Module 4

Probabilistic Based Models

- **Naive Bayes:** Introduction to Naive Bayes Classifier-Bayes' Theorem and Conditional Probability-Gaussian,Multinomial, and Bernoulli Naive Bayes. Bayesian Belief Network-EM algorithm.

Probabilistic Model

- A Probabilistic model in machine learning is a mathematical representation of a real-world process that incorporates uncertain or random variables.
- These are the models that incorporate random variables and probability distributions.
- Random variables represent the potential outcomes of an uncertain event
- Probability distribution assign probabilities to the various potential outcomes.
- Probabilistic models are used in practice because realistic decision making often necessitates recognizing uncertainty.

What is probabilistic model?????

- Let’s consider a classification problem with N classes. If the classification model (classifier) is probabilistic, for a given input, it will provide probabilities for each class (of the N classes) as the output. In other words, a probabilistic classifier will provide a probability distribution over the N classes. Usually, the class with the highest probability is then selected as the Class for which the input data instance belongs.

Probabilistic ModelVs Non probabilistic -An example....

- Take the task of classifying an image of an animal into five classes — {Dog, Cat, Deer, Lion, Rabbit} as the problem. As input, we have an image (of a dog). For this example, let's consider that the classifier works well and provides correct/ acceptable results for the particular input we are discussing. When the image is provided as the input to the probabilistic classifier, it will provide an output such as (Dog (0.6), Cat (0.2), Deer(0.1), Lion(0.04), Rabbit(0.06)). But, if the classifier is non-probabilistic, it will only output "Dog".

Examples

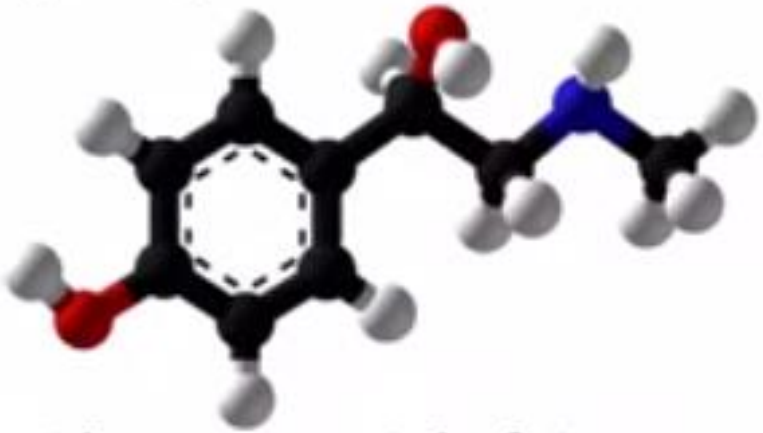
Oil prices

If you run an energy intensive business, an airline for example, then the price of oil is a key determinant of your profitability

For medium or long-term investment planning (buying new planes) the future price of oil is an important consideration

But who knows the price of oil in ten years? No-one. But we may be able to put a probability distribution around the future price and incorporate the uncertainty into the decision making process

Valuing a drug development company



- A company has 10 drugs in a development portfolio
- Given a drug has been approved, you have predicted its revenue
- But whether a drug is approved or not is an uncertain future event (a random variable). You have estimated the probability of approval
- You only wish to invest in the company if the company's expected total revenue for the portfolio is over $10B in 5 years time
- You need to calculate the ***probability distribution*** of the total revenue to understand the investment risk

Categories Of Probabilistic Models

These models can be classified into the following categories:

- Generative models
- Discriminative models.
- Graphical models
- **Generative models** aim to model the **joint distribution** of the input and output variables. These models generate new data based on the probability distribution of the original dataset. A generative model can model a distribution by producing fake data that looks like it's drawn from that distribution.
- eg: GAN, Bayesian Network
- **The discriminative model** aims to model the **conditional distribution** of the output variable given the input variable. They learn a decision boundary that separates the different classes of the output variable.eg:Logistic Regression

- **Graphical models** use **graphical representations** to show the conditional dependence between variables. They are commonly used for tasks such as image recognition, natural language processing, and causal inference.eg:DAG

Advantages

- Ability to take into account uncertainty and variability in data. This allows for more accurate predictions and decision-making, particularly in complex and unpredictable situations.
- Provide insights into how different factors influence outcomes and can help identify patterns and relationships within data.

Terms Related to Bayes Theorem

- **Hypotheses:** Events happening in the sample space **E_1, E_2,… E_n is called the hypotheses .**
- **Eg:**S={1,2,3,4,5,6}
- **Hypothesis 1 (E1):** The outcome is an **even number**.
- This event can be written as: E1={2,4,6}
- **Hypothesis 2 (E2):** The outcome is a **prime number**.
- The prime numbers in the sample space are: E2={2,3,5}
- **Hypothesis 3 (E3):** The outcome is a **number greater than 4**.
- This event can be written as: E3={5,6}
- Each of these hypotheses (E1, E2, E3) represents a different scenario or set of possible outcomes in the experiment.

- **Priori Probability:** **A priori probability** (also known as **theoretical probability**) refers to the probability that is calculated before any experiment or observation is conducted. It is determined based on known information.
- For a discrete sample space S, the a priori probability P(A) of an event Ais given by:

$$P(A) = \frac{\text{Number of favorable outcomes for } A}{\text{Total number of possible outcomes in the sample space } S}$$

- Suppose you roll a fair 6-sided die. The sample space S consists of 6 outcomes: S={1,2,3,4,5,6}
- The even numbers in the sample space are: {2,4,6}
- The number of favorable outcomes (even numbers) is 3. The total number of possible outcomes is 6.
- priori probability of rolling an even number is:P(A) =3/6

- **Posterior Probability:** Posterior Probability is the updated probability of an event after considering new information.
- Bayes' Theorem provides a way to calculate the **posterior probability**. P(H|E) of a hypothesis H given the evidence E. The formula is:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

- P(H|E) is the **posterior probability** — the probability of the hypothesis H after observing the evidence E.
- P(E|H) is the **likelihood** — the probability of observing the evidence E given that the hypothesis H is true.
- P(H) is the **prior probability** — the probability of the hypothesis H before observing the evidence E.
- P(E) is the **marginal likelihood** or the total probability of observing the evidence E across all possible hypotheses.

- For example, let there be two urns, urn A having 5 black balls and 10 red balls and urn B having 10 black balls and 5 red balls. Now if an urn is selected at random, the probability that urn A is chosen is 0.5. This is the a *priori probability*.

- An **additional piece of information t**hat a ball was drawn at random from the selected urn, and that ball was black, what is the probability that the chosen urn is urn A? **Posterior probability takes into account this additional information** and revises the probability downward from 0.5 to 0.333 according to Bayes´ theorem, because a black ball is more probable from urn B than urn A.

- **Conditional Probability**
- The probability of an event A based on the occurrence of another event B is termed conditional Probability.
- It is denoted as **P(A|B)** and represents the probability of A when event B has already happened
- **Joint Probability**
- When the probability of two more events occurring together and at the same time is measured it is marked as Joint Probability. For two events A and B, it is denoted by joint probability is denoted as, **P(A∩B).**
- **Random Variables**
- Real-valued variables whose possible values are determined by random experiments are called random variables. The probability of finding such variables is the experimental probability.

Baye's Theorem

- Bayes theorem works on the principle of Conditional Probability.
- The general statement of Bayes' theorem is "**The conditional probability of an event A, given the occurrence of another event B, is equal to the product of the event of B, given A and the probability of A divided by the probability of event B.**

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

- where,
- **P(A)** and **P(B)** are the probabilities of events A and B
- **P(A|B)** is the probability of event A when event B happens
- **P(B|A)** is the probability of event B when A happens

Bayes Theorem

Proof of Bayes Theorem

The probability of two events A and B happening, P(A∩B), is the probability of A, P(A), times the probability of B given that A has occurred, P(B|A).

$$P(A \cap B) = P(A)P(B|A) \tag{1}$$

On the other hand, the probability of A and B is also equal to the probability of B times the probability of A given B.

$$P(A \cap B) = P(B)P(A|B) \tag{2}$$

Equating the two yields:

$$P(B)P(A|B) = P(A)P(B|A) \tag{3}$$

and thus

$$P(A|B) = P(A)\frac{P(B|A)}{P(B)} \tag{4}$$

This equation, known as Bayes Theorem is the basis of statistical inference.

