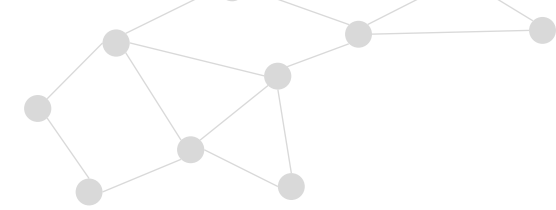
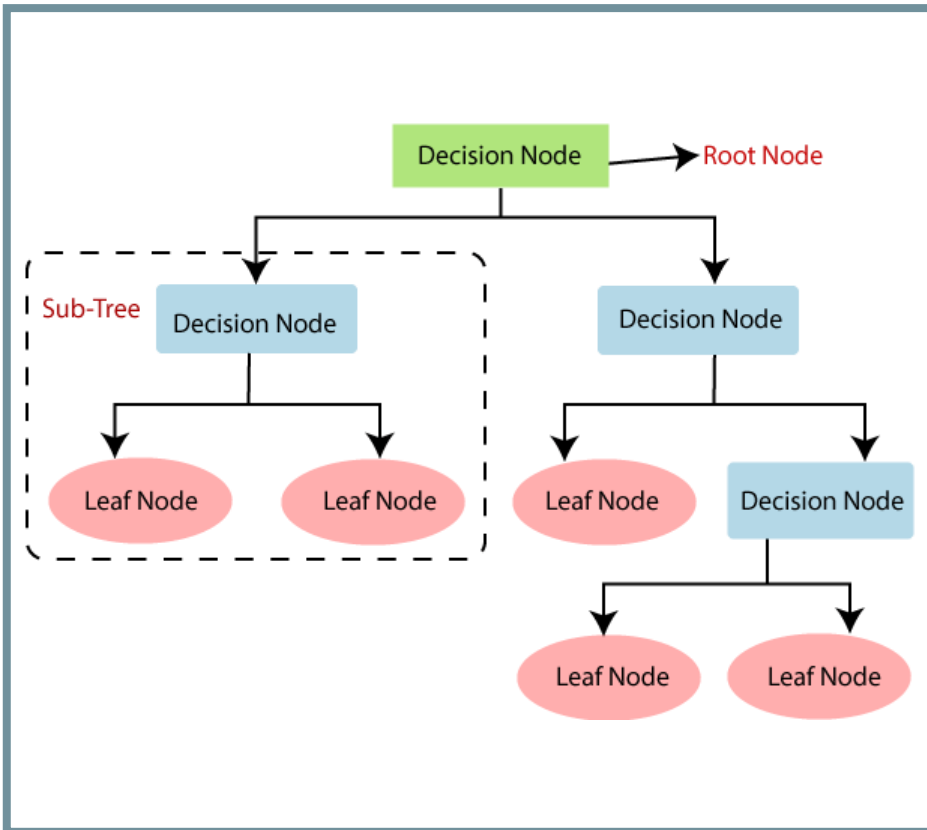


Classification: Decision Tree Introduction



Decision Tree

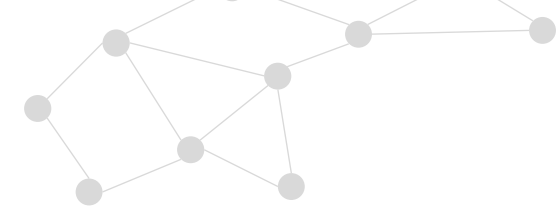


Decision Tree is a **Supervised learning technique** that can be used for both Classification and Regression problems, but mostly it is preferred for solving Classification problems.

It is a tree-structured classifier, where:

- internal nodes represent the features of a dataset.
- branches represent the decision rules.
- each leaf node represents the outcome.

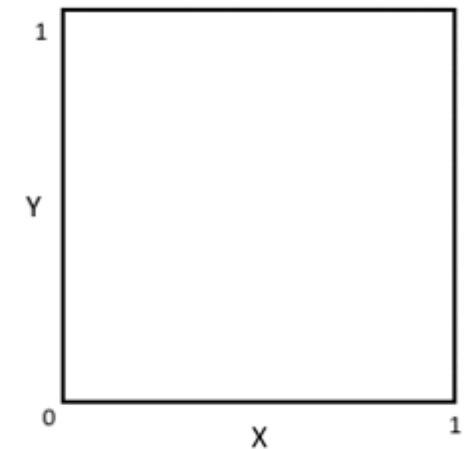
Credit: <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>



Decision Tree

The Decision Tree algorithm intuition is as follows:

1. For each attribute in the dataset, the Decision-Tree algorithm forms a node. The most important attribute is placed at the root node.
2. For evaluating the task in hand, we start at the root node and we work our way down the tree by following the corresponding node that meets our condition or decision.
3. This process continues until a leaf node is reached. It contains the prediction or the outcome of the Decision Tree.



