GURU TEGH BAHADUR 4TH CENTENARY ENGINEERING COLLEGE

G-8 AREA, RAJOURI GARDEN, NEW DELHI-110064

MACHINE LEARNING

ASSIGNMENT – 02

Course Code: CIE-421P

Semester: 7th

Submitted to :

Ms. Arshi Kaur

Submitted by :

SAIMA

03423802722

CSE 1

UNIT – 1 : Introduction to Machine Learning

1. Define Machine Learning. Explain its types - supervised, unsupervised, semi-supervised, reinforcement (with examples).

Machine Learning (ML) is a field of artificial intelligence where systems learn from data, identify patterns, and make decisions with minimal human intervention. The primary goal is to enable computers to learn from experience (data).

Type	Definition	Key Feature	Example
Supervised Learning	Learns from a dataset where the **output is known** (labeled data). The model maps input to a known output.	Uses **Labeled Training Data**	Predicting house price based on size (Regression) or classifying an email as Spam/Not Spam (Classification).
Unsupervised Learning	Learns from a dataset where the **output is not known** (unlabeled data). The model finds hidden patterns or structures in the data.	Uses **Unlabeled Training Data**	Grouping customers with similar purchasing habits (Clustering) or reducing the number of variables (Dimensionality Reduction).
Semi-Supervised Learning	Uses a large amount of **unlabeled data** and a small amount of **labeled data** for training.	Mixes **Labeled and Unlabeled Data**	Training a model for face recognition where only a few faces are labeled.
Reinforcement Learning	The model (agent) learns to make decisions by performing actions in an **environment** to maximize a cumulative **reward**.	**Agent, Environment, Action, Reward**	Training a model to play chess or a self-driving car navigating a road network.

2. Differentiate between supervised and unsupervised learning.

Feature	Supervised Learning	Unsupervised Learning
Data Type	**Labeled** data (input-output pairs are provided).	**Unlabeled** data (only input data is provided).

Feature	Supervised Learning	Unsupervised Learning
Goal	**Predict** an outcome or **classify** data.	**Discover** patterns, structures, or groupings.
Algorithms	Linear Regression, Logistic Regression, SVM, Decision Trees.	K-Means, DBSCAN, PCA, Hierarchical Clustering.
Complexity	Generally simpler to evaluate due to clear targets.	More complex to evaluate, as there's no ground truth.

3. Explain the Bias-Variance trade-off. Why does it matter in model performance?

The Bias-Variance trade-off is a central concept in machine learning that concerns the relationship between a model's complexity and its error. Total Error is approximated as:

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

- **Bias (Bias^2):** The error introduced by approximating a real-world problem, which may be complicated, by a simplified model. **High Bias** suggests the model is too simple (**underfitting**) and consistently misses relevant relations between features and target output.
- **Variance (Variance):** The error introduced due to the model's sensitivity to small fluctuations in the training data. **High Variance** suggests the model is too complex (**overfitting**) and learns the noise in the training data too well.

Why it matters: The goal is to find the **"sweet spot"**—a model complexity that minimizes the total error by balancing low bias (to capture the underlying patterns) and low variance (to generalize well to new, unseen data).

4. Discuss Overfitting and Underfitting. Give prevention methods (regularization, cross-validation, pruning, etc.).

- **Overfitting:** Occurs when a model is **too complex** and learns the training data, including its noise and random fluctuations, *too well*. It performs exceptionally on the training data but **poorly on new, unseen data** (High Variance).
- **Underfitting:** Occurs when a model is **too simple** and cannot capture the underlying structure or patterns in the training data. It performs **poorly on both training and new data** (High Bias).

Prevention Method	Description
Regularization	Adds a penalty term (e.g., L1 - Lasso, L2 - Ridge) to the loss function to discourage overly complex models by reducing the magnitude of coefficients.
Cross-Validation	Splits the data into multiple folds and trains the model on different subsets, ensuring the model's performance is stable across various data splits.
Pruning (Decision Trees)	In decision trees, removing branches that have low importance or that model noise.
More Data	Increasing the size of the training dataset helps the model generalize better and reduces overfitting.
Feature Selection	Choosing only the most relevant features to prevent the model from learning from noise in irrelevant variables.

5. Explain Function Approximation in ML.

Function Approximation is the core concept underlying supervised machine learning. In mathematical terms, the goal is to find a function $f: X \to Y$ that maps input variables X to output variables Y by learning from the training data.

The ML algorithm builds a **model ($\hat{f}$)** that is an **approximation** of the true, unknown function (f).