- We built a robot that can detect defective items produced in our factory:
- if an item is defective, it is spotted with 98% probability by the robot;
- when an item is not defective, the robot will not signal any defect with 99% probability.
- We draw an item at random from a production lot in which 0.1% of items are defective.
- If the robot tells us that the drawn item is defective, what is the probability that the robot is right?

$$P(\text{robot says defective} \mid \text{defective}) = 0.98$$

$$P(\text{robot says defective} \mid \text{not defective}) = 1 - 0.99 = 0.01$$

$$P(\text{defective}) = 0.001$$

$$P(\text{not defective}) = 1 - 0.001 = 0.999$$

the unconditional probability that the robot signals a defective item can be derived using the law of total probability

$$\begin{aligned} & P(\text{robot says defective}) \\ = \; & P(\text{robot says defective} \mid \text{defective}) P(\text{defective}) \\ & + P(\text{robot says defective} \mid \text{not defective}) P(\text{not defective}) \\ = \; & 0.98 \cdot 0.001 + 0.01 \cdot 0.999 \\ = \; & 0.00098 + 0.00999 \\ = \; & 0.01097 \end{aligned}$$

law of total probability

$$P(E) = P(E|I_1)P(I_1) + \ldots + P(E|I_n)P(I_n)$$

- Therefore, Bayes' rule gives

$$
\begin{aligned}
&P(\text{defective} \mid \text{robot says defective}) \\
&= \frac{P(\text{robot says defective} \mid \text{defective})P(\text{defective})}{P(\text{robot says defective})} \\
&= \frac{0.98 \cdot 0.001}{0.01097} \\
&= \frac{0.00098}{0.01097} \simeq 0.08933
\end{aligned}
$$

Example-Bayes Theorem

- Given:
  - A doctor knows that Cold causes fever 50% of the time
  - Prior probability of any patient having cold is 1/50,000
  - Prior probability of any patient having fever is 1/20

- If a patient has fever, what's the probability he/she has cold?

$$P(C \mid F) = \frac{P(F \mid C)P(C)}{P(F)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Naïve Bayes Classifier

- A Naive Bayes classifier is a probabilistic machine learning model that is used for classification task
- It is based on Baye's Theorem

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. The assumption made here is that the predictors/features are independent.

It is mainly used in *text classification* that includes a high-dimensional training dataset

It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Eg: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.

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.
- **Problem:** Players will play if the weather is sunny. Is this statement correct?
- So to solve this problem, we need to follow the below steps:

1.Convert the given dataset into frequency tables.

2.Generate Likelihood table by finding the probabilities of given features.

3.Now, use Bayes theorem to calculate the posterior probability.

	Weather	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	5

Likelihood table weather condition:

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)**

Bayes Algorithm

Cosider the below example dataframe

f1	f2	f3	f4.....	f_{d-1}	y

nxd

Features = x = f1 f2 f3 f4..... f_{d-1} , y = y

x_q — | f1 | f2 | f3 | f4..... | f_{d-1} | yq | y_q = ?

The query point (Xq) will belong to that class for which P(Ci|Xq) is the highest.

$$y_q = \underset{i=1}{\overset{K}{\arg\max}} \{P(C_i|X_q)\}$$

Ci is list of all possible classes
Cardinality of y = K
K = No.of.Classes

Posteriori Probability

$$y_q = \underset{i=1}{\overset{K}{\arg\max}} \left\{ P(C_i|X_q) \right\}$$

From Bayes theorem

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

$$\Rightarrow P(C_i|X_q) = \frac{P(X_q|C_i) * P(C_i)}{P(X_q)}$$

Using the probabilistic approach, given Xq, compute the probability of Xq belonging to each of the classes, whichever is maximum assign that class to Xq.

Now,

$$P(C_1|X_q) = \frac{P(X_q|C_1) * P(C_1)}{P(X_q)} \propto P(X_q|C_1) * P(C_1)$$

$$P(C_2|X_q) = \frac{P(X_q|C_2) * P(C_2)}{P(X_q)} \propto P(X_q|C_2) * P(C_2)$$

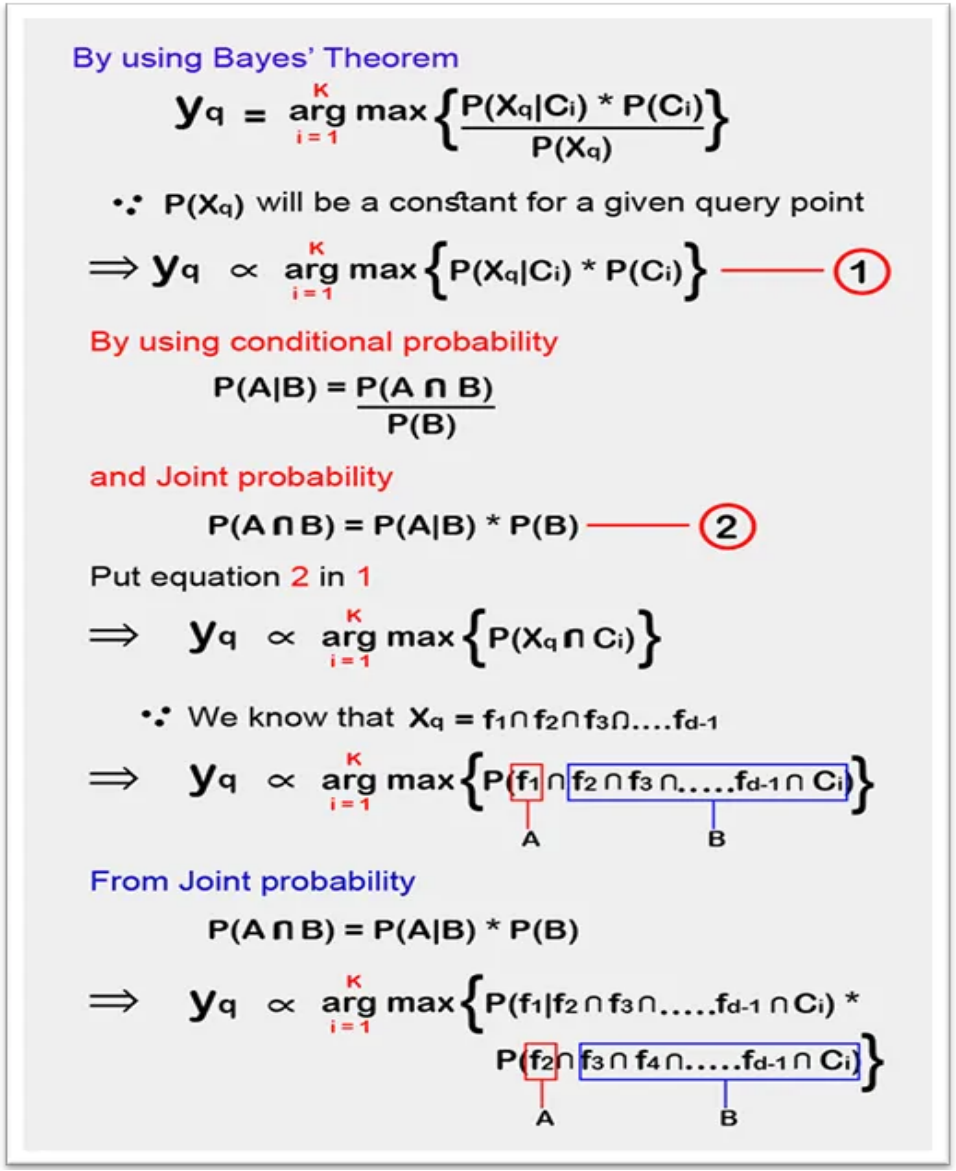
.
.
.
.

$$P(C_k|X_q) = \frac{P(X_q|C_k) * P(C_k)}{P(X_q)} \propto P(X_q|C_k) * P(C_k)$$

The denominator is common we can say that L.H.S will be proportional to the numerator on R.H.S.

Also, when comparing fractions with same denominator it is the numerator that decides which fraction is bigger. e.g. 1/5 < 3/5.

$$\Rightarrow \because y_q = \underset{i=1}{\overset{K}{\arg\max}} \left\{ P(C_i|X_q) \right\}$$



Doing this Joint probability process iteratively is known as the **CHAIN RULE OF CONDITIONAL PROBABILITY.**

$$\Rightarrow \quad y_q \propto \underset{i=1}{\overset{K}{\arg\max}} \Big\{ P(f_1|f_2 \cap f_3 \cap \ldots\ldots f_{d-1} \cap C_i) * P(f_2|f_3 \cap f_4 \cap \ldots\ldots f_{d-1} \cap C_i) * P(f_3|f_4 \cap f_5 \cap \ldots\ldots f_{d-1} \cap C_i) * \ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots * \ldots\ldots\ldots * P(f_{d-2}|f_{d-1} \cap C_i) * P(f_{d-1}|C_i) * P(C_i) \Big\}$$

To solve the above complicated equation, we take an assumption that each features are conditionally independent.

