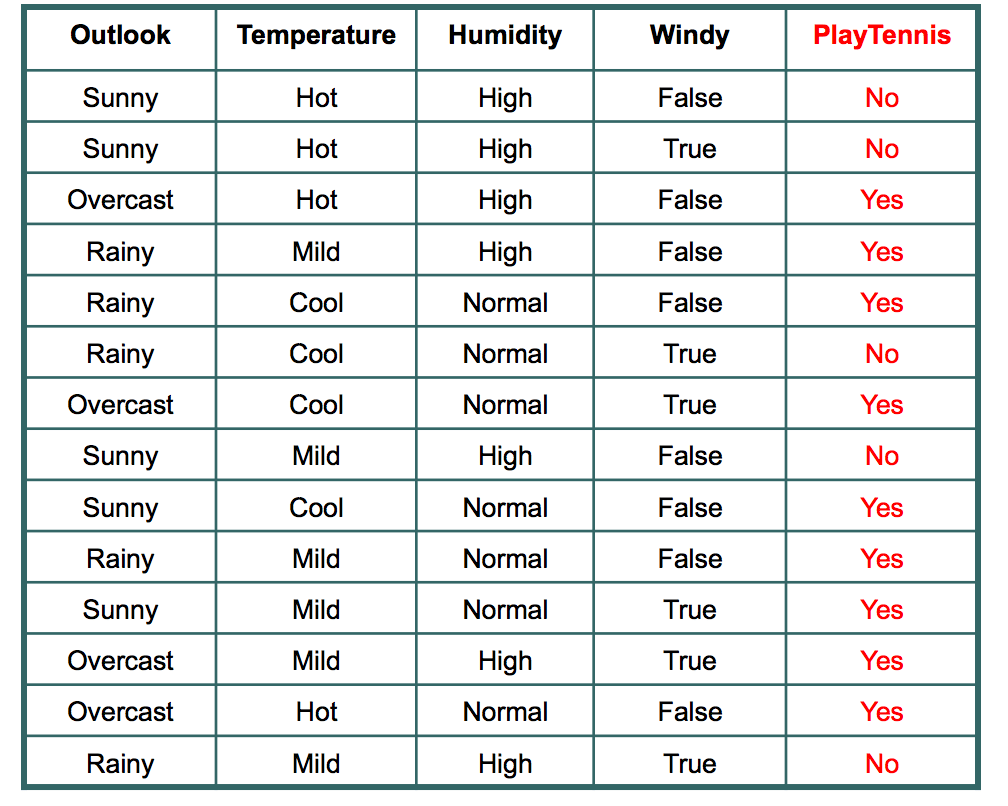
**DECISION TREES**

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

## 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. In ZeroR model there is no predictor, in OneR model we try to find the single best predictor, naive Bayesian includes all predictors using Bayes' rule and the independence assumptions between predictors but decision tree includes all predictors with the dependence assumptions between predictors.

## Entropy

Entropy is a measure of randomness. In other words, its a measure of unpredictability. We will take a moment here to give entropy in case of binary event(like the coin toss, where output can be either of the two events, head or tail) a mathematical face:

Entropy = -(probability(a) \* log2(probability(a))) – (probability(b) \* log2(probability(b)))

*Entropy = -p(1)\*log2(p(1)) -p(2)\*log2(p(2))-p(3)\*log2(p(3))………………………..p(n)\*log(2p(n))*

## Information Gain

It is defined as I.G. = E(Parent) – Weighed sum x E(child)

Where weighed sum = size of child / total size

## HOW TO APPROCH

|  |  |
| --- | --- |
|  |  |
| Step 1: Calculate entropy of the target. |  |  |
|  |  |  |

In the given 14 days, we played tennis on 9 occasions and we did not play on 5 occasions.

Probability of playing tennis:

Number of favourable events : 9

Number of total events : 14

Probability =  (Number of favourable events) / (Number of total events)

= 9/14

= 0.642

Now, we will see probability of not playing tennis.

Probability of not playing tennis:

Number of favourable events : 5

Number of total events : 14

Probability =  (Number of favourable events) / (Number of total events)

=5/14

=0.357

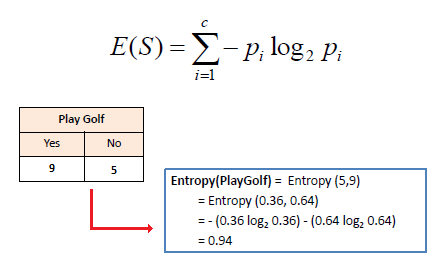
And now entropy of outcome,

Entropy at source= -(probability(a) \* log2(probability(a))) – (probability(b) \* log2(probability(b)))

= -(Probability of playing tennis) \* log2(Probability of playing tennis) – (Probability of not playing tennis) \* log2(Probability of not playing tennis)

= -0.652 \* log2(0.652) – 0.357\*log2(0.357)

=0.940



**So, entropy of whole system before we make our firest question is 0.940**

We will first find I.G. for all the features and max I.G. value secure the first node or root node position.

Now, we have four features to make decision and they are:

1. Outlook
2. Temperature
3. Windy
4. Humidity

Let’s see what happens to entropy when we make our first decision on the basis of Outlook.

## 1.Outlook

### Outlook

If we make a decison tree divison at this level 0 based on outlook, we have three branches possible; either it will be Sunny or Overcast or it will be Raining.

1. Sunny : In the given data, 5 days were sunny. Among those 5 days, tennis was played on 2 days and tenis was not played on 3 days. What is the entropy here?

