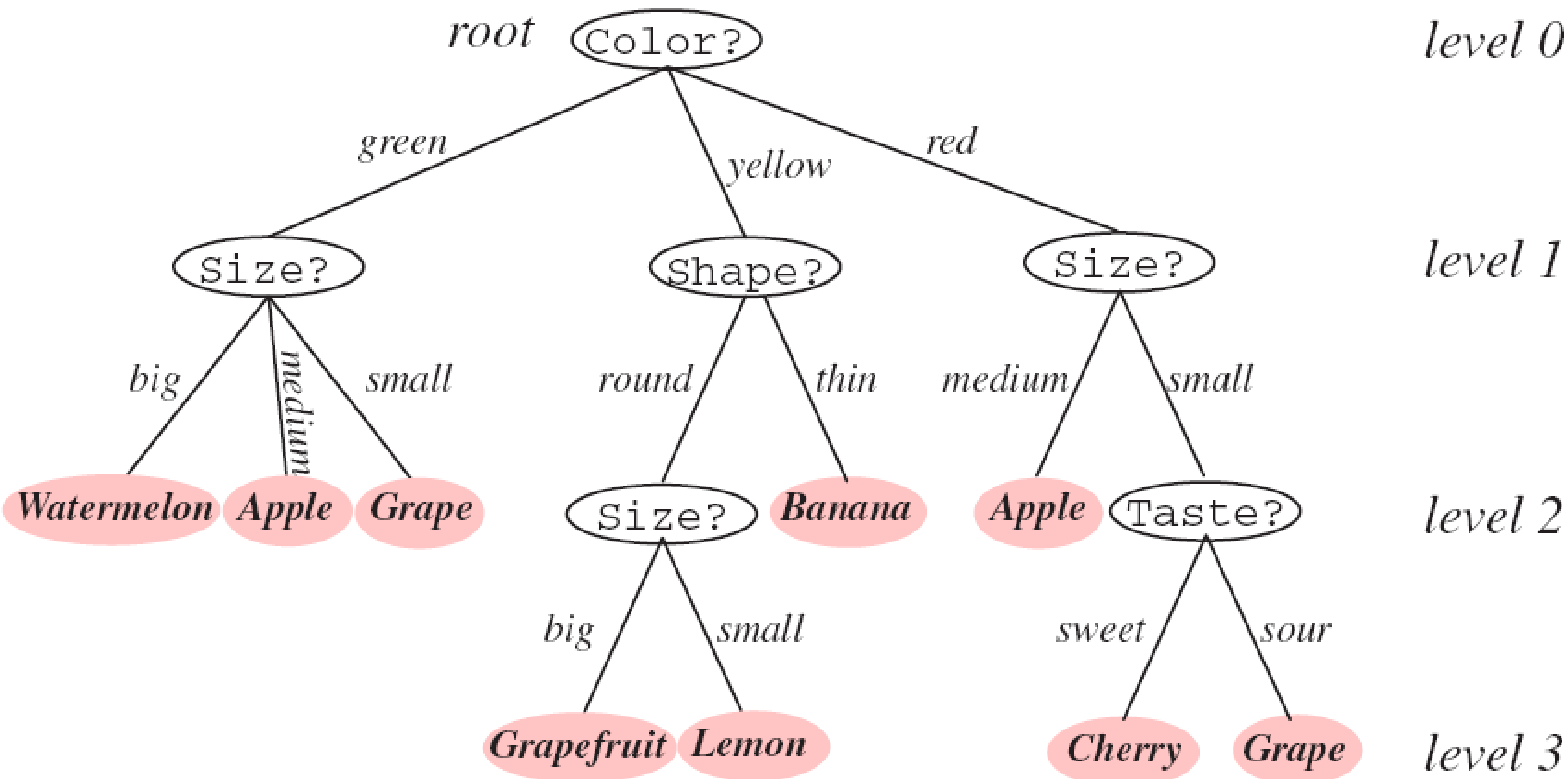
Decision Tree Learning

Introduction

- Decision tree learning is one of the most widely used and practical methods for inductive inference.
- It is a method for approximating discrete-valued functions, in which the learned function is represented by a decision tree.
- It is robust to noisy data and capable of learning disjunctive expressions.
- decision tree learning algorithms that includes widely used algorithms such as ID3, CART, and **C4.5.**
- Learned trees can also be re-represented as sets of ***if-then rules*** to improve human readability.
- Decision tree algorithms transform raw data to rule based decision making trees.

