In [1]:
```
from IPython.display import Image
Image(filename='logo.PNG', height=340, width=900)
```

Out[1]:

DECISION TREE ALGORITHM

Decision Tree learning

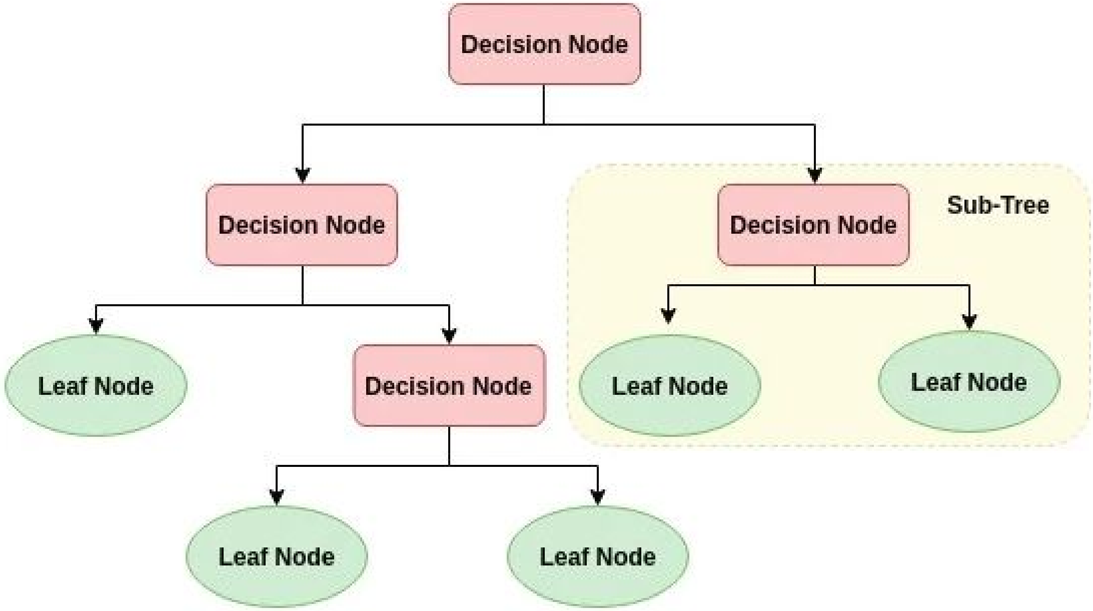
- Decision Tree algorithm belongs to the family of `supervised learning algorithms`
- It is one of the most `widely used practical methods` for inference over supervised data.
- Decision tree algorithm can be used for solving both `regression as well as classification` problems
- A decision tree represents a procedure for classifying `categorical data based on their attributes`
- The construction of decision tree `does not require any domain knowledge or parameter setting`, and therefore appropriate for exploratory knowledge discovery.

- Their representation of acquired knowledge in tree form is `intuitive and easy to assimilate` by humans

In [2]:

```
from IPython.display import Image
Image(filename='DecisionTree_Introduction.jpg')
```

Out[2]:



- A decision tree is a `flowchart-like tree structure` where an internal node represents `feature(or attribute)`, the branch represents a `decision rule`, and each leaf node represents `the outcome`.
- The topmost node in a decision tree is known as the `root node`. It learns to partition on the basis of the attribute value.
- This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are `easy to understand and interpret`.

TYPES OF DECISION TREES

Types of decision tree is based on the type of target variable we have. It can be of two types:

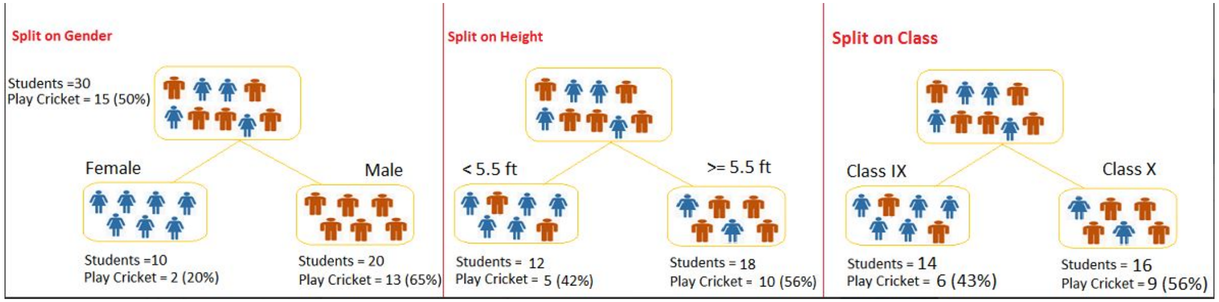
- 1. `Categorical Variable Decision Tree:` Decision Tree which has categorical target variable then it called as **Categorical Variable Decision Tree**.
- 1. `Continuous Variable Decision Tree:` Decision Tree has continuous target variable then it is called as **Continuous Variable Decision Tree**.

