# MACHINE LEARNING

*Unit IV — Unsupervised Learning and Ensemble Learning*

**Topics Covered in this Unit**

Introduction to Clustering: Need for Clustering, Taxonomy & Real-World Domain Applications
Hierarchical Clustering: Agglomerative (Bottom-Up), Divisive (Top-Down), Dendrograms & Linkage Criteria
Partitioning Clustering: K-Means Algorithm (WCSS / Inertia), Elbow Method & Silhouette Coefficient
K-Medoids (PAM) Algorithm: Medoid Centroids & Outlier-Resistant Partitioning Mechanics
Density-Based Clustering: DBSCAN Algorithm, Epsilon, MinPts, Core, Border & Noise Point Classifications
Distribution-Based Clustering: Gaussian Mixture Models (GMM) & Expectation-Maximization (EM)
Introduction to Ensemble Learning: Homogeneous vs. Heterogeneous Ensembles, Bias-Variance Reduction
Basic Ensemble Techniques: Max Voting (Hard vs. Soft), Simple Averaging & Weighted Averaging
Advanced Ensembles: Bagging, Random Forest (Subspace Sampling), AdaBoost, Gradient Boosting & Stacking
High-Yield University Exam Question Bank with Model Outlines, Comparisons & Unit Summary

*Compiled in a structured, easy-to-understand, textbook style format*
*Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science & Data Science Students*

# COURSE SYLLABUS & MAPPING

**[ UNIT IV SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM ]**

*This section contains the official syllabus for Unit IV. Verify with your university curriculum mapping.*

**Unit IV Syllabus Breakdown:**

Unit IV - Unsupervised Learning and Ensemble Learning: Introduction to clustering, need for clustering, types of clustering, Hierarchical Clustering – Agglomerative and Divisive methods, Partitioning Methods: K-Means clustering algorithm, advantages and limitations, Elbow method, Silhouette method; K-Medoids, Density-Based Clustering: DBSCAN algorithm, working mechanism, advantages and limitations. Distribution-Based Clustering: Gaussian Mixture Model. Applications, introduction to Ensemble Learning, homogeneous and heterogeneous ensemble methods, advantages and limitations. Basic Ensemble Techniques: Voting (Max Voting, Averaging, Weighted Averaging). Advanced Ensemble Techniques: Bagging and Random Forest. Boosting: AdaBoost, Gradient Boosting , Stacking.

• Course Code: _____ • Semester / Year: _____

# TABLE OF CONTENTS

4.1 Introduction to Clustering & Taxonomies ..... 1
Taxonomy of Clustering Approaches ..... 1
4.2 Hierarchical Clustering (Agglomerative & Divisive) ..... 1
1. Agglomerative (Bottom-Up) vs. Divisive (Top-Down) ..... 1
2. Linkage Criteria (Distance Between Clusters A and B) ..... 1
4.3 Partitioning Methods: K-Means and K-Medoids ..... 1
1. The K-Means Clustering Algorithm ..... 1
Determining Optimal K: Elbow Method & Silhouette Analysis ..... 1
2. K-Medoids Clustering (PAM — Partitioning Around Medoids) ..... 1
4.4 DBSCAN & Gaussian Mixture Models (GMM) ..... 1
1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) ..... 1
2. Gaussian Mixture Models (GMM) & Expectation-Maximization ..... 1
4.5 Ensemble Learning Techniques ..... 1
1. Basic Ensemble Techniques: Voting & Averaging ..... 1
2. Bagging & Random Forest (Variance Reduction) ..... 1
3. Boosting (Bias Reduction) & Stacking ..... 1
Master Comparison Table: Bagging vs. Boosting vs. Stacking ..... 1
4.6 High-Yield University Exam Question Bank ..... 1
Part A: Short Answer Questions (2 to 5 Marks) ..... 1
Q1. Explain the Elbow Method and Silhouette Analysis in K-Means. [4 Marks] ..... 1
Q2. Differentiate between Core, Border, and Noise points in DBSCAN. [3 Marks] ..... 1
Q3. Differentiate between Bagging and Boosting. [4 Marks] ..... 1
Part B: Long Answer Questions (10 to 15 Marks Outline Guides) ..... 1
4.7 Unit Summary & Key Takeaways ..... 1

