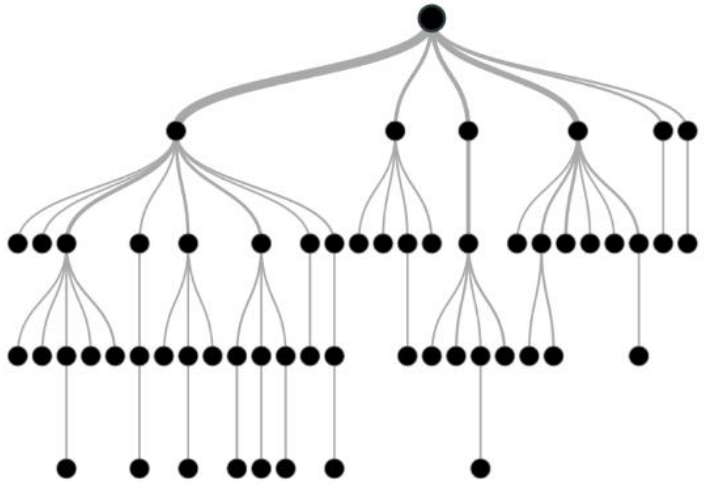
Decision Tree

What is decision Tree?

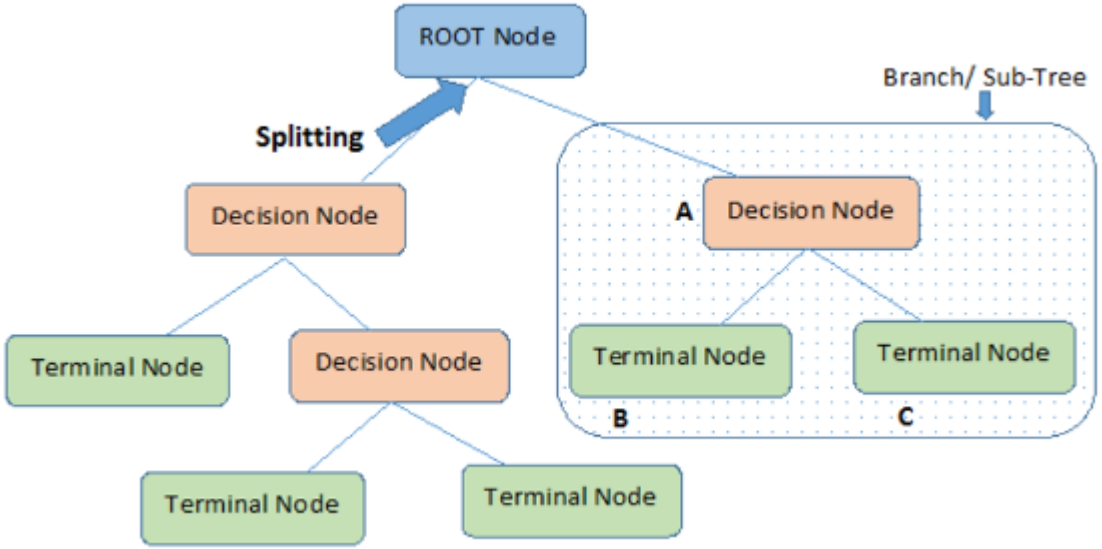
- Decision tree is a popular machine learning algorithm that can be used for both regression and classification tasks.
- It is easy to understand, interpret, and implement.
- A decision tree is a hierarchical model used in decision support that depicts decisions and their potential outcomes.



Decision Tree Terminologies

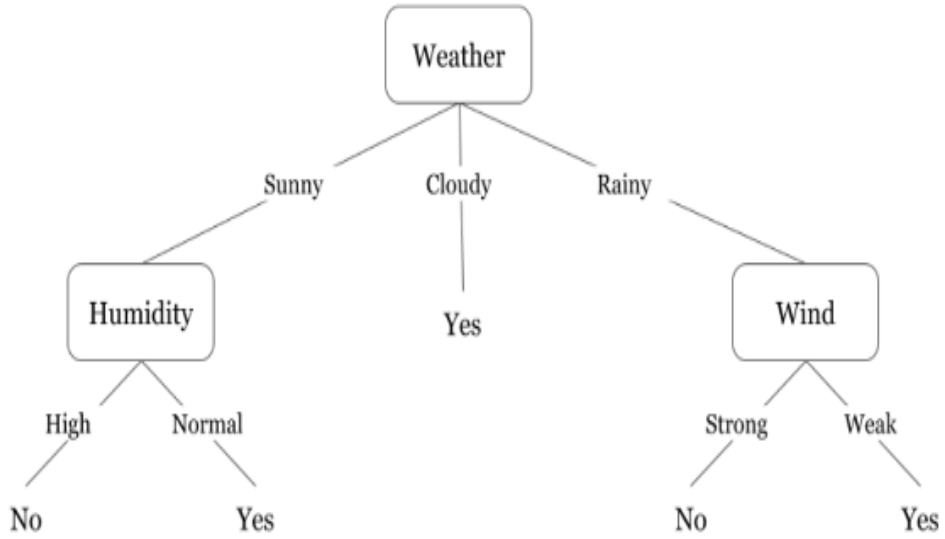
- ***Root Node****:* The initial node at the beginning of a decision tree, where the entire population or dataset starts dividing based on various features or conditions.
- **Internal Nodes (Decision Nodes)** *:* Nodes resulting from the splitting of root nodes are known as decision nodes. These nodes represent intermediate decisions or conditions within the tree.
- ***Leaf Nodes****:* Nodes where further splitting is not possible, often indicating the final classification or outcome. Leaf nodes are also referred to as terminal nodes.

