**What is a Decision Tree?**

A Supervised Machine Learning Algorithm, used to build classification and regression models in the form of a tree structure.

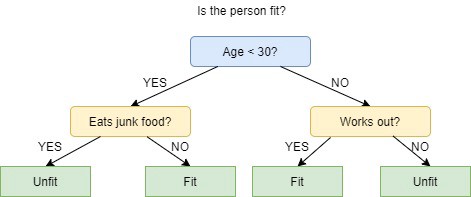
A decision tree is a tree where each –

* Node - a feature(attribute)
* Branch - a decision(rule)
* Leaf - an outcome (categorical or continuous)

**What are Decision Trees?**

In simple words, a decision tree is a structure that contains nodes (rectangular boxes) and edges(arrows) and is built from a dataset (table of columns representing features/attributes and rows corresponds to records). Each node is either used to ***make a decision****(*known as decision node)or ***represent an outcome***(known as leaf node).

**Decision tree Example**



The picture above depicts a decision tree that is used to classify whether a person is **Fit**or **Unfit.**

The decision nodes here are questions like ‘*Is the person less than 30 years of age?’*, *‘Does the person eat junk?’*, etc.andthe leaves are one of the two possible outcomes viz. **Fit**and **Unfit**.

The initial node is called the **root node** (coloured in blue), the final nodes are called the **leaf nodes**(coloured in green) and the rest of the nodes are called **intermediate**or **internal**nodes.

The root and intermediate nodes represent the decisions while the leaf nodes represent the outcomes.

## **What are appropriate problems for Decision tree learning?**

Although a variety of decision tree learning methods have been developed with somewhat differing capabilities and requirements, decision tree learning is generally best suited to problems with the following characteristics:

**1. Instances are represented by attribute-value pairs.**

“Instances are described by a fixed set of attributes (e.g., Temperature) and their values (e.g., Hot). The easiest situation for decision tree learning is when each attribute takes on a small number of disjoint possible values (e.g., Hot, Mild, Cold). However, extensions to the basic algorithm allow handling real-valued attributes as well (e.g., representing Temperature numerically).”

**2. The target function has discrete output values.**

“The decision tree is usually used for Boolean classification (e.g., yes or no) kind of example. Decision tree methods easily extend to learning functions with more than two possible output values. A more substantial extension allows learning target functions with real-valued outputs, though the application of decision trees in this setting is less common.”

**3. Disjunctive descriptions may be required.**

Decision trees naturally represent disjunctive expressions.

**4. The training data may contain errors.**

“Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.”

**5. The training data may contain missing attribute values.**

“Decision tree methods can be used even when some training examples have unknown values (e.g., if the Humidity of the day is known for only some of the training examples).”

**There are 4 popular types of decision tree algorithms:**

1. ID3
2. CART (Classification and Regression Trees),
3. Chi-Square and
4. Reduction in Variance.

# What is an ID3 Algorithm?

ID3 stands for **Iterative Dichotomiser 3**

It is a classification algorithm that follows a greedy approach by selecting a best attribute that yields maximum Information Gain (IG) or minimum Entropy(H).

## **What is Entropy and Information gain?**

**Entropy is a measure of the amount of uncertainty in the dataset S. Mathematical Representation of Entropy is shown here -**

**Entropy(S) = ∑ – p(I). log2p(I)**

**Gain (S, A) = Entropy(S) – ∑ [ p(S|A). Entropy(S|A)]**

**Where,**

* S - The current dataset for which entropy is being calculated (changes every iteration of the ID3 algorithm).
* l - Set of classes in S {example - l = {yes, no}}
* p(l) - The proportion of the number of elements in class l to the number of elements in set S.

**What are the steps in ID3 algorithm?**

The **steps in ID3 algorithm** are as follows:

1. Calculate entropy for dataset.
2. For each attribute/feature.  
   2.1. Calculate entropy for all its categorical values.  
   2.2. Calculate information gain for the feature.
3. Find the feature with maximum information gain.
4. Make a decision tree node using the feature with the maximum Information gain.
5. If all rows belong to the same class, make the current node as a leaf node with the class as its label.
6. Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

#### **Data set**

For instance, the [following table](https://www.goodreads.com/book/show/213030.Machine_Learning) informs about decision making factors to play tennis at outside for previous 14 days.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 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 |

We can summarize the ID3 algorithm as illustrated below

Entropy(S) = ∑ – p(I). log2p(I)

Gain (S, A) = Entropy(S) – ∑ [ p(S|A). Entropy(S|A)]

## **Entropy**

We need to calculate the entropy first. Decision column consists of 14 instances and includes two labels: yes and no. There are 9 decisions labelled yes, and 5 decisions labelled no.

Entropy (Decision) = – p(Yes). log2p(Yes) – p(No). log2p(No)

Entropy (Decision) = – (9/14). log2(9/14) – (5/14). log2(5/14) = 0.940

Now, we need to find the most dominant factor for decisioning.

## **Wind factor on decision**

Gain (Decision, Wind) = Entropy (Decision) – ∑ [ p(Decision|Wind). Entropy(Decision|Wind) ]

**Wind attribute has two labels: weak and strong.**

We would reflect it to the formula.

Gain (Decision, Wind) = Entropy (Decision) – [ p(Decision|Wind=Weak). Entropy (Decision|Wind=Weak)] – [ p(Decision|Wind=Strong). Entropy (Decision|Wind=Strong)]

Now, we need to calculate (Decision|Wind=Weak) and (Decision|Wind=Strong) respectively.

