1. Provide an example of the concepts of Prior, Posterior, and Likelihood.

**Understanding Prior Probability**

The prior probability of an event will be revised as new data or information becomes available, to produce a more accurate measure of a potential outcome. That revised probability becomes the posterior probability and is calculated using [Bayes' theorem](https://www.investopedia.com/terms/b/bayes-theorem.asp). In statistical terms, the posterior probability is the probability of event A occurring given that event B has occurred.

Example

For example, three acres of land have the labels A, B, and C. One acre has reserves of oil below its surface, while the other two do not. The prior probability of oil being found on acre C is one third, or 0.333. But if a drilling test is conducted on acre B, and the results indicate that no oil is present at the location, then the posterior probability of oil being found on acres A and C become 0.5, as each acre has one out of two chances.

1. What role does Bayes' theorem play in the concept learning principle?

# Bayes Theorem in Machine learning

Bayes’ theorem is fundamental in machine learning, especially in the context of Bayesian inference. It provides a way to update our beliefs about a hypothesis based on new evidence.

## What is Bayes theorem?

Bayes’ theorem is a fundamental concept in [probability theory](https://www.geeksforgeeks.org/probability-theory/)that plays a crucial role in various machine learning algorithms, especially in the fields of Bayesian statistics and probabilistic modelling. It provides a way to update probabilities based on new evidence or information. In the context of machine learning, Bayes’ theorem is often used in Bayesian inference and probabilistic models.

*The theorem can be mathematically expressed as:*

*𝑃(𝐴∣𝐵)=𝑃(𝐵∣𝐴)⋅𝑃(𝐴)𝑃(𝐵)P(A∣B)=P(B)P(B∣A)⋅P(A)​*

***where***

* P*(*A*∣*B*) is the posterior probability of event A given event B.*
* *(*B*∣*A*) is the likelihood of event B given event A.*
* P*(*A*) is the prior probability of event A.*
* P*(*B*) is the total probability of event B.*

In the context of modeling hypotheses, Bayes’ theorem allows us to infer our belief in a hypothesis based on new data. We start with a prior belief in the hypothesis, represented by P(A), and then update this belief based on how likely the data are to be observed under the hypothesis, represented by P(B∣A). The posterior probability P(A∣B) represents our updated belief in the hypothesis after considering the data.

### Key Terms Related to Bayes Theorem

1. Likelihood(P(B∣A)):
   * Represents the probability of observing the given evidence (features) given that the class is true.
   * In the Naive Bayes algorithm, a key assumption is that features are conditionally independent given the class label. In other words, Naive Bayes works best with discrete features.
2. Prior Probability (P(A)):
   * In machine learning, this represents the probability of a particular class before considering any features.
   * It is estimated from the training data.
3. Evidence Probability( P(B) ):
   * This is the probability of observing the given evidence (features).
   * It serves as a normalization factor and is often calculated as the sum of the joint probabilities over all possible classes.
4. Posterior Probability( P(A∣B) ):
   * This is the updated probability of the class given the observed features.
   * It is what we are trying to predict or infer in a classification task.

Now, to utilise this in terms of machine learning we use the Naive Bayes Classifier but in order to understand how precisely this classifier works we must first understand the maths behind it.

## Applications of Bayes Theorem in Machine learning

### 1. Naive Bayes Classifier

The [Naive Bayes classifier](https://www.geeksforgeeks.org/naive-bayes-classifiers/) is a simple probabilistic classifier based on applying Bayes’ theorem with a strong (naive) independence assumption between the features. It is widely used for text classification, spam filtering, and other tasks involving high-dimensional data. Despite its simplicity, the Naive Bayes classifier often performs well in practice and is computationally efficient.

#### How it works?

* **Assumption of Independence:** The “naive” assumption in Naive Bayes is that the presence of a particular feature in a class is independent of the presence of any other feature, given the class. This is a strong assumption and may not hold true in real-world data, but it simplifies the calculation and often works well in practice.
* **Calculating Class Probabilities:** Given a set of features x1​,x2​,…,xn​, the Naive Bayes classifier calculates the probability of each class Ck​ given the features using Bayes’ theorem:  
  𝑃(𝐶𝑘​∣𝑥1​,𝑥2​,…,𝑥𝑛​)=𝑃(𝑥1​,𝑥2​,…,𝑥𝑛​∣𝐶𝑘​)⋅𝑃(𝐶𝑘​)𝑃(𝑥1​,𝑥2​,…,𝑥𝑛​)*P*(*Ck*​​∣*x*1​,*x*2​,…,*xn*​)=*P*(*x*1​,*x*2​,…,*xn*​)*P*(*x*1​,*x*2​,…,*xn*​∣*Ck*​​)⋅*P*(*Ck*​​)​,
  + the denominator P(x1​,x2​,…,xn​) is the same for all classes and can be ignored for the purpose of comparison.
* **Classification Decision:** The classifier selects the class Ck​ with the highest probability as the predicted class for the given set of features.

### 2. Bayes optimal classifier

The Bayes optimal classifier is a theoretical concept in machine learning that represents the best possible classifier for a given problem. It is based on Bayes’ theorem, which describes how to update probabilities based on new evidence.

In the context of classification, the Bayes optimal classifier assigns the class label that has the highest posterior probability given the input features. Mathematically, this can be expressed as:

𝑦^​=𝑎𝑟𝑔𝑚𝑎𝑥𝑦​(𝑦∣𝑥)*y*​​=*argmaxy*​​*P*(*y*∣*x*)

where 𝑦^*y*​​ is the predicted class label, y is a class label, **x** is the input feature vector, and P(y∣**x**) is the posterior probability of class y given the input features.

