IML Final Paper solution

2Marks

d) Elaborate the terms training data, validation data, and test data.

1. Training Data

- **Definition:**
  The **training data** is the dataset used to train a machine learning model. It consists of input-output pairs where the model learns the relationship between the inputs (features) and the outputs (labels or target values).
- **Purpose:**
  Helps the model **learn patterns** and **adjust internal parameters** (like weights in neural networks).
- **Example:**
  In a house price prediction model, training data may include house features like size, location, and number of rooms, along with the actual prices.

2. Validation Data

- **Definition:**
  The **validation data** is a separate portion of the dataset used during the training process to **fine-tune** the model's hyperparameters (e.g., learning rate, number of layers).
- **Purpose:**
  Helps in:
  - **Model selection** (choosing the best-performing model).
  - **Hyperparameter tuning** without overfitting the training data.
  - **Early stopping** (ending training when performance on validation data stops improving).

3. Test Data

- **Definition:**
  The **test data** is an entirely separate dataset used **after training** is complete, to **evaluate the final model's performance**.
- **Purpose:**
  Provides an **unbiased estimate** of how well the model will perform on **new, unseen data**.
- **Example:**
  Continuing the house price example, test data includes new houses the model hasn't seen before, used to assess real-world accuracy.
- Here is the **difference between training data and testing data** in a clear **table format**:

Feature	Training Data	Testing Data
Purpose	Used to **train** the machine learning model	Used to **evaluate** the model's performance
Used During Training	✅ Yes	❌ No (used only after training)
Affects Model Weights	✅ Yes (used to adjust model parameters)	❌ No (only used for prediction/evaluation)
Size (usually)	Typically **larger portion** of the dataset (e.g., 70-80%)	Typically **smaller portion** (e.g., 20-30%)
Contains Labels	✅ Yes	✅ Yes (used to compare predictions with actual outcomes)
Goal	To help the model **learn patterns**	To **test generalization** to unseen data
Example Use	Learning relationships between input and output	Calculating accuracy, precision, recall, etc.

f) Consider a dataset with 1000 labeled images of handwritten digits. Which classification method would be more efficient, and why?

Recommended Method: Convolutional Neural Network (CNN)

✅ **Why CNN is Efficient:**

1. **Specialized for Image Data**:
   CNNs are designed to **automatically detect spatial patterns** like edges, shapes, and textures in images.
2. **Local Connectivity & Parameter Sharing**:
   CNNs reduce the number of parameters compared to fully connected networks, making them **computationally efficient** for image classification.
3. **High Accuracy in Digit Recognition Tasks**:
   CNNs have been **proven highly effective** for handwritten digit datasets like **MNIST**, which is very similar to your scenario.

Discuss overfitting and underfitting in Machine Learning models. Define these terms and explain their impact on model performance. How can they be addressed?

✅ **1. Overfitting**

Definition:

Overfitting occurs when a model learns not only the underlying patterns in the training data but also the **noise and irrelevant details**. It performs **very well on training data** but **poorly on test/validation data**.

Impact:

- High accuracy on training data

- Low accuracy on unseen (test) data
- Poor generalization

Causes:

- Model is too complex (too many parameters)
- Not enough training data
- Training too long

Solutions to Overfitting:

- Use **simpler models**
- Apply **regularization** (L1, L2)
- Use **cross-validation**
- **Early stopping** during training
- Use **more training data**
- Apply **dropout** (in neural networks)
- Perform **data augmentation**

✅ **2. Underfitting**

Definition:

Underfitting happens when a model is too simple to capture the underlying pattern of the data. It fails to learn from both training and test data.

Impact:

- Poor accuracy on training and test data
- Model fails to capture the complexity of the problem

Causes:

- Model is too simple (e.g., linear model for non-linear data)
- Not enough training time
- Inadequate features or poor data preprocessing

Solutions to Underfitting:

- Use a **more complex model**
- Train for **more epochs**
- Use **feature engineering** or **add more relevant features**
- Reduce **regularization strength**

e**) Define clustering in ML and give an example of its application.**

Clustering is an unsupervised machine learning task that involves grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups. Unlike supervised learning, there are no predefined labels or output variables for the data; the algorithm discovers inherent patterns and structures within the data itself. The goal is to find natural groupings or clusters within a dataset based on the similarity of features among the data points.

Example of its application: Customer Segmentation in Marketing:

A common application of clustering is in **customer segmentation** for marketing. A company might have a large database of customer information, including their purchase history, Browse behavior, demographics, and interactions with the company. By applying clustering algorithms (e.g., K-Means), the company can group customers into distinct segments.

For example, the clustering algorithm might identify:

- **High-Value Customers:** Those who make frequent and large purchases.
- **New Customers:** Those who have recently started purchasing.
- **Price-Sensitive Customers:** Those who primarily buy discounted items.
- **Loyal Customers:** Those who frequently purchase from the same brand or category.

b) How machine learning is useful in recommender system?

Machine learning powers recommender systems by learning user preferences and item characteristics from data (explicit and implicit feedback). It uses algorithms like collaborative filtering and content-based filtering to identify patterns, predict user interest in unviewed items, and handle large, sparse datasets, ultimately providing personalized and relevant recommendations.

