

Naive Bayes in Classification

Naive Bayes is a **supervised machine learning classification technique** based on **Bayes' Theorem**.

It is used to classify data into different classes by calculating the **probability of each class** and selecting the class with the highest probability.

The word “**Naive**” means that the algorithm assumes that the features are **independent of each other**.

Naive Bayes predicts the class of a new data item by using the probability of its features and selecting the most probable class.

Example: Email Classification

Suppose an email contains the words:

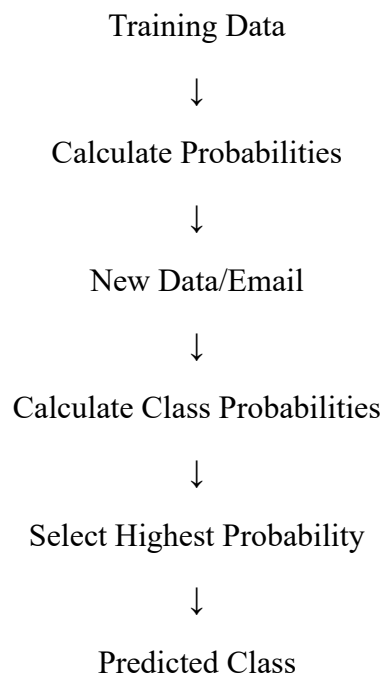
“Win free money”

We want to classify it as:

- **Spam**
- **Not Spam**

Naive Bayes checks the probability of these words occurring in spam and non-spam emails. If: **Probability of Spam > Probability of Not Spam** then the email is classified as: **Spam**

Working



Bayes' Theorem

Bayes' Theorem



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

Where:

- $P(A|B)$ = Probability of A when B is known
- $P(B|A)$ = Probability of B when A is known
- $P(A)$ = Probability of A
- $P(B)$ = Probability of B

Applications of Naive Bayes

1. Spam email detection
2. Text classification
3. Sentiment analysis
4. Document classification
5. News classification
6. Medical diagnosis

Advantages

- Simple and easy to implement.
- Fast in training and prediction.
- Works well with large datasets.
- Particularly useful for text classification.
- Requires relatively less training data compared with some other methods.

Naive Bayes is a supervised classification algorithm based on Bayes' Theorem that assumes the features are independent and classifies a data item according to the class having the highest probability.

Naive Bayes – Numerical Example

Suppose we want to classify an email as **Spam** or **Not Spam**.

We have the following training data:

Email	Free	Class
1	Yes	Spam
2	Yes	Spam
3	No	Not Spam
4	No	Not Spam
5	Yes	Spam

Now, classify a new email:

Free = Yes

Step 1: Calculate Prior Probability

Total emails = 5

Spam emails = 3

Not Spam emails = 2

$P(\text{Spam}) = 3 / 5 = 0.6$

$P(\text{Not Spam}) = 2 / 5 = 0.4$

Step 2: Calculate Conditional Probability

Among 3 Spam emails, all 3 contain “Free = Yes”.

$P(\text{Free}|\text{Spam}) = 3 / 3 = 1$

Among 2 Not Spam emails, none contains “Free = Yes”.

$P(\text{Free} | \text{Not Spam}) = 0 / 2 = 0$ **Step**

3: Apply Bayes' Formula

Bayes' theorem:

$$P(\text{Class} | \text{Feature}) = (P(\text{Feature} | \text{Class}) * P(\text{Class})) / P(\text{Feature})$$

For Spam:

$$P(\text{Spam} | \text{Free}) = (P(\text{Free} | \text{Spam}) * P(\text{Spam})) / P(\text{Free})$$

$$= (1 * 0.6) / P(\text{Free})$$

For Not Spam:

$$P(\text{Not Spam} | \text{Free}) = (0 * 0.4) / P(\text{Free}) = 0$$

Since the probability of **Spam** is higher, the new email is classified as:

Spam

Rule: - Calculate probability for each class → Compare probabilities → Choose the class with the highest probability.

Clustering Technique – K-Means

K-Means is an **unsupervised machine learning clustering technique** used to divide a dataset into **K groups (clusters)** based on the similarity of data points.

In K-Means, each data point is assigned to the cluster whose **centroid (center)** is nearest to it.

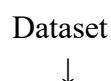
Important Terms

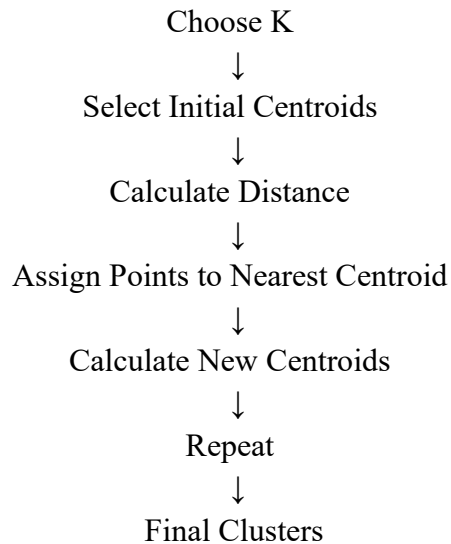
- **K** → Number of clusters required.
- **Centroid** → Center point of a cluster.
- **Cluster** → Group of similar data points.
- **Distance** → Used to determine which centroid is closest to a data point.

Working of K-Means

1. Choose the number of clusters **K**.
2. Select **K initial centroids**.
3. Calculate the distance of each data point from each centroid.
4. Assign each data point to the **nearest centroid**.
5. Recalculate the centroid of each cluster.
6. Repeat the assignment and centroid calculation.
7. Stop when the centroids become stable or there is no significant change.