Probablity of playing tennis = 2/5  = 0.4

Probablity of not playing tennis = 3/5 = 0.6

Entropy when sunny = -0.4 \* log2(0.4) – 0.6 \* log2(0.6)

= 0.97

2. Overcast: In the given data, 4 days were overcast and tennis was played on all the four days. Let

Probablity of playing tennis = 4/4  = 1

Probablity of not playing tennis = 0/4 = 0

Entropy when overcast = 0.0

3. Rain: In the given data, 5 days were rainy. Among those 5 days, tennis was played on 3 days and tenis was not played on 2 days. What is the entropy here?

Probablity of not playing tennis = 2/5  = 0.4

Probablity of playing tennis = 3/5 = 0.6

Entropy when rainy = -0.4 \* log2(0.4) – 0.6 \* log2(0.6)

= 0.97

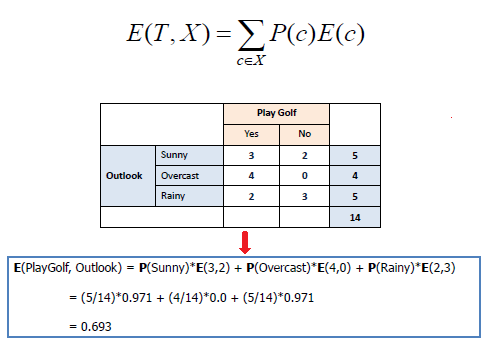
Entropy among the three branches:

Entropy among three branches = ((number of sunny days)/(total days) \* (entropy when sunny)) + ((number of overcast days)/(total days) \* (entropy when overcast)) + ((number of rainy days)/(total days) \* (entropy when rainy))

= ((5/14) \* 0.97) + ((4/14) \* 0) + ((5/14) \* 0.97)

= 0.69

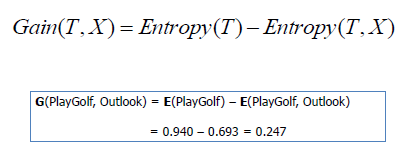
What is the reduction is randomness due to choosing outlook as a decsion maker?



Reduction in randomness = entropy source – entropy of branches

= 0.940 – 0.69

= 0.246



This reduction in randomness is called **Information Gain**. Similar calculation can be done for other features.

## 2.Temperature

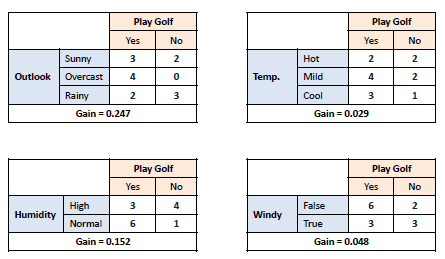
Information Gain = 0.029

## 3.Windy

Information Gain = 0.048

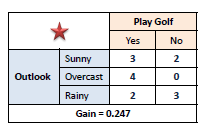
## 4.Humidity

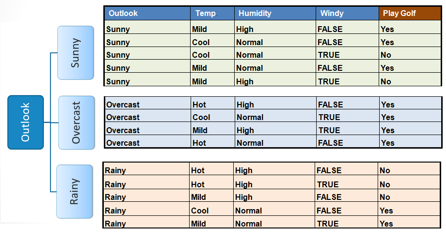
Information Gain = 0.152



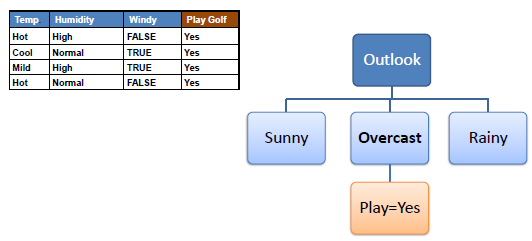
We can see that decrease in randomness, or information gain is most for Outlook. So, we choose root node or first decision maker as **Outlook**.

Step 2: Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

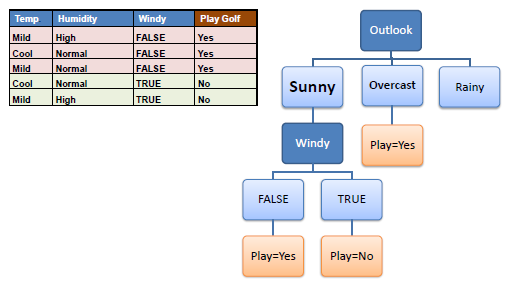




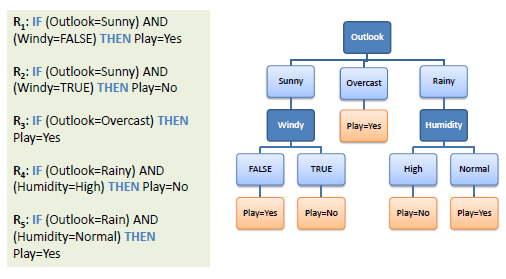
# *Step 3*: A branch with entropy of 0 is a leaf node.



# step 4: A branch with entropy more than 0 needs further splitting



# Step 5: The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified



## Pseudocode

1. Create feature list, attribute list.

Example: Feature List : Outlook, Windy, Temperature and Humidity

Attributes for Outlook are Sunny, Overcast and Rainy.

2.  Find the maximum information gain among all the features. Assign it root node.

Outlook in our example and it has three branches: Sunny, Overcast and Rainy.

3. Remove the feature assigned in root node from the feature list and again find the maximum increase in information gain for each branch. Assign the feature as child  node of each brach and remove that feature from featurelist for that branch.

Sunny Branch for outlook root node has humidity as child node.

4. Repeat step 3 until you get branches with only pure leaf. In our example, either yes  or no.