PVP 23

Code: 23DS3501

III B.Tech - I Semester - Regular Examinations - NOVEMBER 2025

MACHINE LEARNING

(CSE - DS)

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)	Differentiate between Supervised and Unsupervised Learning.	L2	CO1
1.b)	Define deployment in the context of ML projects.	L2	CO1
1.c)	What is the role of gradient descent in regression?	L2	CO2
1.d)	What is binary classification in logistic regression?	L2	CO4
1.e)	Interpret attribute selection measure in decision trees.	L2	CO2
1.f)	List any two distance metrics used in KNN.	L2	CO4
1.g)	What is meant by a margin in SVM?	L2	CO1
1.h)	Summarize the role of activation functions in neural networks.	L2	CO2
1.i)	State two differences between similarity and dissimilarity measures.	L2	CO3
1.j)	Define divisive hierarchical clustering.	L2	CO3

PART – B

			BL	CO	Max. Marks
UNIT-I					
2	a)	Define Machine Learning and explain its need in modern systems.	L2	CO1	4 M
	b)	Survey various types of Machine Learning with examples.	L4	CO1	6 M
OR					
3	a)	Interpret the phases of CRISP-DM methodology.	L3	CO1	6 M
	b)	Compare CRISP-DM with the End-to-End ML workflow.	L4	CO1	4 M
UNIT-II					
4	a)	Derive the mathematical equation for simple linear regression.	L3	CO2	5 M
	b)	Explain the process of model fitting in simple regression with an example.	L2	CO2	5 M
OR					
5	a)	Define polynomial regression with an example.	L2	CO4	5 M
	b)	Examine any two applications of polynomial regression.	L3	CO4	5 M
UNIT-III					
6	a)	Explain the concept of decision tree induction.	L2	CO1	5 M
	b)	Describe the representation of a decision tree with a simple example.	L2	CO2	5 M
OR					
7	a)	Explain any two applications of Naive Bayes.	L4	CO2	5 M
	b)	Analyze the impact of different distance metrics on the performance of KNN algorithm, provide examples.	L4	CO4	5 M
UNIT-IV					
8	a)	Explain the relationship between biological neurons and artificial neurons.	L2	CO1	5 M
	b)	Draw and explain the structure of an artificial neuron model.	L3	CO2	5 M
OR					
9	a)	Explain the concept of a feedforward neural network with a diagram.	L2	CO2	5 M
	b)	Illustrate the concept of margin and support vectors in SVM. How do the contribute to the classification process?	L3	CO1	5 M
UNIT-V					
10	a)	Explain the K-Means algorithm with suitable pseudocode.	L2	CO3	5 M
	b)	Compare and Contrast K-Means and K-Medoids.	L4	CO4	5 M
OR					
11	a)	Explain the working of agglomerative hierarchical clustering.	L2	CO3	5 M
	b)	Demonstrate linkage criteria (single, complete, average) with examples.	L3	CO4	5 M

E1

Prasad V Potluri Siddhartha Institute Of Technology
Department of CSE(Data Science)
III B.Tech I Sem
Machine Learning
Short Key

PART -A

1a) **Difference between supervised and unsupervised learning**	2M
b) **Define Model Deployment**	2M
c) **The role of gradient descent in regression**	2M
d) **Binary classification in logistic regression**	2M
e) **Attribute selection measure in decision trees**	2M
f) **Any two distance metrics in KNN**	2M
g) **Margin in SVM**	2M
h) **Role of activation functions in Neural Networks**	2M
i) **Two differences between similarity and dissimilarity measures**	2M
j) **Definition of divisive hierarchical clustering**	2M

PART- B

2a) Definition of Machine Learning and its need in modern system	4M
b) Various types of machine learning with examples	6M
3a) Phases of CRISP-DM:	6M
b) Any 3 Comparisons between CRISP-DM and End -to – End ML workflow	4M
4a) Derivation of equation for simple linear regression	5M
4 b) Process of model fitting in simple regression with an example	5M
5a) Definition of Polynomial Regression	5M
b) Any Two applications	5M
6a) Concept of Decision tree induction	5M
b) Representation of a decision tree with example	5M
7a) Explanation of any two applications of Naïve Bayes	5M
b) Impact of metrics on the performance of KNN	5M
8a) Relation between biological neuron and artificial neurons	5M
b) Structure of artificial neuron	5M
9a) Explanation of feed forward neural network	5M
b) Concept of Margin and Support Vector Machine	5M
10a) Explanation of K-Means algorithm with pseudocode	5M
b) Compare K-Means and K-Medoids	5M
11a) Working of agglomerative hierarchical clustering	5M
b) Linkage criteria	5M

PVP23

Prasad V Potluri Siddhartha Institute of Technology
Department of CSE (Data Science)
III B.Tech I sem