Example for Categorical Variable Decision Tree:-

- Let’s say we have a sample of **30 students** with three variables **Gender (Boy/ Girl), Class(IX/ X) and Height (5 to 6 ft)**. **15 out of these 30 play cricket** in leisure time.
- Now, I want to create a model to predict **who will play cricket during leisure period**?
- In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.
- This is where decision tree helps, it will segregate the students based on all values of three variable and identify the variable, which creates the best homogeneous sets of students (which are heterogeneous to each other).
- In the snapshot below, you can see that variable Gender is able to identify best homogeneous sets compared to the other two variables.

In [3]:
```
Image(filename='Example1-Categorical Variable DT.jpg')
```

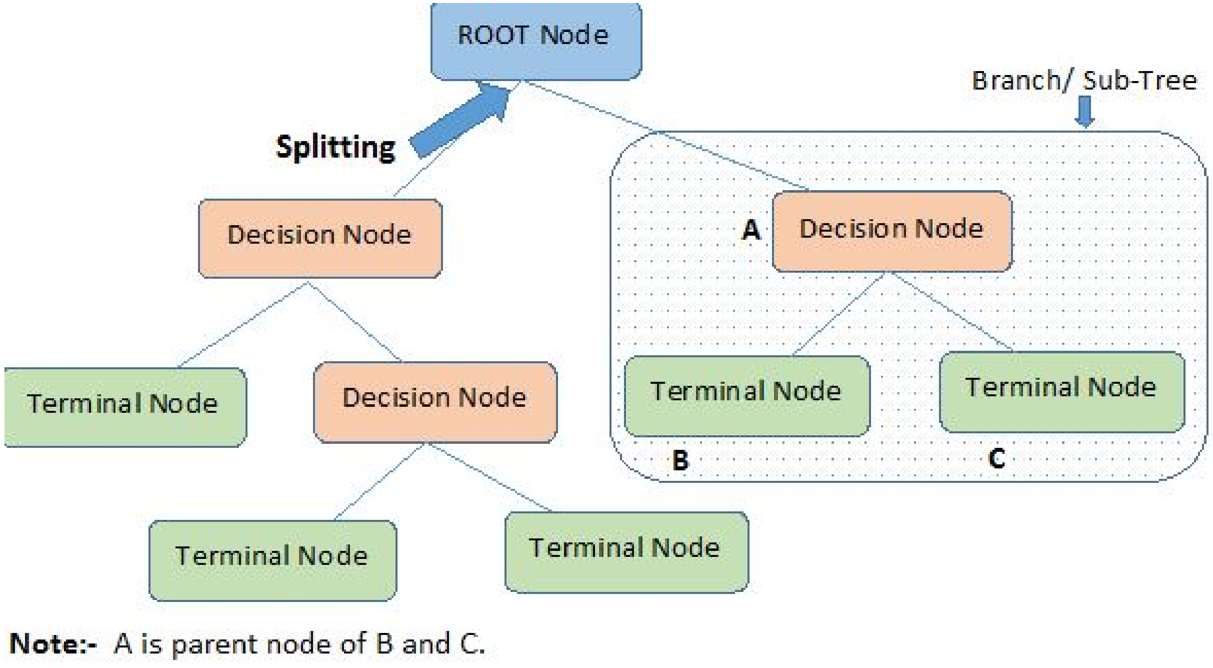
Out[3]:



Terms related to Decision Trees

In [4]:
```
Image(filename='DT_Terms.jpg', width=900)
```

Out[4]:



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

Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.

Splitting: It is a process of dividing a node into two or more sub-nodes.

Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.

Leaf/ Terminal Node: Nodes do not split is called Leaf or Terminal node.

Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.

`Branch / Sub-Tree:` A sub section of entire tree is called branch or sub-tree.

`Parent and Child Node:` A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

ADVANTAGES & DISADVANTAGES OF DECISION TREES

ADVANTAGES **DISADVANTAGES**

Easy to Understand: Decision tree output is very easy to understand even for people from non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis|**Over fitting:** Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning **Useful in Data exploration:** Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict target variable. It can also be used in data exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.| **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree| **Data type is not a constraint:** It can handle both numerical and categorical variables.

HOW DOES A DECISION TREE WORK? HOW DOES A TREE DECIDE WHERE TO SPLIT?

The algorithm selection is also based on type of target variables. Let’s look at the four most commonly used algorithms in decision tree:

01 - GINI INDEX

Gini says, if we select two items from a population at random then they must be of same class and probability for this is 1, if population is pure.

- It works with categorical target variable “Success” or “Failure”.
- It performs only Binary splits
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.

Steps to Calculate Gini for a split

1. `Calculate Gini for sub-nodes`, using formula sum of square of probability for success and failure:

$$(p^2 + q^2)$$

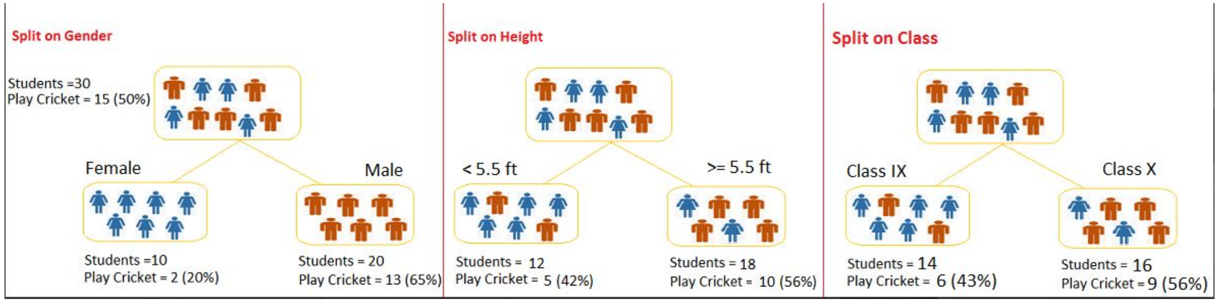
Where,

- **p** is the probability of **"Success"**;
- **q** is the probability of **"Failure"**

1. `Calculate Gini for split` using `weighted Gini score of each node` of that split

In [5]:
```
Image(filename='Example1-Categorical Variable DT.jpg')
```

Out[5]:



`SPLIT ON GENDER:`

In [6]:

```
# GINI for Sub-Node Female:
gf=round((0.2**2)+(0.8**2),2)
print("GINI for Sub-Node Female:", gf)

# GINI for Sub-Node Male:
gm=round((0.65**2)+(0.35**2),2)
print("GINI for Sub-Node Male:", gm)

# Calculate weighted Gini for Split Gender:
weightedgini=round((10/30)*gf + (20/30)*gm,2)
print("Weighted Gini for Gender Split:", weightedgini)
```

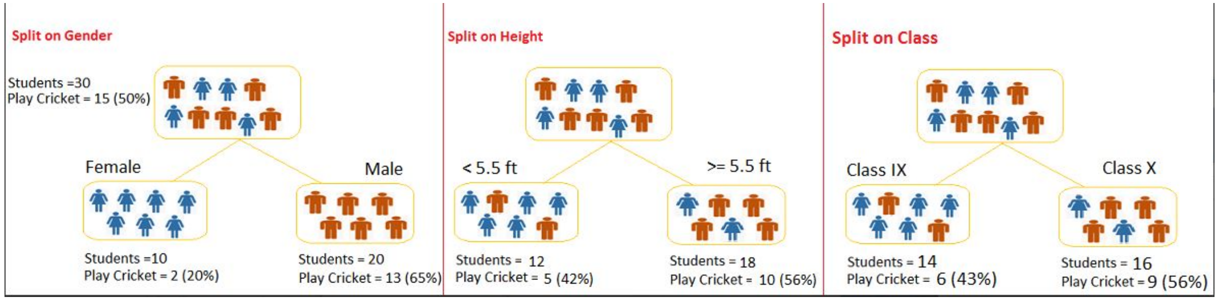
```
GINI for Sub-Node Female: 0.68
GINI for Sub-Node Male: 0.55
Weighted Gini for Gender Split: 0.59
```