Flow





Distance Formula

Usually, **Euclidean Distance** is used:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

where:

- d = Distance between two points
- x_1, y_1 = Coordinates of first point
- x_2, y_2 = Coordinates of second point

Example

Suppose we have students based on their **study hours and marks**:

Student	Study Hours	Marks
A	1	30
B	2	35
C	3	40
D	8	80
E	9	85
F	10	90

Let:

$K=2$

K-Means can form two clusters:

Cluster 1: A, B, C

Students with lower study hours and marks.

Cluster 2: D, E, F

Students with higher study hours and marks.

Thus, K-Means automatically groups similar students into different clusters.

Applications

1. Customer segmentation
2. Image segmentation
3. Document clustering
4. Market analysis
5. Student performance grouping
6. Recommendation systems
7. Pattern recognition
8. Anomaly detection

Advantages

- Simple and easy to implement.
- Fast for large datasets.
- Easy to understand.
- Works well when clusters are clearly separated.
- Computationally efficient.

Disadvantages

- Value of **K** must be chosen in advance.
- Sensitive to initial centroid selection.
- Sensitive to outliers.
- Does not work well with irregularly shaped clusters.
- Different initial centroids may produce different results.

K-Means is an unsupervised clustering algorithm that divides data into K clusters by assigning each data point to the nearest centroid and repeatedly updating the centroids until the clusters become stable.

2-Dimensional K-Means Numerical Example

Question

Consider the following 2-dimensional data points:

Point	X	Y
A	1	1
B	2	2
C	3	3
D	8	8
E	9	9
F	10	10

Take:

K=2

Let the initial centroids be:

$$C_1 = (1, 1)$$

$$C_2 = (8, 8)$$



Step 1: Calculate Euclidean Distance

For a 2-dimensional point:

$$d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

where (x, y) is the data point and (x_c, y_c) is the centroid.

Step 2: Calculate Distance of Each Point

Point A = (1,1)

Distance from $C_1 = (1, 1)$:

$$\begin{aligned}d(A, C_1) &= \sqrt{(1-1)^2 + (1-1)^2} \\&= 0\end{aligned}$$

Distance from $C_2 = (8, 8)$:

$$\begin{aligned}d(A, C_2) &= \sqrt{(1-8)^2 + (1-8)^2} \\&= \sqrt{49 + 49} = \sqrt{98} = 9.90\end{aligned}$$

Therefore, A → **Cluster 1**

Point B = (2,2)

$$\begin{aligned}d(B, C_1) &= \sqrt{(2-1)^2 + (2-1)^2} \\&= \sqrt{2} = 1.41\end{aligned}$$

$$\begin{aligned}d(B, C_2) &= \sqrt{(2-8)^2 + (2-8)^2} \\&= \sqrt{72} = 8.49\end{aligned}$$

Therefore, B → **Cluster 1**

Point C = (3,3)

$$\begin{aligned}d(C, C_1) &= \sqrt{(3-1)^2 + (3-1)^2} \\&= \sqrt{8} = 2.83\end{aligned}$$

$$\begin{aligned}d(C, C_2) &= \sqrt{(3-8)^2 + (3-8)^2} \\&= \sqrt{50} = 7.07\end{aligned}$$

Therefore, C → **Cluster 1**

Point D = (8,8)

$$\begin{aligned}d(D, C_1) &= \sqrt{(8-1)^2 + (8-1)^2} \\&= \sqrt{98} = 9.90\end{aligned}$$

$$d(D, C_2) = 0$$

Therefore, D → **Cluster 2**

Point E = (9,9)

$$\begin{aligned}d(E, C_1) &= \sqrt{(9-1)^2 + (9-1)^2} \\&= \sqrt{128} = 11.31\end{aligned}$$

$$\begin{aligned}d(E, C_2) &= \sqrt{(9-8)^2 + (9-8)^2} \\&= \sqrt{2} = 1.41\end{aligned}$$

Therefore, E → **Cluster 2**



Point F = (10,10)

$$d(F, C_1) = \sqrt{(10 - 1)^2 + (10 - 1)^2}$$

$$= \sqrt{162} = 12.73$$

$$d(F, C_2) = \sqrt{(10 - 8)^2 + (10 - 8)^2}$$

$$= \sqrt{8} = 2.83$$

Therefore, F → Cluster 2

Step 3: Form Clusters

Cluster	Points
Cluster 1	A, B, C
Cluster 2	D, E, F

Step 4: Calculate New Centroids

New Centroid of Cluster 1

Points:

$$(1, 1), (2, 2), (3, 3)$$

Formula:

$$C_1 = \left(\frac{\sum X}{n}, \frac{\sum Y}{n} \right)$$

$$C_1 = \left(\frac{1 + 2 + 3}{3}, \frac{1 + 2 + 3}{3} \right)$$

$$C_1 = (2, 2)$$

New Centroid of Cluster 2

Points:

$$(8, 8), (9, 9), (10, 10)$$

$$C_2 = \left(\frac{8 + 9 + 10}{3}, \frac{8 + 9 + 10}{3} \right)$$

$$C_2 = (9, 9)$$

Final Result

$$C_1 = (2, 2)$$

$$C_2 = (9, 9)$$

Therefore:

Cluster 1 $\rightarrow$ A (1,1), B (2,2), C (3,3)

Cluster 2 $\rightarrow$ D (8,8), E (9,9), F (10,10)

What Happens After Calculating New Centroids?

$$C_1 = (2, 2)$$

$$C_2 = (9, 9)$$

Now we repeat the same process using these new centroids.