- **Regression:** $\hat{f}$ is a function that outputs a continuous value.
- **Classification:** $\hat{f}$ is a function that outputs a discrete class label.

The process involves selecting a model (e.g., a neural network, a linear equation) and optimizing its parameters (weights and biases) to minimize the error between the model's predictions ($\hat{y} = \hat{f}(x)$) and the actual output (y).

6. Explain perspectives and key issues in Machine Learning.

Perspectives on ML

1. **Engineering Perspective:** ML is seen as a set of tools and algorithms used to build practical, deployable systems that solve real-world problems (e.g., spam detection, recommendation systems).
2. **Cognitive Science Perspective:** ML is used to understand the principles of intelligence and learning, often drawing inspiration from the human brain (e.g., neural networks).

3. **Mathematical/Statistical Perspective:** ML is viewed as a discipline focused on building mathematical models and statistical inferences from data, often involving concepts like probability, optimization, and hypothesis testing.

Key Issues in ML

- **Data Quality/Quantity:** ML models are highly dependent on the quality and size of the training data. Issues include missing values, noisy data, and insufficient samples.
- **Overfitting/Underfitting:** The perpetual challenge of balancing model complexity to generalize well to new data (addressed in Q4).
- **Bias and Fairness:** Models can inherit and amplify biases present in the training data, leading to unfair or discriminatory outcomes against certain groups.
- **Interpretability/Explainability (XAI):** Complex models (like deep neural networks) can act as "black boxes." Understanding *why* a model made a specific prediction is crucial for trust and debugging.
- **Scalability:** Developing algorithms and systems that can handle and process massive, ever-growing datasets efficiently.

UNIT - 2 : Regression Analysis

7. Explain Simple Linear Regression. Write its equation, assumptions, and explain how coefficients are estimated.

Simple Linear Regression (SLR) is a statistical method used to model the relationship between a **single independent variable (X)** and a **single dependent variable (Y)** by fitting a straight line to the observed data.

Equation

The equation of the straight line is:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Where:

- Y: Dependent variable (the value being predicted).
- X: Independent variable (the predictor).
- β_0: **Y-intercept** (the value of Y when $X=0$).
- β_1: **Slope** (the change in Y for a one-unit change in X).
- ϵ: The **error term** (or residual), which accounts for the variability in Y that cannot be explained by X.

Assumptions (Gauss-Markov Assumptions)

1. **Linearity:** The relationship between X and Y is linear.
2. **Independence of Errors:** The error terms (ϵ) are independent of each other.
3. **Homoscedasticity:** The variance of the errors is constant across all levels of X.
4. **Normality of Errors:** The error term is normally distributed.

5. **No measurement error** in X.

Coefficient Estimation (Ordinary Least Squares - OLS)

The coefficients (β_0 and β_1) are typically estimated using the Ordinary Least Squares (OLS) method. OLS aims to find the line that minimizes the sum of the squared differences (or residuals) between the actual Y values and the Y values predicted by the line ($\hat{Y}$).

The estimated coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$ are:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where $\bar{x}$ and $\bar{y}$ are the means of X and Y.

8. Differentiate between Simple and Multiple Linear Regression.

Feature	**Simple Linear Regression (SLR)**	**Multiple Linear Regression (MLR)**
Independent Variables	**One** independent variable (X).	**Two or more** independent variables ($X_1, X_2, \ldots, X_k$).
Equation	$Y = \beta_0 + \beta_1 X + \epsilon$	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \epsilon$
Relationship	Models the relationship between Y and a single X.	Models the relationship between Y and a **set of X's**.
Interpretation	β_1 is the change in Y for a one-unit change in X.	β_i is the change in Y for a one-unit change in X_i, **holding all other X variables constant** (ceteris paribus).

9. Explain Ordinary Least Squares (OLS) and discuss R^2, Standard Error, F-statistic, and p-values.

Ordinary Least Squares (OLS)

As mentioned in Q7, **OLS** is a method for estimating the unknown parameters (coefficients) in a linear regression model. It works by minimizing the **Sum of Squared Errors (SSE)**, which is the sum of the squared differences between the observed values and the values predicted by the model.

