

# Unit 1: Machine Learning

---

Introduction to machine learning, Applications, Classification; Supervised Learning-> Linear Regression Cost function, Gradient descent; Logistic Regression, Nearest-Neighbors, Gaussian function.

---

## Introduction to Machine Learning

---

### What is Machine Learning?

---

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that enables systems to learn and improve from experience without being explicitly programmed. It focuses on developing algorithms that can analyse data, identify patterns, and make predictions or decisions.

### Key Features of Machine Learning

---

- **Data-Driven:** ML systems learn from data rather than relying on predefined rules.
- **Automation:** Reduces human intervention by automating tasks.
- **Improvement Over Time:** Models improve as they are exposed to more data.

### Definition

---

 **Arthur Samuel (1959):** Machine Learning is the "field of study that gives computers the ability to learn without being explicitly programmed."

---

## Components of Machine Learning

---

### 1. Data:

- Raw information used to train the model.
- Example: Images, text, numerical data, etc.

### 2. Model:

The mathematical representation of the real-world process.

### 3. Algorithms:

- Procedures used to train models.
- Examples: Linear Regression, Decision Trees.

### 4. Evaluation:

Assessing the performance of the model using metrics like accuracy or precision.

---

## Characteristics of Machine Learning

---

click here to visit

- **Scalability:** ML models can handle large datasets effectively.
  - **Adaptability:** Models can adjust to new, unseen data.
  - **Non-linearity:** Capable of modelling complex relationships.
- 

## Types of Machine Learning

---

### **1. Supervised Learning:**

The model learns from labelled data.

Example: Predicting house prices.

### **2. Unsupervised Learning:**

The model finds patterns in unlabelled data.

Example: Customer segmentation.

### **3. Reinforcement Learning:**

The model learns through trial and error to maximise rewards.

Example: Game-playing AI like AlphaGo.

---

## Applications of Machine Learning

---

- **Healthcare:** Predicting diseases, drug discovery.
  - **Finance:** Fraud detection, stock price prediction.
  - **Retail:** Personalised recommendations, inventory management.
  - **Autonomous Systems:** Self-driving cars, robotics.
- 

## Advantages of Machine Learning

---

- Automates decision-making processes.
  - Provides accurate and scalable solutions.
  - Identifies hidden patterns in large datasets.
- 

## Disadvantages of Machine Learning

---

- **Data Dependency:** Requires a large amount of high-quality data.
  - **Complexity:** Algorithms and models can be challenging to interpret.
  - **Computational Cost:** Demands significant computational resources.
- 

**NOTE: Machine Learning is a foundational technology powering advancements in AI, Big Data, and Analytics.**

---

## Classification in Supervised Learning

---

### What is Supervised Learning?

---

Supervised Learning is a type of Machine Learning where the model is trained on labelled data. Each input comes with an associated output, and the model learns to map inputs to the correct outputs.

## Classification in Supervised Learning

---

Classification is a supervised learning task where the model predicts a discrete output, or **class label**, based on input features.

---

## Key Characteristics of Classification

---

- **Discrete Output:** The output belongs to a predefined set of categories or classes.
  - **Examples of Classification Tasks:**
    - Email Spam Detection: Spam or Not Spam.
    - Disease Diagnosis: Positive or Negative.
    - Image Recognition: Cat, Dog, or Bird.
- 

## Workflow of a Classification Model

---

### 1. Data Collection:

- Gather labelled data.
- Example: A dataset of customer emails with labels like "spam" or "not spam."

### 2. Feature Engineering:

- Extract meaningful features from raw data.
- Example: Email length, number of keywords, etc.

### 3. Model Selection:

Choose a suitable classification algorithm, such as:

- Decision Tree
- Support Vector Machine (SVM)
- Logistic Regression
- Neural Networks

### 4. Training:

Train the model using the labelled dataset.

### 5. Testing and Evaluation:

Evaluate the model using metrics like:

- **Accuracy:** Percentage of correct predictions.
- **Precision and Recall:** Performance on specific classes.
- **F1 Score:** Balances precision and recall.

