VIT-AP UNIVERSITY

Decision Tree in Machine Learning

Dr. Kuppusamy .P

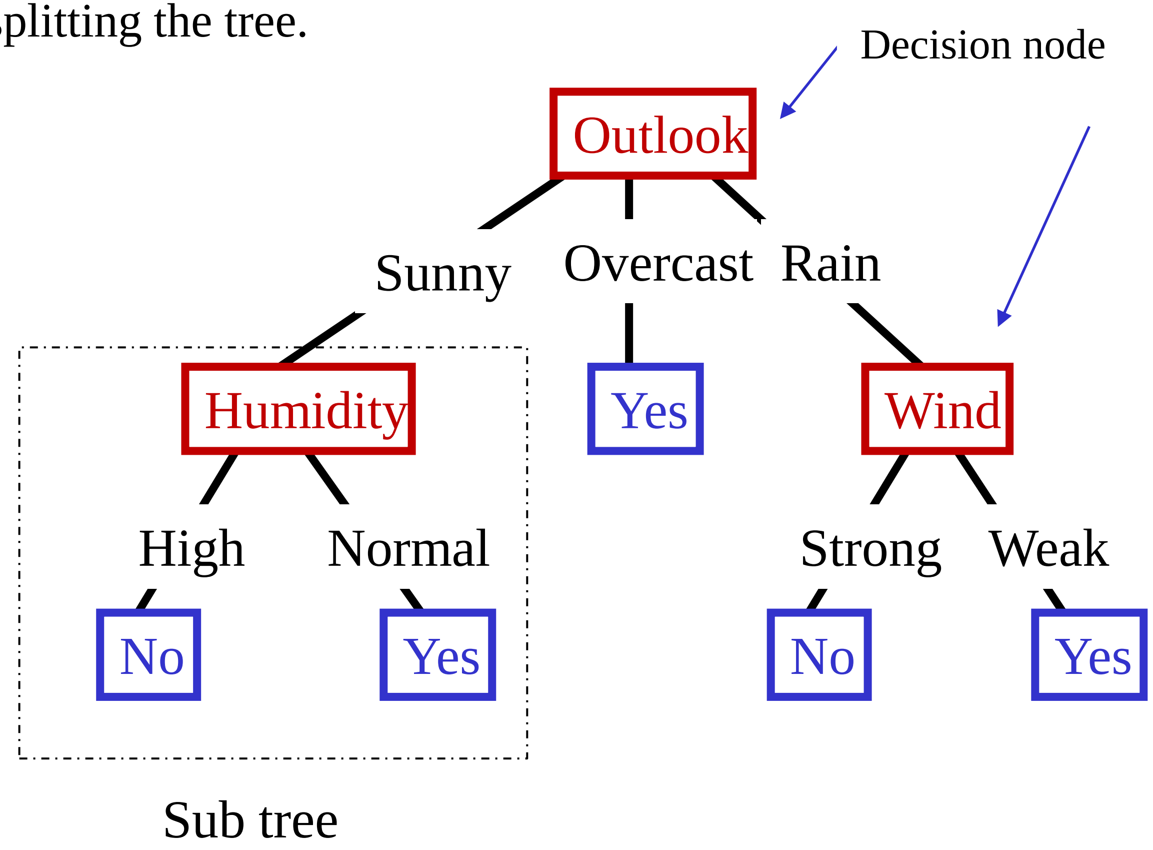
Associate Professor / SCOPE

Decision Tree Learning – Supervised Learning

- Decision tree is a **tree** structure that **makes decision (**classification or regression) by learning **knowledge**.
- It is a graphical representation for getting all the possible solutions/decisions based on given conditions.
- At the end of the learning process, the decisions or test are performed based on features of the given dataset.
  - The decision tree can be thought of as a set sentences (in Disjunctive Normal Form) written propositional logic.
  - It deals both categorical and numerical data.
  - Some characteristics of problems in Decision Tree Learning:
    - Attribute-value paired elements
    - Discrete target function
    - Disjunctive descriptions (of target function)
    - Works well with missing or erroneous training data

Terminologies in Decision Tree

- **Root node** initiates the decision tree that represents the entire dataset D. Then D is divided into two or more homogeneous sets.
- **Internal** (Decision) nodes represent the features of a dataset that make the decision, branches represent the decision rules to make decision, and each leaf node represents the decision (outcome).
- **Splitting** process divides the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree** is formed by splitting the tree.
- **Pruning** process removes the unwanted branches from the tree.



Training Examples

Day	**Outlook**	**Temperature**	**Humidity**	**Wind**	PlayTennis
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

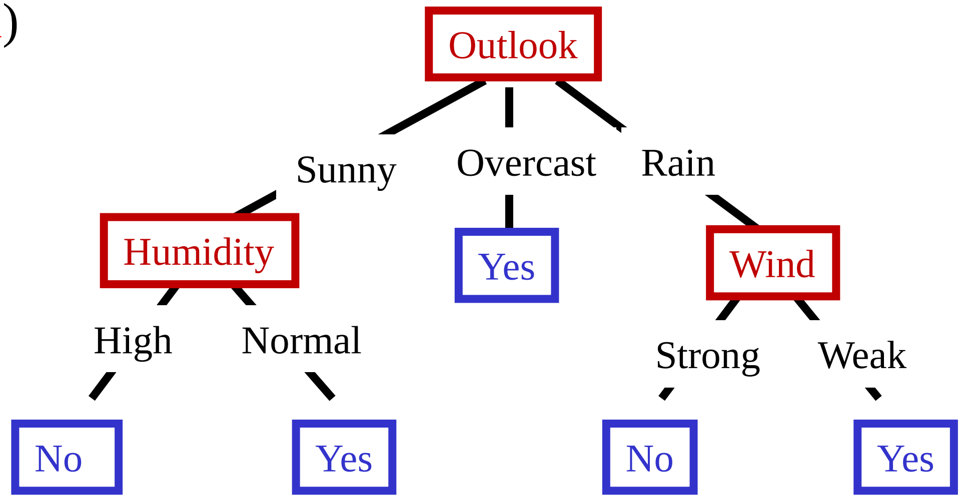
Decision Tree Representation

- Decision trees represent a **disjunction of conjunctions of constraints** on the attribute values of instances.
- Each path from the root to a leaf corresponds to a conjunction of attribute tests, and
- The tree itself is a disjunction of these conjunctions.
- Each node represents a feature, and each link represents a decision.
- Each leaf node represents an outcome (classification)

(Outlook = Sunny ∧ Humidity = Normal)

∨ (Outlook = Overcast)

∨ (Outlook = Rain ∧ Wind = Weak)



Building a Decision Tree

1. First test all attributes and select the **one attribute** that would function as the **best** root.
2. Break-up the training set into **subsets** based on the branches of the root node.
3. Test the **remaining attributes** to check which one fit best underneath the **branches** of the root node;
4. Continue this process for all other branches until
   a. all examples of a subset are of one type
   b. there are no examples left (return majority classification of the parent)
   c. there are **no more** attributes left (default value should be majority classification)

When to Consider Decision Trees?