Example: Here's a short example of how machine learning is used in YouTube's recommendation system:

- **Learning User Preferences:** YouTube's system analyzes your viewing history, searches, and engagement (likes, shares, watch time) to understand what you like.
- **Candidate Generation:** From billions of videos, machine learning models select a smaller subset (hundreds or thousands) that might interest you.
- **Scoring and Ranking:** Another model scores these candidate videos based on various factors and ranks them to select the top videos to show you.
- **Personalization:** The system continuously learns and adapts to your preferences, improving recommendations over time.

c) Why is multiple linear regression so called?

Multiple linear regression is so called because it involves **multiple independent variables** (or predictor variables) to predict a **single dependent variable** (or outcome variable).

- **Linear:** The "linear" part refers to the fact that the relationship between the dependent variable and each independent variable is assumed to be linear. This means the model

fits a straight line (or a hyperplane in higher dimensions) to the data. The equation of the model is a linear combination of the independent variables and their respective coefficients.

- **Multiple:** The "multiple" part signifies that there are **two or more** independent variables in the model. In contrast, "simple linear regression" uses only one independent variable.

d) Outline the significance Of optimal separating hyperplane in SVM?

✅ **Significance of Optimal Separating Hyperplane in SVM (Support Vector Machine)**

🔹 **What is a Separating Hyperplane?**

In SVM, a **hyperplane** is a decision boundary that separates data points of different classes in a feature space.

- In **2D**, the hyperplane is a **line**.
- In **3D**, it's a **plane**.
- In **higher dimensions**, it's called a **hyperplane**.

🧠 **What is the Optimal Separating Hyperplane?**

- It is the **best hyperplane** that **maximally separates** the data points of different classes.
- It is chosen such that it **maximizes the margin** — the distance between the hyperplane and the nearest data points from each class (called **support vectors**).

🎯 **Significance of the Optimal Separating Hyperplane:**

Point	Explanation
1. **Maximizes Margin**	Leads to better generalization and performance on unseen data.
2. **Robust Classification**	Less sensitive to small changes or noise in the data.
3. **Unique Solution**	For linearly separable data, SVM guarantees a unique optimal hyperplane.
4. **Depends on Support Vectors Only**	Makes the model efficient as only key data points determine the boundary.
5. **Good for High-Dimensional Data**	SVM with optimal hyperplane performs well even in complex, high-dimensional spaces.

e) Why we should use the random forest algorithm?

🔍 **Key Reasons to Use Random Forest:**

Reason	Explanation
1. High Accuracy	Combines many decision trees to produce more accurate and stable predictions.
2. Reduces Overfitting	By averaging multiple trees, it reduces the risk of overfitting common in single decision trees.
3. Handles Missing Values	Can maintain performance even if some data is missing.
4. Works with Categorical & Numerical Data	Suitable for various types of input features.
5. Feature Importance	Can measure which features are most important for prediction.
6. Robust to Noise & Outliers	Less affected by noisy data and outliers due to averaging.
7. Good Generalization	Performs well on both training and unseen test data.
8. Handles Large Datasets	Efficient for large datasets with high dimensionality.

b) What are the main 012jectivcs of machine learning?

Objective	Explanation
1. Prediction	Predict future outcomes based on historical data (e.g., stock prices, sales).
2. Classification	Categorize data into predefined classes (e.g., spam vs. not spam, disease vs. no disease).
3. Clustering	Group similar data points together without labels (e.g., customer segmentation).
4. Pattern Recognition	Detect patterns or trends in data (e.g., handwriting recognition, face detection).
5. Anomaly Detection	Identify unusual data points (e.g., fraud detection, network intrusion).
6. Recommendation	Suggest relevant items to users (e.g., movies on Netflix, products on Amazon).
7. Automation of Decision-Making	Help machines make autonomous decisions (e.g., self-driving cars, robots).

d) What is K value in KNN?

✅ **What is the K Value in KNN?**

In the **K-Nearest Neighbors (KNN)** algorithm:

- **K** is the **number of nearest neighbors** to consider when making a prediction about the class (for classification) or value (for regression) of a new data point.

🔍 **Role of K:**

K Value	Effect on KNN
Small K (e.g., 1 or 3)	Model is sensitive to noise and may overfit — decision boundary is very flexible.
Large K	Model is smoother, less sensitive to noise but may underfit — losing detail.

📝 **How KNN Uses K:**

- For a new data point, the algorithm finds the **K closest points** (neighbors) in the training data based on a distance metric (like Euclidean distance).
- It **assigns the class** (or predicts the value) based on the **majority vote** (classification) or average (regression) of these K neighbors.

How do you handle missing or corrupted data in a dataset?

Method	Description	When to Use
1. Remove Rows or Columns	Delete rows with missing/corrupted values or entire columns if too many values are missing.	When missing data is small and random.
2. Imputation	Fill missing values with estimated values like:	Most common approach for moderate missingness.
- Mean/Median/Mode Imputation	Replace missing values with mean (numeric), median (numeric), or mode (categorical) of that feature.	When data is missing at random and numerical/categorical data is available.
- Forward/Backward Fill	Use previous or next valid value (common in time series).	For time-dependent data.
- Predictive Imputation	Use machine learning models to predict missing values based on other features.	When complex relationships exist in data.
3. Use Algorithms Handling Missing Data	Some algorithms (like XGBoost, Random Forest) can handle missing data internally.	When using robust models that support it.