DECISION TREE REPRESENTATION



- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

APPROPRIATE PROBLEMS FOR DECISION TREE LEARNING

- ***Instances are represented by attribute-value pairs.*** Instances are described by a fixed set of attributes (e.g., ***Temperature)*** and their values (e.g., ***Hot, Mild, Cold)***
- ***The target function has discrete output values.*** assigns a Boolean classification (e.g., ***yes*** or ***no)*** to each example.
- ***Disjunctive descriptions may be required.***
- ***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.

Example

Examples:

- Equipment or medical diagnosis
- Credit risk analysis
- Modeling calendar scheduling preferences

Different Algorithms for Decision Tree

- ID3
- CART
- C4.5

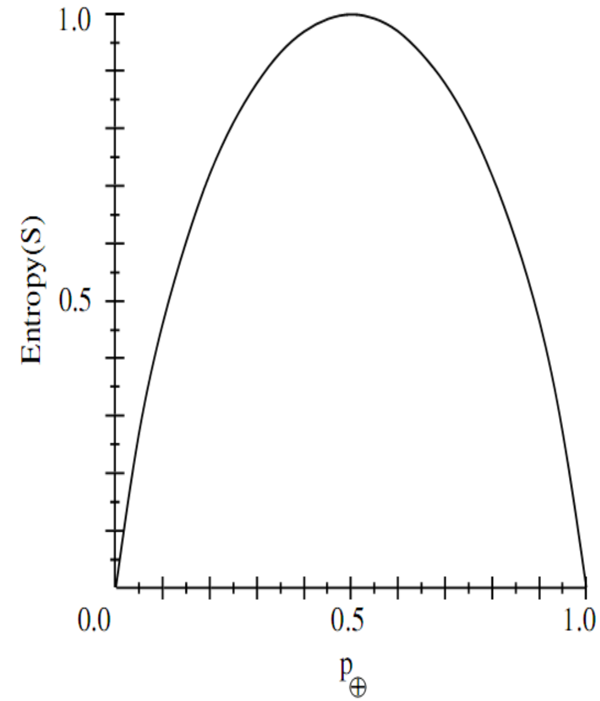
ID3 Algorithm

- ID3 is one of the most common decision tree algorithm.
- Firstly, It was introduced in 1986 and it is acronym of **Iterative Dichotomiser**.
- Dichotomisation means dividing into two completely opposite things.
- So, the algorithm iteratively divides attributes into two groups which are the most dominant attribute and others to construct a tree.
- Then, it calculates the entropy and information gains of each attribute.
- In this way, the most dominant attribute can be founded.
- After then, the most dominant one is put on the tree as decision node. Thereafter, entropy and gain scores would be calculated again among the other attributes.
- Thus, the next most dominant attribute is found.
- Finally, this procedure continues until reaching a decision for that branch.
- That’s why, it is called Iterative Dichotomiser.

Entropy

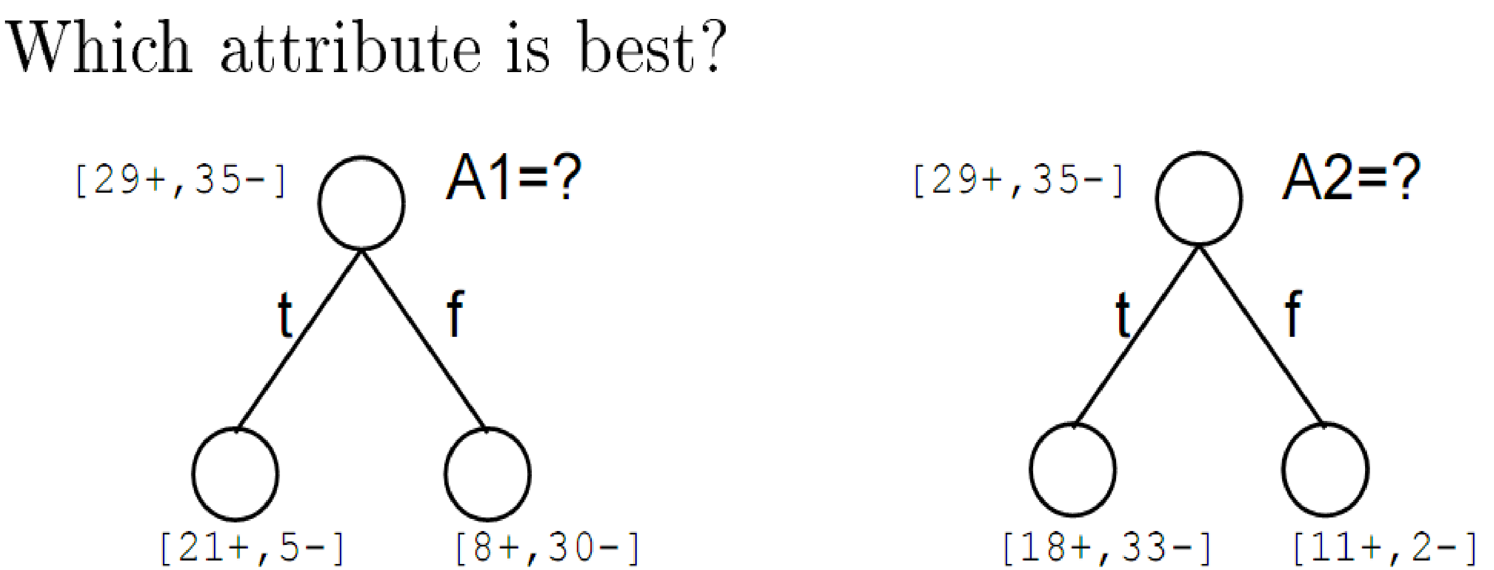
- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$Entropy(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$



