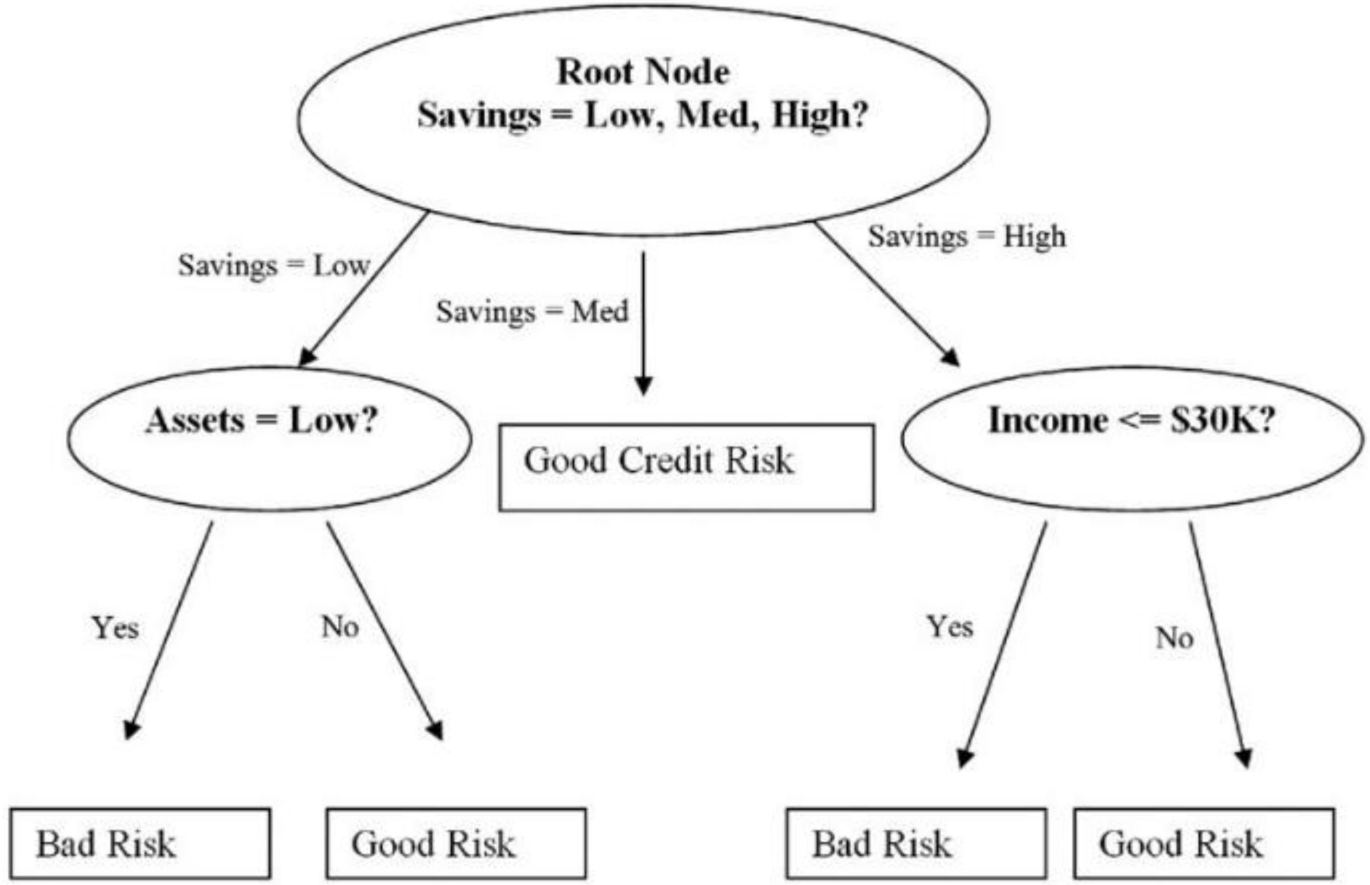
Decision Tree

A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.
A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as **CART (Classification and Regression Trees)**.


Root Node
Savings = Low, Med, High?
Savings = Low
Savings = Med
Savings = High
Assets = Low?
Good Credit Risk
Income <= $30K?
Yes
No
Yes
No
Bad Risk
Good Risk
Bad Risk
Good Risk

Common Terms used with Decision Tree.