Method	Description	When to Use
4. Flag Missing Data	Create a new binary feature indicating whether data was missing or not.	To allow the model to learn missingness patterns.
5. Data Correction	Detect and fix corrupted data based on domain knowledge or validation rules.	When corrupted data is identifiable and fixable.

d) What are unsupervised Machine learning techniques?

1.clusting 2.neural network

What is Principal Component Analysis?

Principal Component Analysis (PCA) is a popular **dimensionality reduction** technique used in machine learning and data analysis.

🔍 **What PCA Does:**

- PCA transforms a large set of correlated variables into a smaller set of **uncorrelated variables** called **principal components**.
- These principal components capture the **most important information (variance)** in the data.
- The first principal component captures the most variance, the second captures the next most, and so on.
- This reduces the data's dimensionality while preserving as much variability as possible.

4Marks

Q3.Discuss the concept of Dimensionality Reduction and its significance in Machine Learning.

Dimensionality reduction is the process of **reducing the number of input variables (features)** in a dataset while **preserving as much important information as possible**.

- It transforms high-dimensional data into a lower-dimensional form.
- Helps simplify the data without losing the essence of the original information.

🎯 **Why Dimensionality Reduction is Important:**

Significance	Explanation
1. Reduces Computational Cost	Fewer features mean faster training and less memory usage in models.

Significance	Explanation
2. Avoids the Curse of Dimensionality	High-dimensional data can make learning algorithms less effective and more prone to overfitting.
3. Improves Model Performance	Removes irrelevant or redundant features, helping models generalize better.
4. Enables Visualization	Reduces data to 2D or 3D for easier visualization and understanding of patterns.
5. Removes Noise	Reduces noisy or redundant data that can confuse models.

Q4. Compare K-Nearest Neighbors and Support Vector Machine in classification tasks.7

Aspect	K-Nearest Neighbors (KNN)	Support Vector Machine (SVM)
Type of Algorithm	Instance-based, lazy learning	Model-based, eager learning
How It Works	Classifies based on majority class among K nearest neighbors	Finds an optimal hyperplane that separates classes with maximum margin
Training Phase	No explicit training; just stores data	Training involves solving optimization to find separating hyperplane
Prediction Phase	Calculates distance to all training points; slow for large data	Uses the learned hyperplane; usually faster at prediction
Handling Non-linear Data	Uses distance metric, can use weighted voting but less effective	Uses kernel trick (e.g., RBF, polynomial) to handle non-linear boundaries
Parameter Tuning	Number of neighbors (K), distance metric	Kernel type, regularization parameter (C), kernel parameters
Sensitivity to Noise	Sensitive, especially with small K	Less sensitive due to margin maximization and regularization
Scalability	Poor for large datasets because of distance computations	Better scalability with optimized solvers
Interpretability	Simple to understand; “neighbors decide the class”	More complex; decision boundary and support vectors define model
Use Cases	Simple problems, smaller datasets, recommendation systems	Complex classification tasks, high-dimensional data

Q5.A retail store wants to segment its customers based on their buying patterns. Which technique would you suggest? Justify your choice.

🔍 **Recommended Technique: Clustering (Unsupervised Learning)**

- Clustering groups customers into distinct segments based on similarities in their buying patterns without using any predefined labels.
- Common clustering algorithms include **K-Means**, **Hierarchical Clustering**, and **DBSCAN**.

🎯 **Why Clustering for Customer Segmentation?**

Reason	Explanation
No labeled data required	Customer segments are usually unknown beforehand (unsupervised).
Identifies natural groups	Groups customers with similar buying behavior automatically.
Helps target marketing strategies	Enables personalized promotions, improving sales and satisfaction.
Handles large datasets	Efficient algorithms like K-Means can handle big customer data.
Insight into buying patterns	Reveals hidden patterns like frequent buyers, discount seekers, etc.

Q7. An e-commerce platform uses Machine Learning for product recommendations. Explain how a Self-Organizing Map (SOM) could be used for this task.

🔍 **What is a Self-Organizing Map (SOM)?**

- SOM is an **unsupervised neural network** used for **clustering and visualization**.
- It maps high-dimensional data onto a low-dimensional (usually 2D) grid, preserving the topological relationships.
- Similar data points are mapped close together on the grid.

🎯 **How SOM Can Be Used for Product Recommendations:**

1. **Input Data:**
    - Customer behavior data like purchase history, browsing patterns, product ratings, and product features.
    - Each customer or product is represented as a high-dimensional vector.
2. **Training SOM:**

- The SOM learns to organize customers or products into clusters based on similarity.
- Similar customers or products get mapped near each other on the SOM grid.

