Bayes Theorem

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- $P(h)$ = prior probability of hypothesis h
- $P(D)$ = prior probability of training data D
- $P(h/D)$ = probability of h given D
- $P(D/h)$ = probability of D given h

Choosing Hypotheses

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

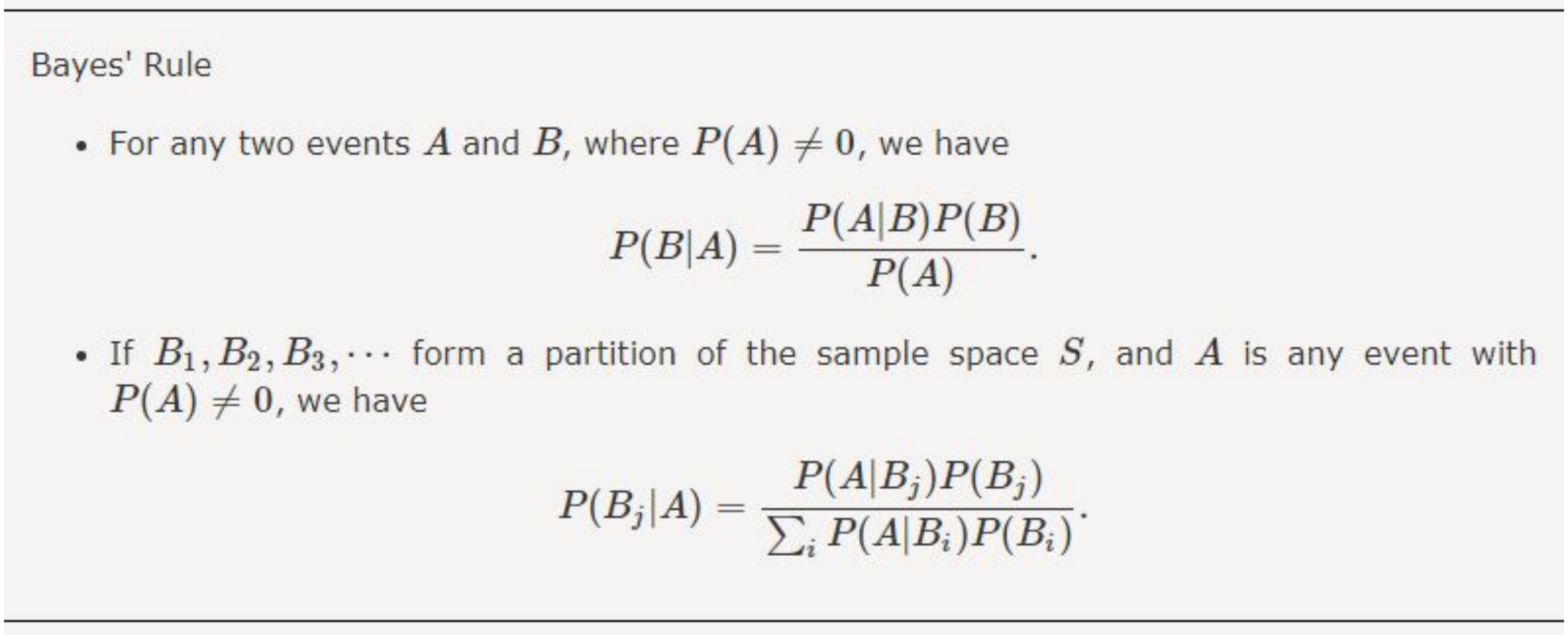
Generally want the most probable hypothesis given the training data

Maximum a posteriori hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h \mid D) \\ &= \arg\max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D \mid h)P(h) \end{aligned}$$

If we assume $P(h_i)=P(h_j)$ then can further simplify, and choose the *Maximum likelihood*(ML) hypothesis

$$h_{ML} = \arg\max_{h_i \in H} P(D \mid h_i)$$



Bayes' Rule

- For any two events A and B, where $P(A) \neq 0$, we have

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}.$$

- If $B_1, B_2, B_3, \cdots$ form a partition of the sample space S, and A is any event with $P(A) \neq 0$, we have

$$P(B_j|A) = \frac{P(A|B_j)P(B_j)}{\sum_i P(A|B_i)P(B_i)}.$$

Bayes Theorem

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, 0.8% of the entire population have this cancer.

$P(\text{cancer}) =$ $P(\neg\text{cancer}) =$

$P(+|\text{cancer}) =$ $P(-|\text{cancer}) =$

$P(+|\neg\text{cancer}) =$ $P(-|\neg\text{cancer}) =$

$P(\text{cancer}|+) =$

$P(\neg\text{cancer}|+) =$

- The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present.
- Furthermore, .008 of the entire population have cancer.

$P(\text{cancer}) = .008$	$P(\text{notcancer}) = .992$
$P(+\|\text{cancer}) = .98$	$P(-\|\text{cancer}) = .02$
$P(+\|\text{notcancer}) = .03$	$P(-\|\text{notcancer}) = .97$

- A patient takes a lab test and the result comes back positive.