- ***Pruning***: The process of removing or cutting down specific nodes in a decision tree to prevent overfitting and simplify the model.
- **Splitting**: The process of dividing a node into two or more sub-nodes based on a decision criterion.



Example of a Decision Tree

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No



Working Principle of Decision Tree

- **Starting at the Root**: The algorithm begins at the top, called the “root node,” representing the entire dataset.
- **Asking the Best Questions:** It looks for the most important feature or question that splits the data into the most distinct groups.
- **Branching Out**: Based on the answer to that question, it divides the data into smaller subsets, creating new branches.
- **Repeating the Process**: The algorithm continues asking questions and splitting the data at each branch until it reaches the final “leaf nodes,” representing the predicted outcomes or classifications.

Entropy

- The entropy of any random variable or random process is the average level of uncertainty involved in the possible outcome of the variable or process.
- To understand it more let’s take an example of a coin flip.
- Where we have only two probabilities either it will be a tail, or it will be a head and if the probability of tail after flip is p then the probability of a head is 1-p. And the maximum uncertainty is for p = ½ when there is no reason to expect one outcome over another.

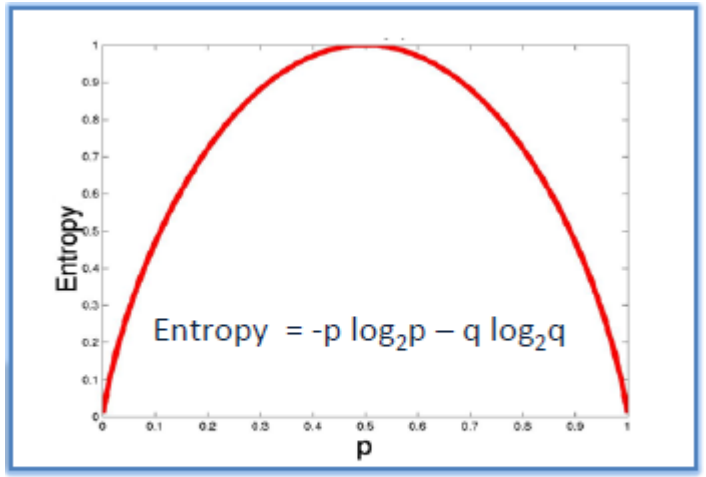
Measure of Entropy

- Maximum entropy value is 1 and it is measured as

H(X)=-$\sum_{i=1}^{n} P(x_i) log P(x_i)$

Where

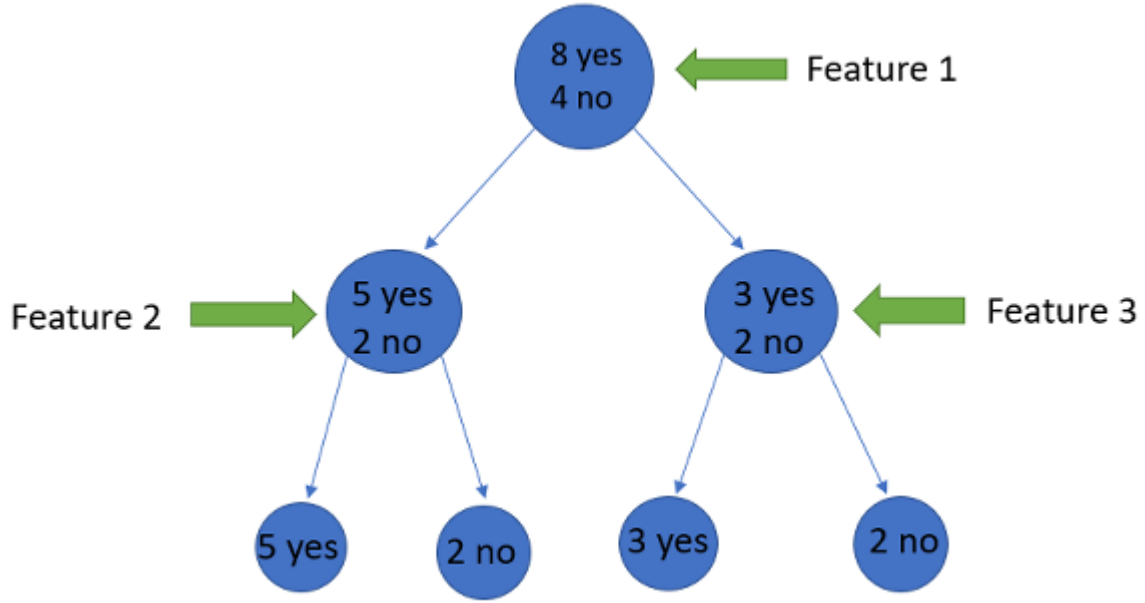
- X = random variable or process
- x_i = possible outcomes
- n= Total possible outcomes
- P(x_i) = probability of possible outcomes.