3. **Cluster Formation:**
   - Identify clusters of customers with similar buying patterns or clusters of similar products.
   - For example, customers who buy similar types of electronics will be grouped together.
4. **Making Recommendations:**
   - For a given customer, find their position on the SOM.
   - Recommend products favored by other customers or located in the same cluster/neighborhood on the map.
   - This helps suggest products that are likely relevant based on similar user profiles.

Q2. A real estate company needs to predict house prices based on features like bedrooms,size, location, and age. They have historical data available. Elaborate on which regression model would you choose and why?

🎯 **Recommended Model: Multiple Linear Regression**

🔍 **Why Multiple Linear Regression?**

Reason	Explanation
Multiple features (independent variables)	Since price depends on several variables, multiple linear regression fits best.
Interpretability	Easy to understand the impact of each feature (e.g., how size affects price).
Efficient with structured data	Works well with numerical/tabular data like real estate records.
Baseline model	Acts as a strong starting point before trying more complex models.

Q3. A company aims to recognize handwritten digits (0-9) from scanned documents using a labelled dataset. Which neural network would you choose and why? Also, elaborate on the network architecture and training process.

🧠 **Recommended Neural Network: Convolutional Neural Network (CNN)**

🎯 **Why CNN?**

Reason	Explanation
Designed for image data	CNNs are ideal for processing pixel data from scanned images.
Captures spatial features	Detects edges, shapes, and patterns (important for recognizing digits).
Reduces parameters	Uses shared weights and pooling, making it more efficient than dense networks.
Proven success	CNNs are the standard in handwritten digit recognition tasks (e.g., MNIST).

Q7. Compare and contrast the K-means and hierarchical clustering algorithms, discussing their advantages and disadvantages.

✅ **K-Means vs. Hierarchical Clustering**

Aspect	**K-Means Clustering**	**Hierarchical Clustering**
Approach	Partition-based clustering	Tree-based (hierarchical) clustering
Cluster Shape	Assumes spherical (equal size) clusters	Can capture complex and nested clusters
Number of Clusters	Must be specified in advance (K)	No need to specify initially; you can cut the tree at any level
Scalability	Efficient for large datasets	Computationally expensive for large datasets
Algorithm Type	Iterative refinement (updates centroids and assigns points)	Agglomerative (bottom-up) or divisive (top-down)
Time Complexity	O(n * k * i) — linear with data size (n), clusters (k), iterations (i)	O(n^2) — due to pairwise distance computations
Memory Usage	Low memory usage	High memory usage
Stability	Sensitive to initial centroid selection; may give different results	More stable; deterministic results (especially in agglomerative)
Interpretability	Harder to interpret clustering process	Dendrogram makes the process visually interpretable

✅ **Advantages**

K-Means

✅ Fast and efficient on large datasets

✅ Easy to implement

✅ Works well when clusters are well-separated and balanced

Hierarchical Clustering

✅ No need to predefine number of clusters

✅ Dendrogram gives insight into data structure

✅ Can capture complex relationships between data points

❌ **Disadvantages**

K-Means

❌ Requires K to be set manually

❌ Sensitive to outliers and noise

❌ Works poorly with non-spherical or overlapping clusters

Hierarchical Clustering

❌ High computational cost on large datasets

❌ Once merged/split, cannot undo decisions

❌ May be sensitive to distance metrics and linkage criteria

Q2. Explain the difference between linear and logistics regression with example.

✅ **Linear Regression vs Logistic Regression**

Aspect	**Linear Regression**	**Logistic Regression**
Purpose	Predicts a **continuous numerical value**.	Predicts a **categorical value (usually binary)**.
Output Range	Any real number from -∞ to +∞.	Output is between 0 and 1, representing a probability.
Type of Problem	Regression problem	Classification problem
Linearity	Assumes linear relationship between input and output	Assumes linear relationship between input and **log-odds** of output
Example	Predicting **house price** based on size, location, etc.	Predicting if an email is **spam or not spam** based on word frequency, etc.
Output Interpretation	Direct numeric value (e.g., ₹300,000 for house price)	Probability (e.g., 0.85 probability email is spam, interpreted as class = "spam")
Algorithm Goal	Minimize Mean Squared Error (MSE)	Maximize Likelihood Function

Q6. What are the benefits of pruning in decision tree induction? Explain different approaches to tree pruning?

🔷 What is Pruning in Decision Trees?

Pruning is the process of **removing unnecessary branches or nodes** from a decision tree that do not provide significant power in classifying instances. It helps to simplify the model and prevent overfitting.

✅ Benefits of Pruning:

Benefit	Explanation
Reduces Overfitting	Eliminates parts of the tree that fit noise or anomalies in the training data.
Improves Accuracy on Test Data	Leads to better generalization on unseen (test) data.
Simplifies the Model	Makes the tree more interpretable and easier to understand.
Reduces Complexity and Size	Smaller trees are faster to evaluate and consume less memory.
Increases Prediction Speed	Shallower trees reduce computation time for predictions.

🔷 Types of Pruning Approaches:

1. Pre-Pruning (Early Stopping):

- Stops tree growth **early** (before it becomes too complex).
- A split is made **only if** it improves performance beyond a threshold.

Methods:

- Limit **tree depth**.
- Set a **minimum number of samples per node**.
- Require a **minimum improvement in impurity** (e.g., Gini or entropy).