$P(+|\text{cancer})\ P(\text{cancer}) = .98 * .008 = .0078$

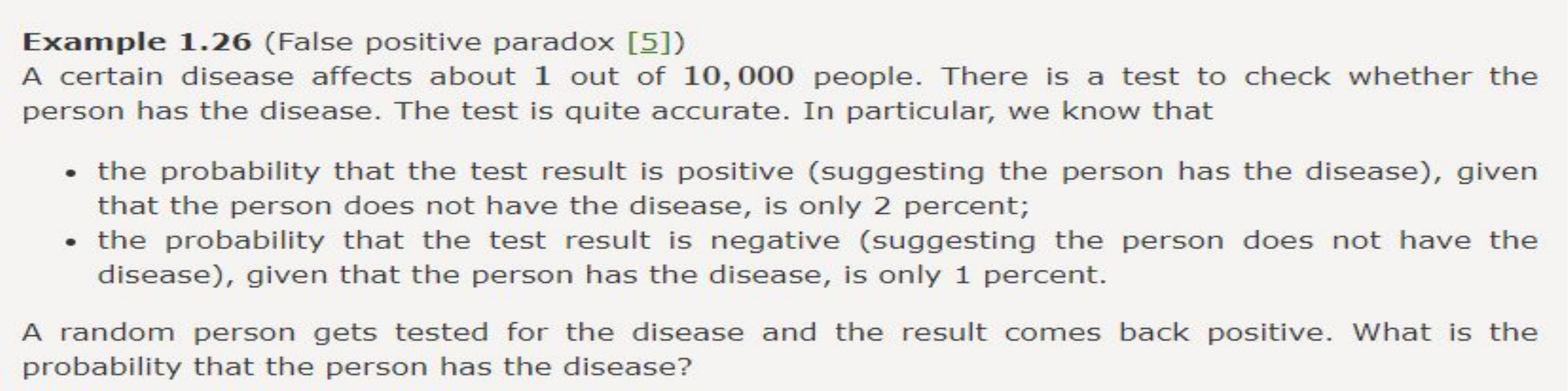
$P(+|\text{notcancer})\ P(\text{notcancer}) = .03 * .992 = .0298$ ➔ ***h_{MAP} is notcancer***

- Since $P(\text{cancer}|+) + P(\text{notcancer}|+)$ must be 1

$P(\text{cancer}|+) = .0078 / (.0078+.0298) = .21$

$P(\text{notcancer}|+) = .0298 / (.0078+.0298) = .79$

- 1% of a population have a certain disease and the remaining 99% are free from this disease. A test is used to detect this disease. This test is positive in 95% of the people with the disease and is also (falsely) positive in 2% of the people free from the disease.
  If a person, selected at random from this population, has tested positive, what is the probability that she/he has the disease?

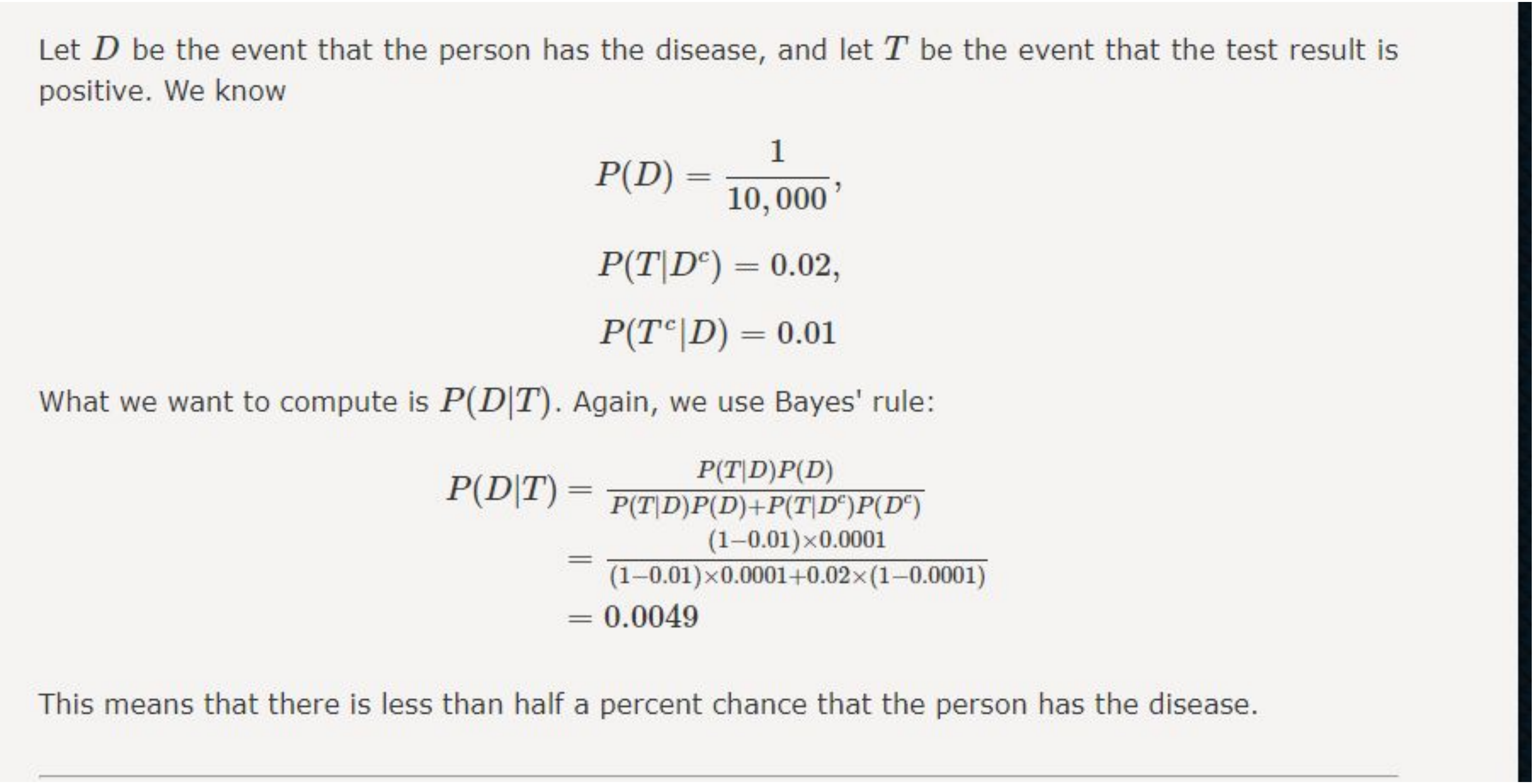


Example 1.26 (False positive paradox [5])

A certain disease affects about 1 out of $10,000$ people. There is a test to check whether the person has the disease. The test is quite accurate. In particular, we know that

- the probability that the test result is positive (suggesting the person has the disease), given that the person does not have the disease, is only 2 percent;
- the probability that the test result is negative (suggesting the person does not have the disease), given that the person has the disease, is only 1 percent.

A random person gets tested for the disease and the result comes back positive. What is the probability that the person has the disease?