### 3. Bayesian Optimization

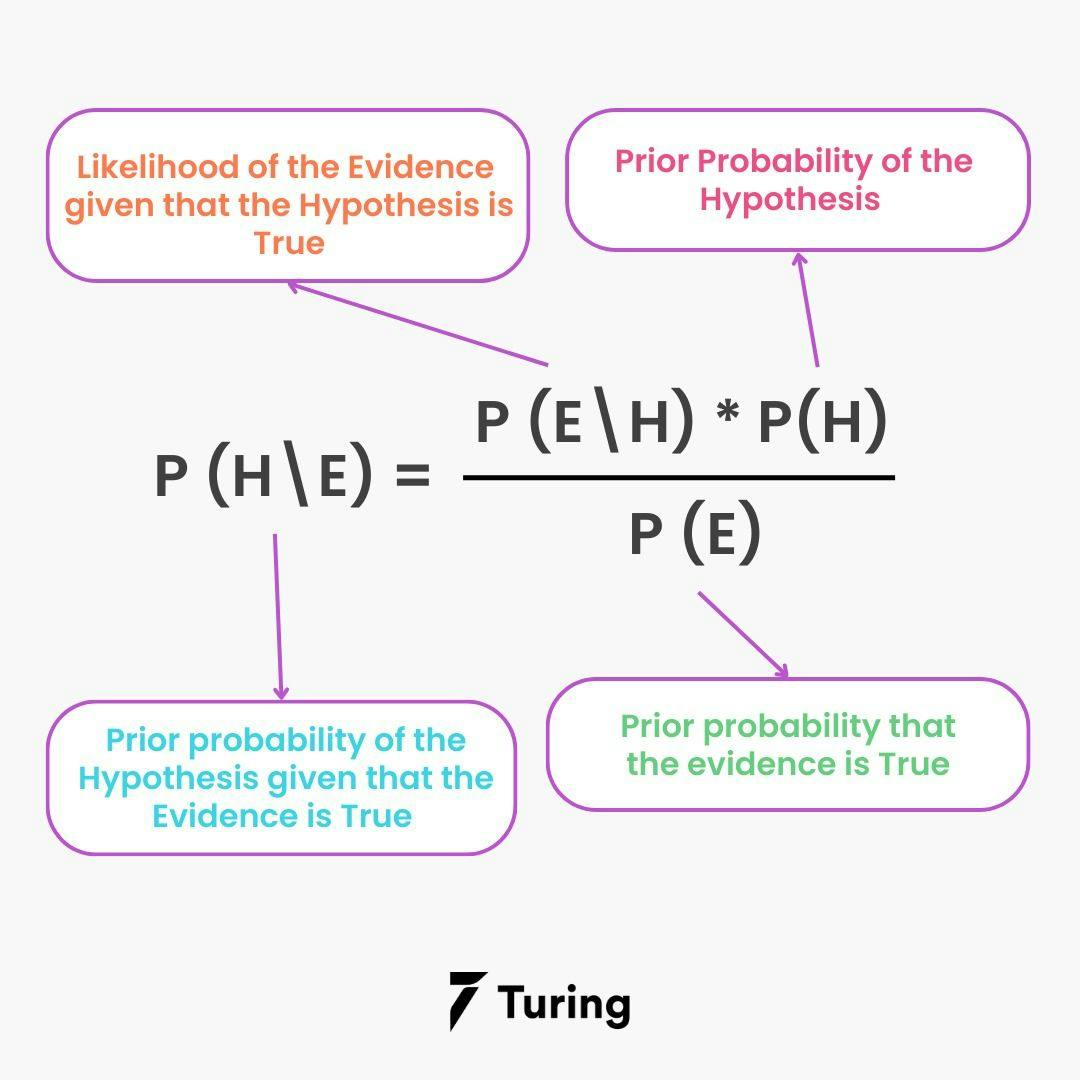
[Bayesian optimization](https://www.geeksforgeeks.org/hyperparameter-optimization-based-on-bayesian-optimization/) is a powerful technique for global optimization of expensive-to-evaluate functions. To choose which point to assess next, a probabilistic model of the objective function—typically based on a Gaussian process—is constructed. Bayesian optimization finds the best answer fast and requires few evaluations by intelligently searching the search space and iteratively improving the model. Because of this, it is especially well-suited for activities like machine learning model hyperparameter tweaking, where each assessment may be computationally costly.

### 4. Bayesian Belief Networks?

[Bayesian Belief Networks (BBNs)](https://www.geeksforgeeks.org/basic-understanding-of-bayesian-belief-networks/), also known as Bayesian networks, are probabilistic graphical models that represent a set of random variables and their conditional dependencies using a directed acyclic graph (DAG).The graph’s edges show the relationships between the nodes, which each represent a random variable.  
  
BBNs are employed for modeling uncertainty and generating probabilistic conclusions regarding the network’s variables. They may be used to provide answers to queries like “What is the most likely explanation for the observed data?” and “What is the probability of variable A given the evidence of variable B?”  
  
BBNs are extensively utilized in several domains, including as risk analysis, diagnostic systems, and decision-making. They are useful tools for reasoning under uncertainty because they provide complicated probabilistic connections between variables a graphical and understandable representation.

3. Offer an example of how the Nave Bayes classifier is used in real life.

## What is the Naive Bayes Algorithm?



It is an algorithm that learns the probability of every object, its features, and which groups they belong to. It is also known as a probabilistic classifier. The Naive Bayes Algorithm comes under supervised learning and is mainly used to solve classification problems.

For example, you cannot identify a bird based on its features and color as there are many birds with similar attributes. But, you make a probabilistic prediction about the same, and that is where the Naive Bayes Algorithm comes in.

## Probability, Bayes Theory, and Conditional Probability

Probability is the base for the Naive Bayes algorithm. This algorithm is built based on the probability results that it can offer for unsolvable problems with the help of prediction. You can learn more about probability, Bayes theory, and conditional probability below:

### Probability

Probability helps to predict an event's occurrence out of all the potential outcomes. The mathematical equation for probability is as follows:

Probability of an event = Number of Favorable Events/ Total number of outcomes

0 < = probability of an event < = 1. The favorable outcome denotes the event that results from the probability. Probability is always between 0 and 1, where 0 means no probability of it happening, and 1 means the success rate of that event is likely.