For more tutorials: annalysin.wordpress.com

Credit: <https://algobeans.com/2016/07/27/decision-trees-tutorial/>



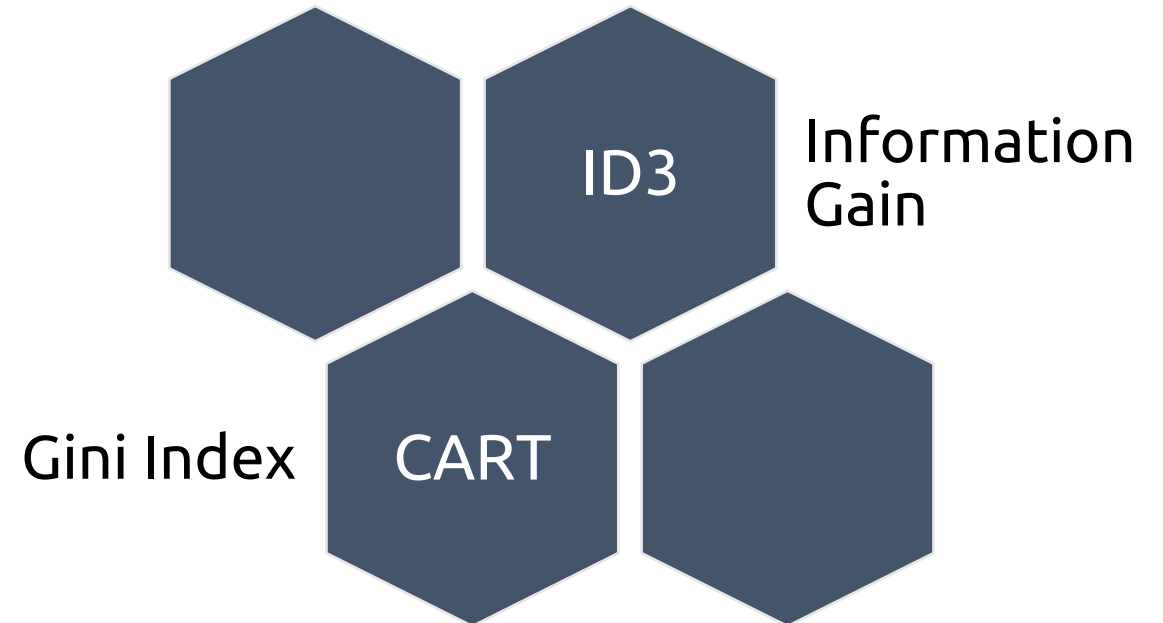
Decision Tree Algorithms

There are 2 popular tree building-algorithm.

The main difference between these two models is the cost function that they use.

The **cost function** decides which question to ask and how each node being split, i.e. **attribute selection**.

There are different attributes selection measure to identify the attribute which can be considered as the root node at each level.





Decision Tree Advantages

It can capture nonlinear relationships: They can be used to classify non-linearly separable data.

Easy to understand, interpret, visualize.

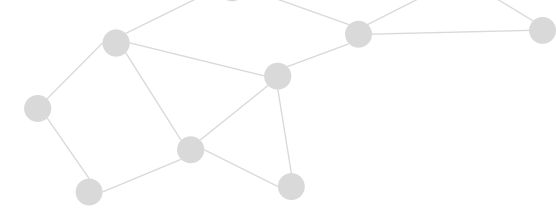
Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.

A decision tree does not require normalization of data.

A decision tree does not require scaling of data as well.

Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.

A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

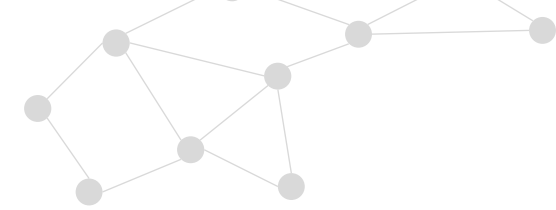


Decision Tree Disadvantages

A small change in the data can cause a large change in the structure of the decision tree causing **instability**.

Adding a new data point can lead to re-generation of the overall tree and all nodes need to be recalculated and recreated.

It can't be used in big data: If the size of data is too big, then one single tree may grow a lot of nodes which might result in complexity and leads to **overfitting**.



Thanks!