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

$$\Rightarrow P(A|B \cap C) = P(A|C) \Rightarrow \left[\begin{array}{c} \text{Assuming A and B are independent} \\ \text{but A and C are not independent} \end{array} \right]$$

$$\Rightarrow \quad y_q \propto \underset{i=1}{\overset{K}{\arg\max}} \Big\{ P(f_1|C_i) * P(f_2|C_i) * P(f_3|C_i) * \ldots\ldots\ldots\ldots * P(f_{d-1}|C_i) * P(C_i) \Big\}$$

So sum it up

$$y_q \propto \underset{i=1}{\overset{K}{\arg\max}} \left\{ P(C_i) * \left[\prod_{J=1}^{d-1} P(f_J|C_i) \right] \right\}$$

Humidity	Temperature	Rain
High	Hot	No
High	Hot	No
High	Cool	Yes
Moderate	Mild	Yes
Moderate	Cool	Yes
Low	Cool	Yes
Low	Mild	No
Moderate	Cool	No
Moderate	Mild	No
High	Cool	No
Low	Hot	Yes

Xq = Moderate and Hot → Yq → Rain = Yes/No?

Given N = 11

Input = [Humidity and Temperature], Output = [Rain]

Xq → Humidity = Moderate and Temperature = Hot

Y → $[\text{Rain} = \text{Yes/No}]$ is categorical → Classification Task

Applying chain rule and assume the features are conditionally independent.

P(Yes|Xq) → P(Rain = Yes) * P(Moderate|Yes) * P(Hot|Yes)

P(No|Xq) → P(Rain = No) * P(Moderate|No) * P(Hot|No)

Frequency Table for Humidity Vs Rain

	Yes	No	
High	1	3	4
Moderate	2	2	4
Low	2	1	3
	5	6	11

Frequency Table for Humidity Vs Rain

	Yes	No	
High	1	3	4
Moderate	2	2	4
Low	2	1	3
	5	6	11

$$P(Yes|Xq) \rightarrow P(Rain = Yes) * P(Moderate|Yes) * P(Hot|Yes)$$

$$\rightarrow \frac{5}{11} * \frac{2}{5} * \frac{1}{5} = \frac{10}{275} = \frac{2}{55}$$

$$P(No|Xq) \rightarrow P(Rain = No) * P(Moderate|No) * P(Hot|No)$$

$$\rightarrow \frac{6}{11} * \frac{1}{3} * \frac{1}{3} = \frac{6}{99} = \frac{2}{33}$$

Comparing these probabilities, we see that P(Yes|Xq) = 2/55 is smaller than P(No|Xq) = 2/33. Therefore, based on the probabilities, the class for the feature values Xq (Humidity = Moderate and Temperature = Hot) would be classified as “No” for Rain.

Naïve Bayes Classifier

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.

Difference Between Conditional Probability and Bayes Theorem

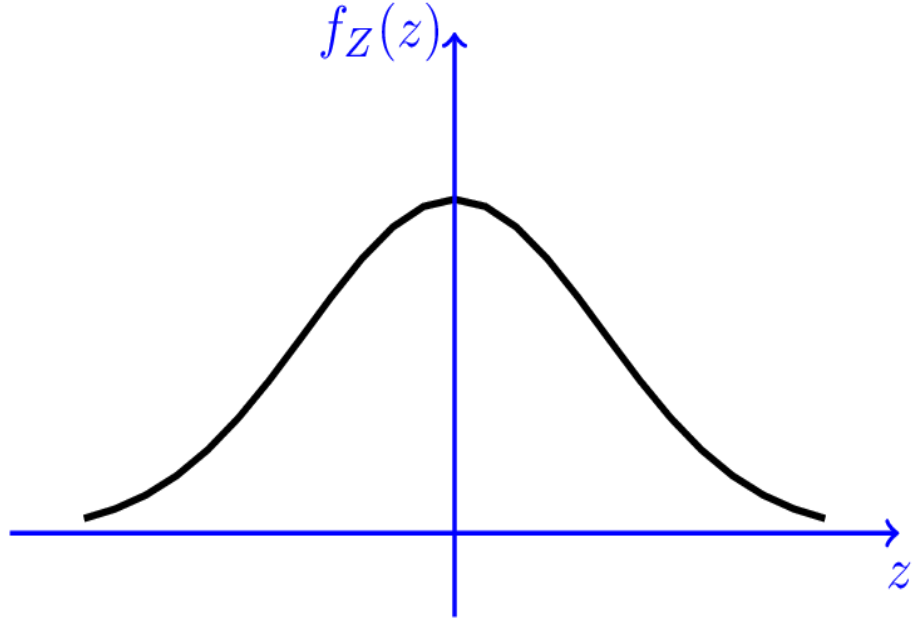
Bayes' Theorem	Conditional Probability
Bayes' Theorem is derived using the definition of conditional probability. It is used to find the reverse probability.	Conditional Probability is the probability of event A when event B has already occurred.
Formula: $P(A\|B) = [P(B\|A)P(A)] / P(B)$	**Formula:** $P(A\|B) = P(A \cap B) / P(B)$

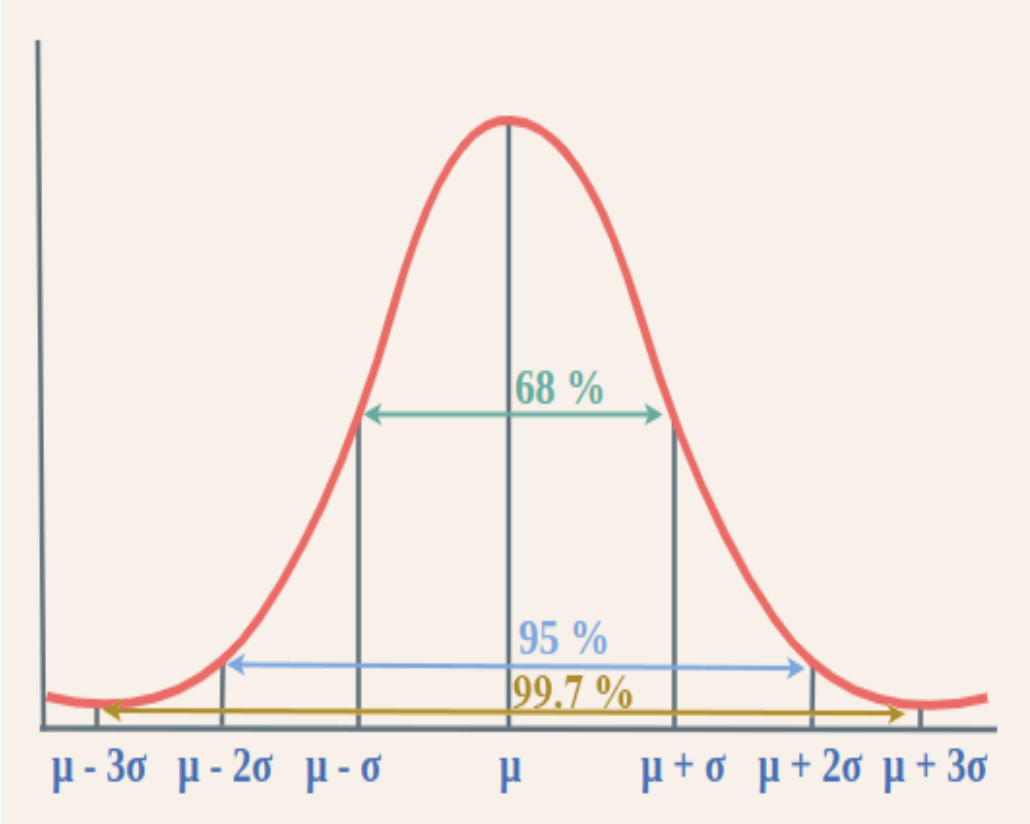
Gaussian Distribution

- The Gaussian distribution, also known as the normal distribution, is a fundamental concept in statistics and probability theory. It describes how the values of a variable are distributed.
- The formula for the Gaussian distribution is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- where:
- x is a value in the distribution.
- μ is the mean.
- σ is the standard deviation.
- Exp denotes the exponential function.



