Tree Based Models

Decision Tree

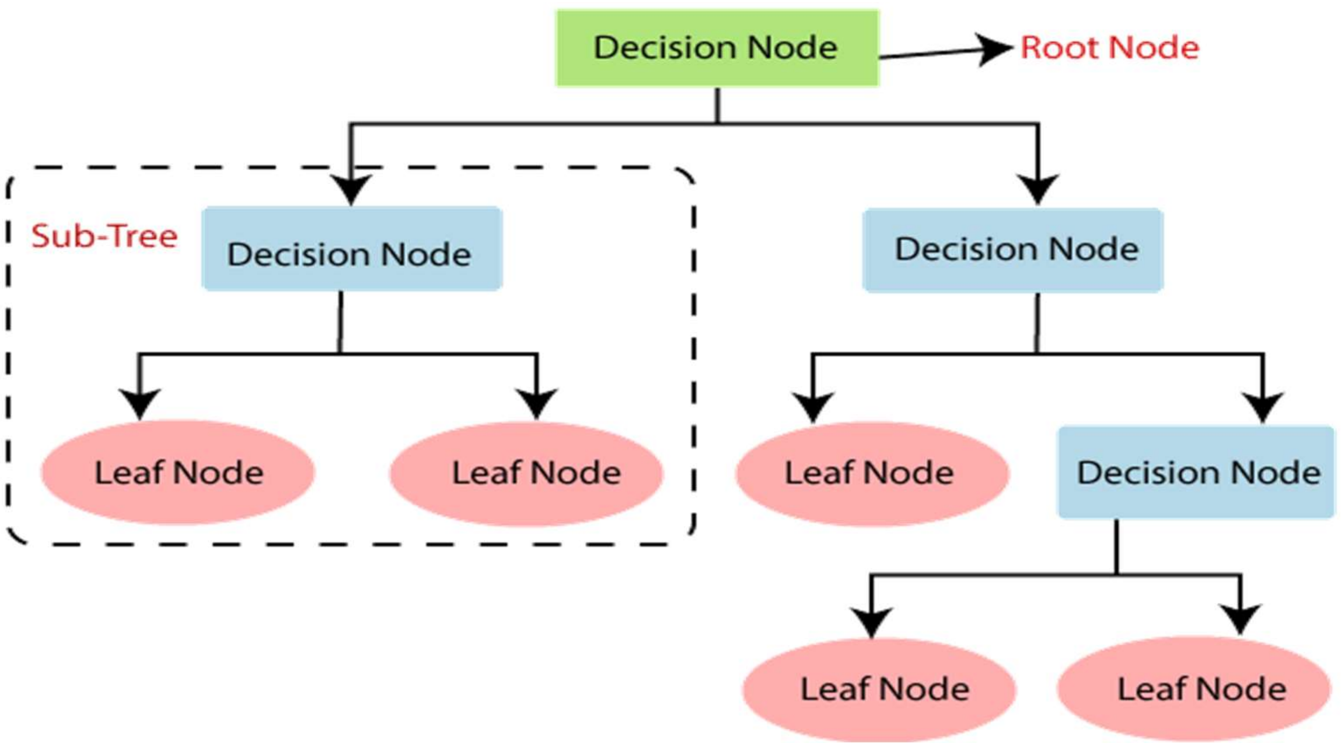
Tree-based models

- Tree-based models use a decision tree to represent how different input variables can be used to predict a target value.
- Machine learning uses tree-based models for both classification and regression problems, such as the type of animal or value of a home.
- The input variables are repeatedly segmented into subsets to build the decision tree, and each branch is tested for prediction accuracy and evaluated for efficiency and effectiveness.
- Splitting the variables in a different order can reduce the number of layers and calculations required to produce an accurate prediction.
- Generating a successful decision tree results in the most important variables (most influential on the prediction) being at the top of the tree hierarchy, while irrelevant features get dropped from the hierarchy.