Entropy = -0.5 $\log_2$0.5 – 0.5 $\log_2$0.5 = 1

How do Decision Trees use Entropy?

- Entropy basically measures the impurity of a node. Impurity is the degree of randomness; it tells how random our data is. A **pure sub-split** means that either you should be getting “yes”, or you should be getting “no”.
- Suppose a *feature* has 8 “yes” and 4 “no” initially, after the first split the left node *gets 5 ‘yes’ and 2 ‘no’* whereas right node *gets 3 ‘yes’ and 2 ‘no’.*



For Feature 2

$$\Rightarrow -\left(\frac{5}{7}\right)log_2\left(\frac{5}{7}\right) - \left(\frac{2}{7}\right)log_2\left(\frac{2}{7}\right)$$

$$\Rightarrow -(0.71 * -0.49) - (0.28 * -1.83)$$

$$\Rightarrow -(-0.34) - (-0.51)$$

$$\Rightarrow 0.34 + 0.51$$

$$\Rightarrow 0.85$$

For Feature 3

$$\Rightarrow -\left(\frac{3}{5}\right)log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right)log_2\left(\frac{2}{5}\right)$$

$$\Rightarrow -(0.6 * -0.73) - (0.4 * -1.32)$$

$$\Rightarrow -(-0.438) - (-0.528)$$

$$\Rightarrow 0.438 + 0.528$$

$$\Rightarrow 0.966$$

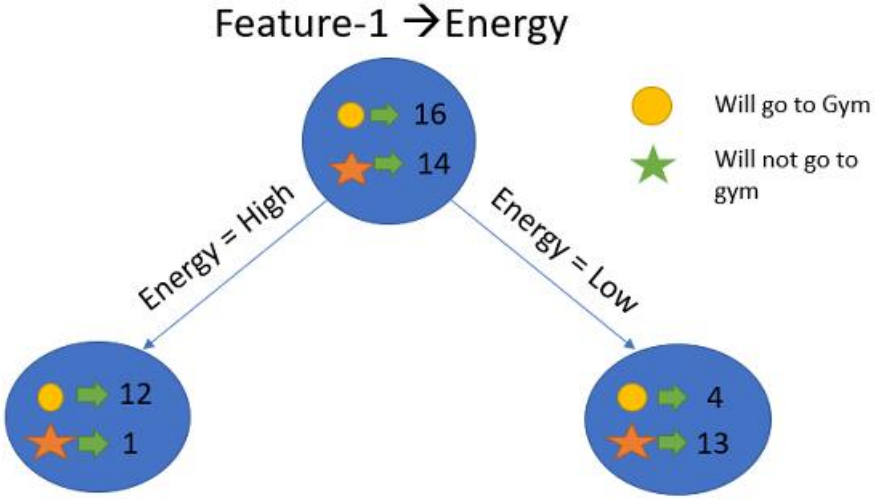
Information Gain

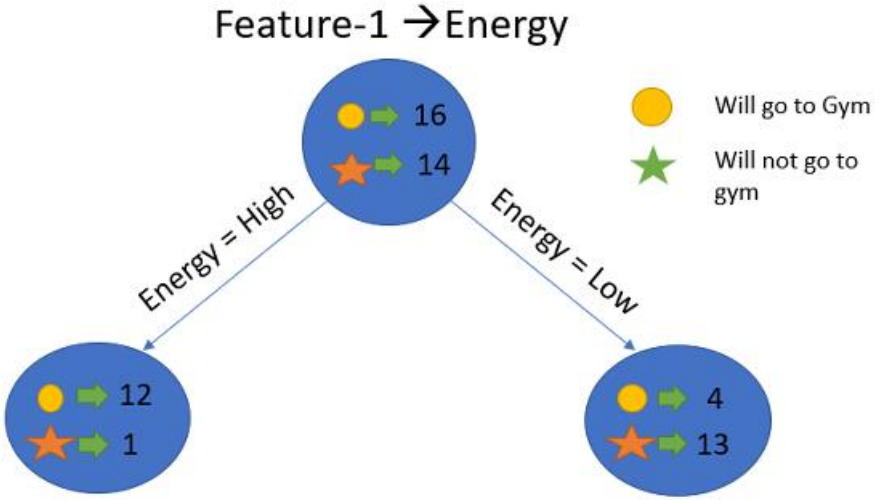
- Information gain measures the reduction of uncertainty given some feature and it is also a deciding factor for which attribute should be selected as a decision node or root node.