Let D be the event that the person has the disease, and let T be the event that the test result is positive. We know

$$P(D) = \frac{1}{10,000},$$

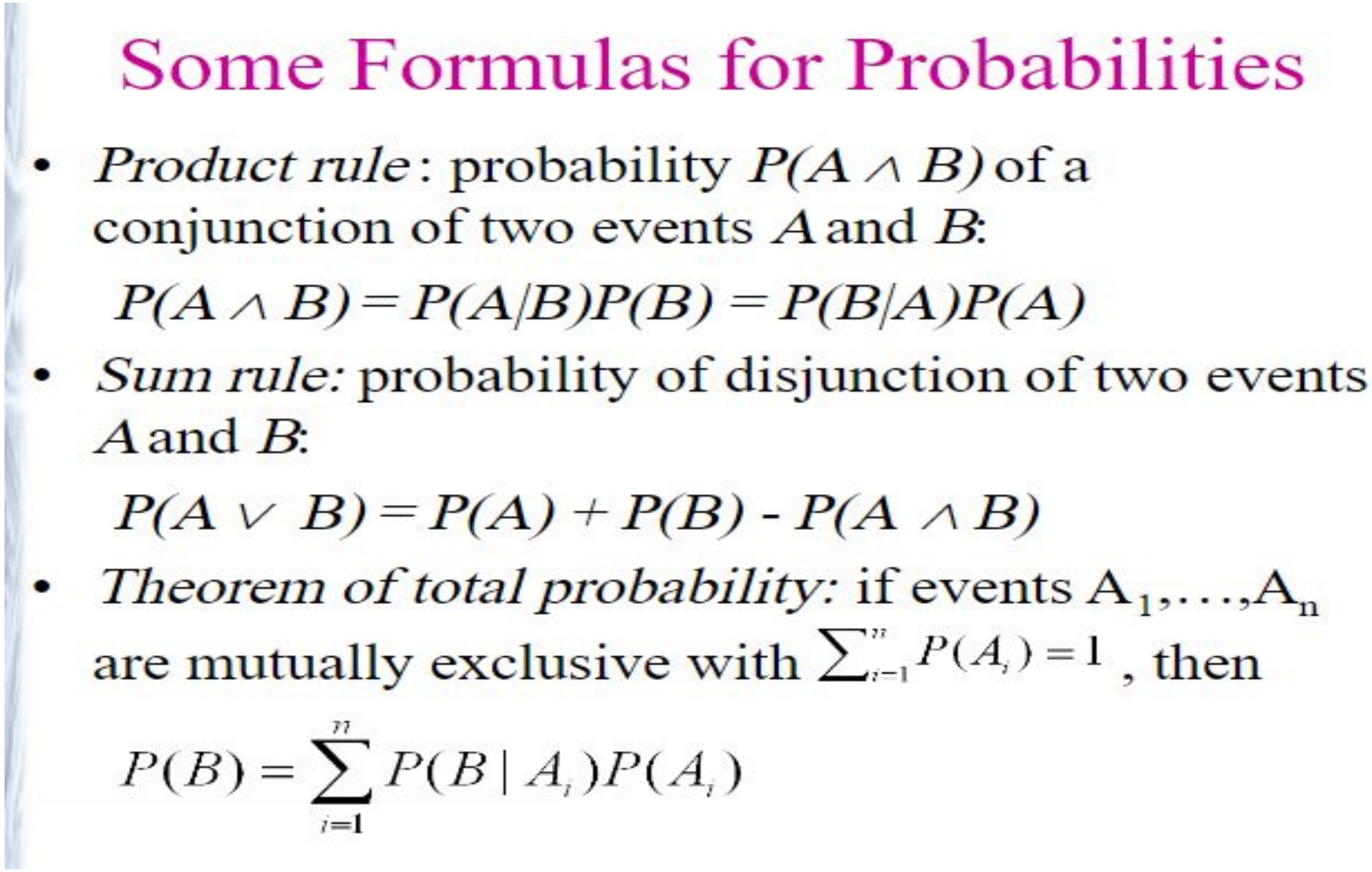
$$P(T|D^c) = 0.02,$$

$$P(T^c|D) = 0.01$$

What we want to compute is $P(D|T)$. Again, we use Bayes' rule:

$$\begin{aligned} P(D|T) &= \frac{P(T|D)P(D)}{P(T|D)P(D)+P(T|D^c)P(D^c)} \\ &= \frac{(1-0.01)\times 0.0001}{(1-0.01)\times 0.0001+0.02\times(1-0.0001)} \\ &= 0.0049 \end{aligned}$$

This means that there is less than half a percent chance that the person has the disease.



Some Formulas for Probabilities

- *Product rule*: probability $P(A \wedge B)$ of a conjunction of two events A and B:

  $$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

- *Sum rule:* probability of disjunction of two events A and B:

  $$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- *Theorem of total probability:* if events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then

  $$P(B) = \sum_{i=1}^{n} P(B \mid A_i) P(A_i)$$

Brute Force MAP Hypothesis Learner

1. For each hypothesis *h* in *H*, calculate the posterior probability

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \arg\max_{h \in H} P(h \mid D)$$

Naïve Bayes' Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naïve Bayes?

- The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem

Working of Naïve Bayes' Classifier:

- Working of Naïve Bayes' Classifier can be understood with the help of the below example:
- Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:
- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.

$$X = (x_1, x_2, x_3, ..., x_n)$$

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remain static. Therefore, the denominator can be removed and a proportionality can be introduced.

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

In our case, the class variable(**y**) has only two outcomes, yes or no. There could be cases where the classification could be multivariate. Therefore, we need to find the class **y** with maximum probability.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

- Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Applying Bayes'theorem:

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)

P(Sunny|Yes)= 3/10= 0.3
P(Sunny)= 0.35
P(Yes)=0.71

So P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**
P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)
P(Sunny|NO)= 2/4=0.5

P(No)= 0.29
P(Sunny)= 0.35
So P(No|Sunny)= 0.5*0.29/0.35 = **0.41**
So as we can see from the above calculation that **P(Yes|Sunny)>P(No|Sunny)**
Hence on a Sunny day, Player can play the game.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

Types of Naïve Bayes Model:

There are three types of Naive Bayes Model, which are given below:

- **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc.
The classifier uses the frequency of words for the predictors.

- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

Bayesian Belief Network or Bayesian Network or Belief Network

"A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."

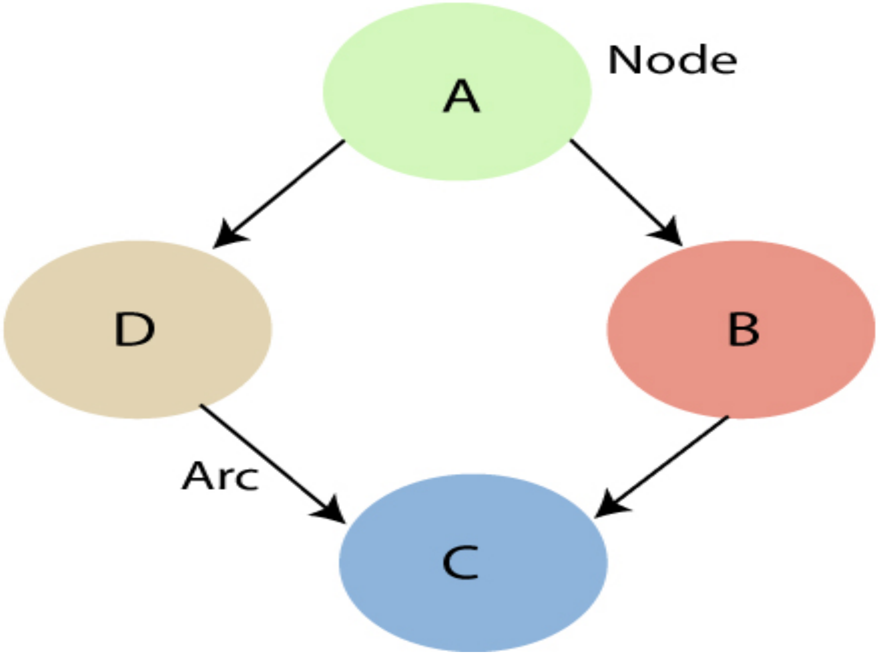
- It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.
- Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.
- Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision making under uncertainty**.

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

Directed Acyclic Graph

Table of conditional probabilities.

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**



- Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph. These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
  - **In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.**
  - **If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.**
  - **Node C is independent of node A.**

Note: The Bayesian network graph does not contain any cyclic graph. Hence, it is known as a directed acyclic graph or DAG.

The Bayesian network has mainly two components:

- **Causal Component**
- **Actual numbers**

Each node in the Bayesian network has condition probability distribution **P(Xi |Parent(Xi))**, which determines the effect of the parent on that node

Joint probability distribution:

If we have variables x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.

P[x1, x2, x3,....., xn], it can be written as the following way in terms of the joint probability distribution.

= P[x1| x2, x3,....., xn]P[x2, x3,....., xn]

= P[x1| x2, x3,....., xn]P[x2|x3,....., xn]....P[xn-1|xn]P[xn].

Explanation of Bayesian network:

Let's understand the Bayesian network through an example by creating a directed acyclic graph:

Example: Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbours David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm. Here we would like to compute the probability of Burglary Alarm.

Problem:

Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry.

Solution:

- The Bayesian network for the above problem is given below. The network structure is showing that burglary and earthquake is the parent node of the alarm and directly affecting the probability of alarm's going off, but David and Sophia's calls depend on alarm probability.
- The network is representing that our assumptions do not directly perceive the burglary and also do not notice the minor earthquake, and they also not confer before calling.
- The conditional distributions for each node are given as conditional probabilities table or CPT.
- Each row in the CPT must be sum to 1 because all the entries in the table represent an exhaustive set of cases for the variable.
- In CPT, a boolean variable with k boolean parents contains 2K probabilities. Hence, if there are two parents, then CPT will contain 4 probability values

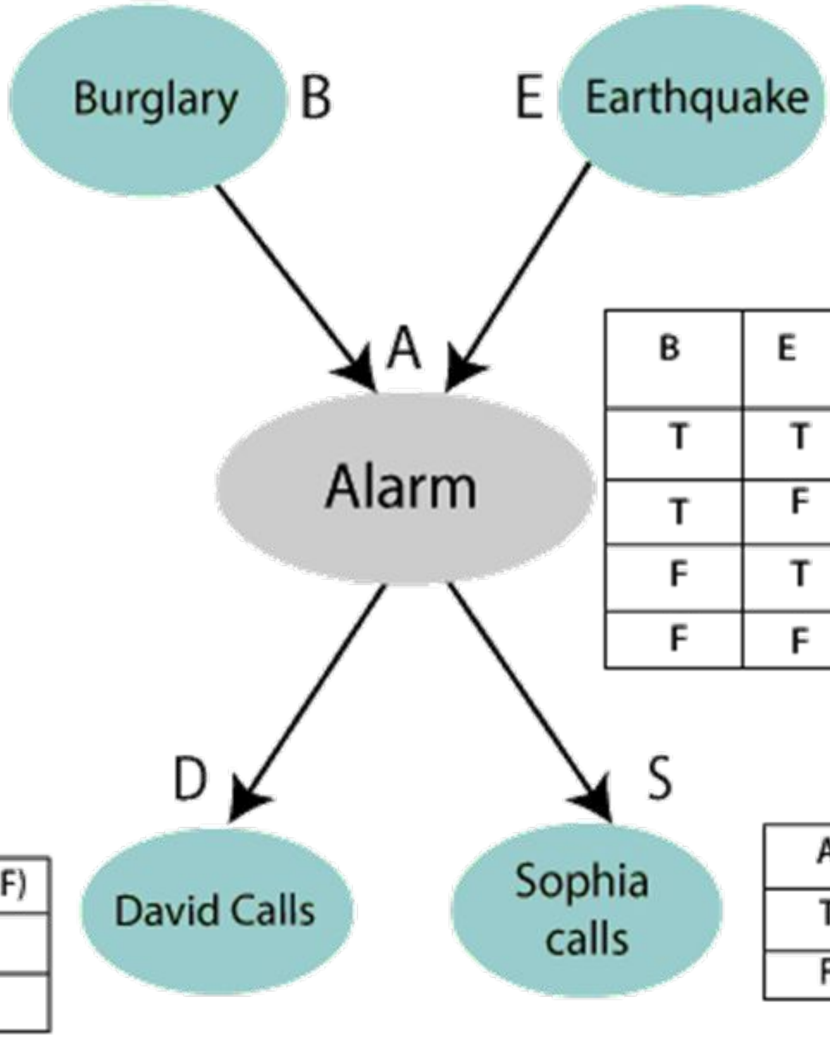
List of all events occurring in this network:

- **Burglary (B)**
- **Earthquake(E)**
- **Alarm(A)**
- **David Calls(D)**
- **Sophia calls(S)**

We can write the events of problem statement in the form of probability: **P[D, S, A, B, E]**, can rewrite the above probability statement using joint probability distribution:

P[D, S, A, B, E]= P[D | S, A, B, E]. P[S, A, B, E]
=P[D | S, A, B, E]. P[S | A, B, E]. P[A, B, E]
= P [D| A]. P [S| A, B, E]. P[A, B, E]
= P[D | A]. P[S | A]. P[A| B, E]. P[B, E]
= P[D | A]. P[S | A]. P[A| B, E]. P[B |E]. P[E]

T	0.002
F	0.998



T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

Let's take the observed probability for the Burglary and earthquake component:

P(B= True) = 0.002, which is the probability of burglary.

P(B= False)= 0.998, which is the probability of no burglary.

P(E= True)= 0.001, which is the probability of a minor earthquake

P(E= False)= 0.999, Which is the probability that an earthquake not occurred.

We can provide the conditional probabilities as per the below tables:

Conditional probability table for Alarm A:

The Conditional probability of Alarm A depends on Burglar and earthquake:

Conditional probability table for David Calls:

B	E	P(A= True)	P(A= False)
True	True	0.94	0.06
True	False	0.95	0.04
False	True	0.31	0.69
False	False	0.001	0.999

A	P(D= True)	P(D= False)
True	0.91	0.09

The Conditional probability of David that he will call depends on the probability of Alarm.

A	P(S= True)	P(S= False)
True	0.75	0.25
False	0.02	0.98

P(S, D, A, ¬B, ¬E) = P (S|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E).

= 0.75* 0.91* 0.001* 0.998*0.999

= 0.00068045.

Hence, a Bayesian network can answer any query about the domain by using Joint distribution.

The semantics of Bayesian Network:

There are two ways to understand the semantics of the Bayesian network, which is given below:

1. To understand the network as the representation of the Joint probability distribution.

It is helpful to understand how to construct the network.

2. To understand the network as an encoding of a collection of conditional independence statements.

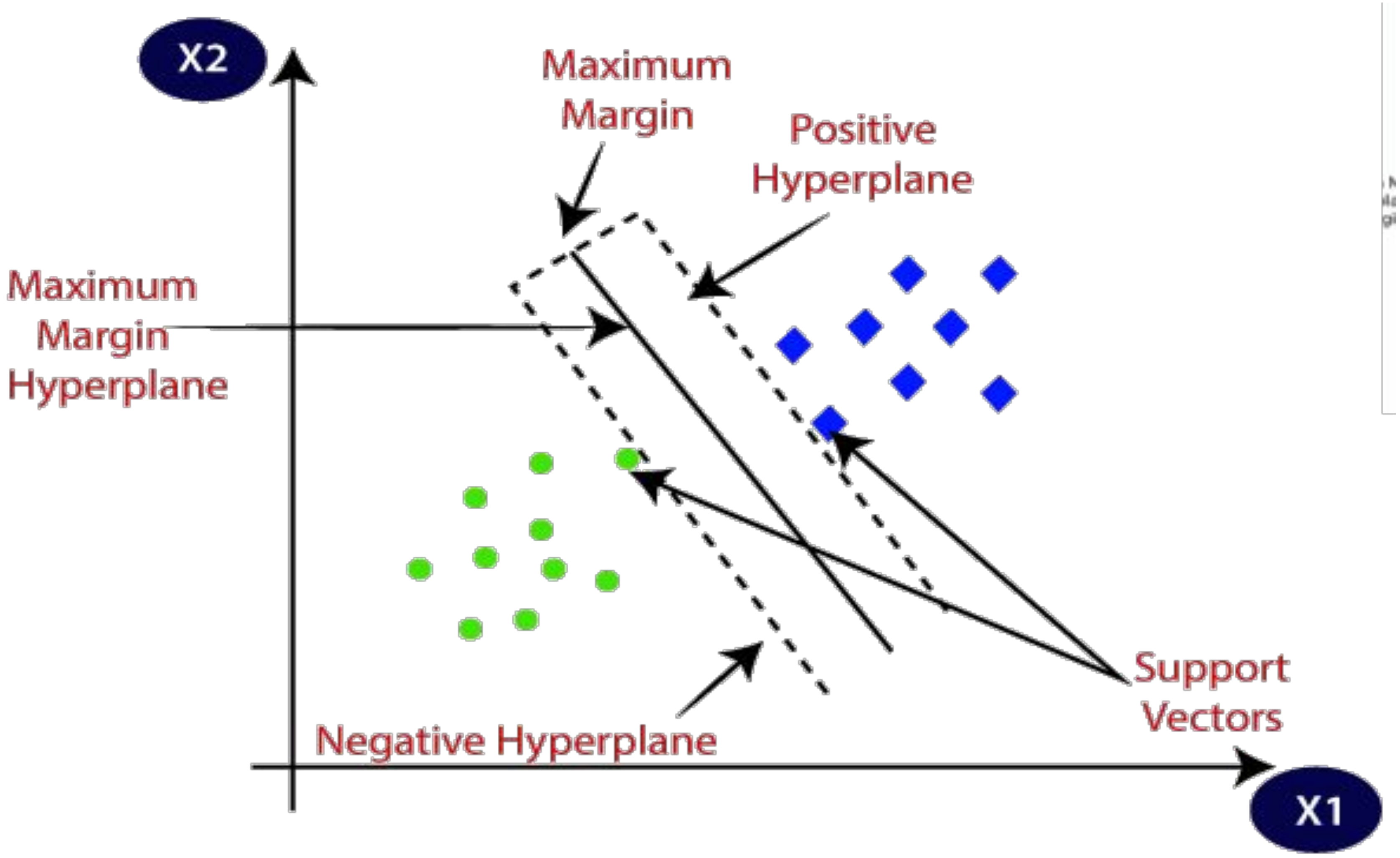
It is helpful in designing inference procedure.