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** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.**
- 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.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***

General structure of Decision tree



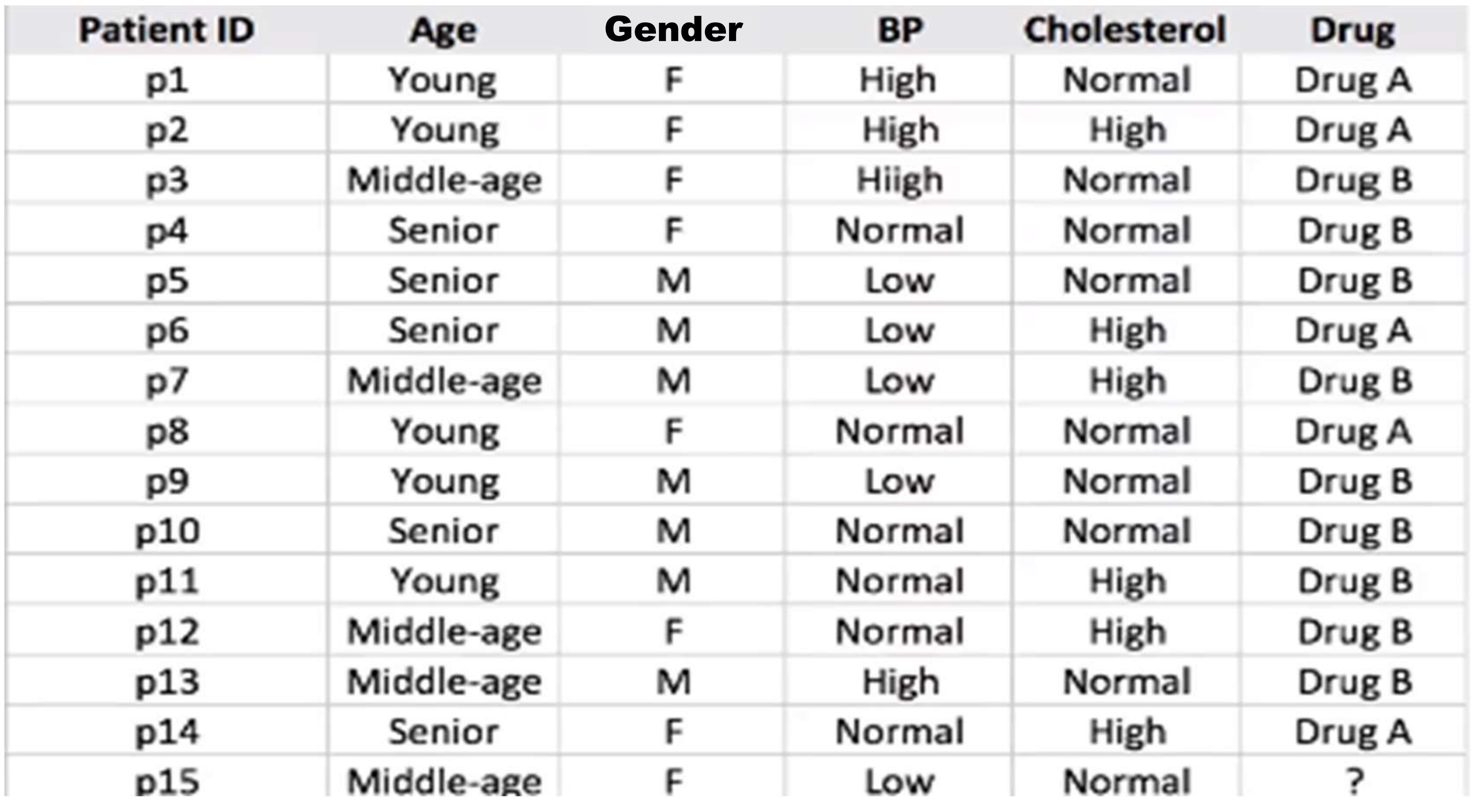
Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