$$Information\ Gain\ =\ E(Y)\ -\ E(Y|X)$$

- It is just entropy of the full dataset – entropy of the dataset given some feature.
- To understand this better let’s consider an example:Suppose our entire population has a total of 30 instances. The dataset is to predict whether the person will go to the gym or not. Let’s say 16 people go to the gym and 14 people don’t

- Now we have two features to predict whether he/she will go to the gym or not.
- Feature 1 is **“Energy”** which takes two values *“high” and “low”*
- Feature 2 is **“Motivation”** which takes 3 values *“No motivation”, “Neutral” and “Highly motivated”.*





$$E(Parent) = -\left(\frac{16}{30}\right)\log_2\left(\frac{16}{30}\right) - \left(\frac{14}{30}\right)\log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Parent|Energy = "high") = -\left(\frac{12}{13}\right)\log_2\left(\frac{12}{13}\right) - \left(\frac{1}{13}\right)\log_2\left(\frac{1}{13}\right) \approx 0.39$$

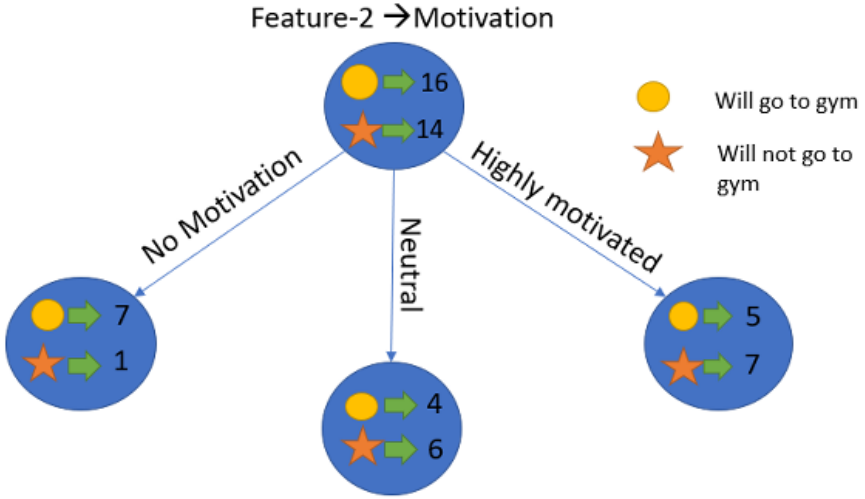
$$E(Parent|Energy = "low") = -\left(\frac{4}{17}\right)\log_2\left(\frac{4}{17}\right) - \left(\frac{13}{17}\right)\log_2\left(\frac{13}{17}\right) \approx 0.79$$

$$E(Parent|Energy) = \frac{13}{30} * 0.39 + \frac{17}{30} * 0.79 = 0.62$$

$$Information\ Gain = E(parent) - E(parent|energy)$$

$$= 0.99 - 0.62$$

$$= 0.37$$



$$E(Parent) = 0.99$$

$$E\left(Parent|Motivation = "No\ motivation"\right) = -\left(\frac{7}{8}\right)log_2\left(\frac{7}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) = 0.54$$

$$E(Parent|Motivation = "Neutral") = -\left(\frac{4}{10}\right)log_2\left(\frac{4}{10}\right) - (\frac{6}{10})\log_2\left(\frac{6}{10}\right) = 0.97$$

$$E(Parent|Motivation = "Highly\ motivated") = -\left(\frac{5}{12}\right)log_2\left(\frac{5}{12}\right) - (\frac{7}{12})\log_2\left(\frac{7}{12}\right) = 0.98$$

$$E(Parent|Motivation) = \frac{8}{30} * 0.54 + \frac{10}{30} * 0.97 + \frac{12}{30} * 0.98 = 0.86$$