Machine Learning

Scheme of Evaluation

1a) Any **two differences between supervised and unsupervised learning** **2M**

Supervised ,	**Unsupervised**
1)Learning with labeled data where input-output mapping is known.	1)Learning with unlabeled data to find hidden patterns or structure.
2)Regression, Classification (e.g., Linear Regression, SVM, Decision Trees).	2)Clustering, Dimensionality Reduction (e.g., K-Means, PCA).
Performance is measured using metrics like accuracy, precision, recall.	Performance is evaluated by the interpretability of patterns (e.g., Silhouette Score for clustering).

b) **Model Deployment:** **2M**

- **Save the Model**: Serialize the model using formats like joblib or pickle.
- **API Creation**: Deploy the model using a REST API (e.g.,Flask or FastAPI).

c)**Role of Gradient Decent in regression** **2M**

Gradient Descent is a powerful optimization algorithm for minimizing the cost function in Linear Regression. By iteratively updating the parameters, it finds the best-fitting regression line.

d) **Binary classification in logistic regression** **2M**

To predict one of two possible outcomes (e.g., yes/no, true/false, 0/1).

e) **Attribute selection measure in decision trees** **2M**

It is a heuristic for selecting the splitting criterion that "best" separates a given data partition, D, of a class- labeled training tuples into individual classes.

Or

The methods are used for attribute selection as follows:

1. **Entropy**
2. **Information Gain**
3. **Gain Ratio**
4. **Gini Index**

f) **Two distance metrics in KNN** **2M**

Euclidean Distance, Manhattan Distance, Minkowski Distance, Cosine Similarity

g) **Margin in SVM** **2M**

The margin is the distance between the hyperplane (decision boundary) and the closest data points from each class.

h) **Role of activation functions in neural networks** **2M**

Activation functions decide whether a neuron should be activated or not. Used to introduce non-linearity, enabling the network to learn complex functions.

i)**Two differences between similarity and dissimilarity** **2M**

Similarity vs. Dissimilarity

o Similarity Measures:

Indicate how alike two data points are.

Higher values mean greater similarity.

Used in algorithms like hierarchical clustering and cosine similarity-based clustering.

o Dissimilarity (Distance) Measures:

Indicate how different two data points are.

Lower values mean greater similarity

Used in algorithms like K-Means, DBSCAN, and hierarchical clustering (with distance metrics).

j) **Definition of divisive hierarchical clustering** **2M**

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.

PART – B

2a) Definition of Machine Learning and its need in modern system **4M**

Any definition of ML **2M**

A computer program is said to learn from experience *E* with respect to some class of tasks ***T*** and performance measure ***P***, if its performance at tasks in *T*, as measured by ***P***, improves with experience ***E***

Need -Any 2 **2M**

A conventional program takes some inputs and produces output(s) based on a set of instructions or rules. Conventional programming, also known as rule-based programming, involves explicitly coding rules and instructions for the system to follow.

While it has been successful in many domains, it has several limitations that highlight the need for advanced approaches like Machine Learning (ML).

Rigid and Inflexible

o Conventional programs operate strictly within the predefined rules coded by developers.

o They cannot adapt to new or unexpected situations without manual intervention or reprogramming.

o Explicitly defining rules for every possible scenario can become infeasible as the problem domain grows.

o **Example:** In natural language processing, creating grammar rules to handle every linguistic nuance is nearly impossible.

Scalability Issues

o As the size and complexity of the problem increase, the number of rules grows exponentially, making the system difficult to manage and maintain.

o **Example**: Defining rules for an e-commerce recommendation system would require an impractical number of conditions to account for user preferences and behavior.

Inability to Generalize

o Rule-based systems operate strictly within predefined conditions and fail when encountering new or unseen situations.

o **Example**: A spam email filter based on specific keywords might miss new spam patterns unless those are explicitly programmed.

High Development and Maintenance Cost

o Writing and maintaining a large set of rules requires significant time, effort, and domain expertise.

o Any change in requirements demands updating numerous interdependent rules.

o **Example**: Updating a tax calculation system to comply with new regulations can involve rewriting large portions of the code.

Difficulty with Noisy or Incomplete Data

o Conventional programming cannot handle noisy, ambiguous, or missing data effectively without extensive error-handling code.

o **Example**: In image recognition, defining pixel-level rules for identifying objects is impractical due to variations in lighting, angles, and noise.