- Instances are represented by attribute-value pairs.
  - Fixed set of attributes, and the attributes take a small number of disjoint possible values.
- The target function has discrete output values.
  - Decision tree learning is appropriate for a boolean classification, but it easily extends to learning functions with more than two possible output values.
- Disjunctive descriptions may be required.
  - decision trees naturally represent disjunctive expressions.
- The training data may contain errors.
  - Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.
- The training data may contain missing attribute values.
  - Decision tree methods can be used even when some training examples have unknown values.
- Decision tree learning has been applied to problems such as learning to classify
  - medical patients by their disease,
  - equipment malfunctions by their cause, and
  - loan applicants by their likelihood of defaulting on payments.

Attribute (Feature) Selection Measures

1. Information Gain   2. Gini Index

Information gain:

- ***Information gain*** measures how well a given attribute separates the training examples according to their target classification.
- Information Gain refers to the **decline** (changes) in entropy after the dataset is split (**Entropy Reduction)**.
- It calculates how much information gained from a feature about a class.
- Split the node based on information gain value and build the decision tree.
- Decision tree algorithm always attempts to maximize the information gain value.
- Node/attribute contains the highest information gain is split first.

Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Gain(S,A) = **Entropy(S)**- **`I(attribute)`**

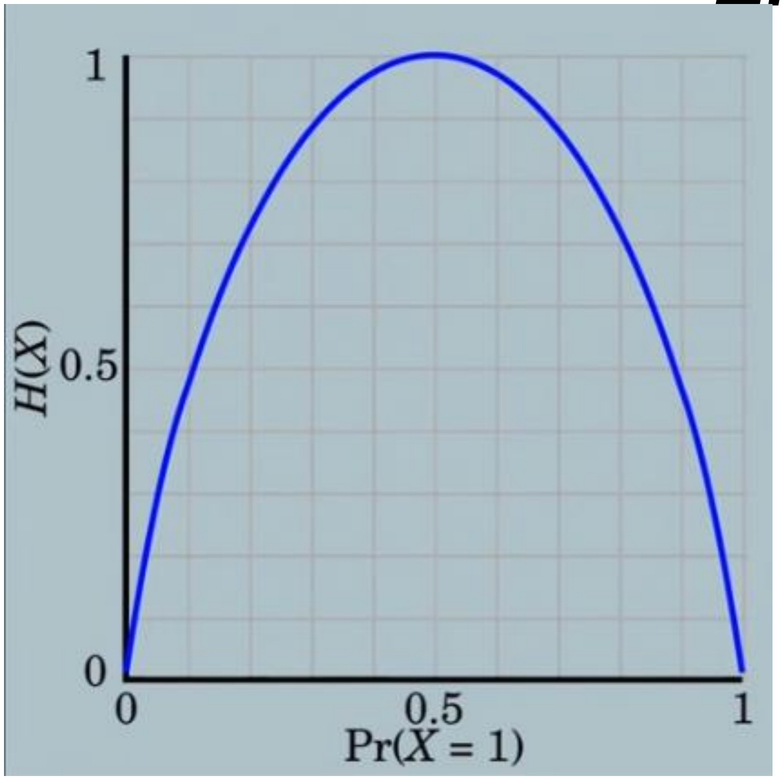
Entropy

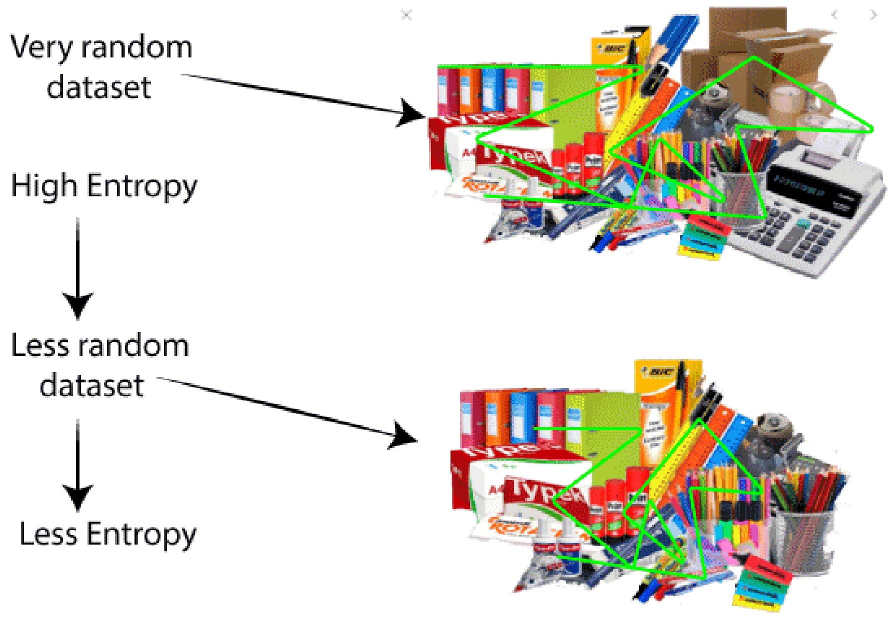
- ***Entropy*** characterizes (measures) the impurity in a given attribute of arbitrary training examples.
- It specifies degree of randomness (uncertainty) in data.
- Entropy values used in splitting i.e., which node is to be split first.
- Given a collection S, containing positive and negative examples of some target concept, the ***entropy of S*** relative to this Boolean classification is:

$$\textbf{Entropy(S)} = -p/(p+q) * \log_2(p/(p+q)) - q/(p+q) * \log_2(q/(p+q))$$

- **S** is a total number of training samples
- **p** is the proportion of positive classes
- **q** is the proportion of negative classes.