### 6. Prediction:

Use the trained model to classify new, unseen data.

## Common Algorithms for Classification

---

### **1. Logistic Regression:**

- Predicts probabilities for binary classification tasks.
- Example: Predicting if a customer will purchase a product (Yes/No).

### **2. Decision Trees:**

- Splits data based on feature values to classify samples.
- Example: Diagnosing a disease based on symptoms.

### **3. Support Vector Machines (SVM):**

Finds the best boundary (hyperplane) to separate classes.

### **4. Naive Bayes:**

- Based on Bayes' theorem and assumes feature independence.
- Example: Text classification.

### **5. K-Nearest Neighbours (KNN):**

Assigns the class of the nearest neighbours in the feature space.

### **6. Neural Networks:**

Uses interconnected layers of neurons to classify complex data.

---

## Evaluation Metrics for Classification

---

- **Confusion Matrix:** Provides a summary of prediction results on a classification problem.

- True Positive (TP): Correctly predicted positive cases.
- True Negative (TN): Correctly predicted negative cases.
- False Positive (FP): Incorrectly predicted as positive.
- False Negative (FN): Incorrectly predicted as negative.

- **Accuracy:**

- **Precision:**

- **Recall:**

- **F1 Score:**
- 

## Applications of Classification

---

- **Medical Diagnosis:** Predicting diseases (e.g., cancer detection).
- **Fraud Detection:** Identifying fraudulent transactions.
- **Customer Segmentation:** Grouping customers based on purchase behaviour.
- **Natural Language Processing (NLP):** Sentiment analysis, spam filtering.

**CAUTION: Ensure the dataset is balanced (similar class proportions) to avoid biased predictions.**

---

## Linear Regression: Cost Function

---

### What is Linear Regression?

---

Linear Regression is a supervised learning algorithm used to model the relationship between a dependent variable (target) and one or more independent variables (features). It aims to find the best-fitting line that minimises the error between predicted and actual values.

---

### Cost Function in Linear Regression

---

The **Cost Function** in Linear Regression measures how well the model's predictions match the actual target values. It quantifies the error by comparing predicted values () with actual values () in the training dataset.

---

### Mean Squared Error (MSE) Cost Function

---

The most commonly used cost function in Linear Regression is the **Mean Squared Error (MSE)**, defined as:

Where:

- : Cost function.
  - : Number of training examples.
  - : Predicted value for the -th example.
  - : Actual value for the -th example.
  - : Parameters (weights and bias) of the linear model.
- 

### Steps to Minimise the Cost Function

---

#### 1. Initialisation:

Start with random values for (weights and bias).

#### 2. Gradient Descent Algorithm:

Use Gradient Descent to update iteratively to minimise .

#### Update Rule:

Where:

- : Learning rate (controls step size).
- : Partial derivative of the cost function with respect to parameter .

click here to visit

### 3. Convergence:

Repeat the update until the cost function converges to a minimum value.

---

## Why Divide by in MSE?

- The factor simplifies the derivative of the cost function during Gradient Descent.
  - The factor cancels out when differentiating .
- 

## Example: Calculating MSE

Given a dataset with examples:

- Actual values:
  - Predicted values:
1. Compute the squared error for each example:
  2. Sum the squared errors:
  3. Compute MSE:
- 

## Importance of the Cost Function

- Helps measure model performance during training.
  - Guides the optimisation process using Gradient Descent.
- 

**NOTE: The cost function is crucial for tuning the parameters to achieve the best-fit line and minimise prediction error.**

---

## Gradient Descent

### What is Gradient Descent?

Gradient Descent is an optimisation algorithm used to minimise a cost function by iteratively adjusting model parameters ( $\theta$ ). It is widely used in Machine Learning to train models by finding the optimal parameters that minimise the error.