2 b) **Various types of machine learning with examples** **6M**

3 Types of learning **3X2=6M**

ML can be divided into different types based on the data and the problem. There are three primary categories for machine learning:

Supervised learning

Unsupervised learning

Reinforcement learning

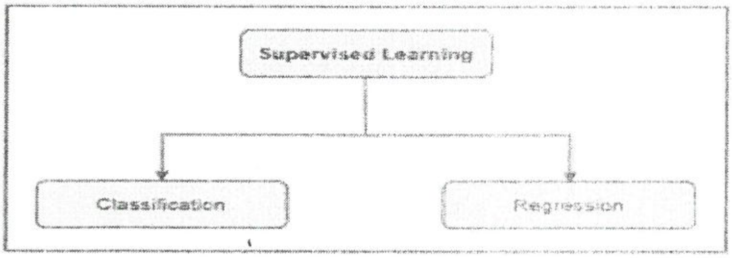
Supervised Learning

Supervised learning involves training the model using a labeled dataset, where the input data is paired with the corresponding desired output (labels). The goal is for the model to learn a mapping function from inputs to outputs and generalize to unseen data.

Types of supervised learning : There are two main types of supervised machine learning: regression and classification.

Classification

Regression



Categories of supervised learning

Classification is a type of supervised learning that aims to predict a categorical (discrete) label for each input.

•Any example

- **Regression** is the type of supervised learning where the goal is to predict a continuous output variable.
    - Any example

Unsupervised learning:-

- Unsupervised learning is a type of machine learning where a model is trained on a dataset without labeled output.
- The primary goal is to uncover hidden patterns, structures, or relationships within the input data.

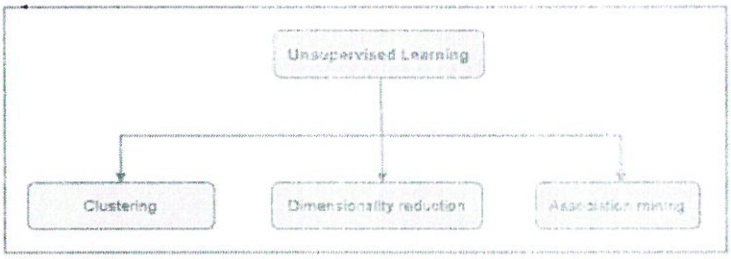
Types of unsupervised learning

Unsupervised learning is classified into three types:

Clustering: Clustering is a machine learning approach in which data points are grouped together. Theoretically, data points belonging to the same group should have similar characteristics, but data points belonging to other groups should have considerably different characteristics.

Dimensionality reduction: The process of reducing the number of features or variables in a dataset while keeping the most crucial information is known as dimensionality reduction, and it's applied in machine learning and data analysis.

Association mining: Another key aspect of unsupervised learning is association mining. This method is used to find interesting relationships or patterns in huge datasets.



Categories of unsupervised learning

Reinforcement Learning (RL):-

- Reinforcement learning involves an agent that learns to make decisions by interacting with an environment.
- The agent receives rewards or penalties based on its actions, and its goal is to maximize the cumulative reward.

Any Example

3a) **Phases of CRISP-DM:** **6M**

six major phases:

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

3 b) **Any 2 Comparisons between CRISP-DM and End -to – End ML workflow** **4M**

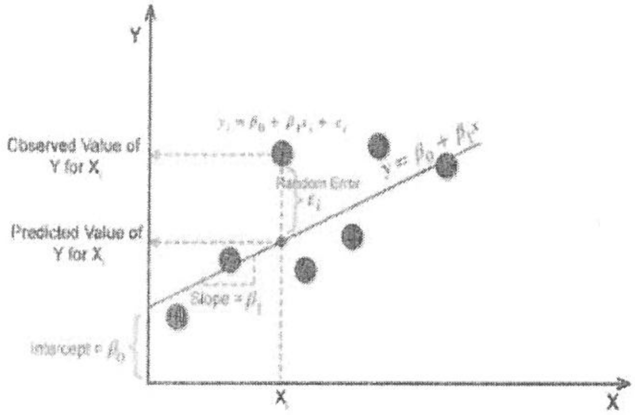
CRISP-DM	End-to-End Machine Learning
A generic framework for Machine Learning projects, with a strong emphasis on the initial stages of business and data understanding.	A more specific lifecycle for machine learning, including deployment, monitoring, and continuous retraining.
Lacks specific phases for continuous deployment and model management after initial deployment, which is critical for ML applications.	Can be more complex to implement due to its integration of software engineering practices like CI/CD.
Highly flexible and adaptable to different industries and project types.	Can be adapted, but is specifically tailored for the needs of machine learning projects.
Treats the final model as the end of the project, rather than the beginning of its operational life.	Views the model as a continuous process that requires ongoing monitoring, evaluation, and retraining in response to new data.
Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment.	Includes the CRISP-DM phases but extends them with activities like continuous integration, continuous deployment, and ongoing monitoring and retraining.