Entropy

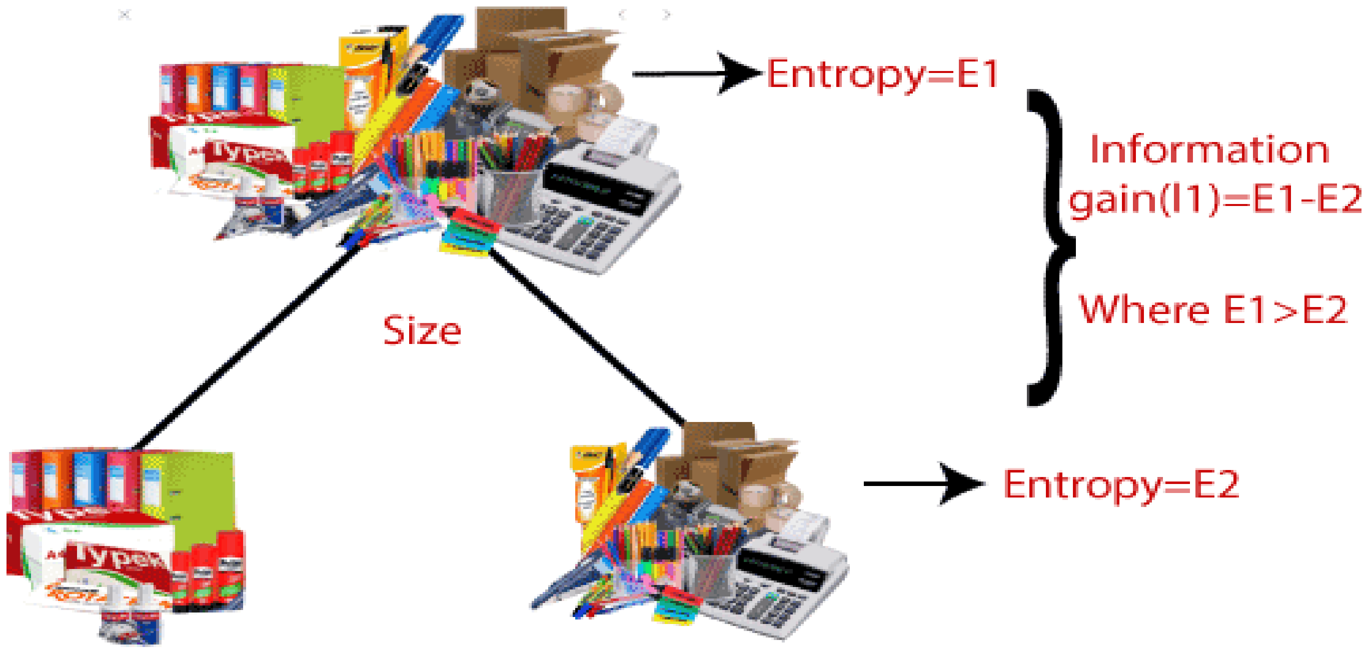




- If data is completely (highly) pure / (highly) impure, randomness is 0.
- If impurity is 0.5, randomness (entropy) is 1.

Decision tree - ID3 algorithm

- **Iterative Dichotomiser 3 (**ID3) algorithm uses this *information gain* measure to select among the candidate attributes at each step that return the highest data gain while growing the tree.



javatpoint.com

Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

- 9 positive instances and 5 negative instance

Examples: Entropy calculation

- **Entropy(S) = -p/(p+q)*log2(p/(p+q)) - q/(p+q)*log2 (q/(p+q))**
- 9 positive instances and 5 negative instances.

- Entropy value range is from 0 to 1.
- Leaf nodes with greater entropy value is considered for further splitting.
- Entropy(S)= Entropy([9+,5-]) = $-(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$

  (94% impure or non-homogeneous)

- In given dataset, If **50% is positive** and **50% is negative** after the splitting, the entropy value is 1 (worst case).

  Entropy([8+,8-] = $-(8/16)\log_2(8/16) - (8/16)\log_2(8/16) = 1.0$

Examples: Entropy calculation

- In a given dataset, **All examples are positive.**

  $\text{Entropy}([8+,0-] = -(8/8)\log_2(8/8) - (0/8)\log_2(0/8) = 0.0$

- In a given dataset, **All examples are Negative.**

  $\text{Entropy}([0+,8-] = -(0/8)\log_2(0/8) - (8/8)\log_2(8/8) = 0.0$

Calculate Average Information Entropy of Attribute:

I(attribute) = $\mathbf{(p_i + q_i)/(p+q)}$ **Entropy(A)**

- pi,$q_{i\,-}$ +ve, -ve values of corresponding attribute (A) possibility value
- p, q – total +ve, -ve values of dataset

ID3 - Algorithm

ID3*(Examples, TargetAttribute, Attributes*)

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = -
- If *Attributes* is empty, Return the single-node tree Root, with label = most common value of *TargetAttribute* in *Examples*
- Otherwise Begin
  – A = The attribute from list of *Attributes* that best classifies *Examples*
  – The decision attribute for *Root* ← A
  – For each possible value, v_i, of attribute A
    - Add a new tree branch below *Root* corresponding to the test A = v_i
    - Let $Examples_{vi}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{vi}$ is empty
      – Then below this new branch add a leaf node with label = most common value of *TargetAttribute* in *Examples*
      – else below this new branch add the subtree
        ID3($Examples_{vi}$, *TargetAttribute*, *Attributes* – {A})
- end
- return *Root*

ID3 - Training Examples

Day	Outlook	Temperature	Humidity	Wind	**PlayTennis**
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

- 9 positive instances and 5 negative instance

Calculate the Entropy for the entire data set

Create a *Root* node for the tree

- Entropy([9+,5-] = – (9/14) $\log_2$(9/14) – (5/14) $\log_2$(5/14) = 0.940

- Select root node from 4 features outlook, temperature, humidity and windy.