Model Evaluation Metrics

Metric	Explanation
R-squared (R^2)	The **coefficient of determination**. It represents the proportion of the variance in the dependent variable (Y) that is predictable from the independent variables (X). R^2 ranges from 0 to 1. A higher value means a better fit.
Standard Error (SE) of the Regression	Also called **Root Mean Squared Error (RMSE)**. It is an absolute measure of the average distance that the observed data points fall from the regression line. It's expressed in the units of the dependent variable (Y).
F-statistic	Tests the **overall significance** of the model. It determines if at least one of the independent variables significantly contributes to the prediction of Y. A large F-statistic and a small corresponding p-value indicate the model is statistically significant.
p-values (for individual coefficients)	Assesses the **statistical significance of each individual predictor** (X_i). A small p-value (typically < 0.05) for a coefficient indicates that the predictor variable is likely a meaningful addition to the model, as its coefficient is significantly different from zero.

10. Explain Logistic Regression. Why is it used for classification?

Logistic Regression is a statistical model used for **classification** tasks, despite its name. It models the probability that an instance belongs to a certain class.

How it works:

1. It first calculates a **linear combination** of the input features (similar to linear regression): $\text{Score} = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$.
2. It then uses the Sigmoid (or Logistic) function to transform this linear score into a probability P that the output is of a specific class (usually $P \in [0, 1]$).

$$\text{Probability} = P(Y=1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_i X_i)}}$$

3. Finally, a **threshold** (usually 0.5) is applied to the probability: if $P \geq 0.5$, classify as Class 1; otherwise, classify as Class 0.

Why it's used for Classification:

Standard linear regression outputs a continuous value $(-\infty, \infty)$, which is not suitable for probability or class prediction. The **Sigmoid function** in Logistic Regression constrains the output to the range **$[0, 1]$**, which is a perfect fit for modeling probability and binary classification (two classes).

11. Discuss Feature Selection and Dimensionality Reduction techniques - PCA and LDA.

Both **Feature Selection** and **Dimensionality Reduction** aim to reduce the number of features (variables) in a dataset, which helps simplify models, speed up training, and reduce overfitting.

1. Feature Selection

The process of selecting a **subset of relevant features** from the original set. It **keeps the original features** and discards irrelevant or redundant ones.

- *Methods:* Filter methods (e.g., correlation), Wrapper methods (e.g., recursive feature elimination), Embedded methods (e.g., Lasso regression).

2. Dimensionality Reduction

The process of transforming the data into a **lower-dimensional space** by creating **new, combined features** (components) that capture most of the original information. It **does not keep the original features**.

Technique	Acronym	Description	Goal/Basis
Principal Component Analysis	**PCA** (Unsupervised)	Finds the directions (principal components) that maximize the **variance** in the data. Projects the data onto a lower-dimensional subspace defined by these components.	**Maximizes variance**; best for data compression and visualization.
Linear Discriminant Analysis	**LDA** (Supervised)	Finds the directions (linear discriminants) that **maximize the separation** between classes. Projects the data onto a lower-dimensional subspace to improve class separation.	**Maximizes class separability**; only used for classification problems.

UNIT - 3 : Classification Algorithms

12. What is Classification? Describe its general approach and steps.

Classification is a supervised machine learning task where the model learns to assign a **class label** (or category) to a new, unseen input instance based on patterns learned from labeled training data. The output is **discrete** (e.g., *Yes/No*, *Cat/Dog*, *Disease A/Disease B*).

General Approach and Steps:

1. **Data Collection & Preprocessing:** Gather the labeled data. Clean it by handling missing values, outliers, and preparing features (e.g., normalization, encoding categorical variables).
2. **Feature Engineering:** Select, create, and transform features to maximize the model's predictive power.

3. **Split Data:** Divide the dataset into **Training Set** (to train the model) and **Testing Set** (to evaluate the model's performance on unseen data).
4. **Model Selection & Training:** Choose a suitable classification algorithm (e.g., k-NN, SVM) and train it on the Training Set. The model learns the mapping from input features (X) to output class labels (Y).
5. **Model Evaluation:** Use the Testing Set to measure the model's performance using metrics like Accuracy, Precision, Recall, F1-Score, etc.
6. **Hyperparameter Tuning:** Optimize the model's internal settings (hyperparameters) to improve performance and generalize better.
7. **Deployment:** Deploy the final, optimal model to make predictions on real-world, new data.

13. Explain k-Nearest Neighbor (k-NN) algorithm with a worked example.

The **k-Nearest Neighbor (k-NN)** algorithm is a non-parametric, lazy learning classification method. It does not learn a fixed model during training; instead, it memorizes the entire training dataset.