1. **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or 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 is called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
6. **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.

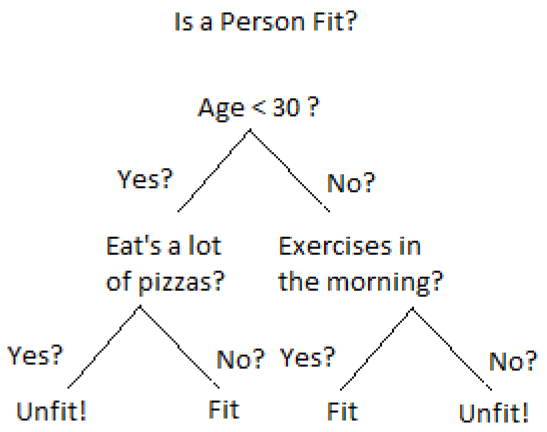
Decision-Tree In-depth

Decision trees have a natural "if … then … else …" construction that makes it fit easily into a programmatic structure. They also are well suited to categorization problems where attributes or features are systematically checked to determine a final category. For example, a decision tree could be used effectively to determine the species of an animal.
The Application of Decision Tree are somehow similar to Logistic Regression. Predicting the classes based on other attributes

Other sophisticated areas of Application

- Evaluation of brand expansion opportunities for a business using historical sales data
- Determination of likely buyers of a product using demographic data to enable targeting of limited advertisement budget
- Prediction of likelihood of default for applicant borrowers using predictive models generated from historical data