Select best decision attribute:

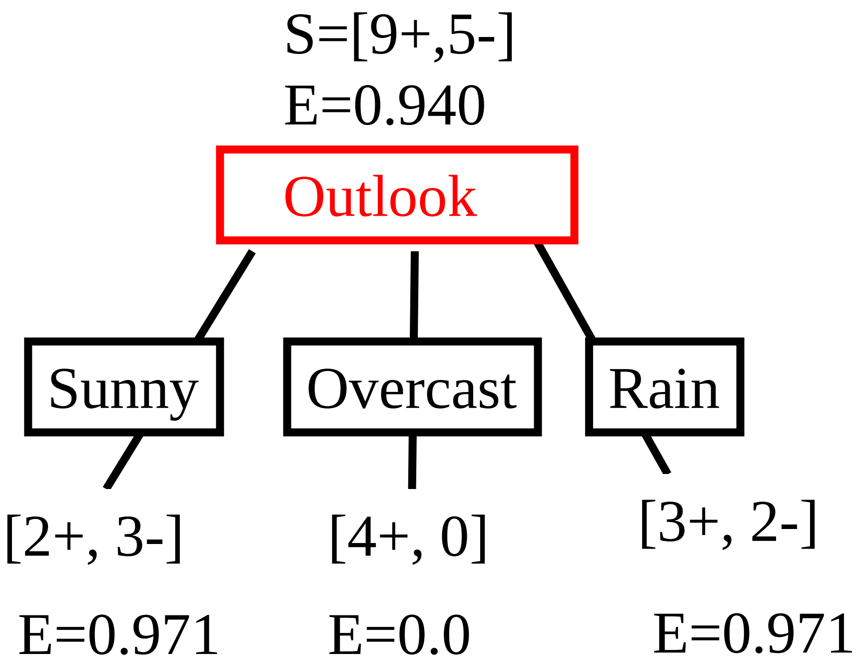
- Let first select feature "Outlook". Outlook contains three possibilities such as sunny, rainy, overcast.
- Hence, calculate entropy for each possibility value of outlook.

Calculate the Entropy for Attribute

Outlook	PlayTennis
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

Outlook	PlayTennis
Rainy	Yes
Rainy	Yes
Rainy	No
Rainy	Yes
Rainy	No

Outlook	PlayTennis
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes



Entropy of Outlook = Sunny

Entropy of Outlook = Overcast

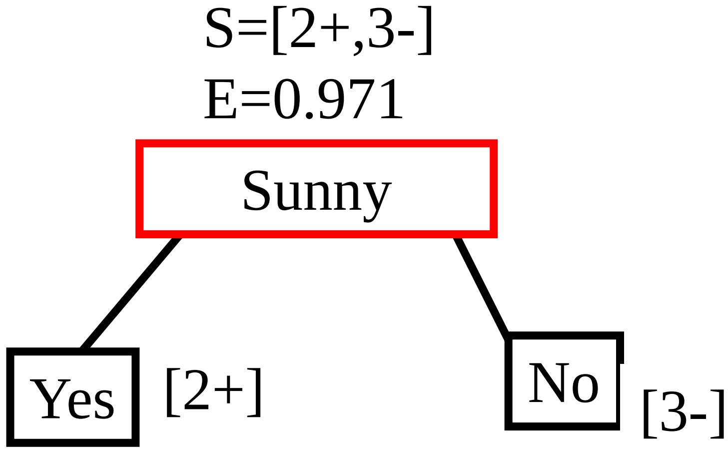
Gain(S,Outlook) =

$$0.940 - [(5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971]$$

= **0.247**

Entropy of Outlook = Rain

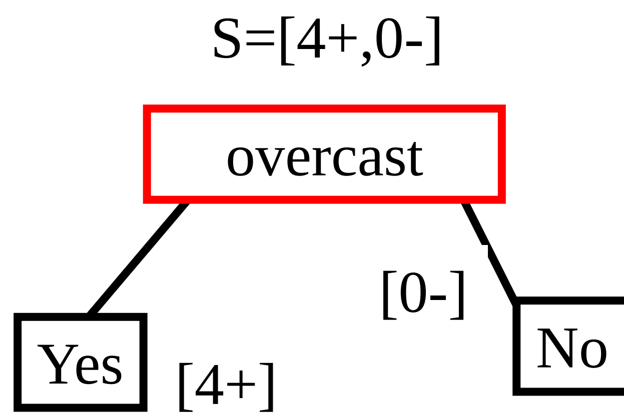
Calculate the Entropy for Attribute



Outlook	PlayTennis
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

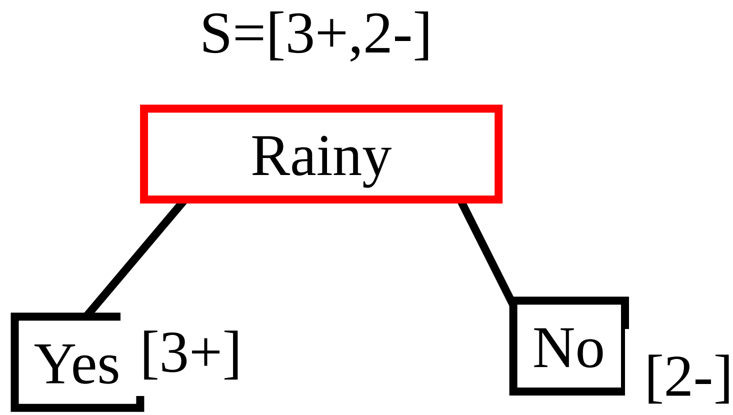
- Entropy([outlook = sunny] = $-(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$

Outlook	PlayTennis
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes



- Entropy([outlook = overcast] = $-(4/4)\log_2(4/4) - (0/4)\log_2(0/4) = 0$

Calculate the Entropy for Attribute



Outlook	PlayTennis
Rainy	Yes
Rainy	Yes
Rainy	No
Rainy	Yes
Rainy	No

- Entropy([outlook = sunny] = $-(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971$