Algorithm Steps:

1. **Choose k:** Select the number of neighbors (k).
2. **Calculate Distance:** For a new data point, calculate its distance (e.g., Euclidean distance) to **all** points in the training set.
3. **Find Neighbors:** Select the k data points that are closest to the new point (the k nearest neighbors).
4. **Vote for Class:** Assign the new data point the class label that is **most frequent** among its k neighbors (majority vote).

Worked Example:

Suppose we have a 2D dataset of 'Apple' and 'Banana' data points. A new point (X) arrives, and we set **$k=3$**.

1. **Calculate Distances:** Find the distance from X to all existing points.
2. **Identify 3 Neighbors:** The 3 closest points are found: **2 Apples** and **1 Banana**.
3. **Classify:** By majority vote, the new point **X is classified as 'Apple'**.

14. Explain Support Vector Machine (SVM). Discuss kernel trick and hyperplane.

Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression, primarily for classification.

- **Hyperplane:** In an SVM, the goal is to find the optimal **hyperplane** that separates the data points of different classes in the feature space. In a 2D space, the hyperplane is a line; in 3D, it's a plane, and in higher dimensions, it's an $n-1$ dimensional subspace. The **optimal hyperplane** is the one that

has the **maximum margin** (distance) to the nearest training data points of any class, called the **Support Vectors**.

- **Support Vectors:** These are the data points closest to the optimal hyperplane. They are the critical elements of the dataset, as they directly influence the position and orientation of the hyperplane.

Kernel Trick

The **Kernel Trick** is a technique used to handle **non-linearly separable data**.

- If the data cannot be separated by a straight line (linear hyperplane) in its original low-dimensional space, the Kernel Trick implicitly maps the data into a **higher-dimensional space** where a linear separation (hyperplane) *is* possible.
- Crucially, it does this **without explicitly calculating the coordinates** in the high-dimensional space. It uses a **kernel function** (e.g., Polynomial, Radial Basis Function/RBF) that calculates the dot product between the data points in the high-dimensional space directly from the original coordinates, making the computation feasible.

15. Explain Decision Tree Algorithm (ID3 / CART). Define Entropy and Information Gain.

A **Decision Tree** is a non-parametric supervised learning method used for both classification and regression. It builds a model in the structure of a tree, where:

- **Internal nodes** represent a feature test (e.g., "Is age ≥ 30?").
- **Branches** represent the outcome of the test.
- **Leaf nodes** represent the final class label or a numerical value (prediction).

Algorithms:

- **ID3 (Iterative Dichotomiser 3):** Uses **Information Gain** to select the splitting feature. Best suited for categorical features and multi-way splits.
- **CART (Classification and Regression Trees):** Uses the **Gini Index** (for classification) or Squared Error (for regression) to select the splitting feature. Only allows binary splits.

Key Concepts:

1. Entropy: A measure of the impurity or randomness in a set of examples. High entropy means the set is mixed (e.g., 50% Class A, 50% Class B). Low entropy (or zero) means the set is pure (e.g., 100% Class A). The goal of the decision tree is to minimize entropy at the leaf nodes.

$$\text{Entropy}(S) = - \sum_{i=1}^{c} P_i \log_2(P_i)$$

where P_i is the proportion of data belonging to class i.

2. Information Gain (IG): The expected reduction in entropy achieved by splitting the data based on a particular feature. The algorithm selects the feature with the highest Information Gain to be the split node, as this feature best separates the data into purer subsets.

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

16. Explain Naïve Bayes Classifier. State its assumptions and give one example.

The **Naïve Bayes Classifier** is a collection of simple probabilistic classification algorithms based on **Bayes' Theorem** with a strong, or "naïve," independence assumption.

Bayes' Theorem

The classifier calculates the probability of a class Y, given a set of features $X = \{x_1, x_2, \ldots, x_n\}$:

$$P(Y|X) = \frac{P(X|Y) P(Y)}{P(X)}$$

The classifier then predicts the class that has the highest posterior probability $P(Y|X)$.

Naïve Assumption

The core assumption is that all features (x_i) are conditionally independent of each other, given the class Y. That is:

$$P(X|Y) = P(x_1, x_2, \ldots, x_n|Y) = P(x_1|Y) \times P(x_2|Y) \times \ldots \times P(x_n|Y)$$

While this assumption is rarely true in the real world, Naïve Bayes often performs surprisingly well, especially for large datasets.