For better understanding, you can also consider a case where you predict a fruit based on its color and texture. Here are some possible assumptions that you can make. You can either choose the correct fruit that you have in mind or get confused with similar fruits and make mistakes. Either way, the probability of choosing the right fruit is 50%.

### Bayes Theory

Bayes Theory works on coming to a hypothesis (H) from a given set of evidence (E). It relates to two things: the probability of the hypothesis before the evidence P(H) and the probability after the evidence P(H|E). The Bayes Theory is explained by the following equation:

P(H|E) = (P(E|H} \* P(H))/P(E)

In the above equation,

* P(H|E) denotes how event H happens when event E takes place.
* P(E|H) represents how often event E happens when event H takes place first.
* P(H) represents the probability of event X happening on its own.
* P(E) represents the probability of event Y happening on its own.

The Bayes Rule is a method for determining P(H|E) from P(E|H). In short, it provides you with a way of calculating the probability of a hypothesis with the provided evidence.

### Conditional Probability

Conditional probability is a subset of probability. It reduces the probability of becoming dependent on a single event. You can compute the conditional probability for two or more occurrences.

When you take events X and Y, the conditional probability of event Y is defined as the probability that the event occurs when event X is already over. It is written as P(Y|X). The mathematical formula for this is as follows:

P(Y|A) = P(X and Y) /P(X)

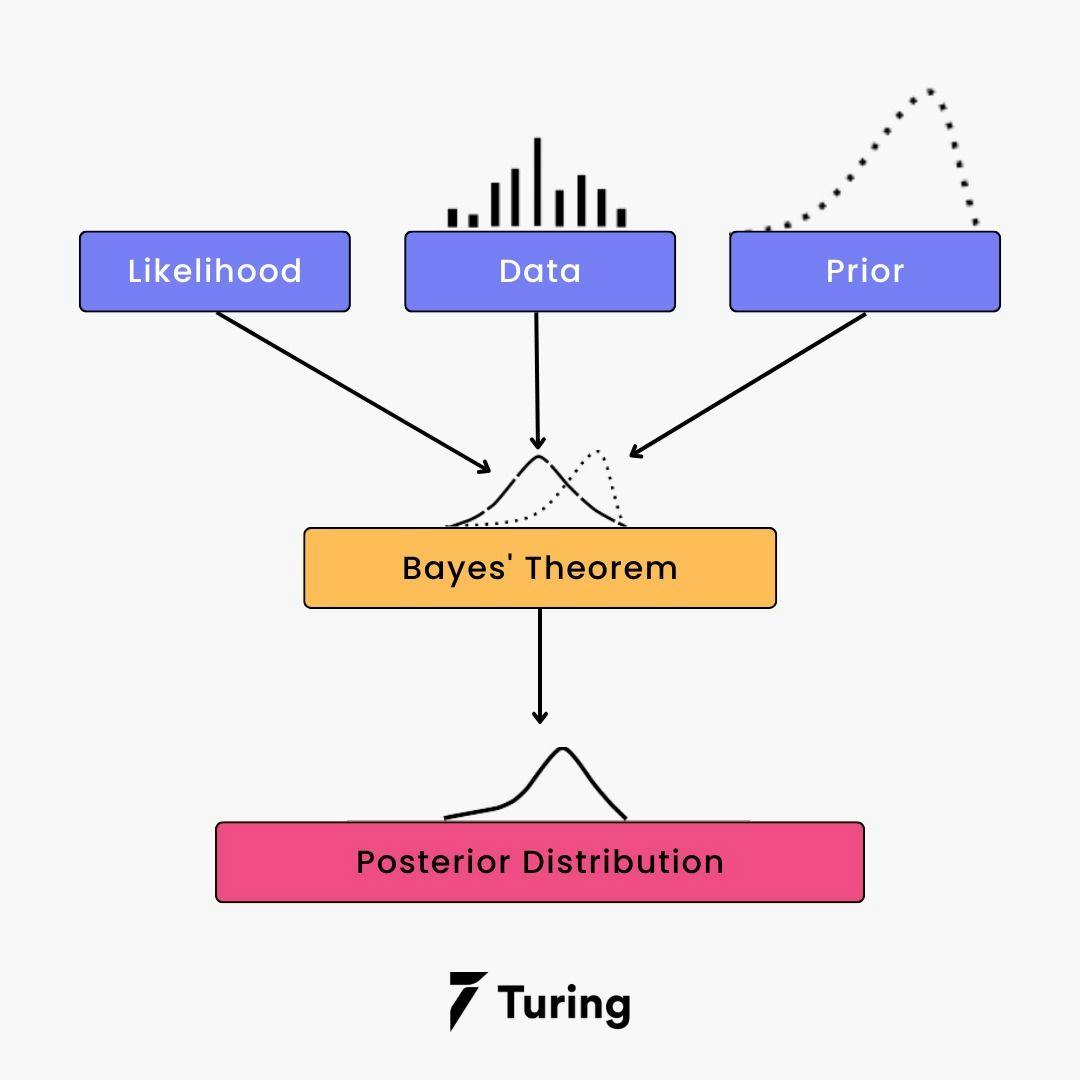
### Bayesian Probability

Bayesian Probability allows to calculate the conditional probabilities. It enables to use of partial knowledge for calculating the probability of the occurrence of a specific event. This algorithm is used for developing [models for prediction](https://www.turing.com/kb/all-you-have-to-know-about-predictive-modeling) and classification problems like Naive Bayes.

The Bayesian Rule is used in probability theory for computing - conditional probabilities. What is important is that you cannot discover just how the evidence will impact the probability of an event occurring, but you can find the exact probability.

## Bayes Theory from a machine learning standpoint

There are training data to train your model and make it functional. You then need to validate the data for evaluating the model and making new predictions. Finally, you need to call the input attributes “evidence” and label them “outputs” in the training data.



Using conditional probability denoted by P(E|O), you can calculate the probability of the evidence from the given outputs. Your ultimate goal is to compute P(O|E) - the probability of output based on the current attributes.

When the problem has two outputs, you can calculate the probability of every outcome and say which one wins. Whereas if you have various input attributes, then the Naive Bayesian Algorithm will be needed.