Support Vector Machine Algorithm

- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:


X2
Maximum Margin
Positive Hyperplane
Maximum Margin Hyperplane
Support Vectors
Negative Hyperplane
X1

SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.
The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

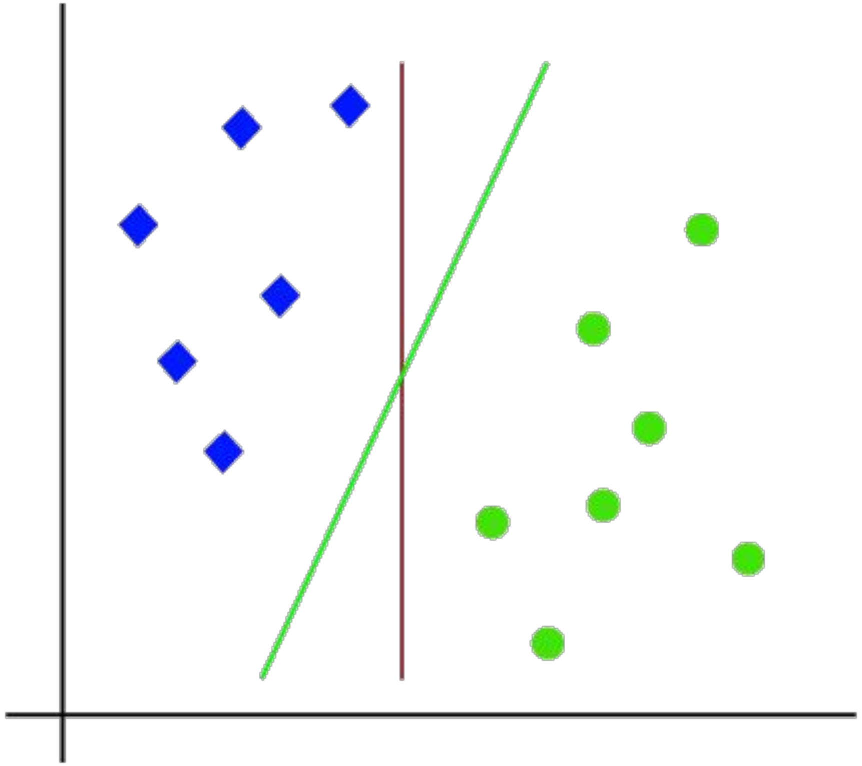
Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

How does SVM works?

Linear SVM:

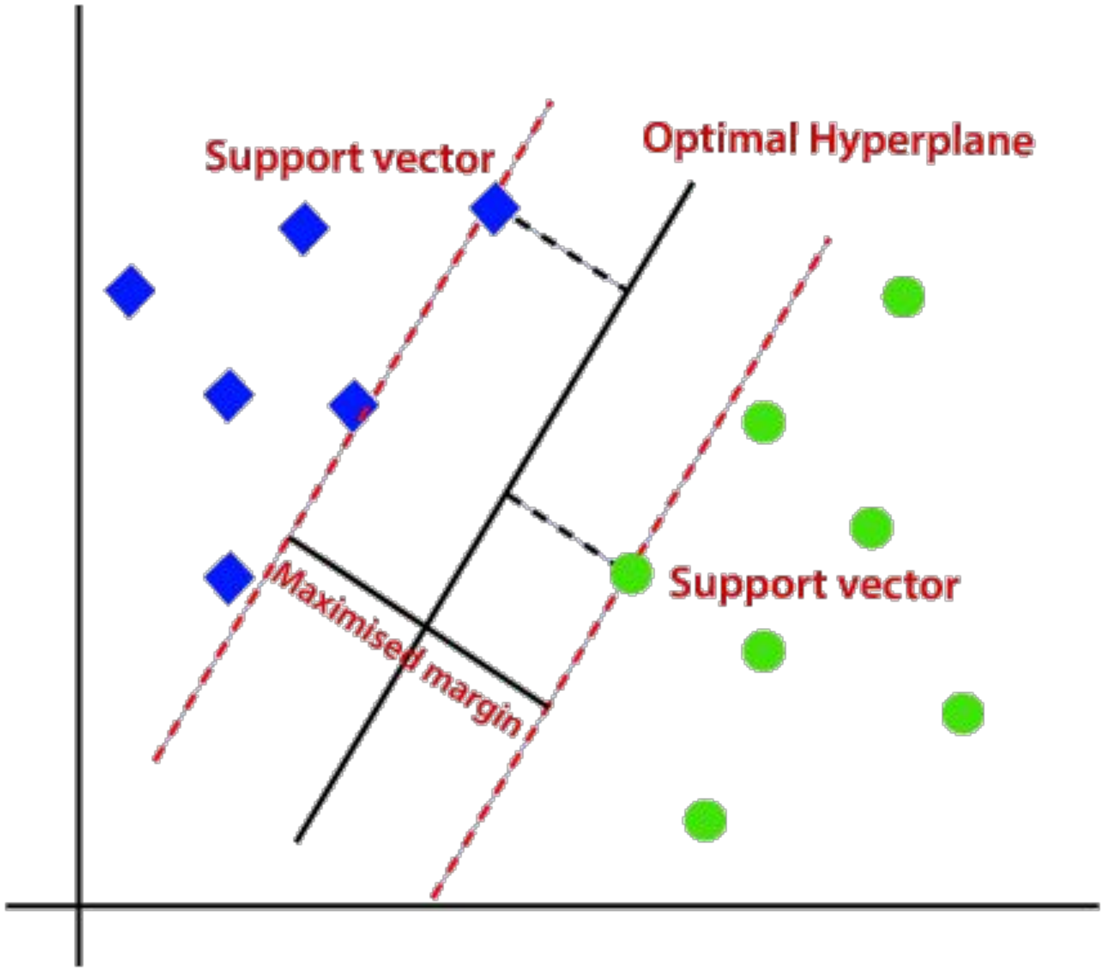
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



- Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyper-plane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors.

