Times asked: 6 times
5 times
4 times
3 times
2 times
1 time
# indicates 5-mark question

Machine Learning Question bank

1. Introduction to Machine Learning

1. Explain issues in Machine Learning.
2. Explain the steps of developing Machine Learning applications.
3. Explain the terms overfitting, underfitting, bias & variance trade-off w.r.t Machine Learning.
4. Explain how to choose the right algorithm for an ML application. #
5. Explain any five applications of Machine Learning.

2. Learning with Regression and Trees

6. Explain performance evaluation metrics for classification with suitable examples.
7. Problem on Gini Index & Decision Tree Construction, PYQ:

   Suppose we want Gini index to decide whether the car will be stolen or not. The target classification is "car is stolen?" which can be Yes or No, create a decision tree for the given data below:

Car no	Colour	Type	Origin	Stolen ?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes

Car no	Colour	Type	Origin	Stolen ?
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

8. Explain Logistic Regression. #
9. Explain Regression line, Scatter plot, Error in prediction and Best fitting line. #
10. Problem on Linear Regression.

    Following table shows the midterm and final exam grades obtained for students in a database course. Use linear regression to predict the final exam grade of a student who received 86 in the midterm exam.

Midterm exam (X)	72	50	81	74	94	86	59	83	86	33	88	81
Final exam (Y)	84	53	77	78	90	75	49	79	77	52	74	90

11. Explain Linear Regression with an example.

3. Ensemble Learning

12. Explain the Random Forest algorithm in detail.
13. Explain different ways to combine classifiers.
14. Explain the necessity of cross validation in ML applications and K-fold cross validation in detail.

4. Learning with Classification

15. Describe Multiclass classification.
16. Explain the concept of margin and support vectors in Support Vector Machines (SVM). Define related terms: hyperplane, hard margin, soft margin, and kernel.
17. Explain support vector machine as a constrained optimization problem.
18. Write a detailed not on SVM Kernel trick.

5. Learning with Clustering

19. Write a detailed note on the DBSCAN algorithm with a suitable example.
20. Demonstrate Minimal Spanning Tree (MST) algorithm for clustering with a suitable example.
21. Explain the Expectation-Maximization (EM) algorithm in detail.

6. Dimensionality Reduction

22. Write a detailed note on Principal Component Analysis for Dimension Reduction.
23. Write a detailed note on Linear Discriminant Analysis for Dimension Reduction.

	1	2	3	4	5	6
2025 May	15	45	20	15	20	20
2024 Dec	35	30	20	20	20	10
2024 May	15	30	25	20	20	15
2023 Dec	15	30	20	35	20	15
2023 May	15	40	20	10	25	25
2022 Dec	15	40	20	15	25	20
Estimate	**15**	**30-40**	**20**	**20**	**20**	**15-20**
Total	110	215	125	115	130	105

Asked once:

# indicates 5-mark question

1. Introduction to Machine Learning

1. Explain Training error and Generalization error. #
2. Differentiate between Supervised and Unsupervised Learning. #

2. Learning with Regression and Trees

3. Demonstrate CART method along with an example.
4. Explain Gini index along with an example. #
5. Explain performance evaluation measures for regression. #
6. Differentiate between Linear regression and Logistic regression. #
7. Explain Multivariate Linear regression method.

3. Ensemble Learning

8. Write detailed note on XGBoost ensemble method.
9. Compare Bagging and Boosting with reference to ensemble learning. Explain how these methods help to improve the performance of the machine learning model.

4. Learning with Classification

10. Consider the use case of Email spam detection. Identify and explain the suitable machine learning technique for this task.
11. Differentiate between Logistic regression and Support vector machine. #

5. Learning with Clustering

12. Explain K-means algorithm. #
13. Explain the distance metrics used in clustering. #

6. Dimensionality Reduction

14. What is dimensionality reduction? Explain how it can be utilized for classification and clustering task in Machine Learning. #
15. Explain the concept of feature selection and extraction. #
16. Find SVD for A = $\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$
17. Compute the Linear Discriminant projection for the following two-dimensional dataset.
    X1 = (x1, x2) = { (4,1), (2,4), (2,3), (3,6), (4,4) } and
    X2 = (x1, x2) = { (9,10), (6,8), (9,5), (8,7), (10,8) }

Machine Learning Answer bank

# indicates 5-mark question

1. Introduction to Machine Learning

1. Explain issues in Machine Learning.

1. Data Collection

- Gathering sufficient and relevant data is often difficult due to privacy, accessibility, and cost constraints.
- Poor collection methods can lead to biased or incomplete datasets.

2. Insufficient Data / Labelled Data

- Supervised learning needs large amounts of labelled data, which is expensive and time-consuming to obtain.
- Small datasets cause poor generalization and unstable model predictions.

3. Non-representative Data

- Training data must reflect real-world diversity; biased data leads to unfair or inaccurate predictions.
- Example: A model trained mainly on one demographic group may fail for others.

4. Poor Quality Data

- Missing values, noise, or incorrect labels reduce model accuracy.
- Requires preprocessing steps like cleaning, normalization, and validation.

5. Irrelevant Features

- Unrelated or redundant features add noise and increase computational cost.
- Feature selection helps retain only the most relevant inputs.

6. Underfitting

- Occurs when the model is trained with very little data, producing incomplete and inaccurate results.
- The model is too simple to understand the base structure of the data.

7. Overfitting

- You train a model on one dataset, and it gives good results, but when applied to another dataset, it produces inaccurate results.
- This happens because the model learns details specific to the training data and fails to generalize.

8. Software Integration

- Accurate models may fail if they cannot integrate with existing software.
- API compatibility and system limitations are common challenges.

9. Offline Learning / Deployment

- Models trained offline may become outdated due to changes in real-world data.

10. Cost Involved

- Hidden costs include server deployment, storage, and maintenance.
- Cloud infrastructure costs may scale rapidly with model usage.

2. Explain the steps of developing Machine Learning applications.

1. **Problem Definition**
   - Clearly define the objective of the ML application — what task needs automation or prediction (e.g., spam detection, price prediction).
   - This ensures the right data and algorithm are chosen.
2. **Data Collection**
   - Collect relevant raw data from files, sensors, APIs, or databases, as its quality and quantity directly affect model performance.
3. **Data Understanding and Analysis**
   - Analyze the dataset to understand its structure, feature types, and relationships.
   - Check for imbalances, missing values, outliers, or noise in data.
4. **Data Visualization**
   - Use charts and plots (scatter plots, histograms, heatmaps) to visually explore trends, correlations, and key influencing patterns.
5. **Data Preprocessing**
   - Clean the data by handling missing values, encoding categories, and scaling features.
   - Ensures the data is in a suitable format for the ML algorithm.
6. **Feature Selection and Extraction**
   - Select the most important features that influence the output or derive new meaningful ones. This improves model accuracy and reduces complexity.
7. **Model Selection**
   - Choose an appropriate ML algorithm (e.g., Decision Tree, SVM, Logistic Regression) based on the problem type — classification, regression, or clustering.
8. **Model Training and Testing**
   - Train the model on the training dataset to learn patterns, and evaluate it on the test dataset to check performance on unseen data. Helps detect overfitting or underfitting.
9. **Model Evaluation**
   - Assess the model using metrics such as accuracy, precision, recall based on the task type.
10. **Deployment and Optimization**
   - Deploy the final model into a real-world environment (e.g., web app, API) for practical use.
   - Continuously monitor its performance and optimize it using techniques like hyperparameter tuning, better features, or retraining with new data.

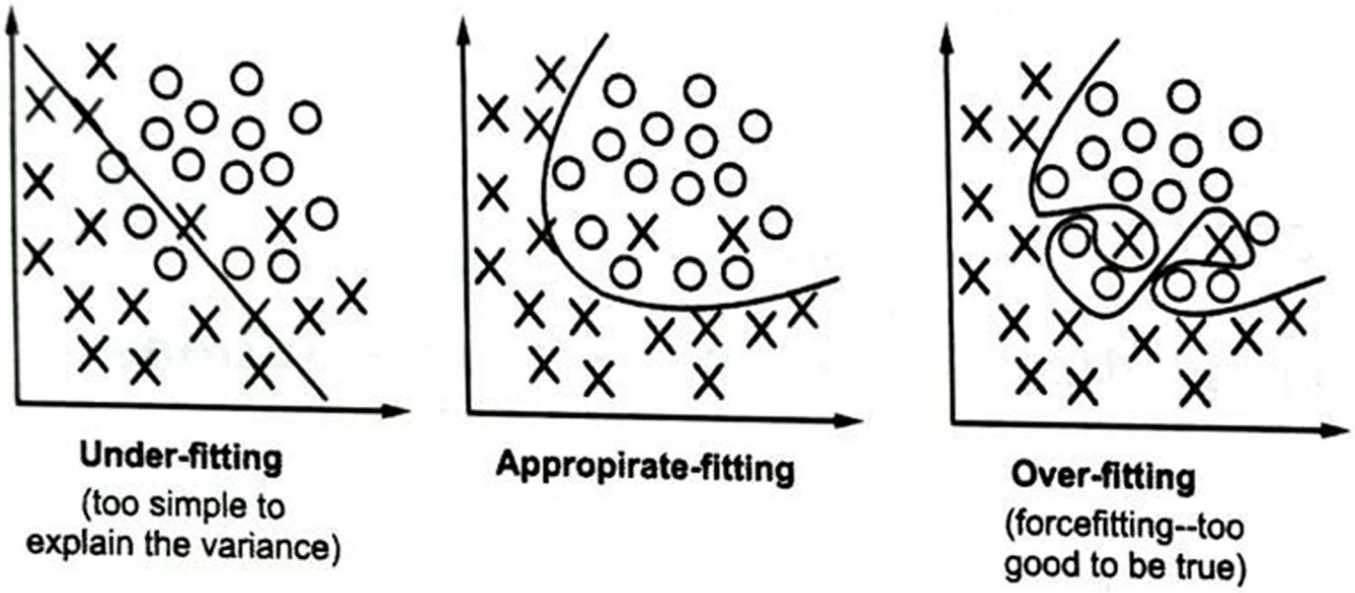
3. Explain the terms overfitting, underfitting, bias & variance trade-off w.r.t Machine Learning.

Underfitting:

- Underfitting occurs when the model is too simple to capture the underlying patterns of the data.
- It happens when the model is trained with very little data or lacks sufficient complexity, producing incomplete and inaccurate results.
- The model performs poorly on both training and test sets.

Overfitting:

- Overfitting occurs when a model performs very well on training data but poorly on new, unseen data.
- You train a model on one dataset, and it gives good results, but when applied to another dataset, it produces inaccurate results.
- This happens because the model learns not just the patterns but also the noise and details specific to the training data, failing to generalize.



Bias:

- Bias refers to the error caused by simplifying assumptions made by the model.
- High bias leads to underfitting because the model ignores important data patterns.

Variance:

- Variance refers to the model's sensitivity to small changes in the training data.
- High variance leads to overfitting because the model learns random noise instead of true patterns.

Bias–Variance Trade-off:

- The total error of a model = Bias^2 + Variance + Irreducible error.
- Reducing bias often increases variance, and vice versa.
- The goal is to find the optimal balance between bias and variance to achieve the best generalization performance.

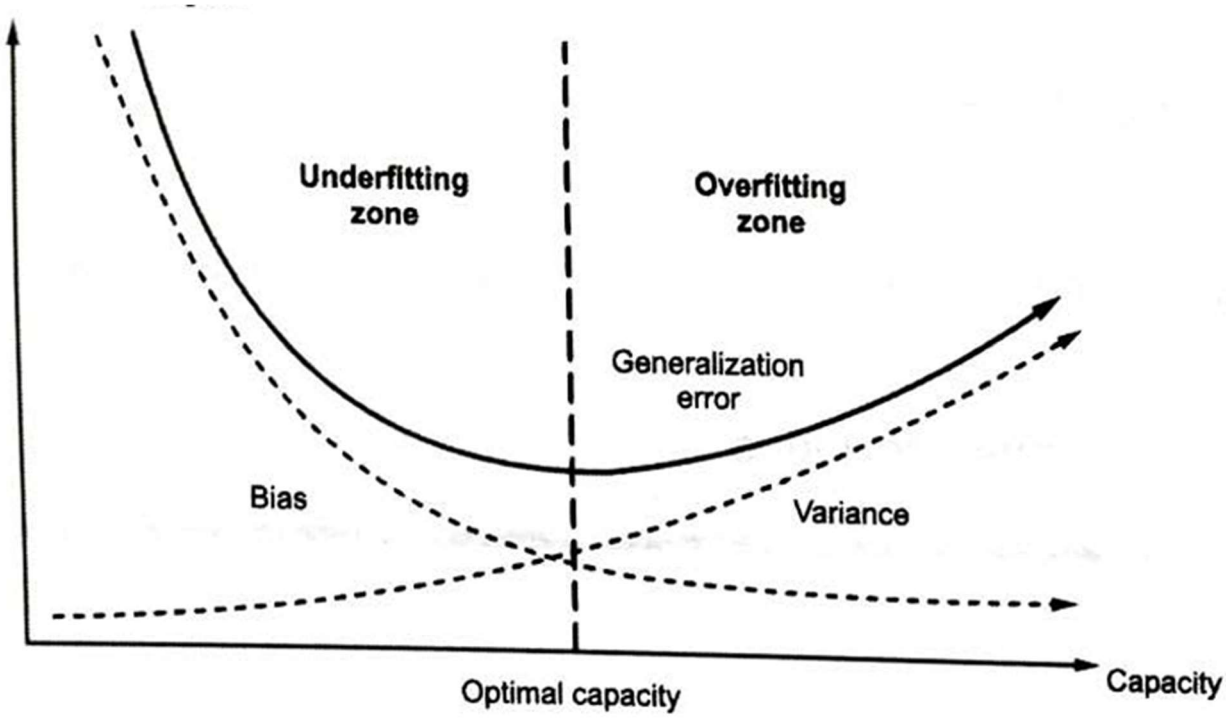


Fig. 1.9.2 : Bias variance Tradeoff as a function of model capacity

4. Explain how to choose the right algorithm for an ML application. #

To choose the right algorithm, we mainly consider goal and data:

1. **Goal:**
    - If the task involves predicting or forecasting a target value, use Supervised Learning.
        - If the target value is discrete (e.g. Yes/No, A/B/C) → use Classification.
        - If the target value is continuous (e.g. 0–100, -99–99) → use Regression.
    - If there is no target value, use Unsupervised Learning.
        - If the aim is to group data into categories → use Clustering.
        - If the aim is to find data distribution → use Density Estimation.
2. **Data:**
    - Check if features are continuous or categorical, and handle missing values or outliers properly.
    - These properties help narrow down which algorithm suits your dataset.

