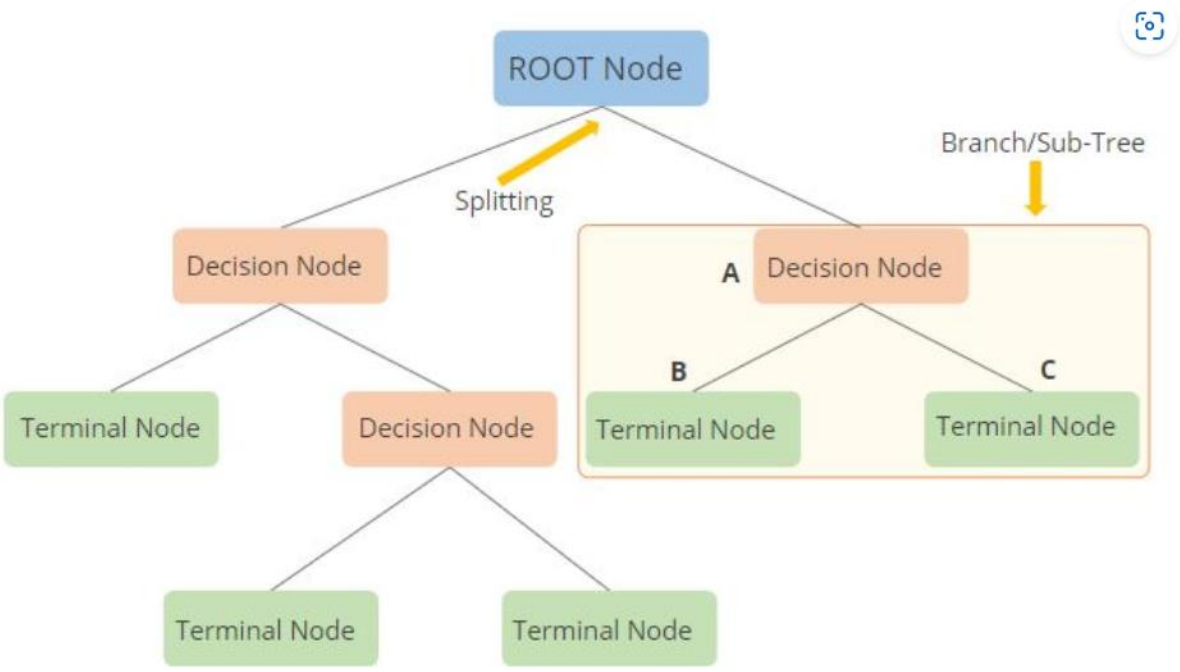
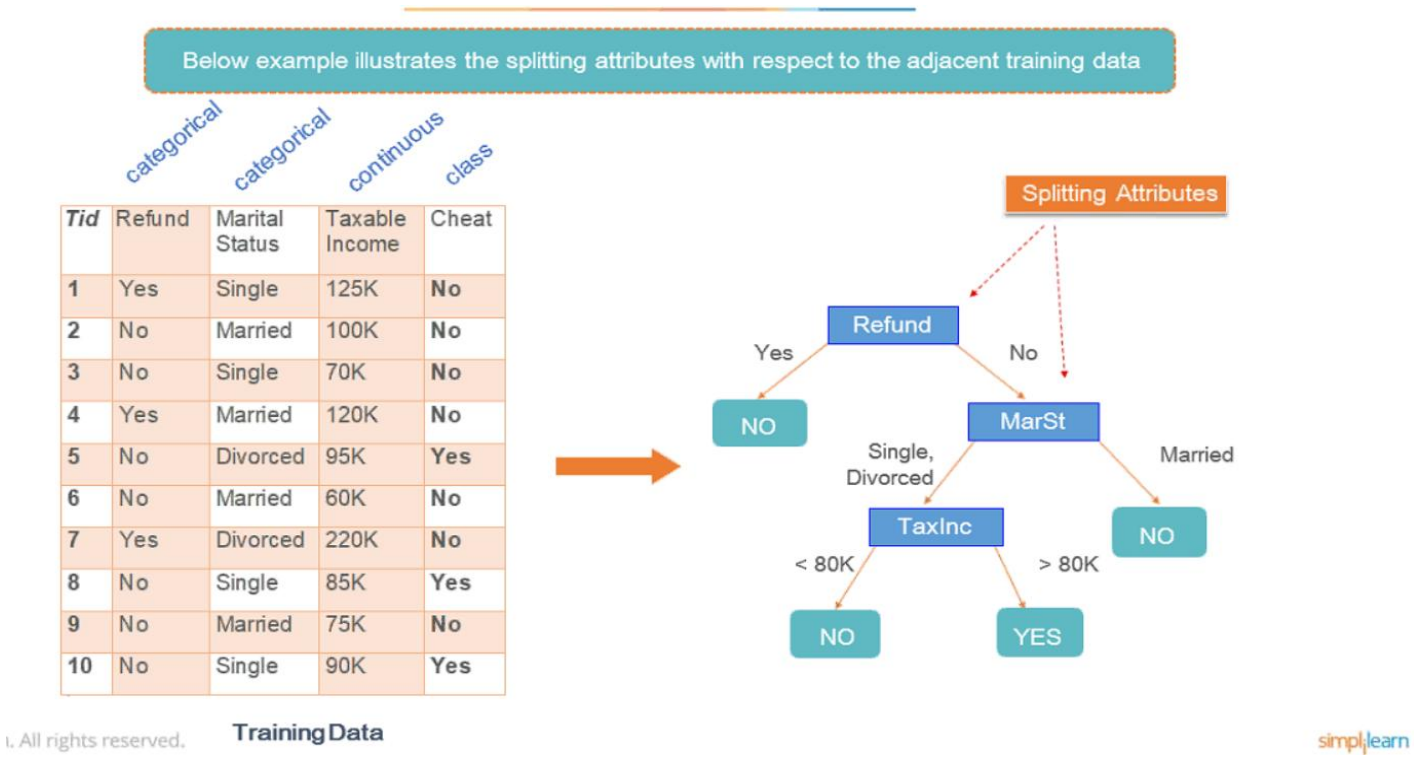
Decision Tree