Which Attribute Is the Best Classifier?

- What is a good quantitative measure of the worth of an attribute?
- We will define a statistical property, called ***information gain.***



Entropy and Information Gain

- $Entropy(S) = \sum - p(I) \,.\, \log_2 p(I)$
- $Gain(S, A) = Entropy(S) - \sum [\, p(S|A) \,.\, Entropy(S|A) \,]$

Data set

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Entropy

- We need to calculate the entropy first.
- Decision column consists of 14 instances and includes two labels: yes and no.
- There are 9 decisions labeled yes, and 5 decisions labeled no.
- Entropy(Decision)

$$= -\, p(Yes) \,.\, \log_2 p(Yes) - p(No) \,.\, \log_2 p(No)$$

$$= -\,(9/14) \,.\, \log_2(9/14) - (5/14) \,.\, \log_2(5/14)$$

$$= 0.940$$

- Now, we need to find the most dominant factor for decision.

Wind factor on decision

- Gain(Decision, Wind) = Entropy(Decision) – ∑ [p(Decision|Wind) . Entropy(Decision|Wind)]
- Wind attribute has two labels: weak and strong.
- Now, we need to calculate (Decision|Wind=Weak) and (Decision|Wind=Strong) respectively.

Weak wind factor on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes

- . There are 8 instances for weak wind. Decision of 2 items are no and 6 items are yes.

- $\text{Entropy(Decision|Wind=Weak)} = – p(No) . log_2p(No) – p(Yes) . log_2p(Yes)$
- $\text{Entropy(Decision|Wind=Weak)} = – (2/8) . log_2(2/8) – (6/8) . log_2(6/8) = 0.811$

Strong wind factor on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
2	Sunny	Hot	High	Strong	No
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rain	Mild	High	Strong	No

- Here, there are 6 instances for strong wind. Decision is divided into two equal parts.
- Entropy(Decision|Wind=Strong) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)
- Entropy(Decision|Wind=Strong) = – (3/6) . $\log_2$(3/6) – (3/6) . $\log_2$(3/6) = 1

- Now compute Gain(Decision, Wind)
- Gain(Decision, Wind) =

Entropy(Decision) – [p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak)]

– [p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong)]

=0.940 – [(8/14) . 0.811] – [(6/14). 1] = 0.048

- Calculations for wind column is over. Now, we need to apply same calculations for other columns to find the most dominant factor on decision.

Outlook Factor on decision

- outlook attribute has three labels: sunny, rain and overcast.
- Now, we need to calculate (Decision|outlook=sunny) , (Decision|outlook=overcast) and (Decision|outlook=rain).

Overcast outlook on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

- Here, there are 4 instances for outlook=overcast. Decision is consisting only yes.
- Entropy(Decision|outlook=overcast) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)
- Entropy(Decision|outlook=overcast) = – (0/4) . $\log_2$(0/4) – (4/4) . $\log_2$(4/4) = 0
- If Entropy is zero, that means data is belonging to single class.
- We can analyse data from table as well, that all examples are falling in 'yes' class.