Learning Type	**Discrete Target**	**Continuous Target**
Supervised Learning	Classification	Regression
Unsupervised Learning	Clustering	Density Estimation

5. Explain any five applications of Machine Learning.

1. **Spam Detection** – Machine Learning algorithms analyze the content, sender information, and user behaviour to automatically classify emails or messages as spam or legitimate, helping users manage their inbox efficiently.
2. **Image & Speech Recognition** – ML models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), enable systems to recognize faces, handwriting, and spoken words, powering applications like facial authentication and voice assistants such as Siri or Alexa.
3. **Recommendation Systems** – ML predicts user preferences by analyzing past activity, purchase history, and behaviour of similar users, allowing platforms like Netflix, Amazon, and YouTube to provide personalized suggestions for movies, products, or music.
4. **Fraud Detection** – ML detects unusual patterns and anomalies in banking transactions, credit card usage, or online payments, helping financial institutions prevent fraud and secure customer accounts in real time.
5. **Self-Driving Cars** – ML models help autonomous vehicles detect lanes, obstacles, traffic signs, and pedestrians, enabling safe navigation and decision-making in complex and dynamic driving environments.
6. **Natural Language Processing (NLP)** – ML powers chatbots, sentiment analysis, language translation, and text summarization, enabling computers to understand, interpret, and respond to human language effectively.

2. Learning with Regression and Trees

6. Explain performance evaluation metrics for classification with suitable examples.

Key Terms:

- **TP (True Positive):** Correctly predicted positive cases (e.g., spam predicted as spam).
- **TN (True Negative):** Correctly predicted negative cases (e.g., not-spam predicted as not-spam).
- **FP (False Positive):** Incorrectly predicted positive cases (e.g., not-spam predicted as spam).
- **FN (False Negative):** Incorrectly predicted negative cases (e.g., spam predicted as not-spam).

1. **Confusion Matrix** – A table showing the actual vs predicted classifications. It helps visualize all types of correct and incorrect predictions.

Actual \ Predicted	Positive	Negative
Positive	TP=45	FN=15
Negative	FP=5	TN=35

2. **Accuracy** – Measures the overall correctness of the model.

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

Example: If a model correctly classifies 80 out of 100 emails as spam or not, accuracy = 80%.

3. **Precision** – Measures how many predicted positive cases are actually positive.

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

Example: Out of 50 emails predicted as spam, if 45 are actually spam, precision = 45/50 = 90%.

4. **Sensitivity (Recall)** – Measures how many actual positive cases are correctly identified.

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

Example: If there are 60 actual spam emails and the model detects 45, recall = 45/60 = 75%.

5. **Specificity** – Measures how many actual negative cases are correctly identified.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Example: Out of 40 non-spam emails, if 35 are correctly identified, specificity = 35/40 = 87.5%.

6. **F1 Score (F-Measure)** – Harmonic mean of precision and recall, balancing both metrics.

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

   Example: Precision = 90%, Recall = 75% → F1 ≈ 81.8%.

7. **Kappa Statistics** – Measures agreement between predicted and actual classifications, adjusting for chance. Values range from -1 (disagreement) to 1 (perfect agreement). Example: Kappa = 0.8 indicates strong agreement beyond chance.

8. **ROC Curve (Receiver Operating Characteristic Curve)** – Plots True Positive Rate (Sensitivity) against False Positive Rate (1 – Specificity) to evaluate performance at different thresholds. A higher area under the curve (AUC) indicates better model performance. Example: A spam detection model with AUC = 0.95 is highly effective.

7. Problem on Gini Index & Decision Tree Construction, PYQ:

Suppose we want Gini index to decide whether the car will be stolen or not. The target classification is "car is stolen?" which can be Yes or No, create a decision tree for the given data below:

Car no	Colour	Type	Origin	Stolen ?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

☑ **Soln. :**

- In this example there are two classes Yes and No.

  No. of records for Yes = 5

  No. of records for No = 5

  Total No. of records = 10
- Now we will calculate Gini of the complete database as,

$$\text{Gini (T)} = 1-\left(1-\left(\left(\frac{5}{10}\right)^2+\left(\frac{5}{10}\right)^2\right)\right)=0.5$$

- Next we will calculate Split for all attributes, i.e. Colour, Type and Origin.

Colour

$$\text{Split} = \frac{2}{4}=\frac{5}{10}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right]+\frac{5}{10}\left[1-\left(\left(\frac{2}{5}\right)^2+\left(\frac{3}{5}\right)^2\right)\right]=0.48$$

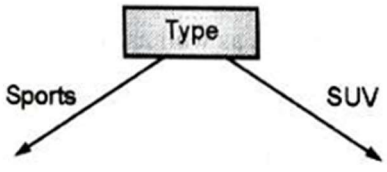
Type

$$\text{Split} = \frac{6}{10}\text{gini (Sports)}+\frac{4}{10}\text{gini (SUV)}=\frac{6}{10}\left[1-\left(\left(\frac{4}{6}\right)^2+\left(\frac{2}{6}\right)^2\right)\right]+\frac{4}{10}\left[1-\left(\left(\frac{1}{4}\right)^2+\left(\frac{3}{4}\right)^2\right)\right]=0.42$$

Origin

$$\text{Split} = \frac{5}{10}\text{gini (Domestic)}+\frac{5}{10}\text{gini (imported)}$$

$$=\frac{5}{10}\left[1-\left(\left(\frac{2}{5}\right)^2+\left(\frac{3}{5}\right)^2\right)\right]+\frac{5}{10}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right]=0.48$$



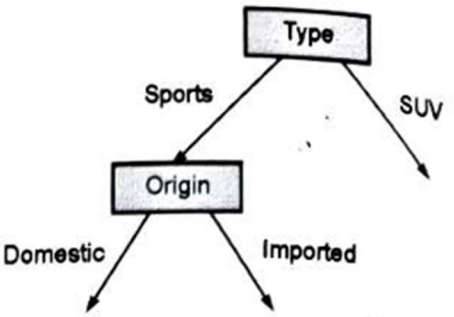
- Split value of Type is smallest, so we will select Type as root node.
- Next we will check for Type = Sports
- As attribute Type at root, we have to decide on remaining tree attribute for Sports branch.
- Consider only Colour and Origin for Type = Sports

Car no	Colour	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
10	Red	Sports	Imported	Yes

Colour

$$\text{Split} = \frac{4}{6}\text{gini (Red)} + \frac{2}{6}\text{gini (Yellow)}$$

$$= \frac{4}{6}\left[1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right]+\frac{2}{6}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right]=0.417$$



Origin>

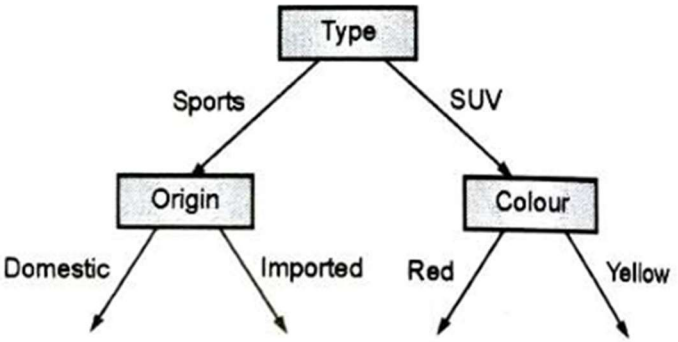
$$\text{Split} = \frac{4}{6}\text{gini (Domestic)} + \frac{4}{6}\text{gini (Imported)}$$

$$= \frac{4}{6}\left[1-\left(\left(\frac{2}{4}\right)^2+\left(\frac{2}{4}\right)^2\right)\right]+\frac{2}{6}\left[1-\left(\left(\frac{2}{2}\right)^2+\left(\frac{0}{2}\right)^2\right)\right]=0.33$$

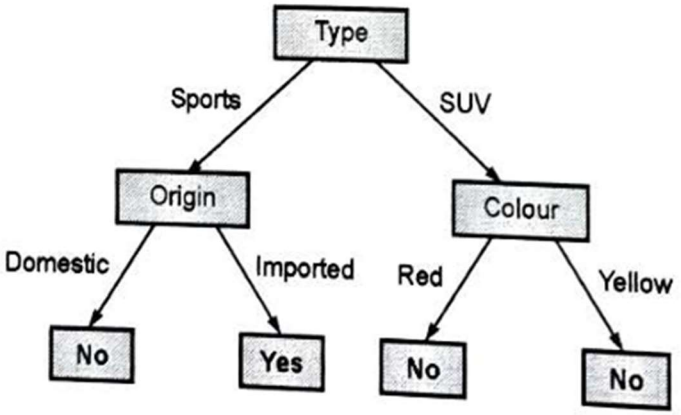
- Split value of Origin is smallest, so we will select Origin as next node.
- Next we will check for Type = SUV
- As attribute Type and Origin is already chosen, we have to decide on only remaining Colour attribute for SUV branch.
- Now we will check the value of 'Stolen?' from the database, for all branches,

  For, Type = Sports and Origin = Domestic, Stolen? = Yes as well as No

- So for this type of case we have to select the most common class. In this example there are 2 instances for Yes as well as No, so we can select any one. Let's we select No.
  - For, Type = Sports and Origin = Imported, Stolen? = Yes
  - For, Type = SUV and Colour = Red, Stolen? = No
  - For, Type = SUV and Colour = Yellow, Stolen? = Yes as well as No



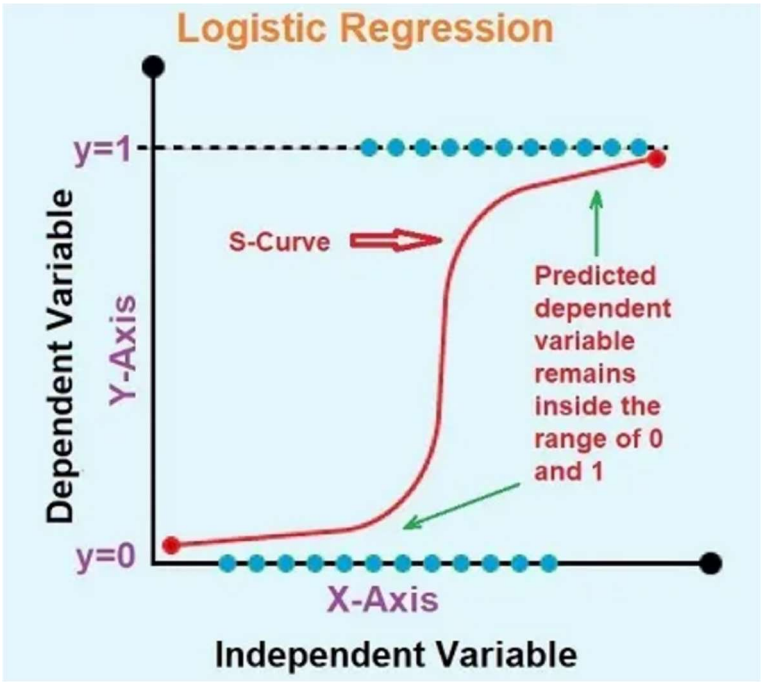
- So for this type of case we have to select the most common class (Since all attributes are already considered). In this example there are 2 instances for No and 1 instance of Yes, so we will select No.
- Final Decision Tree is



8. Explain Logistic Regression. #

Logistic Regression is a statistical method used for binary classification problems, where we need to predict the probability of an outcome that can take one of two possible categorical values (e.g., Yes/No, Pass/Fail, Spam/Not Spam).

The core idea is to model the probability that a given input belongs to a specific class. Unlike linear regression, which predicts continuous values, logistic regression predicts the probability of a discrete outcome using the sigmoid (logistic) function.



The Logistic Function (Sigmoid Function): It converts the linear combination of inputs into a probability (range 0 to 1).

$$\text{Sigmoid}(z) = \frac{1}{1 + e^{-z}} \quad \text{where } z = w_1 x_1 + w_2 x_2 + \cdots + b$$

Example:

Predicting if a student will pass an exam based on study hours:

- Input: Study hours (x)
- Output: Probability of passing (0 to 1)
- If probability > 0.5 → Predict "Pass"
- If probability ≤ 0.5 → Predict "Fail"

Applications:

- **Medical Diagnosis:** Disease/No Disease
- **Email Filtering:** Spam/Not Spam

9. Explain Regression line, Scatter plot, Error in prediction and Best fitting line. #

1. Regression Line

- Represents the relationship between independent (X) and dependent (Y) variables.
- Used to predict the value of Y for any given X based on historical data.

