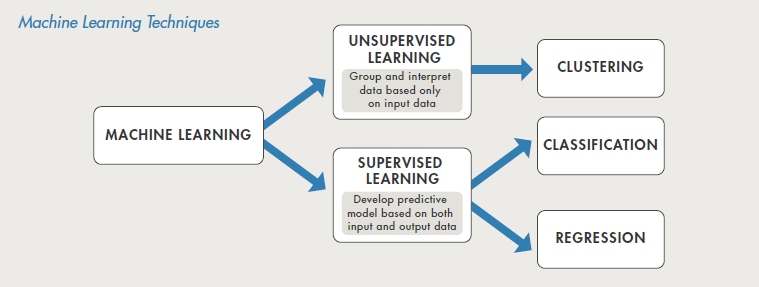
**Cross validation** allows us to compare different machine learning methods and get a sense of how well they will work in practice.

We can use 75% of the data for training and 25% of the data for testing. We could then compare methods by seeing how well each one categorized the test data.

Machine learning teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases.

Machine learning uses two types of techniques:

* Supervised Learning, which trains a model on known input and output data so that it can predict future outputs
* Unsupervised Learning, which finds hidden patterns or essential structures in input data.



**Supervised Learning**

The aim of supervised machine learning is to build a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (outputs) and trains a model to generate reasonable predictions for the response to new data.

Supervised learning uses classification and regression techniques to develop predictive models.

* **Classification techniques** predict discrete responses – for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition and credit scoring.
* **Regression techniques** predict continuous responses – for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading.

**Unsupervised Learning**

Unsupervised learning finds hidden patterns or essential structures in data. It is used to draw inferences from datasets consisting of input data without labeled responses.

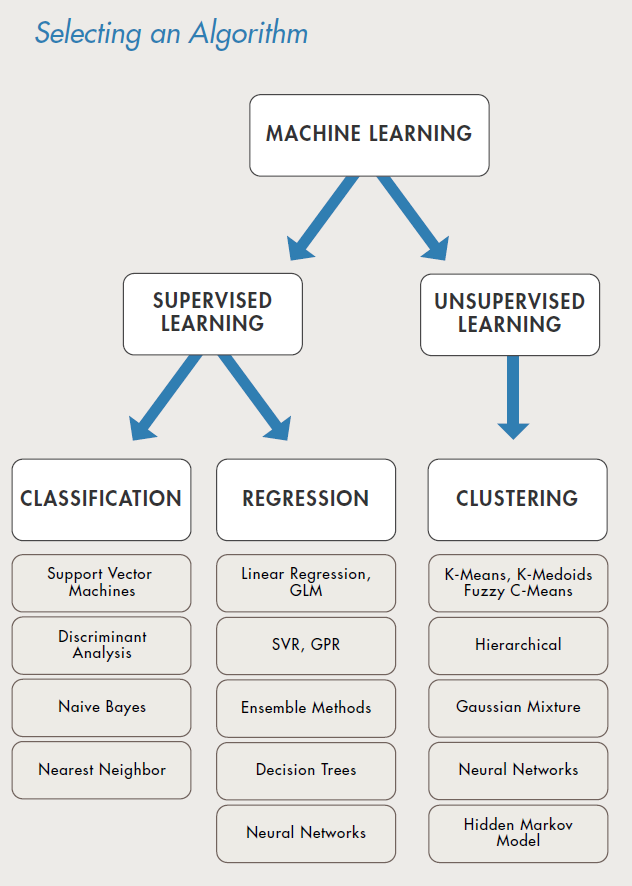
**Clustering** is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data.

Applications for clustering include gene sequence analysis, market research, and object recognition.

**How do you decide which algorithm to use?**

Choosing the right algorithm can seem overwhelming-there are dozens of supervised and unsupervised machine learning algorithms, and each takes a different approach to learning.

There is no best method or one size fits all. Finding the right algorithm is partly just trial and error – even highly experienced data scientists can’t tell whether an algorithm will work without trying it out. But algorithm selection also demands on the size and type of data you’re working with, the insights you want to get from the data, and how those insights will be used.

