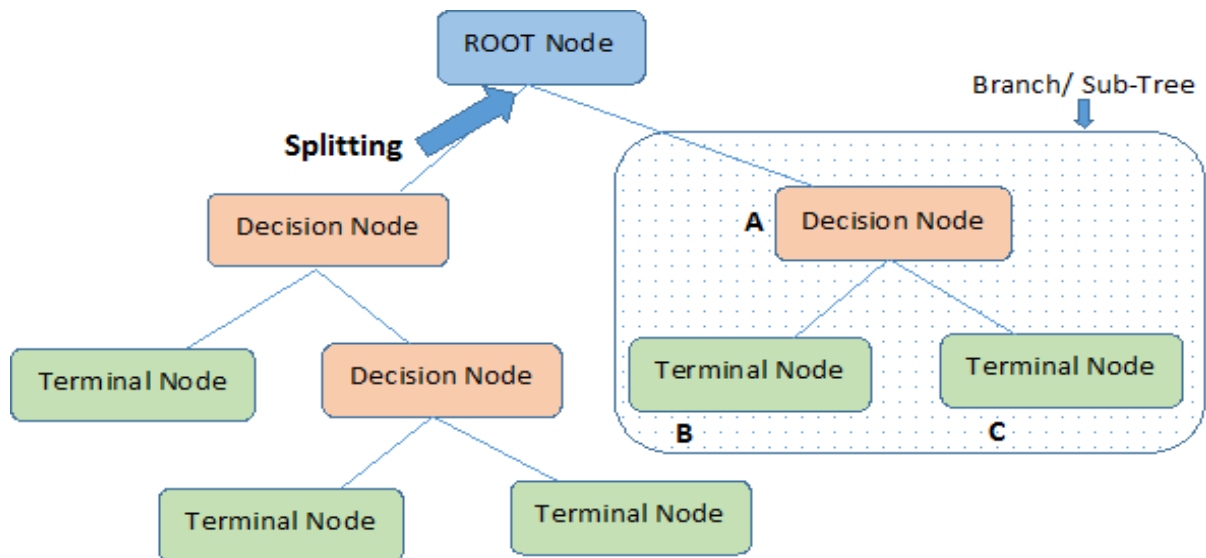


**Contents:**

1. Introduction.....	(2-3)
Graphical Representation of Decision Trees	
Terminologies related to Decision Trees	
Why Decision Trees	
How Decision Tree Algorithm works	
Decision Tree Algorithm pseudo code	
2. Implementation of Decision Tree.....	(4)
Problem Statement	
Data Set Information	
Structure of the Data Set	
Tree Construction performed by Algorithm	
3. Decision Tree Construction (Logic).....	(5-7)
4. Summary.....	8

**Introduction:**

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on **most significant splitter / differentiator** in input variables

**Graphical Representation of Decision Trees:**

**Note:-** A is parent node of B and C.

## Terminologies related to Decision Trees

- 1) **Root Node** - It represents the entire population or sample
- 2) **Splitting** - It is process of dividing node into more sub nodes
- 3) **Decision Node** - When a sub-node splits into further sub-nodes, then it is called decision node.
- 4) **Leaf/Terminal Node** - Nodes do not split and is the final node
- 5) **Pruning** - when we remove sub-nodes of a decision node, the process is called pruning

**Why Decision Trees ?**

Decision Tree Classifier is a supervised learning algorithm used to solve classification and regression problems.

- Decision trees are powerful and popular tools for classification and prediction.
- Decision trees represent *rules*, which can be understood by humans and used in knowledge system such as database.

**How Decision Tree Algorithm works:  
(recursive partitioning algorithm)**

- 1) Decision tree algorithm tries to solve a problem, by using tree representation.
- 2) Each Internal node of the tree represents an attribute (in our model: Humidity, Outlook, Temperature, etc..) and each leaf node corresponds to a class label (in our model: yes/no)

**Decision Tree Algorithm Pseudocode:**

- 1) Place the best Attribute of the data set at the root of the tree.  
Eg: The important feature of the dataset would be usually sit at the root (in our example Outlook)
- 2) Divide the training examples based on selected attributes using statistical measures
- 3) Repeat step1 and step2 until the leaf node is found (i.e in our model: Yes/No)

**Implementation of Decision Tree:**1) Problem Statement:

To predict whether we can play Golf (or) not.