4a) **Derivation of equation for simple linear regression** **5M**

Simple linear regression models the relationship between a single independent variable (x) and a dependent variable (y) using a linear equation

Model Equation:

$y = \beta 0 + \beta 1x + \epsilon$



y is the dependent variable (target)

x is the independent variable (feature)

β0 is the intercept

(the value of y when X is 0)

$\beta 1$ is the slope (coefficient)

ϵ is the error term

- The goal of simple linear regression is to estimate the values of $\beta 0$ and $\beta 1$ that minimize the sum of squared residuals, typically using the least squares method.
- $\beta 0$ represents the value of the dependent variable when the independent variable is zero (the intercept).
- $\beta 1$ indicates the change in the dependent variable for a one-unit change in the independent variable.

4 b) **Process of model fitting in simple regression with an example** **5M**

Process of model fitting **4M**

1.Data Collection:

- Gather data on the dependent and independent variables.

2 **Model Specification:**

- Specify the linear relationship between y and X.

3 **Parameter Estimation:**

- Use the method of least squares to estimate the parameters $\beta 0$ and $\beta 1$. This method minimizes the sum of the squared residuals.

OR

- Gradient Descent is an optimization algorithm used to minimize a cost function (e.g., the sum of squared residuals). It iteratively adjusts the parameters to find the values that minimize the cost.

4 **Model Evaluation:**

- Assess the fit of the model using statistics like $R2$, and check the residuals for patterns that might suggest model inadequacy.

5 **Prediction:**

Use the regression equation to predict the dependent variable for new values of the independent variable

Any example **1M**

5a) **Definition of Polynomial Regression** **5M**

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.

Explanation of any example **3M**

5 b) **Any Two applications** **5M**

1. Economics

- **Demand Forecasting:**
  - Model the relationship between price and demand, which is often nonlinear.
- **Income vs. Consumption:**
  - Analyze how consumption patterns change with income levels, which may follow a polynomial trend.

2. Biology

- **Growth Rates:**
  - Model the growth of organisms or populations over time, which may

follow a polynomial curve.

- **Dose-Response Relationships**:
  - Study the effect of drug dosage on biological responses, which may not be linear.

3. **Engineering**
   - **Material Properties**:
     - Predict material properties (e.g., strength, elasticity) under varying conditions (e.g., temperature, pressure).
   - **Signal Processing**:
     - Approximate nonlinear signals or systems using polynomial models.
4. **Environmental Science**
   - **Climate Modeling**:
     - Model relationships between environmental factors (e.g., temperature, CO_2 levels) and outcomes (e.g., ice melt, sea level rise).
   - **Pollution Analysis**:
     - Study the relationship between pollutant concentrations and their effects on health or ecosystems.
5. **Social Sciences**
   - **Education**:
     - Analyze the relationship between study hours and academic performance, which may not be linear.
   - **Psychology**:
     - Model the relationship between stress levels and productivity.
6. **Physics**
   - **Motion Analysis**:
     - Model the trajectory of objects under the influence of forces (e.g., quadratic motion under gravity).
   - **Thermodynamics**:
     - Study the relationship between temperature and energy, which may follow a polynomial trend.
7. **Finance**
   - **Stock Market Analysis**:
     - Model the relationship between time and stock prices, which may exhibit nonlinear trends.
   - **Risk Assessment**:
     - Analyze the relationship between risk factors and financial outcomes.
8. **Medicine**
   - **Drug Efficacy**:
     - Study the relationship between drug dosage and patient response.
   - **Disease Progression**:
     - Model the progression of diseases over time, which may follow a polynomial curve.

9. Agriculture

- **Crop Yield Prediction:**
    - Model the relationship between factors like rainfall, fertilizer use, and crop yield.
- **Soil Analysis:**
    - Study the relationship between soil properties and plant growth.

6a) Concept of Decision tree induction 5M

Decision tree Induction: It is the learning of decision trees from class-labeled training tuples.

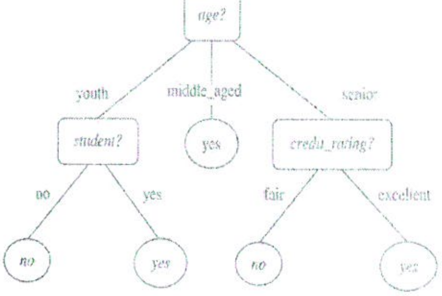
Decision tree:

A decision tree is a flowchart-like tree structure, where each internal node (non leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node.

Decision Tree is a supervised learning method used in data mining for classification and regression methods. It is a tree that helps us in decision-making purposes. The decision tree creates classification or regression models as a tree structure.

It separates a data set into smaller subsets, and at the same time, the decision tree is steadily developed. The final tree is a tree with the decision nodes and leaf nodes. A decision node has at least two branches. The leaf nodes show a classification or decision. Decision trees can deal with both categorical and numerical data.

The following decision tree is for the concept buy a computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class

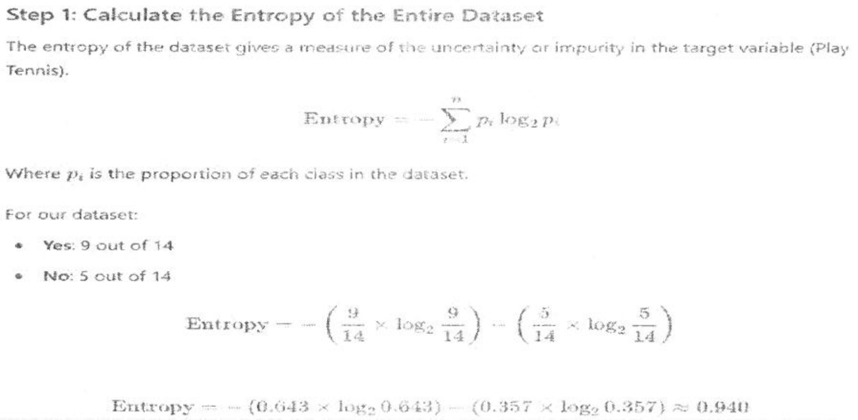


6 b) Representation of a decision tree with example 5M

Example Problem:

Suppose we have a small dataset of 14 samples, each classified as either "Yes" or "No" for whether a person will play tennis based on two features: **Outlook** and **Wind**.

Outlook	Wind	Play Tennis
Sunny	Weak	No
Sunny	Strong	No
Overcast	Weak	Yes
Rain	Weak	Yes
Rain	Weak	Yes
Rain	Strong	No
Overcast	Strong	Yes
Sunny	Weak	No
Sunny	Weak	Yes
Rain	Weak	Yes
Sunny	Strong	Yes
Overcast	Weak	Yes
Overcast	Strong	Yes
Rain	Strong	No



Step 1: Calculate the Entropy of the Entire Dataset

The entropy of the dataset gives a measure of the uncertainty or impurity in the target variable (Play Tennis).

$$\text{Entropy} = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Where p_i is the proportion of each class in the dataset.

For our dataset:

- **Yes**: 9 out of 14
- **No**: 5 out of 14

$$\text{Entropy} = -\left(\frac{9}{14} \times \log_2 \frac{9}{14}\right) - \left(\frac{5}{14} \times \log_2 \frac{5}{14}\right)$$

$$\text{Entropy} = -(0.643 \times \log_2 0.643) - (0.357 \times \log_2 0.357) \approx 0.940$$

Step 2: Calculate the Information Gain for Each Feature

For **Outlook**:

- Sunny:
  - Yes: 2, No: 3
  - Entropy = $-\left(\frac{2}{5} \times \log_2 \frac{2}{5}\right) - \left(\frac{3}{5} \times \log_2 \frac{3}{5}\right) \approx 0.971$
- Overcast:
  - Yes: 4, No: 0
  - Entropy = $-\left(\frac{4}{4} \times \log_2 \frac{4}{4}\right) = 0$
- Rain:
  - Yes: 3, No: 2
  - Entropy = $-\left(\frac{3}{5} \times \log_2 \frac{3}{5}\right) - \left(\frac{2}{5} \times \log_2 \frac{2}{5}\right) \approx 0.971$

Overall Information Gain for Outlook:

$$\text{Information Gain}_{\text{Outlook}} = 0.940 - \left(\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971\right) \approx 0.247$$

For **Wind**:

- Weak:
  - Yes: 6, No: 2
  - Entropy = $-\left(\frac{6}{8} \times \log_2 \frac{6}{8}\right) - \left(\frac{2}{8} \times \log_2 \frac{2}{8}\right) \approx 0.811$
- Strong:
  - Yes: 3, No: 3
  - Entropy = $-\left(\frac{3}{6} \times \log_2 \frac{3}{6}\right) - \left(\frac{3}{6} \times \log_2 \frac{3}{6}\right) = 1$

Overall Information Gain for Wind:

$$\text{Information Gain}_{\text{Wind}} = 0.940 - \left(\frac{8}{14} \times 0.811 + \frac{6}{14} \times 1\right) \approx 0.048$$

Step 3: Calculate the Split Information

For **Outlook**:

$$\text{Split Information}_{\text{Outlook}} = -\left(\frac{5}{14} \times \log_2 \frac{5}{14}\right) - \left(\frac{4}{14} \times \log_2 \frac{4}{14}\right) - \left(\frac{5}{14} \times \log_2 \frac{5}{14}\right) \approx$$

For **Wind**:

$$\text{Split Information}_{\text{Wind}} = -\left(\frac{8}{14} \times \log_2 \frac{8}{14}\right) - \left(\frac{6}{14} \times \log_2 \frac{6}{14}\right) \approx 0.985$$

Step 4: Calculate the Gain Ratio

- **Gain Ratio for Outlook**:

$$\text{Gain Ratio}_{\text{Outlook}} = \frac{0.247}{1.577} \approx 0.157$$

- **Gain Ratio for Wind**:

$$\text{Gain Ratio}_{\text{Wind}} = \frac{0.048}{0.985} \approx 0.049$$

or

Any example

7a) **Any two applications of Naïve bayes** **5M**

1. Text Classification & Spam Filtering
Naïve Bayes is widely used for email spam detection by classifying emails based on word occurrences. It also categorizes documents into topics like sports, politics, etc., by treating words as features.

2. Sentiment Analysis
This classifier can determine whether reviews or social media posts arepositive, negative, or neutral. It works by analyzing the frequencies of emotionally charged words in the text.

3. Medical Diagnosis
Doctors and AI systems use Naïve Bayes to predict diseases based on symptoms and patient history. It estimates the likelihood of a condition given specific symptoms.

4. Credit Scoring & Risk Management
Banks use Naïve Bayes to assess the probability of loan defaults by analyzing attributes like income, credit scqre, and employment status. This helps in automating loan approvals.

5. Recommendation Systems
Platforms like Netflix or Amazon use Naïve Bayes to predict what users might like based on past behavior. It calculates the probability that a user will prefer a particular item.

6. Weather Forecasting
Weather systems use Naïve Bayes to predict conditions like rain or snow by analyzing variables such as temperature, humidity, and pressure. It helps in estimating future weather based on past patterns.

7. Intrusion Detection

Naïve Bayes helps detect abnormal activities in network traffic, like hacking attempts. It classifies network behavior into "normal" or "suspicious" categories based on observed patterns.

7.b) **Impact of different metrics on performance of KNN algorithm, provide examples** **5M**

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
Cosine	Yes	Excellent	No (normalized)	Text, embeddings

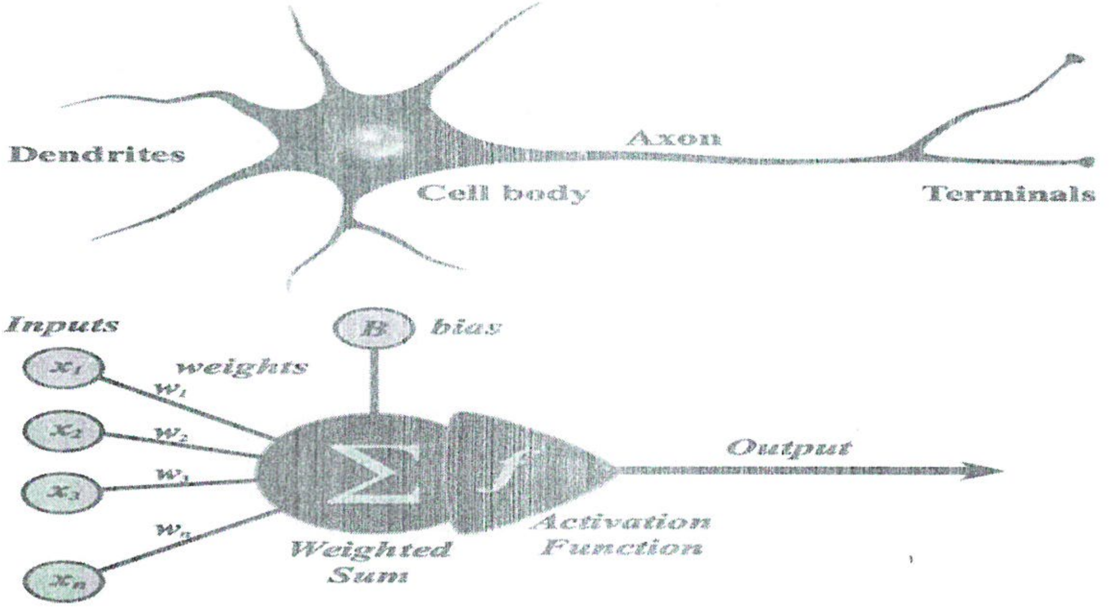
8a) **The relationship between biological neuron and artificial neurons** **5M**

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) **Structure of an artificial neuron** 5M

Artificial Neurons: An artificial neuron, also known as a node or unit, is a simplified mathematical model of a biological neuron. It receives input, processes it, and produces an output.



Key Components of an Artificial Neuron:

Inputs (x1,x2,...,xn): Represent the signals received from other neurons or external sources. Each input is typically associated with a weight.

Weights (w1,w2,...,wn): Represent the strength or importance of each input connection. A higher weight indicates a stronger influence.

Bias (b): An additional input with a constant value (usually 1) and its own weight. The bias allows the neuron to be activated even when all inputs are zero.

Weighted Sum: The inputs are multiplied by their corresponding weights and then summed together

Activation Function (σ): A non-linear function that determines the output of the neuron based on the weighted sum and bias. (Sigmoid (Logistic), Tanh (Hyperbolic Tangent), ReLU (Rectified Linear Unit), etc).

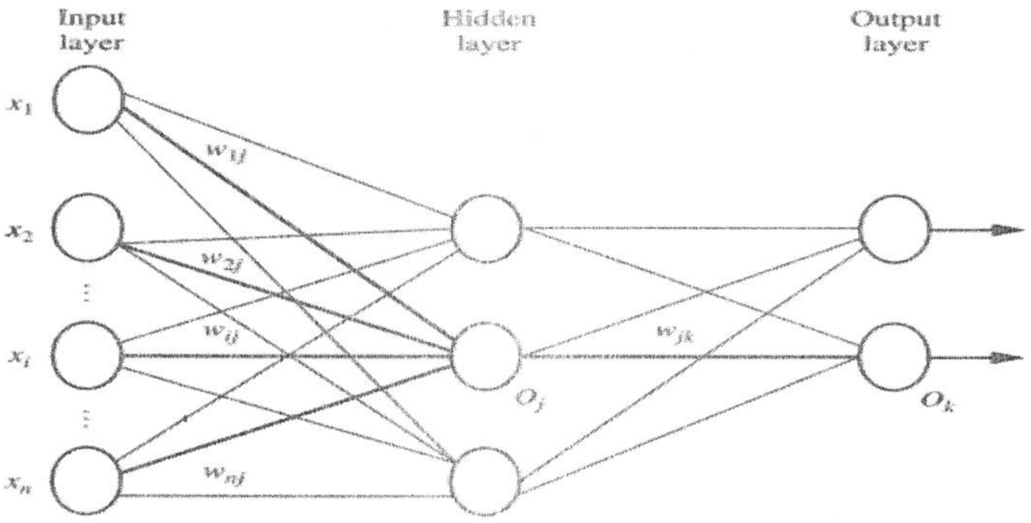
9a) **Concept of a feed forward neural network with a diagram** **5M**

A Multi-Layer Perceptron (MLP) is a type of feedforward artificial neural network that consists of multiple layers of interconnected nodes (neurons).

o 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.

o Feedforward Structure: Data flows in one direction (input → hidden layers → output).

It's a powerful tool for learning complex, non-linear relationships in data, making it suitable for a wide range of supervised learning tasks like classification and regression.
Architecture of MLP:



Input Layer:
o Takes input features.

o No computation, only passes data to the next layer.

Hidden Layer(s):
o Located between the input and output layers.

o There can be one or multiple hidden layers, making the network "deep."

o Each hidden layer neuron receives weighted inputs from all neurons in the previous layer.

o These neurons perform a non-linear transformation on the weighted sum of their inputs using an **activation function**.

o Hidden layers are responsible for extracting complex features and patterns from the input data.

o The number of hidden layers and the number of neurons in each hidden layer are **hyperparameters** that need to be carefully chosen

Output Layer:

o Produces the final output of the network.

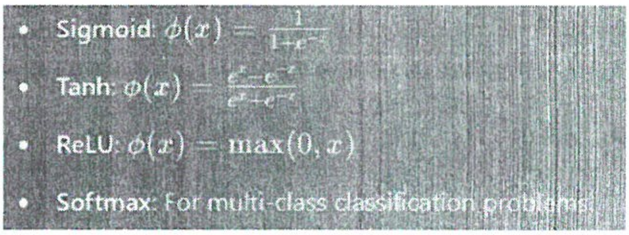
o The number of neurons in the output layer depends on the task:

o **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.

o **Multi-class Classification:** Multiple neurons (with softmax), where each neuron's output represents the probability of the input belonging to a specific class.

Activation Functions:

Used to introduce **non-linearity**, enabling the network to learn complex functions.



- Sigmoid: $\phi(x) = \frac{1}{1+e^{-x}}$
- Tanh: $\phi(x) = \frac{e^x-e^{-x}}{e^x+e^{-x}}$
- ReLU: $\phi(x) = \max(0, x)$
- Softmax: For multi-class classification problems.

9 b) **Concept of margin and support vectors in SVM, The contribution to the classification process** **5M**

Concept of Margin: **2M**

The **margin** is the distance between the hyperplane (decision boundary) and the **closest data points** from each class.

SVM aims to find the hyperplane with the **maximum margin** because a larger margin implies better generalization to unseen data.

There are two types of margins:

o **Hard Margin**: Assumes data is linearly separable with no misclassification.

o **Soft Margin**: Allows some misclassification to handle non-linearly separable data and prevent overfitting.

Mathematically, the margin is defined as:

- Margin = 2 / ||w||

Where w is the weight vector perpendicular to the hyperplane.

Support Vectors: **2M**

o **Support Vectors** are the **data points closest to the hyperplane**.

o These points lie on the boundary of the margin and are critical in defining,the decision boundary.

o Removing a support vector can change the position of the hyperplane.

o Only the support vectors are used to calculate the optimal hyperplane, making the algorithm sparse and efficient.

Contribution to the decision boundary **1M**

The classification of a new data point is determined by which side of the hyperplane it falls on, which is defined by the support vectors.

During prediction, only the support vectors are needed to calculate the classification of a new data point, making the process computationally efficient.

10a) **K-Means algorithm** **5M**

K-Means clustering
K-Means clustering is a popular unsupervised machine learning algorithm used to partition a dataset into a set of k distinct, non-overlapping clusters.

The goal of this algorithm is to group data points with similar characteristics together and separate data points with dissimilar characteristics into different clusters.

The number of clusters, k, is pre-defined by the user.

It is an iterative algorithm that assigns data points to clusters such that the sum of squared distances between points and their cluster centroids is minimized.

A centroid is the mean (average) position of all the points in a cluster. It serves as the "center" of the cluster.

Algorithm: *K*means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.
Input:
K: the number of clusters,
D: a data set containing n objects.
Output: A set of k clusters
Method:
1. Randomly select K objects from D as the initial cluster centers;

2. **Repeat:**
o (re)assign each object to the cluster to which the object is the most similar.

o update the cluster means, that is, calculate the mean value of the objects for each cluster.
3. **until no change;**

10 **b) Compare and Contrast K-Means and K-Medoids** **5M**

Feature	K-Medoids	K-Means
Center Type	Actual data point (medoid)	Mean (centroid)
Robustness	Resistant to outliers	Sensitive to outliers
Distance Metric	Works with any (Manhattan, Euclidean)	Prefers Euclidean distance
Speed	Slower ($O(n^2)$)	Faster ($O(n)$)
Use Case	Small datasets, noisy data	Large datasets, spherical clusters

11a) **Working of agglomerative hierarchical clustering** **5M**

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

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
   - o Merge the two closest clusters
   - o Update the proximity matrix
4. **Until** only a single cluster remains

Explanation:

- **Input**

  A dataset DDD of n objects {x1,x2,…,xn}.

- **Output**

A **dendrogram**—a binary tree whose leaves are the original objects and whose internal nodes record which clusters were merged at what distance.

Step 1: Compute the Proximity Matrix

Choose a distance (or dissimilarity) metric d(xi,xj). Common choices: Euclidean, Manhattan, cosine-dissimilarity, etc.

Build an n×n symmetric matrix P, where Pij = d(xi, xj),

Step 2: Initialize Each Point as Its Own Cluster

- Let C={{x1},{x2},…,{xn}}
- At this stage there are n clusters, each containing exactly one data point.

Step 3: Repeat Until One Cluster Remains

- Merge the Two Closest Clusters
- Based on the chosen linkage criterion and the current proximity matrix, the algorithm identifies the two clusters that have the smallest distance between them.

- These two clusters are then merged into a single new cluster.
- Update the Proximity Matrix: After merging two clusters, the proximity matrix needs to be updated to reflect the distances between this new cluster and all the remaining clusters. The way this update is performed depends on the linkage criterion used.

Step 4: Terminate When One Cluster Remains

11 b) Linkage Criteria in Hierarchical Clustering: **5M**

Linkage criteria are fundamental to Agglomerative Hierarchical Clustering as they determine how distances between clusters are computed during the merging process.

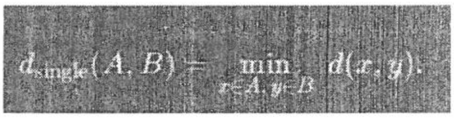
The choice of linkage method significantly impacts the shape and quality of the resulting clusters.

Linkage criteria define how to measure the distance between two clusters based on their constituent points.

The four most widely used methods are:

o Single Linkage
o Complete Linkage
o Average Linkage

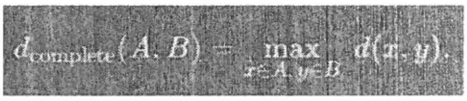
Single Linkage: The distance between two clusters is the minimum distance between any two points in the clusters.



$$d_{\text{single}}(A, B) = \min_{x \in A,\, y \in B} d(x, y).$$

Any example

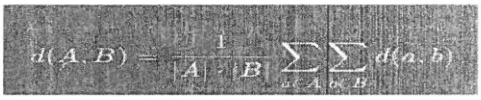
Complete Linkage: The distance between two clusters is the maximum distance between any two points in the clusters.



$$d_{\text{complete}}(A, B) = \max_{x \in A,\, y \in B} d(x, y).$$

Any example

Average Linkage: The distance between two clusters is the average distance between all pairs of points in the clusters.



$$d(A, B) = \frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$$

Any Example