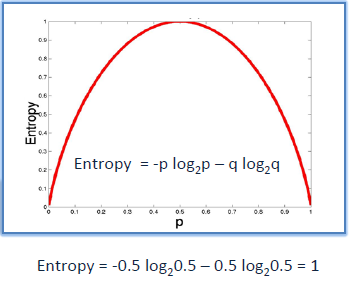
**Decision Tree**

**What is Decision Tree?**

Decision tree is a tree shaped diagram used to determine a course of action. Each branch of the tree represents a possible decision, occurrence or reaction.

**Entropy**

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.



**Algorithm**

The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree

**Problems that Decision Tree can solve.**

1. Classification
2. Regression

A **Classification** tree will determine a set of logical if-then conditions to classify problems. For example, determining between three types of flowers based on certain features.

**Regression** is used when the target variable is numerical or continuous in nature. We fit a regression model to the target variable using each of the independent variables. Each split is made based on the sum of squared error.

**Advantages**

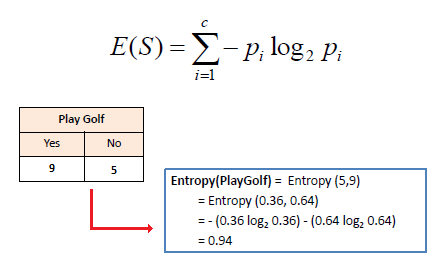
1. Little effort required for data preparation. For example, no data scaling required
2. Can handle both numerical and categorical data.
3. Non-linear parameters don’t affect its performance.

**Disadvantages**

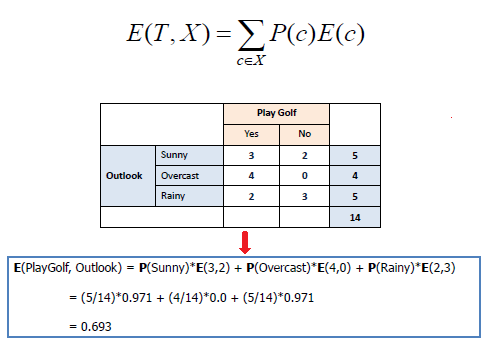
1. Overfitting occurs when the algorithm captures noise in the data
2. High variance- the model can get unstable due to small variation in data.
3. Low biased Tree- A highly complicated tree trends to have a low bias which makes it difficult for the model to work with new data.

To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:

a) Entropy using the frequency table of one attribute:

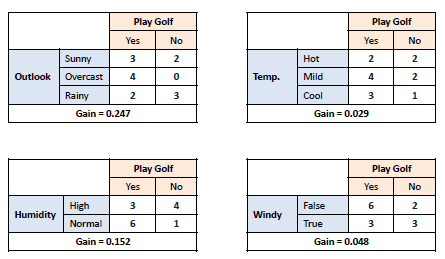


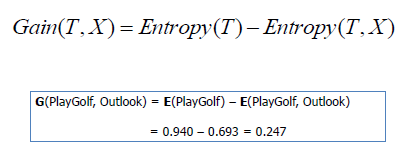
b) Entropy using the frequency table of two attributes:



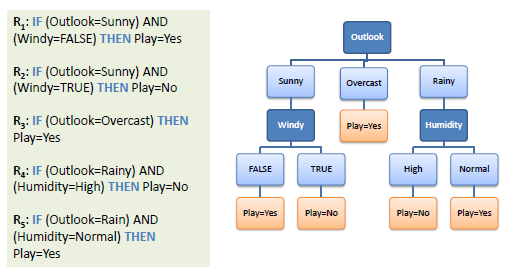
**Information Gain:**

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches)





**Decision Tree to Decision Rules**



**Random Forest Tree**

Random forest decision forest is a method that operates by constructing multiple decision trees during training phase. The Decision of most of the tress is chosen by the random forest as the final decision.

**Advantages:**

1. No Overfitting- Use of multiple risk of over lifting. So, Training time is less
2. High Accuracy- Runs efficiently on large database. For large data, it produces highly accurate predictions. Example, Big data
3. Estimate Missing data- Random forest can maintain accuracy when a large proportion of data is missing.

**Entropy**

Entropyis the measure of randomness or unpredictability in the dataset

**Information Gain**

Informationgain is the measure in entropy after the dataset is slpit.

**Leaf Node**

Leaf node carries the classification decision.

**Decision Node**

Decision node has two or more branches.

**Root Node**

The topmost decision node is knowns as the root node.

**Decision tree in Machine learning using R**

install.packages('rpart')

* Installing the packages from the internet we use this “rpart” library.

library(rpart)

* We use this library in the directory which we are working

help(rpart)

* Details of the rpart library

str(kyphosis)

* Details of the dataset

print(head(kyphosis))

* Print the dataset

tree<-rpart(Kyphosis~.,method = 'class',data=kyphosis)

* Apply the datset in the rpart for the decision tree

printcp(tree)

* print the decision tree

plot(tree,uniform = T,main='kyphosis tree')

* Plot the decision tree with extra featuring to it.

text(tree,use.n = ,all=T)

* Adding the text to the tree for better explanation

install.packages('rpart.plot')

library(rpart.plot)

prp(tree)

**Random Forest tree in Machine learning using R**

install.packages('randomForest')

* Installing the random forest packages in the directory

library(randomForest)

* Adding the library in the directory

rf.model<- randomForest(Kyphosis~.,data = kyphosis)

* Adding the dataset to the random forest library

print(rf.model)

* Printing the model

rf.model$predicted

rf.model$confusion