2. Scatter Plot

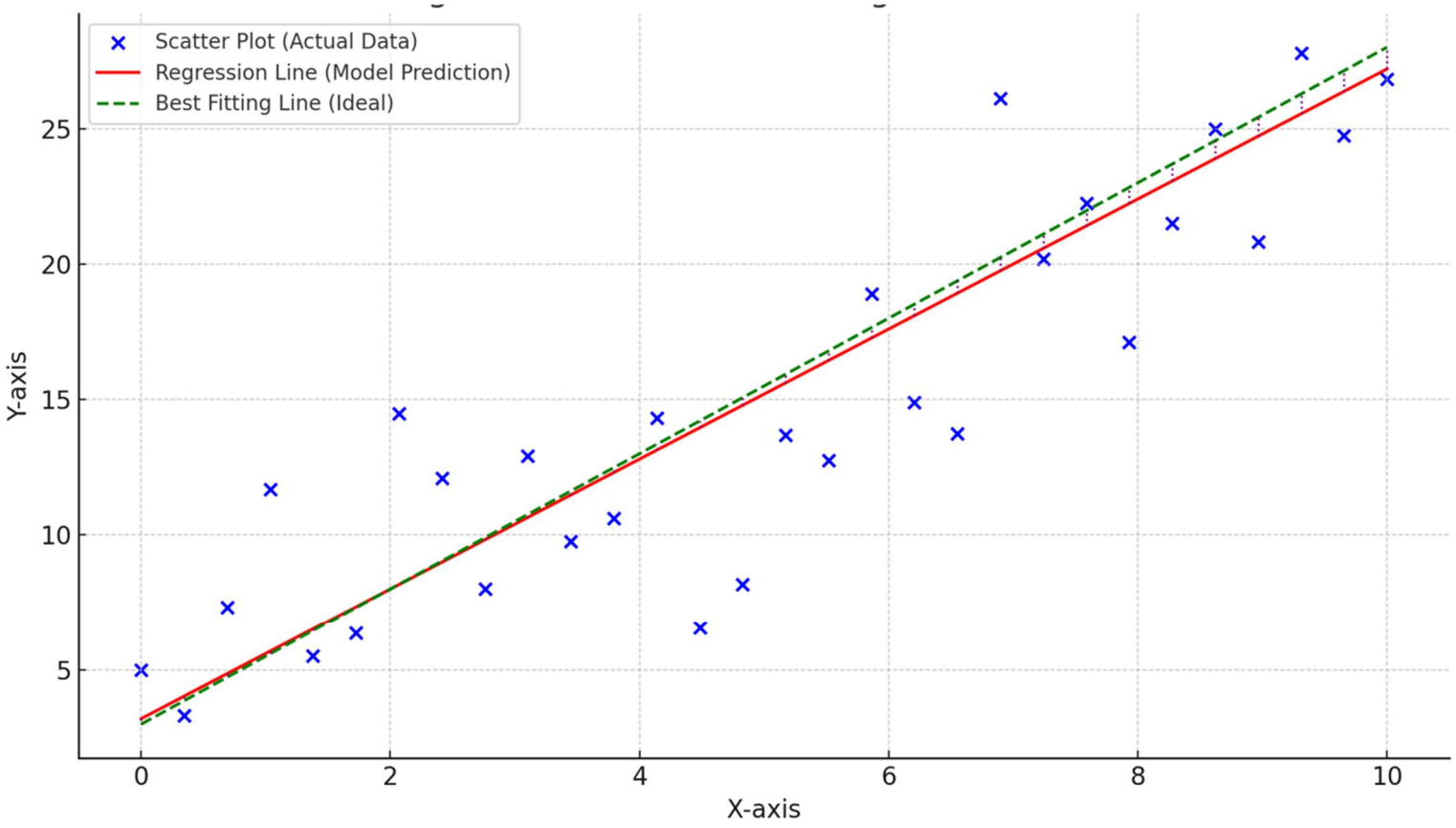
- A graphical representation where each point denotes a data pair (X, Y).
- Helps visualize data distribution and identify patterns or correlations.

3. Error in Prediction

- The difference between the actual and predicted output values.
- Calculated as: Error = Actual Y − Predicted Y; smaller errors indicate better accuracy.

4. Best Fitting Line

- The regression line that minimizes the overall prediction error.
- Determined using techniques like Least Squares, it best captures the data trend.



10. Problem on Linear Regression.

Following table shows the midterm and final exam grades obtained for students in a database course. Use linear regression to predict the final exam grade of a student who received 86 in the midterm exam.

Midterm exam (X)	72	50	81	74	94	86	59	83	86	33	88	81
Final exam (Y)	84	53	77	78	90	75	49	79	77	52	74	90

Sol^n

Let intercept be β_0 and slope be β_1

$\therefore Y = \beta_0 + \beta_1 X$: $\bar{X} = 73.917$

$\therefore \beta_0 = \bar{Y} - \beta_1 \bar{X}$: $\bar{Y} = 73.167$

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^{n} (x_i - \bar{X})^2}$$

Calculate the mean of X and Y and then calculate the other values as required

X	Y	$x_i - \bar{X}$	$y_i - \bar{Y}$	$(x_i - \bar{X}) \cdot (y_i - \bar{Y})$	$(x_i - \bar{X})^2$
72	84	-1.92	10.83	-20.7638	3.6736
50	53	-23.92	20.17	482.3194	572.0069
81	77	7.08	3.83	27.1527	50.1736
74	78	0.08	4.83	0.4027	0.0069
94	90	20.08	16.83	338.0694	403.3402
86	75	12.08	1.83	22.1527	146.0096
59	49	-14.92	-24.17	360.4861	222.5069
83	79	9.08	5.83	52.9861	82.5096
86	77	12.08	3.83	46.3194	146.0069
33	52	-40.92	-21.17	866.0699	1674.1736
88	74	14.08	0.83	11.7361	198.3402
81	190	7.08	16.83	119.2361	50.1736
				$\Sigma = 2306.1667$	$\Sigma = 3548.916$

$$\therefore \beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} = \frac{2306.1667}{3548.9167}$$

$$\beta_1 \approx 0.6498$$

$$\therefore \beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$= 73.1667 - 0.6493 \cdot (73.9167)$$

$$\beta_0 \approx 25.1339$$

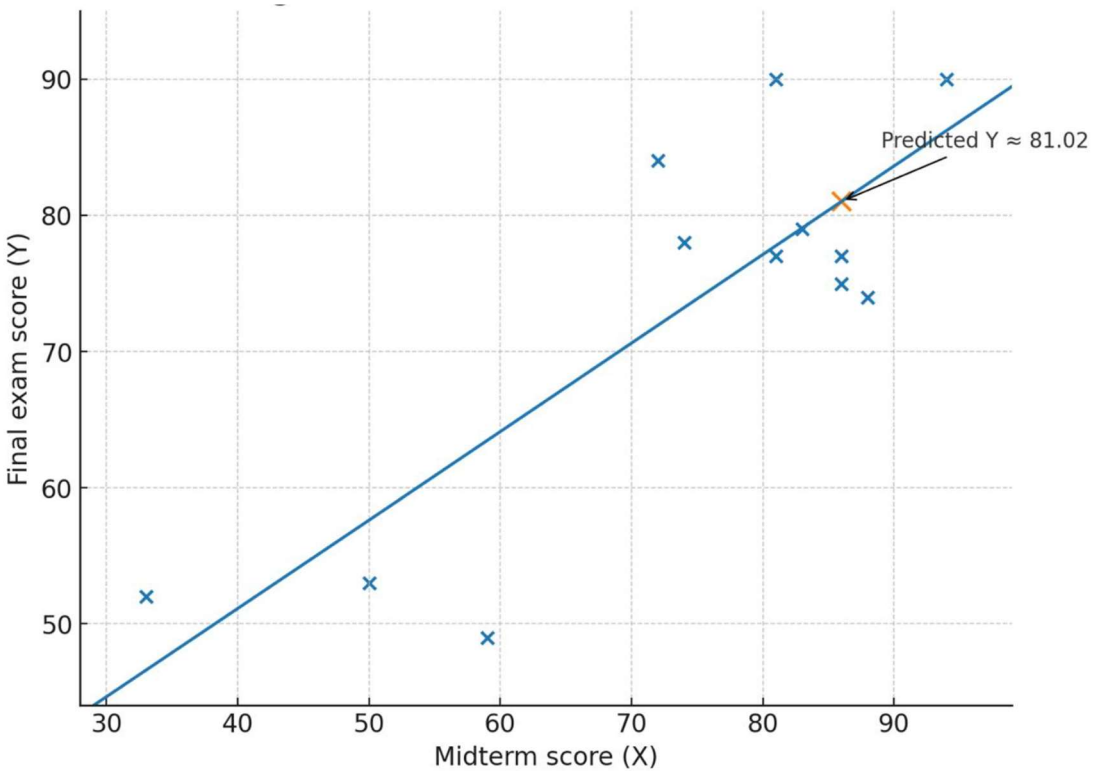
So, the regression equation is :

$$y = 25.1339 + 0.6498(x)$$

Prediction for $X = 86$

$$y = 25.1339 + 0.6498(86) \approx 81.0186$$

⇒ The predicted final exam grade for a student who scored 86 in the midterm is $y = 81.02$.



11. Explain Linear Regression with an example.

Linear Regression is a supervised machine learning algorithm used to predict a continuous numeric value based on one or more input features.
It assumes a linear relationship between the independent variables (inputs) and the dependent variable (output).

Types of Linear Regression:

- **Simple Linear Regression:** One input feature.
- **Multiple Linear Regression:** Two or more input features.

Equation:

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n + \epsilon$$

Where:

- y= Predicted (dependent) variable
- $x_1, x_2, \ldots, x_n$= Input (independent) features
- b_0= Intercept
- $b_1, b_2, \ldots, b_n$= Coefficients representing the impact of each feature
- ϵ= Error term (difference between actual and predicted values)

Working Steps:

1. Collect and prepare data.
2. Identify input features (independent variables) and output (dependent variable).
3. Fit the linear model using least squares to minimize prediction errors.
4. Use the model to predict new values.
5. Evaluate performance using metrics like Mean Squared Error (MSE) or R^2 score.

Example:
Predict a student's exam score (y) based on hours studied (x).

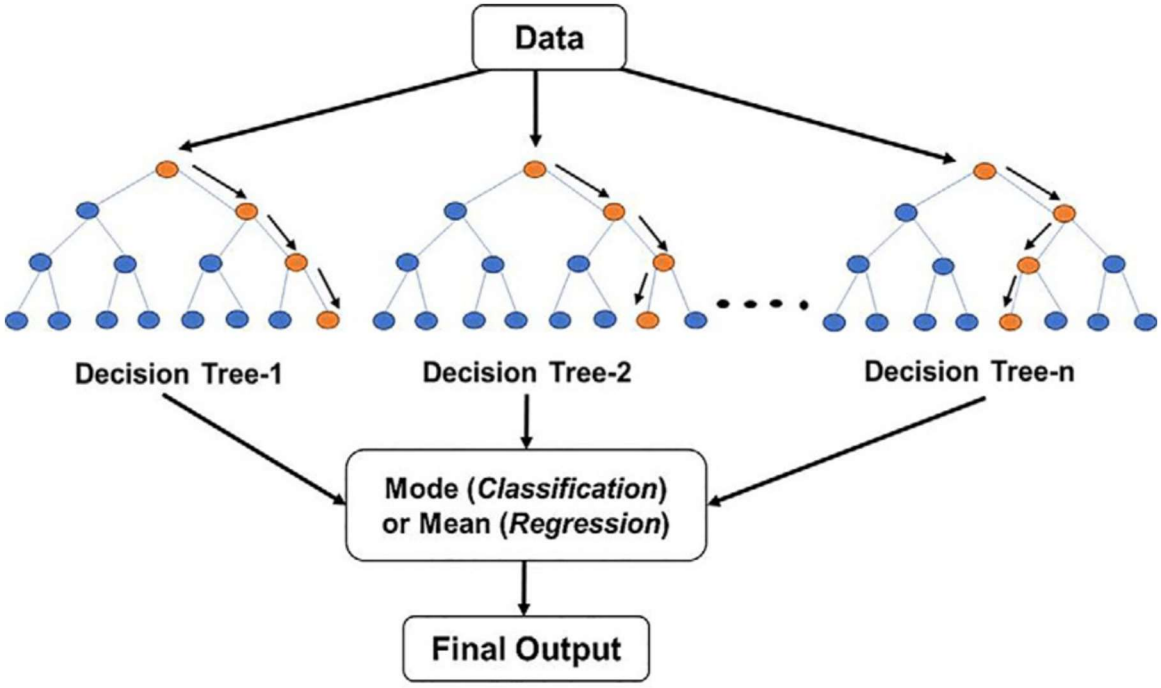
If the model finds the line $y = 5 + 10x$, it predicts that a student studying 3 hours will score $5 + 10 \times 3 = 35$.

Applications of Linear Regression:

- Predicting house prices based on size, location, etc.
- Forecasting sales based on advertising spend.
- Predicting crop yield based on rainfall, soil quality, and fertilizer usage.

3. Ensemble Learning

12. Explain the Random Forest algorithm in detail.



Random Forest is a supervised machine learning algorithm used for both classification and regression tasks.
It works by combining many decision trees and using their collective prediction (majority vote or average) to make the final decision. This approach is based on the idea that a "crowd" of weak learners can perform better than a single strong learner.

Working of Random Forest Algorithm

1. **Create Many Trees (Bagging)**
   Make several decision trees, each trained on a random sample of the original data (with replacement).
2. **Random Feature Selection:**
   At each split, only a few randomly chosen features are considered for finding the best split. This keeps the trees different.
3. **Each Tree Predicts:**
   Every tree makes its own prediction based on what it learned from its sample.
4. **Combine Predictions:**
   - **Classification:** The final class is the one most trees vote for (majority voting).
   - **Regression:** The final value is the average of all tree predictions.

Example: Predict whether a patient has a disease:

- Tree 1 → "Yes"
- Tree 2 → "No"
- Tree 3 → "Yes"
  Final output (majority vote) → "Yes"

Advantages

- High accuracy and robustness.
- Works well with both categorical and numerical data.
- Reduces overfitting due to averaging of multiple trees.
- Can handle missing values effectively.