1. Again Calculate Distances

For every point, calculate its distance from the new C_1 and new C_2 .

For example, for point C = (3,3):

Distance from $C_1 = (2, 2)$:

$$\begin{aligned} d(C, C_1) &= \sqrt{(3-2)^2 + (3-2)^2} \\ &= \sqrt{2} = 1.41 \end{aligned}$$

Distance from $C_2 = (9, 9)$:

$$\begin{aligned} d(C, C_2) &= \sqrt{(3-9)^2 + (3-9)^2} \\ &= \sqrt{72} = 8.49 \end{aligned}$$

So C remains in **Cluster 1**.

2. Do This for All Points

Point	Distance from C_1 (2,2)	Distance from C_2 (9,9)	Cluster
A (1,1)	1.41	11.31	C_1
B (2,2)	0	9.90	C_1
C (3,3)	1.41	8.49	C_1
D (8,8)	8.49	1.41	C_2
E (9,9)	9.90	0	C_2
F (10,10)	11.31	1.41	C_2

So, the clusters are **still**:

Cluster1 = {A, B, C}

Cluster2 = {D, E, F}

3. Calculate Centroids Again

Since the clusters have not changed:

$$C_1 = \left(\frac{1+2+3}{3}, \frac{1+2+3}{3} \right)$$
$$C_1 = (2, 2)$$

And:

$$C_2 = \left(\frac{8+9+10}{3}, \frac{8+9+10}{3} \right)$$
$$C_2 = (9, 9)$$

The centroids are **exactly the same as before**.

4. Stop the Algorithm

Now we compare:

Old centroids:

$$C_1 = (2, 2), \quad C_2 = (9, 9)$$

New centroids:

$$C_1 = (2, 2), \quad C_2 = (9, 9)$$

There is **no change**.

Therefore, K-Means stops.

Final Clusters

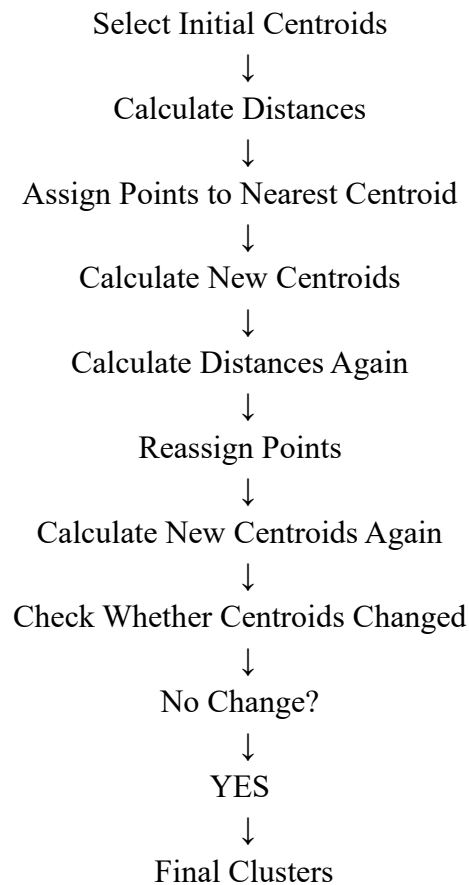
$$\boxed{\text{Cluster 1} = \{A, B, C\}}$$

$$\boxed{\text{Cluster 2} = \{D, E, F\}}$$

Complete K-Means Process

Choose K





Classification Metrics in Machine Learning

Classification metrics are used to evaluate the performance of a machine learning model and measure how accurately it predicts the correct class.

1. Accuracy

Accuracy is the ratio of correctly predicted observations to the total number of observations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP (True Positive): Actual positive and predicted positive.
- TN (True Negative): Actual negative and predicted negative.
- FP (False Positive): Actual negative but predicted positive.
- FN (False Negative): Actual positive but predicted negative. **Given Values Let:**
- True Positive (TP) = 40

- True Negative (TN) = 30
- False Positive (FP) = 20
- False Negative (FN) = 10

For MAE, let actual values be 10, 20, 30 and predicted values be 12, 18, 25.

$$Accuracy = \frac{40 + 30}{40 + 30 + 20 + 10} \times 100$$

$$= \frac{70}{100} \times 100$$

$$Accuracy = 70\%$$

Use: Measures the overall correctness of a classification model.

2. Precision

Precision measures how many of the observations predicted as positive are actually positive.

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{40}{40 + 20} \times 100$$

$$= \frac{40}{60} \times 100$$

$$Precision = 66.67\%$$

Use: Precision is important when false positive predictions must be minimized.

3. Recall

Recall measures how many actual positive observations are correctly identified by the model.

TP

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{40}{40 + 10} \times 100$$

$$= \frac{40}{50} \times 100$$

$$Recall = 80\%$$

Use: Recall is important when missing positive cases is costly, such as detecting diseases.

4. F1 Score

F1 Score is the harmonic mean of precision and recall. It provides a balance between precision and recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1 = 2 \times \frac{66.67 \times 60}{66.67 + 60}$$

$$F1 = \frac{10667.2}{146.67}$$

$$F1 \approx 72.73\%$$

Use: F1 Score is useful when both precision and recall are important, especially for imbalanced datasets.

5. MAE (Mean Absolute Error)

Mean Absolute Error is the average of the absolute differences between actual values and predicted values.