$$Information\ Gain = E(Parent) - E(Parent|Motivation)$$

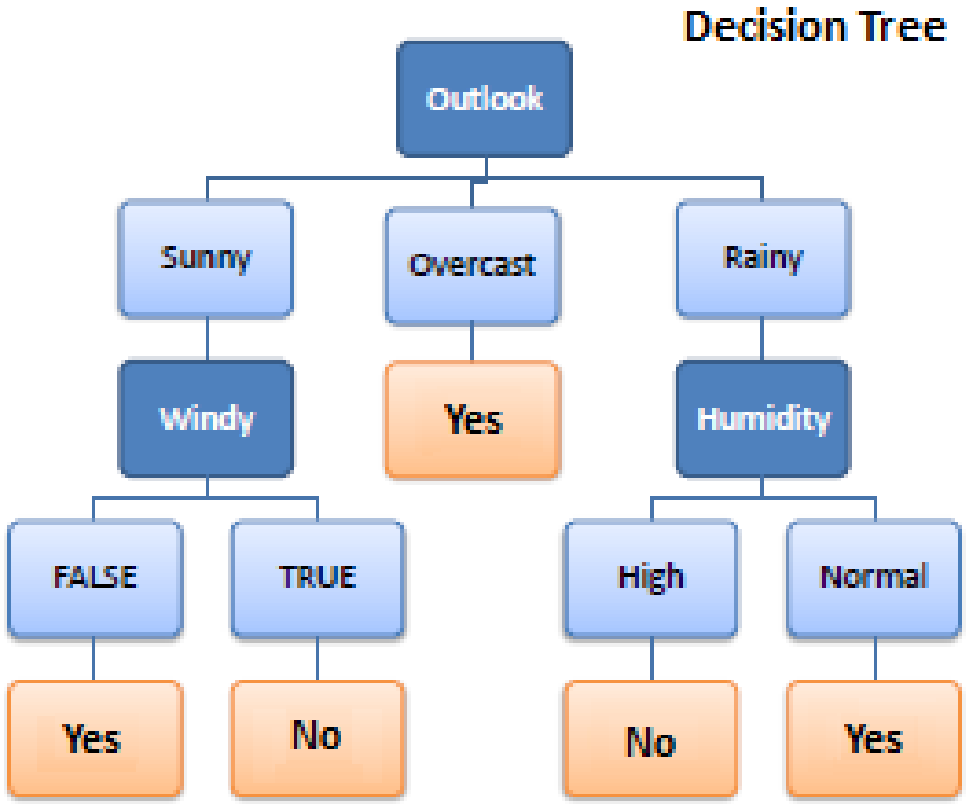
$$= 0.99 - 0.86$$

$$= 0.13$$

- We now see that the “Energy” feature gives highest information gain and hence we will select the “Energy” feature to split the nodes.

Complete example

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



Calculate the Entropy of the Target

$$\text{Entropy(PlayGolf)} = \text{Entropy}(5,9)$$

$$= \text{Entropy}(0.36, 0.64)$$

$$= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64)$$

$$= 0.94$$

Calculating Entropy and Information Gain of a specific split

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

E(PlayGolf, Outlook) = **P**(Sunny)**E**(3,2) + **P**(Overcast)**E**(4,0) + **P**(Rainy)**E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

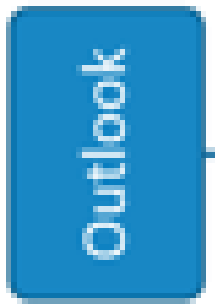
- The dataset is then split on the different attributes. The entropy for each branch is calculated.

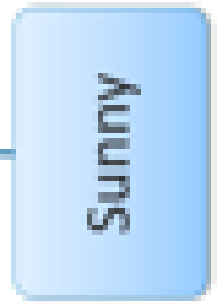
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

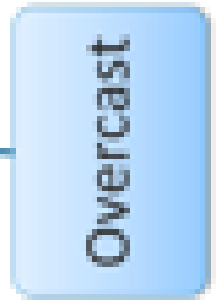
		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

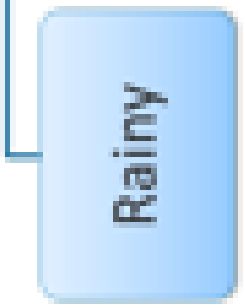
		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			





