Module 3

Decision Tree Algorithm Examples In Data Mining

June 27, 2023

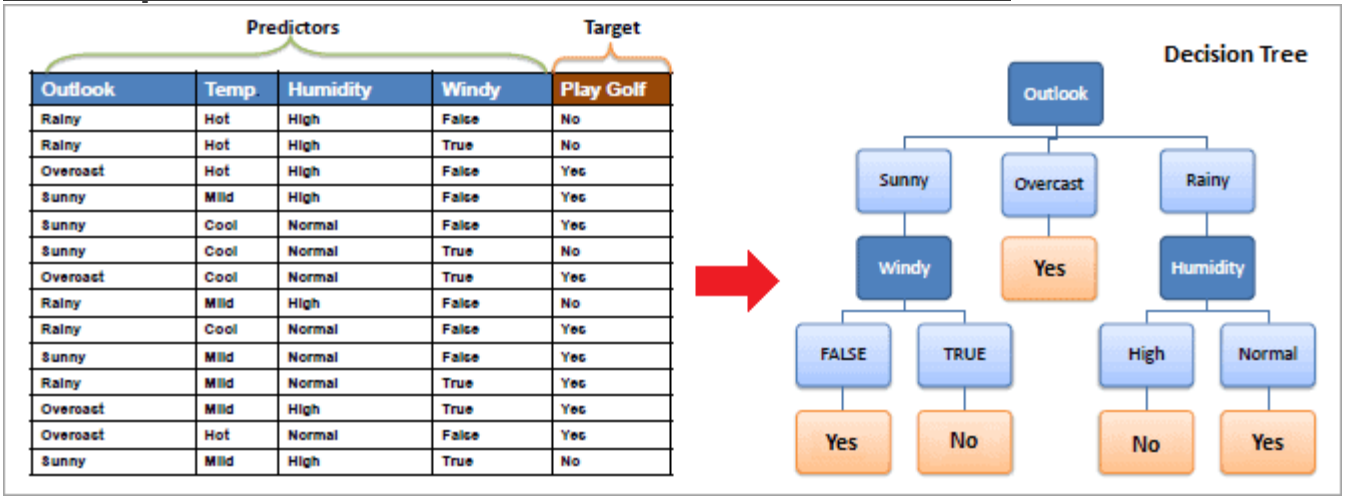
This In-depth Tutorial Explains All About Decision Tree Algorithm In Data Mining. You will Learn About Decision Tree Examples, Algorithm & Classification:

We had a look at a couple of **Data Mining Examples** in our previous tutorial in **Free Data Mining Training Series**.

Decision Tree Mining is a type of data mining technique that is used to build Classification Models. It builds classification models in the form of a tree-like structure, just like its name. This type of mining belongs to supervised class learning.

In supervised learning, the target result is already known. Decision trees can be used for both categorical and numerical data. The categorical data represent gender, marital status, etc. while the numerical data represent age, temperature, etc

An example of a decision tree with the dataset is shown below.



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

What Is The Use Of A Decision Tree?

Decision Tree is used to build classification and regression models. It is used to create data models that will predict class labels or values for the decision-making process. The models are built from the training dataset fed to the system (supervised learning).

Using a decision tree, we can visualize the decisions that make it easy to understand and thus it is a popular data mining technique.