## How Naive Bayes Classifier works?

You can now try to build a classification model that uses Sklearn to see how the Naive Bayes Classifier works. Sklearn is also known as [Scikit-Learn](https://www.turing.com/kb/scikit-learn-cheatsheet-methods-for-classification-and-regression" \t "_self). It is an open-source machine-learning library that is written in Python.

For instance, you are using the social\_media\_ads dataset. With this problem, you can predict if a user has purchased a product by clicking on the ad, depending on her age and other attributes. You can understand the working of the Naive Bayes Classifier by following the below steps:

**Step 1 - Import basic libraries**

You can use the below command for importing the basic libraries required.

# Importing basic libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

**Step 2 - Importing the dataset**

Using the below code, import the dataset, which is required.

# Importing the dataset

dataset = pd.read\_csv(‘Social\_Media\_Ads.csv’)

X = dataset.iloc[:, [3, 4]]

y = dataset.iloc[:, 5]

print(“Prediction evidence:\n”, X.head())

print(“\nFinal Target:\n”, y.head())

**Step 3 - Data preprocessing**

The below command will help you with the data preprocessing.

# Conversion of variables into arrays

X = X.values

y = y.values

# Dataset splitting into training and test datasets(70:30)

from sklearn.selection\_of\_model import splitting\_of\_train\_test\_dataset

X\_train, X\_test, y\_train, y\_test = splitting\_of\_train\_test\_dataset(X, y, test\_size = 0.30)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.transform\_fit(X\_train)

X\_test = sc.transform(X\_test)

In this step, you have to split the dataset into a training dataset (70%) and a testing dataset (30%). Next, you have to do some basic feature scaling with the help of a standard scaler. It will transform the dataset in a way where the mean value will be 0, and the standard deviation will be 1.

**Step 4 - Training the model**

You should then write the following command for training the model.

# Fitting of Naive Bayes Algorithm to the Training Dataset

from sklearn.naive\_bayes\_algorithm import GaussianNB

classifier = GaussianNB()

classifier.fit(X\_train, y\_train)

**Step 5 - Testing and evaluation of the model**

The code for testing and evaluating the model is as below:

# Prediction of the test dataset outcomes

y\_pred = classifier.predict(X\_test)

# Constructing the confusion matrix

import seaborn as sns

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True)

A confusion matrix helps to understand the quality of the model. It describes the production of a classification model on a set of test data for which you know the true values. Every row in a confusion matrix portrays an actual class, and every column portrays the predicted class.

**Step 6 - Visualizing the model**

Finally, the below code will help in visualizing the model.

# Visualizing the test dataset results

from matplotlib.colors import ColormapListed

X\_datsetset, y\_datasetset = X\_test, y\_test

X1, X2 = np.meshgrid(np.arrange(start = X\_dataset[:,0].min()-1, stop = X\_dataset[:. 0].max() + 1, step =

np.arrange(start = X\_dataset[:, 1[.min() -1, stop = X\_dataset[:, 1].max() +1, step = 0.02))

plt.contourf(X1, X2, Classifier.predict(np.array([X1.ravel(), X2.ravel()].T).rescape(X1.shape),

alpha = 0.3, cmap = ColormapListed((‘yellow’, ‘blue’)))

plt.xlim(X1.min(), X1.max())

plt.ylim(X2.min(), X2.max())

for u, v in enumerate(np.unique(y\_set)):

plt.scatter(X\_dataset[y\_dataset == v, 0], X\_dataset[y\_dataset== v, 1],

c = ColormapListed((‘yellow’,’blue’))(i), label = v)

plt.xlabel(‘Current\_age’)

plt.ylabel(‘Gross\_salary’)

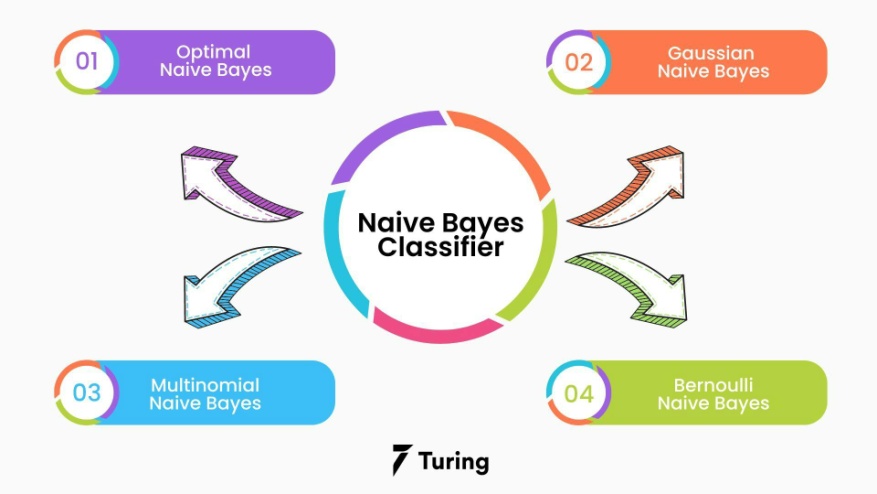
plt.legend()

plt.show()

**Code Idea:** Towardsdatascience.com

However, in some cases, these steps might not be absolutely necessary. But the above-mentioned example provides a clear idea and information about how data points can be classified.

## Types of the Naive Bayes Model



There are four types of the Naive Bayes Model, which are explained below:

### Gaussian Naive Bayes

It is a straightforward algorithm used when the attributes are continuous. The attributes present in the data should follow the rule of Gaussian distribution or normal distribution. It remarkably quickens the search, and under lenient conditions, the error will be two times greater than Optimal Naive Bayes.

### Optimal Naive Bayes

Optimal Naive Bayes selects the class that has the greatest posterior probability of happenings. As per the name, it is optimal. But it will go through all the possibilities, which is very slow and time-consuming.

### Bernoulli Naive Bayes

Bernoulli Naive Bayes is an algorithm that is useful for data that has binary or boolean attributes. The attributes will have a value of yes or no, useful or not, granted or rejected, etc.