Disadvantages

- Slower and more resource-intensive than a single decision tree.
- Less interpretable compared to individual trees.

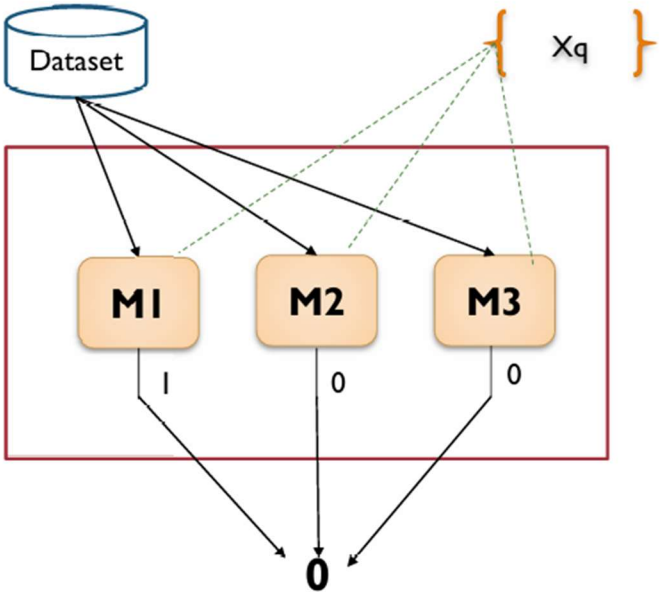
Applications

- Medical diagnosis (e.g., disease prediction)
- Stock market prediction
- Image and text classification

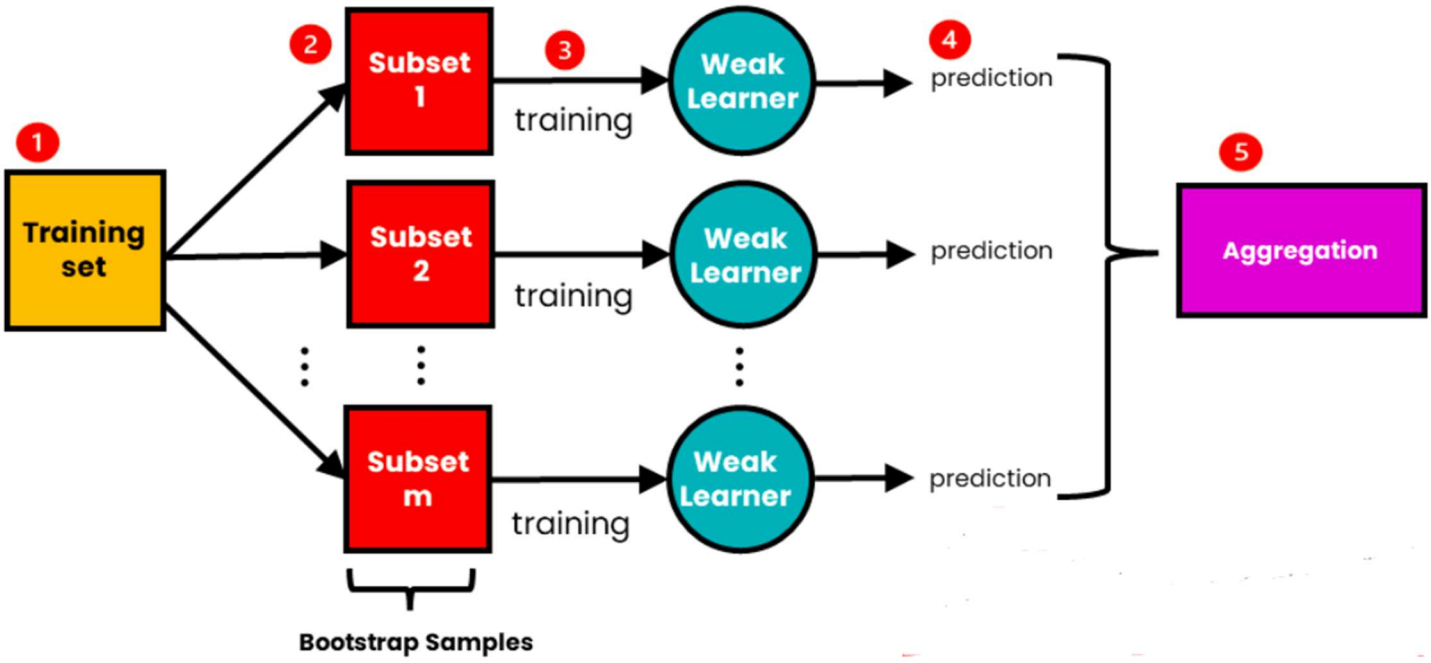
13. Explain different ways to combine classifiers.

Combining classifiers in machine learning, also known as ensemble learning, involves aggregating the predictions of multiple individual models to achieve better performance than any single model could achieve alone. The main ways to combine classifiers are:

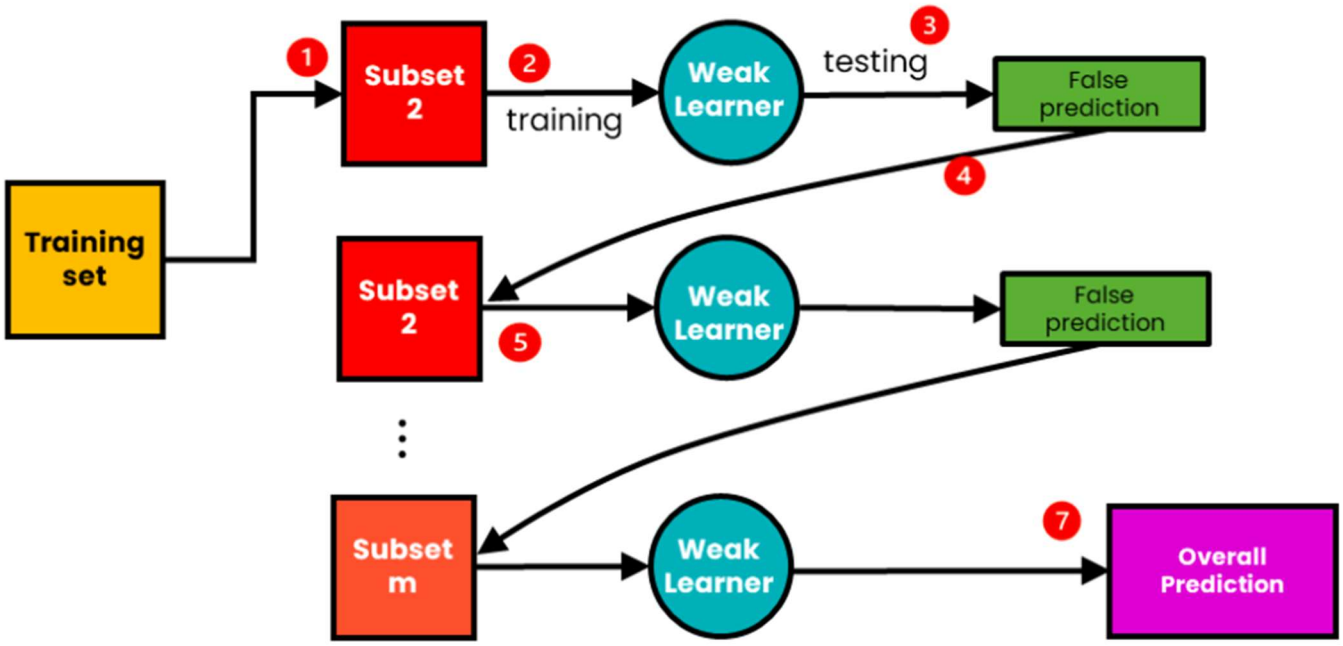
1. **Voting Ensemble** – Multiple models are trained independently, and their predictions are combined.
    - **Hard voting** → Majority class is chosen.
    - **Soft voting** → Average of predicted probabilities is taken.



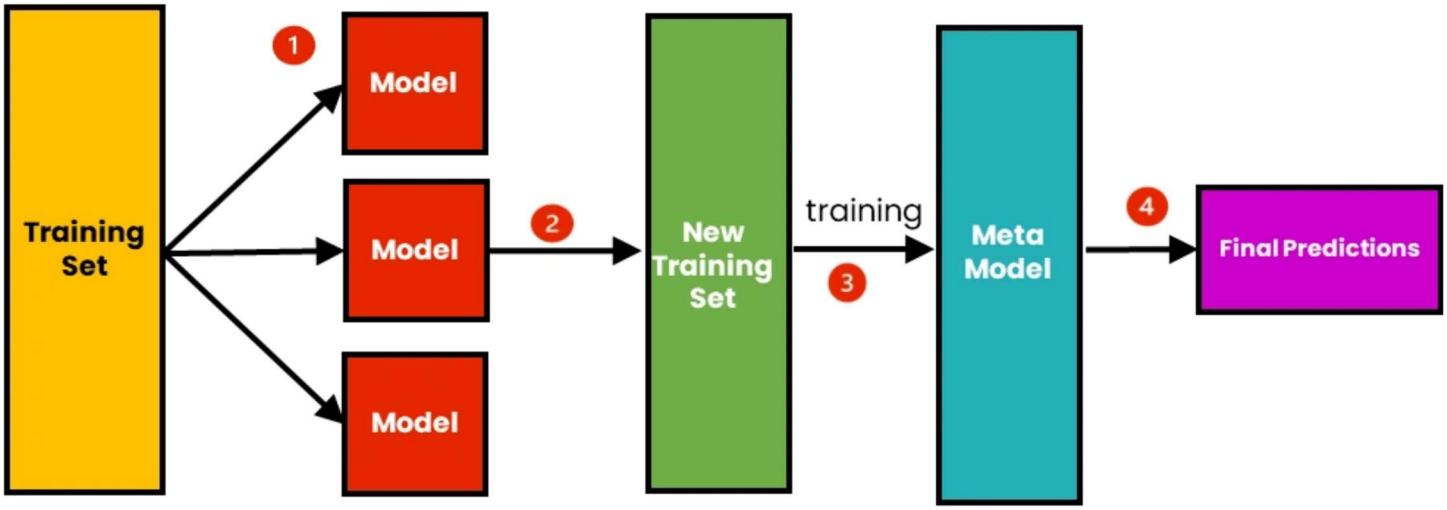
2. **Bagging (Bootstrap Aggregation)** – It works by taking several random subsets of the training data (called bootstrap samples, created with replacement), and training a weak learner (a simple model) on each subset independently and in parallel. Each weak learner then makes its prediction, and all predictions are combined—using majority vote for classification or averaging for regression—producing the final result. This method helps reduce variance and improve stability, since it relies on the collective output of multiple models trained on different versions of the data.



3. **Boosting** – It trains weak learners one after another in a sequential manner. Each learner gives extra attention to the errors or incorrect predictions made by its predecessor, focusing its training on the data points that were previously misclassified. This way, every next learner is better at correcting earlier mistakes. The outputs from all learners are then combined (typically in a weighted manner) to provide the final prediction. Boosting reduces bias and variance and results in a model that can handle complex patterns.



4. **Stacking** – Stacking involves training multiple base models (which can be of different algorithm types, such as SVMs, decision trees, etc.) on the same training set. Instead of just combining their outputs directly, the predictions from these base models are used to create a new dataset—a set of meta-features. A special meta-model, or stacker, is then trained on this new dataset to generate the final predictions. Stacking takes advantage of the strengths of various models and allows the meta-model to learn how best to blend their predictions for improved accuracy.



14. Explain the necessity of cross validation in ML applications and K-fold cross validation in detail.

Cross validation is crucial in machine learning because it helps evaluate how well a model's predictions will generalize to unseen data. Without cross validation, a model might perform well on its training set but poorly on new, real-world data due to overfitting.

Necessity:

- Ensures reliable model performance estimation.
- Detects overfitting or underfitting.
- Helps in selecting the best model and hyperparameters.

K-fold cross validation is the most popular cross validation method. It systematically partitions the entire dataset into multiple subsets ("folds") and uses each fold for validation exactly once.

Process:

- The entire dataset is divided into K equal-sized parts called "folds" (common K values are 5 or 10).
- The process is repeated K times:
  - Each time, one fold is used as the testing (or validation) set, and the remaining K-1 folds are used for training.
  - The model is trained and tested so every data point is used once for validation and K-1 times for training.
- After K iterations, all performance scores are averaged to produce a final estimate.

Example: K=5 (5-Fold Cross Validation)

Iteration	Training Set (4 Folds)	Testing/Validation Set (1 Fold)	Performance Score
1	Folds 2, 3, 4, 5	Fold 1	Score 1
2	Folds 1, 3, 4, 5	Fold 2	Score 2
3	Folds 1, 2, 4, 5	Fold 3	Score 3
4	Folds 1, 2, 3, 5	Fold 4	Score 4
5	Folds 1, 2, 3, 4	Fold 5	Score 5

Final performance is averaged:

$$\textbf{Final Performance} = \frac{\text{Score 1} + \text{Score 2} + \text{Score 3} + \text{Score 4} + \text{Score 5}}{5}$$

4. Learning with Classification

15. Describe Multiclass classification.

Multiclass Classification is a supervised learning technique where the model classifies input data into three or more distinct categories.
Unlike binary classification, which deals with only two classes (e.g., spam or not spam), multiclass classification handles multiple outcomes — for example, classifying an image as cat, dog, or horse.

Approaches for Multiclass Classification:
a) **One-vs-Rest (OvR):**

- A separate classifier is trained for each class against all others.
- Example: For 3 classes (A, B, C), three models are trained — A vs (B,C), B vs (A,C), and C vs (A,B).
- The class with the highest confidence score is chosen.

b) **One-vs-One (OvO):**

- A classifier is trained for every pair of classes.
- For 3 classes, 3 classifiers are created: A vs B, A vs C, and B vs C.
- The final class is decided by majority voting.

c) **Multinomial (Native) Approach:**

- Some algorithms like Softmax Regression or Decision Trees handle multiple classes directly without dividing the problem.