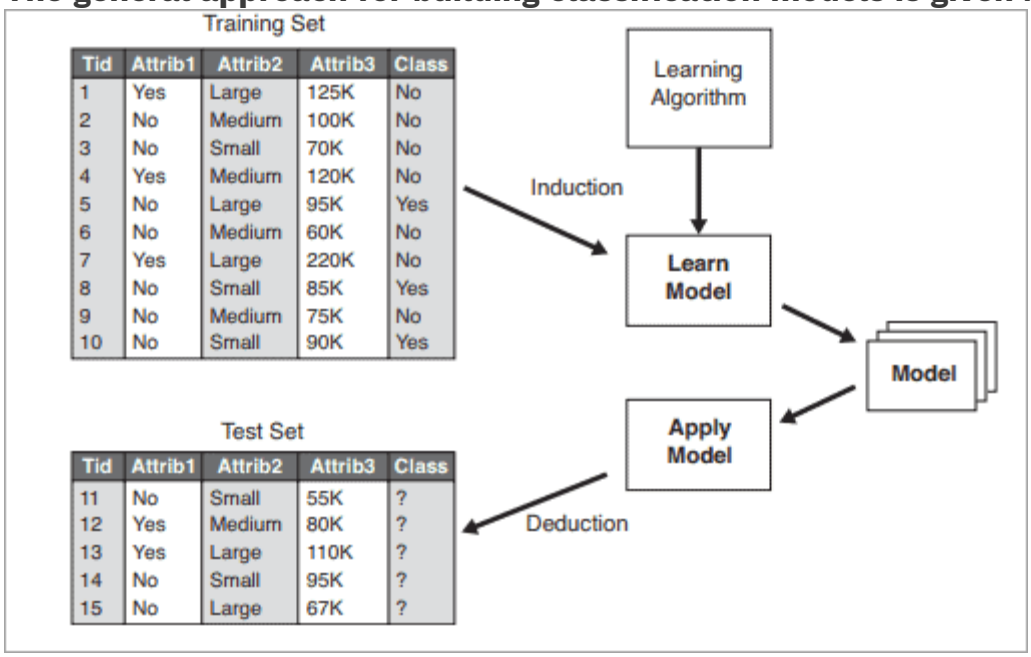
Classification Analysis

Data Classification is a form of analysis which builds a model that describes important class variables. For example, a model built to categorize bank loan applications as safe or risky. Classification methods are used in machine learning, and pattern recognition. Application of classification includes fraud detection, medical diagnosis, target marketing, etc. The output of the classification problem is taken as "Mode" of all observed values of the terminal node.

A two-step process is followed, to build a classification model.

1. In the first step i.e. learning: A classification model based on training data is built.
2. In the second step i.e. Classification, the accuracy of the model is checked and then the model is used to classify new data. The class labels presented here are in the form of discrete values such as "yes" or "no", "safe" or "risky".

The general approach for building classification models is given below:



Regression Analysis

Regression analysis is used for the prediction of numeric attributes.

Numeric attributes are also called continuous values. A model built to predict the continuous values instead of class labels is called the regression model. The output of regression analysis is the "Mean" of all observed values of the node.

How Does A Decision Tree Work?

A decision tree is a supervised learning algorithm that works for both discrete and continuous variables. It splits the dataset into subsets on the basis of the most significant

attribute in the dataset. How the decision tree identifies this attribute and how this splitting is done is decided by the algorithms.

The most significant predictor is designated as the root node, splitting is done to form sub-nodes called decision nodes, and the nodes which do not split further are terminal or leaf nodes.

In the decision tree, the dataset is divided into homogeneous and non-overlapping regions. It follows a top-down approach as the top region presents all the observations at a single place which splits into two or more branches that further split. This approach is also called a *greedy approach* as it only considers the current node between the worked on without focusing on the future nodes.
The decision tree algorithms will continue running until a stop criteria such as the minimum number of observations etc. is reached.

Once a decision tree is built, many nodes may represent outliers or noisy data. Tree pruning method is applied to remove unwanted data. This, in turn, improves the accuracy of the classification model.