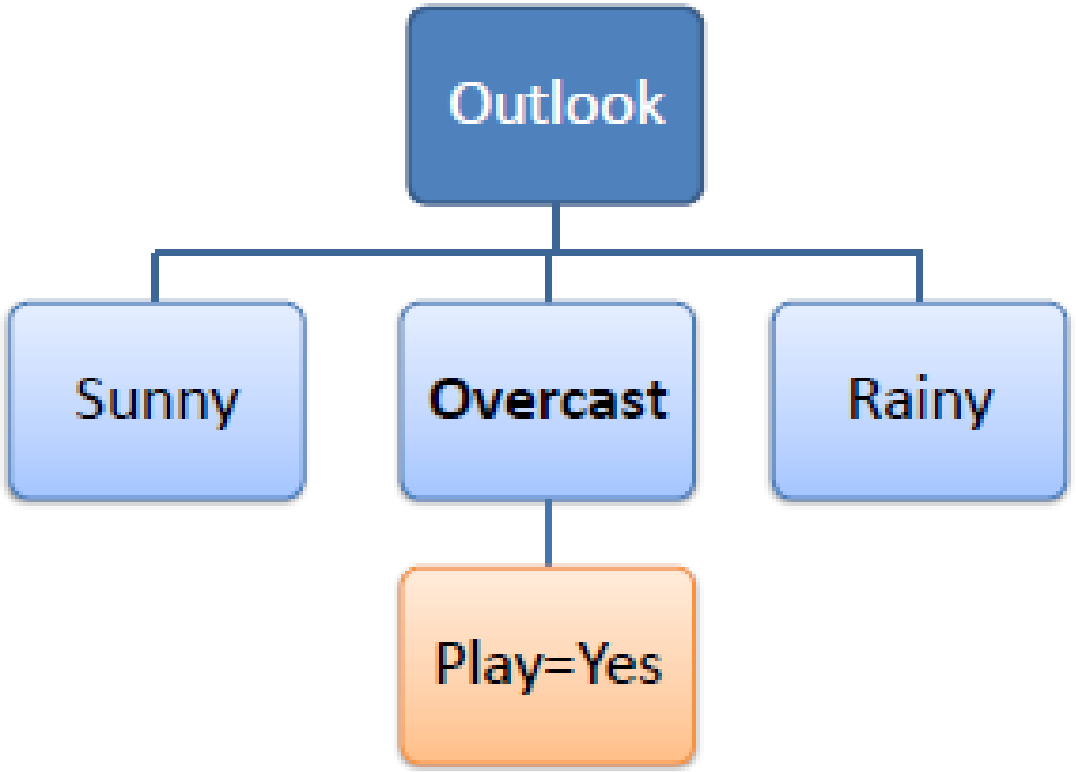




Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

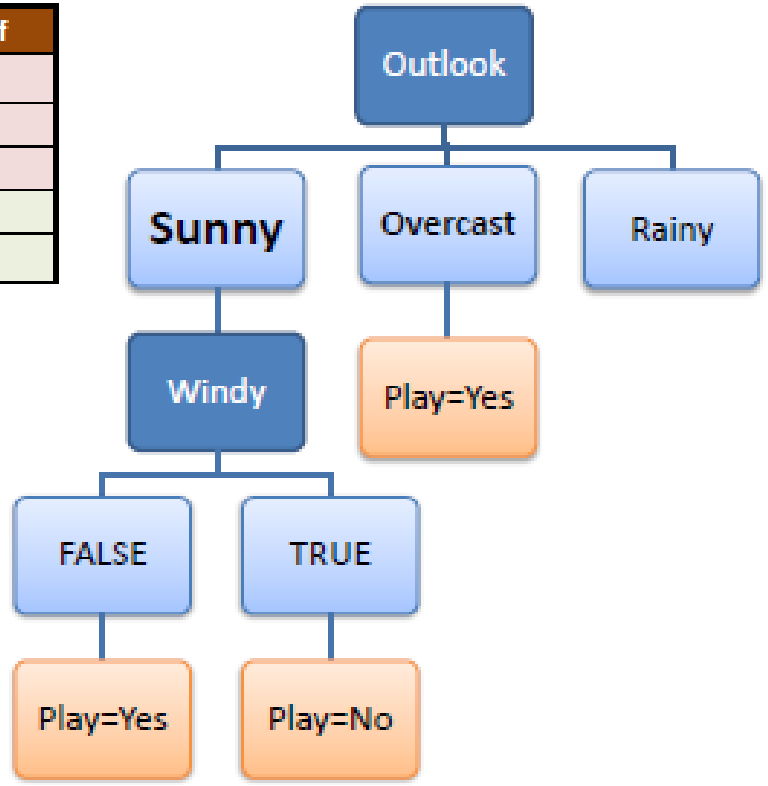
A branch with entropy of 0 is a leaf node

Temp.	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



A branch with entropy more than 0 needs further splitting.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Decision Tree to Decision Rules

- A decision tree can easily be transformed to a set of rules by mapping from the root node to the leaf nodes one by one.