Pros: Fast and simple
Cons: May miss important patterns (underfitting)

2. Post-Pruning (Backward Pruning):

- Tree is built **completely first**, then unnecessary branches are **cut back**.
- Subtrees are replaced by leaf nodes if it improves or doesn't reduce accuracy.

Methods:

- **Reduced Error Pruning:**
    - Remove nodes and check validation set accuracy.
    - If accuracy doesn't decrease, keep the change.

- **Cost Complexity Pruning (a.k.a. weakest link pruning):**
    - Adds a penalty for tree complexity (used in CART).
    - Prune nodes based on a trade-off between error and complexity.

Pros: More reliable, better accuracy
Cons: Computationally expensive

Suppose 10,000 patients get tested for flu. Out of them, 9,000 are actually healthy and 1,000 are actually sick.For the sick people, the test was positive for 620 patients and negative for 380 patients. For the healthy people, the same test was positive for 180 patients and negative for 8,820 patients. Construct a confusion matrix for the data and compute the precision and recall.

12 marks

Explain the concept of confusion matrix. Assume 1000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data.

✅ **Given Data:**

- Total patients = **10,000**
- **Actually sick** = 1000
    - Test **positive** for sick = 620 → **True Positives (TP)**
    - Test **negative** for sick = 380 → **False Negatives (FN)**
- **Actually healthy** = 9000
    - Test **positive** for healthy = 180 → **False Positives (FP)**
    - Test **negative** for healthy = 8820 → **True Negatives (TN)**

📊 **Confusion Matrix:**

	Predicted: Positive	Predicted: Negative
Actual: Sick	TP = 620	FN = 380
Actual: Healthy	FP = 180	TN = 8820

📈 **Precision and Recall Calculations:**

- **Precision** (Positive Predictive Value):

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{620}{620 + 180} = \frac{620}{800} = 0.775$$

  - Precision = **77.5%**

Describe Support Vector Machine (SVM). How is the vector developed in the training pattern?

✅ **How is the vector developed in the training pattern?**

1. **Input Data**: The SVM receives labeled training data points (features with class labels).

2. **Linear Separation**: It tries to find a straight line (in 2D) or a hyperplane (in higher dimensions) that separates the data points of one class from the other with the **maximum margin**.
3. **Support Vectors**: The vectors (data points) that lie closest to the decision boundary (hyperplane) are chosen as **support vectors**. They play a critical role in defining the position and orientation of the hyperplane.
4. **Optimization**: SVM uses a mathematical optimization technique to **maximize the margin** and **minimize classification error**.
5. **Kernels (for Non-linear Data)**: If the data is not linearly separable, SVM applies a **kernel trick** (e.g., polynomial, RBF) to map the data into a higher-dimensional space where a linear separator can be found.

Q4. Define Multilayer Perceptron. How does a Multilayer Perceptron solve the XOR problem?

✅ **XOR Problem and Why Single-Layer Perceptron Fails:**

The **XOR (exclusive OR)** function returns true (1) when inputs are different and false (0) when inputs are the same.

Input X1	Input X2	XOR Output
0	0	0
0	1	1
1	0	1
1	1	0

The XOR function is **not linearly separable**, meaning it cannot be solved by a **single-layer perceptron** using a straight line.`

✅ **How MLP Solves XOR Problem:**

A **Multilayer Perceptron** can solve the XOR problem by using **one hidden layer with non-linear activation functions**.

Architecture to solve XOR:

- **Input Layer**: 2 neurons (X1 and X2)
- **Hidden Layer**: At least 2 neurons with **non-linear activation** (e.g., sigmoid or ReLU)
- **Output Layer**: 1 neuron to produce the final XOR output

How it works:

- The hidden neurons learn to **transform the input space** into a form where the XOR function becomes **linearly separable**.

- The output neuron then learns to classify based on the transformed representation.

Q5. Discuss the major drawbacks of K-Nearest Neighbour (KNN) learning algorithm and how it can be corrected.

✅ **Drawbacks of K-Nearest Neighbour (KNN):**

1. **High Computational Cost:**
   - KNN is a **lazy learner**, meaning it stores all training data and performs computation at prediction time.
   - As dataset size grows, prediction becomes **slower and memory-intensive**.
2. **Curse of Dimensionality:**
   - In high-dimensional spaces, **distance measures become less meaningful**.
   - The algorithm may perform poorly when irrelevant or many features exist.
3. **Sensitive to Noisy Data and Outliers:**
   - KNN can be heavily affected by **noisy data or outliers**, especially if k is small.
4. **Feature Scaling Required:**
   - Features with larger scales can **dominate the distance calculation**, leading to biased results.
5. **Choosing Optimal K Value:**
   - A too-small k leads to **overfitting**, while a too-large k may result in **underfitting**.
   - There's **no fixed rule** to determine the best value for k.
6. **Imbalanced Data Issues:**
   - If one class dominates the dataset, KNN may **bias toward the majority class**, causing poor performance for the minority class.