2) Data Set Information:

	Outlook	Temperature	Humidity	Windy	Play
1	Rainy	Hot	High	FALSE	No
2	Rainy	Hot	High	TRUE	No
3	Overcast	Hot	High	FALSE	Yes
4	Sunny	Mild	High	FALSE	Yes
5	Sunny	Cool	Normal	FALSE	Yes
6	Sunny	Cool	Normal	TRUE	No

3) Structure of the Data Set

```
> str(Data)
'data.frame': 14 obs. of 5 variables:
 $ outlook   : Factor w/ 3 levels "Overcast","Rainy",...: 2 2 1 3 3 3 1 2 2 3 ...
 $ Temperature: Factor w/ 3 levels "Cool","Hot","Mild": 2 2 2 3 1 1 1 3 1 3 ...
 $ Humidity   : Factor w/ 2 levels "High","Normal": 1 1 1 1 2 2 2 1 2 2 ...
 $ windy      : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 1 1 2 2 1 1 1 ...
 $ Play       : Factor w/ 2 levels "No","Yes": 1 1 2 2 2 1 2 1 2 2 ...
```

4) Tree Construction performed by Algorithm

C5.0 [Release 2.07 GPL Edition]

Fri Mar 16 12:12:02 2018

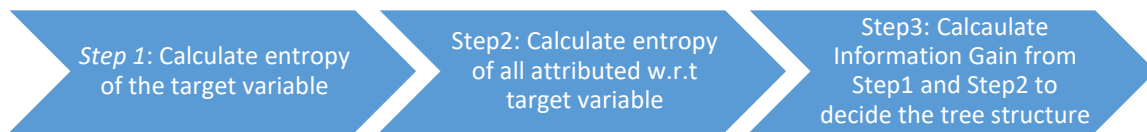
Class specified by attribute 'outcome'

Read 14 cases (5 attributes) from undefined.data

Decision tree:

```
outlook = overcast: Yes (4)
outlook = Rainy:
...Humidity = High: No (3)
: Humidity = Normal: Yes (2)
outlook = Sunny:
...windy = FALSE: Yes (3)
: windy = TRUE: No (2)
```

### Decision Tree Construction:



#### Step 1: Calculate entropy of the target variable:

Mathematical Formula

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

Play Golf	
Yes	No
9	5



$$\begin{aligned} \text{Entropy(PlayGolf)} &= \text{Entropy}(5,9) \\ &= \text{Entropy}(0.36, 0.64) \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94 \end{aligned}$$

#### Step 2: Calculate entropy of all attributes w.r.t target variable:

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

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



$$\begin{aligned} E(\text{PlayGolf}, \text{Outlook}) &= P(\text{Sunny}) * E(3,2) + P(\text{Overcast}) * E(4,0) + P(\text{Rainy}) * E(2,3) \\ &= (5/14) * 0.971 + (4/14) * 0.0 + (5/14) * 0.971 \\ &= 0.693 \end{aligned}$$

#### Step 3: Calculate Information Gain from Step 1 and Step 2:

Eg of Outlook w.r.t target variable is shown here:

$$\text{Gain}(T, X) = \text{Entropy}(T) - \text{Entropy}(T, X)$$

$$\begin{aligned} G(\text{PlayGolf}, \text{Outlook}) &= E(\text{PlayGolf}) - E(\text{PlayGolf}, \text{Outlook}) \\ &= 0.940 - 0.693 = 0.247 \end{aligned}$$

To calculate Information Gain(Required to split the tree) via Mathematical Formulae and Package

>>The below information gives the logic for calculating attribute importance of outlook w.r.t (target variable)

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

$$G(\text{PlayGolf}, \text{Outlook}) = E(\text{PlayGolf}) - E(\text{PlayGolf}, \text{Outlook})$$

$$= 0.940 - 0.693 = 0.247$$



```
> best_features <- information.gain(Play ~ ., Data, unit = "log2")
> best_features
```

	attr_importance
outlook	0.24674982
Temperature	0.02922257
Humidity	0.15183550
windy	0.04812703

Similarly,

>>Calculating Information Gain for all attributes w.r.t target variable(Play Golf)