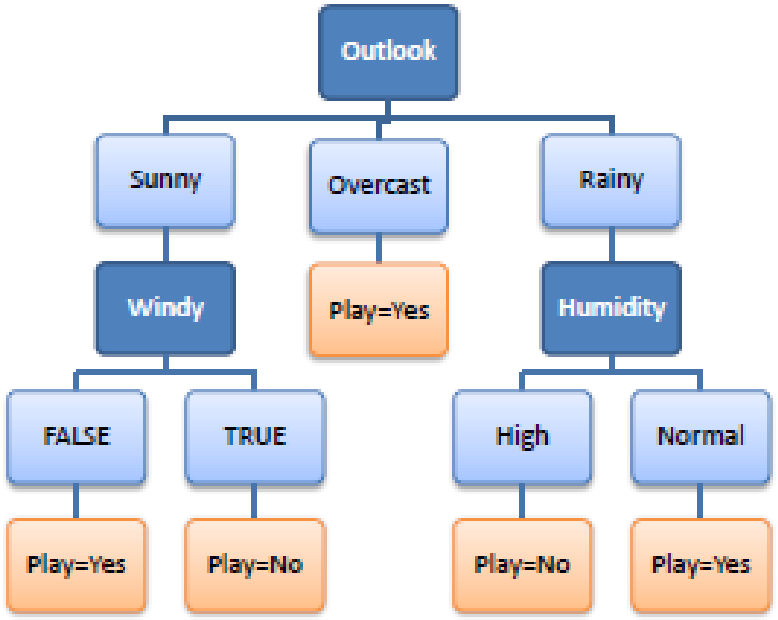
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes



- **Gini Index:** The other way of splitting a decision tree is via the Gini Index.
- The formula for Gini Index is

$$\text{Gini\_index} = 1 - \sum_{i=1}^{n} \big(P(i)\big)^2$$

- The lower the Gini Index, lower the likelihood of misclassification.
- The Gini Index has a minimum (highest level of purity) of 0. It has a maximum value of .5.

Example with Gini Index

Resp srl no	Target variable	Predictor variable	Predictor variable	Predictor variable
	Exam Result	Other online courses	Student background	Working Status
1	Pass	Y	Maths	NW
2	Fail	N	Maths	W
3	Fail	y	Maths	W
4	Pass	Y	CS	NW
5	Fail	N	Other	W
6	Fail	Y	Other	W
7	Pass	Y	Maths	NW
8	Pass	Y	CS	NW
9	Pass	n	Maths	W
10	Pass	n	CS	W
11	Pass	y	CS	W
12	Pass	n	Maths	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Maths	W

- Now let us calculate the Gini index for the root node for Student Background attribute.
- Maths sub node: 4Pass, 3Fail

$$Gini_{maths} = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = .4897$$

- CS sub node: 4Pass, 0 Fail

$$Gini_{CS} = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{5}\right)^2 = 0$$

- Others sub node: 0Pass, 4 Fail

$$Gini_{others} = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$$

- The overall Gini Index for this split is calculated similarly to the entropy as weighted average of the distribution across the 3 nodes.

$$Gini_{bkgrd} = \frac{7}{15} * .4897 + \frac{4}{15} * 0 + \frac{4}{15} * 0 = .2286$$

- Similarly, we can also compute the Gini Index for Working Status and Online Courses.

$$Gini_{working} = 1 - \left(\frac{6}{9}\right)^2 - \left(\frac{3}{9}\right)^2 = .44$$

$$Gini_{notworking} = 1 - \left(\frac{5}{6}\right)^2 - (6)^2 = .278$$

$$Gini_{workstatus} = \frac{9}{15} * .44 + \frac{6}{15}.278 = .378$$

$$Gini_{online} = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2 = .4688$$

$$Gini_{notonline} = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = .4898$$

$$Gini_{online} = \frac{8}{15} * .4688 + \frac{7}{15}.4898 = .479$$

- The Gini Index is lowest for the Student Background variable. Hence, we pick this variable for the root node.

Handling Continuous Predictor Variable

- This can be accomplished by dynamically defining new discretevalued attributes that partition the continuous attribute value into a discrete set of intervals.
- In particular, for an attribute A that is continuous-valued, the algorithm can dynamically create a new boolean attribute A_c, that is true if A < c and false otherwise.
- how to select the best value for the threshold c?