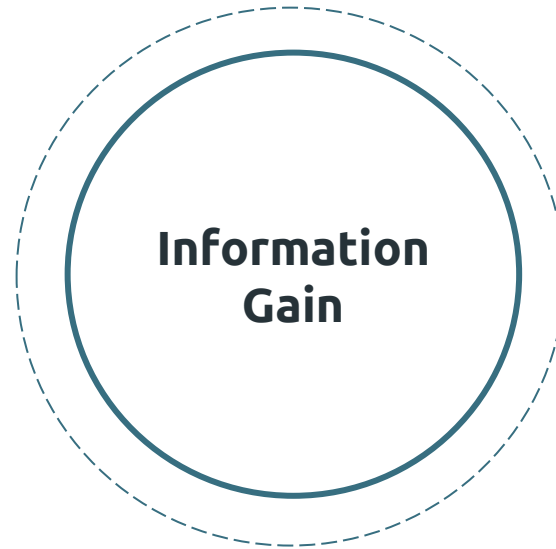
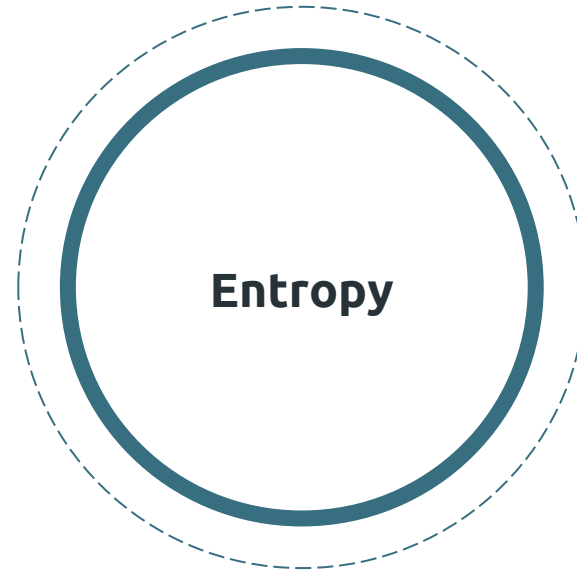
This is the end of Classification: Decision Tree
Introduction, see you in the next topic.



Classification: Decision Tree with ID3 Algorithm



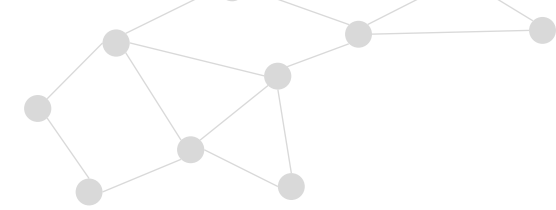
Entropy measures the impurity in the given dataset. In Physics and Mathematics, entropy is referred to as the randomness or uncertainty of a random variable X . In information theory, it refers to the impurity in a group of examples.



Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values.

ID3 Algorithm

The ID3 (Iterative Dichotomiser) Decision Tree algorithm uses entropy to calculate information gain. So, by calculating decrease in entropy measure of each attribute we can calculate their information gain. The attribute with the highest information gain is chosen as the splitting attribute at the node.



ID3 Algorithm

$$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

Entropy, also called as Shannon Entropy is denoted by $H(S)$ for a finite set S , is the measure of the amount of uncertainty or randomness in data. Highest entropy is when there's no way of determining what the outcome.

Consider a coin which has heads on both sides. Since we know beforehand that it'll always be heads, this event has no randomness and its entropy is zero.

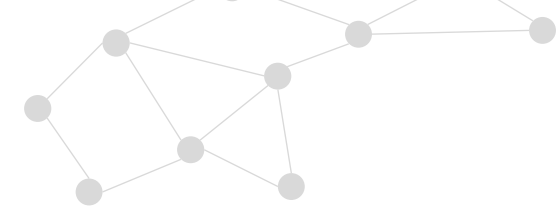
In other words, lower values imply less uncertainty while higher values imply high uncertainty.