Common Algorithms Used:

- Logistic Regression (Softmax)
- Decision Trees and Random Forests
- K-Nearest Neighbours (KNN)
- Support Vector Machines (using OvR or OvO)

Evaluation Metrics:

- **Accuracy:** Percentage of correctly predicted classes.
- **Precision/Recall:** For each class, how well it distinguishes correct class vs. others.
- **Confusion Matrix:** Table showing actual vs. predicted classes, helps visualize performance.

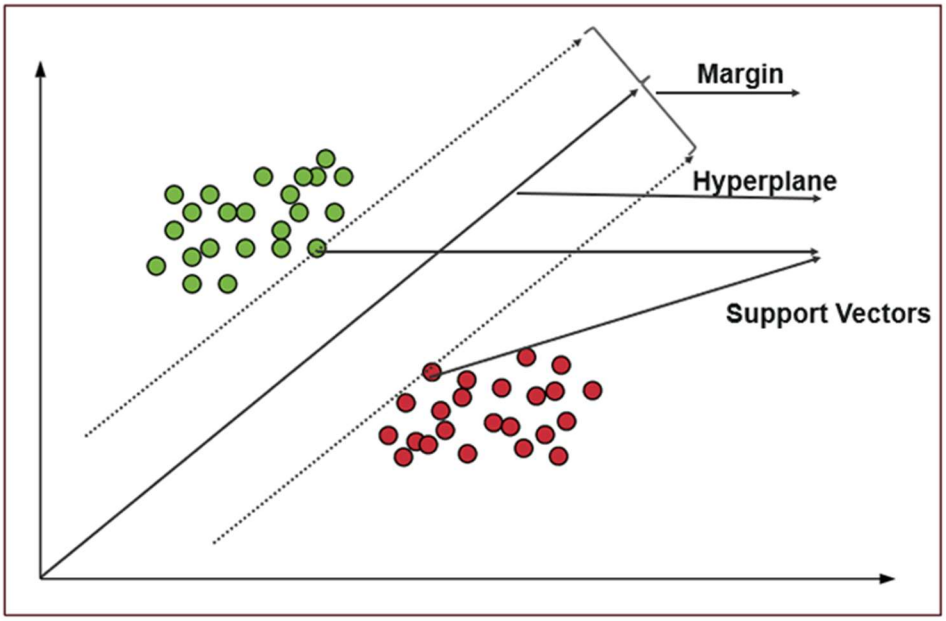
16. Explain the concept of margin and support vectors in Support Vector Machines (SVM). Define related terms: hyperplane, hard margin, soft margin, and kernel.

1. Margin

- In Support Vector Machines (SVM), the margin is the distance between the decision boundary (hyperplane) and the nearest data points from any class.
- A large margin means the classifier is more confident and generalizes better.
- The goal of SVM is to find the optimal hyperplane that maximizes this margin.

2. **Support Vectors**

- Support vectors are the data points that lie closest to the decision boundary (hyperplane).
- They are critical because:
  - They define the position and orientation of the hyperplane.
  - If we remove non-support vector points, the hyperplane doesn't change.
  - If we remove or move a support vector, the hyperplane shifts.



3. Hyperplane

- A hyperplane is the decision boundary that separates data points of different classes.
- In 2D, it's a line; in 3D, it's a plane; and in higher dimensions, it's called a hyperplane.

4. Hard Margin

- Used when data is perfectly linearly separable (no overlap or noise).
- The classifier aims to find a hyperplane that completely separates the classes without any misclassification.

5. Soft Margin

- Used when data is not perfectly separable or contains noise.
- Allows some misclassification to improve generalization.

- A penalty parameter (C) controls how much error is allowed — high C means fewer misclassifications but lower generalization.

6. Kernel

- A mathematical function that transforms non-linear data into higher dimensions to make it linearly separable.
- Common kernels: Linear, Polynomial, RBF (Radial Basis Function).

17. Explain support vector machine as a constrained optimization problem.

Support Vector Machine (SVM) aims to find the optimal hyperplane that separates data points of different classes with the maximum margin.
This can be formulated as a constrained optimization problem, where we maximize the margin while ensuring all data points are correctly classified.

Equation of the Hyperplane:
The separating hyperplane is defined as:

$$w \cdot x + b = 0$$

Where,

- w = weight vector
- b = bias term

Constraints for Classification:
For a dataset (x_i, y_i), where $y_i \in \{+1, -1\}$:

$$y_i(w \cdot x_i + b) \geq 1$$

This ensures that all data points lie on the correct side of the margin boundary.

Objective Function (Maximizing Margin)
The margin between the two classes is given by $\frac{2}{||w||}$.
Maximizing the margin is equivalent to minimizing $|| w ||^2$.
Hence, the optimization problem becomes:

Minimize:

$$\frac{1}{2} || w ||^2$$

Subject to:

$$y_i(w \cdot x_i + b) \geq 1, \forall i$$

Soft Margin SVM (for Non-Separable Data)
When data isn't perfectly separable, slack variables ξ_i are introduced to allow misclassification:

Minimize:

$$\frac{1}{2} || w ||^2 + C \sum_i \xi_i$$

Subject to:

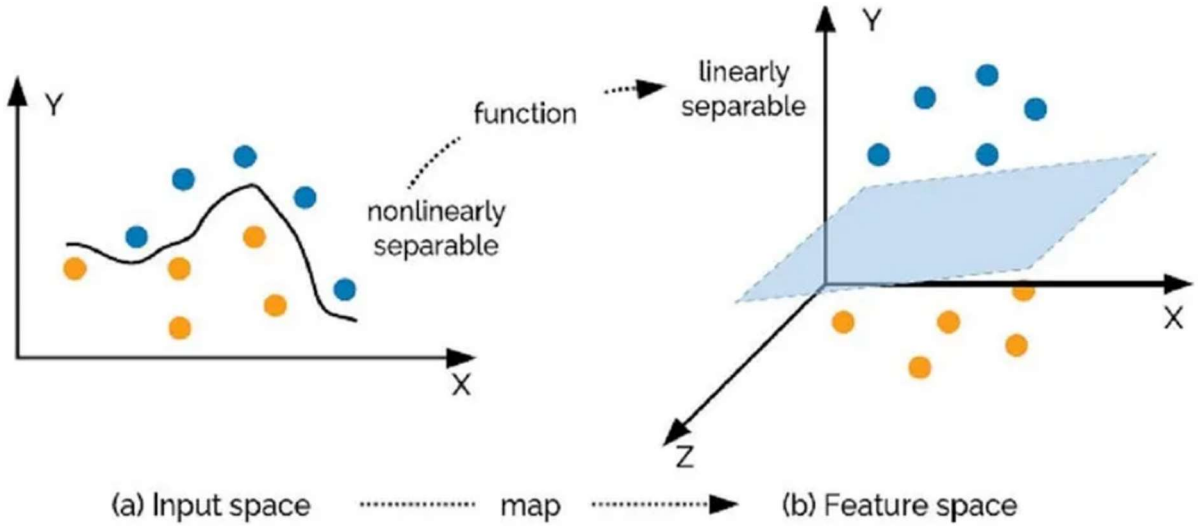
$$y_i(w \cdot x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0$$

Here, C is a regularization parameter that balances margin size and classification error.

Conclusion
Thus, SVM solves a quadratic optimization problem with linear constraints to find the hyperplane that achieves the best trade-off between margin maximization and classification accuracy.

18. Write a detailed not on SVM Kernel trick.

The Kernel Trick is a method used in Support Vector Machines (SVM) to handle non-linearly separable data by mapping it into a higher-dimensional space where it becomes linearly separable.
SVM uses a kernel function to compute the inner product in the higher-dimensional space efficiently.



How it Works

- SVM calculates decision boundaries based on dot products of data points.
- With the kernel trick, the dot product is replaced by a kernel function (e.g., linear, polynomial, RBF, or sigmoid) that measures similarity in a transformed space.

Example:

Linear kernel uses the standard dot product:

$$K(x, x') = x \cdot x'$$

This allows SVM to find straight-line boundaries in the original feature space.

Data remains in its original form; the kernel function determines how separation is computed.

Types of Kernels

- **Linear Kernel:**
  Works for data already separable by a straight line or plane.
  Formula: $K(x_i, x_j) = x_i^T x_j$
- **Polynomial Kernel:**
  Can separate data with curved boundaries (e.g., parabolas).
  Formula: $K(x_i, x_j) = (x_i^T x_j + c)^d$
- **Radial Basis Function (RBF) Kernel:**
  Handles clusters, rings, or complex patterns; can create circular or flexible decision boundaries.
  Formula: $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2)$

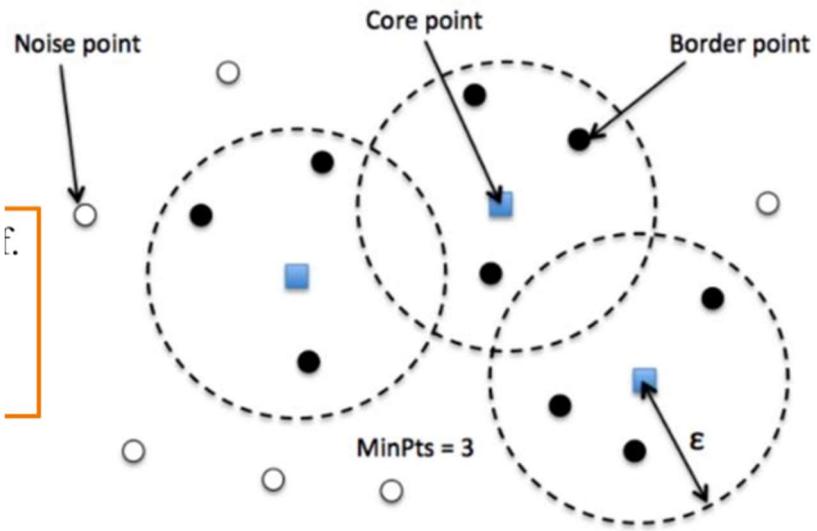
5. Learning with Clustering

19. Write a detailed note on the DBSCAN algorithm with a suitable example.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm that groups together data points that are closely packed (high-density regions) and marks points in low-density regions as noise or outliers. Unlike k-means, it does not require the number of clusters in advance and can discover clusters of arbitrary shapes.

Key Concepts

1. **ε (epsilon):** The neighbourhood radius — how far we look around a point.
2. **MinPts:** Minimum number of points required to form a dense region.
3. **Core Point:** A point having at least MinPts neighbours within distance ε.
4. **Border Point:** A point that is not a core point but falls within the ε-neighbourhood of a core point.
5. **Noise (Outlier):** A point that is neither a core point nor a border point.



Algorithm Steps

1. Pick an unvisited point.
2. If it has at least MinPts neighbours within ε → mark it as a core point and form a new cluster.
3. Expand the cluster by recursively including all points that are density-reachable from the core point.
4. If a point is not density-reachable from any core point → mark it as noise.
5. Repeat until all points are visited.

Example:

Consider a set of GPS coordinates representing trees in a forest. DBSCAN can identify dense clusters of trees (forest patches) and label isolated trees as noise. This is useful in environmental studies for mapping forest density or detecting sparse regions.

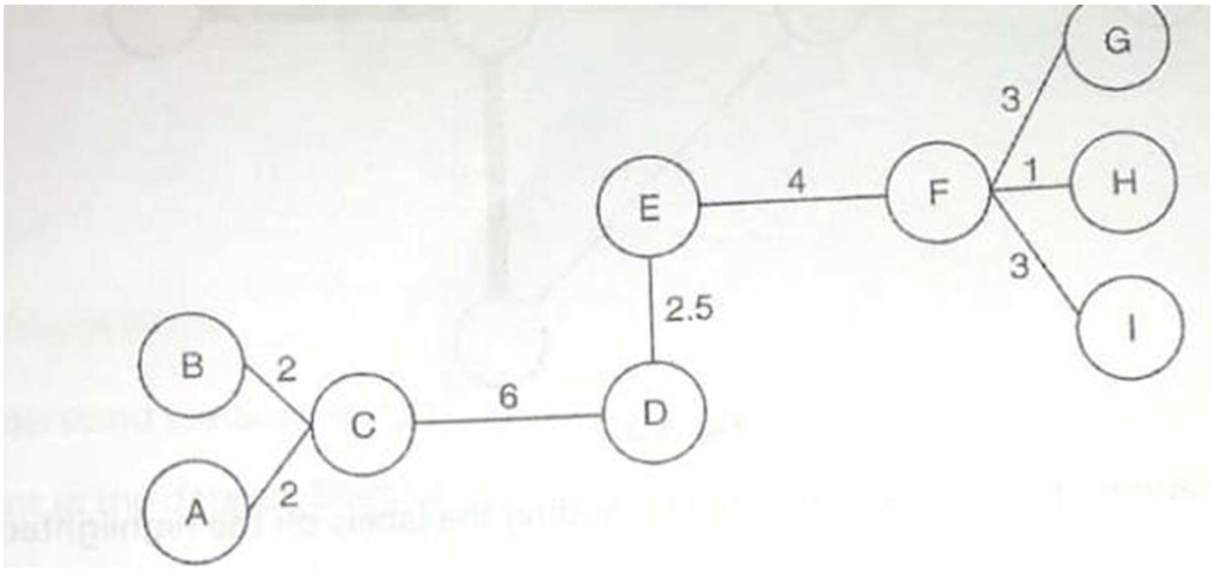
20. Demonstrate Minimal Spanning Tree (MST) algorithm for clustering with a suitable example.