✅ **How to Correct or Improve KNN:**

1. **Reduce Dimensionality:**
   - Apply techniques like **PCA (Principal Component Analysis)** to reduce features and focus on the most relevant ones.
2. **Feature Scaling/Normalization:**
   - Use **Min-Max Scaling** or **Standardization** so all features contribute equally to distance calculations.
3. **Use Efficient Data Structures:**

- Implement data structures like **KD-Trees** or **Ball Trees** for faster nearest neighbor search.

4. **Optimize the Value of K:**
   - Use **cross-validation** to find the best k that gives the lowest error rate.
5. **Weighting Neighbors:**
   - Assign **weights to neighbors** based on their distance (e.g., closer points have more influence).
6. **Handle Imbalanced Data:**
   - Use **resampling techniques** (like SMOTE) or **weighted voting** to improve performance on imbalanced datasets.

Q2.What is 'training Set' and 'test Set' in a Machine Learning Model? How Much Data

Will You Allocate for Your Training, Validation, and Test Sets? Justify.

🔹 **Training Set:**

- A **training set** is the portion of the dataset used to **train the machine learning model**.
- The model learns the relationships, patterns, or rules from this data.
- It includes both **input features** and **labels** (in supervised learning).

🔸 *Example:*
In a house price prediction task, the training set may contain house features (bedrooms, size, location) and their known prices.

🔹 **Test Set:**

- A **test set** is used **after training** to evaluate the model's performance on **unseen data**.
- It assesses how well the model **generalizes** beyond the training data.
- It must never be used during the training process to avoid data leakage.

✅ **Data Allocation for Training, Validation, and Test Sets**

Dataset	Purpose	Typical Proportion
Training Set	To train the model and learn patterns	**60–70%**
Validation Set	To tune hyperparameters and prevent overfitting	**15–20%**
Test Set	To evaluate final model performance	**15–20%**

🔍 **Justification:**

- **Training Set (60–70%)**: A large portion is needed so the model can learn effectively.

- **Validation Set (15–20%)**: Used for **model selection** and **hyperparameter tuning**, helping to improve generalization.
- **Test Set (15–20%)**: Ensures **unbiased evaluation** on data the model has never seen before.

Explain the Confusion Matrix with Respect to Machine Learning Algorithm

✅ Confusion Matrix in Machine Learning

A **Confusion Matrix** is a performance measurement tool used in **classification problems**. It compares the actual (true) labels with the predicted labels produced by the model, allowing us to evaluate the model's accuracy and identify specific types of errors.

📊 Structure of a Confusion Matrix (for Binary Classification):

	Predicted: Positive	**Predicted: Negative**
Actual: Positive	True Positive (TP)	False Negative (FN)
Actual: Negative	False Positive (FP)	True Negative (TN)

🔍 Terminology Explained:

- **True Positive (TP):** The model correctly predicted the positive class.
- **False Positive (FP):** The model incorrectly predicted positive for a negative sample (Type I Error).
- **False Negative (FN):** The model missed a positive case and predicted it as negative (Type II Error).
- **True Negative (TN):** The model correctly predicted the negative class.

📐 Common Evaluation Metrics from the Confusion Matrix:

1. **Accuracy** – Overall correctness

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

2. **Precision** – Correctness of positive predictions

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

3. **Recall (Sensitivity)** – Ability to detect positives

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

4. **F1-Score** – Harmonic mean of precision and recall

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

c) What are the main types of classification algorithms used in machine learning? Provide examples of each type.

- 1. Logistic Regression

- Type: Linear Classification
- Description: Predicts the probability of a binary outcome using a logistic function.
- Use Case: Email spam detection, disease prediction.
- Example: Predicting whether a student will pass or fail based on study hours.

◆ 2. Decision Tree

- Type: Tree-Based Model
- Description: Splits data into branches to make decisions based on feature values.
- Use Case: Loan approval, credit risk classification.
- Example: Classifying loan applicants as approved or rejected based on income and credit score.

◆ 3. Random Forest

- Type: Ensemble Learning
- Description: Combines multiple decision trees to improve accuracy and reduce overfitting.
- Use Case: Customer churn prediction, fraud detection.
- Example: Classifying if a transaction is fraudulent or legitimate.

◆ 4. K-Nearest Neighbors (KNN)

- Type: Instance-Based Learning
- Description: Classifies data based on the majority class among the k-nearest neighbors.
- Use Case: Handwriting recognition, recommendation systems.
- Example: Recognizing a digit by comparing with the closest known digits.

◆ 5. Support Vector Machine (SVM)

- Type: Margin-Based Classification
- Description: Finds the optimal hyperplane that separates different classes.
- Use Case: Face detection, text classification.
- Example: Classifying emails as spam or not spam using SVM.

◆ 6. Naive Bayes

- Type: Probabilistic Classifier
- Description: Uses Bayes' theorem with the assumption of feature independence.
- Use Case: Sentiment analysis, document classification.

- Example: Classifying movie reviews as positive or negative.

◆ 7. Neural Networks

- Type: Deep Learning
- Description: Mimics the human brain using layers of interconnected neurons.
- Use Case: Image and speech recognition.
- Example: Recognizing objects in images (cats, dogs, etc.).

12marks
Discuss the various applications of ML across different domains and elaborate on why ML is considered the future of many industries.