### Multinominal Naive Bayes

Multinominal Naive Bayes is used on documentation classification issues. The features needed for this type are the frequency of the words converted from the document.

## Advantages of a Naive Bayes Classifier

Here are some advantages of the Naive Bayes Classifier:

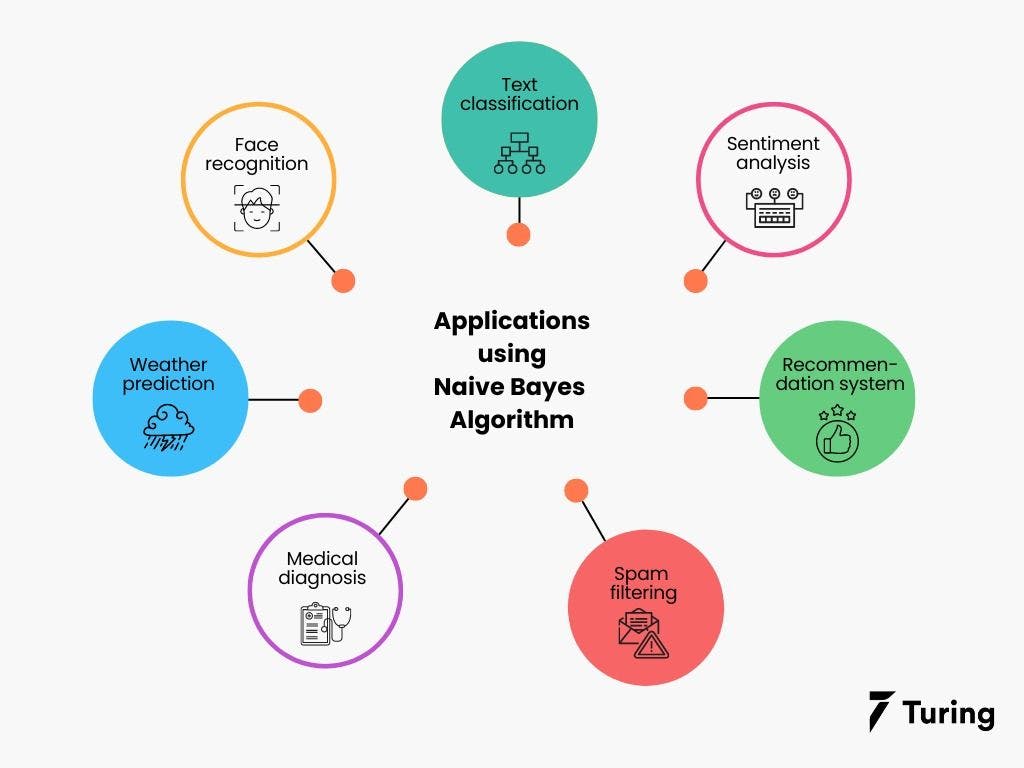
* It doesn’t require larger amounts of training data.
* It is straightforward to implement.
* Convergence is quicker than other models, which are discriminative.
* It is highly scalable with several data points and predictors.
* It can handle both continuous and categorical data.
* It is not sensitive to irrelevant data and doesn’t follow the assumptions it holds.
* It is used in real-time predictions.

## Disadvantages of a Naive Bayes Classifier

The disadvantage of the Naive Bayes Classifier are as below:

* The Naive Bayes Algorithm has trouble with the ‘zero-frequency problem’. It happens when you assign zero probability for categorical variables in the training dataset that is not available. When you use a smooth method for overcoming this problem, you can make it work the best.
* It will assume that all the attributes are independent, which rarely happens in real life. It will limit the application of this algorithm in real-world situations.
* It will estimate things wrong sometimes, so you shouldn’t take its probability outputs seriously.

## Applications that use Naive Bayes



The Naive Bayes Algorithm is used for various real-world problems like those below:

* **Text classification:** The Naive Bayes Algorithm is used as a probabilistic learning technique for text classification. It is one of the best-known algorithms used for [document classification](https://www.turing.com/kb/document-classification-using-naive-bayes) of one or many classes.
* **Sentiment analysis:** The Naive Bayes Algorithm is used to analyze sentiments or feelings, whether positive, neutral, or negative.
* **Recommendation system:** The Naive Bayes Algorithm is a collection of collaborative filtering issued for building hybrid recommendation systems that assist you in predicting whether a user will receive any resource.
* **Spam filtering:** It is also similar to the text classification process. It is popular for helping you determine if the mail you receive is spam.
* **Medical diagnosis:** This algorithm is used in medical diagnosis and helps you to predict the patient’s risk level for certain diseases.
* **Weather prediction:** You can use this algorithm to predict whether the weather will be good.
* **Face recognition:** This helps you identify faces.

## Wrapping Up

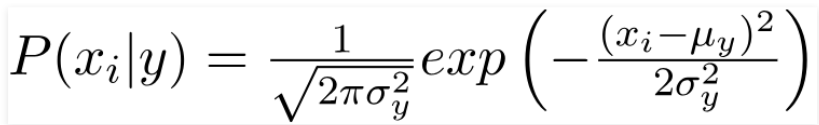
4. Can the Nave Bayes classifier be used on continuous numeric data? If so, how can you go about doing it?

**Gaussian Naive Bayes**

So far, we have discussed how to predict probabilities if the predictors take up discrete values. But what if they are continuous? For this, we need to make some more assumptions regarding the distribution of each feature. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of P(xi | y). Here we’ll discuss Gaussian Naïve Bayes.

Gaussian Naïve Bayes is used when we assume all the continuous variables associated with each feature to be distributed according to **Gaussian Distribution.**Gaussian Distribution is also called Normal distribution.

The conditional probability changes here since we have different values now. Also, the (PDF)  probability density function of a normal distribution is given by:



We can use this formula to compute the probability of likelihoods if our data is continuous.