$$IG(S, A) = H(S) - H(S, A)$$

$$IG(S, A) = H(S) - \sum_{i=0}^n P(x) * H(x)$$

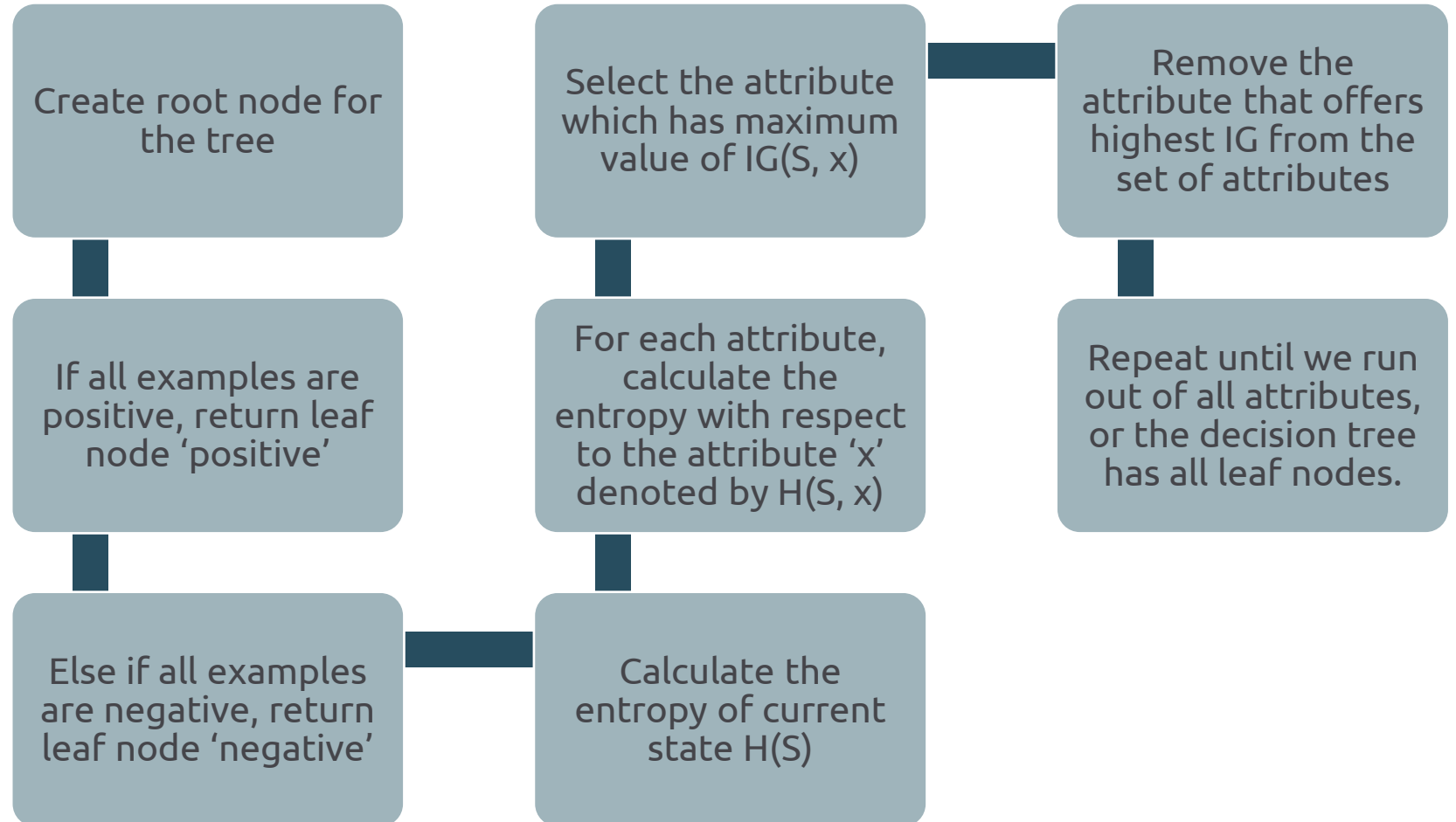
Information Gain denoted by $IG(S, A)$ for a set S is the effective change in entropy after deciding on a particular attribute A . It measures the relative change in entropy with respect to the independent variables.

where $IG(S, A)$ is the information gain by applying feature A . $H(S)$ is the Entropy of the entire set, while the second term calculates the Entropy after applying the feature A , where $P(x)$ is the probability of event x .



ID3 Algorithm

ID3 Algorithm will perform following tasks recursively:





ID3 Algorithm - Example

Day	Outlook	Temperature	Humidity	Wind	Play Golf
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

Let's understand the ID3 algorithm with the help of a simple example.

Consider a piece of data collected over the course of 14 days where the features are Outlook, Temperature, Humidity, Wind and the outcome variable is whether Golf was played on the day.



ID3 Algorithm - Example

Day	Outlook	Temperature	Humidity	Wind	Play Golf
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

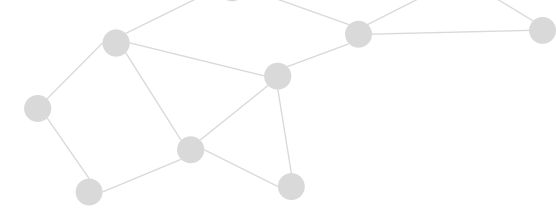
In the example, we can see in total there are 5 No's and 9 Yes's.

$$Entropy(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

$$Entropy(S) = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right)$$

$$= 0.940$$

Remember that the Entropy is 0 if all members belong to the same class (no uncertainty), and 1 when half of them belong to one class and other half belong to other class (high randomness/high uncertainty). Here it's 0.94 which means the distribution is fairly random. Now the next step is to choose the attribute that gives us highest possible Information Gain which we'll choose as the root node.



ID3 Algorithm - Example

1

Let's start with 'Wind'. Amongst all the 14 examples we have 8 places where the wind is weak and 6 where the wind is Strong.

$$P(S_{weak}) = \frac{\text{Number of Weak}}{\text{Total}} = \frac{8}{14}$$

$$P(S_{strong}) = \frac{\text{Number of Strong}}{\text{Total}} = \frac{6}{14}$$

2

Out of the 8 Weak examples, 6 of them were 'Yes' for Play Golf and 2 of them were 'No' for 'Play Golf'

$$\begin{aligned} Entropy(S_{weak}) &= -\left(\frac{6}{8}\right)\log_2\left(\frac{6}{8}\right) - \left(\frac{2}{8}\right)\log_2\left(\frac{2}{8}\right) \\ &= 0.811 \end{aligned}$$

Similarly, out of 6 Strong examples, we have 3 examples where the outcome was 'Yes' for Play Golf and 3 where we had 'No' for Play Golf (this is perfect randomness)

$$\begin{aligned} Entropy(S_{strong}) &= -\left(\frac{3}{6}\right)\log_2\left(\frac{3}{6}\right) - \left(\frac{3}{6}\right)\log_2\left(\frac{3}{6}\right) \\ &= 1.000 \end{aligned}$$

