These two models are often used in **classifiers**.

$\equiv$ a machine learning system that makes decisions on input.

the inputs are called characteristics or instance.

the output is called a decision.

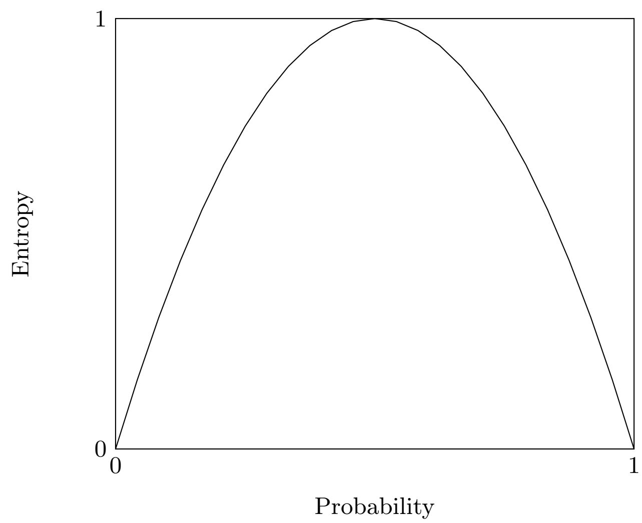
We can construct them from Bayesian Networks.

We must first introduce the idea of **entropy**.

$$\text{ENT}(X) = -\sum_X Pr(x)log_2(Pr(x))$$

Notice that this is identical to the cross entropy between any distribution and itself.
We can show it with the following data table:

	E	B	A
T	0.1	0.2	0.2442
F	0.9	0.8	0.7556
ENT	0.469	0.722	0.802



The form we tend to utilize is **CONDITIONAL PROBABILITY**.
If we have ENT(X) and we learn that Y=y, we have

$$E(X|y) = -\sum_X Pr(x|y)log_2(Pr(x|y))$$

Alternatively, if we plan to observe Y but do not yet know the value

$$E(X|Y) = -\sum_Y Pr(y)\text{ENT}(X|y)$$

It also turns out that information can never increase average entropy, ie

$$\text{ENT}(X|Y) \leq \text{ENT}(X)$$

Note that this specifies average; the entropy of a single value may increase:

	B	B|A	B|$\neg A$
T	0.2	0.741	0.025
F	0.8	0.259	0.975
ENT	0.722	0.825	0.169

ENT(X|Y) = ENT(B|a) Pr(a) + ENT(B|$\bar{a}$) Pr($\bar{a}$) = 0.328 $\leq$ 0.722

These are used to build classifiers by supervised learning of labeled data.
Our CPT thus effectively functions as our model.

We will now use the notion of a decision tree/random forest to solve a problem.
We will use the following data and corresponding tree:

![$Wehave12labeledvariables$]($https://paper-attachments.dropbox.com/s_66C470E465947B218F12E6833ACF54B222DBA6F$

This model is called **interpretable** because it is easy to read, as opposed to a neural network.
Classifying a variable is as easy as parsing the tree!

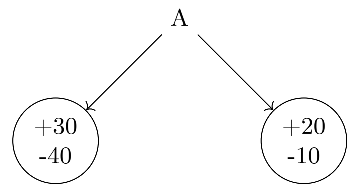
Consider X_{12} — we can just walk; this probability happens to match, but it won't be in general.

![]($https : //paper - attachments.dropbox.com/s_6 6C470E465947B218F12E6833ACF54B222DBA6F153DEEC1F4A1A4D0690$

The depth of the decision tree is a sign of its complexity.
Splitting is as easy as making a choice.
Nodes represent attributes.
Leaves represent decisions.

We can equivalently build:
this has 4 attributes rather than the 10 from above
this is much shallower, and thus simpler

The algorithm itself is very simple;
we just split repeatedly as if tracing the tree.
This assumes a black box for choosing variables,
but developing one is not hard
How do we choose which attribute to split on at a given depth?
We use conditional entropy as a score to determine our next split.
A snapshot of the algorithm is as right.