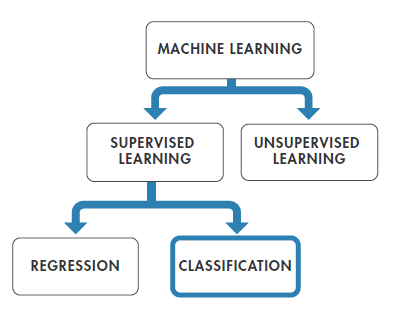


Every machine learning workflow begins with three questions:

* What kind of data are you working with?
* What insights do you want to get from it?
* How and where will those insights be applied?

Chose supervised learning if you need to train a model to make a prediction – for example, the future value of a continuous variable, such as temperature or a stock price, or a classification - for example, identify makes of cars from webcam video footage.

Choose unsupervised learning if you need to explore your data and want to train a model to find a good internal representation, such as splitting data up into clusters.



**Common Classification Algorithms**

**Logistic Regression**

How it works:

Fits a model that can predict the probability of binary response belonging to one class or the other.

Because of its simplicity, logistic regression is commonly used as a starting point for binary classification problems.

Best Used:

* When data can be clearly separated by a single linear boundary
* As a baseline for evaluating more complex classification methods

**K Nearest Neighbor (kNN)**

How it works:

kNN categorizes objects based on the classes of their nearest neighbors in the dataset. kNN predictions assume that objects near each other are similar. Distance metrics, such as Euclidean, city block, cosine, and Chebychev, are used to find the nearest neighbor.

Best used:

* When you need a simple algorithm to establish benchmark learning rules
* When memory usage of the trained model is a lesser concern
* When prediction speed of the trained model is a lesser concern

**Neural Network**

How it works:

Inspired by the human brain, a neural network consists of highly connected networks of neurons that relate the inputs to the desired outputs. The network is trained by iteratively modifying the strengths of the connections so that given inputs map to the correct response.

Best used:

* For modeling highly, nonlinear systems
* When data is available incrementally and you wish to constantly update the model
* When there could be unexpected changes in your input data
* When model interpretability is not a key concern

**Naïve Bayes**

How it works:

A naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It classifies new data based on the highest probability of its belonging to a particular class.

Best used:

* For a small dataset containing many parameters
* When you need a classifier that’s easy to interpret
* When the model will encounter scenarios that weren’t in the training data, as is the case with many financial and medical applications.

**Decision Tree:**

How it works:

A decision tree lets you predict responses to data by following the decisions in the tree from the root down to a leaf node. A tree consists of branching condition where the value of a predictor is compared to a trained weight. The number of branches and the values of weights are determined in the training process. Additional modification, or pruning, may be used to simplify the model.

Best used:

* When you need an algorithm that is easy to interpret and fast to fit
* To minimize memory usage
* When high predictive accuracy is not a requirement

**Linear Regression**

How it works:

Linear regression is a statistical modeling technique used to describe a continuous response variable as a linear function of one or more predictor variables. Because linear regression models are simple to interpret and easy to train, they are often the first model to be fitted to a new dataset.

Best used:

* When you need an algorithm that is easy to interpret and fast to fit
* As a baseline for evaluating other, more complex, regression models

**Nonlinear Regression**

How it works:

Nonlinear regression is a statistical modeling technique that helps describe nonlinear relationships in experimental data. Nonlinear regression models are generally assumed to be parametric, where the model is described as a nonlinear equation.

Nonlinear refers to a fit function that is a nonlinear function of the parameters. For example, if the fitting parameters are b0, b1, b2: the equation y = b0 + b1 x + b2 x2 is a linear function of the fitting parameters, whereas y = (b0xb1)/(x+b2) is a nonlinear function of the fitting parameters.

Best used:

* When data has strong nonlinear trends and cannot easily transformed into a linear space
* For fitting custom models to data­

**What is a Time Series?**

* A sequence of values or events where the next event is determined by the events that precede it
* The next step in a time series may be determined by 1 or more of the previous steps. The number of steps is known as the order of the time series.