Sunny outlook on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

- Here, there are 5 instances for outlook=sunny. Decision of 3 items are No and decision of 2 items are yes.
- Entropy(Decision|outlook=sunny) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)
- Entropy(Decision|outlook=sunny) = – (3/5) . $\log_2$(3/5) – (2/5) . $\log_2$(2/5) = 0.971

Rain outlook on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

- Here, there are 5 instances for outlook=rain. Decision of 2 items are No and decision of 3 items are yes.
- Entropy(Decision|outlook=rain) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)
- Entropy(Decision|outlook=rain) = – (2/5) . $\log_2$(2/5) – (3/5) . $\log_2$(3/5) = 0.971

- Now compute Gain(Decision, outlook)
- Gain(Decision, outlook) =

Entropy(Decision)

– [p(Decision|outlook=overcast) . Entropy(Decision|outlook=overcast)]

– [p(Decision|outlook=sunny) . Entropy(Decision|outlook=sunny)]

- [p(Decision|outlook=rain) . Entropy(Decision|outlook=rain)]

=0.940 – [(4/14) . 0] – [(5/14). 0.971] - [(5/14). 0.971] = 0.246

Exercise

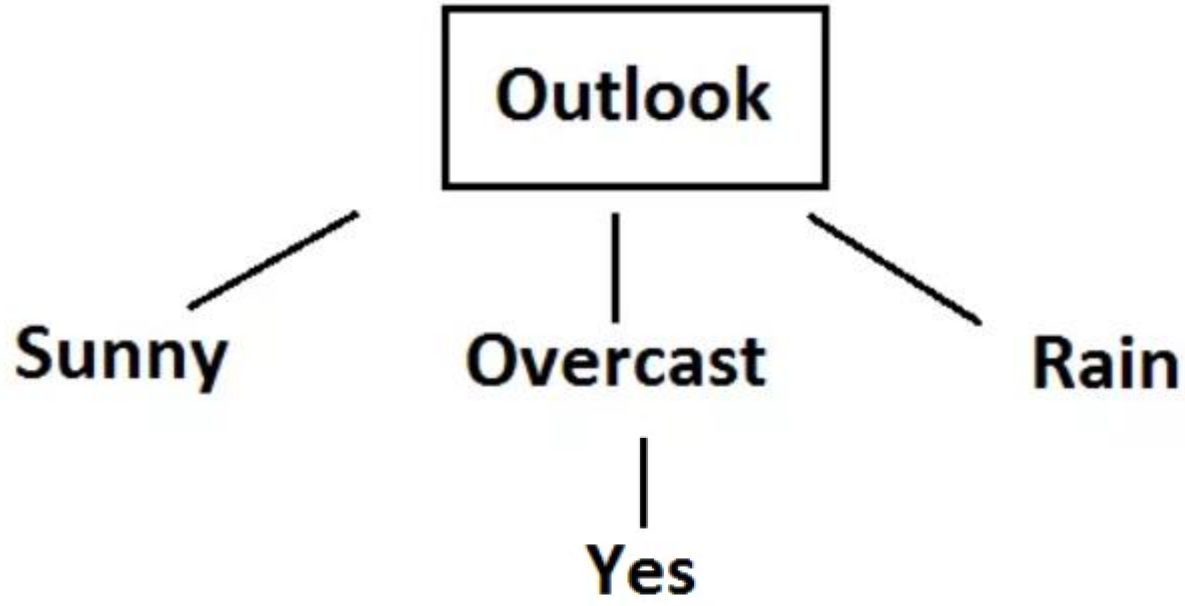
- Similarly, compute gain for humidity and temperature

Information Gain of All Attributes

- 1- Gain(Decision, Outlook) = 0.246
- 2- Gain(Decision, Temperature) = 0.029
- 3- Gain(Decision, Humidity) = 0.151
- 4-Gain(Decision, Wind) = 0.048

- As seen, outlook factor on decision produces the highest score.
- That’s why, outlook decision will appear in the root node of the tree.

So, Root node of Decision Tree



Second Level (Outlook = sunny)

- Now, we need to test dataset for custom subsets of outlook attribute.
- 1- Gain(Outlook=Sunny|Temperature) = 0.570
- 2- Gain(Outlook=Sunny|Humidity) = 0.970
- 3- Gain(Outlook=Sunny|Wind) = 0.019

- Now, humidity is the decision because it produces the highest score if outlook were sunny.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

- At this point, decision will always be no if humidity were high.