>>Mathematical way of calculating information gain:

		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	

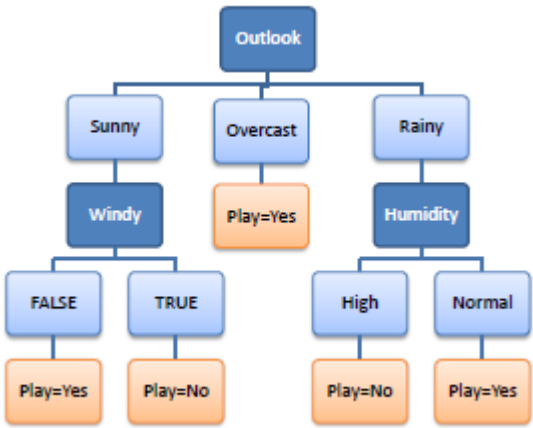
		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
		Gain = 0.048	

Importing Libraries directly to calculate information gain:

```
> best_features
```

	attr_importance
outlook	0.24674982
Temperature	0.02922257
Humidity	0.15183550
windy	0.04812703

### Final Tree Construction:

Decision Tree Construction via Algorithm	Decision Tree Construction via Graphical Representation
<p><b>Decision tree:</b></p> <pre> outlook = Overcast: Yes (4) outlook = Rainy: ...Humidity = High: No (3) : Humidity = Normal: Yes (2) outlook = Sunny: ...Windy = FALSE: Yes (3)   Windy = TRUE: No (2)           </pre> <p>C5.0 [Release 2.07 GPL Edition]</p> <p>-----</p> <p>Class specified by attribute 'outcome'</p>	 <pre> graph TD     Outlook[Outlook] --&gt; Sunny[Sunny]     Outlook --&gt; Overcast[Overcast]     Outlook --&gt; Rainy[Rainy]     Overcast --&gt; PlayYes1[Play=Yes]     Sunny --&gt; Windy[Windy]     Windy --&gt; FALSE[FALSE]     Windy --&gt; TRUE[TRUE]     FALSE --&gt; PlayYes2[Play=Yes]     TRUE --&gt; PlayNo1[Play=No]     Rainy --&gt; Humidity[Humidity]     Humidity --&gt; High[High]     Humidity --&gt; Normal[Normal]     High --&gt; PlayNo2[Play=No]     Normal --&gt; PlayYes3[Play=Yes]           </pre>

### Decision Tree to Decision Rule:

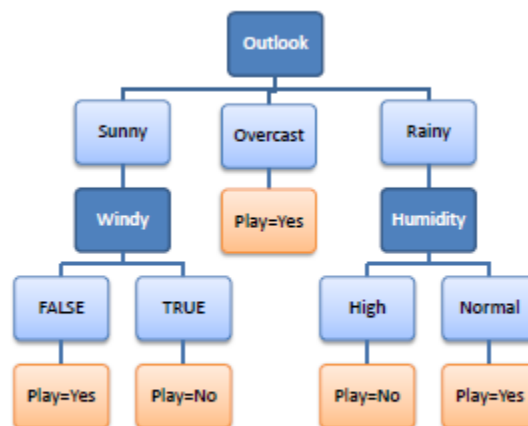
$R_1$ : IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

$R_2$ : 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



Summary:

Once you have decided that Decision tree is the way , then please find the insights of the Algorithm

**Flow Chart:**

**Step 1:- How to find the Root Node**

Use Information gain to understand the each attribute information w.r.t target variable and place the attribute with the highest information gain as root node

**Step 2:- How to Find the Information Gain**

Please apply the entropy(Mathematical Foemulae) to calculate Information Gain.

$$\text{Gain}(T,X) = \text{Entropy}(T) - \text{Entropy}(T,X)$$

here,T represent target variable and X represent features

**Step3: Identification of Terminal Node**

Based on the information gain value obtained from the above steps,identify the second most highest information gain and place it as the terminal node

**Step 4: Predicted Outcome**

Recursively iterate the step4 till we obtain the leaf node which would be our predicted target variable

**Step 5:Tree Pruning and optimization for good results**

It helps to reduce the size of decision trees by removing sections of the tree to avoid overfitting.