**Unsupervised Learning**

## **What is Unsupervised Learning?**

Unsupervised learning is a type of machine learning where the model is **not given labeled data.**  
 This means:

* There are no target/output values (like “spam” or “not spam”).
* The algorithm tries to find hidden patterns, structures, or relationships in the data on its own.

Simply: The model learns from raw data without being told what the correct output should be.

**Why Use Unsupervised Learning?**

It’s useful when:

* You don’t have labelled data (labelling is expensive or impossible).
* You want to discover natural groupings or reduce data complexity.

**Key Goals**

* **Clustering:** Group similar data points together.
* **Dimensionality Reduction:** Simplify data by reducing the number of features while keeping important information (like PCA).
* **Association:** Find rules or relationships between variables (like market basket analysis).

**Types of Unsupervised Learning**

1. **Clustering** → Finding groups in data (K-Means, Hierarchical, DBSCAN)
2. **Dimensionality Reduction** → Summarizing data (PCA, t-SNE)
3. **Association Rule Learning** → Finding rules (Apriori, FP-Growth)

**Clustering Introduction**

**What is Clustering?**

Clustering is the process of **grouping data points** so that:

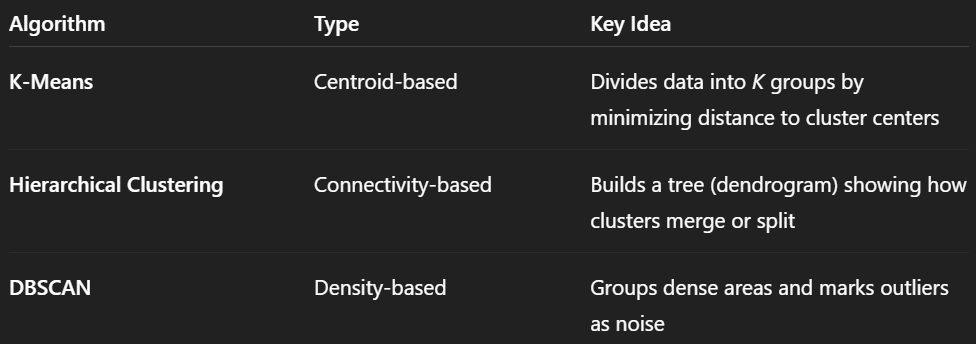
* Points **in the same cluster** are **more** **similar** to each other.
* Points **in different clusters** are **more dissimilar**.

Example: In a dataset of customers, clustering can automatically group customers with similar purchasing habits.

**Goal of Clustering**

To find hidden structure or natural groupings in data without any labels.

**Common Clustering Algorithms**



**How Clustering Works (General Idea)**

1. Choose the number of clusters (sometimes automatically found).
2. Assign data points to clusters based on similarity (e.g., Euclidean distance).
3. Update clusters until they stabilize (no big changes).
4. Evaluate results (using metrics like silhouette score or visual plots).

**K-Mean Algorithm**

**Introduction**

K-Means is one of the most popular unsupervised learning algorithms used for clustering — i.e., grouping similar data points together without using labels.

The goal is simple:

Divide the dataset into K clusters, such that points within the same cluster are more like each other than to those in other clusters.

### **Working of K-Means**

K-Means works in an **iterative** way and follows these main steps:

### **Step 1: Choose K**

Decide how many clusters (K) you want to divide your data into.

### **Step 2: Initialize Centroids**

Randomly pick K points from the dataset as the **initial centroids** (these act like the centres of clusters).

### **Step 3: Assign Points to Nearest Centroid**

For each data point, calculate its distance (usually **Euclidean distance**) to each centroid, and assign it to the nearest one.

### **Step 4: Update Centroids**

For each cluster, compute the **mean** of all points assigned to it.  
 This mean becomes the new centroid.

**Step 5: Repeat**

Repeat steps 3 and 4 until:

* The centroids stop changing significantly, or
* You reach a maximum number of iterations.

### **Example (Simple Intuition)**

Let’s say we have 10 data points representing customers’ annual income.

We choose K = 2 (we want 2 groups).

1. Randomly select 2 points as initial centroids.
2. Assign each customer to the nearest centroid → Group A or Group B.
3. Compute the average income of each group → update centroids.
4. Reassign customers using the new centroids.
5. Repeat until centroids don’t move much.

Result: two clusters, maybe low-income customers and high-income customers.