# 4.1 Introduction to Clustering & Taxonomies

**Definition: Clustering (Cluster Analysis)**

*Clustering is an unsupervised machine learning task that partitions an unlabelled dataset X = {x_1, x_2, ..., x_n} in R^d into distinct homogeneous subgroups (Clusters) such that instances within the same cluster exhibit High Intra-Cluster Similarity, while instances belonging to different clusters exhibit High Inter-Cluster Dissimilarity.*

## Taxonomy of Clustering Approaches

- **1. Hierarchical Clustering:** Builds a nested tree of clusters (Dendrogram) without requiring a pre-specified cluster count K.
- **2. Partitioning Clustering:** Divides data into K non-overlapping spherical partitions, optimizing a centroid-based distance criterion (e.g., K-Means, K-Medoids).
- **3. Density-Based Clustering:** Discovers clusters as dense spatial regions separated by low-density noise regions; finds arbitrary non-spherical shapes (e.g., DBSCAN, OPTICS).
- **4. Distribution-Based Clustering:** Assumes data is generated by a mixture of underlying parametric probability distributions (e.g., Gaussian Mixture Models - GMM).

# 4.2 Hierarchical Clustering (Agglomerative & Divisive)

## 1. Agglomerative (Bottom-Up) vs. Divisive (Top-Down)

- **Agglomerative Approach (Bottom-Up):** Starts with n individual clusters (each data point is a singleton cluster). At each step, the two closest clusters are iteratively merged according to a linkage metric until all points belong to a single root cluster. Represented as a Dendrogram tree. Time complexity: O(n^3) or O(n^2 log n).
- **Divisive Approach (Top-Down):** Starts with all n data points in one single all-inclusive cluster. Recursively splits clusters into smaller sub-clusters using flat clustering until each data point forms its own cluster. Computationally expensive (O(2^n)).

## 2. Linkage Criteria (Distance Between Clusters A and B)

- • **Single Linkage (Minimum Distance):** d(A, B) = min { d(x, y) : x in A, y in B }. Can capture non-elliptical curved shapes; prone to Chaining Effect (long stringy clusters joined by single noisy points).

- • **Complete Linkage (Maximum Distance):** d(A, B) = max { d(x, y) : x in A, y in B }. Finds compact, spherical clusters with equal diameters; sensitive to outliers.
- • **Average Linkage:** d(A, B) = (1 / (|A| * |B|)) * sum_{x in A} sum_{y in B} d(x, y). Balances single and complete linkage; robust to noise.
- • **Ward's Minimum Variance Method:** Merges the pair of clusters that minimizes the increase in total Within-Cluster Sum of Squares (variance). Highly popular for clean, spherical clusters.

# 4.3 Partitioning Methods: K-Means and K-Medoids

## 1. The K-Means Clustering Algorithm

K-Means partitions n observations into K clusters, where each point belongs to the cluster with the nearest mean (Cluster Centroid mu_k).

- **Objective Function (Inertia / Within-Cluster Sum of Squares - WCSS):** min J = sum_{k=1}^K sum_{x_i in C_k} ||x_i - mu_k||^2.
- **Step-by-Step Algorithm (Lloyd's Algorithm):**
  - 1. Initialization: Randomly select K initial centroids mu_1, mu_2, ..., mu_K from dataset (or use K-Means++ for smart spread initialization).
  - 2. Assignment Step: Assign each data point x_i to the nearest centroid: c_i = argmin_k ||x_i - mu_k||^2.
  - 3. Update Step: Recalculate each centroid as the arithmetic mean of all data points assigned to that cluster: mu_k = (1 / |C_k|) * sum_{x_i in C_k} x_i.
  - 4. Convergence: Repeat steps 2 and 3 until centroids stop moving (change in WCSS < threshold epsilon).

***Determining Optimal K: Elbow Method & Silhouette Analysis***

- **1. The Elbow Method:** Plot WCSS (Inertia) versus number of clusters K (from K=1 to 10). WCSS monotonically decreases as K increases. The optimal K is selected at the sharp 'Elbow Point' where the rate of decrease abruptly flattens out.
- **2. Silhouette Analysis (Silhouette Coefficient s(i)):** Measures how well-separated cluster boundaries are for sample i: s(i) = [ b(i) - a(i) ] / max( a(i), b(i) ) • a(i): Mean intra-cluster distance from sample i to all other points in its OWN cluster. • b(i): Mean nearest-cluster distance from sample i to points in the closest NEIGHBORING cluster. • Score Range: s(i) in [-1.0, +1.0]. Values near +1.0 indicate excellent clustering; near 0 indicate boundary ambiguity; negative values indicate misclustering.