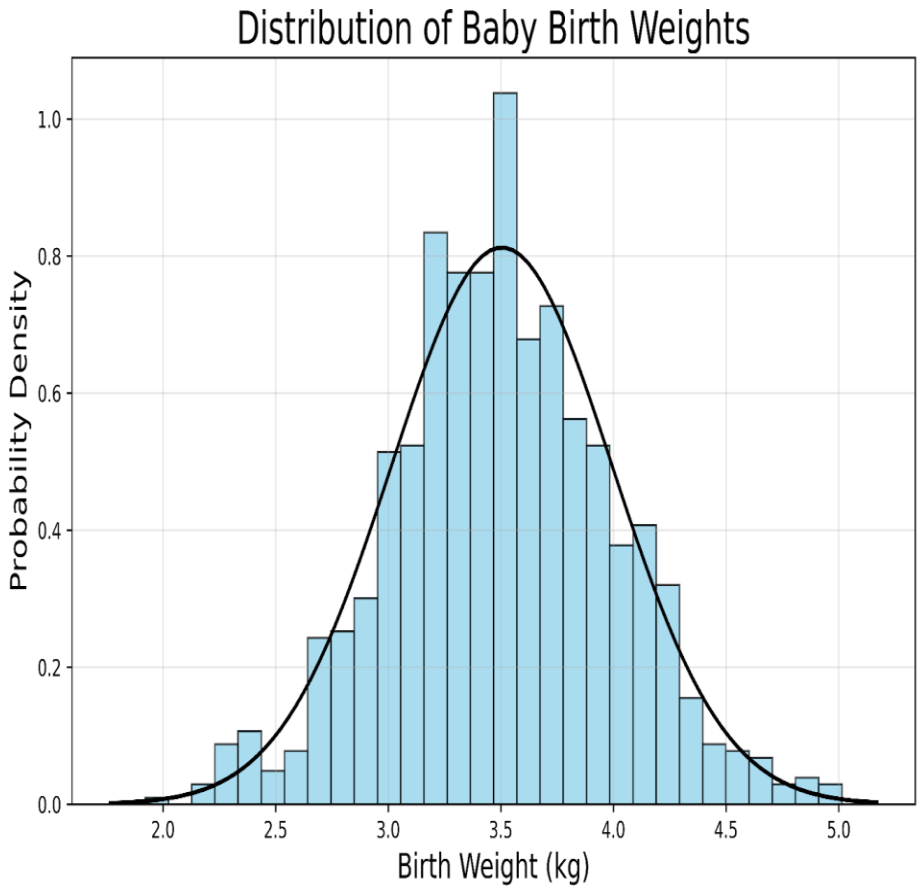


•The standard deviations are used to subdivide the area under the normal curve. Each subdivided section defines the percentage of data, which falls into the specific region of a graph.

•**Analysis :** A **smaller standard deviation** results in a **narrower and taller bell curve**, indicating that data points are clustered closely around the mean. Conversely, a **larger standard deviation** leads to a **wider and shorter bell curve,** suggesting that data points are **more spread out from the mean**.

•The **Empirical Rule**, also known as the **68-95-99.7 rule**, quantifies the proportion of data falling within certain intervals around the mean in a normal distribution. It provides a quick way to estimate the spread of data without performing detailed calculations.

Example



- Most babies' birth weights cluster around an average value (the peak of the curve).
- Fewer babies have birth weights that deviate significantly from this average.
- Very few babies have extreme birth weights (very high or very low).

GaussianDistribution

- **Calculate the probability density function of normal distribution using the following data. x = 2, μ = 3 and σ = 4.**
- Solution: *Variable (x) = 2*
- *Mean = 3*
- *Standard Deviation = 4*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Hence, *f(2, 3, 4) = 0.09666703*

Example

- **If the value of the random variable is 4, the mean is 4 and the standard deviation is 3, then find the probability density function of the Gaussian distribution.**

Given,

Variable (x) = 4

Mean = 4

Standard Deviation = 3

Simplifying,

$f(4, 4, 3) = 1/(3\sqrt{2\pi})e^{0}$

$f(4, 4, 3) = 0.13301$

- Let's classify whether a patient has **Disease A** based on their **Blood Pressure** and **Cholesterol Level**, assuming we have data on these features.

Training Data:

Blood Pressure (mm Hg)	Cholesterol (mg/dL)	Disease A
130	200	Yes
140	220	Yes
125	190	No
120	180	No
135	210	Yes

Step 1: Calculate Prior Probabilities

From the training data:

- $P(\text{Disease A} = \text{Yes}) = \frac{3}{5} = 0.6$
- $P(\text{Disease A} = \text{No}) = \frac{2}{5} = 0.4$

Step 2: Estimate Mean and Variance for Each Feature in Each Class

For **Blood Pressure**:

- Mean $\mu_{\text{Yes}} = \frac{130+140+135}{3} = 135$
- Variance $\sigma^2_{\text{Yes}} = \frac{(130-135)^2+(140-135)^2+(135-135)^2}{3} = 16.67$
- Mean $\mu_{\text{No}} = \frac{125+120}{2} = 122.5$
- Variance $\sigma^2_{\text{No}} = \frac{(125-122.5)^2+(120-122.5)^2}{2} = 12.5$

For **Cholesterol**:

- Mean $\mu_{\text{Yes}} = \frac{200+220+210}{3} = 210$
- Variance $\sigma^2_{\text{Yes}} = \frac{(200-210)^2+(220-210)^2+(210-210)^2}{3} = 66.67$
- Mean $\mu_{\text{No}} = \frac{190+180}{2} = 185$
- Variance $\sigma^2_{\text{No}} = \frac{(190-185)^2+(180-185)^2}{2} = 25$

Step 3: Predict for a New Patient

Suppose we have a new patient with:

- **Blood Pressure** = 128
- **Cholesterol** = 195

We calculate the likelihood for each class:

- **For Disease A = Yes:**

$$P(\text{Blood Pressure} = 128|\text{Yes}) = \frac{1}{\sqrt{2\pi \cdot 16.67}} \exp\left(-\frac{(128-135)^2}{2 \cdot 16.67}\right) = 0.070$$

$$P(\text{Cholesterol} = 195|\text{Yes}) = \frac{1}{\sqrt{2\pi \cdot 66.67}} \exp\left(-\frac{(195-210)^2}{2 \cdot 66.67}\right) = 0.046$$

$$P(\text{Yes}|X) \propto P(\text{Yes}) \cdot 0.070 \cdot 0.046 = 0.6 \cdot 0.070 \cdot 0.046 = 0.001932$$

- **For Disease A = No:**

$$P(\text{Blood Pressure} = 128|\text{No}) = \frac{1}{\sqrt{2\pi \cdot 12.5}} \exp\left(-\frac{(128-122.5)^2}{2 \cdot 12.5}\right) = 0.078$$

$$P(\text{Cholesterol} = 195|\text{No}) = \frac{1}{\sqrt{2\pi \cdot 25}} \exp\left(-\frac{(195-185)^2}{2 \cdot 25}\right) = 0.048$$

$$P(\text{No}|X) \propto P(\text{No}) \cdot 0.078 \cdot 0.048 = 0.4 \cdot 0.078 \cdot 0.048 = 0.0014976$$

Step 4: Compare Posterior Probabilities

Since $P(\text{Yes}|X) = 0.001932$ is greater than $P(\text{No}|X) = 0.0014976$, we classify the patient as **likely to have Disease A.**

Properties

- **Symmetry:** The normal distribution is symmetric around its mean. This means the left side of the distribution mirrors the right side.
- **Mean, Median, and Mode:** In a normal distribution, the mean, median, and mode are all equal and located at the center of the distribution.
- **Bell-shaped Curve:** The curve is bell-shaped, indicating that most of the observations cluster around the central peak, and the probabilities for values further away from the mean taper off equally in both directions.
- **Standard Deviation:** The spread of the distribution is determined by the standard deviation. About 68% of the data falls within one standard deviation of the mean, 95% within two standard deviations, and 99.7% within three standard deviations.

Multinomial Distribution

- The multinomial distribution is a generalization of the binomial distribution. While the binomial distribution deals with scenarios where there are only two possible outcomes (success or failure), the multinomial distribution handles situations where there are more than two.
- The multinomial distribution applies to experiments in which the following conditions are true:
- **Repeated:** The experiment consists of repeated trials, such as rolling a die five times instead of just once.
- **Independent:** Each trial must be independent of the others. For example, if you roll two dice, the outcome of one die does not impact the outcome of the other die.
- **Same probability:** The probability of each outcome must be the same across each instance of the experiment. For example, if a fair, six-sided die is used, then there must be a one-in-six chance of each number being given on each roll.
- **Specific outcome:** Each trial must produce a specific outcome, such as a number between two and 12 if rolling two six-sided dice.