**4 different types of behavior:**

**Level**

* Level is simply the average value of the time series
* If the average level is the same throughout its length then the series is said to be ‘stationary’
* A stationary system might get pushed off its level by a sudden shock, but it will return to this level quite quickly

**Trend**

* A process that produces values that get continually larger over time is said to have a trend
* The average level for such data is of no use as the data will never be that value again
* Trend can be a function of time or precious values

**Seasonality**

* In time series analysis, a season is any period of time that repeats through the data e.g.
  + Monday, Tuesday, Wednesday
  + March, April, May
  + 1pm, 2pm, 3pm
* Seasonality is always of a fixed and known period.
* Each season will have an impact of the data produced during that season
  + Sales may be much higher during December
  + Temperatures are higher in summer

**Cycles**

* Cycles often add together to produce complex wave forms
  + Sound, images, other vibrations, etc.

**Cycles vs Seasonality**

* Cyclic pattern are not of fixed period
* Seasonal pattern is unchanging and associated with some aspect of the calendar
* The average length of cycles is no longer than the length of a seasonal pattern

**What is Bayesian Network?**

A Bayesian Network (BN) is a way of describing the relationships between causes and effects, and is made up of nodes and arcs.

Mainly is used to support decision making and to find strategies to solve tasks under uncertainty.

**Uncertainty:** The lack of certainty. A state of having limited knowledge where it is impossible to exactly describe the existing state, a future outcome, or more than one possible outcome.

**Measurement of uncertainty:** A set of possible states or outcomes where probabilities are assigned to each possible state or outcome.

**Risk:** A state of uncertainty where some possible outcomes have an undesired effect or significant loss.

**Measurement of Risk:** A set of measured uncertainties where some possible outcomes are losses, and the magnitudes of those losses.

**How to build a Bayesian network?**

Step 1: Collect information

* List information we are given
* Determine information we can deduct from the information we are given

Step 2: Convert information into a BN

* Determine the nodes
* Determine Relationships between Nodes
* Convert information into a Network

**Multi-layer Perceptron:**

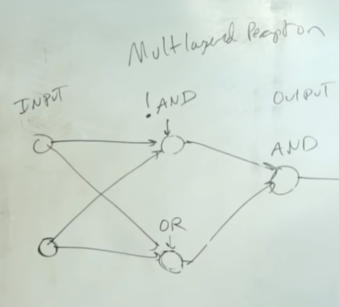
An x and y signals can connect to a single perceptron/neuron with weight and the single perceptron can create a weighted sum of all the inputs multiplied by their weights pass through the activation function to generate the output. The reason why a single perceptron can’t have multiple outputs is because a single perceptron can only solve linearly separable problems. What that means? Let’s take an example with Boolean expressions. AND (&&), OR (||). The idea behind that is to create a truth table. Let’s say A&&B, need both to be true. This is a linearly separable problem.

|  |  |  |
| --- | --- | --- |
|  | **T** | **F** |
| **T** | **T** | **F** |
| **F** | **F** | **F** |

True is on one side and false is on other side. A single perceptron can handle that. But there is another Boolean expression. The XOR which the X stands for exclusive! That means it is ONLY true if there is one False and one True. NOT 2 false NOT 2 true.

|  |  |  |
| --- | --- | --- |
|  | **T** | **F** |
| **T** | **F** | **T** |
| **F** | **T** | **F** |

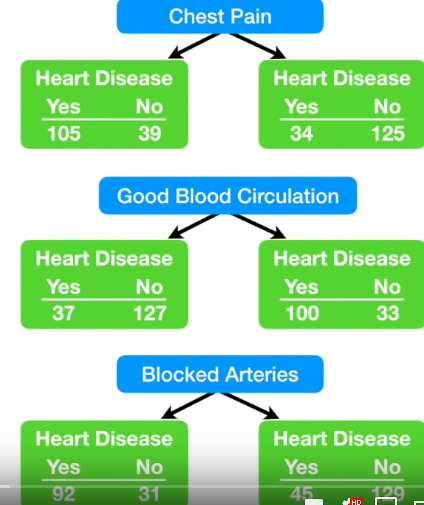
This mean a single perceptron CAN’T solve this. The answer to this is to use multi-layer perceptron.



**Decision Trees:**

A decision tree asks a question and then classifies that based on the answer. Decision trees are intuitive to work with. You start from the top and work your way down till you get to a point where you can’t go any further, and that’s how you will classify a sample. First, we need to decide which of the attributes should go first to be the root node.

Here we see that none of the leaf nodes are 100% YES heart Disease or 100% NO heart Disease. So, they are all “impure”.