- A decision tree is a type of supervised machine learning used to categorize or make predictions.
- Decision tree is a tree-like structure in which the internal node represents the test on an attribute.
- The topmost node in a decision tree is known as the root node. It learns to partition based on the attribute value.
- Each branch represents the outcome of the test and each leaf node represents the class label.
- A path from root to leaf represents the classification rules.





1. Which color ? / Which brand ?

- Why is not the "Which color?" question good starting point ?
- Because this is not helping you to gain much information to your objective. OR
- if I say this is not going to help you decrease your uncertainity.
- But asking which brand is definitely going to lead you to some information.
- So the idea is we ask the questions in such an order such that it leads to gain in maximum information.
- Now In decision tree building how do we model this ?
- We use **Entropy** and **Information gain** to build the tree.

- Entropy : Measure of uncertainity 0 =< E <= 1

- Information Gain : Reduction in Entropy

Lets go back to the very same game of asking questions about the car.

Initially the entropy is maximum == 1

No information you have

Question 1 :

1. Which brand ?

Answer : Hyundai

Entropy might have decreased. Lets say it has decreased to 0.7.

How much information did you gain ? 1 - 0.7 = 0.3

Question 2 :

1. Which Type ?

Answer : Sedan

Entropy might have decreased. Lets say it has decreased to 0.4.

How much information did you gain? 0.7 - 0.4 = 0.3

Question 3 :

1. Price range ?

Answer : 10-15

Entropy might have decreased. Lets say it has decreased to 0.1.

How much information did you gain ? 0.4 - 0.1 = 0.3

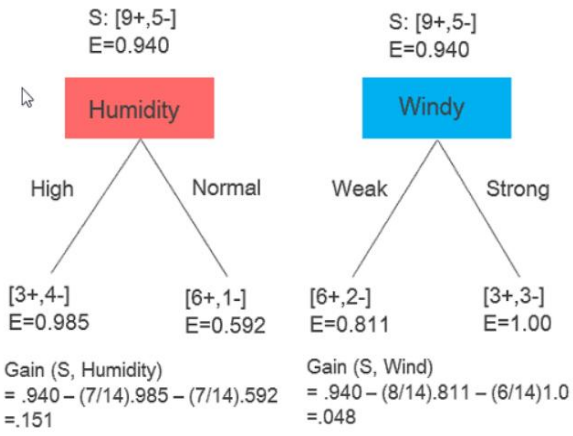
- The whole idea of selecting which feature to split is based upon the entropy and information gain of each split. We will calculate the information gain on each possible split and select the