Multinomial Naive Bayes Classifier

- This algorithm is particularly effective for text classification tasks such as spam detection or sentiment analysis.
- Multinomial Naive Bayes classifier assumes that the features (e.g., words in text data) follow a multinomial distribution given the class label. In this case, the probability of observing a particular word in a document is modeled using a multinomial distribution
- During training, the model estimates the probabilities of each feature (word) occurring in each class, and these probabilities are used to predict the class of new documents.
- For a given class c, the probability of a document with features x1,x2,x3…xn s given by:

$$P(c \mid x_1, x_2, \ldots, x_n) \propto P(c) \prod_{i=1}^{n} P(x_i \mid c)$$

where $P(x_i \mid c)$ follows a multinomial distribution.

- If the feature vector consists of counts (e.g., word frequencies), then the likelihood of observing a word w_i given class c_k is computed as:

$$P(w_i|c_k) = \frac{count(w_i, c_k) + \alpha}{\sum_{j=1}^{V} count(w_j, c_k) + \alpha V}$$

- count(w_i, c_k) is the number of times word w_i appears in documents of class c_k
- count(w_j, c_k) Total no words in class ck
- V is the size of the vocabulary (the number of unique words).
- α is a smoothing parameter (typically set to a small value, such as 1, to avoid zero probabilities for words that don't appear in the training set).

Training Dataset: simple_train = ['call you tonight', 'Call me a cab', 'please call me.. please']

Out[14]:

	cab	call	me	please	tonight	you	Label
0	0	1	0	0	1	1	Ham
1	1	1	1	0	0	0	Spam
2	0	1	1	2	0	0	Ham

Testing Dataset: " Please Don't Call me"

cab	call	me	please	tonight	you	Label

Bernoulli Naïve Bayes

- Bernoulli Distribution:Bernoulli distribution is used for discrete probability calculation. It either calculates success or failure. Here the random variable is either 1 or 0 whose chance of occurring is either denoted by p or (1-p) respectively.

$$f(x) = \begin{cases} p^x * (1-p)^{1-x} & \text{if x=0,1} \\ 0 \; otherwise \end{cases}$$

- If x=1 then the value of f(x) is p
- x=0 then the value of f(x) is 1-p.
- Here, p denotes the success of an event.

Bernoulli Naive Bayes

- Bernoulli Naive Bayes is a subcategory of the Naive Bayes Algorithm.
- The “Bernoulli” in its name comes from the assumption that each feature is binary-valued.
- It is used for the classification of binary features such as ‘Yes’ or ‘No’, ‘1’ or ‘0’, ‘True’ or ‘False’ etc.
- Used for spam detection, text classification, Sentiment Analysis, used to determine whether a certain word is present in a document or not.
- It is not suitable for multiclass problem.

- The decision rule of Bernoulli NB is given as follows

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

- *p(xi |y) is the conditional probability of xi occurring provided y has occurred.*
- *i is the event*
- *xi holds binary value either 0 or 1*

Step-by-Step with BNB

- **Data Preparation**: Begin with a set of binary data. Each row signifies a data sample while columns represent features.

DATA PREPARATION

Feature 1	Feature 2	Class
1	0	0
0	1	0
1	1	0
0	0	0
1	0	1
0	1	1
0	0	1
1	1	1
1	0	0
0	1	0

- Each row in the table represents a data sample
- each column represents a binary feature.
- For instance, "Feature 1" represents the presence (1) or absence (0)

Training Phase

- **Training Phase** — Class Priors: Compute the prior probabilities of each class based on your training data.
- Calculate the prior probabilities for each class based on the training data:
- Total samples = 10
- Class 0 samples = 6
- Class 1 samples = 4
- Thus:
- $P\ (Class\ 0) = 6/10 = 0.6$
- $P\ (Class\ 1) = 4/10 = 0.4$

- **Training Phase — Feature Priors:** Calculate conditional probabilities for each feature based on their presence (1) or absence (0).
- Next, compute the conditional probabilities for each feature based on its presence (1) or absence (0):
- **For Feature 1:**

$P(Feature\ 1=1|Class\ 0) = 3/6 = 0.5$

- $P(Feature\ 1=0|Class\ 0) = 3/6 = 0.5$
- **$(Feature\ 1=0|Class\ 1) = 2/4 = 0.5$**
- **$Feature\ 1=1|Class\ 1) = 2/4 = 0.5$**

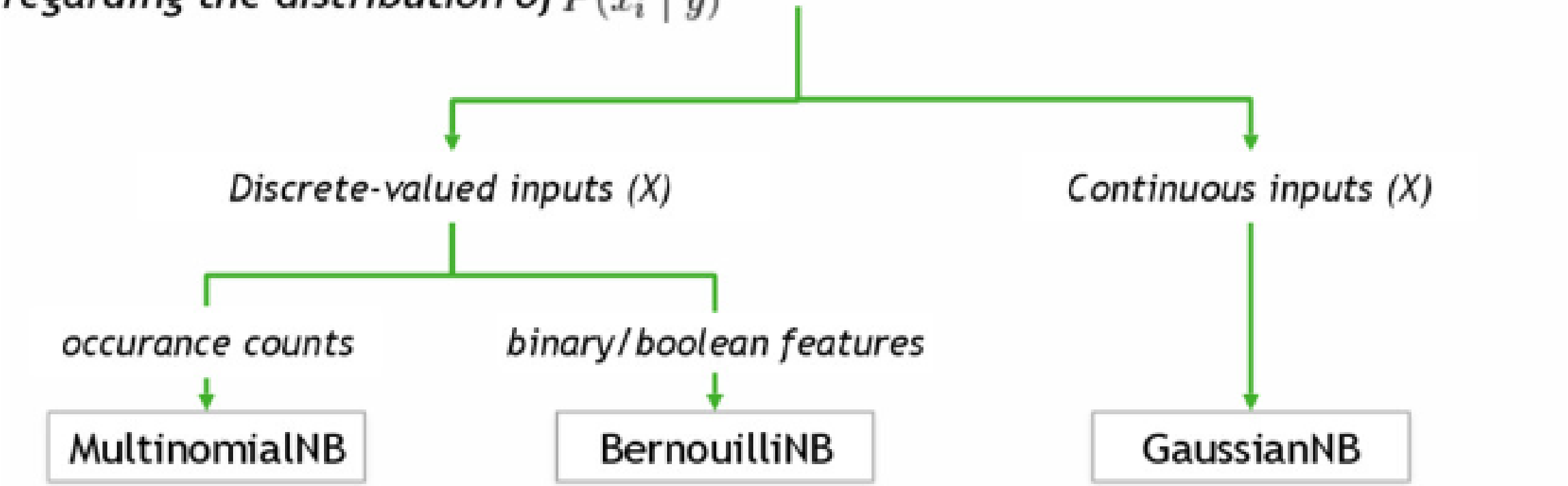
- **For Feature 2:** $P(Feature\ 2=1|Class\ 0) = 3/6 = 0.5$
  - $P(Feature\ 2=1|Class\ 1) = 2/4 = 0.5$
  - $P(Feature\ 2=0|Class\ 0) = 3/6 = 0.5$
  - $P(Feature\ 2=0|Class\ 1) = 2/4 = 0.5$

- 4. **Prediction:** Using the trained model, classify new samples.
- Calculate the likelihood of each class for the given features and compute the unnormalized posterior probability.
- **For Class 0:**
  P(Class 0 /Feature 1=1, Feature 2=0) = P(Feature 1=1 | Class 0) × P(Feature2=0 | Class 0)
  = 0.5 × 0.5
  = 0.25
- **Unnormalized Posterior**: P(Class 0) × Likelihood = 0.6 × 0.25 = 0.15

- **For Class 1:**
  P (Feature 1=1, Feature 2=0 | Class 1) = P (Feature 1=1 | Class 1) × P (Feature 2=0 | Class 1)
  = 0.5 × 0.5
  = 0.25
- **Unnormalized Posterior**: P (Class 1) × Likelihood = 0.4 × 0.25 = 0.10