Example: Spam Filtering

- **Goal:** Classify an email as **Spam** or **Not Spam**.
- **Features (X):** The presence or absence of certain words in the email (e.g., x_1= 'free', x_2= 'money', x_3= 'dear').
- **Model:** The classifier learns the probability of each word appearing, given the email is Spam, $P(\text{'free' | Spam})$, and given it is Not Spam, $P(\text{'free' | Not Spam})$.
- **Prediction:** For a new email, it combines the probabilities of all the words appearing, using the Naïve assumption, to calculate the overall probability $P(\text{Spam} | \text{Email})$ and $P(\text{Not Spam} | \text{Email})$, and selects the higher one.

17. Explain Ensemble Learning - Bagging, Boosting, AdaBoost, Random Forest.

Ensemble Learning is a technique that combines the predictions of **multiple base learners (or models)** to produce a single, superior prediction. The main principle is that a group of "weak learners" can form a "strong learner" when aggregated.

Technique	**Method**	**Goal**	**Core Algorithm Example**
Bagging	**Bootstrap Aggregating:** Trains **multiple base models** (often of the same type, like Decision Trees) on different **bootstrap samples** (random samples with replacement) of the training data. The final prediction is by averaging	**Reduces Variance** and helps prevent overfitting.	**Random Forest** (see below)

Technique	Method	Goal	Core Algorithm Example
	(regression) or majority vote (classification).		
Boosting	Sequentially trains **multiple weak learners**, where each subsequent model focuses on the instances that the previous models **misclassified** or predicted poorly. Weights are assigned to data points and updated in each iteration.	**Reduces Bias** and converts weak learners into strong ones.	**AdaBoost** (see below), Gradient Boosting
AdaBoost (Adaptive Boosting)	An early, popular boosting algorithm. It initially assigns equal weights to all data points. In each iteration, it: 1) Trains a weak learner, 2) **Increases the weights** of the misclassified data points, and 3) **Decreases the weights** of correctly classified points. Finally, it combines the weighted votes of all weak learners.	Focuses on **hard-to-classify** data points.	N/A
Random Forest	An extension of **Bagging** specifically for Decision Trees. It trains multiple Decision Trees using a random subset of the training data *and* a **random subset of features** for determining the best split at each node.	**Reduces Variance** significantly by decorrelating the trees. Excellent for high accuracy and robustness.	N/A

18. Discuss Model Evaluation Metrics: Precision, Recall, F1-Score, ROC Curve, Sensitivity & Specificity.

These metrics are crucial for evaluating the performance of a **classification model**, especially in binary classification. They are based on the **Confusion Matrix** (True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN)).

Metric	Formula	Description	When to Use
Precision	$\text{Precision} = \frac{TP}{TP + FP}$	Of all the instances the model **predicted positive**, how many were **actually positive**?	E.g., Spam detection: When the cost of a FP (marking a good email as spam) is high.

Metric	Formula	Description	When to Use
		(Minimizing False Positives)	
Recall (Sensitivity)	$\text{Recall} = \frac{TP}{TP + FN}$	Of all the instances that were **actually positive**, how many did the model **correctly predict positive**? (Minimizing False Negatives)	E.g., Disease detection: When the cost of a FN (missing a disease) is high.
F1-Score	$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	The **harmonic mean** of Precision and Recall. It provides a single score that balances both metrics.	When you need a balance between Precision and Recall, especially with imbalanced classes.
Sensitivity	$\text{Sensitivity} = \text{Recall}$	The same as Recall (True Positive Rate).	In medical/technical contexts.
Specificity	$\text{Specificity} = \frac{TN}{TN + FP}$	Of all the instances that were **actually negative**, how many did the model **correctly predict negative**? (True Negative Rate)	When correctly identifying negatives is important.
ROC Curve	**Receiver Operating Characteristic Curve**. A plot of the **True Positive Rate (Sensitivity/Recall)** against the **False Positive Rate (1 - Specificity)** at various threshold settings.	Used to visualize the performance of a binary classifier across all possible classification thresholds. The **Area Under the Curve (AUC)** measures the overall performance; a higher AUC is better.	When evaluating model performance across different thresholds.

UNIT – 4 : Clustering Methods

19. Explain K-Means Clustering Algorithm with example and steps.

K-Means Clustering is a popular, simple, and effective **unsupervised learning** algorithm used to partition n data points into K distinct, non-overlapping clusters.

Goal:

To partition the data such that the total **within-cluster sum of squares (WCSS)** is minimized. This means points within a cluster are as similar as possible (close to their centroid), and clusters are as far apart as possible.