- The distance between the vectors and the hyper-plane is called as **margin**. And the goal of SVM is to maximize this margin.

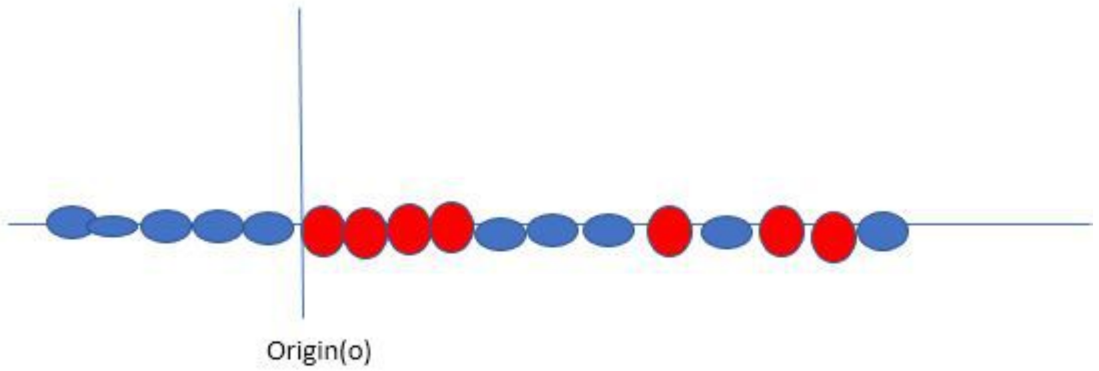
- The **hyper-plane** with maximum margin is called the **optimal hyper-plane**.



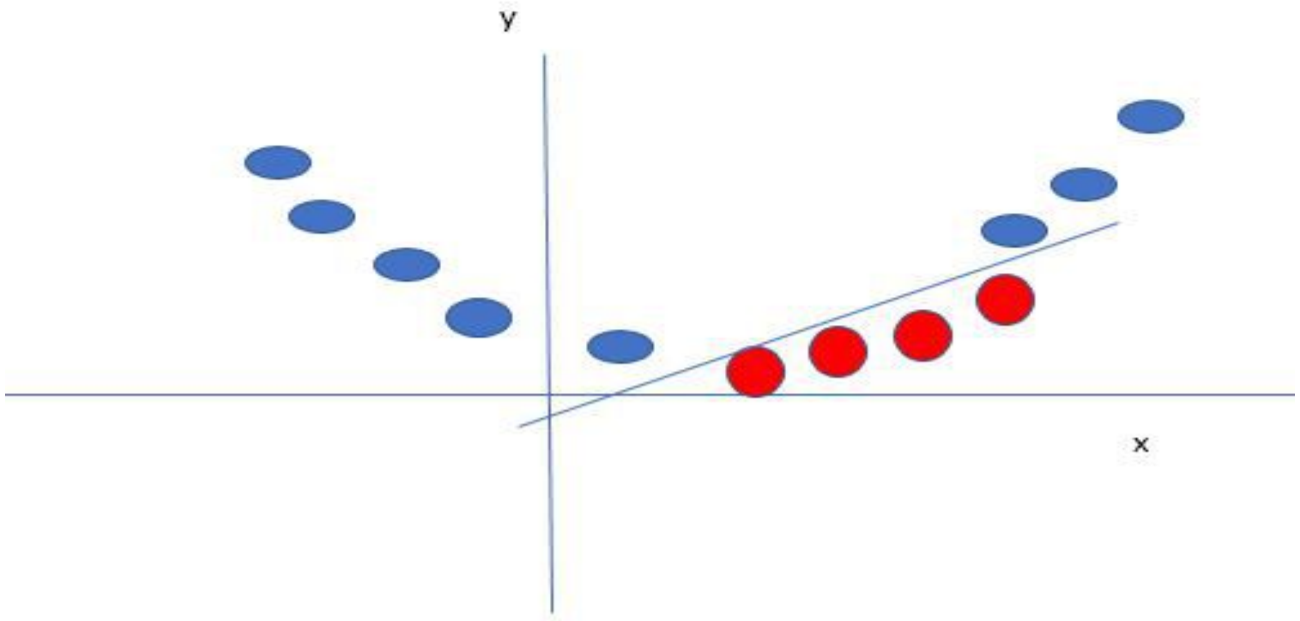
Non-Linear SVM:

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Say, our data is like shown in the figure above.SVM solves this by creating a new variable using a kernel. We call a point x_i on the line and we create a new variable y_i as a function of distance from origin o.so if we plot this we get something like as shown below



In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as kernel.

SVM Kernel:

The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts non separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put the kernel, it does some extremely complex data transformations then finds out the process to separate the data based on the labels or outputs defined.

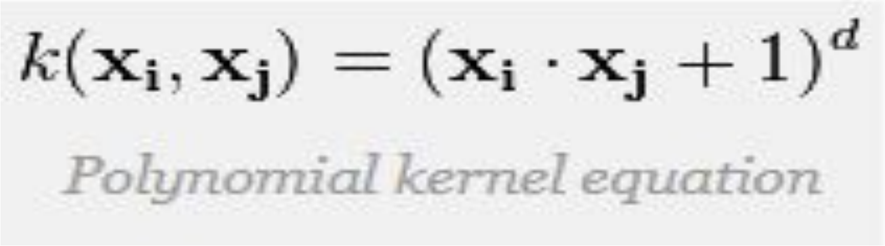
SVM Applications

SVMs are utilized in applications like

- **Handwriting recognition,**
- **Intrusion detection,**
- **Face detection,**
- **Email classification,**
- **Gene classification.**
- **for pattern classification problems.**
- **classification of natural text documents into a fixed number of predefined categories based on their content**

SVM help us to find complex relationships among the provided dataset without you involving in plenty of transformations.