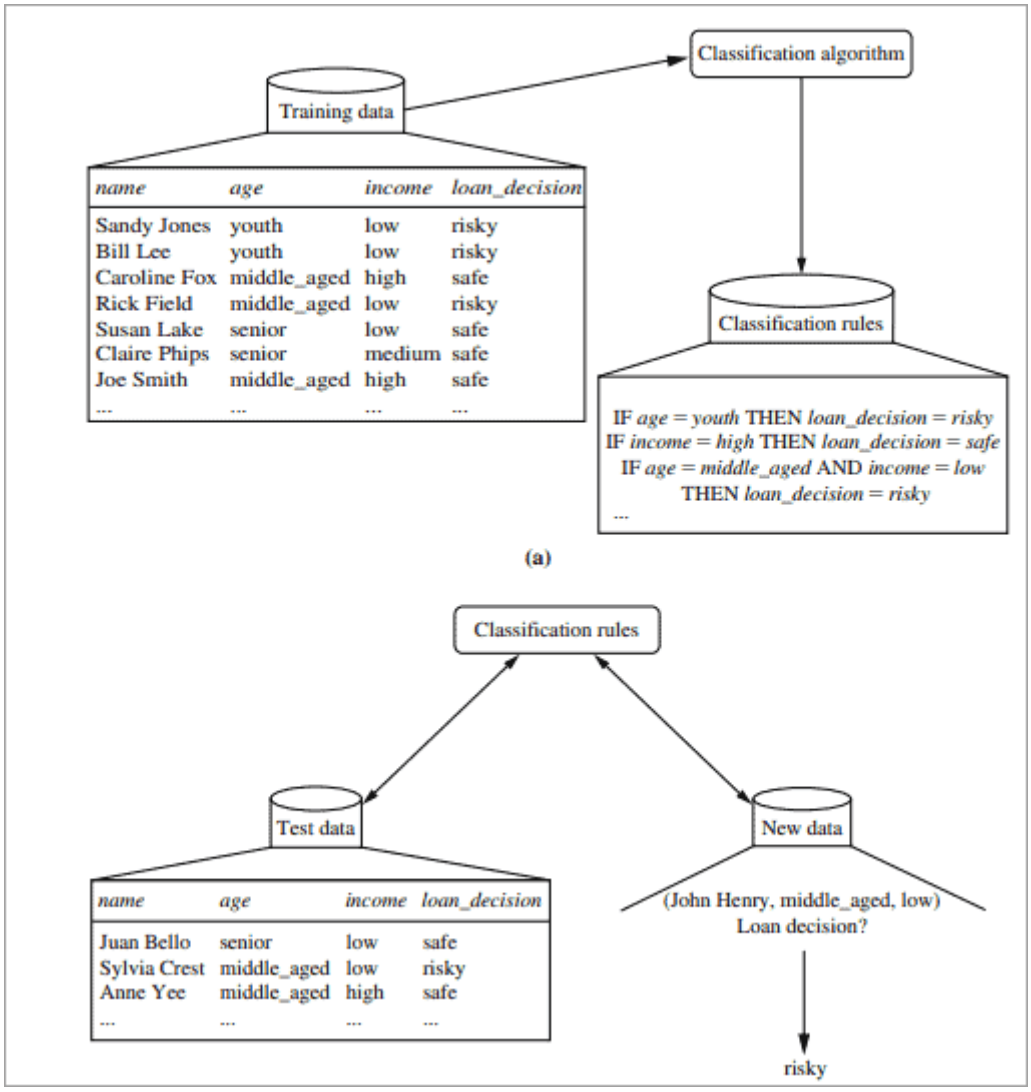
To find the accuracy of the model, a test set consisting of test tuples and class labels is used. The percentages of the test set tuples are correctly classified by the model to identify the accuracy of the model. If the model is found to be accurate then it is used to classify the data tuples for which the class labels are not known.

Some of the decision tree algorithms include Hunt's Algorithm, ID3, CD4.5, and CART.

Example of Creating a Decision Tree
(Example is taken from Data Mining Concepts: Han and Kimber)

#1) Learning Step: The training data is fed into the system to be analyzed by a classification algorithm. In this example, the class label is the attribute i.e. "loan decision". The model built from this training data is represented in the form of decision rules.
#2) Classification: Test dataset are fed to the model to check the accuracy of the classification rule. If the model gives acceptable results then it is applied to a new dataset with unknown class variables.

Semester : V Subject :DWM Academic Year: 2023 - 2024



Decision Tree Induction Algorithm

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition, *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split-point* or *splitting subset*.

Output: A decision tree.