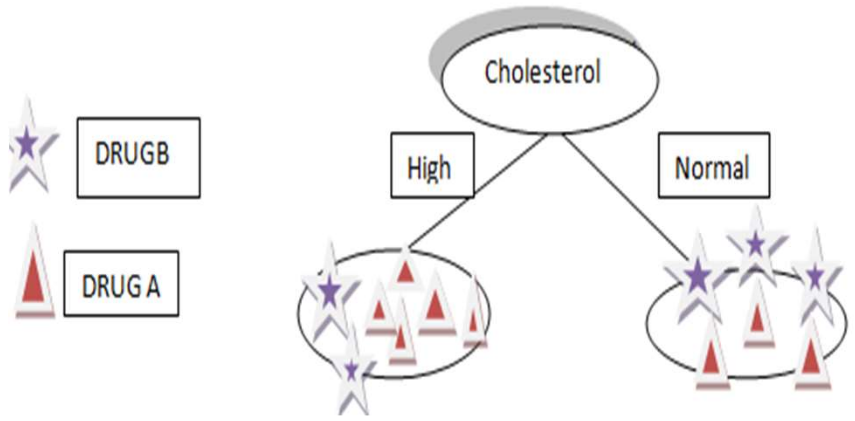
- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example

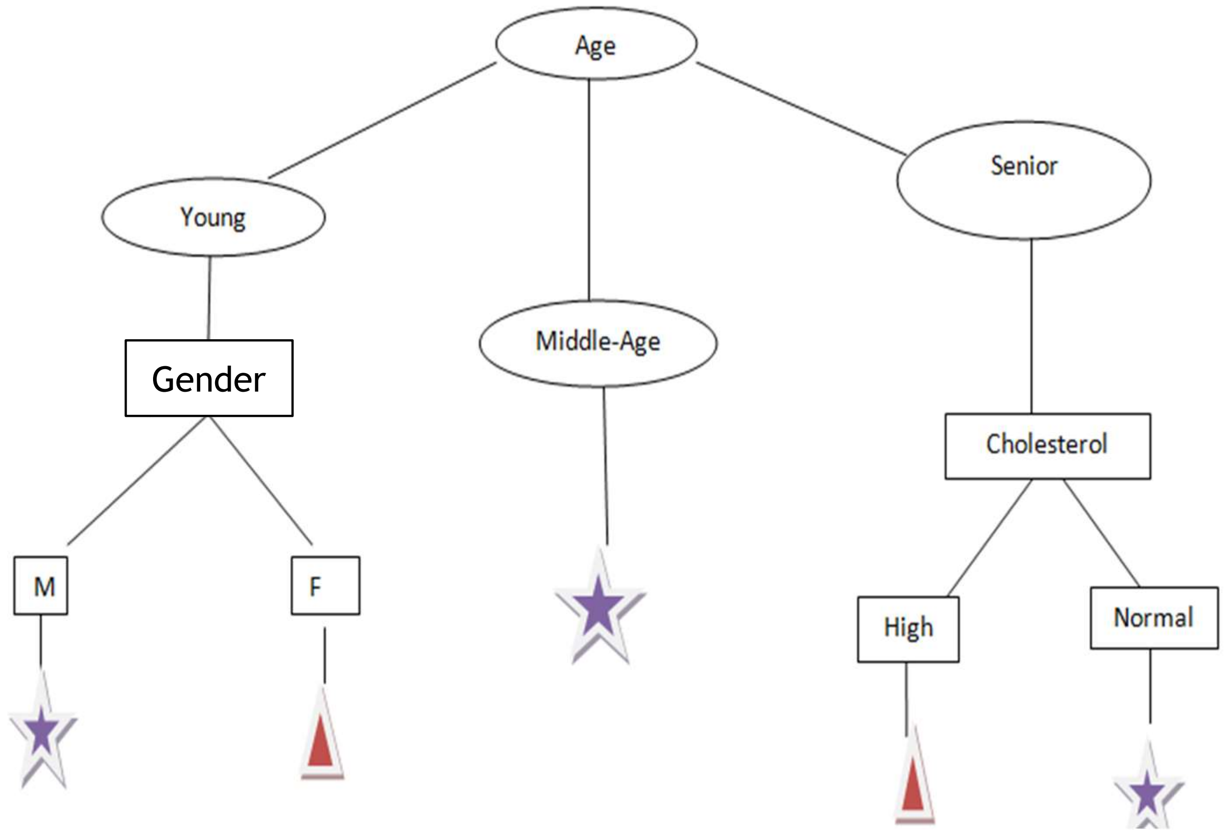


Patient ID	Age	Gender	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A
p15	Middle-age	F	Low	Normal	?

- Suppose we have a sample of 14 patient data set and we have to predict which drug to suggest to the patient A or B.
- Let's say we pick cholesterol as the first attribute to split data.



- It will split our data into two branches High and Normal based on cholesterol, as you can see in the above figure.
- Let's suppose our new patient has high cholesterol by the above split of our data we cannot say **whether** Drug B or Drug A will be suitable for the patient.
- Also, If the patient cholesterol is normal we still do not have an idea or information to determine that either Drug A or Drug B is Suitable for the patient.
- Let us take Another Attribute Age, as we can see age has three categories in it Young, middle age and senior let's try to split.



- From the above figure, Now we can say that we can easily predict which Drug to give to a patient based on his or her reports.

Assumptions that we make while using the Decision tree:

- In the beginning, we consider the whole training set as the root.
- Feature values are preferred to be categorical, if the values continue then they are converted to discrete before building the model.
- Based on attribute values records are distributed recursively.
- We use a statistical method for ordering attributes as a root node or the internal node.

Attribute Selection Measures

- While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes.