**Endnotes**

Naive Bayes [algorithms](https://www.analyticsvidhya.com/blog/2023/01/naive-bayes-algorithms-a-complete-guide-for-beginners/) are mostly used in face recognition, weather prediction, Medical Diagnosis, News classification, Sentiment Analysis, etc. In this article, we learned the mathematical intuition behind Naive bayes algorithm in Machine learning.  You have already taken your first step to master this algorithm and from here all you need is practi

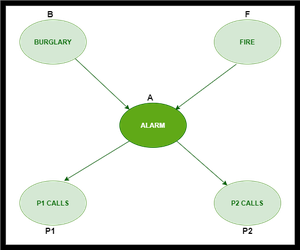
5. What are Bayesian Belief Networks, and how do they work? What are their applications? Are they capable of resolving a wide range of issues?

# Basic Understanding of Bayesian Belief Networks

**Bayesian Belief Network**is a graphical representation of different probabilistic relationships among random variables in a particular set. It is a classifier with no dependency on attributes i.e it is condition independent. Due to its feature of joint probability, the probability in Bayesian Belief Network is derived, based on a condition — P(attribute/parent) i.e probability of an attribute, true over parent attribute.

(Note: A classifier assigns data in a collection to desired categories.)

* Consider this example:



* In the above figure, we have an alarm ‘A’ – a node, say installed in a house of a person ‘gfg’, which rings upon two probabilities i.e burglary ‘B’ and fire ‘F’, which are – parent nodes of the alarm node. The alarm is the parent node of two probabilities P1 calls  ‘P1’ & P2 calls ‘P2’ person nodes.
* Upon the instance of burglary and fire, ‘P1’ and ‘P2’ call person ‘gfg’, respectively. But, there are few drawbacks in this case, as sometimes ‘P1’ may forget to call the person ‘gfg’, even after hearing the alarm, as he has a tendency to forget things, quick.  Similarly, ‘P2’, sometimes fails to call the person ‘gfg’, as he is only able to hear the alarm, from a certain distance.

**Q)** Find the probability that ‘P1’ is true (P1 has called ‘gfg’), ‘P2’ is true (P2 has called ‘gfg’) when the alarm ‘A’ rang, but no burglary ‘B’ and fire ‘F’ has occurred.

=> **P ( P1, P2, A, ~B, ~F)** [ where- P1, P2 & A are ‘true’ events and ‘~B’ & ‘~F’ are ‘false’ events]

[ **Note:** The values mentioned below are neither calculated nor computed. They have observed values ]

***Burglary ‘B’ –***

* **P (B=T) = 0.001** (‘B’ is true i.e burglary has occurred)
* **P (B=F) = 0.999**(‘B’ is false i.e burglary has not occurred)

***Fire ‘F’ –***

* **P (F=T) = 0.002** (‘F’ is true i.e fire has occurred)
* **P (F=F) = 0.998** (‘F’ is false i.e fire has not occurred)

***Alarm ‘A’ –***

|  |  |  |  |
| --- | --- | --- | --- |
| **B** | **F** | **P (A=T)** | **P (A=F)** |
| T | T | 0.95 | 0.05 |
| T | F | 0.94 | 0.06 |
| F | T | 0.29 | 0.71 |
| F | F | 0.001 | **0.999** |

* The alarm ‘A’ node can be ‘true’ or ‘false’ ( i.e may have rung or may not have rung). It has two parent nodes burglary ‘B’ and fire ‘F’ which can be ‘true’ or ‘false’ (i.e may have occurred or may not have occurred) depending upon different conditions.

***Person ‘P1’ –***

|  |  |  |
| --- | --- | --- |
| **A** | **P (P1=T)** | **P (P1=F)** |
| T | **0.95** | 0.05 |
| F | 0.05 | 0.95 |

* The person ‘P1’ node can be ‘true’ or ‘false’ (i.e may have called the person ‘gfg’ or not) . It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung ,upon burglary ‘B’ or fire ‘F’).

***Person ‘P2’ –***

|  |  |  |
| --- | --- | --- |
| **A** | **P (P2=T)** | **P (P2=F)** |
| T | **0.80** | 0.20 |
| F | 0.01 | 0.99 |

* The person ‘P2’ node can be ‘true’ or false’ (i.e may have called the person ‘gfg’ or not). It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung, upon burglary ‘B’ or fire ‘F’).

**Solution:** Considering the observed probabilistic scan –

With respect to the question —  **P ( P1, P2, A, ~B, ~F)**, we need to get the probability of ‘P1’. We find it with regard to its parent node – alarm ‘A’. To get the probability of ‘P2’, we find it with regard to its parent node — alarm ‘A’.

We find the probability of alarm ‘A’ node with regard to ‘~B’ & ‘~F’ since burglary ‘B’ and fire ‘F’ are parent nodes of alarm ‘A’.

From the observed probabilistic scan, we can deduce –

**P ( P1, P2, A, ~B, ~F)**

**= P (P1/A) \* P (P2/A) \* P (A/~B~F) \* P (~B) \* P (~F)**

**= 0.95 \* 0.80 \* 0.001 \* 0.999 \* 0.998**

**= 0.00075**

6. Passengers are checked in an airport screening system to see if there is an intruder. Let I be the random variable that indicates whether someone is an intruder I = 1) or not I = 0), and A be the variable that indicates alarm I = 0). If an intruder is detected with probability P(A = 1|I = 1) = 0.98 and a non-intruder is detected with probability P(A = 1|I = 0) = 0.001, an alarm will be triggered, implying the error factor. The likelihood of an intruder in the passenger population is P(I = 1) = 0.00001. What are the chances that an alarm would be triggered when an individual is actually an intruder?

# Bayesian Belief Network in artificial intelligence

Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:

"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.

Backward Skip 10sPlay VideoForward Skip 10s

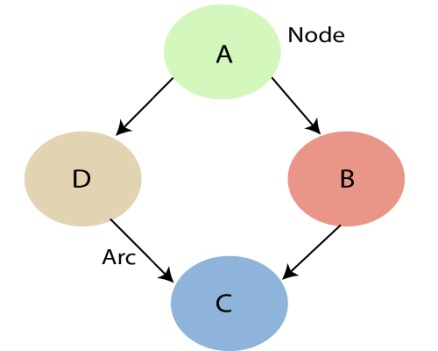
Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision 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**.