Summary

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid y)$

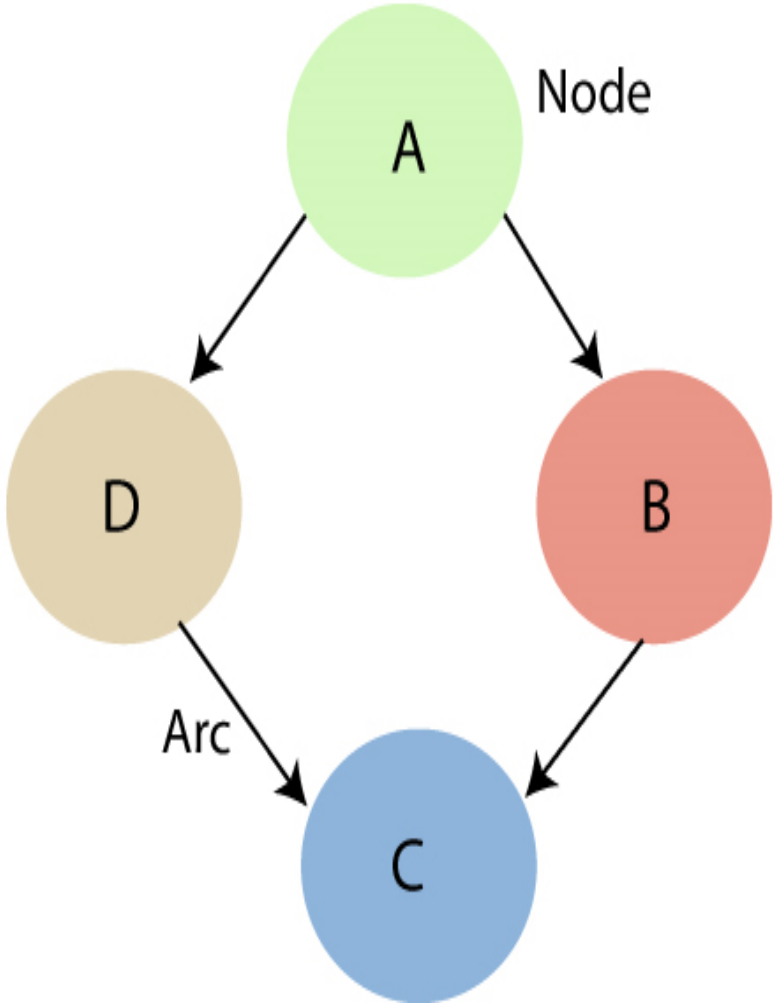


Bayesian Belief Network

- 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.

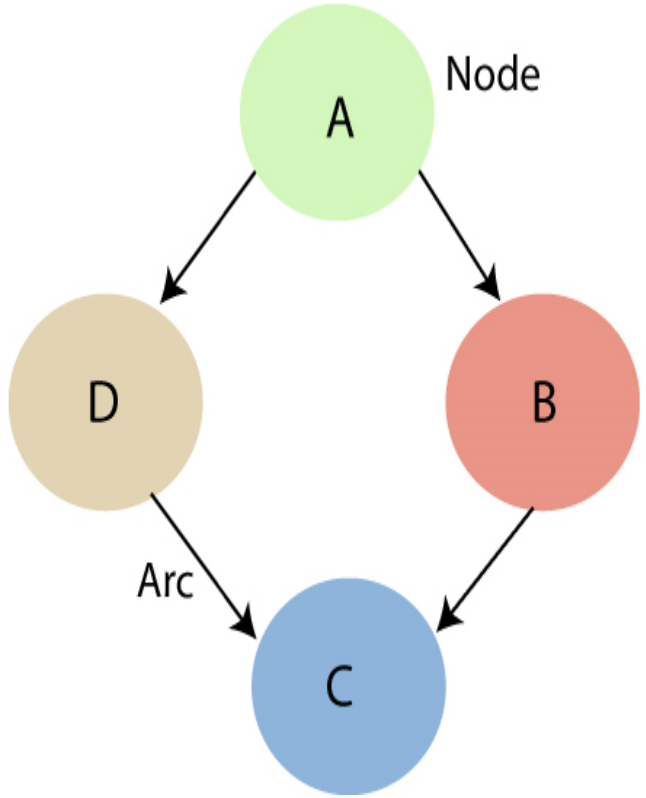
Bayesian Network Graph

- **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.

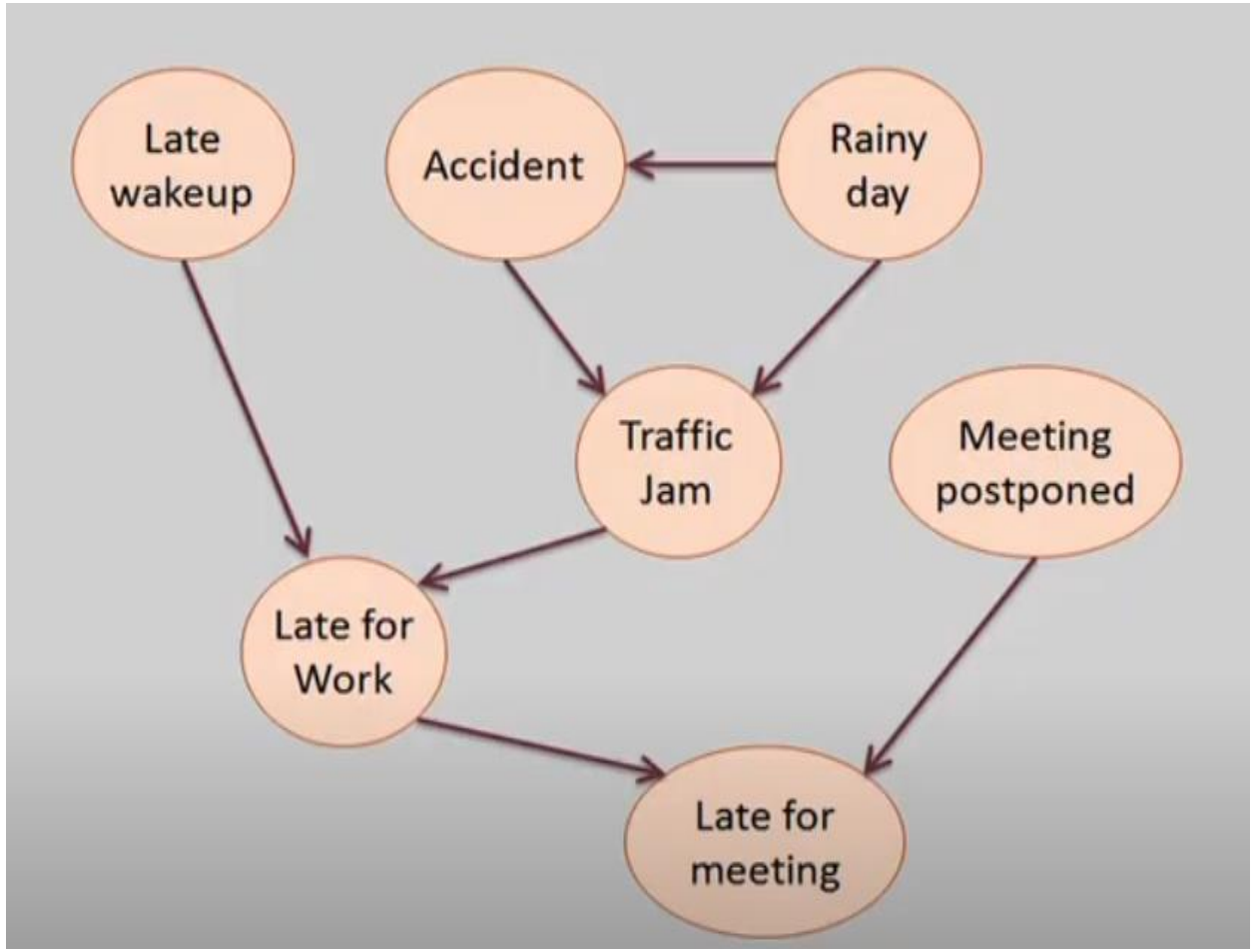


Bayesian Network Graph

- **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.**
- Each node in the Bayesian network has condition probability distribution **$P(X_i \mid Parent(X_i))$**, which determines the effect of the parent on that node.