MAE is primarily used for regression problems, rather than classification problems.

$$MAE_{\hat{y}_i} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Where:

- n = Total number of observations
- y_i = Actual value
- $\hat{y}_i$ = Predicted value
- $| |$ = Absolute value **Example:**

Actual values: 10, 20, 30

Predicted values: 12, 18, 25

Here, $n = 3$

$$MAE = \frac{|10 - 12| + |20 - 18| + |30 - 25|}{3}$$

$$MAE = \frac{2 + 2 + 5}{3} = \frac{9}{3}$$

$$MAE = 3$$

Use: Measures the average magnitude of prediction errors. A lower MAE indicates better predictions.

Difference Between Classification Metrics

Metric	Measures	Best Value
Accuracy	Overall correct predictions	100%
Precision	Correct positive predictions among predicted positives	100%
Recall	Correctly identified actual positives	100%
F1 Score	Balance between precision and recall	100%
MAE	Average absolute prediction error	0

Accuracy measures overall correctness, precision measures the reliability of positive predictions, recall measures the identification of actual positive cases, F1 Score balances precision and recall, and MAE measures average prediction error.

Neural Networks

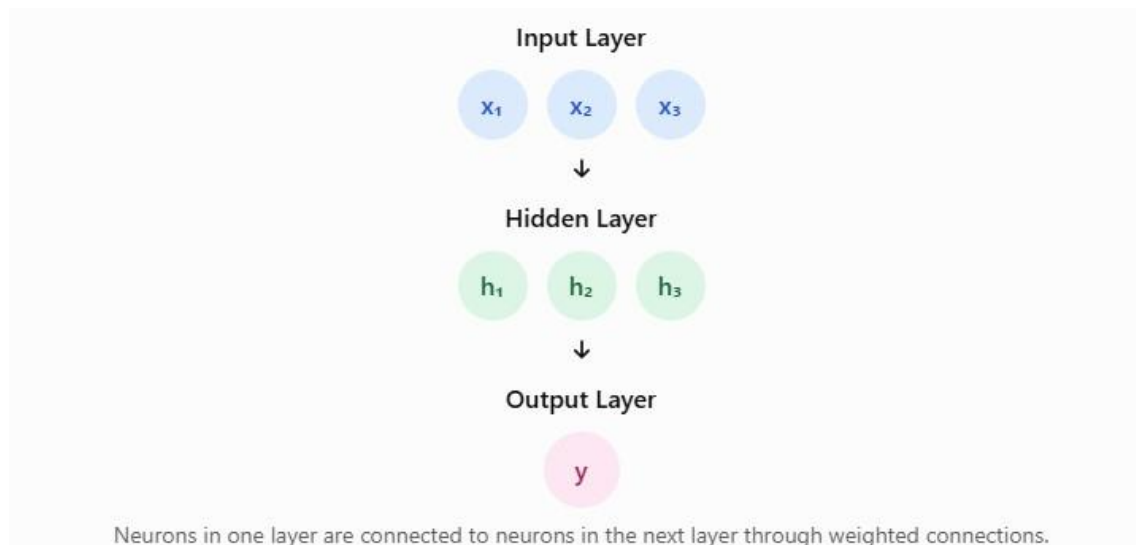
(Don't write bracket line in exam, it's just for understanding)

1. Definition

A Neural Network is a machine learning model inspired by the working of the human brain. It consists of interconnected neurons arranged in layers that process data and learn patterns to make predictions or decisions.

(Example: A neural network can learn to identify whether an image contains a cat or a dog by studying many cat and dog images.)

2. Simple Diagram of a Neural Network



Neurons in one layer are connected to neurons in the next layer through weighted connections.

(Example: If we give a picture of a cat to the neural network, the input layer receives the picture, the hidden layer identifies patterns such as ears and eyes, and the output layer predicts “Cat.”)

3. Neurons

A neuron is the basic processing unit of a neural network. It receives inputs, multiplies them by weights, adds a bias, and applies an activation function to produce an output.

(For **example**, when identifying a cat, one neuron may respond to the presence of pointed ears, while another may respond to the shape of the eyes.)

Functions of a neuron:

1. Receives input values.
2. Multiplies inputs by their respective weights.
3. Adds a bias.

4. Applies an activation function.
5. Passes the output to the next layer.

4. Layers in a Neural Network

A neural network consists of three main types of layers.

1. Input Layer: Receives input data such as images, numbers, or text.

(**Example:** If we want to identify cats and dogs, the input layer receives the image data.)

2. Hidden Layer: Processes the input data and learns patterns using weights, biases, and activation functions. A network can have one or more hidden layers.

(**Example:** While identifying an animal, hidden layers may learn patterns such as ears, eyes, fur, and face shape.)

3. Output Layer: Produces the final result, such as a predicted value or a class label.

(**Example:** After examining an image, the output layer may predict “Cat” or “Dog.”)

5. Weights

Weights are numerical values assigned to the connections between neurons. They determine the importance of each input in producing the output.

(**Example:** Suppose a neural network predicts whether a student will pass an examination based on study hours and attendance.)

- A larger absolute weight generally indicates a stronger influence on the neuron's output.
- Positive weights increase the weighted sum, while negative weights decrease it.
- During training, weights are adjusted to reduce prediction errors.