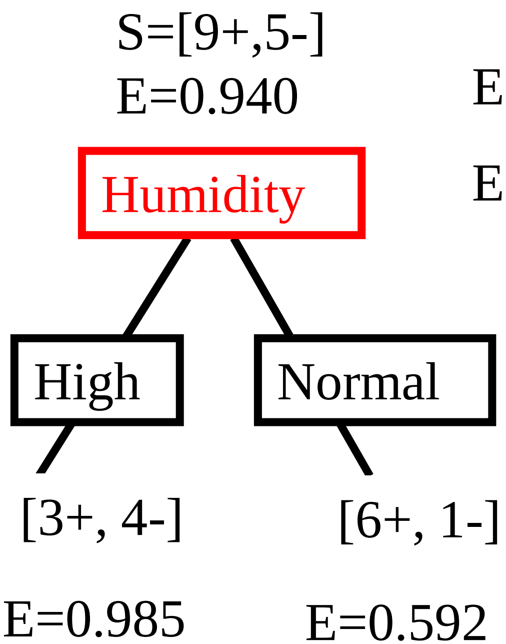
Calculate Average Information Entropy

- I(outlook) = $(P_{sunny} + N_{sunny})/(p+n)$ (Entropy (outlook = sunny)) +
  $(P_{overcast} + N_{overcast})/(p+n)$ (Entropy (outlook = overcast))+
  $(P_{rainy} + N_{rainy})/(p+n)$ (Entropy (outlook = rainy))

- I(outlook) = $(2+3)/(9+5) * 0.971 + (4+0)/((9+5) *0 +$
  $(3+2)/(9+5) * 0.971 =$ **0.693**
- **Info Gain(**S,Outlook) = Entropy (S) – I(Outlook)
- **Info Gain(**S,Outlook) = 0.940 - 0.693
  = **0.247**

Entropy for Humidity

$E(S) = E([9+,5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$



$E(\text{Humidity}=\textbf{High}) = -(3/7)\log_2(3/7) - (4/7)\log_2(4/7) = 0.985$

$E(\text{Humidity}=\textbf{Normal}) = -(6/7)\log_2(6/7) - (1/7)\log_2(1/7) = 0.592$

Info. Gain(S, **Humidity)** = E(S) – I(Humidity)

= 0.940- [(7/14)*0.985+(7/14)*0.592]

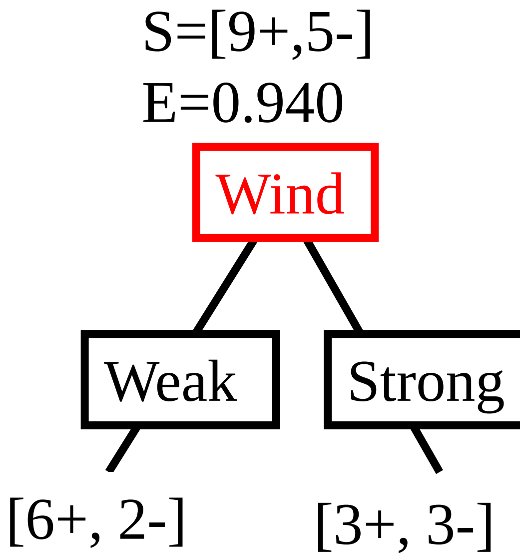
= **0.151**

Entropy for wind

$E(S) = E([9+,5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$

$E(\text{Wind}=\textbf{Weak}) = -(6/8)\log_2(6/8) - (2/8)\log_2(2/8) = 0.811$

$E(\text{Wind}=\textbf{Strong}) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$

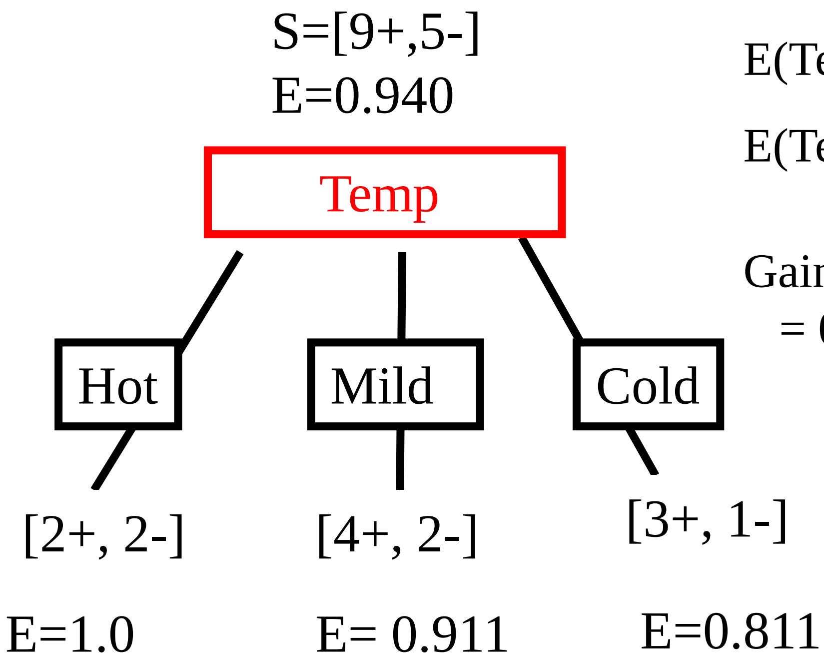


$\text{Gain}(S,\text{Wind}) = E(S) - I(\text{Wind})$

$= 0.940 - [(8/14)*0.811+(6/14)*1.0]$

$= \mathbf{0.048}$

Entropy for Temperature



E(Temp=**Hot**) = – (2/4) $\log_2$(2/4) – (2/4) $\log_2$(2/4) = 1

E(Temp=**Mild**) = – (4/6) $\log_2$(4/6) – (4/6) $\log_2$(4/6) = 0.911

E(Temp=**Cold**) = – (3/4) $\log_2$(3/4) – (1/4) $\log_2$(1/4) = 0.811

Gain(S,Temperature) = E(S) – I (Temp)
= 0.940- [(4/14)*1.0 + (6/14)*0.911+ (4/14)*0.811]
= **0.029**

Information Gain of 4 Attributes:

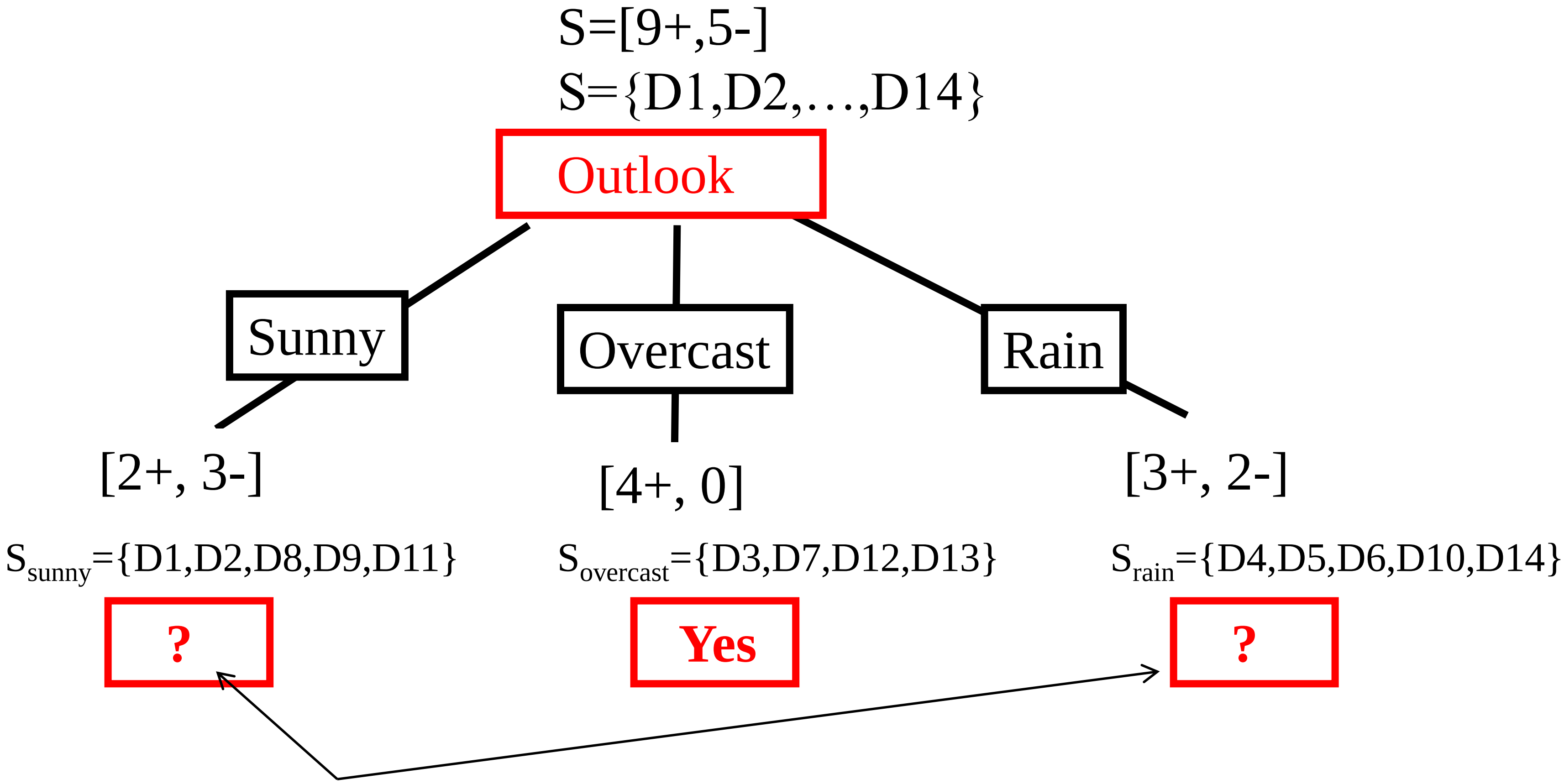
- Info Gain(S,Outlook) = 0.247
- Info. Gain(S, Humidity) = 0.151
- Info. Gain(S,Wind) = 0.048
- Info. Gain(S,Temperature) = 0.029

Select the attribute which contains Max Information Gain i.e. Info Gain(S,**Outlook**) = 0.247

Best Attribute - Outlook

Algorithm:

1. First test all attributes and select the **one attribute** that would function as the **best** root.
2. Break-up the training set into **subsets** based on the branches of the root node.

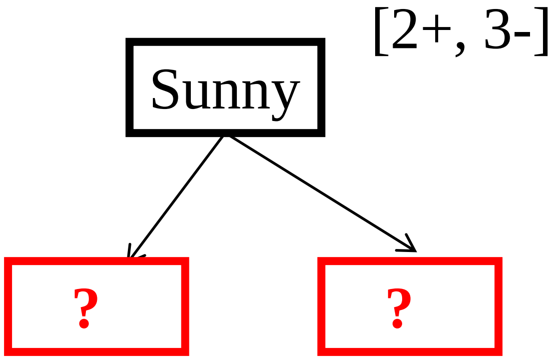


Which attribute should be tested here?

Repeat the same process for the sub-trees till get the tree

Outlook	Temp	Humidity	Windy	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

S_{sunny}={D1,D2,D8,D9,D11}



If Outlook = **Sunny**

p= 2 and n=3

Entropy(Sunny) = -$2/(2+3)\ \log_2(2/(2+3)) - 3/(2+3)\log_2(3/(2+3))$

= 0.970

Calculate the entropy value for Humidity

Outlook	Humidity	Play Tennis
Sunny	High	No
Sunny	High	No
Sunny	High	No
Sunny	Normal	Yes
Sunny	Normal	Yes

Humidity	p	n	Entropy
High	0	3	0
Normal	2	0	0

E(Humidity=**High**) = – (0/3) $\log_2$(0/3) – (3/3) $\log_2$(3/3) = 0

E(Humidity=**Normal**) = – (2/2) $\log_2$(2/2) – (0/2) $\log_2$(0/2) = 0

Average Info. Entropy (Humidity) = (3/5) * 0.0 + 2/5 *(0.0) = 0

Gain(S_{sunny} , Humidity) = 0.970 - 0 = **0.97**

Calculate the entropy value for each Windy

Outlook	Windy	Play Tennis
Sunny	Weak	No
Sunny	Strong	No
Sunny	Weak	No
Sunny	Weak	Yes
Sunny	Strong	Yes

Windy	p	n	Entropy
Strong	**1**	**1**	**1**
Weak	**1**	**2**	**0.8962**

E(Windy=**Strong**) = – (1/2) $\log_2$(1/2) – (1/2) $\log_2$(1/2) = 1

E(Windy=**Weak**) = – (1/3) $\log_2$(1/3) – (2/3) $\log_2$(2/3) = - 0.3 (-1.585) – 0.6 * (-0.5851)