- So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.**
- By this measurement, we can easily select the best attribute for the nodes of the tree.
- There are two popular techniques for ASM, which are:
  - **Information Gain**
  - **Gini Index**

Information Gain

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first.
- The split with the highest information gain will be taken as the first split and the process will continue until all children nodes are pure, or until the information gain is 0.

Information Gain

- Information gain computes the difference between entropy before and after split and specifies the impurity in class elements.

Information Gain = Entropy before splitting - Entropy after splitting

- Given a probability distribution such that

$P = (p_1 , p_2 ,\ldots\ldots p_n),$

- and where (p_i) is the probability of a data point in the subset of Di of a dataset D

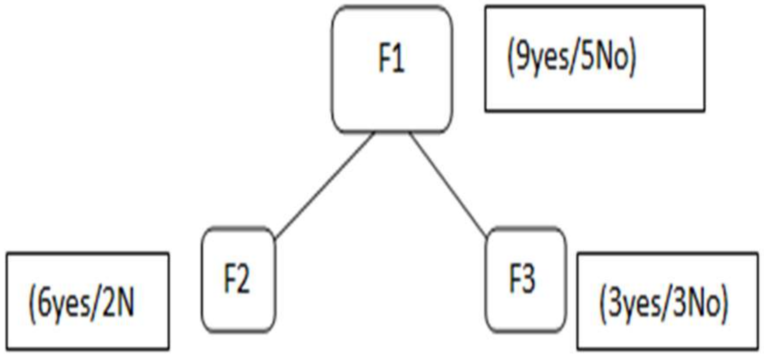
Entropy

- **Mathematics behind Decision tree algorithm:** Before going to the Information Gain first we have to understand entropy
- Entropy: **Entropy** *is the measures of **impurity**, **disorder,** or **uncertainty** in a bunch of examples.*
- ***Purpose of Entropy*:**
- *Entropy controls how a Decision Tree decides to **split** the data. It affects how a **Decision Tree** draws its boundaries.*
- “Entropy values range from 0 to 1”, Less the value of entropy more it is trusting able.

```
Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
```

Where,

- **S= Total number of samples**
- **P(yes)= probability of yes**
- **P(no)= probability of no**