Various Applications of ML Across Different Domains:

1. Healthcare:

- **Disease Diagnosis and Prediction:** ML algorithms analyze medical images (X-rays, MRIs, CT scans) to detect anomalies (tumors, lesions) with high accuracy, assisting radiologists. They can also predict disease onset based on patient data, genetic information, and lifestyle factors.
- **Drug Discovery and Development:** Accelerates the identification of potential drug candidates by analyzing complex biological and chemical data, predicting molecular interactions, and optimizing drug design.
- **Personalized Medicine:** Creates tailored treatment plans by combining individual patient health data with predictive analytics to optimize drug dosages and therapies.
- **Hospital Management:** Optimizes resource allocation, predicts patient admissions, and improves operational efficiency within healthcare facilities.
- **Electronic Health Records (EHR) Analysis:** Extracts insights from unstructured clinical notes and patient histories for research and improved care.

2. Finance:

- **Fraud Detection:** Real-time analysis of transaction patterns and customer behavior to identify and prevent fraudulent activities (e.g., credit card fraud, money laundering).
- **Algorithmic Trading:** ML models analyze market data to predict stock price movements and execute trades automatically at high speeds.
- **Credit Scoring and Risk Assessment:** More accurately assesses creditworthiness and loan default risk by analyzing vast amounts of financial history and macroeconomic indicators.
- **Personalized Financial Advice (Robo-Advisors):** Provides automated, data-driven investment recommendations and portfolio management based on individual risk tolerance and financial goals.

- **Customer Service:** AI-powered chatbots and virtual assistants handle customer inquiries, streamline banking processes, and enhance the overall customer experience.

3. Retail and E-commerce:

- **Recommendation Systems:** Powers personalized product recommendations on e-commerce platforms (e.g., "Customers who bought this also bought..."), significantly boosting sales and user engagement.
- **Demand Forecasting:** Predicts future customer demand for products, optimizing inventory management, reducing waste, and improving supply chain efficiency.
- **Dynamic Pricing:** Adjusts product prices in real-time based on factors like demand, competitor prices, inventory levels, and seasonality to maximize revenue.
- **Customer Segmentation:** Groups customers based on purchasing behavior, preferences, and demographics for highly targeted marketing campaigns.
- **Fraud Detection:** Identifies suspicious transactions and abnormal customer behavior to prevent retail fraud and protect both businesses and consumers.

4. Automotive Industry:

- **Autonomous Driving:** The core technology behind self-driving cars, enabling vehicles to perceive their environment, navigate, detect obstacles, and make real-time decisions using sensors, cameras, and ML algorithms.
- **Predictive Maintenance:** Analyzes vehicle sensor data (e.g., engine temperature, fuel usage, tire pressure) to predict potential component failures, allowing for proactive maintenance and reducing downtime.
- **Driver Assistance Systems (ADAS):** Powers features like adaptive cruise control, lane-keeping assist, automatic emergency braking, and blind-spot detection.
- **Manufacturing and Quality Control:** Optimizes production lines, performs automated inspections of parts and vehicles with high precision, and detects defects early.
- **Personalized Driving Experience:** Learns driver preferences for climate control, seat positions, infotainment, and even driving style.

Compare a) AI and ML b) Clustering and Classification c) Supervised and unsupervised learning, elaborate each comparison with some suitable example of each.

Aspect	**Clustering**	**Classification**
Type of Learning	Unsupervised Learning	Supervised Learning
Purpose	Grouping data into clusters based on similarity	Assigning predefined labels to data points

Aspect	Clustering	Classification
Input Data	Unlabeled data	Labeled data
Output	Groups or clusters of similar data points	Predicted class or category for each input
Goal	Discover hidden patterns or structure in data	Predict class labels based on training data
Examples	Segmenting customers into groups based on buying behavior	Email spam detection: classifying emails as "spam" or "not spam"
Common Algorithms	K-Means, Hierarchical Clustering, DBSCAN	Decision Trees, Support Vector Machines, Random Forest
Evaluation Metrics	Silhouette Score, Davies-Bouldin Index (unsupervised metrics)	Accuracy, Precision, Recall, F1-score (supervised metrics)
Nature of Labels	No prior labels; clusters are formed from data	Predefined class labels used during training
Use Cases	Market segmentation, image segmentation, anomaly detection	Fraud detection, sentiment analysis, medical diagnosis

Compare a) K means clustering with hierarchical Clustering techniques b) instance based learning vs. model based learning.

Aspect	Instance-Based Learning	Model-Based Learning
Definition	Learns by storing training instances and makes predictions using those stored examples directly.	Learns a general model (like equations or rules) from training data and uses it for prediction.
Approach	Lazy learning — little or no training phase; generalization happens at query time.	Eager learning — builds a model during training, then uses it for fast predictions.
Examples	K-Nearest Neighbors (KNN), Locally Weighted Regression	Linear Regression, Decision Trees, Support Vector Machines, Neural Networks
Training Time	Low (just storing data)	Usually high (building a model involves optimization or parameter estimation)
Prediction Time	High (needs to compare query with many instances)	Low (apply the learned model directly)

Aspect	Instance-Based Learning	Model-Based Learning
Memory Requirement	High (stores all training data)	Low to moderate (stores model parameters only)
Handling Noise	Sensitive to noisy data as predictions rely on raw data	Can generalize better by learning underlying patterns
Flexibility	Can model complex decision boundaries by using all instances	May be limited by the model structure or assumptions
Generalization	Generalizes locally based on nearby instances	Generalizes globally using a learned model

Describe the significance of Kernel functions in S V M. List any two kernel functions.