- **SVM Kernel Functions**
- SVM algorithms use a group of mathematical functions that are known as kernels. The function of a kernel is to require data as input and transform it into the desired form.
- Different SVM algorithms use differing kinds of kernel functions. These functions are of different kinds—for instance, linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.
- The most preferred kind of kernel function is RBF. Because it's localized and has a finite response along the complete x-axis.
- The kernel functions return the scalar product between two points in an exceedingly suitable feature space. Thus, by defining a notion of resemblance, with a little computing cost even in the case of very high-dimensional spaces.
- Some of the Kernel functions are;
- **i)Polynomial:** A polynomial mapping is a popular method for non-linear modeling. It is popular in image processing.

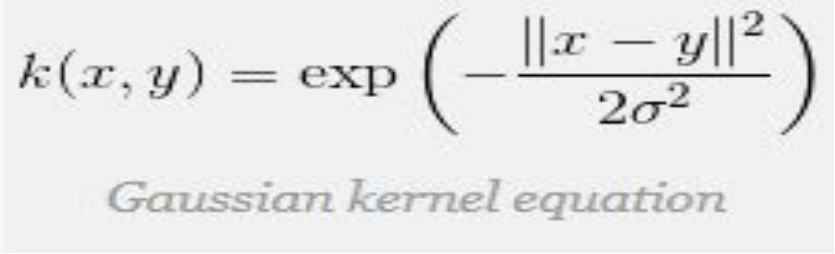


$$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$$

Polynomial kernel equation

ii) Gaussian kernel

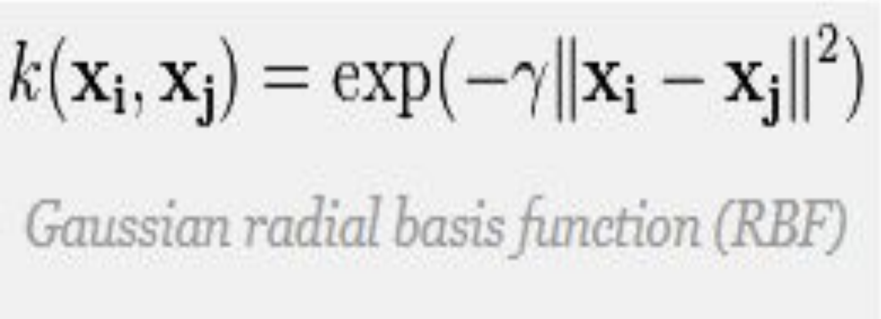
It is most widely used. It is a general-purpose kernel; used when there is no prior knowledge about the data. Equation is:



$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

Gaussian kernel equation

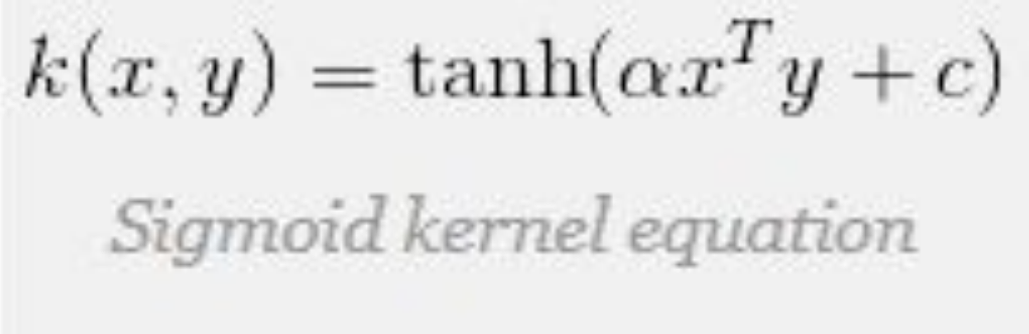
- **iii)Gaussian radial basis function (RBF)**
- It is a general-purpose kernel; used when there is no prior knowledge about the data.
  Equation is:



$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma\|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

Gaussian radial basis function (RBF)

- **iv) Sigmoid kernel**
- We can use it as the proxy for neural networks. Equation is



$$k(x, y) = \tanh(\alpha x^T y + c)$$

Sigmoid kernel equation

SVM Main Properties

- Known for their robustness, good generalization ability, and unique global optimum solutions, SVMs are probably the most popular machine learning approach for supervised learning, yet their principle is very simple. What makes SVM an attractive machine learning framework can be summarized by the following properties:
- •**SVM is a sparse technique.** Like nonparametric methods, SVM requires that all the training data be available, that is, stored in memory during the training phase, when the parameters of the SVM model are learned. However, once the model parameters are identified, SVM depends only on a subset of these training instances, called support vectors, for future prediction
- **SVM is a kernel technique.** SVM uses the kernel trick to map the data into a higher-dimensional space before solving the machine learning task .
- **SVM is a maximum margin separator.** Beyond minimizing the error or a cost function, based on the training datasets (similar to other discriminant machine learning techniques), SVM imposes an additional constraint on the optimization problem: the hyperplane needs to be situated such that it is at a maximum distance from the different classes

Advantages of Support Vector Machine (SVM)

1. Regularization capabilities: SVM has L2 Regularization feature. So, it has good generalization capabilities which prevent it from over-fitting.
2. Handles non-linear data efficiently: SVM can efficiently handle non-linear data using Kernel trick.

3. Solves both Classification and Regression problems: SVM can be used to solve both classification and regression problems. SVM is used for classification problems while **SVR (Support Vector Regression)** is used for regression problems.

4. Stability: A small change to the data does not greatly affect the hyperplane and hence the SVM. So the SVM model is stable.
5. SVM is more effective in high dimensional spaces.
6. SVM is effective in cases where the number of dimensions is greater than the number of samples.

Disadvantages of Support Vector Machine (SVM)

1. Choosing an appropriate Kernel function is difficult: Choosing an appropriate Kernel function (to handle the non-linear data) is not an easy task. It could be tricky and complex. In case of using a high dimension Kernel, you might generate too many support vectors which reduce the training speed drastically.

2. Extensive memory requirement: Algorithmic complexity and memory requirements of SVM are very high. You need a lot of memory since you have to store all the support vectors in the memory and this number grows abruptly with the training dataset size.

3. Requires Feature Scaling: One must do feature scaling of variables before applying SVM.

4. Long training time: SVM takes a long training time on large datasets.

5. Difficult to interpret: SVM model is difficult to understand and interpret by human beings unlike Decision Trees.

6. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.