3

Now calculate the information gain

$$IG(S, Wind) = H(S) - \sum_{i=0}^n P(x) * H(x)$$

$$\begin{aligned} IG(S, Wind) &= H(S) - P(S_{weak}) * H(S_{weak}) - P(S_{strong}) * H(S_{strong}) \\ &= 0.940 - \left(\frac{8}{14}\right)(0.811) - \left(\frac{6}{14}\right)(1.00) \\ &= 0.048 \end{aligned}$$



ID3 Algorithm - Example

Counting information gain of all attributes:

$$IG(S, Outlook) = 0.246$$

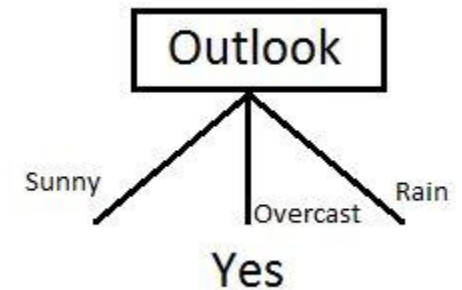
$$IG(S, Temperature) = 0.029$$

$$IG(S, Humidity) = 0.151$$

$$IG(S, Wind) = 0.048 \text{ (Previous example)}$$

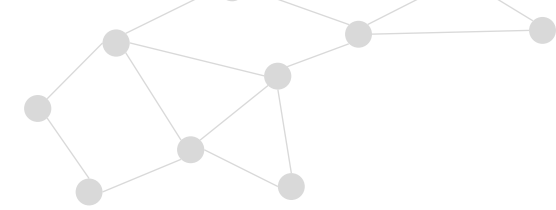


We can clearly see that $IG(S, Outlook)$ has the highest information gain of 0.246, hence we chose Outlook attribute as the root node. At this point, the decision tree looks like:



Whenever the outlook is Overcast, Play Golf is always 'Yes', it's no coincidence by any chance, the simple tree resulted because of the highest information gain is given by the attribute Outlook. Now how do we proceed from this point? We can simply apply recursion, you might want to look at the algorithm steps described earlier.

Now that we've used Outlook, we've got three of them remaining Humidity, Temperature, and Wind. And, we had three possible values of Outlook: Sunny, Overcast, Rain. The Overcast node already ended up having leaf node 'Yes', so we're left with two subtrees to compute: Sunny and Rain.



ID3 Algorithm - Example

For **Sunny Subtree**, the table looks like this:

Temperature	Humidity	Wind	Play Golf
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

In the similar fashion, we compute the following values

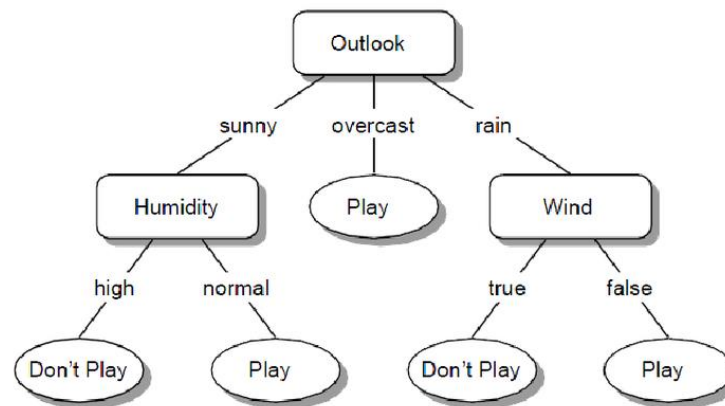
$$IG(S_{\text{sunny}}, \text{Humidity}) = 0.96$$

$$IG(S_{\text{sunny}}, \text{Temperature}) = 0.57$$

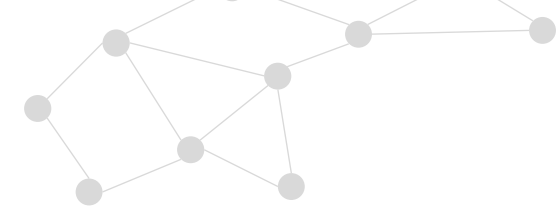
$$IG(S_{\text{sunny}}, \text{Wind}) = 0.019$$

We get **Humidity** as the highest information gain

After doing the similar thing in rain subtree, the **final decision tree** will look like this:



Your Turn: For Rain Subtree, we get Wind as the highest information gain, try calculate this by yourself 😊



Thanks!

This is the end of Classification: Decision Tree with ID3
Algorithm, see you in the next topic.



Classification: Decision Tree with CART Algorithm



CART Algorithm

Classification And Regression Trees (CART) algorithm is a classification algorithm for building a decision tree based on Gini's impurity index as splitting criterion. Gini impurity is calculated using following formula:

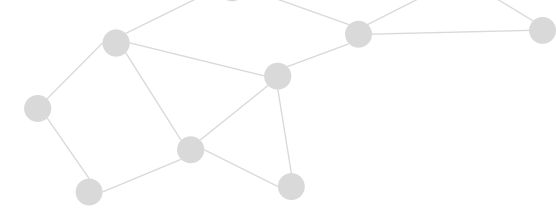
$$GiniIndex = 1 - \sum_j p_j^2$$

Gini impurity measures the frequency at which any element of the dataset will be mislabelled when it is randomly labeled.