---

## Key Idea

Gradient Descent works by:

1. Calculating the slope (gradient) of the cost function with respect to each parameter.
  2. Updating the parameters in the opposite direction of the gradient to reduce the cost.
-

## Gradient Descent Algorithm

---

For each parameter (where ), update the value using the following rule:

Where:

- : Parameter to be updated.
  - : Learning rate (controls the size of the steps).
  - : Partial derivative of the cost function with respect to .
- 

## Steps in Gradient Descent

---

### 1. Initialise Parameters:

Start with random or zero values for all parameters .

### 2. Compute Predictions:

Use the current parameters to make predictions () .

### 3. Calculate the Cost Function:

Compute the error using the cost function .

### 4. Update Parameters:

Use the gradient of the cost function to adjust the parameters .

### 5. Repeat Until Convergence:

Stop when the cost function does not decrease significantly.

---

## Learning Rate ()

---

The learning rate determines the step size for updating parameters:

- **Small** : Slow convergence (long training time).
- **Large** : May overshoot or fail to converge.

 **TIP:** Use techniques like learning rate scheduling or adaptive methods (e.g., Adam, RMSProp) to optimise .

---

## Types of Gradient Descent

---

### 1. Batch Gradient Descent:

- Uses the entire training dataset to compute the gradient.
- **Advantages:** Stable convergence.
- **Disadvantages:** Computationally expensive for large datasets.

## 2. Stochastic Gradient Descent (SGD):

- Updates parameters using a single training example at each step.
- **Advantages:** Faster updates; works well for large datasets.
- **Disadvantages:** Noisy convergence; may overshoot.

## 3. Mini-Batch Gradient Descent:

- Combines the advantages of Batch and Stochastic methods.
  - Updates parameters using small subsets (mini-batches) of the training dataset.
  - **Advantages:** Efficient and more stable than SGD.
- 

## Gradient Descent for Linear Regression

For Linear Regression, the gradient of the cost function is:

Where:

- : Predicted value.
  - : Actual value.
- 

## Convergence

Gradient Descent is said to converge when:

1. The cost function becomes stable (does not change significantly between iterations).
  2. The gradients approach zero.
- 

## Example

Given a simple dataset and a linear model:

- (true equation).
- Initial .

### Steps:

1. Predict .
2. Compute the cost .
3. Calculate gradients:
4. Update , .
5. Repeat until convergence.

**NOTE: Gradient Descent can be sensitive to the choice of learning rate and initialisation. Experimentation and tuning are often required.**

---

## Logistic Regression

---

### What is Logistic Regression?

Logistic Regression is a supervised learning algorithm used for binary classification problems. Instead of predicting continuous values (as in Linear Regression), it predicts the probability that a given input belongs to one of two classes.

---

### Key Concept: Sigmoid Function

Logistic Regression uses the **sigmoid function** to map the output of a linear equation to a probability value between 0 and 1.

The sigmoid function is defined as:

Where:

- : Linear combination of input features and weights.
  - : Sigmoid function.
  - : Predicted probability.
- 

### Decision Rule

The output of the sigmoid function is interpreted as a probability:

- If , predict **class 1**.
  - If , predict **class 0**.
- 

### Cost Function for Logistic Regression

The cost function for Logistic Regression is designed to handle probabilities and ensure efficient optimisation:

Where:

- : Number of training examples.
  - : Actual label for the -th training example.
  - : Predicted probability for the -th training example.
- 

### Gradient Descent for Logistic Regression

Logistic Regression uses Gradient Descent to optimise parameters . The update rule is:

Where:

- : Learning rate.

- : Partial derivative of the cost function with respect to .
- 

## Applications of Logistic Regression

---

- **Healthcare:** Predicting the likelihood of diseases (e.g., heart attack, diabetes).
  - **Marketing:** Customer conversion prediction (e.g., click-through rate).
  - **Finance:** Fraud detection and credit risk assessment.
  - **Natural Language Processing (NLP):** Sentiment analysis, spam filtering.
- 

## Assumptions of Logistic Regression

---

1. **Linear Relationship:** Assumes a linear relationship between input features and the log-odds of the output.
  2. **Independence:** Features are assumed to be independent of each other.
  3. **Binary Output:** Typically used for two-class problems, though it can be extended to multi-class classification.
- 

## Advantages of Logistic Regression

---

- **Simplicity:** Easy to implement and interpret.
  - **Efficient:** Computationally inexpensive for small datasets.
  - **Probability Output:** Provides probabilistic predictions for better decision-making.
- 

## Disadvantages of Logistic Regression

---

- **Linear Boundary:** Cannot model non-linear relationships without feature transformations.
  - **Feature Engineering Required:** Sensitive to irrelevant or highly correlated features.
  - **Large Datasets:** Performance may degrade on very large datasets.
- 

## Example

---

### Problem:

---

Predict whether a student will pass an exam based on study hours.