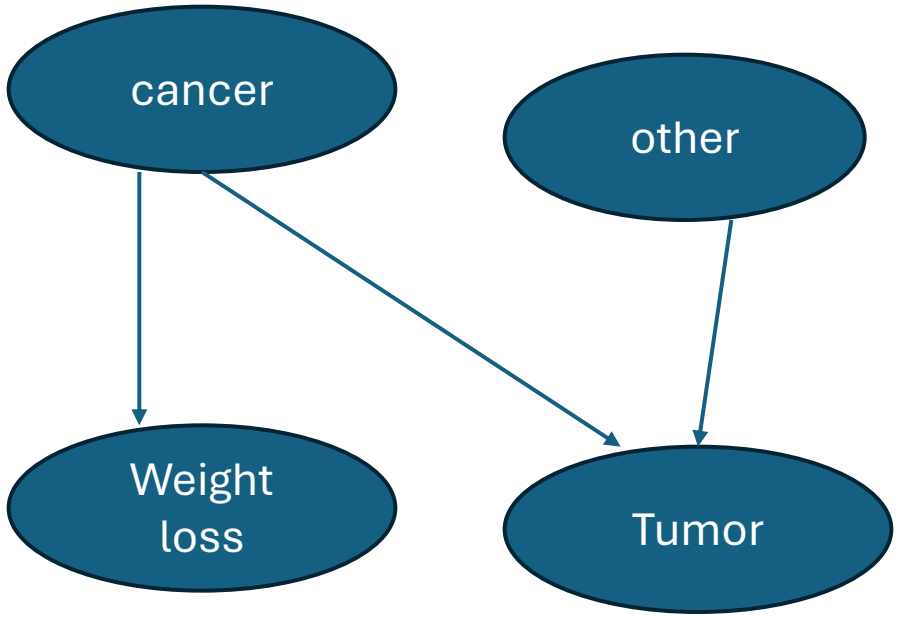
Example -BBN



- Each node is asserted to be conditionally independent of its non descendents, given its immediate parents.
- Eg: late wakeup and late for meeting are conditionally independent , given late for work.
- Inference: Compute posterior probability given evidences.

Applications

- Diagnosis:P(Cause| Symptom)=?
- Given the symptom, we can find the cause of the disease
- Prediction: P(Cause| Symptom)=?
- If a person has cancer, what is the probability of weight loss.



Bayesian Network Graph

- it consists of two parts:
- **Directed Acyclic Graph**
- **Table of conditional probabilities.**

Bayesian network is based on Joint probability distribution and conditional probability.

It is a compact representation of 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.

- **$P[x_1, x_2, x_3, ..., x_n]$**, it can be written as the following way in terms of the joint probability distribution.
- **$= P[x_1 | x_2, x_3, ..., x_n]P[x_2, x_3, ..., x_n]$**
- **$= P[x_1 | x_2, x_3, ..., x_n]P[x_2 | x_3, ..., x_n]....P[x_{n-1} | x_n]P[x_n]$.**
- In general for each variable Xi, we can write the equation as:

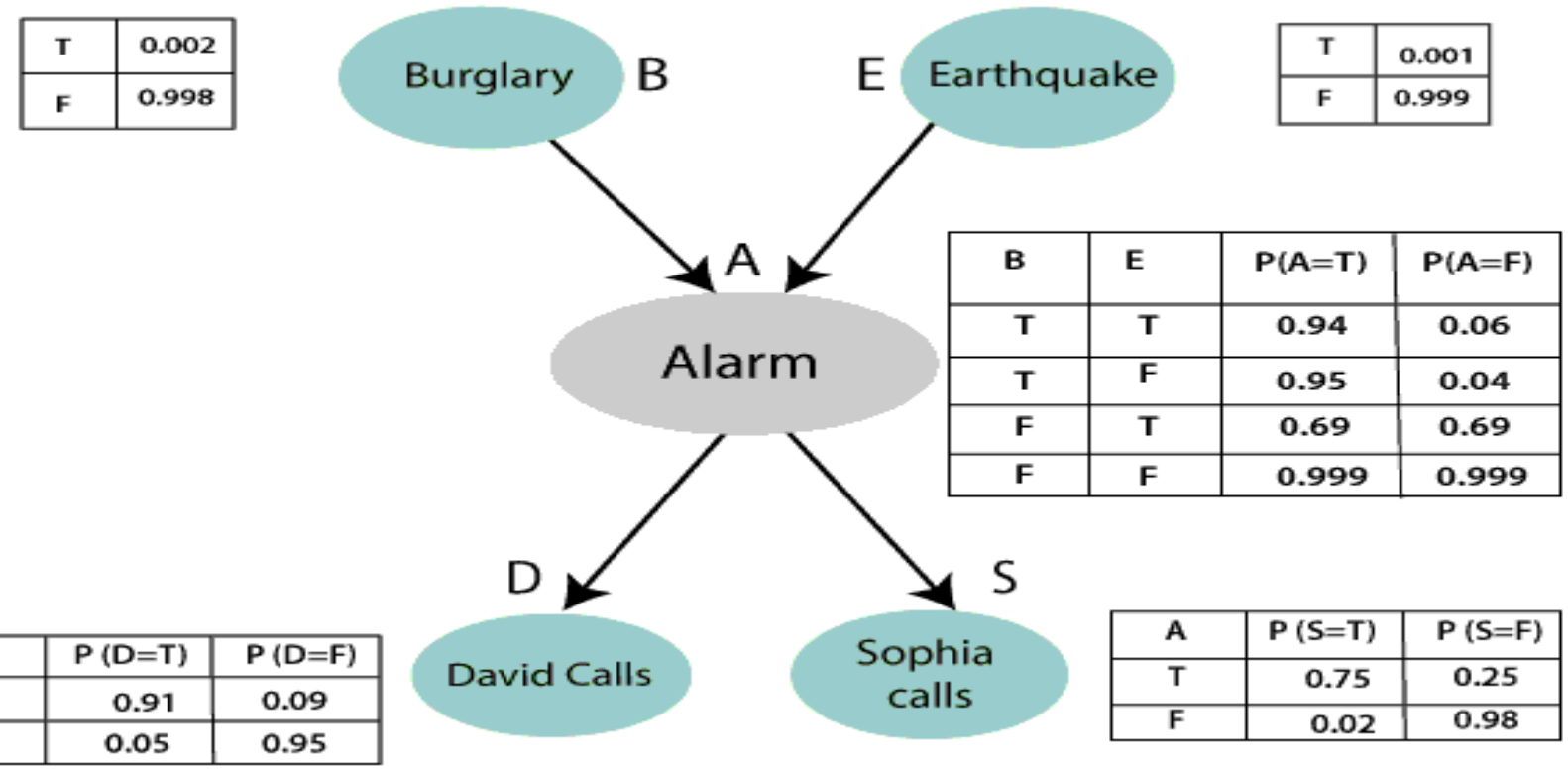
$$P(X_i|X_{i-1}, ..., X_1) = P(X_i | Parents(X_i))$$

- **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 2^K 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]**
-

Example



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:

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.

- **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

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

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.

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.**

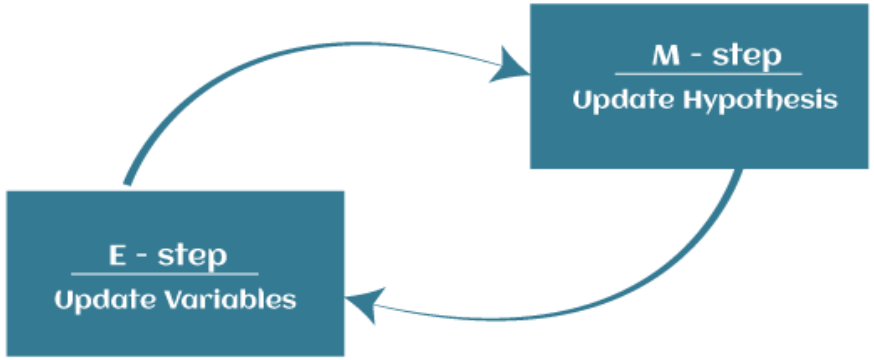
Hence, a Bayesian network can answer any query about the domain by using Joint distribution.

EM Algorithm

- The Expectation-Maximization (EM) algorithm is defined as the combination of various unsupervised machine learning algorithms, which is used to determine the **local maximum likelihood estimates (MLE)** or **maximum a posteriori estimates (MAP)** for unobservable variables in statistical models.
- It is a technique to find maximum likelihood estimation when the latent variables are present. It is also referred to as the latent variable model.
- A latent variable model consists of both observable and unobservable variables where observable can be predicted while unobserved are inferred from the observed variable. These unobservable variables are known as latent variables.