SPLIT ON CLASS:

In [7]:

```
Image(filename='Example1-Categorical Variable DT.jpg')
```

Out[7]:



In [8]:

```
# GINI for Sub-Node IX:
g9=round((0.43**2)+(0.57**2),2)
print("GINI for Sub-Node Class IX:", g9)

# GINI for Sub-Node X:
g10=round((0.56**2)+(0.44**2),2)
print("GINI for Sub-Node Class X:", g10)

# Calculate weighted Gini for Split Class:
```

```
weightedgini=round((14/30)*g9 + (16/30)*g10,2)
print("Weighted Gini for Class Split:", weightedgini)
```

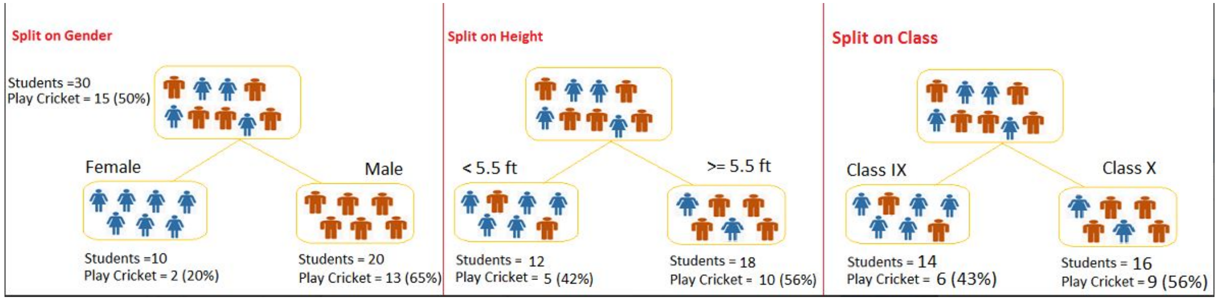
```
GINI for Sub-Node Class IX: 0.51
GINI for Sub-Node Class X: 0.51
Weighted Gini for Class Split: 0.51
```

`SPLIT ON HEIGHT`

In [9]:

```
Image(filename='Example1-Categorical Variable DT.jpg')
```

Out[9]:



In [10]:

```
# GINI for Sub-Node less than 5.5ft:
gless=round((0.42**2)+(0.58**2),2)
print("GINI for Sub-Node Height Less than 5.5ft:", gless)

# GINI for Sub-Node more than 5.5ft:
gmore=round((0.56**2)+(0.44**2),2)
print("GINI for Sub-Node Height More than 5.5ft:", gmore)

# Calculate weighted Gini for Split Height:
weightedgini=round((12/30)*gless + (18/30)*gmore,2)
print("Weighted Gini for Height Split:", weightedgini)
```

```
GINI for Sub-Node Height Less than 5.5ft: 0.51
GINI for Sub-Node Height More than 5.5ft: 0.51
Weighted Gini for Height Split: 0.51
```

Gini score for Split on Gender is higher than Split on Class, hence, the node split will take

place on Gender

You might often come across the term ‘Gini Impurity’ which is determined by subtracting the gini value from 1. So mathematically we can say,

$$GiniImpurity = 1 - Gini$$

Are tree based models better than linear models?

“If I can use logistic regression for classification problems and linear regression for regression problems, why is there a need to use trees”? Many of us have this question. And, this is a valid one too.

Actually, you can use any algorithm. It is dependent on the type of problem you are solving. Let’s look at some key factors which will help you to decide which algorithm to use:

1. If the relationship between dependent & independent variable is well **approximated by a linear model, linear regression will outperform tree based model**.
2. If there is a **high non-linearity & complex relationship** between dependent & independent variables, **a tree model will outperform a classical regression method**.
3. If you need to build a model **which is easy to explain to people**, a decision tree model will always do better than a linear model. Decision tree models are even simpler to interpret than linear regression!

In [ ]: