**PVP 23**

Code: 23AM3401

**II B.Tech - II Semester – Regular / Supplementary Examinations**
**APRIL 2026**

# MACHINE LEARNING
## (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

## PART – A

| | | BL | CO |
|---|---|---|---|
| 1.a) | Explain the role of feature selection in Machine Learning. | L2 | CO1 |
| 1.b) | Define Accuracy. | L2 | CO1 |
| 1.c) | What is sigmoid function. | L2 | CO1 |
| 1.d) | Write the applications of Regression. | L2 | CO1 |
| 1.e) | Define conditional Probability. | L2 | CO1 |
| 1.f) | Define Entropy. | L2 | CO1 |
| 1.g) | Define Perceptron. | L2 | CO1 |
| 1.h) | Write the applications of SVM. | L2 | CO1 |
| 1.i) | Explain Complete Linkage. | L2 | CO1 |
| 1.j) | Write the measures of similarity. | L2 | CO1 |

## PART – B

<table>
<tr><th></th><th></th><th></th><th>BL</th><th>CO</th><th>Max. Marks</th></tr>
<tr><td colspan="6">UNIT-I</td></tr>
<tr><td rowspan="2">2</td><td>a)</td><td>Discuss various algorithms used in supervised and unsupervised learning with their applications.</td><td>L2</td><td>CO1</td><td>5 M</td></tr>
<tr><td>b)</td><td>What are the major challenges in training machine learning models? Explain with examples.</td><td>L2</td><td>CO1</td><td>5 M</td></tr>
<tr><td colspan="6">OR</td></tr>
<tr><td rowspan="2">3</td><td>a)</td><td>Describe five real-world applications of machine learning in different domains.</td><td>L2</td><td>CO1</td><td>5 M</td></tr>
<tr><td>b)</td><td>Explain the key steps involved in an end-to-end machine learning project.</td><td>L2</td><td>CO1</td><td>5 M</td></tr>
<tr><td colspan="6">UNIT-II</td></tr>
<tr><td>4</td><td colspan="2">Discuss about working process of linear regression and Predict the price of a 1700 sq. ft. House for the following data
<table>
<tr><th></th><th>House size(sq.ft)</th><th>price</th></tr>
<tr><td>1</td><td>1000</td><td>150</td></tr>
<tr><td>2</td><td>1200</td><td>180</td></tr>
<tr><td>3</td><td>1500</td><td>200</td></tr>
<tr><td>4</td><td>1800</td><td>240</td></tr>
<tr><td>5</td><td>2000</td><td>260</td></tr>
</table></td><td>L3</td><td>CO2</td><td>10 M</td></tr>
<tr><td colspan="6">OR</td></tr>
</table>

<table>
<tr><td>5</td><td>a)</td><td>Apply Logistic Regression and Linear Regression for classification tasks.</td><td>L3</td><td>CO2</td><td>5 M</td></tr>
<tr><td></td><td>b)</td><td>Interpret the algorithm used for Non-Linear Regression and how it works.</td><td>L3</td><td>CO2</td><td>5 M</td></tr>
<tr><td colspan="6">UNIT-III</td></tr>
<tr><td>6</td><td colspan="3"><table>
<tr><th>Contains “offer” (Yes/No)</th><th>Sender in Contacts (Yes/No)</th><th>Email Length (Short/Long)</th><th>Spam (Yes/No)</th></tr>
<tr><td>Yes</td><td>No</td><td>Short</td><td>Yes</td></tr>
<tr><td>Yes</td><td>Yes</td><td>Long</td><td>No</td></tr>
<tr><td>No</td><td>No</td><td>Long</td><td>Yes</td></tr>
<tr><td>No</td><td>Yes</td><td>Short</td><td>No</td></tr>
<tr><td>Yes</td><td>No</td><td>Long</td><td>Yes</td></tr>
</table>
i. Build a Decision Tree model to classify emails as spam or not.<br>
ii. Predict whether an email with "Offer = No, Sender in Contacts = No, Email Length = Short" is spam.<br>
iii. Compute the Information Gain for splitting on "Contains Offer".</td><td>L3</td><td>CO2</td><td>10 M</td></tr>
<tr><td colspan="6">OR</td></tr>
<tr><td>7</td><td>a)</td><td>Use different distance metrics in KNN, such as Euclidean, Manhattan, and Minkowski distances. How does the choice of distance metric impact classification performance?</td><td>L3</td><td>CO2</td><td>5 M</td></tr>
<tr><td></td><td>b)</td><td>Illustrate the step-by-step process of constructing a Decision Tree using the ID3.</td><td>L3</td><td>CO2</td><td>5 M</td></tr>
</table>

| | | | | | |
|---|---|---|---|---|---|
| **UNIT-IV** | | | | | |
| 8 | a) | Compare and contrast biological neuron and artificial neural networks. | L4 | CO4 | 5 M |
| | b) | Apply the concept of SVM to solve linearly separable problem. | L3 | CO2 | 5 M |
| **OR** | | | | | |
| 9 | a) | What is Backpropagation in artificial neural networks? Analyze Backpropagation Algorithm to update weights and bias. | L4 | CO4 | 5 M |
| | b) | Explain about Multilayer perceptron and its layers. | L3 | CO2 | 5 M |
| **UNIT-V** | | | | | |
| 10 | | Use K Means clustering to cluster the following data into three groups. Assume cluster centroids are A1(2,10), A4(5,8), A7(1,2). The distance function used is Euclidean distance. A1(2,10), A2(2,5), A3(8,4), A4(5,8), A5(7,5), A6(6,4), A7(1,2), A8(4,9). | L4 | C04 | 10 M |
| **OR** | | | | | |
| 11 | a) | Explain the Agglomerative Hierarchical Clustering approach. How does it differ from Divisive Hierarchical Clustering? | L3 | CO3 | 5 M |
| | b) | Write the metrics for evaluating clustering performance and explain. | L3 | CO3 | 5 M |

PVP23

Prasad V Potluri Siddhartha Institute of Technology
Department of CSE (AI & ML)
II B. Tech II Sem
Machine Learning-23AM3401
Short Key

## PART -A

| | |
|---|---|
| 1a) The role of Feature selection in machine learning | 2M |
| b) Define Accuracy | 2M |
| c) What is sigmoid function | 2M |
| d) applications of Regression. | 2M |
| e) Define conditional probability. | 2M |
| f) Define Entropy. | 2M |
| g) Define Perceptron. | 2M |
| h) applications of SVM. | 2M |
| i) Define complete linkage | 2M |
| j) measures of similarity | 2M |

## PART- B

| | |
|---|---|
| 2a) algorithms used in supervised and unsupervised learning with their applications. | 5M |
| b) major Challenges in machine learning | 5M |
| 3a) Five applications of Machine learning | 5M |
| b) Steps in End - to - End ML project | 5M |
| 4) Problem - linear regression | 10M |
| 5a) Apply Logistic regression and Linear regression in classification tasks | 5M |
| b) Algorithm of Nonlinear regression | 5M |
| 6) Problem – Decision tree | 10M |
| 7a) Impact of metrics on the performance of KNN | 5M |
| b) step-by-step process of constructing a Decision Tree using the ID3. | 5M |
| 8a) Compare and contrast biological neuron and artificial neurons | 5M |
| b) Apply the concept of SVM to solve linearly separable problem | 5M |
| 9a) What is Backpropagation in artificial neural networks? Analyse Backpropagation Algorithm to update weights and bias. | 5M |
| b) Apply the concept of a Multilayer Perceptron (MLP) to explain how data flows through its different layer | 5M |
| 10) Problem- K-Means | 10M |
| 11a) agglomerative hierarchical clustering | 5M |
| b) Evaluate the performance of metrics in clustering | 5M |

PVP23

Prasad V Potluri Siddhartha Institute of Technology
Department of CSE (AI & ML)
II B. Tech II sem
Machine Learning
Scheme of Evaluation

**1a) role of feature selection in Machine Learning.** **2M**
It is the process of selecting the most relevant features for a machine learning model.

**b) Accuracy** **2M**
the proportion of correctly classified instances out of the total instances.

**C) Sigmoid function** **2M**
converts any real number into a probability value between 0 and 1.

**d) applications of Regression (Any 2 aplications)** **2M**

house prices, sales forecasting, stock market trends, and weather prediction.

**e) Conditional probability** **2M**
Conditional probability is the probability of an event occurring given that another event has already occurred.

**f) Entropy** **2M**
is the measure of randomness or impurity contained in a dataset.

**g) Perceptron** **2M**
It's a type of linear binary classifier that makes decisions by weighing input signals and applying an activation function.

**h) Applications of SVM** **2M**
Any 2 applications
image classification, text categorization, handwriting recognition, and bioinformatics (e.g., gene classification).

**i)Complete Linkage** **2M**
how distances between clusters are computed during the merging process.

**j) measure of similarity** **2M**
How similar two data points are.

## PART – B

**2 a) Discuss various algorithms used in supervised and unsupervised learning with their applications. Write any 4 (Supervised 3 and unsupervised 1) or (2+ 2)** **5M**

**Supervised Learning Algorithms**

In supervised learning, data is labelled (input + output).

Algorithms & Applications:

Linear Regression

→ Predicting house prices, sales forecasting

Logistic Regression

→ Disease prediction, spam email detection

K-Nearest Neighbours (KNN)

→ Recommendation systems, pattern recognition

Decision Tree

→ Credit risk analysis, medical diagnosis

Naive Bayes

→ Spam filtering, text classification

Support Vector Machine (SVM)

→ Image classification, face detection

Artificial Neural Networks (ANN)

→ Speech recognition, image recognition

**Unsupervised Learning Algorithms**

In unsupervised learning, data is unlabeled.

Algorithms & Applications:

K-Means Clustering

→ Customer segmentation, grouping similar users

Hierarchical Clustering

→ Biological data analysis, document grouping

**2 b) What are the major challenges in training machine learning models? Explain with examples. (write any 3 give 5 marks)** **5M**

**Data Quality and Quantity:**

o Insufficient Data: Many real-world problems lack the massive datasets required to train effective machine learning models.

o Data Quality Issues: Inaccurate, incomplete, or biased data can lead to unreliable models and erroneous predictions.

o Data Privacy and Security: Collecting and storing sensitive data raises concerns about privacy violations and security breaches.

**2. Model Interpretability and Explainability:**

o "Black Box" Problem: Many complex models, especially deep learning models, are difficult to understand. This lack of transparency can hinder trust and make it hard to debug or identify biases.

**3. Overfitting and Underfitting:**

o Overfitting: Models that perform well on training data but poorly on new, unseen data.

o Underfitting: Models that fail to capture the underlying patterns in the data, resulting in poor performance on both training and new data.

**4. Computational Resources:**

o High Computational Costs: Training and deploying complex models can require significant computational power and time, making them expensive and resource-intensive.

**5. Bias and Fairness:**

o Algorithmic Bias: If training data reflects existing societal biases, the model may perpetuate or amplify those biases, leading to unfair or discriminatory outcomes.

**6. Evolving Environments:**

o Concept Drift: The underlying relationships between features and targets in the data can change over time, requiring models to be constantly updated and retrained.

**7. Ethical Considerations:**

o Autonomous Weapons: The development of autonomous weapons systems raises serious ethical concerns about the potential for misuse.

**3 a) Describe five real-world applications of machine learning in different domains. (any 3 give 5 marks)** **5M**

**1. Healthcare**

Disease diagnosis (e.g., cancer detection using imaging data).
Personalized medicine and drug discovery.
Medical imaging analysis (e.g., X-rays, MRIs, ultrasound reports).

**2. Finance**

Fraud detection and prevention.
Credit scoring and risk assessment.
Algorithmic trading and stock market prediction.
Customer segmentation and personalized banking.

**3. Retail and E-commerce**

Product recommendations (e.g., Amazon, Netflix).
Inventory management and demand forecasting.
Customer sentiment analysis and feedback mining.
Price optimization and dynamic pricing.

**4. Transportation**

Autonomous vehicles and self-driving cars.
Traffic prediction and route optimization (e.g., Google Maps, Waze).
Predictive maintenance of vehicles.
Logistics and supply chain optimization.

**5. Education**

Personalized learning and adaptive tutoring systems.
Automated grading and evaluation.
Dropout prediction and intervention.
Content recommendation for e-learning platforms.
Language translation and virtual classrooms.

**6. Entertainment**

- Recommendation systems for movies, music, and books (e.g., Spotify, Netflix).
- Game AI and dynamic content generation.
- Video and image enhancement.
- Audience behavior prediction and targeted advertisements.

**7. Manufacturing and Industry**

- Predictive maintenance of machinery.
- Quality control using image recognition.
- Supply chain optimization.
- Process automation and robotics.
- Defect detection in products.

**8. Agriculture**

- Crop monitoring and yield prediction.
- Weed and pest detection.
- Precision farming with drones and sensors.
- Soil health analysis.
- Weather forecasting and irrigation management.

**9. Energy**

- Energy demand forecasting and grid optimization.
- Predictive maintenance for energy equipment.
- Renewable energy optimization (e.g., wind and solar).

**10. Security**

- Cybersecurity and threat detection.
- Facial recognition and biometric authentication.
- Anomaly detection in network traffic.
- Spam and phishing email detection.
- Surveillance and video analysis.

**3 b) Explain the key steps involved in an end-to-end machine learning project. 5M**

End-to-End Machine Learning Project:

1. Frame the Problem

Objective: Clearly define the problem you are solving, determine the problem type (classification, regression, etc.), and establish clear goals.

- Example: If you are tasked with predicting house prices in a specific region, identify that the problem is a regression problem, as the target variable (house price) is continuous.

Steps:

1. Understand the Business Context: Determine the underlying business need. For example, a real estate company may need price predictions to assist with pricing strategies.
2. Define the Objective: Set a clear objective, such as predicting house prices based on various features like the number of bedrooms, location, and square footage.

3. Specify the Success Criteria: Identify appropriate evaluation metrics, such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), to assess model performance.

2. Get the Data
Objective: Acquire the necessary data to build and train your model.
• Example: Obtain a dataset containing house prices along with relevant features, which could be sourced from a public dataset like the Boston Housing dataset or a proprietary real estate database.

Steps:
1. Identify Data Sources: Determine where the data is stored (e.g., databases, CSV files, APIs).
2. Access the Data: Download or query the data and load it into a working environment, such as a pandas DataFrame in Python.
3. Ensure Data Sufficiency: Verify that the data includes all required features and target variables needed to solve the problem.

3. Explore and Visualize the Data to Gain Insights
Objective: Perform exploratory data analysis (EDA) to understand the data, detect patterns, and identify any anomalies or missing values.

• Example: Visualize how house prices vary with different features like the number of bedrooms, or examine the geographical distribution of house prices.
Steps:
1. Conduct Descriptive Statistics: Use statistical summaries to understand the distribution of data (e.g., using .describe() in pandas).
2. Data Visualization: o Histograms: Visualize the distribution of individual variables, such as house prices.
o Scatter Plots: Analyze relationships between features (e.g., house size vs. price).
o Box Plots: Identify outliers in the data, such as unusually high or low prices.

3. Correlation Analysis: Create a correlation matrix to examine relationships between features and the target variable.
4. Identify Missing Data: Detect missing values using methods like .isnull().sum() in pandas, and assess the extent of the missing data.

4. Prepare the Data for Machine Learning Algorithms
Objective: Pre-process the data to ensure it is clean, well-structured, and in a format suitable for machine learning algorithms.
• Example: Address missing values, encode categorical variables, and scale features to prepare the data for modelling.
Steps:
1. Handle Missing Values: o Imputation: Replace missing values with appropriate substitutes, such as the mean or median for numerical features, or the most frequent category for categorical features.

2. Feature Scaling: o Standardization: Adjust features to have a mean of zero and a standard deviation of one.
o Normalization: Scale features to a specific range, such as 0 to 1.
3. Encode Categorical Variables: o One-Hot Encoding: Convert categorical variables into binary vectors, creating separate columns for each category.

o Label Encoding: Transform categorical labels into numerical values.
4. Feature Engineering: Create new features or modify existing ones to enhance model performance, such as creating interaction terms or polynomial features.
5. Data Splitting: Divide the dataset into training and testing subsets (commonly an 80/20 split) to allow for model evaluation on unseen data.
5. Select a Model and Train It
Objective: Choose a machine learning model appropriate for the problem at hand and train it on the prepared data.
• Example: Begin with a simple Linear Regression model as a baseline, and consider more complex models like Random Forests if needed.
Steps:
1. Model Selection: o Start with simpler models, such as Linear Regression, to establish a baseline for comparison.
o Experiment with more complex models, such as Decision Trees, Random Forests, or Gradient Boosting Machines, to potentially improve performance.
2. Model Training: o Fit the selected model to the training data using methods like .fit() in scikit-learn.
o Assess the model's initial performance on the training set to identify potential issues such as underfitting or overfitting.

6. Evaluation
Objective: Evaluate the model's performance using appropriate metrics to determine its effectiveness.
• Example: After training the model, evaluate its accuracy by calculating metrics like RMSE on the test dataset.
Steps:
1. Make Predictions: Use the trained model to generate predictions on the test set.
2. Calculate Performance Metrics: o For Regression: Use metrics such as RMSE, MAE, and $R^2$ to evaluate model performance.
o For Classification: Use metrics like accuracy, precision, recall, and F1-score (if applicable).
3. Cross-Validation: Apply k-fold cross-validation to ensure that the model generalizes well across different subsets of the data.

7. Fine-Tune the Model
Objective: Optimize the model's performance by adjusting hyperparameters and exploring advanced techniques like ensemble learning.
• Example: Use techniques like Grid Search or Randomized Search to optimize hyperparameters for models like Random Forests or Gradient Boosting Machines.
Steps:
1. Hyperparameter Tuning: o Grid Search: Exhaustively search over a specified parameter grid to identify the best combination of hyperparameters.
o Randomized Search: Perform a less exhaustive, randomized search to find good hyperparameter values more quickly.
2. Ensemble Methods: o Bagging: Combine multiple models to reduce variance, such as using Random Forests.
o Boosting: Sequentially train models that correct the errors of previous models, as seen in algorithms like Gradient Boosting or XGBoost.

3. Post-Tuning Evaluation: Re-evaluate the optimized model on the test set to ensure improved performance.

8. Deployment and Maintenance
Objective: Deploy the model into a production environment and implement a plan for ongoing monitoring and maintenance.
• Example: Deploy the house price prediction model as a web service accessible through an API.
Steps:
1. Model Deployment: o Containerization: Use tools like Docker to package the model along with its dependencies for easy deployment.
o API Deployment: Create a RESTful API using frameworks like Flask or FastAPI to allow external applications to access the model.
o Cloud Deployment: Deploy the model on cloud platforms such as AWS, Azure, or Google Cloud for scalability and reliability.
2. Monitoring and Performance Tracking: o Implement monitoring tools to track the model's performance over time, watching for signs of degradation or data drift.
3. Model Maintenance: o Regular Updates: Periodically retrain the model with new data to ensure it remains accurate and relevant.
o A/B Testing: Implement A/B testing to compare the performance of updated models against existing ones, ensuring any changes lead to performance improvements.

4) **Working Process of Linear Regression – 5M**
**Problem -5M (Formula -1 M, Slope -1M, Intercept- 1M, Prediction-2M)**
Linear Regression is a supervised learning algorithm used to model the relationship between an independent variable (house size) and a dependent variable (price) by fitting a straight line.
Steps involved:
Collect Data:
Given house sizes and corresponding prices.
Assume Linear Model:
The relationship is expressed as:
$Y=\beta0+\beta1X$
where
y = predicted price
x = house size
$\beta0$ = intercept
$\beta1$ = slope
Compute Parameters ($\beta0$ and $\beta1$):
Using formulas:
Slope $\beta1 = \frac{\Sigma((x - \bar{x})(y - \bar{y}))}{(\Sigma(x - \bar{x})^2)}$
Intercept = $\beta0 = (\Sigma y / n) - \beta1 * (\Sigma x / n) = \bar{y} - \beta1\ \bar{x}$
Fit the Model:
Substitute values of $\beta0$ and $\beta1$ into the equation.
Prediction:
Use the model to estimate price for a new house size.
Slope=0.107 Intercept=45.5 y=227.4
(Complete problem has been attached on a separate page) — See the last page (18)

**5a) Apply Logistic Regression and Linear Regression for classification tasks. 5M**

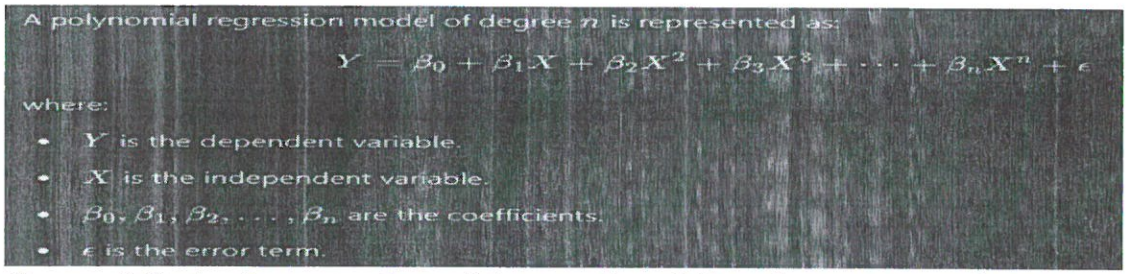
Logistic regression -4M
Linear regression -1M
Logistic Regression (for Classification)
Used when the output is **categorical** (e.g., Yes/No, 0/1).
It predicts **probability** using a sigmoid function.
Decision boundary:
If probability $\geq 0.5 \rightarrow$ Class 1
Else $\rightarrow$ Class 0
Example: Email spam detection, disease prediction.
Linear regression is not suitable for classification tasks.

**5b) Interpret the algorithm used for Non-Linear Regression and how it works. 5M**

**Polynomial regression Definition - 2M**

**Working procedure – 3M**

Polynomial regression is a form of regression analysis in which the relationship between the independent variable X and the dependent variable y is modeled as an nth-degree polynomial. It is a flexible approach that extends linear regression to capture nonlinear relationships in data.
While linear regression models the relationship as a straight line, polynomial regression allows for curvature by including higher-order terms of X.
A higher degree allows the model to fit more complex, nonlinear relationships.




Successfully implement polynomial regression, follow these structured steps:
Step 1: Data Collection & Pre-processing
Gather data that includes both independent variable(s) (features) and dependent variable (target).
Check for **missing values**, **outliers**, and **inconsistencies** in the dataset.
Normalize or standardize data if necessary, to improve performance.
Step 2: Exploratory Data Analysis (EDA)
**Visualize Data:** Create scatter plots to examine the relationship between independent and dependent variables.
**Correlation Analysis:** Check if there is a non-linear relationship that might require a polynomial model instead of linear regression.
**Feature Selection:** Choose the most relevant variables that contribute to the dependent variable.
Step 3: Transform the Features (Generate Polynomial Features)
Convert the original independent variable XXX into polynomial terms such as:
X (original feature),$X^2$ (quadratic feature),$X^3$ (cubic feature), and so on.
Choose the appropriate **polynomial degree** based on data trends.

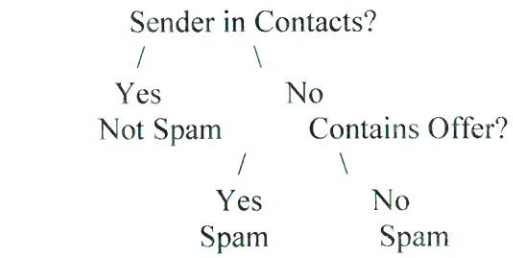
Step 4: Split Data for Training and Testing
Divide the dataset into a **training set** (for model building) and a **testing set** (for model evaluation).
The training set is used to **train the polynomial regression model**, while the test set helps in **evaluating generalization**.
Step 5: Train the Polynomial Regression Model
Fit the polynomial regression model using **least squares estimation** or Gradient Descent optimization algorithm (minimizing the sum of squared residuals).
Solve for the coefficients (β0,β1,β2,...)that best fit the polynomial equation.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_n X^n + \epsilon$$

Step 6: Evaluate Model Performance
To assess the model's accuracy, calculate the following metrics:
**Mean Squared Error (MSE)** – Measures the average squared differences between actual and predicted values.
**Root Mean Squared Error (RMSE)** – Helps understand prediction errors in the original scale.
**R-squared ($R^2$)** – Indicates how well the model explains the variance in the data (closer to 1 is better).
Compare these metrics for different polynomial degrees to determine the best fit.
Step 7: Model Tuning and Optimization
**Choose the Optimal Polynomial Degree:**
A **low-degree polynomial** might underfit the data (high bias, low variance).
A **high-degree polynomial** might overfit (low bias, high variance).
Use **cross-validation** to test different degrees and select the best one.
Step 8: Make Predictions
Once trained and optimized, use the model to predict outcomes for new data points.
Interpret the results to derive meaningful insights.
Step 9: Deployment & Interpretation
Deploy the model into production for real-time predictions.
Periodically update the model with new data for better accuracy.
Visualize the polynomial regression curve to explain insights effectively

**6) Problem – Decision Tree**
**i) Build decision tree – 5M**
**ii)Prediction - 3 M**
**iii)Information Gain - 2M**
Step 1: Dataset Summary
Total samples = 5
Spam = 3, Not Spam = 2
Entropy of Dataset
H(S)=−3/5 log 3/5-2/5 log 2/5=0.971
Step 2: Information Gain for "Contains Offer"
Split data:
Offer = Yes (3 records):
Spam = 2, Not Spam = 1

H(Yes)≈0.918
Offer = No (2 records):
Spam = 1, Not Spam = 1
H(No)=1
Weighted Entropy
Hsplit=3/5(0.918)+2/5(1)=0.951
Information Gain
IG=0.971−0.951=0.020
Information Gain (Contains Offer) ≈ 0.02
Step 3: Build Decision Tree (best splits)
Checking other attributes (briefly):
Sender in Contacts gives perfect separation:
Yes → Not Spam
No → Mostly Spam
So choose Sender in Contacts as root.
Final Decision Tree




Prediction
Given:
Offer = No
Sender in Contacts = No
Email Length = Short
Path:
Sender in Contacts = No → go right
Predict Spam
Final Prediction: YES (Spam)

**7 a) Use different distance metrics in KNN, such as Euclidean, Manhattan, and Minkowski distances. How does the choice of distance metric impact classification performance? 5M**

| **Comparative Analysis: Metric** | **Robust to Outliers** | **High-Dim Performance** | **Feature Scaling Needed?** | **Data Type Suitability** |
|---|---|---|---|---|
| Euclidean | No | Poor | Yes | Continuous, low-dim |
| Manhattan | Yes | Moderate | Yes | High-dim, sparse |
| Minkowski | Depends on p | Moderate | Yes | Flexible (depends on P) |

**7 b) Illustrate the step-by-step process of constructing a Decision Tree using the ID3. 5M**

**Steps in ID3 algorithm:**

1. It **begins with the original set S as the root node.**
2. On **each iteration of the algorithm**, it iterates through the very unused attribute of the set S and **calculates Entropy(H) and Information gain(IG)** of this attribute.
3. It then selects the attribute which has the **smallest Entropy or Largest Information gain.**
4. The **set S is then split by the selected attribute** to produce a subset of the data.

**8 a) Compare and contrast biological neuron and artificial neural networks. 5M**
**(write any 3 give 5 marks)**

| Aspect | Biological Neuron | Artificial Neuron |
|---|---|---|
| **Inspiration** | Natural cell in the human brain | Computational model inspired by biological neurons |
| **Structure** | Dendrites, Soma (cell body), Axon, Synapse | Inputs, Weights, Bias, Activation Function |
| **Signal Transmission** | Electrochemical impulses | Numerical values (real numbers) |
| **Processing** | Complex biochemical processing | Weighted sum of inputs followed by activation |
| **Learning Mechanism** | Synaptic plasticity (e.g., Hebbian learning) | Optimization algorithms (e.g., gradient descent) |
| **Connections** | Highly interconnected ($10^4$ synapses/neuron approx.) | Limited connections, typically layered architecture |

**8 b) Apply the concept of SVM to solve linearly separable problem 5M**

**Concept of SVM- 3M**

**Linearly separable explanation – 2 M**

Let's consider the case where the data is linearly separable, meaning a straight line (or hyperplane) can perfectly divide the different classes.

The linear SVM classification algorithm aims to find this optimal separating hyperplane.

Given a set of training data points (Xi, yi)

Xi be the input vector

yi $\in$ {-1,1} be the class label

the goal is to find a hyperplane defined by:

$w \cdot x + b = 0$

where:

w is the weight vector, which is normal to the hyperplane.

x is the input feature vector.

b is the bias term.

The algorithm seeks to find w and b that satisfy the following conditions for all training examples:

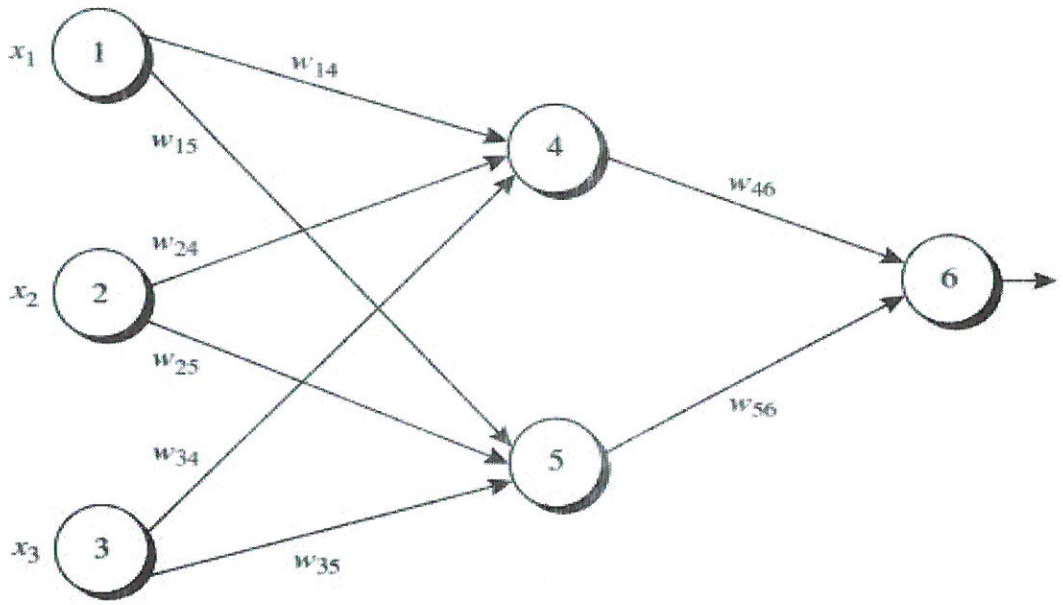
$w \cdot xi + b \geq +1 \; for \; yi = +1$

$w \cdot xi + b \leq -1 \; for \; yi = 1$
These two inequalities can be combined as:
$yi(w \cdot xi + b) \geq 1 \; \forall i$

**9a) What is Backpropagation in artificial neural networks? Analyse Backpropagation Algorithm to update weights and bias.**

**Backpropagation Explanation – 3M**
**Algorithm -2 M**

The backpropagation algorithm is a powerful and essential tool for training artificial neural networks.
It efficiently computes the gradients of the loss function with respect to the weights and biases of the network, which are then used to update these parameters in order to minimize the loss.
By efficiently calculating and propagating error gradients, it enables networks to learn complex patterns and solve a wide range of machine learning problems.
The fundamental principle behind backpropagation is the **chain rule of calculus**. It allows us to calculate the gradient of a complex function (the loss function with respect to network parameters) by breaking it down into a chain of simpler derivatives. The error (the difference between the network's output and the desired output) is propagated backward through the network, layer by layer, to determine the contribution of each weight and bias to this error.




**Algorithm: Backpropagation.** Neural network learning for classification or numeric prediction, using the backpropagation algorithm.
**Input:**
- *D*, a data set consisting of the training tuples and their associated target values;
- *l*, the learning rate;
- *network*, a multilayer feed-forward network.

**Output:** A trained neural network.
**Method:**

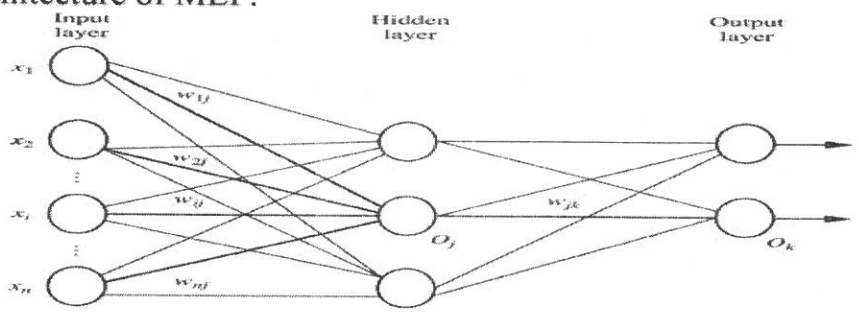
(1) Initialize all weights and biases in *network*;
(2) **while** terminating condition is not satisfied
(3) **for** each training tuple $X$ in $D$
(4) // Propagate the inputs forward:
(5) **for** each input layer unit $j$ {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) **for** each hidden or output layer unit $j$
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit $j$ with respect to the previous layer, $i$
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit $j$
(10) // Backpropagate the errors:
(11) **for** each unit $j$ in the output layer
(12) $Err_j = O_j(1 - O_j)(T_j - O_j)$; // compute the error
(13) **for** each unit $j$ in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, $k$
(15) **for** each weight $w_{ij}$ in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) **for** each bias $\theta_j$ in *network* {
(19) $\Delta \theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
(21) } }

**9 b) Apply the concept of a Multilayer Perceptron (MLP) to explain how data flows through its different layers.** **5M**

Concept of MLP – 3M
Explanation -2M

A Multi-Layer Perceptron (MLP) is a type of feedforward artificial neural network that consists of multiple layers of interconnected nodes (neurons).
It consists of multiple layers of neurons, where each neuron in one layer is connected to every neuron in the next layer — hence the term fully connected.
Feedforward Structure: Data flows in one direction (input → hidden layers → output).

Architecture of MLP:



**Input Layer:**
Takes input features.
No computation, only passes data to the next layer.
**Hidden Layer(s):**
Located between the input and output layers.
There can be one or multiple hidden layers, making the network "deep."
Each hidden layer neuron receives weighted inputs from all neurons in the previous layer.
These neurons perform a non-linear transformation on the weighted sum of their inputs using an **activation function**.
Hidden layers are responsible for extracting complex features and patterns from the input data.
The number of hidden layers and the number of neurons in each hidden layer are **hyperparameters** that need to be carefully chosen.
**Output Layer:**
Produces the final output of the network.
The number of neurons in the output layer depends on the task:
**Binary Classification:** Typically one neuron (with a sigmoid activation) representing the probability of belonging to one class, or two neurons (with softmax) representing the probabilities of each class.

10) Iteration 1: Assign points to nearest centroid - 3M
Compute nearest centroid for each point:

| Point | C1 (2,10) | C2 (5,8) | C3 (1,2) | Cluster |
|---|---|---|---|---|
| A1(2,10) | 0 | 3.61 | 8.06 | C1 |
| A2(2,5) | 5 | 4.24 | 3.16 | C3 |
| A3(8,4) | 8.48 | 5 | 7.28 | C2 |
| A4(5,8) | 3.60 | 0 | 7.21 | C2 |
| A5(7,5) | 7.07 | 3.60 | 6.70 | C2 |
| A6(6,4) | 7.21 | 2.44 | 5.38 | C2 |
| A7(1,2) | 8.06 | 7.21 | 0 | C3 |
| A8(4,9) | 2.23 | 1.41 | 7.61 | C2 |

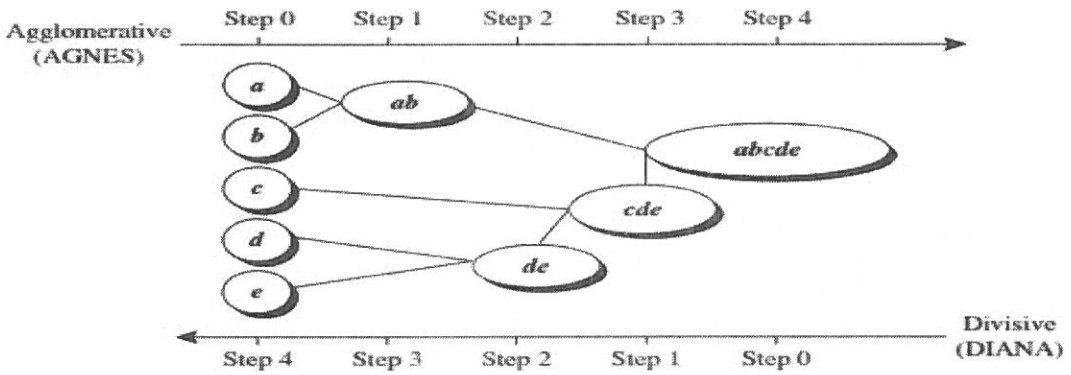
Clusters after Iteration 1
**C1:** {A1}
**C2:** {A3, A4, A5, A6, A8}
**C3:** {A2, A7}
Update Centroids -1M
New C1
(2,10)
New C2
((8+5+7+6+4)/5), (4+8+5+4+9)/5)) = (6,6)
New C3
((2+1)/2),((5+2)/2)=(1.5,3.5)
Iteration 2: Reassign Points -3M

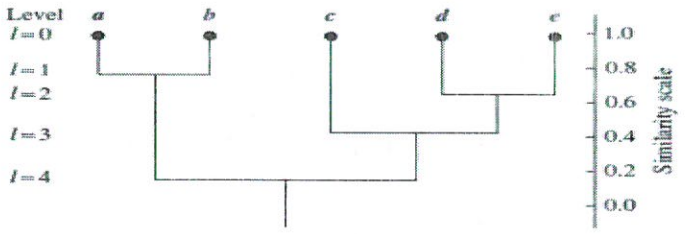
| Point | C1 (2,10) | C2 (6,6) | C3 (1.5,3.5) | Cluster |
|---|---|---|---|---|
| A1 | 0 | 5.65 | 6.51 | C1 |
| A2 | 5 | 4.12 | 1.58 | C3 |
| A3 | 8.48 | 2.82 | 6.51 | C2 |
| A4 | 3.60 | 2.23 | 5.70 | C2 |
| A5 | 7.07 | 1.41 | 5.50 | C2 |
| A6 | 6.32 | 2 | 4.5 | C2 |
| A7 | 8.06 | 6.40 | 1.58 | C3 |
| A8 | 2.23 | 3.60 | 6.04 | C1 |

Clusters after Iteration 2
**C1:** {A1, A8}
**C2:** {A3, A4, A5, A6}
**C3:** {A2, A7}
Update Centroids Again -1M
C1= (3,9.5)
C2= (6.5,5.25)
C3= (1.5,3.5)
Reassigning : no change in clusters -2M

11a) Agglomerative -3M
Differentiate with Divisive -2M

Agglomerative (Bottom-Up) Clustering:
It starts with each data point as a single cluster and then iteratively merges the closest pairs of clusters until all data points belong to a single cluster or a predefined stopping criterion is met.
Divisive (Top-Down) Clustering:
This approach starts with all data points in a single cluster and then iteratively splits clusters into smaller sub-clusters until each data point forms its own cluster or a predefined stopping criterion is met.






Dendrogram representation for hierarchical clustering of data objects $\{a, b, c, d, e\}$.

Agglomerative Hierarchical Clustering:

It is a **bottom-up** clustering method that builds a hierarchy (tree) of clusters by **iteratively merging** the two closest clusters until all points are agglomerated into one.

**Algorithm**: Agglomerative Hierarchical Clustering

Input:

D: a data set containing n objects.

Output:

Dendrogram - A Visual History of Merges

Method

Compute the proximity matrix

Let each data point be a cluster

Repeat

Merge the two closest clusters

Update the proximity matrix

**Until** only a single cluster remains

11b) **Metrics for Evaluating Clustering Performance** **5M**

Evaluating clustering performance is crucial to determine how well a clustering algorithm has grouped the data.

Unlike supervised learning (where we have ground truth labels), clustering evaluation is more challenging because it is **unsupervised**.

Clustering metrics can be categorized into:

Internal Evaluation Metrics (No ground truth)

External Evaluation Metrics (Ground truth available)

**Internal Evaluation Metrics:** Used when true labels are unknown. These measure cluster compactness (intra-cluster similarity) and separation (inter- cluster dissimilarity).

**Silhouette Score**: Measures how similar a point is to its own cluster compared to other clusters.

For a single data point i:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

a(i): Average distance from i to all other points in its cluster (cohesion). b(i): Smallest average distance from i to points in any other cluster

(separation).
Overall Silhouette Score:

$$\text{Silhouette Score} = \frac{1}{N}\sum_{i=1}^{N} s(i)$$

Interpretation:
**Score ∈ [-1, 1]**
o **+1**: Points are well-clustered (tight and far from other clusters).
**0**: Clusters overlap.
**-1**: Points are wrongly clustered.
**Davies-Bouldin Index (DBI)**: Measures the average similarity between each cluster and its most similar cluster (lower = better).

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

- $\sigma_i$ = Average distance of all points in cluster $i$ to its centroid (**cluster spread**).
- $d(c_i, c_j)$ = Distance between centroids of clusters $i$ and $j$.

Interpretation:
Lower DBI = Better clustering (clusters are compact and well- separated).
**External Evaluation Metrics:** Used when true labels are known (compares clustering results to ground truth).
**Adjusted Rand Index (ARI):** Measures similarity between true and predicted clusters, adjusted for chance.

$$ARI = \frac{RI - \text{Expected RI}}{\max(RI) - \text{Expected RI}}$$

**RI (Rand Index) =**
**TP: Pairs correctly clustered together.**
**TN: Pairs correctly kept apart.**

4. Apply linear regression to predict the price of a 1700 sq.ft house for the following data

Ans:- $\bar{x} = \frac{7500}{5} = 1500$ $\quad \bar{y} = \frac{1030}{5} = 206$

$$\text{Slope } \beta = \frac{\Sigma (x-\bar{x})(y-\bar{y})}{\Sigma (x-\bar{x})^2}$$

$$= \frac{(1000-1500)(150-206) + (1200-1500)(180-206) + (1500-1500)(200-206) + (1800-1500)(240-206) + (2000-1500)(260-206)}{(1000-1500)^2 + (1200-1500)^2 + (1800-1500)^2 + (2000-1500)^2}$$

$$= \frac{(-500)(-56) + (-300)(-26) + (300)(34) + (500)(54)}{250000 + 90000 + 90000 + 250000}$$

$$= \frac{28000 + 7800 + 10200 + 27000}{680000} = \frac{73000}{680000} = 0.107$$

$$\text{Intercept } \beta_0 = \bar{y} - b\bar{x} = 206 - 0.107(1500)$$

$$= 206 - 160.5 = 45.5$$

$$y = \beta_0 + \beta_1 x$$

$$= 45.5 + 0.107x$$

predict the price of 1700 sq ft house

$$y = 45.5 + 0.107(1700)$$

$$= 45.5 + 181.9 = 227.4$$