Method:

```
(1)  create a node N;
(2)  if tuples in D are all of the same class, C, then
(3)      return N as a leaf node labeled with the class C;
(4)  if attribute_list is empty then
(5)      return N as a leaf node labeled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  if splitting_attribute is discrete-valued and
         multiway splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)     if Dj is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(Dj, attribute_list) to node N;
     endfor
(15) return N;
```

Decision Tree Induction

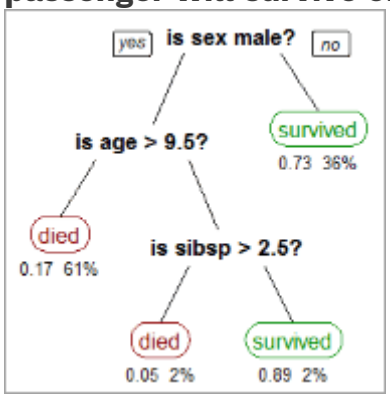
Decision tree induction is the method of learning the decision trees from the training set. The training set consists of attributes and class labels. Applications of decision tree induction include astronomy, financial analysis, medical diagnosis, manufacturing, and production.

A decision tree is a flowchart tree-like structure that is made from training set tuples. The dataset is broken down into smaller subsets and is present in the form of nodes of a tree. The tree structure has a root node, internal nodes or decision nodes, leaf node, and branches.

The root node is the topmost node. It represents the best attribute selected for classification. Internal nodes of the decision nodes represent a test of an attribute of the dataset leaf node or terminal node which represents the classification or decision label. The branches show the outcome of the test performed.

Some decision trees only have *binary nodes*, that means exactly two branches of a node, while some decision trees are non-binary.

The image below shows the decision tree for the Titanic dataset to predict whether the passenger will survive or not.



CART

CART model i.e. Classification and Regression Models is a decision tree algorithm for building models. Decision Tree model where the target values have a discrete nature is called classification models.

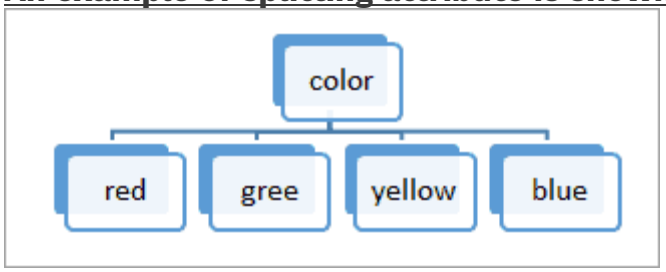
A discrete value is a finite or countably infinite set of values, **For Example,** age, size, etc. The models where the target values are represented by continuous values are usually numbers that are called Regression Models. Continuous variables are floating-point variables. These two models together are called CART.
CART uses Gini Index as Classification matrix.

Decision Tree Induction for Machine Learning: ID3

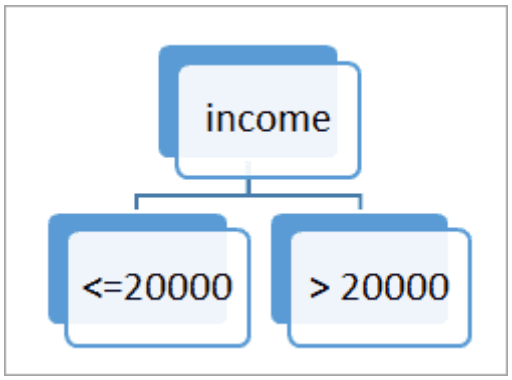
In the late 1970s and early 1980s, J.Ross Quinlan was a researcher who built a decision tree algorithm for machine learning. This algorithm is known as **ID3, Iterative Dichotomiser**. This algorithm was an extension of the concept learning systems described by E.B Hunt, J, and Marin.
ID3 later came to be known as C4.5. ID3 and C4.5 follow a greedy top-down approach for constructing decision trees. The algorithm starts with a training dataset with class labels that are portioned into smaller subsets as the tree is being constructed.

#1) Initially, there are three parameters i.e. **attribute list, attribute selection method and data partition**. The attribute list describes the attributes of the training set tuples.
#2) The attribute selection method describes the method for selecting the best attribute for discrimination among tuples. The methods used for attribute selection can either be Information Gain or Gini Index.
#3) The structure of the tree (binary or non-binary) is decided by the attribute selection method.
#4) When constructing a decision tree, it starts as a single node representing the tuples.