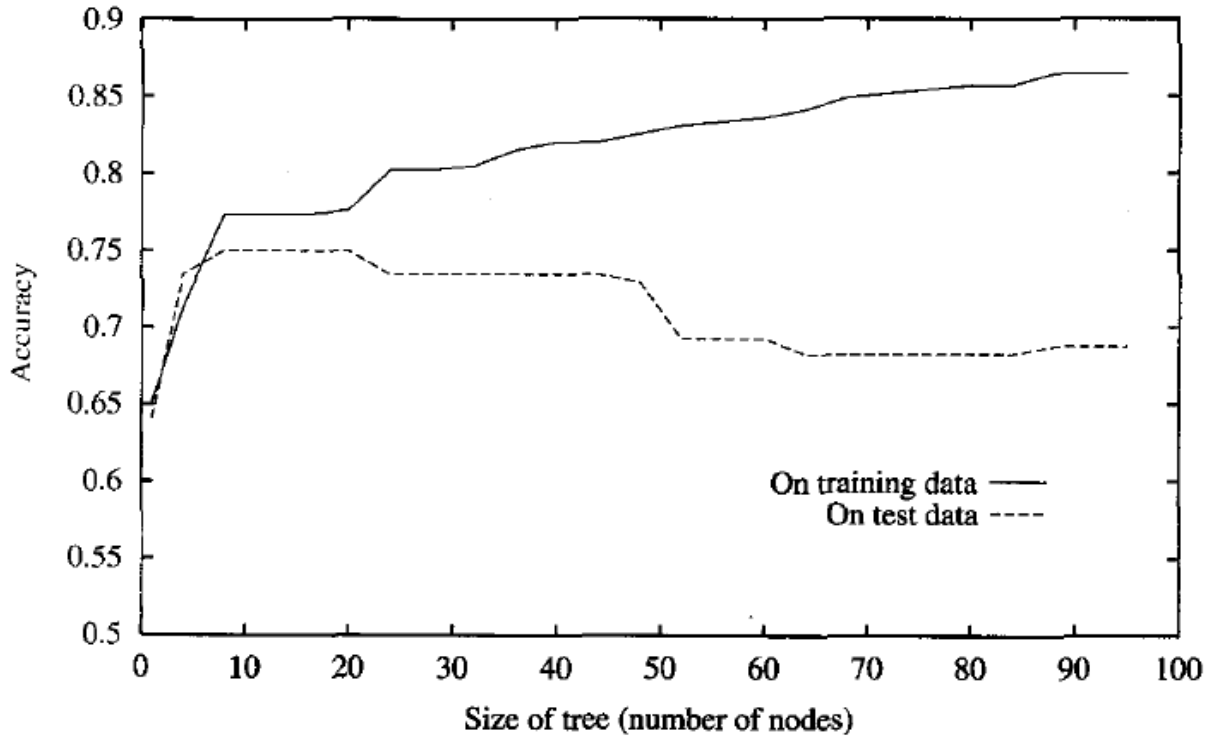
- suppose we wish to include the continuous-valued attribute ***Temperature*** in describing the training example days in the learning task of following Table.
- ***Temperature:*** 40 48 60 72 80 90
- ***PlayTennis:*** No No Yes Yes Yes NO
- Clearly, we would like to pick a threshold, ***c,*** that produces the greatest information gain.
- By sorting the examples according to the continuous attribute **A,** then identifying adjacent examples that differ in their target classification, we can generate a set of candidate thresholds midway between the corresponding values of ***A.***

- These candidate thresholds can then be evaluated by computing the information gain associated with each.
- In the current example, there are two candidate thresholds, corresponding to the values of Temperature at which the value of PlayTennis changes: (48 + 60)/2, and (80 + 90)/2.
- The information gain can then be computed for both the threshold and best can be found.

Overfitting and Decision Trees

- Overfitting can be a big challenge with Decision Trees.
- Algorithm continues to split till it reaches a leaf node (pure node) which implies training error to be 0.
- Often the leaf node may just have one or two instances.
- This will clearly lead to a complex tree structure which may not generalize well to a test scenario.
- There are several ways we can prevent the decision tree from becoming too large.

Overfitting Scenario



- Pre pruning or Early stopping: Preventing the tree from growing too big or deep
- Post Pruning: Allowing a tree to grow to its full depth and then getting rid of various branches based on various criteria

Pre pruning

- The pre-pruning technique refers to the early stopping of the growth of the decision tree.
- The pre-pruning technique involves tuning the hyperparameters of the decision tree model prior to the training pipeline.
- The hyperparameters of the DecisionTreeClassifier in SkLearn include max_depth, min_sample_leaf, min_sample_split which can be tuned to early stop the growth of the tree and prevent the model from overfitting.

Post Pruning

- In this technique, we allow the tree to grow to its maximum depth.
- Convert the learned tree into an equivalent set of rules by creating one rule
- Prune (generalize) each rule by removing any preconditions that result in improving its estimated accuracy.
- Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances.

- Although the first of these approaches might seem more direct, the second approach of post-pruning overfit trees has been found to be more successful in practice.
- This is due to the difficulty in the first approach of estimating precisely when to stop growing the tree.

Advantages and Disadvantages

- Advantages:
  - Trees give a visual schema of the relationship of variables used for classification.
  - White box model which is explainable
  - In general there is less need to prepare and clean data
- Disadvantages:
  - Prone to overfitting and hence lower predictive accuracy
  - Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.