The minimum value of the Gini Index is 0. This happens when the node is pure, this means that all the contained elements in the node are of one unique class. Therefore, this node will not be split again. Thus, the optimum split is chosen by the features with less Gini Index. Moreover, it gets the maximum value when the probability of the two classes are the same.

$$Gini_{min} = 1 - (1^2) = 0$$

$$Gini_{max} = 1 - (0.5^2 + 0.5^2) = 0.5$$



CART Algorithm - Example

Day	Outlook	Temperature	Humidity	Wind	Play Golf
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

Let's understand the CART algorithm with the help of a simple example.

Consider a piece of data collected over the course of 14 days where the features are Outlook, Temperature, Humidity, Wind and the outcome variable is whether Golf was played on the day.



CART Algorithm - Example

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Let's understand the CART algorithm with the help of golf playing decision dataset.

Note that number of classes is 2 for 'No' = not playing golf and 'Yes' = playing golf.

Outlook consists of 3 values (Sunny, Overcast, Rain)

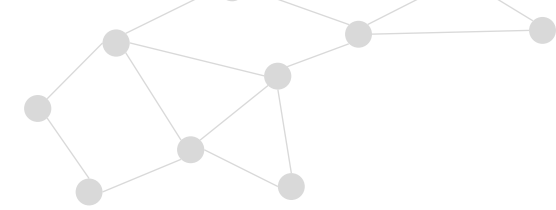
1

- $\text{Gini}(\text{Outlook}=\text{Sunny}) = 1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 = 0.48$
- $\text{Gini}(\text{Outlook}=\text{Overcast}) = 1 - (4/4)^2 - (0/4)^2 = 0$
- $\text{Gini}(\text{Outlook}=\text{Rain}) = 1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 = 0.48$

2

Then, weighted sum of Gini indexes for outlook feature:

$$\text{Gini}(\text{Outlook}) = (5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$$



CART Algorithm - Example

Humidity is a binary class feature and has 2 values (High and Normal)

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

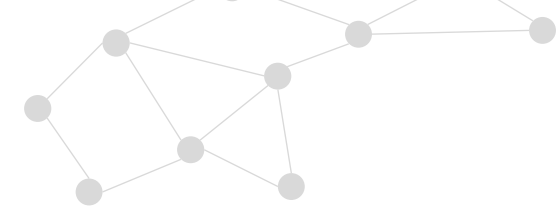
- $\text{Gini}(\text{Humidity}=\text{High}) = 1 - (3/7)^2 - (4/7)^2 = 1 - 0.183 - 0.326 = 0.489$
 - $\text{Gini}(\text{Humidity}=\text{Normal}) = 1 - (6/7)^2 - (1/7)^2 = 1 - 0.734 - 0.02 = 0.244$
- $\text{Gini}(\text{Humidity}) = (7/14) \times 0.489 + (7/14) \times 0.244 = 0.367$

Wind is a binary class feature and has 2 values (Weak and Strong)

Wind	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

- $\text{Gini}(\text{Wind}=\text{Weak}) = 1 - (6/8)^2 - (2/8)^2 = 1 - 0.5625 - 0.062 = 0.375$
 - $\text{Gini}(\text{Wind}=\text{Strong}) = 1 - (3/6)^2 - (3/6)^2 = 1 - 0.25 - 0.25 = 0.5$
- $\text{Gini}(\text{Wind}) = (8/14) \times 0.375 + (6/14) \times 0.5 = 0.428$

Your Turn: Can you calculate Gini index for the next feature, i.e., Temperature?