#5) If the root node tuples represent different class labels, then it calls an attribute selection method to split or partition the tuples. The step will lead to the formation of branches and decision nodes.
#6) The splitting method will determine which attribute should be selected to partition the data tuples. It also determines the branches to be grown from the node according to the test outcome. The main motive of the splitting criteria is that the partition at each branch of the decision tree should represent the same class label.
An example of splitting attribute is shown below:



a. The portioning above is discrete-valued.



b. The portioning above is for continuous-valued.

#7) The above partitioning steps are followed recursively to form a decision tree for the training dataset tuples.
#8) The portioning stops only when either all the partitions are made or when the remaining tuples cannot be partitioned further.
#9) The complexity of the algorithm is described by **n * |D| * log |D|** where n is the number of attributes in training dataset D and |D| is the number of tuples.

What Is Greedy Recursive Binary Splitting?

In the binary splitting method, the tuples are split and each split cost function is calculated. The lowest cost split is selected. The splitting method is binary which is formed as 2 branches. It is recursive in nature as the same method (calculating the cost) is used for splitting the other tuples of the dataset.

This algorithm is called as greedy as it focuses only on the current node. It focuses on lowering its cost, while the other nodes are ignored.

How To Select Attributes For Creating A Tree?

Attribute selection measures are also called splitting rules to decide how the tuples are going to split. The splitting criteria are used to best partition the dataset. These measures provide a ranking to the attributes for partitioning the training tuples.

The most popular methods of selecting the attribute are information gain, Gini index.

#1) Information Gain

This method is the main method that is used to build decision trees. It reduces the information that is required to classify the tuples. It reduces the number of tests that are needed to classify the given tuple. The attribute with the highest information gain is selected.

The original information needed for classification of a tuple in dataset D is given by:

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

Where p is the probability that the tuple belongs to class C. The information is encoded in bits, therefore, log to the base 2 is used. E(s) represents the average amount of information required to find out the class label of dataset D. This information gain is also called **Entropy**.

The information required for exact classification after portioning is given by the formula:

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

Where P (c) is the weight of partition. This information represents the information needed to classify the dataset D on portioning by X.

Information gain is the difference between the original and expected information that is required to classify the tuples of dataset D.

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

Gain is the reduction of information that is required by knowing the value of X. The attribute with the highest information gain is chosen as "best".

#2) Gain Ratio

Information gain might sometimes result in portioning useless for classification. However, the Gain ratio splits the training data set into partitions and considers the number of tuples of the outcome with respect to the total tuples. The attribute with the max gain ratio is used as a splitting attribute.

$$\text{Gain Ratio (A)} = \frac{\text{Gain (A)}}{\text{SplitInfo (D)}}$$

#3) Gini Index

Gini Index is calculated for binary variables only. It measures the impurity in training tuples of dataset D, as

$$\text{Gini} = 1 - \sum_i p(i|t)^2$$

P is the probability that tuple belongs to class C. The Gini index that is calculated for binary split dataset D by attribute A is given by:

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

Where n is the nth partition of the dataset D.

The reduction in impurity is given by the difference of the Gini index of the original dataset D and Gini index after partition by attribute A.

The maximum reduction in impurity or max Gini index is selected as the best attribute for splitting.

Overfitting In Decision Trees

Overfitting happens when a decision tree tries to be as perfect as possible by increasing the depth of tests and thereby reduces the error. This results in very complex trees and leads to overfitting.

Overfitting reduces the predictive nature of the decision tree. The approaches to avoid overfitting of the trees include pre pruning and post pruning.

What Is Tree Pruning?

Pruning is the method of removing the unused branches from the decision tree. Some branches of the decision tree might represent outliers or noisy data.

Tree pruning is the method to reduce the unwanted branches of the tree. This will reduce the complexity of the tree and help in effective predictive analysis. It reduces the overfitting as it removes the unimportant branches from the trees.