List the advantages of SVM and how optimal Hyperplane differ from Hyperplane.

Significance of Kernel Functions in SVM

Support Vector Machines (SVMs) try to find a hyperplane that best separates different classes. However, when the data is **not linearly separable** in the original input space, SVM uses **kernel functions** to map the input data into a higher-dimensional feature space where it *may* become linearly separable.

- **Kernel functions** allow SVM to operate in this high-dimensional space **without explicitly computing** the coordinates of the data in that space (known as the "kernel trick").
- This saves computation and enables SVM to efficiently solve complex, non-linear classification problems.
- Kernels define the similarity measure between pairs of data points in this new space.

Two Common Kernel Functions

1. **Linear Kernel**
   $$K(x, y) = x^\top y$$
   - Suitable for linearly separable data.
   - No mapping to higher dimensions; works in original feature space.
2. **Radial Basis Function (RBF) Kernel (Gaussian Kernel)**
   $$K(x, y) = \exp\left(-\gamma \|x - y\|^2\right)$$
   - Maps data to infinite-dimensional space.
   - Very flexible, widely used for non-linear problems.

Other kernels include Polynomial Kernel, Sigmoid Kernel, etc.

Advantages of SVM

- **Effective in high-dimensional spaces**: Works well when the number of features is large compared to samples.
- **Memory efficient**: Uses a subset of training points (support vectors) for decision making.
- **Robust to overfitting**: Especially with the right choice of kernel and regularization parameter.
- **Works well with clear margin of separation** between classes.
- **Versatile**: Can be customized with different kernels for various data types.

Difference between Optimal Hyperplane and Hyperplane

Aspect	**Hyperplane**	**Optimal Hyperplane**
Definition	Any decision boundary that separates classes in feature space.	The hyperplane that **maximizes the margin** between classes.
Margin	May have small or no margin between classes.	Has the **largest possible margin** (distance) from nearest data points of any class.
Performance	May not generalize well; could lead to misclassification.	Generalizes better with minimal classification error on unseen data.
Support Vectors	May or may not be defined by support vectors.	Defined by support vectors closest to the hyperplane.
Goal	Just separate the classes.	Find the **best separation** with maximum margin for better generalization.

Draw a decision tree for the following set of training examples. Do we require feature scaling for decision trees? Is it possible to have more than one decision tree for same training sample? Explain.

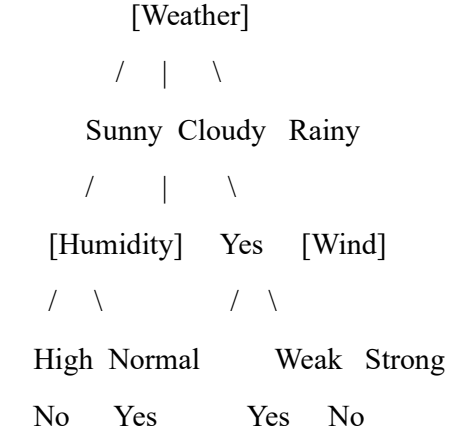
Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

1. Build a Decision Tree

We'll use the **ID3 algorithm** based on **Information Gain** to decide splits. Here's the dataset summarized:

Day	Weather	Temperature	Humidity	Wind	Play
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

- **Decision Tree**



2. Do we require feature scaling for decision trees?

No, feature scaling is **not required** for decision trees. This is because:

- Decision trees **do not use distance metrics** (unlike KNN or SVM).
- They split data based on thresholds or categories.

3. Is it possible to have more than one decision tree for the same training sample?

Yes, it is possible. This can happen when:

- Multiple features yield **equal information gain**.
- A **different tie-breaking rule** leads to different tree structures.
- Ensemble methods like **Random Forests** build multiple trees using subsets of data and features.

c. What is the stopping criteria of decision tree?

Pure Leaf Nodes (Zero Entropy)

- If all samples in a node belong to the same class, the node becomes a leaf.
- **Example:** All samples = "Yes" → No further split needed.

No Remaining Features

- If there are no more features to split on, the tree stops.
- The majority class in that node is assigned as the output.

Minimum Number of Samples per Node

- Splitting stops if a node has fewer than a threshold number of samples.
- This threshold is often called min_samples_split or min_samples_leaf.

Maximum Tree Depth

- A user-defined limit on how deep the tree can grow (e.g., max_depth = 5).
- Prevents excessive growth and overfitting.

Information Gain Threshold

- If the best split gives **very low information gain**, it might not be worth splitting.
- The split is made only if the information gain exceeds a minimum threshold (e.g., min_impurity_decrease in some libraries).

Early Stopping (Validation-based)

- Used in practical implementations: the tree stops growing if performance on a validation set stops improving.