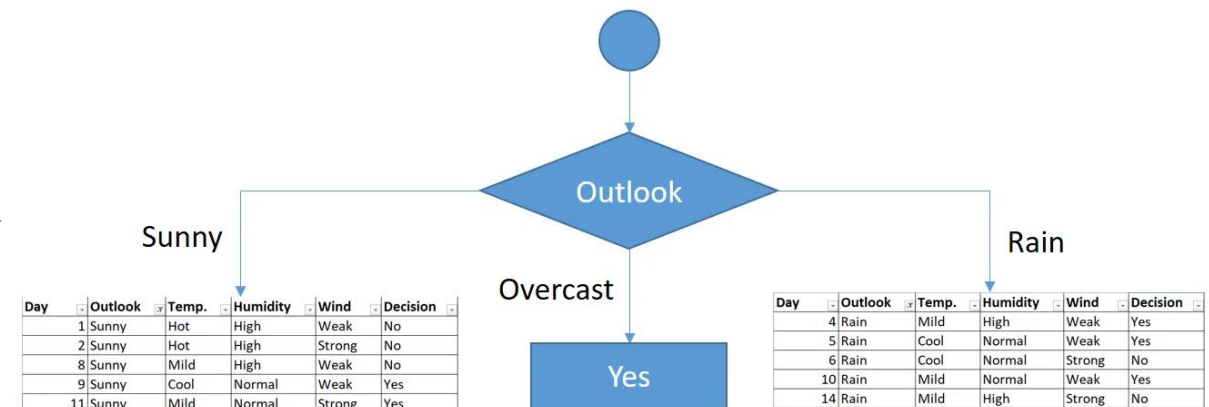
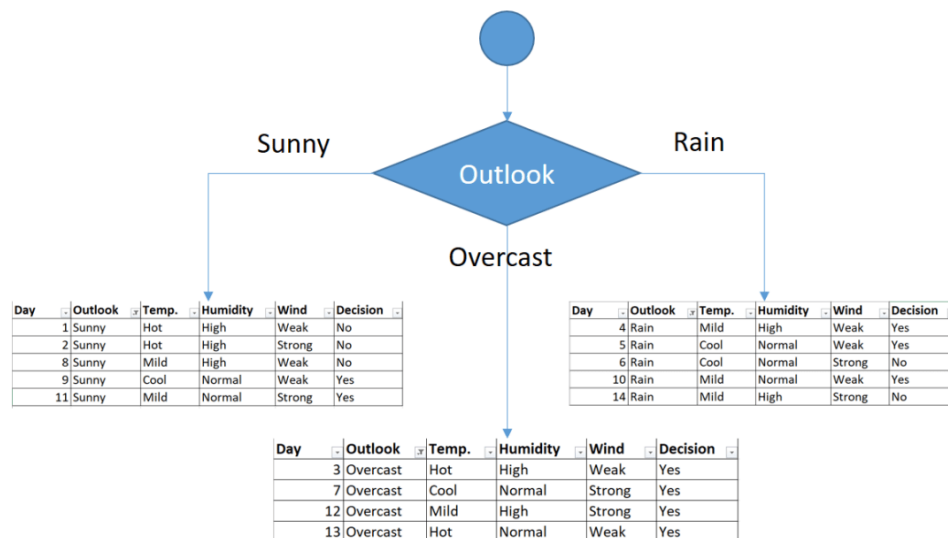


CART Algorithm - Example

Deciding **Root Node**. After calculating gini index for all attributes, we get:

- Gini (Outlook) = 0.342
- Gini (Temperature) = 0.439
- Gini (Humidity) = 0.367
- Gini (Wind) = 0.428

We see Outlook has the lowest index and put it at the top of the tree, i.e. becomes the root node



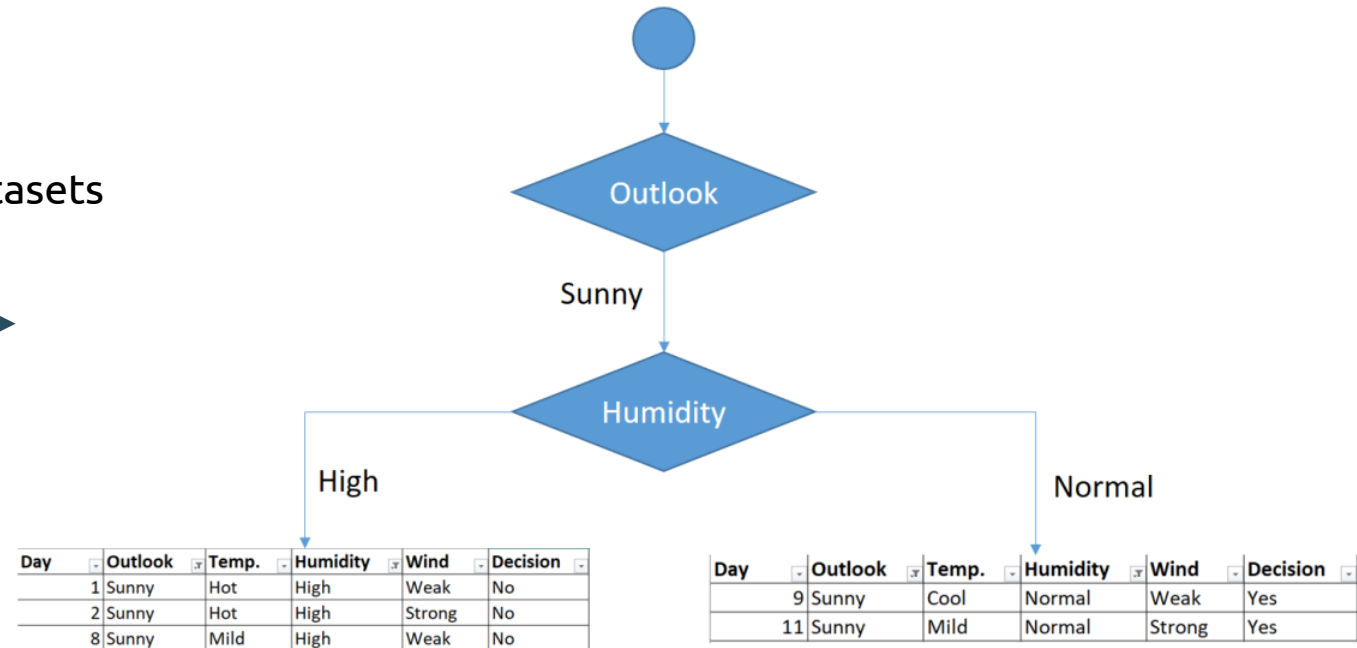


CART Algorithm - Example

Next, we will apply same principles to the next sub datasets

This one is for Sunny Outlook

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



As seen, decision is always no for high humidity and yes for normal humidity. So, this branch is over.