Example:

If inputs are $x_1 = 2$, $x_2 = 3$, and their weights are $w_1 = 0.5$, $w_2 = 0.2$, then:

$$\begin{aligned}\text{Weighted Sum} &= x_1w_1 + x_2w_2 \\ &= (2 \times 0.5) + (3 \times 0.2) \\ &= 1 + 0.6 = 1.6\end{aligned}$$

6. Working of a Neural Network

1. Input data is provided to the input layer.
2. Inputs are multiplied by their weights.
3. Each neuron calculates a weighted sum and adds a bias.
4. An activation function processes the result.
5. Hidden layers extract useful patterns from the data.
6. The output layer produces the prediction.
7. During training, weights are adjusted to improve prediction accuracy.

Neural networks use interconnected neurons, layers, and adjustable weights to learn patterns from data and make predictions.

Bias

Bias is an additional numerical value added to the weighted sum of inputs in a neuron. It helps a neural network adjust its output and learn patterns more effectively.

In simple words, bias helps a neuron adjust how easily it produces a particular output.

Formula

$$z = (x_1 \cdot w_1 + x_2 \cdot w_2) + b$$

Where:

- x_1, x_2 = Input values
- w_1, w_2 = Weights assigned to inputs
- b = Bias
- z = Weighted sum after adding bias

Example

Suppose a neural network predicts whether a student will pass an examination based on study hours.

Given:

- Study hours, $x = 4$
- Weight, $w = 2$
- Bias, $b = -3$

Step 1: Multiply input by weight.

$$x \times w = 4 \times 2 = 8$$

Step 2: Add bias.

$$z = (4 \times 2) + (-3)$$

$$z = 8 - 3 = 5$$

The final weighted sum is 5.

If the bias were 0, the result would be 8. Therefore, bias adjusts the value calculated by the neuron.

Activation Function in Neural Networks

Definition

An activation function is a mathematical function applied to the result calculated by a neuron. It determines the neuron's output and helps the neural network learn complex patterns.

In simple words, the activation function decides how the calculated value should be converted into an output.

Formula

First, the neuron calculates:

$$z = \sum_{i=1}^n x_i w_i + b$$

Then, the activation function is applied:

$$a = f(z)$$

Where:

- x_i = Input values

- w_i = Weights
- b = Bias
- z = Weighted sum plus bias
- $f(z)$ = Activation function
- a = Final output of the neuron

Deep Learning

Deep Learning is a branch of Machine Learning that uses artificial neural networks with multiple layers to learn patterns from large amounts of data and make predictions.

Example: Deep learning can identify cats and dogs by learning features such as ears, eyes, shapes, and fur from images.

What Makes Deep Learning “Deep”?

Deep learning is called “deep” because it uses multiple hidden layers between the input layer and the output layer.

Each layer learns different features from the input data.

Example: Image Recognition

- First layer: Detects edges and lines.
- Second layer: Detects shapes and patterns.
- Third layer: Detects parts such as eyes, ears, and noses.
- Final layer: Identifies the image as a cat or dog.

Input Image

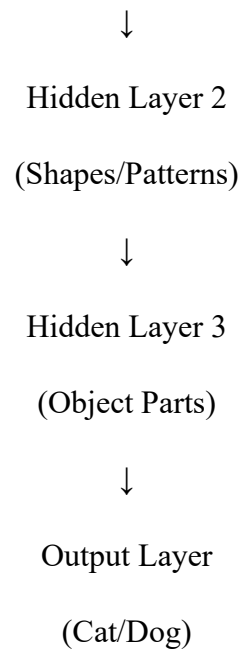


Input Layer



Hidden Layer 1

(Edges/Lines)



Applications of Deep Learning

1. Image and face recognition
2. Speech recognition
3. Self-driving cars
4. Medical image analysis
5. Language translation
6. Chatbots and virtual assistants

2. Why Does Deep Learning Need GPUs?

GPU (Graphics Processing Unit) is a processor designed to perform many calculations simultaneously.

Deep learning models perform millions or billions of mathematical operations during training. GPUs help perform these calculations faster than CPUs for many deep learning tasks.

Reasons for Using GPUs

- 1). Parallel Processing:** GPUs can perform many mathematical operations simultaneously.
- 2). Faster Training:** GPUs can reduce the time required to train large neural networks.

3). Large Data Processing: GPUs are suitable for processing large datasets, especially images and videos.

4). Matrix Calculations: Deep learning uses many matrix and vector operations, which GPUs handle efficiently.

5). Handling Complex Models: GPUs help train deep neural networks containing many layers and parameters.

CPU	GPU
Central Processing Unit	Graphics Processing Unit
Usually has fewer, more general-purpose cores	Has many cores designed for parallel workloads
Suitable for general-purpose tasks	Suitable for highly parallel calculations
Often slower for large-scale neural network training	Often faster for large-scale neural network training

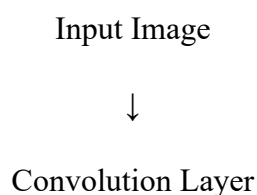
Example: Training an image recognition model on a large dataset may take much longer on a CPU than on a suitable GPU.

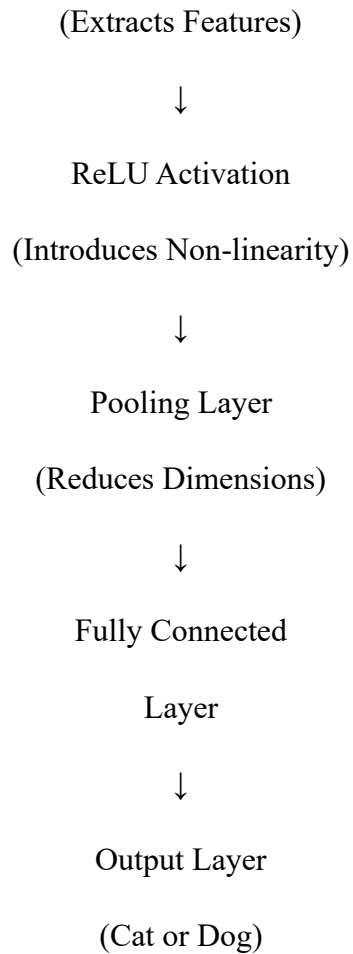
Note: A GPU is not compulsory for every deep learning model. Small models can also be trained using a CPU.