- Suppose we have F1, F2, F3 features we selected the F1 feature as our root node
- F1 contains 9 yes label and 5 no label in it, after splitting the F1 we get F2 which have 6 yes/2 No and F3 which have 3 yes/3 no.
- Now if we try to calculate the Entropy of both F2 by using the Entropy formula...
- Putting the values in the formula:

Entropy

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

$$H(s) = -\frac{6}{8} \log_2 6\backslash 8 - \frac{2}{8} \log_2 \frac{2}{8} = 0.81\ bits$$

- Here, 6 is the number of yes taken as positive as we are calculating probability divided by 8 is the total rows present in the F2.
- Similarly, if we perform Entropy for F3 we will get 1 bit which is a case of an attribute as in it there is 50%, yes and 50% no.
- This splitting will be going on unless and until we get a pure subset.

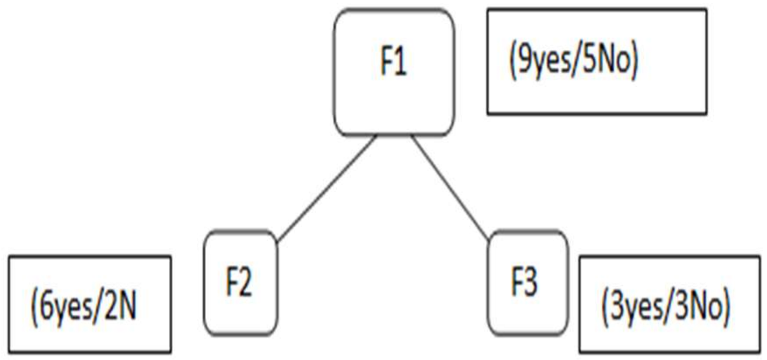
What is a Pure subset?

- The pure subset is a situation where we will get either all yes or all no in this case.
- We have performed this concerning one node what if after splitting F2 we may also require some other attribute to reach the leaf node and we also have to take the entropy of those values and add it up to do the submission of all those entropy values for that we have the concept of information gain.

Information Gain

Formula for Information Gain:

$$Gain(S,A) = H(s) \sum \frac{|Sv|}{|s|} H(Sv)$$



- We can say that in Information gain we are going to compute the average of all the entropy-based on the specific split.
- Sv = Total sample after the split as in F2 there are 6 yes
- S = Total Sample as in F1=9+5=14

Information Gain

- Now calculating the Information Gain:

$$
\begin{aligned}
Gain(S,A) &= H(F1) - \frac{8}{14} * h(F2) - \frac{6}{14} * H(F3) \\
&= 0.91 - \frac{8}{14} * 0.81 - \frac{6}{14} * 1 \\
&= 0.05
\end{aligned}
$$

- **The higher the value of information gain of the split the higher the chance of it getting selected for the particular split.**

Gini Impurity:

- Gini Impurity is a measurement used to build Decision Trees to determine how the features of a data set should split nodes to form the tree.
- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.

$$\text{Gini Index} = 1 - \sum_j P_j^2$$

Gini Index

- The degree of gini index varies from 0 to 0.5,
  - Where 0 depicts that all the elements be allied to a certain class, or only one class exists there.
  - A value of 0.5 denotes the elements are uniformly distributed into some classes.
- Gini index can be calculated using the below formula:

$$Gini(P) = \sum_{i=1}^{n} p_i(1-p_i) = 1 - \sum_{i=1}^{n} (p_i)^2$$

- where P=(p_1 , p_2 ,.......p_n) , and p_i is the probability of an object that is being classified to a particular class.
- Also, an attribute/feature with least Gini index is preferred as root node while making a decision tree.

Entropy vs Gini Impurity

- The maximum value for entropy is 1 whereas the maximum value for Gini impurity is 0.5.
- As the Gini Impurity does not contain any logarithmic function to calculate it takes less computational time as compared to entropy.