= 0.4755 + 0.3516 = 0.8271

Average Info. Entropy (Windy) = (2/5)1.0 + 3/5(0.8271) = 0.4+0.4962

= 0.8962

Gain(S_{sunny} , Windy) = 0.970 - 0.8962 = **0.0738**

Calculate the entropy value for Temperature

Outlook	Temp	Play Tennis
Sunny	Hot	No
Sunny	Hot	No
Sunny	Mild	No
Sunny	Cool	Yes
Sunny	Mild	Yes

Temp	p	n	Entropy
Hot	0	2	0
Mild	1	1	1
Cool	1	0	0

Average Info. Entropy (Temp) = (2/5)0.0 +2/5(1.0) + (1/5)0.0 = 0.4

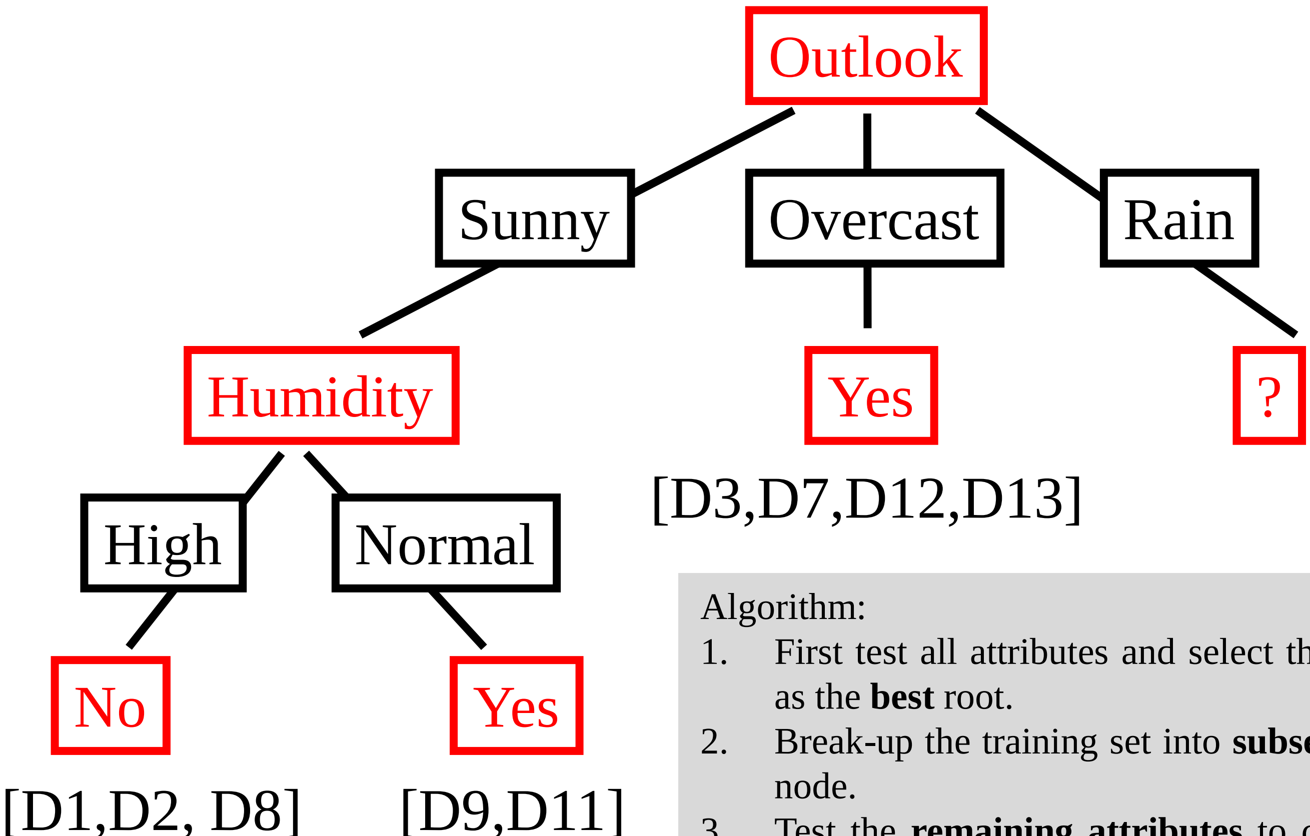
Gain(S_{sunny} , Temp) = 0.970 - 0.4 = **0.571**

Best Attribute:

- Gain(S_{sunny} , Humidity) = 0.970 - 0 = **0.970**
- Gain(S_{sunny} , Wind) = 0.970 - 0.951 = **0.0738**
- Gain(S_{sunny} , Temp) = 0.970 - 0.4 = **0.571**

So, **Humidity** is selected as best attribute.

ID3 - Result



Algorithm:

1. First test all attributes and select the **one attribute** that would function as the **best** root.
2. Break-up the training set into **subsets** based on the branches of the root node.
3. Test the **remaining attributes** to check which one fit best underneath the **branches** of the root node;
4. Continue this process for all other branches until
   a. all examples of a subset are of one type
   b. there are no examples left (return majority classification of the parent)
   c. there are **no more** attributes left (default value should be majority classification)

Repeat the same process for the sub-trees till get the tree

Outlook	**Temp**	**Humidity**	**Windy**	**Play Tennis**
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Rain	Mild	Normal	Weak	Yes
Rain	Mild	High	Strong	No

If Outlook = **Rain**

p= 3 and n=2

Entropy(Rain) = $-3/(3+2)\log_2(3/(3+2)) - 2/(3+2)\log_2(2/(3+2))$

$$= -0.6 * (-0.737) - 0.4 * (-1.322)$$

$$= 0.4422 + 0.5288 = 0.971$$

Calculate the entropy value for Humidity

Outlook	Humidity	Play Tennis
Rain	**High**	**Yes**
Rain	**High**	**No**
Rain	**Normal**	**Yes**
Rain	**Normal**	**No**
Rain	**Normal**	**Yes**

Humidity	p	n	Entropy
High	**1**	**1**	**1**
Normal	**2**	**1**	**0.8271**

E(Humidity=**High**) = – (1/2) $\log_2(1/2)$ – (1/2) $\log_2(1/2)$ = 1

E(Humidity=**Normal**) = – (2/3) $\log_2(2/3)$ – (1/3) $\log_2(1/3)$ = – 0.6 * (-0.5851) - 0.3 (-1.585)

= 0.3516 + 0.4755 = 0.8271

Average Info. Entropy (Humidity) = 2/5(1) + (3/5) * 0.8271 = 0.4 + 0.6 (0.8271)