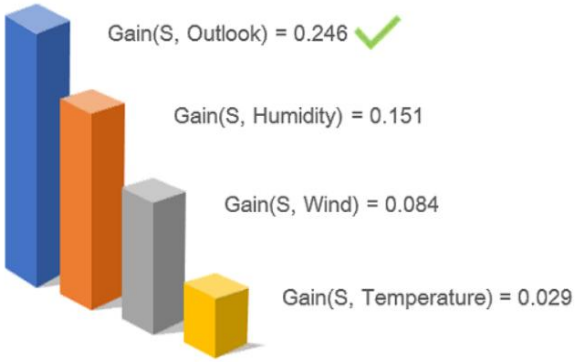
one with max information gain. We keep on doing this for every split till we get terminal or leaf nodes.

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Which Attribute Is the Best Classifier?

The attribute with the highest information gain is selected as the splitting attribute





$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

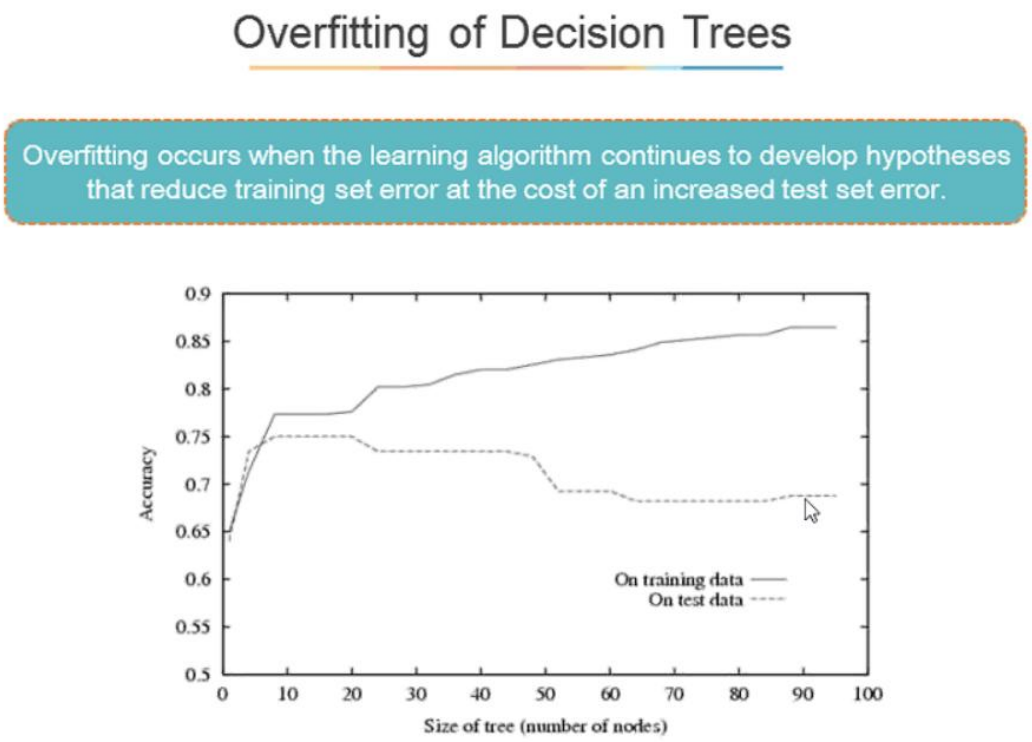
Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Entropy = - (P+) log (P+) - (P-) log (P-)

$$Gain(S, A) = Entropy(S) - \Sigma \frac{|Sv|}{|S|} Entropy(Sv)$$

Gini Index = 1- [(P+)^2 + (P-)^2]

Problem:



Solution : Do not increase number of nodes exponently.

Max_depth = none (**add some val**)

Hyperparameters:

1) Max_depth
2) Min_sample_leaf
3) Criterion