How Decision Tree Works

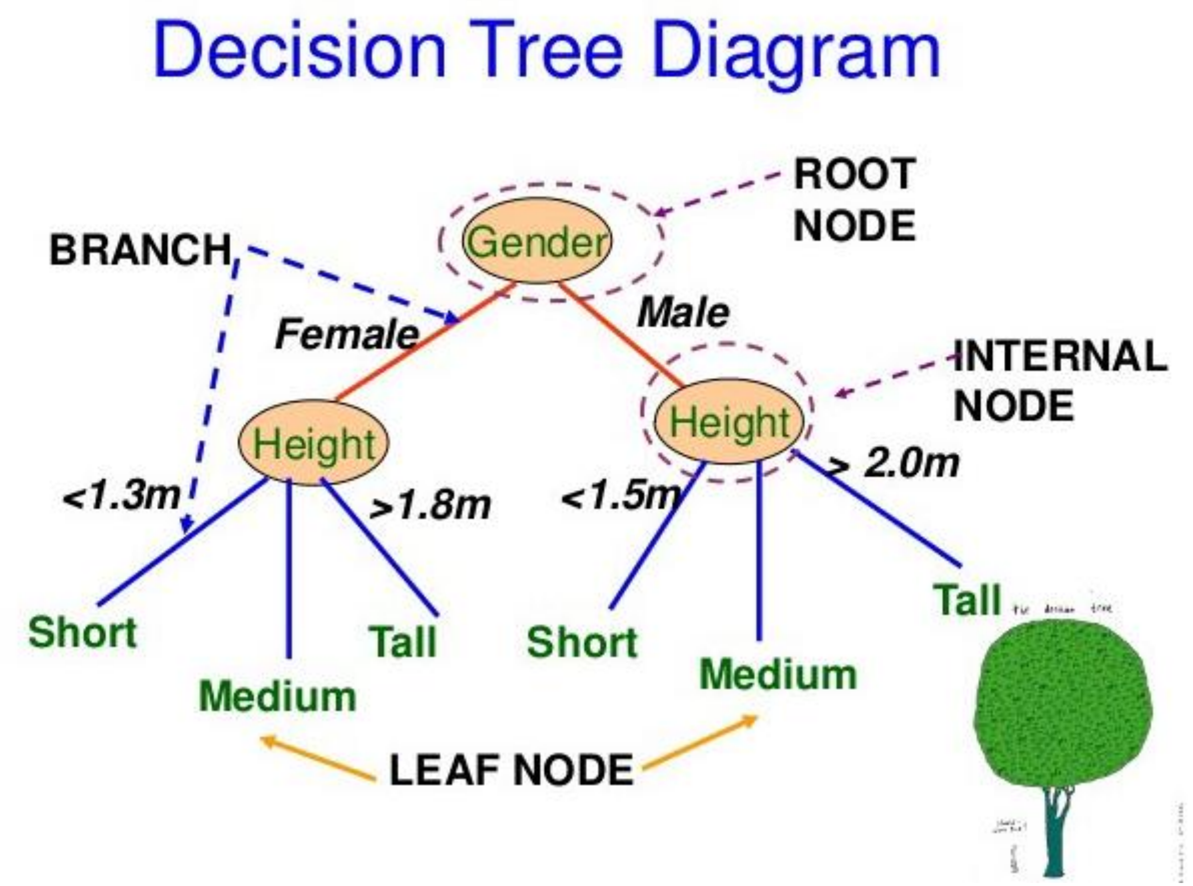


Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets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

Example:-

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, we 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.



Advantages of Decision Tree

- **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.
- Decision trees require relatively **little effort from users for data preparation**.
- **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. Can also *handle multi-output problems.*

Disadvantage of Decision Tree

- **Over fitting:** Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. Over fitting is one of the most practical difficulty for decision tree models
- **Not fit for continuous variables**: While working with continuous numerical variables, decision tree loses information, when it categorizes variables in different categories.
- Calculations can become complex when there are many class label.

How Decision Tree Split into Nodes

Gini Index vs Entropy

Decision trees recursively split features with regard to their target variable's "purity". The entire algorithm is designed to optimize each split on maximizing purity… What is purity? Purity can be thought of as how homogenized the groupings are

If we have 4 red gumballs and 0 blue gumballs, that group of 4 is 100% pure, based on color as the target.
If we have 2 red and 2 blue, that group is 100% impure.
If we have 3 red and 1 blue, that group is either 75% or 81% pure, if we use Gini or Entropy respectively.

GINI Index

1-(yes/(yes+no))sq2 – (no/(yes+no))sq2

After calculating the Gini Index of a attribute, then the weighted average is being calculated for in-depth understanding.

(No. of Instance/Sum of Instances)* Gini Index of attribute's observation

Outlook	Yes	No	No.of Instance
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Outlook	Temp	Humidity	Wind	Decision
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cold	Normal	Weak	Yes
Rain	Cold	Normal	Weak	No
Overcast	Cold	Normal	Strong	Yes
Sunny	Mild	High	Strong	No
Sunny	Cold	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Strong	Yes
Rain	Mild	High	Strong	No

The Table which is right hand side is the major table and the above table is defined from the main table.
The above table will be evaluated for Gini Value
For Sunny
1-(2/5)sq2 – (3/5)sq2 = 0.48
For Overcast
1-(4/4)sq2 – (0/4)sq2 =0
For Rain
1-(3/5)sq2 – (2/5)sq2 =0.48

Now we will calculate the weighted average

((5/14)*0.48) + ((4/14)*0) + ((5/14)*0.48)
0.171 +0+0.171
=0.342 (Gini Index value for Outlook)

With same calculations we will calculate Gini Index Value for Temp variable, Humidity and Wind

For Temp the Gini Value is 0.439
For Humidity the Gini Value is 0.368
For Wind the Gini Value is 0.428
For Outlook the Gini Value was 0.342

Since the Outlook Variable has the most smallest Gini Value among other 3 variables. The Decision Tree will split it's first node on **Outlook.**

With the help of the same calculation the Decision Tree will split other nodes on different variables.

Entropy

Entropy is more computationally heavy due to the log in the equation.

$-P/(P+N) \log2(P/(P+N)) - N/(P+N) \log2(N/(P+N))$

How we can Calculate Entropy value

Step1. Calculate Entropy value for Dependent/Target Variable
-P/(P+N) log2(P/(P+N)) –N/(P+N) log2(N/(P+N))

Step2. Calculate Information Gain for Each Variable
Information Gain= -P/(P+N) log2(P/(P+N)) –N/(P+N) log2(N/(P+N))

Step3. Entropy of the variable's datapoints

Sum of (Pi+Ni)/(P+N) from dependent Variable * Information Gain

Step4. Calculate the Gain Value

Gain=Entropy of Class – Entropy attribute

Hence proved it's bit complicated to split the Decision Tree with Entropy