M	
HI	30/70
LO	40/70

ENT = 0.935

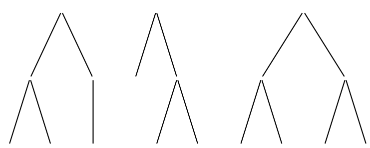
M	
HI	20/30
LO	10/30

ENT = 0.918

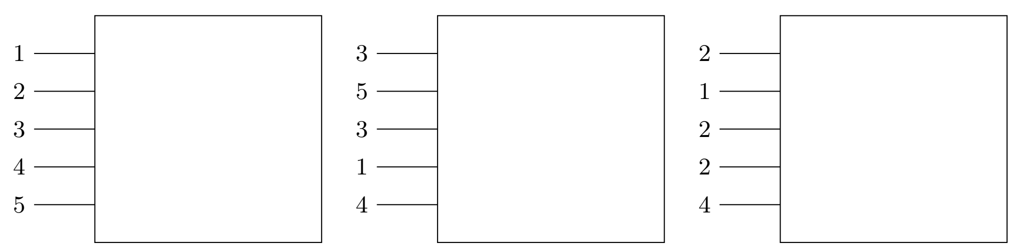
$$\implies \text{ENT}(\text{M}|\text{A}) = (0.7)(0.985) + (0.3)(0.918) = 0.965$$

How do we evaluate an algorithm? We use **cross-validation**.
≡ split the dataset into 80/20 training/testing data & repeat to find average score.

This can be generalized one more time to a **random forest**.
We build a series of trees and majority vote to determine the output.
We call this type of method an ensemble learning method.

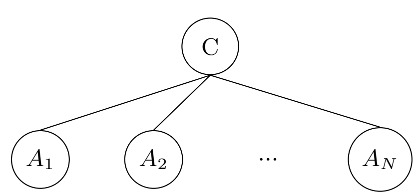


Suppose we have a dataset of 5 values; we may bootstrap data sets by random choice to get:



The count of numbers chosen will be a parameter.
We can test the power using the out of bag examples.

Bayesian Network Classifiers



We set a threshold T to classify inputs st

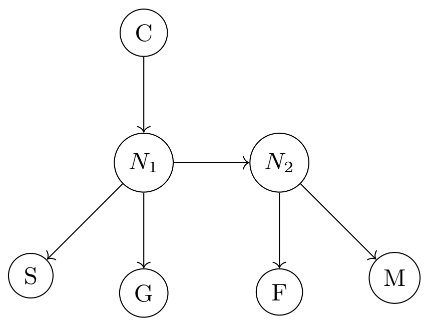
$$C = \left\{ \begin{array}{ll} c & \text{iff } Pr(C|a_1, a_2, ..., a_N) \geq T \\ \neg c & \text{iff } Pr(C|a_1, a_2, ..., a_N) < T \end{array} \right\}$$

A specific subset of these are called **naive**.
These assume the independence of attributes given the parent. If we interpret the above as naive, then

$$
\begin{aligned}
Pr(c|a_1, a_2, ..., a_N) &= \frac{Pr(a_1, a_2, ..., a_N)Pr(c)}{Pr(a_1, a_2, ..., a_N} \\
&= \frac{Pr(a_1|c)Pr(a_2|c)...Pr(a_N|c)Pr(c)}{Pr(a_1, a_2, ..., a_N|c)Pr(c) + Pr(a_1, a_2, ..., a_N|\neg c)Pr(\neg c)} \\
&= \frac{\left[\prod_{i=1}^{N} Pr(a_i|c)\right] Pr(c)}{Pr(\cap_{i=1}^{N} a_i|c)Pr(c) + Pr(\cap_{i=1}^{N} a_i|\neg c)Pr(\neg c)}
\end{aligned}
$$

We can thus observe this directly from the tree!

Traditionally, we want AI to be easily explainable.
Consider the following example:



S	G	F	M	C
-	-	-	-	-
-	-	-	+	+
...	...	...	...	...
+	+	+	+	+

Say we are asked C|{S=1, G=0, F=1, M=1}.
We might say "yes, because F & M!", as S & G are not used.
This is called a **PI-explanation**.
In actuality, we can make a tractable circuit from this data, which is much power powerful.
This, however, is not as interpretable!
In the current day, Random Forests < Bayesian Classifiers < Neural Networks.