**A Bayesian network graph is made up of nodes and Arcs (directed links), where:**



* 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.

Bayesian network is based on Joint probability distribution and conditional probability. So let's first understand the joint probability distribution:

## 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].**

In general for each variable Xi, we can write the equation as:

P(Xi|Xi-1,........., X1) = P(Xi |Parents(Xi ))

## 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 neighbors 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:

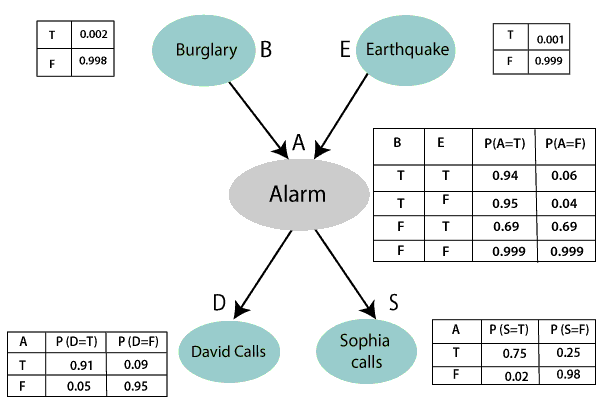
**P[D, S, A, B, E]= P[D | S, A, B, E]. P[S, A, B, E]**

**=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]**



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:

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

**Conditional probability table for David Calls:**

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

|  |  |  |
| --- | --- | --- |
| **A** | **P(D= True)** | **P(D= False)** |
| True | 0.91 | 0.09 |
| False | 0.05 | 0.95 |

**Conditional probability table for Sophia Calls:**

The Conditional probability of Sophia that she calls is depending on its Parent Node "Alarm."

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

From the formula of joint distribution, we can write the problem statement in the form of probability distribution:

**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.

7. An antibiotic resistance test (random variable T) has 1% false positives (i.e., 1% of those who are not immune to an antibiotic display a positive result in the test) and 5% false negatives (i.e., 1% of those who are not resistant to an antibiotic show a positive result in the test) (i.e. 5 percent of those actually resistant to an antibiotic test negative). Assume that 2% of those who were screened were antibiotic-resistant. Calculate the likelihood that a person who tests positive is actually immune (random variable D).

**Suppose a certain disease has an incidence rate of 0.1% (that is, it afflicts 0.1% of the population)**. The percentage 0.1% can be converted to a decimal number by moving the decimal place two places to the left, to get 0.001. In turn, 0.001 can be rewritten as a fraction: 1/1000. This tells us that about 1 in every 1000 people has the disease. (If we wanted we could write P(disease)=0.001.)

**A test has been devised to detect this disease.  The test does not produce false negatives (that is, anyone who has the disease will test positive for it)**. This part is fairly straightforward: everyone who has the disease will test positive, or alternatively everyone who tests negative does not have the disease. (We could also say P(positive | disease)=1.)

**The false positive rate is 5% (that is, about 5% of people who take the test will test positive, even though they do not have the disease)**. This is even more straightforward. Another way of looking at it is that of every 100 people who are tested and do not have the disease, 5 will test positive even though they do not have the disease. (We could also say that P(positive | no disease)=0.05.)

**Suppose a randomly selected person takes the test and tests positive.  What is the probability that this person actually has the disease?** Here we want to compute P(disease|positive). We already know that P(positive|disease)=1, but remember that conditional probabilities are not equal if the conditions are switched.

Rather than thinking in terms of all these probabilities we have developed, let’s create a hypothetical situation and apply the facts as set out above. First, suppose we randomly select 1000 people and administer the test. How many do we expect to have the disease? Since about 1/1000 of all people are afflicted with the disease, 1/1000 of 1000 people is 1. (Now you know why we chose 1000.) Only 1 of 1000 test subjects actually has the disease; the other 999 do not.

We also know that 5% of all people who do not have the disease will test positive. There are 999 disease-free people, so we would expect (0.05)(999)=49.95 (so, about 50) people to test positive who do not have the disease.

Now back to the original question, computing P(disease|positive). There are 51 people who test positive in our example (the one unfortunate person who actually has the disease, plus the 50 people who tested positive but don’t). Only one of these people has the disease, so

P(disease | positive) ≈151≈0.0196≈151≈0.0196

or less than 2%. Does this surprise you? This means that of all people who test positive, over 98% do not have the disease.

The answer we got was slightly approximate, since we rounded 49.95 to 50. We could redo the problem with 100,000 test subjects, 100 of whom would have the disease and (0.05)(99,900)=4995 test positive but do not have the disease, so the exact probability of having the disease if you test positive is

P(disease | positive) ≈1005095≈0.0196≈1005095≈0.0196

which is pretty much the same answer.But back to the surprising result. Of all people who test positive, over 98% do not have the disease.  If your guess for the probability a person who tests positive has the disease was wildly different from the right answer (2%), don’t feel bad. The exact same problem was posed to doctors and medical students at the Harvard Medical School 25 years ago and the results revealed in a 1978 New England Journal of Medicine article. Only about 18% of the participants got the right answer. Most of the rest thought the answer was closer to 95% (perhaps they were misled by the false positive rate of 5%).

So at least you should feel a little better that a bunch of doctors didn’t get the right answer either (assuming you thought the answer was much higher). But the significance of this finding and similar results from other studies in the intervening years lies not in making math students feel better but in the possibly catastrophic consequences it might have for patient care. If a doctor thinks the chances that a positive test result nearly guarantees that a patient has a disease, they might begin an unnecessary and possibly harmful treatment regimen on a healthy patient.  Or worse, as in the early days of the AIDS crisis when being HIV-positive was often equated with a death sentence, the patient might take a drastic action and commit suicide.