### Dataset:

---

- (hours studied).
  - (pass/fail).
- 

### Steps:

---

1. Compute .

2. Apply sigmoid:
3. Use Gradient Descent to optimise based on the cost function.

### **Prediction:**

---

For :

- Compute and .
- Predict if .

**NOTE: Logistic Regression is a foundational algorithm in classification tasks and serves as a baseline for more advanced models.**

---

## **Nearest Neighbors**

---

### **What is the Nearest Neighbors Algorithm?**

---

The **Nearest Neighbors** algorithm is a non-parametric, instance-based supervised learning technique used for classification and regression. It relies on measuring the similarity (or distance) between data points to make predictions.

---

### **Key Idea**

---

The algorithm classifies a data point based on the **majority class** of its nearest neighbours in the feature space.

---

## **Workflow of Nearest Neighbors**

---

### **1. Data Representation:**

Each data point is represented as a vector of features.

### **2. Distance Calculation:**

The algorithm computes the distance between the query point and all points in the dataset.

### **3. Neighbour Selection:**

Select the -nearest data points based on the computed distance.

### **4. Prediction:**

- For classification: Assign the majority class among the -nearest neighbours.
  - For regression: Compute the average (or weighted average) of the -nearest neighbours' target values.
-

## Distance Metrics

---

1. **Euclidean Distance:**
2. **Manhattan Distance:**
3. **Minkowski Distance:**
4. **Hamming Distance (for categorical data):**

Counts the number of differing features.

 **TIP:** Choose the distance metric based on the nature of your data.

---

## Value of (Number of Neighbours)

---

- **Small :** Sensitive to noise and may lead to overfitting.
- **Large :** Provides smoother decision boundaries but may underfit the data.

 **CAUTION:** Use techniques like cross-validation to select an optimal .

---

## Applications of Nearest Neighbors

---

- **Classification:**
  - Spam detection (e.g., spam or not spam).
  - Image recognition (e.g., handwritten digit classification).
- **Regression:**

Predicting house prices based on location and features.
- **Anomaly Detection:**

Identifying unusual patterns in data, like fraud detection.

## Advantages of Nearest Neighbors

---

- **Simplicity:** Easy to understand and implement.
- **No Assumptions:** Does not assume any specific data distribution.
- **Versatility:** Can be used for both classification and regression tasks.

## Disadvantages of Nearest Neighbors

---

- **High Computational Cost:** Distance calculation can be slow for large datasets.
- **Storage Requirement:** Requires storing the entire dataset in memory.
- **Sensitive to Irrelevant Features:** Performance can degrade if irrelevant features are present.

- **Curse of Dimensionality:** Performance decreases as the number of dimensions increases.
- 

## Example: Classification with -NN

---

### Dataset:

---

- Features: Height (cm), Weight (kg).
- Labels: Gender (Male/Female).

**Height    Weight    Label**

170	65	Male
160	55	Female
180	75	Male

### Query Point:

---

Height = 175, Weight = 70.

### Steps:

---

1. Compute the distance from the query point to each data point.
  2. Select the  $-$ nearest neighbours (e.g., ).
  3. Assign the majority label (e.g., "Male").
- 

## Implementation in Python

---

```
from sklearn.neighbors import KNeighborsClassifier

# Data
X = [[170, 65], [160, 55], [180, 75]]
y = ["Male", "Female", "Male"]

# Model
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X, y)

# Prediction
query = [[175, 70]]
prediction = model.predict(query)
print("Predicted label:", prediction)
```

---

## Gaussian Function

---

### What is the Gaussian Function?

---

click here to visit

The **Gaussian Function**, also known as the **Normal Distribution** or **Bell Curve**, is a fundamental function in mathematics and statistics. It describes a symmetric, bell-shaped curve that represents the distribution of data around a central mean.