A Minimum Spanning Tree (MST) connects all points (or nodes) in a weighted graph with the minimum possible total edge weight and no cycles. Apart from being useful in network design, MST is also applied in clustering, where the idea is to remove the heaviest edges to form separate groups of nodes.

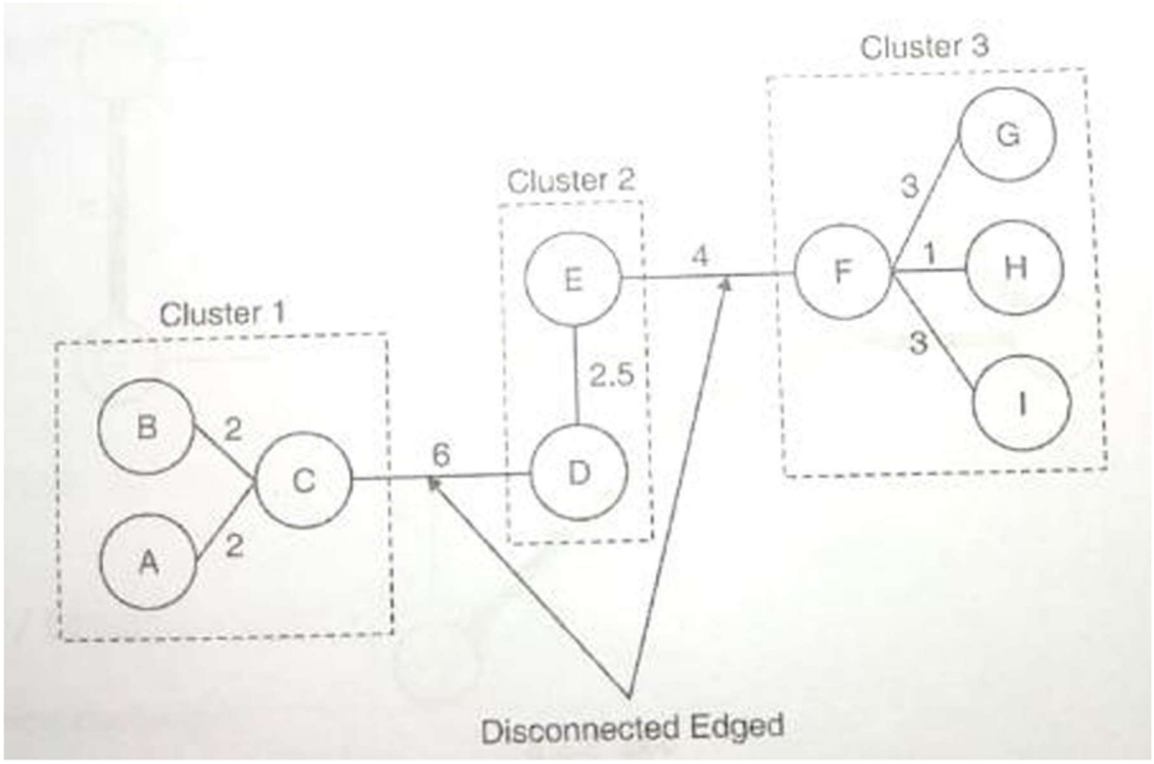
Steps for Clustering with MST

1. Build the MST from the given data points.
2. Remove the edges with the highest weights one by one until the desired number of clusters is obtained.

Example:



1. The MST formed from nodes A, B, C, D, E, F, G, H, I is shown in the figure.
    - It contains edges like A–C (2), B–C (2), D–E (2.5), F–G (3), F–H (1), F–I (3), etc.
2. To create two clusters, remove the edge with the largest weight:
    - The longest edge is C–D (6).
    - After removing it, we get two clusters:
        - Cluster 1 = {A, B, C}
        - Cluster 2 = {D, E, F, G, H, I}
3. To create three clusters, remove the next highest edge:
    - Remove E–F (4).
    - Now the clusters are:
        - Cluster 1 = {A, B, C}
        - Cluster 2 = {D, E}
        - Cluster 3 = {F, G, H, I}



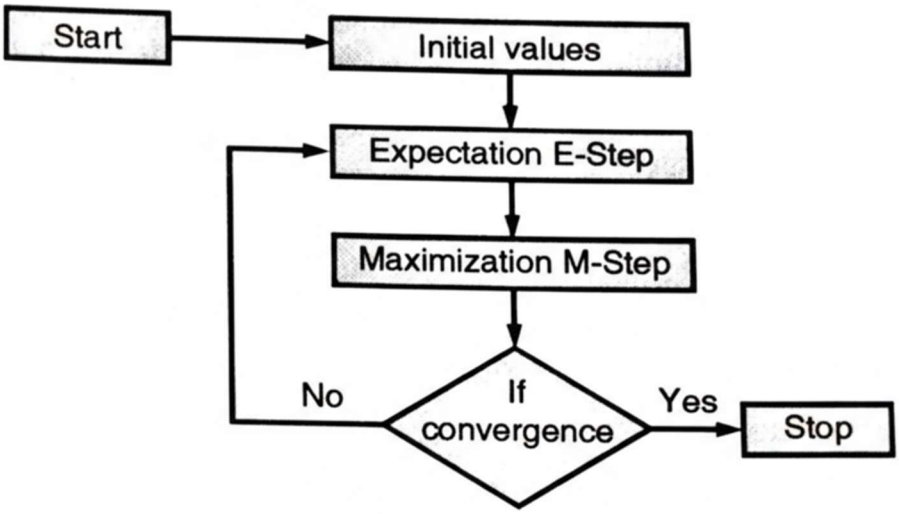
Final Result:

- MST first connects all points with minimum cost.
- Removing edges of highest weight breaks the tree into multiple clusters.
- For this example:
    - 2 clusters = {A, B, C} and {D, E, F, G, H, I}
    - 3 clusters = {A, B, C}, {D, E}, and {F, G, H, I}

21. Explain the Expectation-Maximization (EM) algorithm in detail.

Expectation-Maximization (EM) is an iterative algorithm used to find the best parameters (maximum likelihood estimates) of statistical models when the data is incomplete or has hidden (latent) variables.

EM alternates between estimating the missing data (Expectation step) and optimizing model parameters (Maximization step) until convergence.



Steps:

1. **Start → Initial Values:**
   - Begin by assigning initial guesses for the parameters of your model, such as means, variances, or mixture weights.
2. **Expectation (E) Step:**
   - Using the current parameters, calculate the expected value of the hidden variables.
   - For example, estimate the probability that each data point belongs to a particular cluster.
3. **Maximization (M) Step:**
   - Update the parameters of the model to maximize the likelihood function based on the expectations from the E-step.
   - Example: Recalculate cluster means, variances, or probabilities.
4. **Check Convergence:**
   - Determine if the changes in parameters between iterations are below a threshold.
   - **Yes → Stop:** If parameters have stabilized, the algorithm terminates.
   - **No → Loop Back to E-Step:** If not converged, repeat the E-step and M-step using updated parameters.
5. **Stop:**
   - The algorithm ends when parameters converge, providing the best estimates for the hidden variables and model parameters.

6. Dimensionality Reduction

22. Write a detailed note on Principal Component Analysis for Dimension Reduction.

Principal Component Analysis (PCA) is a widely used statistical technique for dimensionality reduction in machine learning and data analysis.
The main idea behind PCA is to transform a dataset with many variables into a smaller set of new variables called principal components (PCs), while retaining as much of the original information (variance) as possible.

These principal components are linear combinations of the original features and are arranged in such a way that:

- The first component captures the maximum variance in the data.
- The second component captures the next highest variance, and so on.
- Each component is orthogonal (uncorrelated) to the previous one.

Steps in PCA

1. **Standardize the data:** Ensure all features contribute equally by removing mean and scaling to unit variance.
2. **Compute the covariance matrix:** Shows relationships between variables.
3. **Find eigenvalues and eigenvectors:** Eigenvectors define directions of maximum variance (principal components), and eigenvalues show how much variance each captures.
4. **Select top components:** Keep the top *k* components that retain most of the data's variance.
5. **Transform data:** Project the data onto these selected components to form a lower-dimensional representation.

Example:

Suppose you have a dataset of students' performance with three features — marks in Physics, Chemistry, and Math.
Since these subjects are often correlated, PCA can combine them into a single principal component that represents the student's overall academic performance.
This reduces the dataset from 3 dimensions to 1, while still capturing most of the variation in their scores.

Advantages

- Reduces computation time and storage requirements.
- Removes redundant or correlated features.

Limitations

- PCA is a linear technique, not suitable for non-linear data relationships.
- The new components are not easily interpretable.

23. Write a detailed note on Linear Discriminant Analysis for Dimension Reduction.

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction technique that reduces features while keeping classes as separate as possible. Unlike PCA, which focuses only on data variance, LDA uses class labels to find directions that best separate the data.

Goal:
To project data onto a lower-dimensional space where the distance between different classes is large and the spread within each class is small.

Working:

1. **Compute the mean vectors** for each class.
2. **Calculate the within-class scatter matrix (Sw)** – Measures how samples vary within each class.
3. **Calculate the between-class scatter matrix (Sb)** – Measures how class means differ from the overall mean.
4. **Compute the matrix ($Sw^{-1} * Sb$)** and find its eigenvalues and eigenvectors.
5. **Select top eigenvectors** corresponding to the largest eigenvalues to form the new feature space.
6. **Project the data** onto this lower-dimensional subspace.

Example:

In a face recognition system, each image has thousands of pixel features. LDA reduces these features into a smaller set that still separates different people's faces clearly. This makes recognition faster and more accurate while keeping class differences intact.

Real-World Applications:

1. **Email Spam Detection:** Helps classify emails into spam and non-spam by projecting important distinguishing features.
2. **Medical Diagnosis:** Differentiates between healthy and diseased patients based on clinical or lab test data.
3. **Customer Segmentation:** Groups customers with similar buying behaviours for targeted marketing.

Asked once:

# indicates 5-mark question

1. Introduction to Machine Learning

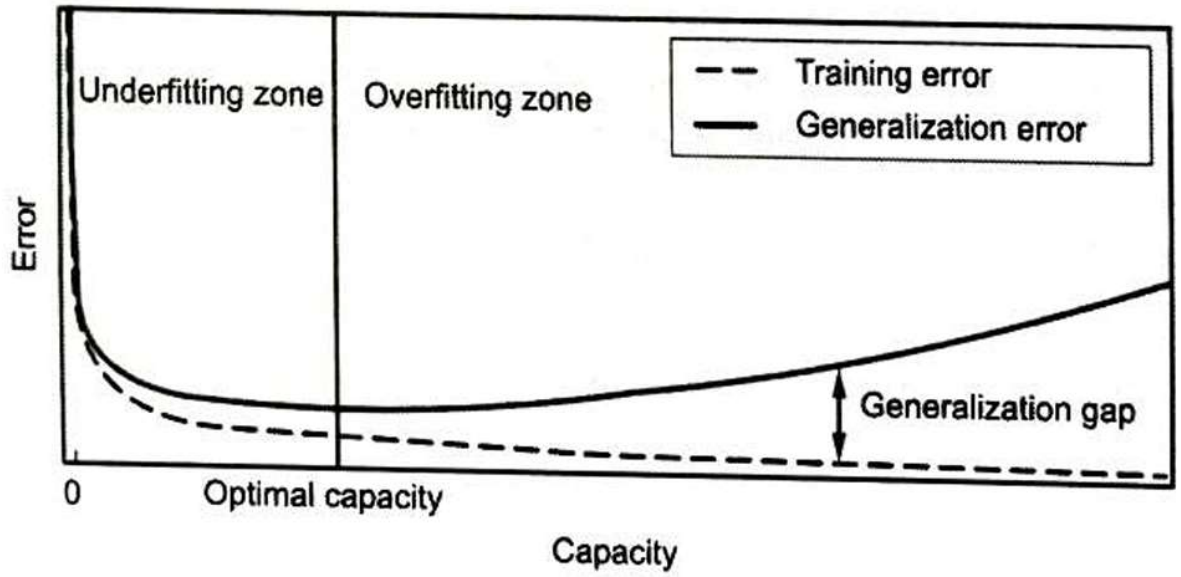
1. Explain Training error and Generalization error. #

Training Error:

- It is the error the model makes on the training data — the same data it was trained on.
- A low training error means the model fits the training data well.
- However, if it's too low, it may indicate overfitting (the model has memorized the data).

Generalization Error:

- It is the error the model makes on unseen or test data that was not used during training.
- It measures how well the model can generalize its learning to new data.
- A model with low training error but high generalization error performs poorly in the real world.



As model capacity increases, training error continues to decrease, but generalization error starts increasing beyond a certain point—illustrating overfitting.

The graph shows the "generalization gap": the difference between training and generalization errors, with an optimal capacity zone where both errors are minimized.

2. Differentiate between Supervised and Unsupervised Learning.

Parameter	Supervised Learning	Unsupervised Learning
Definition	Model learns from labelled data (input–output pairs).	Model learns from unlabelled data without predefined outputs.
Goal	Predict outcomes or classify data.	Discover patterns or structure in data.
Examples	Classification, regression	Clustering, dimensionality reduction
Output type	Predicts a known outcome/class.	Groups/sorts data by similarity.
Algorithm types	Decision trees, SVM	K-means, PCA
Complexity	Simpler to interpret and evaluate.	More complex and exploratory in nature.
Accuracy evaluation	Easy to measure with labels.	Harder to quantify without labels.
Applications	Spam detection, sentiment analysis	Customer segmentation, anomaly detection

2. Learning with Regression and Trees

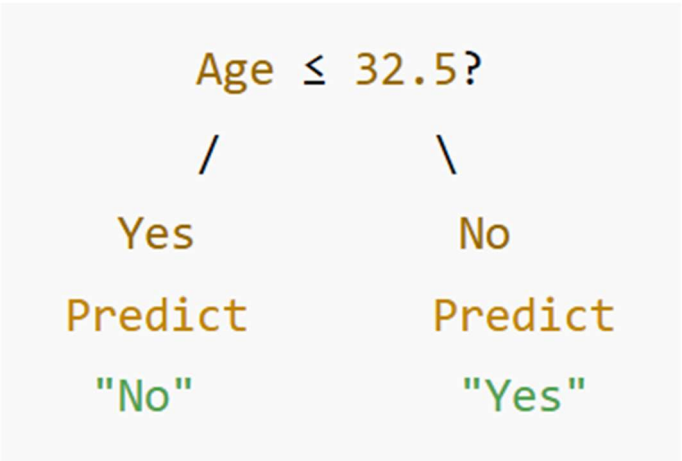
3. Demonstrate CART method along with an example.