3. CNN (Convolutional Neural Network)

CNN (Convolutional Neural Network) is a type of deep learning neural network mainly used to process and recognize images. It automatically learns visual features such as edges, shapes, textures, and objects from images.

Architecture of CNN





Main Layers of CNN

1. Input Layer

The input layer receives the image that we want the computer to recognize.

Example: We provide a picture of a cat to the CNN.

The computer represents the image as numerical values called pixels. These values contain information about the image's colours and brightness.

2. Convolution Layer

The convolution layer uses small filters, also called kernels, to scan an image and detect features such as edges, lines, and shapes.

Example: When processing a cat image, the convolution layer may detect:

- Edges of the ears.
- Lines around the eyes.

- Shapes of the nose.
- Patterns in the fur.

3. ReLU Activation Layer

ReLU stands for Rectified Linear Unit. It replaces negative values with zero while keeping positive values unchanged.

$$f(x) = \max(0, x)$$

It helps the CNN learn complex patterns.

4. Pooling Layer

The pooling layer reduces the spatial size of feature maps while retaining important information.

Example: Max pooling selects the maximum value from a small region of a feature map.

Advantages:

- Reduces computation.
- Reduces the number of values processed by later layers.
- Helps retain important features.

5. Fully Connected Layer

The fully connected layer uses the learned features to make a prediction.

Example: It uses features detected in an image to determine whether the image contains a cat or a dog.

6. Output Layer

The output layer gives the final classification result.

Example:

- Cat: 90%
- Dog: 10%

The CNN predicts Cat because it has the highest probability.

Applications of CNN

1. Face recognition
2. Image classification
3. Object detection
4. Medical image analysis
5. Self-driving cars
6. Handwriting recognition
7. Video surveillance

Advantages of CNN

- 1). Automatically learns features from images.
- 2). Reduces the need for manual feature extraction.
- 3). Works effectively for many image recognition tasks.
- 4). Can recognize complex visual patterns.
- 5). Uses shared filters to reduce the number of parameters compared with some fully connected image-processing networks.

Deep learning uses neural networks with multiple hidden layers to learn complex patterns. GPUs accelerate the large number of calculations required for training. CNNs are specialized deep learning networks widely used for image processing, feature extraction, and image classification.

Computer Vision

Computer Vision is a branch of Artificial Intelligence (AI) that enables computers to see, understand, and interpret images and videos.

It allows machines to identify objects, recognize faces, detect patterns, and analyze visual information.

How AI Sees Images

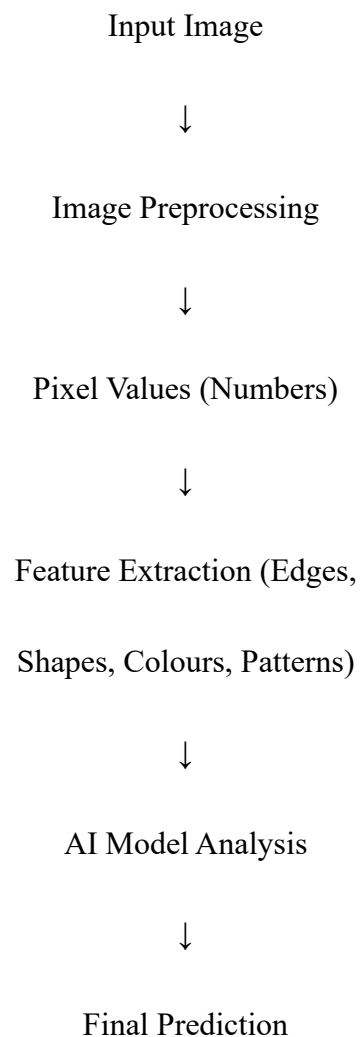
AI does not see images like humans. It processes images as numerical values called pixels.

Each pixel represents a small part of an image. In a grayscale image, pixel values usually range from 0 to 255:

- 0 represents black.
- 255 represents white.
- Values between 0 and 255 represent different shades of gray.

In a colour image, each pixel usually contains three values: Red (R), Green (G), and Blue (B).