## 2. K-Medoids Clustering (PAM — Partitioning Around Medoids)

In K-Means, centroids are artificial mathematical averages highly susceptible to being dragged away by extreme outliers. K-Medoids mandates that the center of each cluster MUST be an actual, physical data point (Medoid) from the dataset.

- **Advantage:** Substantially more robust to noise and outliers than K-Means; supports arbitrary non-Euclidean distance metrics. Limitation: Higher computational complexity O(K * (n - K)^2).

# 4.4 DBSCAN & Gaussian Mixture Models (GMM)

## 1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN defines clusters as continuous dense regions of points separated by regions of low density. Unlike K-Means, DBSCAN does NOT require specifying K in advance and naturally discovers arbitrary non-convex geometric shapes.

- **Two Core Hyperparameters:**
  - Epsilon (eps): The maximum radius distance defining a point's local neighborhood.
  - MinPts: The minimum number of points required within the eps-neighborhood to establish a dense core.
- **Classification of Data Points:**
  - • **Core Point:** Has at least MinPts within its eps-neighborhood: |N_eps(p)| >= MinPts.
  - • **Border Point:** Has fewer than MinPts within eps, but falls within the eps-neighborhood of a Core Point.
  - • **Noise Point (Outlier):** Neither a Core Point nor a Border Point (isolated in low-density space).
- **Advantages & Limitations:** Discovers complex non-linear clusters (rings, spirals); immune to outliers. Limitation: Struggles on datasets with widely varying local densities.

## 2. Gaussian Mixture Models (GMM) & Expectation-Maximization

GMM is a probabilistic soft-clustering method that models data as a weighted mixture of K multivariate Gaussian distributions: P(x) = sum_{k=1}^K pi_k * N(x | mu_k, Sigma_k), where pi_k is the mixture weight, mu_k is the mean vector, and Sigma_k is the covariance matrix.

- **The Expectation-Maximization (EM) Algorithm:**
  - E-Step (Expectation): Calculate the posterior probability (responsibility gamma_ik) that point x_i was generated by Gaussian component k.
  - M-Step (Maximization): Update parameters pi_k, mu_k, and Sigma_k using the soft responsibility weights to maximize the expected log-likelihood.

# 4.5 Ensemble Learning Techniques

**Definition: Ensemble Learning**

*Ensemble Learning is a paradigm in which multiple diverse base machine learning models (Weak Learners) are strategically combined to produce a single composite predictor (Strong Learner) that achieves superior predictive accuracy, robustness, and generalization than any individual constituent model.*

## 1. Basic Ensemble Techniques: Voting & Averaging

- **Max Voting (Majority Voting for Classification):**
  - Hard Voting: Predicts the class label that received the highest number of votes across base models: y_hat = mode( y_1, y_2, ..., y_M ).
  - Soft Voting: Computes the average predicted class probability across models and picks the argmax. Superior when base classifiers are well-calibrated.
- **Averaging & Weighted Averaging (for Regression):** y_hat = sum_{m=1}^M w_m * f_m(x), where weights w_m reflect individual model validation accuracy (sum(w_m) = 1).

## 2. Bagging & Random Forest (Variance Reduction)

- **Bagging (Bootstrap Aggregation - Breiman, 1996):** Draws B bootstrap subsets (sampled with replacement, containing ~63.2% unique original points) from training data. Trains B independent deep models in parallel. Predictions are averaged (regression) or voted (classification). Substantially reduces variance without increasing bias.
- **Random Forest:** Enhances bagging by adding Feature Subspace Sampling: At every split in every decision tree, only a random subset of m = sqrt(p) features is considered. This decorrelates individual trees, making the ensemble immune to dominant features!

## 3. Boosting (Bias Reduction) & Stacking

- **Boosting Concept:** Sequential ensemble where base models are trained iteratively. Each successive model focuses specifically on correcting the residual errors and misclassified instances of previous models by adjusting sample weights.
- **AdaBoost (Adaptive Boosting):** Increases instance weights for misclassified points. Computes model weight alpha_m = (1/2) * ln((1 - err_m) / err_m). Final prediction is weighted vote: y_hat = sign( sum( alpha_m * f_m(x) ) ).