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

- On the other hand, decision will always be yes if humidity were normal

Second Level (Outlook = Rain)

- 1- Gain(Outlook=Rain | Temperature) = 0.01997309402197489
- 2- Gain(Outlook=Rain | Humidity) = 0.01997309402197489
- 3- Gain(Outlook=Rain | Wind) = 0.9709505944546686

- Here, wind produces the highest score if outlook were rain.
- That’s why, we need to check wind attribute in 2nd level if outlook were rain.

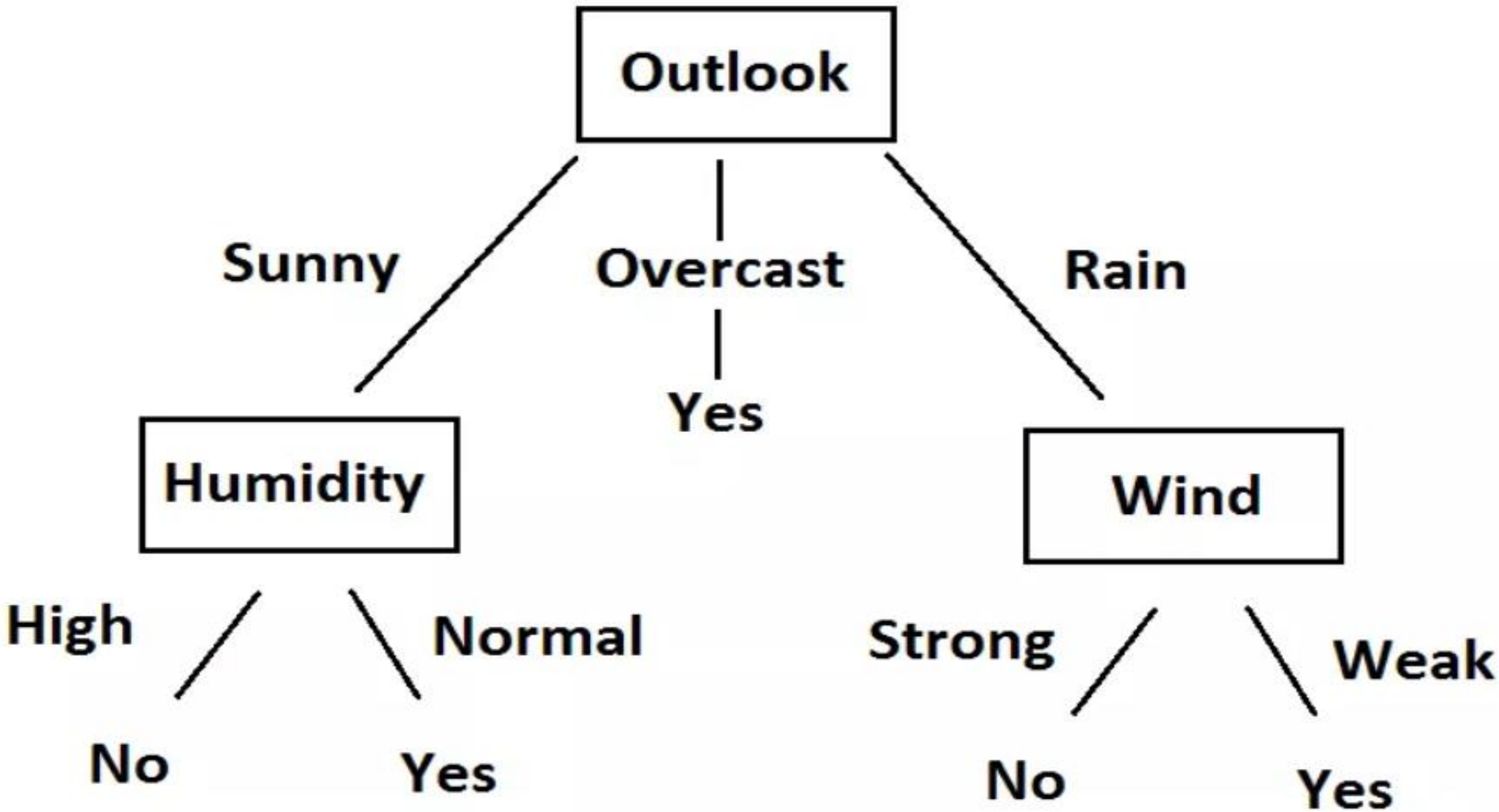
Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

- So, it is revealed that decision will always be yes if wind were weak and outlook were rain.

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

- Decision will be always no if wind were strong and outlook were rain.

Final Decision Tree



Thank You !!!