Working of Computer Vision



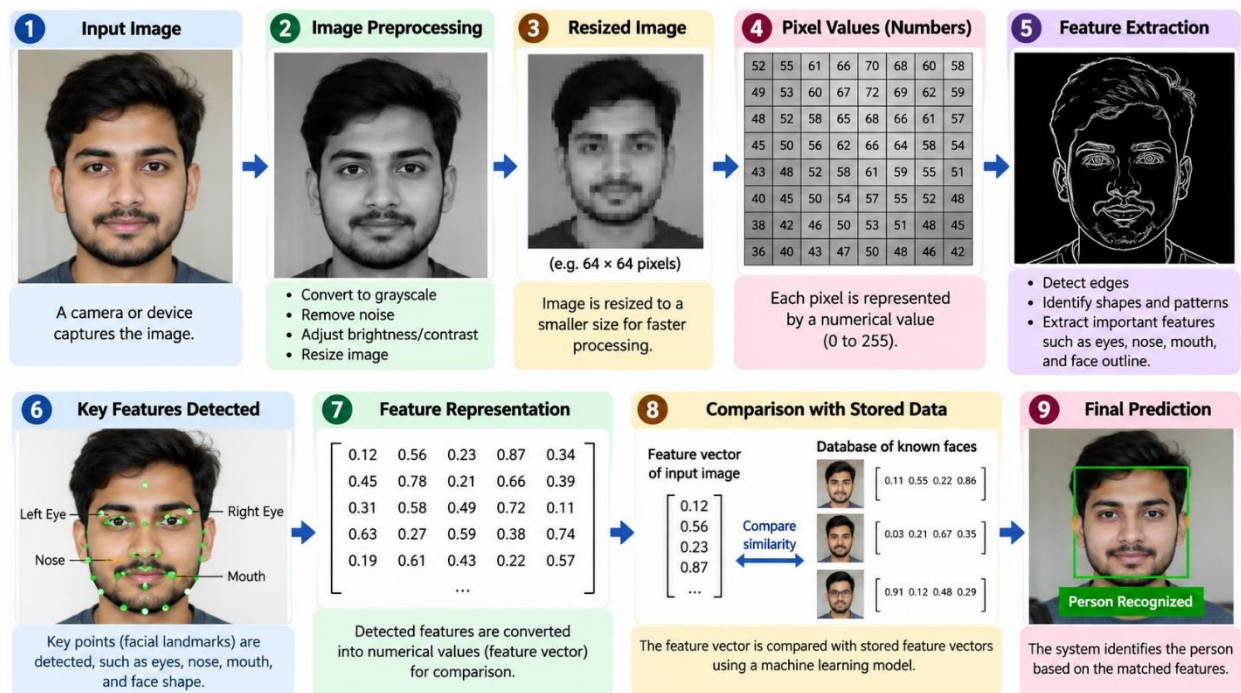
Steps in Computer Vision

- 1). **Image Acquisition:** The computer receives an image from a camera, scanner, or other source.
- 2). **Image Preprocessing:** The image is resized, cleaned, or adjusted to improve its quality.
- 3). **Feature Extraction:** The AI model identifies important features such as edges, shapes, colours, and textures.
- 4). **Image Analysis:** The model analyses the extracted features and identifies patterns.
- 5). **Prediction or Classification:** The system produces a result, such as identifying a person, object, or disease.

Applications of Computer Vision

A. Face Recognition

Face recognition is a computer vision application that identifies or verifies a person using facial features.



Working:

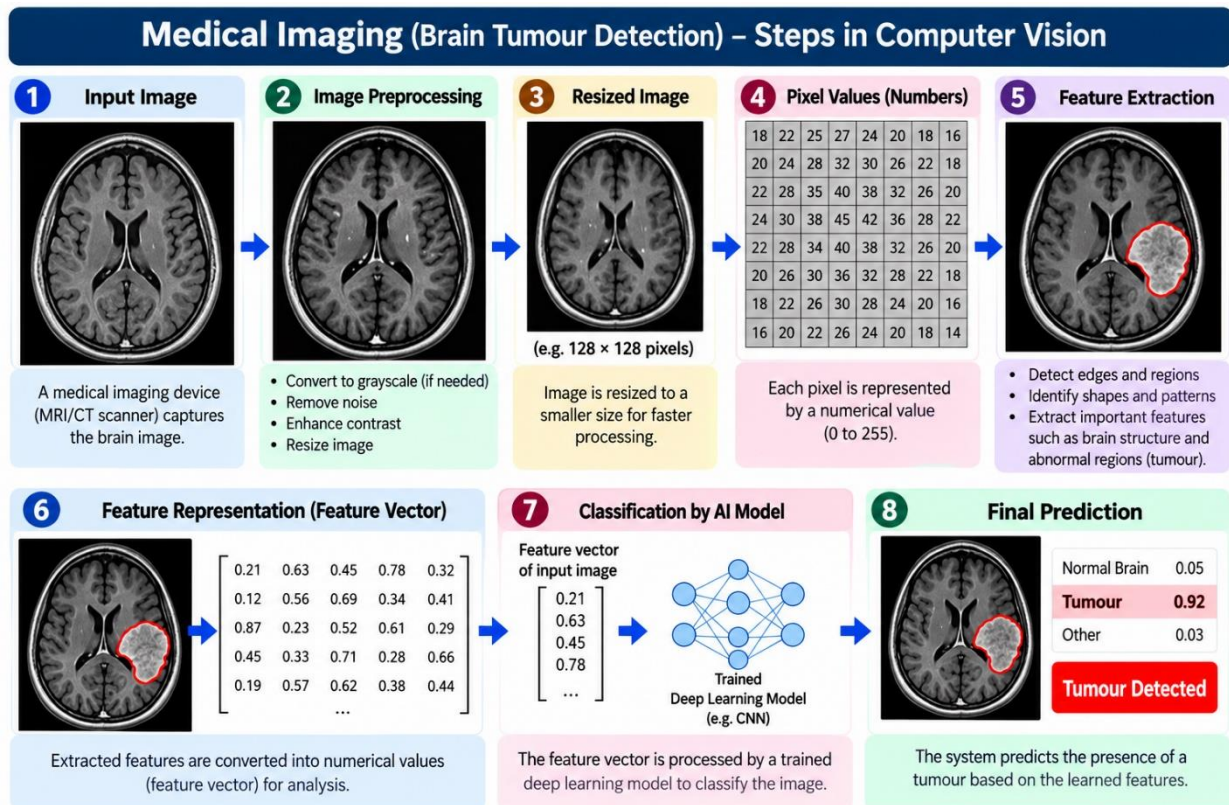
- 1). A camera captures a person's face.
- 2). The system detects the face in the image.
- 3). The AI model extracts facial features, such as the distance between the eyes and the shape of the face.
- 4). The extracted features are compared with stored facial information.
- 5). The system verifies or identifies the person.