There are two ways of pruning the tree:

#1) Prepruning: In this approach, the construction of the decision tree is stopped early. It means it is decided not to further partition the branches. The last node constructed becomes the leaf node and this leaf node may hold the most frequent class among the tuples.

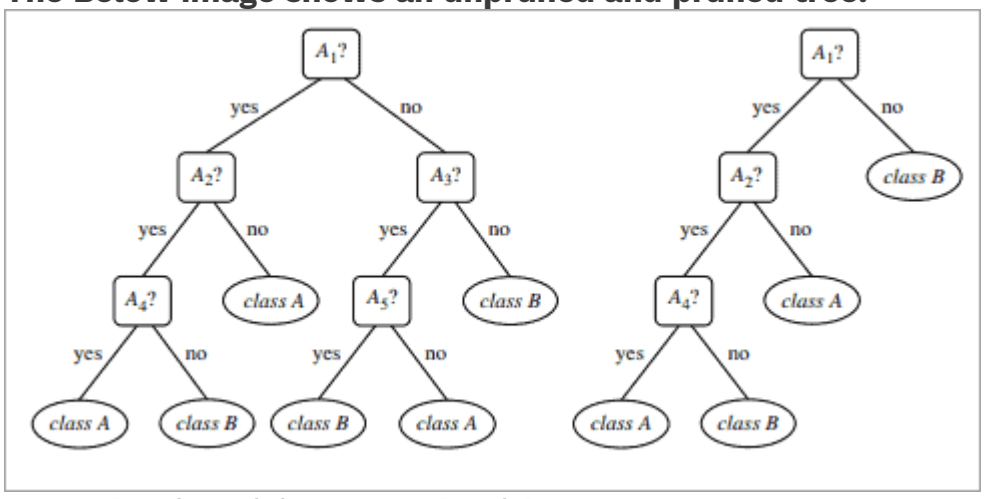
The attribute selection measures are used to find out the weightage of the split. Threshold values are prescribed to decide which splits are regarded as useful. If the portioning of the node results in splitting by falling below threshold then the process is halted.

#2) Postpruning: This method removes the outlier branches from a fully grown tree. The unwanted branches are removed and replaced by a leaf node denoting the most frequent class label. This technique requires more computation than prepruning, however, it is more reliable.

The pruned trees are more precise and compact when compared to unpruned trees but they carry a disadvantage of replication and repetition.

Repetition occurs when the same attribute is tested again and again along a branch of a tree. *Replication* occurs when the duplicate subtrees are present within the tree. These issues can be solved by multivariate splits.

The Below image shows an unpruned and pruned tree.



Example of Decision Tree Algorithm

Example Source

Constructing a Decision Tree

Let us take an example of the last 10 days weather dataset with attributes outlook, temperature, wind, and humidity. The outcome variable will be playing cricket or not. We will use the ID3 algorithm to build the decision tree.

Day	Outlook	Temperature	Humidity	Wind	Play cricket
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Step1: The first step will be to create a root node.
Step2: If all results are yes, then the leaf node "yes" will be returned else the leaf node "no" will be returned.
Step3: Find out the Entropy of all observations and entropy with attribute "x" that is E(S) and E(S, x).
Step4: Find out the information gain and select the attribute with high information gain.
Step5: Repeat the above steps until all attributes are covered.

Calculation of Entropy:

Yes	No
9	5

$$Entropy(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

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

$$= 0.940$$

If entropy is zero, it means that all members belong to the same class and if entropy is one then it means that half of the tuples belong to one class and one of them belong to other class. 0.94 means fair distribution.

Find the information gain attribute which gives maximum information gain.

For Example "Wind", it takes two values: Strong and Weak, therefore, x = {Strong, Weak}.

$$IG(S, Wind) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

Find out H(x), P(x) for x =weak and x= strong. H(S) is already calculated above.