Algorithm Steps:

1. **Initialization: Choose the number of clusters, K**. Randomly select K data points from the dataset as the initial **centroids** (center points of the clusters).
2. **Assignment Step:** Assign **each data point** to the **nearest centroid** (based on distance, usually Euclidean distance). This forms K initial clusters.
3. **Update Step: Recalculate the centroid** of each cluster by taking the **mean** of all data points assigned to that cluster.
4. **Iteration/Convergence:** Repeat the Assignment and Update steps iteratively until the centroids no longer change significantly (convergence) or the maximum number of iterations is reached.

Example: Customer Segmentation

- **Goal:** Group customers based on their purchase history (e.g., features are 'Annual Income' and 'Spending Score').
- **Process:** If $K=3$ is chosen, the algorithm groups the customers into 3 distinct segments (e.g., 'High Value', 'Mid Value', 'Low Value'), allowing the company to tailor marketing strategies for each group.
****

20. Explain DBSCAN Algorithm and how it differs from K-Means.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an **unsupervised** clustering algorithm that groups together data points that are closely packed (high density), marking as **outliers** points that lie alone in low-density regions.

Key Concepts:

1. **Epsilon (ϵ) / MaxDist:** The radius defining the neighborhood around a data point.
2. **MinPts:** The minimum number of data points required to form a dense region (a cluster).
3. **Core Point:** A point having at least MinPts (including itself) within its ϵ-neighborhood.
4. **Border Point:** A point that is within the ϵ-neighborhood of a Core Point but has fewer than MinPts in its own ϵ-neighborhood.
5. **Noise Point (Outlier):** A point that is neither a Core Point nor a Border Point.

Difference from K-Means:

Feature	DBSCAN	K-Means
Number of Clusters (K)	**Does not require** specifying K upfront. Discovers clusters based on density.	**Requires** specifying K (number of clusters) before running.
Cluster Shape	Can find clusters of **arbitrary shapes** (non-spherical).	Assumes and works best with **spherical** (convex) clusters.
Handling Outliers	Explicitly identifies and handles **noise (outliers)**.	Every point is forced into a cluster, making it **sensitive to outliers**.
Density	**Density-based**: Clusters are defined as dense regions separated by low-density areas.	**Centroid-based**: Clusters are defined by their mean position (centroid).

21. Explain Hierarchical Clustering (Agglomerative & Divisive) and draw a Dendrogram.

Hierarchical Clustering is an **unsupervised** method that builds a hierarchy of clusters, represented as a **Dendrogram**. It can be performed in two main ways:

1. Agglomerative (Bottom-Up)

- Starts with **each data point as its own cluster**.
- In each step, the **two closest clusters are merged** into a new cluster.
- This continues until only **one single cluster** remains (the root of the tree).
- *Linkage Methods:* Used to define the "distance" between two clusters (e.g., Single, Complete, Average linkage).

2. Divisive (Top-Down)

- Starts with **all data points in one single cluster**.
- In each step, the single cluster is **split into two sub-clusters** based on a dissimilarity measure.
- This continues until every data point is in its own cluster (the leaves of the tree).

Dendrogram

A **Dendrogram** is a tree-like diagram that records the sequence of merges or splits. The height of the join in the dendrogram represents the **distance or dissimilarity** between the clusters being merged (Agglomerative) or split (Divisive). Cutting the dendrogram horizontally at a chosen distance/height determines the final clusters.

22. Describe Cluster Validity/ Measuring Clustering Goodness.

Cluster Validity refers to the process of evaluating the results of a clustering algorithm to determine how well the clusters fit the data and to compare different clustering results. Since clustering is an unsupervised task, there is no true 'ground truth' for comparison, leading to different evaluation approaches:

1. External Measures (Supervised)

Used when the **true class labels** are known (for testing/validation purposes). They measure the agreement between the clustering result and the known labels.

- **Purity:** Measures the extent to which each cluster contains data points of mostly a single class.
- **Rand Index / Adjusted Rand Index:** Measures the similarity between the ground truth and the clustering results by considering pairs of points.

2. Internal Measures (Unsupervised)

Used when the **true class labels are unknown**. They evaluate the quality of the clustering based on the data used for clustering itself.

- **Cohesion (Intra-cluster similarity):** How closely related the objects within a cluster are (e.g., minimizing WCSS in K-Means).
- **Separation (Inter-cluster dissimilarity):** How distinct or well-separated the clusters are from each other.
- **Silhouette Score:** Measures how similar a data point is to its own cluster compared to other clusters. It ranges from -1 (bad clustering) to $+1$ (dense, well-separated clustering).
- **Davies-Bouldin Index:** A ratio of within-cluster distance to between-cluster distance. A lower value indicates better clustering.