CART stands for Classification and Regression Trees, a decision-tree-based method used for both classification and regression problems. It works by splitting the dataset into smaller and smaller subsets using decision rules until the model can make accurate predictions.

Example - Classification
Problem: Predict whether a person will buy a product based on Age and Income.

Age	Income	Buy
25	High	No
35	Medium	Yes
45	Low	Yes
30	Medium	No
50	High	Yes

1. **Calculate Root Gini**
   Yes = 3, No = 2 → Gini = 0.48
2. **Test All Possible Splits**
   Try Age thresholds (27.5, 32.5, 40, 47.5) and Income categories.
   Compute weighted Gini for each split.
3. **Find Best Split**
   Age ≤ 32.5 gives Weighted Gini = 0, which means perfect separation.
4. **Form Tree**
   Left branch (Age ≤ 32.5) → all No → Leaf "No".
   Right branch (Age > 32.5) → all Yes → Leaf "Yes".



5. **Prediction Rule**
   If Age ≤ 32.5 → predict No
   Else → predict Yes

4. Explain Gini index along with an example.

The Gini Index is a measure of impurity or diversity used by decision tree algorithms to decide the best feature for splitting the data.
A lower Gini index indicates a purer node, meaning the data within that node mostly belongs to a single class.

Formula:

$$Gini = 1 - \sum p_i^2$$

where p_i is the probability of each class in the node.

Example:
Suppose a node has 10 samples: 4 are Class A and 6 are Class B.

- $p_A = 4/10 = 0.4$
- $p_B = 6/10 = 0.6$

Calculate Gini index:

$$\text{Gini} = 1 - (0.4^2 + 0.6^2) = 1 - (0.16 + 0.36) = 1 - 0.52 = 0.48$$

A Gini index of 0 means the node is pure (contains only one class); a value closer to 1 means more mixed classes.

5. Explain performance evaluation measures for regression.

1. Mean Absolute Error (MAE)

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

It measures the average absolute difference between the actual and predicted values.

Lower MAE indicates better model

2. Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

It measures the average squared difference between actual and predicted values.

Lower MSE indicates better performance

3. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE}$$

It is the square root of MSE and represents the standard deviation of prediction errors.
It has the same units as the target variable, making it easier to interpret.

4. R-squared (Coefficient of Determination)

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

It shows how well the regression line fits the data.

- $R^2 = 1$→ Perfect fit
- $R^2 = 0$→ Model explains none of the variability

Higher R^2means better explanatory power.

5. Adjusted R-squared

$$\text{Adjusted } R^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

Used when multiple features are present.
It adjusts $R^2$for the number of predictors to prevent overestimation of model performance.

6. Differentiate between Linear regression and Logistic regression. #

Parameter	Linear Regression	Logistic Regression
Type of Problem	Used for regression with continuous outputs.	Used for classification with discrete outputs.
How Model Learns	Learns by fitting a line that best predicts the output.	Learns by finding parameters that best separate classes.
Output	Predicts a numeric value.	Predicts probability of a class (0 to 1)
Equation / Model	$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n + \epsilon$	$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}}$
Loss Function	Minimizes mean squared error.	Minimizes log loss.
Decision Making	Uses the predicted value directly.	Classifies data using a threshold (usually 0.5).
Prediction	Gives estimated target value.	Gives probability of a class.
Use Cases	Predicts prices, temperatures, etc.	Predicts spam, disease, etc.

7. Explain Multivariate Linear regression method.

Multivariate Linear Regression is an extension of simple linear regression where the model predicts a dependent variable (Y) using two or more independent variables (X_1, X_2, X_3, ...). It helps understand how multiple factors together influence the output.

Equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta n Xn + \varepsilon$$

Where:

- Y= Dependent variable
- $X_1, X_2, \ldots, Xn$= Independent variables
- β_0= Intercept
- $\beta_1, \beta_2, \ldots, \beta n$= Coefficients (effect of each predictor)
- ε= Error term

Example:
Predict house price (Y) using area (X_1), number of rooms (X_2), and age (X_3):

$$Y = 50{,}000 + 200X_1 + 10{,}000X_2 - 500X_3$$

For a 1000 sq. ft, 3-room, 10-year-old house:

$$Y = 50{,}000 + (200 \times 1000) + (10{,}000 \times 3) - (500 \times 10) = ₹2{,}95{,}000$$

Advantages:

- Captures combined effect of multiple predictors
- Provides better accuracy for complex data

Limitations:

- Sensitive to outliers
- Assumes linear relationships
- Poor performance with correlated variables

Applications:

- Business forecasting
- Healthcare (predicting disease risk from multiple factors)

3. Ensemble Learning

8. Write detailed note on XGBoost ensemble method.

XGBoost (Extreme Gradient Boosting) is an advanced version of the Gradient Boosting algorithm that builds an ensemble of many small decision trees (weak learners) sequentially.
Each new tree is trained to correct the errors (residuals) made by the previous ones, making the final model fast, accurate, and efficient. Instead of one large model, it combines several simple models to form a strong and highly predictive system.

Working Steps:

1. **Initialize** the model with an initial prediction (e.g., average of target values).
2. **Compute Residuals** – the difference between actual and predicted values.
3. **Build a New Tree** to predict these residuals.
4. **Update Predictions** using:

$$\text{New Prediction} = \text{Old Prediction} + \eta \times \text{Tree Output}$$

(η = learning rate to control step size)

5. **Repeat** steps until the model converges or reaches the set number of trees.
6. **Final Prediction** = Sum of all tree outputs.

Example:

Predicting loan approval:
The first tree predicts based on income, the next corrects errors based on credit score, and the next focuses on loan amount, together improving accuracy.

Advantages:

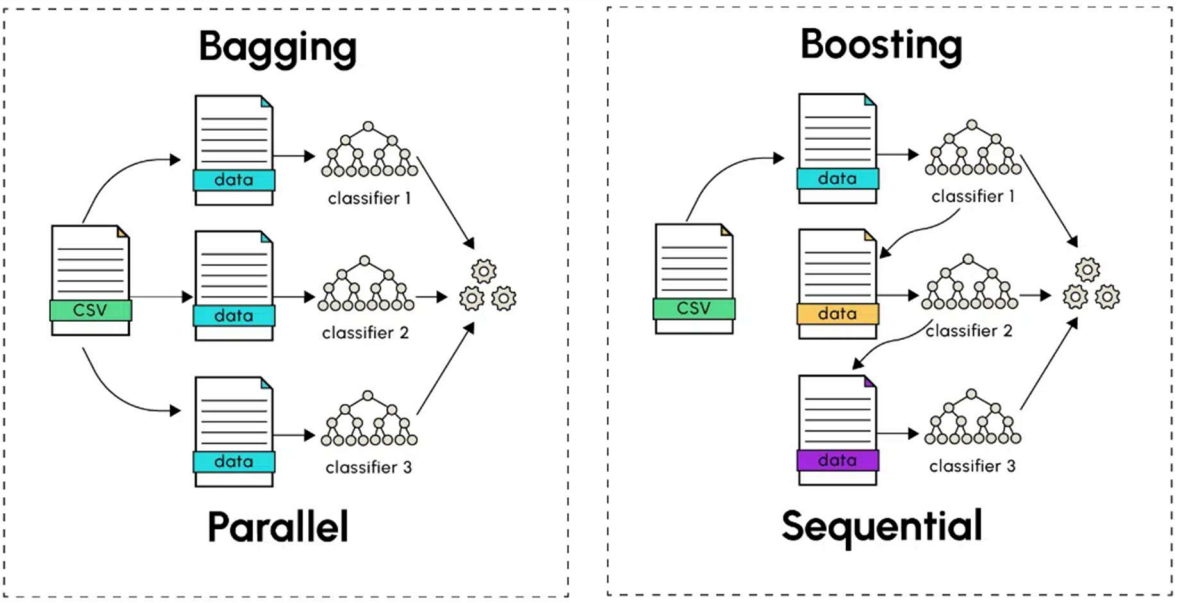
- Prevents overfitting using regularization.
- Works on large datasets.

Limitations:

- Needs careful tuning of parameters.
- High memory usage on large data.

9. Compare Bagging and Boosting with reference to ensemble learning. Explain how these methods help to improve the performance of the machine learning model.

Feature	Bagging	Boosting
Model Training	Models are trained in parallel	Models are trained sequentially, each correcting previous errors
Data Distribution	Uses random subsets of data with replacement	All data used, but gives more focus to misclassified samples
Goal	Primarily reduces variance (overfitting)	Primarily reduces bias (underfitting) and can also reduce variance
Combination Method	Simple voting or averaging of predictions	Weighted voting, giving more weight to accurate models
Model Dependence	Models are independent	Each new model depends on previous models
Use-case Examples	Random Forest, Bagged Decision Trees	AdaBoost, Gradient Boosting



How Bagging and Boosting Improve Model Performance

1. **Bagging:**
   - By training multiple models on different random subsets of data, bagging reduces the impact of noise or variance from any single model.
   - Aggregating predictions stabilizes output and improves overall generalization.
2. **Boosting:**
   - Sequentially trains models, each focusing on the mistakes of the previous one.
   - Assigning higher weight to misclassified examples ensures that the ensemble learns difficult patterns, reducing bias.
   - The combination of weak learners produces a strong model with improved accuracy.

4. Learning with Classification

10. Consider the use case of Email spam detection. Identify and explain the suitable machine learning technique for this task.

Email spam detection aims to automatically identify and filter out unwanted or harmful emails (spam) from legitimate messages. The task is a binary classification problem, where each email must be classified as either spam or not spam.

ML technique for this task:

Support Vector Machine (SVM) is a strong choice for spam detection due to its ability to handle high-dimensional feature spaces (like text data), maximize the margin between classes, and adapt to both linear and nonlinear data patterns with the kernel trick. SVM is known for high accuracy and robustness, making it well-suited for distinguishing between spam and non-spam emails in varied datasets.

Main Steps in Building an SVM-Based Spam Detector

1. Prepare Data: Collect and label a dataset of emails as spam or not spam.
2. Preprocess and Extract Features: Transform the raw emails into numeric form, capturing relevant signs of spam.
3. Train the SVM: Use the labeled feature vectors to teach the SVM how to separate spam from legitimate messages.
4. Test and Evaluate: Assess how well the SVM predicts new, unseen emails by checking metrics like accuracy, precision, and recall.
5. Deploy and Monitor: Integrate the trained SVM model into the email system and monitor its ongoing performance.

11. Differentiate between Logistic regression and Support vector machine.

Parameter	Logistic Regression	Support Vector Machine (SVM)
Type of Problem	Used for classification with discrete outputs.	Used for classification (and regression in SVR).
How Model Learns	Learns by finding parameters that best separate classes.	Learns by finding the hyperplane that maximizes margin between classes.
Output	Predicts probability of a class.	Predicts class labels directly based on the decision boundary.
Equation	$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}}$	$f(x)=w^T x+b$
Loss Function	Minimizes log loss (cross-entropy).	Minimizes hinge loss.
Decision Making	Classifies data using a threshold (usually 0.5).	Classifies data based on which side of the hyperplane it lies.
Handling Non-linear Data	Requires feature engineering or polynomial transformations.	Uses kernel trick to handle non-linear decision boundaries.
Use Cases	Predicts spam, disease, etc.	Image recognition, text classification.

5. Learning with Clustering

12. Explain K-means algorithm.

K-Means is an unsupervised learning algorithm that partitions data into K clusters, where each point belongs to the cluster with the nearest centroid.

Working:

1. **Initialize:** Choose K centroids randomly.
2. **Assign:** Assign each data point to the nearest centroid.
3. **Update:** Recalculate centroids as the mean of all points in each cluster.
4. **Repeat:** Repeat assignment and update steps until centroids don't change (convergence).
5. **Result:** Data points are grouped into K clusters minimizing the distance to their centroids.

Example: Suppose you have customer data with age and income. K-Means can group them into 3 clusters: low, medium, and high spenders based on similarity in age and income.

13. Explain the distance metrics used in clustering.

Distance metrics measure how similar or dissimilar data points are, which helps clustering algorithms group similar points together.

Common Metrics:

1. **Euclidean Distance:** Measures the straight-line distance between two points in n-dimensional space; widely used for numerical data.
2. **Manhattan Distance:** Calculates the sum of absolute differences along each dimension; useful when movement is along a grid.
3. **Minkowski Distance:** A generalized form of Euclidean and Manhattan distances, controlled by a parameter p.
4. **Cosine Similarity:** Measures the angle between two vectors, focusing on their direction rather than magnitude; ideal for text or high-dimensional data.

6. Dimensionality Reduction

14. What is dimensionality reduction? Explain how it can be utilized for classification and clustering task in Machine Learning.

Dimensionality reduction is the process of reducing the number of input features in a dataset while keeping the most important information. It helps simplify data, remove noise, and improve the efficiency of machine learning models.