### **Key Points**

* K must be chosen manually (using the **Elbow Method** helps).
* Works well when clusters are spherical and similar in size.
* Sensitive to initialization and outliers.
* Uses distance-based learning, so scaling features (normalization) is important.

**Principal Component Analysis (PCA)**

## **What is PCA?**

Principal Component Analysis (PCA) is a Dimensionality Reduction technique used to reduce the number of features in a dataset while keeping most of the important information.

It’s especially useful when:

* Your dataset has many correlated features.
* You want to speed up model training or visualize high-dimensional data.

## **Goal of PCA**

PCA finds new axes (directions) called Principal Components along which the data varies the most.

In simpler words:

PCA transforms your data into a new coordinate system where:

* The first principal component (PC1) captures the maximum variance.
* The second principal component (PC2) captures the next highest variance, and so on.

## PCA Steps (How It Works)

1. **Standardize the data**
   1. Mean = 0, Variance = 1 (important since PCA is affected by scale)
2. **Compute the covariance matrix**
   1. It tells how features vary together.
3. **Calculate eigenvectors and eigenvalues**
   1. Eigenvectors → principal component directions
   2. Eigenvalues → amount of variance captured by each component
4. **Sort and select top components**
   1. Choose components with the largest eigenvalues.
5. **Transform the data**
   1. Project the original data onto these new axes.

## **Benefits**

* Reduces model complexity
* Removes redundancy (correlated features)
* Helps visualization (e.g., reducing from 3D → 2D)
* Reduces noise

## **Limitations**

* Loses some information
* Difficult to interpret transformed features
* Assumes linear relationships

## **Why do we perform PCA**

* **Too many features can confuse models** — many may be redundant or irrelevant.
* **Fewer features = faster training** and less memory use.
* **Easier visualization** — PCA lets us see high-dimensional data in 2D or 3D.
* **Removes noise** — helps focus on the most meaningful patterns.
* **Prevents overfitting** — fewer, more important features make models generalize better.

**Principal Component Analysis (PCA)**

**What is Dimensionality Reduction?**

Dimensionality Reduction means reducing the number of input features (dimensions) in your dataset while keeping as much important information as possible.

In simple terms:

It’s like summarizing your data - fewer columns, but still meaningful.

**Why Do We Need It?**

When datasets have many features (high-dimensional data):

1. Models become slow and complex
2. Overfitting may occur (model learns noise)
3. Hard to visualize data
4. Some features may be irrelevant or correlated

So, we reduce dimensions to:

* Simplify the model
* Speed up computation
* Remove noise
* Improve accuracy
* Visualize patterns more easily

**Types of Dimensionality Reduction**

**(A) Feature Selection**

Choose only the most important features.

Example methods:

* Correlation analysis
* Chi-square test
* Recursive Feature Elimination (RFE)

**(B) Feature Extraction**

Create new features from existing ones.

Example methods:

* PCA (Principal Component Analysis)
* LDA (Linear Discriminant Analysis)
* t-SNE (for visualization)

**Example (Simple Intuition)**

Imagine a dataset with:

Feature 1 → height (cm)

Feature 2 → height (inches)

These two features give almost the same information, so we can combine or remove one → this is dimensionality reduction.

**Real-life Uses**

* Face recognition (reducing pixel data)
* Image compression
* Noise removal in sensor data
* Visualization of high-dimensional datasets

**Chi-Square Test**

**What is the Chi-Square Test?**

The Chi-Square (χ²) Test measures how much two categorical variables are related (or independent).

In machine learning, we use it to find which features are most related to the target variable.

In simple terms:

It helps us select the features that have the strongest relationship with the output (target).

**Why Use Chi-Square Test in ML?**

Because not all features affect the output equally.  
 Chi-square helps us rank features by importance — features with higher Chi-square values are more likely to be useful for prediction.

**Conditions for Using Chi-Square Test**

Feature (X) must be categorical or discrete (not continuous).  
Target (Y) must be categorical.  
Observations must be independent.

If your features are numeric, you must first discretize them (convert into bins/categories).

## **Interpretation**

* **High χ² value** → Strong relationship (feature is important).
* **Low χ² value** → Weak or no relationship (feature may be dropped).

## **Example (Concept)**

Suppose you’re predicting whether a person buys a car (Yes/No),  
 and you have a feature like Gender (Male/Female).

The Chi-square test checks if Gender and Car Purchase are independent.  
 If not, Gender is a useful feature.