---

## **Mathematical Formula**

The Gaussian function is defined as:

Where:

- $x$  : Random variable.
  - $\mu$  : Mean (centre of the curve).
  - $\sigma$  : Standard deviation (spread of the curve).
  - $\sigma^2$  : Variance.
- 

## **Key Properties of the Gaussian Function**

### **1. Symmetry:**

The curve is symmetric about the mean  $\mu$ .

### **2. Mean ( $\mu$ ):**

Represents the peak of the curve, where the data is most concentrated.

### **3. Standard Deviation ( $\sigma$ ):**

- Controls the spread of the curve.
- A larger results in a flatter and wider curve.
- A smaller results in a taller and narrower curve.

### **4. Area Under the Curve (AUC):**

The total area under the curve equals 1, representing the entire probability space.

### **5. Inflection Points:**

Located at  $\mu - \sigma$  and  $\mu + \sigma$ .

---

## **Applications of the Gaussian Function**

### **1. Statistics:**

- Models normal distributions for data analysis.
- Example: Heights of individuals in a population.

### **2. Machine Learning:**

Basis for algorithms like Gaussian Naive Bayes and Gaussian Mixture Models.

### 3. Signal Processing:

Used in smoothing filters (e.g., Gaussian blur in image processing).

### 4. Physics:

Models phenomena like heat distribution and quantum mechanics.

### 5. Probability Density Function (PDF):

Represents the probability of a random variable taking on a value near .

---

## **Example: Standard Normal Distribution**

A **Standard Normal Distribution** is a special case of the Gaussian function where:

- 
- 

The formula simplifies to:

## **Properties:**

- Mean = 0.
  - Standard deviation = 1.
  - 68% of the data lies within 1 standard deviation ( ).
  - 95% of the data lies within 2 standard deviations ( ).
- 

## **Relation to Machine Learning**

The Gaussian function plays a critical role in:

- **Gaussian Naive Bayes:** Assumes features follow a Gaussian distribution.
  - **Kernel Methods:** Used in Support Vector Machines (SVM) and Gaussian Processes.
  - **Clustering Algorithms:** Basis of Gaussian Mixture Models (GMM).
- 

**NOTE: The Gaussian function is ubiquitous across disciplines due to its simplicity and ability to model natural phenomena.**

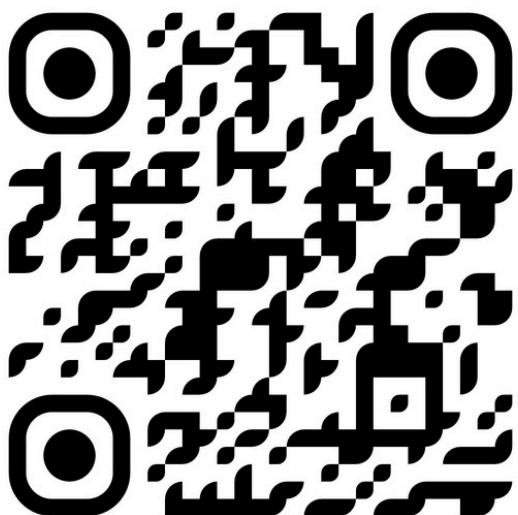
---

# *Well done!*

You've Completed  
Unit 1 – Machine Learning

Visit NotesBuddy.in to:

- Test yourself with quizzes
- Revise fast with flashcards
- Master key concepts with one-shots
- Prepare smartly with important questions
- Learn from topper handwritten notes



NotesBuddy.in

or click here

SCAN IT

A large curved arrow originates from the text "SCAN IT" and points towards the QR code. Another curved arrow originates from the text "or click here" and points towards the URL "NotesBuddy.in".