= 0.4+0.496 = 0.896

Gain(S_{rain} , Humidity) = Entropy(S_{rain}) – ($I_{humidity}$)

= 0.971- 0.896= 0.075

Calculate the entropy value for Temperature

Outlook	Temp	Play Tennis
Rain	Mild	Yes
Rain	Cool	Yes
Rain	Cool	No
Rain	Mild	Yes
Rain	Mild	No

Temp	p	n	Entropy
Hot	**0**	**0**	**0**
Mild	**2**	**1**	**0.8271**
Cool	**1**	**1**	**1**

E(Temp=**Hot**) = 0

E(Temp=Mild) = – (2/3) $\log_2(2/3)$ – (1/3) $\log_2(1/3)$ = – 0.6 * (-0.5851) - 0.3 (-1.585)

= 0.3516 + 0.4755 = 0.8271

E(Temp=**Cool**) = – (1/2) $\log_2(1/2)$ – (1/2) $\log_2(1/2)$ = 1

Average Info. Entropy (Temp) = (0/5) * 0 + (3/5) * 0.8271 +(2/5) * 1

=0+ 0.496+0.4 = 0.896

Gain(S_{rain} , Humidity) = Entropy(S_{rain}) – ($I_{humidity}$)

= 0.971- 0.896= 0.075

Calculate the entropy value for Windy

Outlook	Windy	Play Tennis
Rain	Weak	Yes
Rain	Weak	Yes
Rain	Strong	No
Rain	Weak	Yes
Rain	Strong	No

Windy	p	n	Entropy
Strong	**0**	**2**	**0**
Weak	**3**	**0**	**0**

$$\text{Gain}(S_{rain}, \text{Wind}) = \text{Entropy}(S_{rain}) - (I_{windy})$$

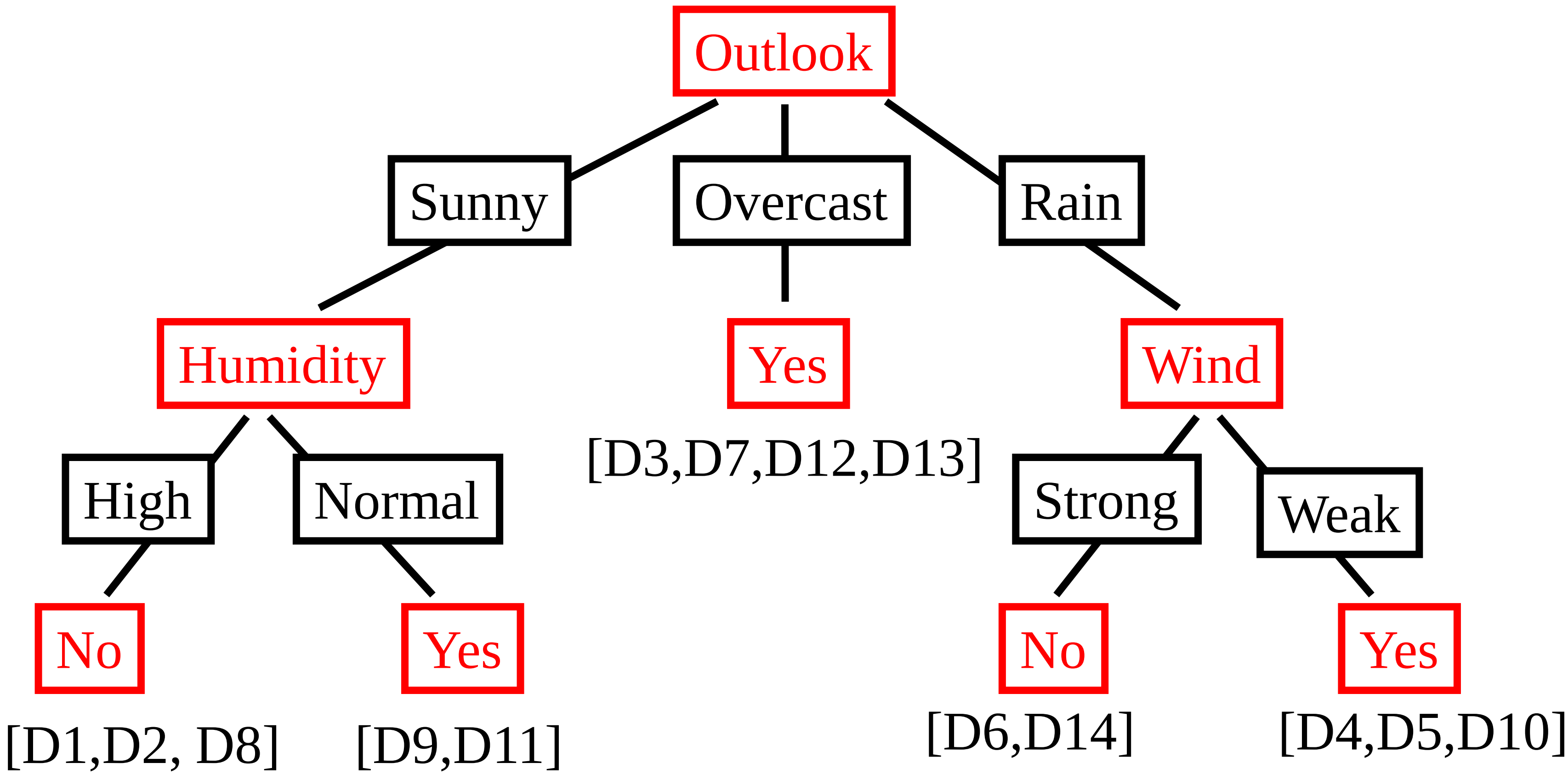
$$= 0.971 - 0 = \mathbf{0.971}$$

Best Attribute

- $\text{Gain}(S_{rain}, \text{Humidity}) = 0.075$
- $\text{Gain}(S_{rain}, \text{Temp.}) = 0.075$
- $\text{Gain}(S_{rain}, \text{Wind}) = \mathbf{0.971}$

So, **Wind** will be selected

ID3 - Result



34

VIT-AP UNIVERSITY

References

1. Tom M. Mitchell, Machine Learning, McGraw Hill , 2017.
2. EthemAlpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2017.
3. Wikipedia