### **Weak wind factor on decision**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 1 | Sunny | Hot | High | Weak | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |

There are 8 instances for weak wind. Decision of 2 items are no and 6 items are yes as illustrated below.

1- Entropy (Decision|Wind=Weak) = – p(No). log2p(No) – p(Yes). log2p(Yes)

2- Entropy (Decision|Wind=Weak) = – (2/8). log2(2/8) – (6/8). log2(6/8) = 0.811

### **Strong wind factor on decision**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 2 | Sunny | Hot | High | Strong | No |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 14 | Rain | Mild | High | Strong | No |

Here, there are 6 instances for strong wind. Decision is divided into two equal parts.

1- Entropy (Decision|Wind=Strong) = – p(No). log2p(No) – p(Yes). log2p(Yes)

2- Entropy (Decision|Wind=Strong) = – (3/6). log2(3/6) – (3/6). log2(3/6) = 1

Now, we can turn back to Gain (Decision, Wind) equation.

Gain (Decision, Wind) = Entropy (Decision) – [ p(Decision|Wind=Weak). Entropy (Decision|Wind=Weak)] – [ p(Decision|Wind=Strong). Entropy (Decision|Wind=Strong)] = 0.940 – [ (8/14). 0.811] – [ (6/14). 1] = 0.048

Calculations for wind column is over. Now, we need to apply same calculations for other columns to find the most dominant factor on decision.

**Other factors on decision**

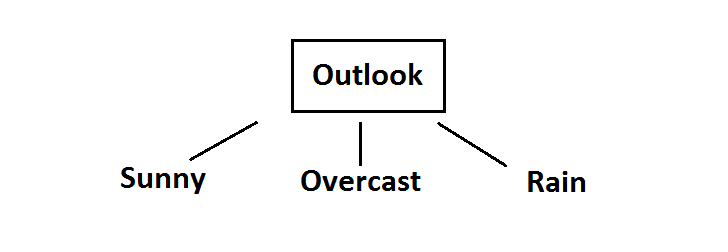
We have applied similar calculation on the other columns.

1- Gain (Decision, Outlook) = 0.246

2- Gain (Decision, Temperature) = 0.029

3- Gain (Decision, Humidity) = 0.151

As seen, outlook factor on decision produces the highest score. That’s why, outlook decision will appear in the root node of the tree.

**Fig: Root decision on the tree**

Now, we need to test dataset for custom subsets of outlook attribute.

## **Overcast outlook on decision**

Basically, decision will always be yes if outlook were overcast.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 3 | Overcast | Hot | High | Weak | Yes |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |

## **Sunny outlook on decision**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |

Here, there are 5 instances for sunny outlook. Decision would be probably 3/5 percent no, 2/5 percent yes.

1- Gain (Outlook=Sunny|Temperature) = 0.570

2- Gain (Outlook=Sunny|Humidity) = 0.970

3- Gain (Outlook=Sunny|Wind) = 0.019

Now, humidity is the decision because it produces the highest score if outlook were sunny.

At this point, decision will always be no if humidity were high.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 8 | Sunny | Mild | High | Weak | No |

On the other hand, decision will always be yes if humidity were normal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |

Finally, it means that we need to check the humidity and decide if outlook were sunny.

**Rain outlook on decision**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

1- Gain (Outlook=Rain | Temperature) = 0.01997309402197489

2- Gain (Outlook=Rain | Humidity) = 0.01997309402197489

3- Gain (Outlook=Rain | Wind) = 0.9709505944546686

Here, wind produces the highest score if outlook were rain. That’s why, we need to check wind attribute in 2nd level if outlook were rain.

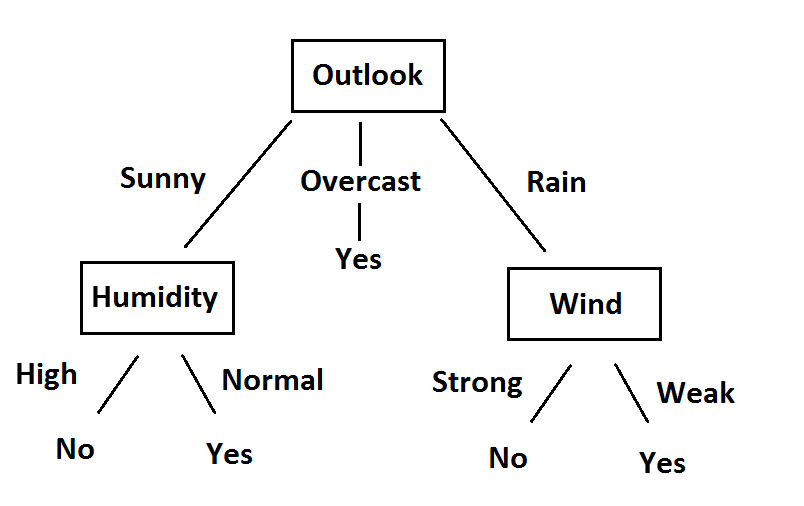
So, it is revealed that decision will always be yes if wind were weak and outlook were rain.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |

What’s more, decision will be always no if wind were strong and outlook were rain.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 6 | Rain | Cool | Normal | Strong | No |
| 14 | Rain | Mild | High | Strong | No |

So, decision tree construction is over. We can use the following rules for decisioning.



**Fig:** Final version of decision tree