- **Gradient Boosting (GBM / XGBoost):** Fits each new base tree directly to the Pseudo-Residuals (negative gradient of the loss function) of the existing ensemble: r_im = - [ d L(y_i, F(x_i)) / d F(x_i) ]. Updates model: F_m(x) = F_{m-1}(x) + learning_rate * h_m(x).
- **Stacking (Stacked Generalization):** Heterogeneous ensemble. Trains diverse Level-0 base models (e.g., SVM, KNN, Random Forest). Their out-of-fold predictions serve as input features to a Level-1 Meta-Classifier (e.g., Logistic Regression) that learns how to optimally combine predictions.

### Master Comparison Table: Bagging vs. Boosting vs. Stacking

| Comparison Parameter | Bagging (e.g., Random Forest) | Boosting (e.g., AdaBoost, XGBoost) | Stacking (Stacked Generalization) |
|---|---|---|---|
| Architecture & Execution | Parallel (Independent base models) | Sequential (Models depend on previous errors) | Two-Tier Hierarchical (Base + Meta Model) |
| Primary Error Targeted | Reduces VARIANCE (Overfitting control) | Reduces BIAS (Builds strong learner from weak) | Reduces both Bias and Variance |
| Base Learners Used | Homogeneous (Deep, high-variance trees) | Homogeneous (Shallow decision stumps / weak trees) | Heterogeneous (SVM, KNN, Trees, Neural Nets) |
| Data Sampling Method | Bootstrap random sampling with replacement | Weighted data sampling / Gradient residual fitting | K-Fold out-of-fold cross-validated meta-data |
| Aggregation Rule | Simple Majority Voting / Average | Weighted sum based on model stage performance | Meta-model regression/classification function |

## 4.6 High-Yield University Exam Question Bank

### Part A: Short Answer Questions (2 to 5 Marks)

***Q1. Explain the Elbow Method and Silhouette Analysis in K-Means. [4 Marks]***

- **Answer:** The Elbow Method plots WCSS vs K to find the inflection elbow point of diminishing returns. Silhouette Analysis measures separation: s(i) = (b(i)-a(i))/max(a(i),b(i)), where a(i) is intra-cluster distance and b(i) is nearest-cluster distance. A score near +1 indicates well-clustered instances.

***Q2. Differentiate between Core, Border, and Noise points in DBSCAN. [3 Marks]***

- **Answer:** Core Point: has >= MinPts within radius eps. Border Point: has < MinPts within eps but is neighbor to a Core point. Noise Point: has < MinPts and is not reachable from any Core point (outlier).

***Q3. Differentiate between Bagging and Boosting. [4 Marks]***

- **Answer:** Bagging trains independent models in parallel on bootstrap samples to reduce variance (e.g., Random Forest). Boosting trains models sequentially, adjusting weights on errors to reduce bias (e.g., AdaBoost, Gradient Boosting).

### Part B: Long Answer Questions (10 to 15 Marks Outline Guides)

- **Q4. Explain K-Means Clustering step-by-step with WCSS optimization formula, convergence criteria, advantages, and limitations. [10 Marks]**
- **Q5. Discuss Hierarchical Clustering (Agglomerative vs Divisive), Dendrogram interpretation, and linkage criteria (Single, Complete, Average, Ward's). [12 Marks]**
- **Q6. Explain DBSCAN clustering algorithm with epsilon, MinPts, density reachability concepts, and compare with K-Means. [10 Marks]**
- **Q7. Explain Advanced Ensemble Techniques: Random Forest feature subspace sampling, AdaBoost weight updates, Gradient Boosting residual fitting, and Stacking. [15 Marks]**

## 4.7 Unit Summary & Key Takeaways

- Clustering groups unlabeled data by maximizing intra-cluster similarity and minimizing inter-cluster similarity.
- Hierarchical clustering builds dendrogram trees using Agglomerative (bottom-up) or Divisive (top-down) with linkage metrics.
- K-Means minimizes WCSS; optimal K is evaluated via Elbow and Silhouette methods; K-Medoids uses actual points as centers.
- DBSCAN discovers arbitrary non-linear shapes based on density (eps, MinPts) and isolates noise outliers.
- GMM performs soft probabilistic clustering using Expectation-Maximization (EM).
- Ensemble learning combines weak learners: Bagging/Random Forest reduces variance in parallel; Boosting (AdaBoost/GBM) reduces bias sequentially; Stacking uses meta-learners.