1. **For Classification:**
    - Removes less useful features, making models like SVM or Logistic Regression faster and less prone to overfitting.
    - Helps focus on the most important features that separate classes clearly.
2. **For Clustering:**
    - Makes clusters more distinct and easier to visualize.
    - Improves performance of algorithms like K-Means by reducing irrelevant dimensions.

Common Techniques:

- PCA (Principal Component Analysis)
- LDA (Linear Discriminant Analysis)
- SVD (Singular Value Decomposition)

15. Explain the concept of feature selection and extraction.

Feature Selection:
It is the process of choosing the most important features from the existing data that contribute the most to the output.

- Removes irrelevant or redundant data.
- Helps reduce overfitting and improves model accuracy.
- **Example:** Selecting only “age” and “income” from many customer attributes to predict spending habits.

Feature Extraction:
It is the process of creating new features from existing ones by transforming or combining them.

- Reduces data dimensionality while keeping essential information.
- **Example:** Using PCA to combine many correlated features into a few uncorrelated ones (principal components).

16. Find SVD for A = $\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$

https://www.youtube.com/watch?v=a4NclubgZoQ&t=73s

Transpose A to get $A^T = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix}$

Calculate $A \times A^T$

$$A \times A^T = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}$$

Calculate $A^T \times A$

$$A^T \times A = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$$

To find the value of left singular vector U, let's find the eigenvalues and eigenvectors of $A \times A^T = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}$

$$\begin{bmatrix} 8-\lambda & 0 \\ 0 & 2-\lambda \end{bmatrix} = 0$$

$$(8-\lambda)(2-\lambda) - 0 \times 0 = 0$$

Hence $\lambda = 8$ or $\lambda = 2$

Let's find the 1st eigenvector corresponding to $\lambda = 8$. Let's call it $v = \begin{bmatrix} x \\ y \end{bmatrix}$.

$$\begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = 8 \times \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 8x \\ 2y \end{bmatrix} - \begin{bmatrix} 8x \\ 8y \end{bmatrix} = 0$$

Hence, you get two equations.

$$8x - 8x = 0 \text{ and } 2y - 8y = 0$$

From these equations, you can pick the value of x = 1 and y could only be 0.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

To make the length of this eigenvector as 1, divide the eigenvector with its current length $\sqrt{(1)^2 + (0)^2} = 1$.

Hence, 1st eigenvector of unit length is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1 \\ 0/1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Let's find the 2nd eigenvector corresponding to $\lambda = 2$. Let's call it $v = \begin{bmatrix} x \\ y \end{bmatrix}$.

$$\begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = 2 \times \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 8x \\ 2y \end{bmatrix} - \begin{bmatrix} 2x \\ 2y \end{bmatrix} = 0$$

Hence, you get two equations.

$$8x - 2x = 0 \text{ and } 2y - 2y = 0$$

From these equations, you can pick the value of y = 1 and x could only be 0.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

To make the length of this eigenvector as 1, divide the eigenvector with its current length $\sqrt{(0)^2 + (1)^2} = 1$.

Hence, 2nd eigenvector of unit length is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0/1 \\ 1/1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

You sort the eigenvectors according to eigenvalues.

Hence,

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

To find the value of right singular vector V^T, let's find the eigenvalues and eigenvectors of $A^T \times A = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$

$$\begin{bmatrix} 5-\lambda & 3 \\ 3 & 5-\lambda \end{bmatrix} = 0$$

$$(5-\lambda)(5-\lambda) - 3 \times 3 = 0$$

$$\lambda^2 - 10\lambda + 25 - 9 = 0$$

$$\lambda^2 - 10\lambda + 16 = 0$$

Hence $\lambda = 8$ or $\lambda = 2$

Let's find the 1st eigenvector corresponding to $\lambda = 8$. Let's call it $v = \begin{bmatrix} x \\ y \end{bmatrix}$.

$$\begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = 8 \times \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 5x + 3y \\ 3x + 5y \end{bmatrix} - \begin{bmatrix} 8x \\ 8y \end{bmatrix} = 0$$

$$\begin{bmatrix} -3x + 3y \\ 3x - 3y \end{bmatrix} = 0$$

Hence, $x = y$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

To make the length of this eigenvector as 1, divide the eigenvector with its current length

$$\sqrt{(1)^2 + (1)^2} = \sqrt{2} = 1.41.$$

Hence, 1st eigenvector of unit length is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1.41 \\ 1/1.41 \end{bmatrix} = \begin{bmatrix} 0.70 \\ 0.70 \end{bmatrix}$$

Let's find the 2nd eigenvector corresponding to$\lambda = 2$. Let's call it $v = \begin{bmatrix} x \\ y \end{bmatrix}$.

$$\begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = 2 \times \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 5x + 3y \\ 3x + 5y \end{bmatrix} - \begin{bmatrix} 2x \\ 2y \end{bmatrix} = 0$$

$$\begin{bmatrix} 3x + 3y \\ 3x + 3y \end{bmatrix} = 0$$

Hence, $x = -y$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

To make the length of this eigenvector as 1, divide the eigenvector with its current length

$$\sqrt{(-1)^2 + (1)^2} = \sqrt{2} = 1.41.$$

Hence, 2nd eigenvector of unit length is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1/1.41 \\ 1/1.41 \end{bmatrix} = \begin{bmatrix} -0.70 \\ 0.70 \end{bmatrix}$$

You sort the eigenvectors according to eigenvalues.

Hence, $$V^T = \begin{bmatrix} 0.70 & -0.70 \\ 0.70 & 0.70 \end{bmatrix}$$

Singular values are square root of eigenvalues and are sorted in descending order. It is a matrix of $m \times n$ where all elements are 0 except the diagonal elements.

$$\varepsilon = \begin{bmatrix} \sqrt{8} & 0 \\ 0 & \sqrt{2} \end{bmatrix} = \begin{bmatrix} 2.82 & 0 \\ 0 & 1.41 \end{bmatrix}$$

Hence, SVD of $\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$ can be written as following.

$$\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = U\varepsilon V^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2.82 & 0 \\ 0 & 1.41 \end{bmatrix} \times \begin{bmatrix} 0.70 & -0.70 \\ 0.70 & 0.70 \end{bmatrix}$$

Just for fun and confirmation, if you go ahead and multiply the matrices you got after decomposition, you will get the original matrix (very close values).

So, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2.82 & 0 \\ 0 & 1.41 \end{bmatrix} \times \begin{bmatrix} 0.70 & -070 \\ 0.70 & 0.70 \end{bmatrix}$ will give you $\begin{bmatrix} 1.974 & -1.974 \\ 0.987 & 0.987 \end{bmatrix}$ which is very close to the matrix that you started with $\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$.

Note here that signs of matrices U and V do not matter. You may get different signs. As long as the absolute values are correct, it is a correct SVD.

17. Compute the Linear Discriminant projection for the following two-dimensional dataset.

X1 = (x1, x2) = { (4,1), (2,4), (2,3), (3,6), (4,4) } and

X2 = (x1, x2) = { (9,10), (6,8), (9,5), (8,7), (10,8) }

https://www.youtube.com/watch?v=6M-8WEFsFjU

Step 1: Calculate means of both classes (μ_1, μ_2)

$$\mu_1 = \left\{ \frac{4+2+2+3+4}{5}, \frac{1+4+3+6+4}{5} \right\}$$

$$\mu_2 = \left\{ \frac{9+6+9+8+10}{5}, \frac{10+8+5+7+8}{5} \right\}$$

$$\mu_1 = \{3, 3.6\}$$

$$\mu_2 = \{8.4, 7.6\}$$

Step 2: Compute covariance matrix (S_1, S_2)

$$(x - \mu_1) = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix}$$

Calculating $S_1 = \sum (x_1 - \mu_1)(x_1 - \mu_1)^T$

$$\begin{bmatrix} 1 \\ -2.6 \end{bmatrix} \begin{bmatrix} 1 & -2.6 \end{bmatrix} = \begin{bmatrix} 1 & -2.6 \\ -2.6 & 6.76 \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ 0.4 \end{bmatrix} \begin{bmatrix} -1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & -0.4 \\ -0.4 & 0.16 \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ -0.6 \end{bmatrix} \begin{bmatrix} -1 & -0.6 \end{bmatrix} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 0.36 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 2.4 \end{bmatrix} \begin{bmatrix} 0 & 2.4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 5.76 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0.4 \end{bmatrix} \begin{bmatrix} 1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 0.16 \end{bmatrix}$$

Adding these 5 matrices and dividing by 5,

$$S_1 = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix}$$

Similarly for S_2,

$$(X_2 - \mu_2) = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}$$

$S_2 = \sum (x_2 - \mu_2)\,(x_2 - \mu_2)^T$

$$\begin{bmatrix} 0.6 \\ 2.4 \end{bmatrix} \begin{bmatrix} 0.6 & 2.4 \end{bmatrix} = \begin{bmatrix} 0.36 & 1.44 \\ 1.44 & 5.76 \end{bmatrix}$$

$$\begin{bmatrix} -2.4 \\ 0.4 \end{bmatrix} \begin{bmatrix} -2.4 & 0.4 \end{bmatrix} = \begin{bmatrix} 5.76 & -0.96 \\ -0.96 & 0.16 \end{bmatrix}$$

$$\begin{bmatrix} 0.6 \\ -2.6 \end{bmatrix} \begin{bmatrix} 0.6 & -2.6 \end{bmatrix} = \begin{bmatrix} 0.36 & -1.56 \\ -1.56 & 6.76 \end{bmatrix}$$

$$\begin{bmatrix} -0.4 \\ -0.6 \end{bmatrix} \begin{bmatrix} -0.4 & -0.6 \end{bmatrix} = \begin{bmatrix} 0.16 & 0.24 \\ 0.24 & 0.36 \end{bmatrix}$$

$$\begin{bmatrix} 1.6 \\ 0.4 \end{bmatrix} \begin{bmatrix} 1.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 2.56 & 0.64 \\ 0.64 & 0.16 \end{bmatrix}$$

Adding these 5 matrices and dividing by 5,

$$S_2 = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

Step 3: Compute within class scatter matrix (SW)

$$S_W = S_1 + S_2$$

$$S_W = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix} + \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

$$S_W = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.24 \end{bmatrix}$$

Step 4: Compute between class scatter matrix (SB)

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$\mu_1 - \mu_2 = \begin{bmatrix} 3 \\ 3.6 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} = \begin{bmatrix} -5.4 \\ -4 \end{bmatrix}$$

$$S_B = \begin{bmatrix} -5.4 \\ -4 \end{bmatrix} \begin{bmatrix} -5.4 & -4 \end{bmatrix} = \begin{bmatrix} 29.16 & 21.60 \\ 21.60 & 16 \end{bmatrix}$$

Step 5: Compute Eigenvalue (λ_1, λ_2) & Eigenvector (V)

$$S_W^{-1} S_B V = \lambda V$$

$$|S_W^{-1} S_B - \lambda I| = 0$$

$$\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.79 \end{bmatrix} - \lambda I = 0$$

$$\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.79 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 11.89 - \lambda & 8.81 \\ 5.08 & 3.79 - \lambda \end{bmatrix} = 0$$

$$(11.89 - \lambda)(3.79 - \lambda) - 44.75 = 0$$

$$45.06 - 11.89\lambda + \lambda^2 - 3.79\lambda - 44.75 = 0$$

$$\lambda^2 - 15.68\lambda + 0.31 = 0$$

$$\lambda_1 = 15.66 \quad \lambda_2 = 0.019$$

$$\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.79 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = 15.66 \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 0.91 \\ 0.39 \end{bmatrix}$$

Or directly written as

$$W^* = S_W^{-1}(M_1 - M_2)$$

$$= \begin{bmatrix} -0.91 & -0.39 \end{bmatrix}^T$$

Previously asked problems:

1. Decision tree construction using Gini index:

1. Explain the concept of decision tree. Consider the dataset given in a table below. The dataset has 3 features as Past trend, Open interest, Trading volume and one class label as Return. Compute the Gini index for all features and specify which node will be chosen as a root node in decision tree.

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

2. Create a decision tree using Gini Index to classify following dataset.

Sr. No.	Income	Age	Own Car
1	Very High	Young	Yes
2	High	Medium	Yes
3	Low	Young	No
4	High	Medium	Yes
5	Very High	Medium	Yes
6	Medium	Young	Yes
7	High	Old	Yes
8	Medium	Medium	No
9	Low	Medium	No
10	Low	Old	No
11	High	Young	Yes
12	Medium	Old	No

3. Create a decision tree using Gini Index to classify following dataset for profit.

Age	Competition	Type	Profit
old	Yes	software	down
old	No	software	Down
old	No	hardware	Down
mid	Yes	software	Down
mid	Yes	hardware	Down
mid	No	hardware	Up
mid	No	software	Up
new	Yes	software	Up
new	No	hardware	Up
new	no	software	Up

4. Consider the dataset given below with 3 features Colour, Wig, Num. Ears and one output variable Emotion.

Color	G	G	G	B	B	R	R	R	R
Wig	Y	N	N	N	N	N	N	N	Y
Num. Ears	2	2	2	2	2	2	2	2	3
Emotion	S	S	S	S	H	H	H	H	H

i) Find root node of decision tree using GINI index.

ii) Explain techniques that can be used to handle over fitting in decision trees?

2. Linear Regression:

1. Explain the concept of regression and enlist its types. A clinical trial gave the data for BMI and cholesterol level for 10 patients as shown in table below. Identify the machine learning method used to solve the above problem and predict the likely value of cholesterol level for someone who has BMI of 27.

BMI	17	21	24	28	14	16	19	22	15	18
Cholesterol	140	189	210	240	130	100	135	166	130	170

2. Consider the example below where the mass, y (grams), of a chemical is related to the time, x (seconds), for which the chemical reaction has been taking place according to the table. Find the equation of the regression line.

Time, x (seconds)	5	7	12	16	20
Mass, y (grams)	40	120	180	210	240