8. In order to prepare for the test, a student knows that there will be one question in the exam that is either form A, B, or C. The chances of getting an A, B, or C on the exam are 30 percent, 20%, and 50 percent, respectively. During the planning, the student solved 9 of 10 type A problems, 2 of 10 type B problems, and 6 of 10 type C problems.

1. What is the likelihood that the student can solve the exam problem?

Let E1: person A get selected

E2: person B get selected

E3: person C get selected

A: Changes introduced but profit not happened

Now, P(E1) = 1/(1+2+4) = 1/7

P(E2) = 2/7 and P(E3) = 4/7

P(A|E1) = P(Profit not happened by the changes introduces by A) = 1 – P(Profit happened by the changes introduces by A) = 1 – 0.8 = 0.2

P(A|E2) = P(Profit not happened by the changes introduces by B) = 1 – P(Profit happened by the changes introduces by B) = 1 – 0.5 = 0.5

P(A|E3) = P(Profit not happened by the changes introduces by C) = 1 – P(Profit happened by the changes introduces by C) = 1 – 0.3 = 0.7

We have to find the probability of not happening profit due to selection of C

𝑃(𝐸3|𝐴)=𝑃(𝐴|𝐸3)𝑃(𝐸3)𝑃(𝐴|𝐸1)𝑃(𝐸1)+𝑃(𝐴|𝐸2)𝑃(𝐸2)+𝑃(𝐴|𝐸3)𝑃(𝐸3)

(𝐸3|𝐴)=0.7×470.2×17+0.5×27+0.7×47

= 7/10.

∴ the required probability is 0.7.

2. Given the student's solution, what is the likelihood that the problem was of form A?

Let,

F: children with flu

M: children with measles

R: children showing the symptom of rash

P(F) = 90% = 0.9

P(M) = 10% = 0.1

P(R|F) = 0.08

P(R|M) = 0.95

𝑃(𝐹|𝑅)=𝑃(𝑅|𝐹)𝑃(𝐹)𝑃(𝑅|𝑀)𝑃(𝑀)+𝑃(𝑅|𝐹)𝑃(𝐹)

(𝐹|𝑅)=0.08×0.90.95×0.1+0.08×0.9

= 0.072/(0.095 + 0.072) = 0.072/0.167 ≈ 0.43

⇒ P(F|R) = 0.43

9. A bank installs a CCTV system to track and photograph incoming customers. Despite the constant influx of customers, we divide the timeline into 5 minute bins. There may be a customer coming into the bank with a 5% chance in each 5-minute time period, or there may be no customer (again, for simplicity, we assume that either there is 1 customer or none, not the case of multiple customers). If there is a client, the CCTV will detect them with a 99 percent probability. If there is no customer, the camera can take a false photograph with a 10% chance of detecting movement from other objects.

1. How many customers come into the bank on a daily basis (10 hours)?

2. On a daily basis, how many fake photographs (photographs taken when there is no customer) and how many missed photographs (photographs taken when there is a customer) are there?

3. Explain likelihood that there is a customer if there is a photograph?

## The correct option is A 0.38

**Average no. of customers**=30/hr=0.5/min=μ **Poisson distribution function is**f(t)=μe−μt[∵**Inter arrival time follow) Exponential distribution]**and P(0<t<t1)=∫t10f(t)dt=1−e−μt1so, P(0≤t≤1)=1−e−μ(1)=1−e−1/2=0.393 and P(0≤t≤3)=1−e−3μ=1−e−32=0.7768 **so, P(successive customer arrival between 1 to 3 min)**=P(1≤t≤3)=0.7768−0.398=0.38

10. Create the conditional probability table associated with the node Won Toss in the Bayesian Belief network to represent the conditional independence assumptions of the Nave Bayes classifier for the match winning prediction problem in Section 6.4.4.

# Bayesian Belief Network in artificial intelligence

Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:

"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.

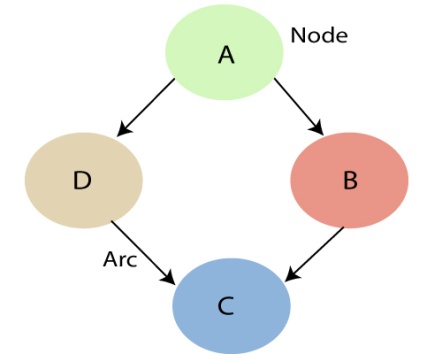
Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision 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**.

**A Bayesian network graph is made up of nodes and Arcs (directed links), where:**



* 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.

Bayesian network is based on Joint probability distribution and conditional probability. So let's first understand the joint probability distribution:

## 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].**

In general for each variable Xi, we can write the equation as:

P(Xi|Xi-1,........., X1) = P(Xi |Parents(Xi ))

## 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 neighbors 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:

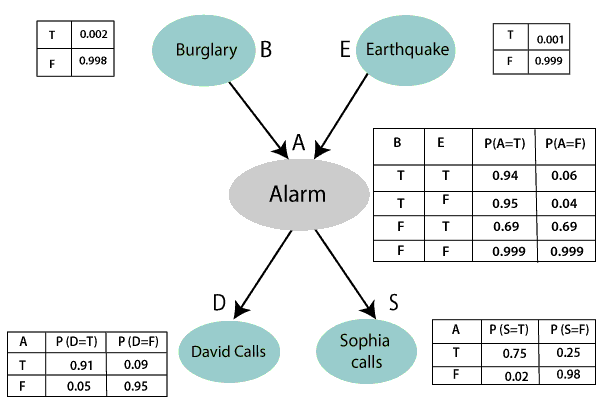
**P[D, S, A, B, E]= P[D | S, A, B, E]. P[S, A, B, E]**

**=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]**



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:

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

**Conditional probability table for David Calls:**

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

|  |  |  |
| --- | --- | --- |
| **A** | **P(D= True)** | **P(D= False)** |
| True | 0.91 | 0.09 |
| False | 0.05 | 0.95 |

**Conditional probability table for Sophia Calls:**

The Conditional probability of Sophia that she calls is depending on its Parent Node "Alarm."

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

From the formula of joint distribution, we can write the problem statement in the form of probability distribution:

**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.