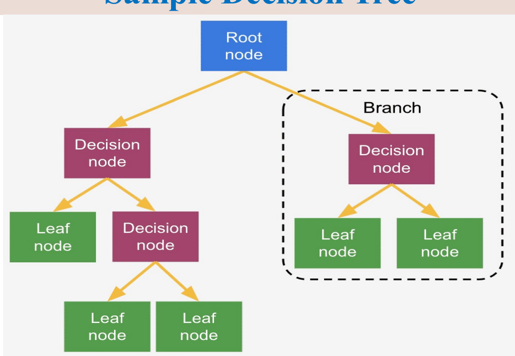
Decision Tree (DT)

- Decision tree builds classification or regression models in the form of a **tree structure**
- In a Decision tree, there are mainly two nodes, which are the **Decision Node (node)** and **Leaf Node (leaf).**
- Decision nodes are used to **make any decision** and have multiple branches, whereas **Leaf nodes** are the **output of those decisions** and do not contain any further branches
- Looks like an upside down tree

Sample Decision Tree



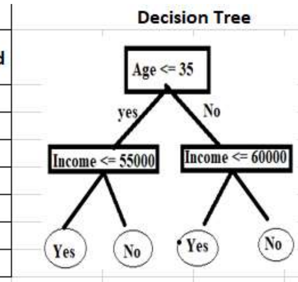
- **Root Node:** This is the topmost node in the tree, representing the entire dataset or a subset of it.
- **Leaf Nodes:** These are the terminal nodes or end points of the tree. Each leaf node represents a class label (for classification) or a numerical value (for regression).
- **Internal Nodes:** These nodes represent tests on attributes

Example

Sample datasets

Age	Income	Purchased
25	50000	No
35	60000	No
45	75000	Yes
20	30000	No
50	80000	Yes
22	40000	Yes
30	70000	No

Decision Tree



Decision trees are;

- Quite easy to interpret its results
- Require minimal pre-processing of data.
- A decision tree handles outliers and missing values automatically.

*disadvantage: A decision tree algorithm is not suitable for a large dataset and is prone to overfitting

Building a Decision Tree

- First test all attributes and select the one that would function as **the best root (root node)**. The root node is selected based on a statistical measure called Information Gain (IG).
- IG - Measures the effectiveness of a particular feature in reducing uncertainty about the classification of data points.
- Attribute with the highest IG is selected as root node.
  i.e Information gain=Parent Entropy – Children Entropy

$$Gain = E_{parent} - E_{children}$$

Example

Predict will **rain or not** based on features: Outlook, Humidity and Wind. Create a suitable decision tree

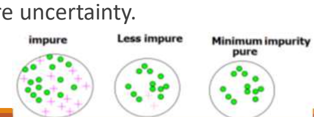
Day	*Outlook*	*Humidity*	*Wind*	*Rainfall*
D1	*Sunny*	*High*	*Weak*	*No*
D2	*Sunny*	*High*	*Strong*	*No*
D3	*Overcast*	*High*	*Weak*	*Yes*
D4	*Rain*	*High*	*Weak*	*Yes*
D5	*Rain*	*Normal*	*Weak*	*Yes*
D6	*Rain*	*Normal*	*Strong*	*No*
D7	*Overcast*	*Normal*	*Strong*	*Yes*
D8	*Sunny*	*High*	*Weak*	*No*
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*

*What prediction would we make for **<outlook=sunny, humidity=high, wind=weak>** ?*

Entropy

A measure of impurity/uncertainty

- For a given node in a decision tree, entropy is calculated based on the distribution of class labels among the instances in that node.
- If a **node is pure** (all instances belong to the same class), the entropy is 0, indicating no uncertainty.
- If a **node is impure** (instances are distributed across multiple classes), the entropy is higher, indicating more uncertainty.



Formula

$E(s) = \Sigma^{c}_{i=1} -p_i \log_2 p_i$

- Where S is a set of training examples,
- c is the number of classes, and
- p_i is the proportion of the training set that is of class I

Parent entropy

- To calculate entropy, start with parent entropy (Eparent) followed by children entropy (Echildren)
- This dataset has a total of **14 rows** of data, **9 yes** classes and **5 no** classes.

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

The **parent entropy** or **(Entropy(rainfall))**=0.94.

Calculating the Entropy of the Training Set.

Children entropy

- Starting with **Outlook** variable:

5 'Sunny' where **2** **'Y'**s and **3** **'N'**s class $\frac{5}{14} * (-\frac{2}{5}log_2\left(\frac{2}{5}\right) - \frac{3}{5}log_2\left(\frac{3}{5}\right))$

$+$

5 'Rainy' **where 3 'Y'**s and **2 'N'**s class $\frac{5}{14} * (-\frac{3}{5}log_2\left(\frac{3}{5}\right) - \frac{2}{5}log_2\left(\frac{2}{5}\right))$

$+$

4 'overcast' where **4'Y'**s $\frac{4}{14} * (-\frac{4}{4}log_2\left(\frac{4}{4}\right))$

- The information necessary to classify the dataset based on outlook (Entropy (outlook) is; 0.347+0.347+0=0.69

- **Info_gain(Outlook)** = **Entropy(parent)**(0.94)- Entropy (outlook)(**0.69**)
- Info_gain(Outlook)= 0.25
- In the same way, we can calculate the Entropy(humidity) and Entropy(wind) which are **0.788** and **0.893** respectively.

  So, the information gain for humidity and wind are:
- Info_gain(humidity) = 0.94–**0.788** = 0.152
- Info_gain(wind) = 0.94–**0.893** = 0.047

- From the calculation above, the attribute with the highest information gain is **Outlook**. So, will be the **root node.** ie. attribute that results in smallest expected size of subtrees rooted at its children
- Entropy is **uncertainty/ randomness** in the data, the more the randomness the higher will be the entropy
- The greater the **information gain,** the more useful the attribute is for making accurate predictions.

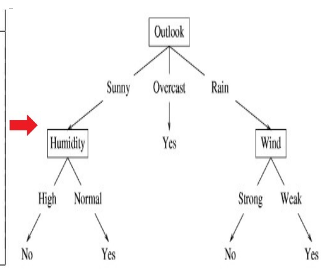
How do we classify it?

1. Outlook =Sunny, Humidity=High, Wind=Strong
2. Outlook = overcast, Humidity=normal, Wind=strong?

Finally the decision tree for the datasets is as shown below;

Decision tree

Day	Outlook	Humidity	Wind	Rainfall
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



Practical exercise

Given the following training set with dogs data:

Color	Fur	Size	Class
brown	ragged	small	well-behaved
black	ragged	big	dangerous
black	smooth	big	dangerous
black	curly	small	well-behaved
white	curly	small	well-behaved
white	smooth	small	dangerous
red	ragged	big	well-behaved