With the same calculation as before you'll get

- Gini (Sunny, Temperature) = 0.2
- Gini (Sunny, Humidity) = 0
- Gini (Sunny, Wind) = 0.466

We choose humidity and the decision tree of the sunny outlook

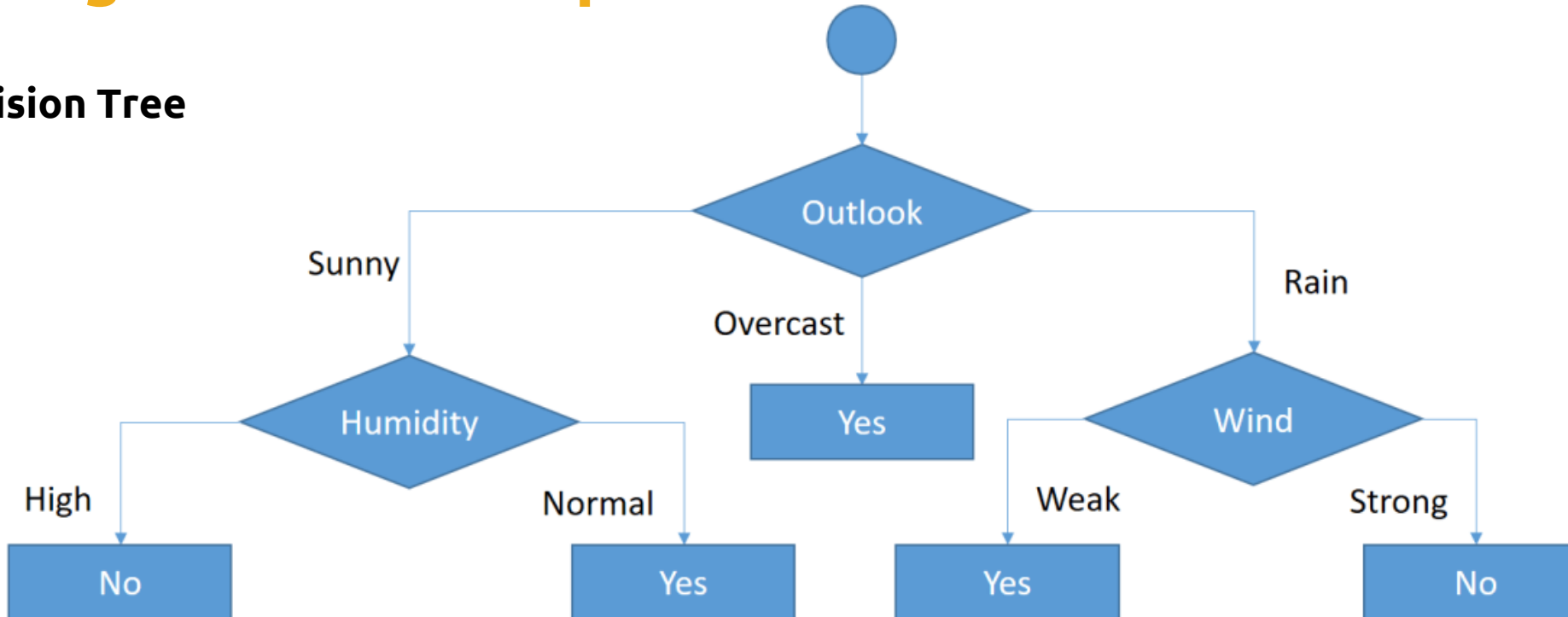
Your Turn:

Can you calculate the Gini indexes for the Rain Outlook branch?

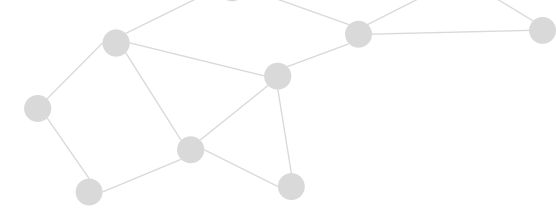


CART Algorithm - Example

Final Decision Tree



You might realize that we've created exactly the same tree as in ID3 example. This does not mean that ID3 and CART algorithms always produce same trees. This simple example fortunately generates the same tree. But it's not always the case. For more complex system, both algorithms might result in different trees.



Thanks!

This is the end of Classification: Decision Tree with CART
Algorithm, see you in the next topic.