Applications:

- Mobile phone unlocking
- Attendance systems
- Security and surveillance
- Identity verification

B. Medical Imaging

Medical imaging uses computer vision to analyse medical images such as X-rays, CT scans, and MRI scans.



Working:

- 1). A medical image is captured using a medical imaging device.
- 2). The image is processed to improve its quality.
- 3). The AI model identifies patterns and unusual regions.
- 4). The model detects possible abnormalities, such as tumours or fractures.
- 5). The system provides results that help doctors examine the image.

Applications:

- Tumour detection
- Detection of bone fractures
- Identification of abnormalities in X-rays
- Analysis of MRI and CT scans

- Assistance in medical diagnosis

Advantages of Computer Vision

- 1). It can analyse large numbers of images quickly.
- 2). It helps automate repetitive visual tasks.
- 3). It can detect patterns that may be difficult to notice.
- 4). It improves efficiency in security and medical image analysis.
- 5). It can work continuously without fatigue.

Limitations of Computer Vision

- 1). It requires sufficient training data.
- 2). Poor lighting or low-quality images may reduce accuracy.
- 3). Incorrect predictions may occur.
- 4). Training complex models may require high computing power.
- 5). Face recognition can raise privacy concerns.

Computer Vision enables computers to interpret images and videos using AI techniques. It is widely used in face recognition, medical imaging, object detection, autonomous vehicles, and security systems.

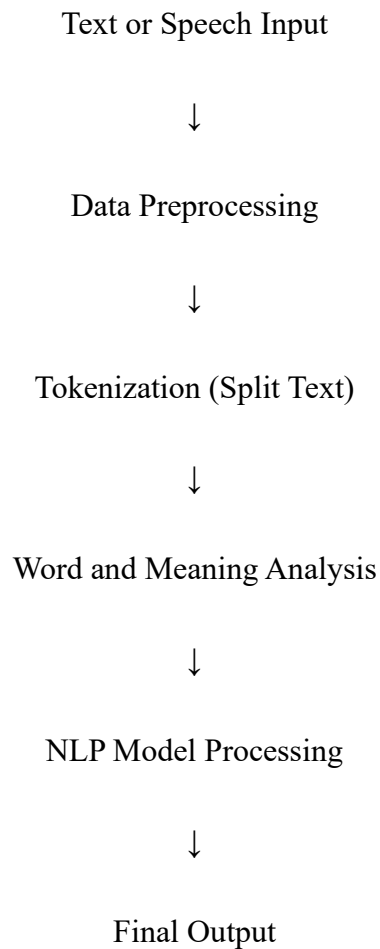
Natural Language Processing (NLP)

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that enables computers to understand, interpret, process, and generate human languages such as English and Hindi.

NLP helps computers communicate with humans through text and speech.

How AI Understands Text and Speech

AI processes human language through several steps.



Steps Involved in NLP

- 1). **Input:** The system receives text or speech from the user.
- 2). **Preprocessing:** Unnecessary symbols, extra spaces, or unwanted words may be removed.
- 3). **Tokenization:** The text is divided into smaller units called tokens, such as words or subwords.
- 4). **Word Analysis:** The system analyses words, their forms, and their relationships.
- 5). **Meaning Analysis:** The NLP model uses context to understand the intended meaning of the text.

6). Model Processing: The AI model processes the language and identifies patterns.

7). Output Generation: The system produces a response, translation, summary, or other result.

For speech input, speech recognition first converts spoken words into text for further NLP processing.

Example of NLP

Input: “I am feeling happy today.”

- **Tokenization:** I, am, feeling, happy, today
- **Analysis:** The system examines the words and their context.
- **Sentiment detection:** Positive sentiment.
- **Output:** “The sentence expresses happiness.”

Applications of NLP

A. Chatbots

NLP enables chatbots to understand user questions and generate relevant responses.

Example: ChatGPT answering a question asked in English.

B. Machine Translation

NLP translates text from one language into another.

Example: Translating English sentences into Hindi using Google Translate.

C. Sentiment Analysis

NLP identifies emotions or opinions expressed in text, such as positive, negative, or neutral sentiment.

Example: Analysing customer reviews of a product.

D. Speech Recognition

Speech recognition converts spoken language into text.

Example: Converting a spoken message into written text.

E. Text Summarization

NLP generates a shorter version of a long document while preserving its main points.

Example: Summarizing a long chapter into short notes.

F. Virtual Assistants

NLP helps virtual assistants understand voice commands and respond to users.

Example: Google Assistant answering a spoken question.

Advantages of NLP

- 1). Enables communication between humans and computers using natural language.
- 2). Processes large amounts of text quickly.
- 3). Automates translation and text summarization.
- 4). Helps analyse customer feedback and opinions.
- 5). Improves chatbot and virtual assistant functionality.

Limitations of NLP

- 1). Human language can be ambiguous.
- 2). Sarcasm and emotions can be difficult to interpret correctly.
- 3). Regional accents and background noise can affect speech recognition.
- 4). Different meanings of the same word can create confusion.
- 5). Biased or insufficient training data can produce incorrect results.