To determine which separation is the best, we need a way to measure and compare “impurity” using Information gain & Entropy.

The entropy of target: We have 8 records with negative class and 8 records with positive class. So, we can directly estimate the entropy of target as 1.

E(8,8) = -1\*((P(+ve)\*log(p(+ve)) + (p(-ve)\*log(p(-ve)) )

= -1\*((8/16) \*log­2­(8/16)) + (8/16)\*log­2(8/16))

= 1

For Var A (>=5) == positive: 5/12

For Var A (>=5) == negative: 7/12

Entropy (5,7) = -1\*(5/12) \*log2(5/12) + (7/12) \*log2(7/12) = 0.9799

For Var A (<5) == positive ¾

For Var A (<5 == negative ¼

Entropy (3,1) = -1\*(3/4) \*log2(3/4) + (1/4) \*log2(1/4) = 0.81128

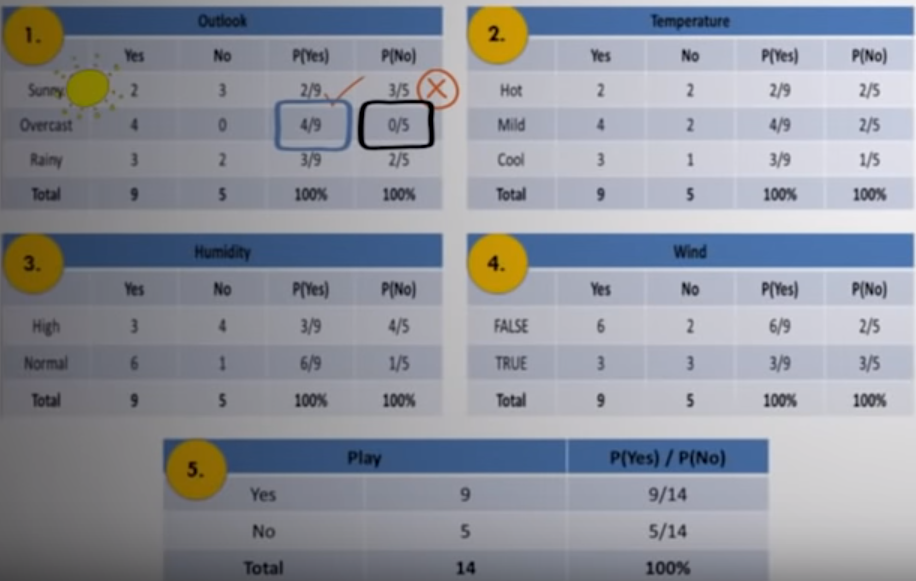
Entropy (Target A) = (12/16) \*0.9799 + (4/16) \*0.81128 = 0.937745

Information Gain = 1 – 0.937745 = 0.063

**Naïve Bayes Classifier**

Dataset P(YES) = 9/14 P(NO) = 5/14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play** |
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |



**Probability that we can play golf.**

> P (Outlook=Sunny | Play = Yes) = 2/9

> P (Temperature=Cool | Play = Yes) = 3/9

> P (Humidity=Hight |Play = Yes) = 3/9

> P (Wind=Strong | Play = Yes) = 3/9

> P (Play=Yes) = 9/14

**Probability that we cannot play golf.**

> P (Outlook=Sunny | Play = No) = 3/5

> P (Temperature=Cool | Play = No) = 1/5

> P (Humidity=Hight |Play = No) = 4/5

> P (Wind=Strong | Play = No) = 3/5

> P (Play=No) = 5/14

P(X|Play=Yes)P(Play=Yes) = (2/9)\*(3/9)\* (3/9)\* (3/9)\*(9/14) = 0.0053

P(X|Play=No)P(Play=No) = (3/5)\*(1/5)\*(4/5)\*(3/5)\*(5/14) = 0.0206

>P(X) = P(Outlook=Sunny) \*P(Temperature=Cool) \*P(Humidity=High) \*P(Wind=Strong)

>P(X) = (5/14) \* (4/14) \* (7/14) \* (6/14)

>P(X) = **0.02186**

>P(Play=Yes|X) = 0.0053/**0.002186** = 0.2424

>P(Play=No|X) = 0.0206/**0.002186** = 0.9421 – Highest Value. So there is no golf today.