Weak= 8

Strong= 8

$$P(S_{weak}) = \frac{Number\ of\ Weak}{Total}$$

$$= \frac{8}{14}$$

$$P(S_{strong}) = \frac{Number\ of\ Strong}{Total}$$

$$= \frac{6}{14}$$

For "weak" wind, 6 of them say "Yes" to play cricket and 2 of them say "No". So entropy will be:

$$Entropy(S_{weak}) = -\left(\frac{6}{8}\right)\log_2\left(\frac{6}{8}\right) - \left(\frac{2}{8}\right)\log_2\left(\frac{2}{8}\right)$$

$$= 0.811$$

For "strong" wind, 3 said "No" to play cricket and 3 said "Yes".

$$Entropy(S_{strong}) = -\left(\frac{3}{6}\right)\log_2\left(\frac{3}{6}\right) - \left(\frac{3}{6}\right)\log_2\left(\frac{3}{6}\right)$$
$$= 1.000$$

This shows perfect randomness as half items belong to one class and the remaining half belong to others.

Calculate the information gain,

$$IG(S, Wind) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$
$$IG(S, Wind) = H(S) - P(S_{weak}) * H(S_{weak}) - P(S_{strong}) * H(S_{strong})$$
$$= 0.940 - \left(\frac{8}{14}\right)(0.811) - \left(\frac{6}{14}\right)(1.00)$$
$$= 0.048$$

Similarly the information gain for other attributes is:

$$IG(S, Outlook) = 0.246$$
$$IG(S, Temperature) = 0.029$$
$$IG(S, Humidity) = 0.151$$

The attribute outlook has the **highest information gain** of 0.246, thus it is chosen as root. Overcast has 3 values: Sunny, Overcast and Rain. Overcast with play cricket is always "Yes". So it ends up with a leaf node, "yes". For the other values "Sunny" and "Rain".

Table for Outlook as "Sunny" will be:

Temperature	Humidity	Wind	Golf
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

Entropy for "Outlook" "Sunny" is:

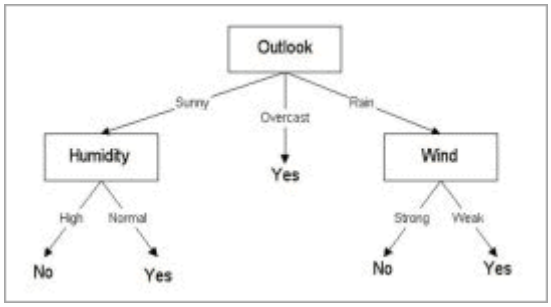
$$H(S_{sunny}) = \left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) = 0.96$$

Information gain for attributes with respect to Sunny is:

$$IG(S_{sunny}, Humidity) = 0.96$$
$$IG(S_{sunny}, Temperature) = 0.57$$
$$IG(S_{sunny}, Wind) = 0.019$$

The information gain for humidity is highest, therefore it is chosen as the next node. Similarly, Entropy is calculated for Rain. **Wind gives the highest information gain**.

The decision tree would look like below:



What Is Predictive Modelling?

The classification models can be used to predict the outcomes of an unknown set of attributes.

When a dataset with unknown class labels is fed into the model, then it will automatically assign the class label to it. This method of applying probability to predict outcomes is called predictive modeling.

Advantages Of Decision Tree Classification

Enlisted below are the various merits of Decision Tree Classification:

1. Decision tree classification does not require any domain knowledge, hence, it is appropriate for the knowledge discovery process.
2. The representation of data in the form of the tree is easily understood by humans and it is intuitive.
3. It can handle multidimensional data.
4. It is a quick process with great accuracy.

Disadvantages Of Decision Tree Classification

Given below are the various demerits of Decision Tree Classification:

1. Sometimes decision trees become very complex and these are called overfitted trees.
2. The decision tree algorithm may not be an optimal solution.
3. The decision trees may return a biased solution if some class label dominates it.

Conclusion

Decision Trees are data mining techniques for classification and regression analysis.

This technique is now spanning over many areas like medical diagnosis, target marketing, etc. These trees are constructed by following an algorithm such as ID3, CART. These algorithms find different ways to split the data into partitions.

It is the most widely known supervised learning technique that is used in machine learning and pattern analysis. The decision trees predict the values of the target variable by building models through learning from the training set provided to the system.