EM Algorithm

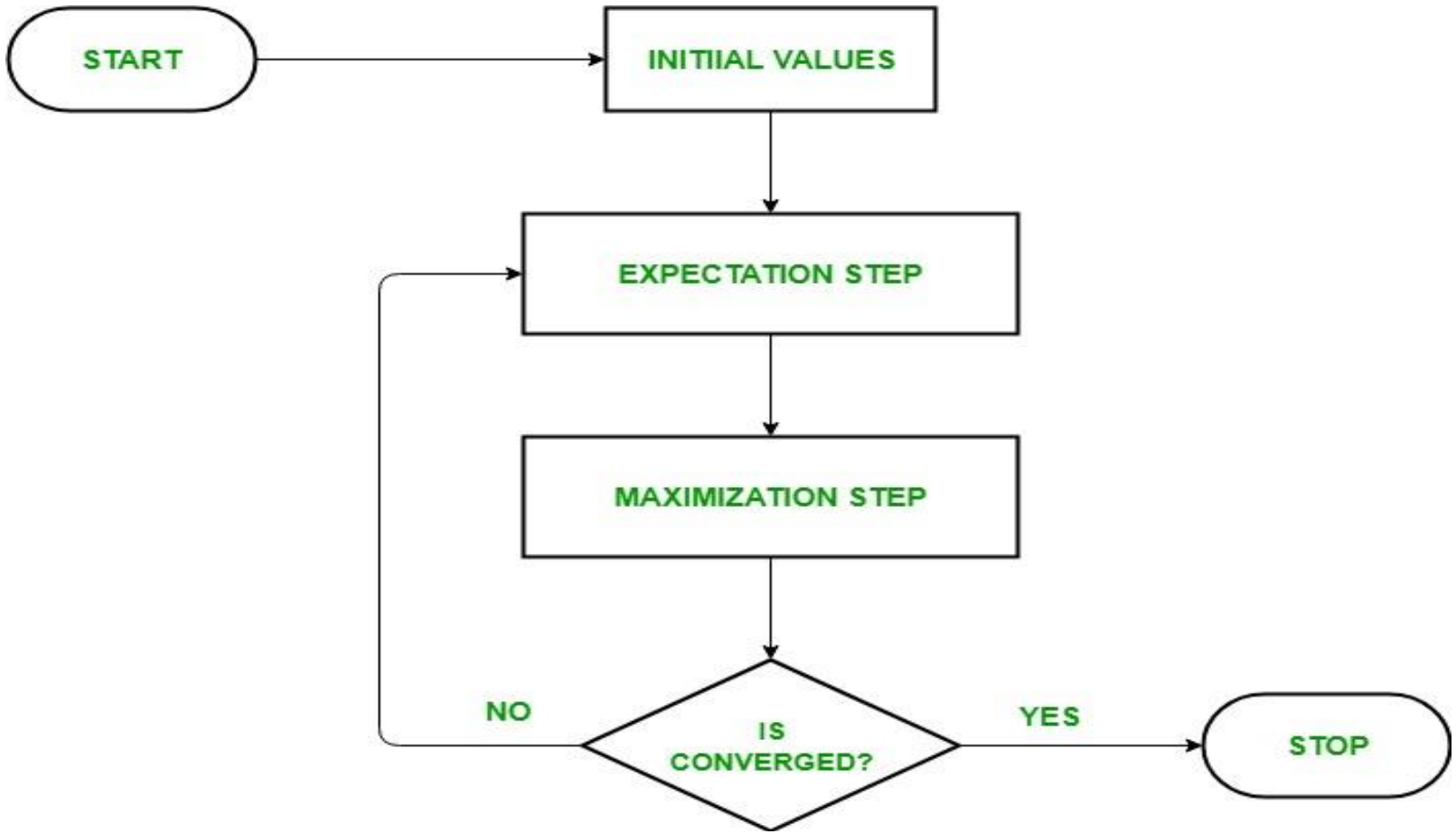
- The EM algorithm is the combination of various unsupervised ML algorithms, such as the **k-means clustering algorithm**. Being an iterative approach, it consists of two modes. In the first mode, we estimate the missing or latent variables. Hence it is referred to as the **Expectation/estimation step (E-step**). Further, the other mode is used to optimize the parameters of the models so that it can explain the data more clearly. The second mode is known as the **maximization-step or M-step.**



- **Expectation step (E - step):** It involves the estimation (guess) of all missing values in the dataset so that after completing this step, there should not be any missing value.
- **Maximization step (M - step):** This step involves the use of estimated data in the E-step and updating the parameters.
- **Repeat E-**step and M-step until the convergence of the values occurs.

Convergence in the EM algorithm

- Convergence refers to the condition when the EM algorithm has reached a stable solution. It is typically determined by checking if the change in the log-likelihood or the parameter estimates falls below a predefined threshold.
- The essence of the Expectation-Maximization algorithm is to use the available observed data of the dataset to estimate the missing data and then use that data to update the values of the parameters.


START
INITIIAL VALUES
EXPECTATION STEP
MAXIMIZATION STEP
IS CONVERGED?
NO
YES
STOP

- **Initialization: A** set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model.

2. **E-Step (Expectation Step):** we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables.
    - Compute the posterior probability or responsibility of each latent variable given the observed data and current parameter estimates.
    - Estimate the missing or incomplete data values using the current parameter estimates.
    - Compute the log-likelihood of the observed data based on the current parameter estimates and estimated missing data.
-

3. **M-step (Maximization Step):** In this step, we use the complete data generated in the preceding “Expectation” – step in order to update the values of the parameters. It is basically used to update the hypothesis.
    3. Update the parameters of the model by maximizing the expected complete data log-likelihood obtained from the E-step.
    4. This typically involves solving optimization problems to find the parameter values that maximize the log-likelihood.
    5. The specific optimization technique used depends on the nature of the problem and the model being used.

•

4. **Convergence**: In this step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat *step-2* and *step-3* i.e. “Expectation” – step and “Maximization” – step until the convergence occurs.
    4. Check for convergence by comparing the change in log-likelihood or the parameter values between iterations.
    5. If the change is below a predefined threshold, stop and consider the algorithm converged.
    6. Otherwise, go back to the E-step and repeat the process until convergence is achieved.

Code

```
import numpy as np
import seaborn as sns
from scipy.stats import norm
from scipy.stats import gaussian_kde
import matplotlib.pyplot as plt
# Generate a dataset with two Gaussian components
mu1, sigma1 = 2, 1
mu2, sigma2 = -1, 0.8
X1 = np.random.normal(mu1, sigma1, size=200)
X2 = np.random.normal(mu2, sigma2, size=600)
X = np.concatenate([X1, X2])
```

```
# Plot the density estimation using seaborn
sns.kdeplot(X)
plt.xlabel('X')
plt.ylabel('Density')
plt.title('Density Estimation of X')
plt.show()
```

- **# Initialize parameters**

```
mu1_hat, sigma1_hat = np.mean(X1), np.std(X1)
mu2_hat, sigma2_hat = np.mean(X2), np.std(X2)
pi1_hat, pi2_hat = len(X1) / len(X), len(X2) / len(X)
```

```
# Perform EM algorithm for 20 epochs
num_epochs = 20
log_likelihoods = []
for epoch in range(num_epochs):
    # E-step: Compute responsibilities
    gamma1 = pi1_hat * norm.pdf(X, mu1_hat, sigma1_hat)
    gamma2 = pi2_hat * norm.pdf(X, mu2_hat, sigma2_hat)
    total = gamma1 + gamma2
    gamma1 /= total
    gamma2 /= total
```

```
 # M-step: Update parameters
    mu1_hat = np.sum(gamma1 * X) / np.sum(gamma1)
    mu2_hat = np.sum(gamma2 * X) / np.sum(gamma2)
    sigma1_hat = np.sqrt(np.sum(gamma1 * (X - mu1_hat)**2) / np.sum(gamma1))
    sigma2_hat = np.sqrt(np.sum(gamma2 * (X - mu2_hat)**2) / np.sum(gamma2))
    pi1_hat = np.mean(gamma1)
    pi2_hat = np.mean(gamma2)
       # Compute log-likelihood
    log_likelihood = np.sum(np.log(pi1_hat * norm.pdf(X, mu1_hat, sigma1_hat)
                                   + pi2_hat * norm.pdf(X, mu2_hat, sigma2_hat)))
    log_likelihoods.append(log_likelihood)
```

```
# Plot log-likelihood values over epochs
plt.plot(range(1, num_epochs+1), log_likelihoods)
plt.xlabel('Epoch')
plt.ylabel('Log-Likelihood')
plt.title('Log-Likelihood vs.Epoch')
plt.show()
```