CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | |

Internal Assessment Test - I

Sub:	Machine Learning							Code:	22MBABA403
Date:	**21.08.2025**	Duration:	**90 minutes**	Max Marks:	**50**	Sem:	**IV**	Branch:	**MBA**

SET- II
Scheme of Evaluation

		Marks	OBE CO	OBE RBT
	Part A - Answer Any Two Full Questions (2* 20 = 40 marks)			
1 (a)	What is Machine Learning? Machine Learning is a branch of Artificial Intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to learn patterns from data and make predictions or decisions without being explicitly programmed for every task.	[03]	CO1	L2
(b)	Discuss different types of errors that can occur in a python program	[07]	CO1	L2

1. Syntax Errors

- Occur when the code doesn't follow Python's grammar rules.
- Detected before the program runs.
- Example:
- print("Hello"   # Missing parenthesis
- Error: Syntax Error

2. Runtime Errors (Exceptions)

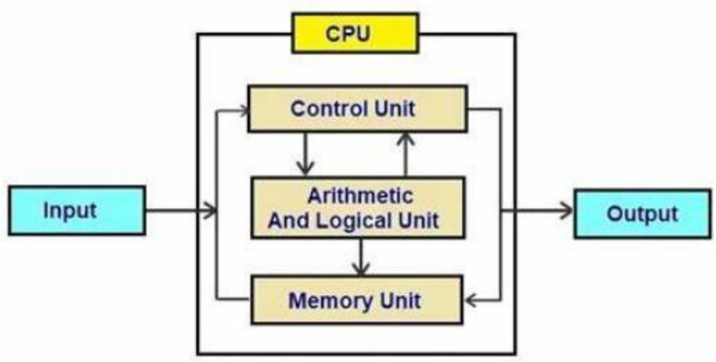
- Happen while the program is running.
- Caused by invalid operations like dividing by zero, accessing invalid index, wrong data type, etc.
- Example:
- x = 10 / 0
- Error: ZeroDivisionError

3. Logical Errors

- Program runs without crashing, but the output is wrong or unexpected.
- Hardest to detect since Python doesn't flag them as errors.
- Example:
- # Intended: find average
- numbers = [10, 20, 30]
- avg = sum(numbers) / len(numbers) # Correct
- avg = sum(numbers) / 100   # Logical error

print(avg)   # Output is wrong, but no error

(c)	Examine the functioning of a computer system with a block diagram.	[10]	CO 1	L3

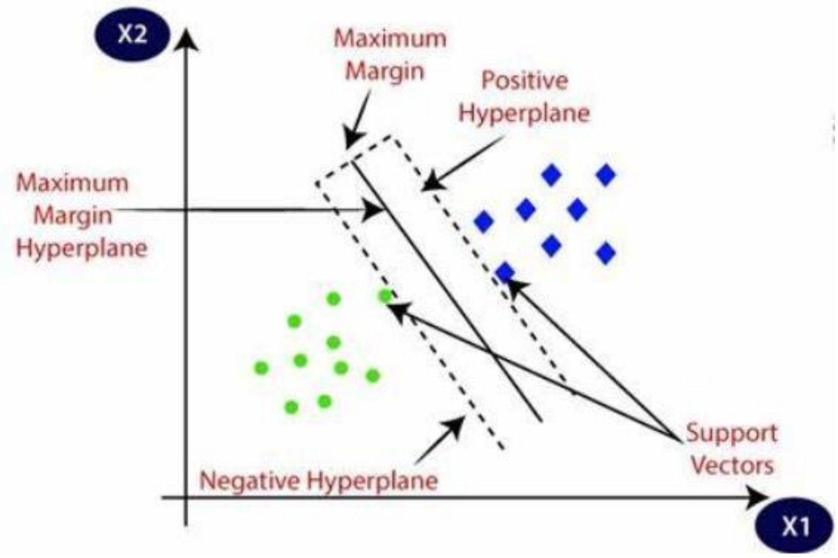


Block Diagram of a Computer

2 (a)	Differentiate supervised and unsupervised machine learning.	[03]	CO 1	L1

Aspect	Supervised Learning	Unsupervised Learning
Definition	The model is trained on a labeled dataset (input + correct output).	The model is trained on an unlabeled dataset (only no output).
Goal	Learn the mapping between inputs and outputs, so it can predict outcomes for new data.	Discover hidden patterns, groupings, or structures
Data Requirement	Requires labeled data (each input has a known output).	Works with unlabeled data (no predefined outcom
Output	Predicts continuous values (regression) or categories (classification).	Identifies clusters, associations, or dimensionality
Examples	Predicting house prices, detecting spam emails, medical diagnosis.	Customer segmentation, market basket analysis, ai detection.
Techniques	Linear Regression, Logistic Regression, Decision Trees, Random Forest, Neural Networks.	K-Means Clustering, Hierarchical Clustering, Prin Component Analysis (PCA), Autoencoders.
Difficulty	Easier to evaluate since predictions can be compared to actual labels.	Harder to evaluate since no ground truth is availab

(b) Illustrate the functioning of SVM with a neat diagram.	[07]	CO1	L3



(c) Critically analyze the implementation of K- means clustering using centroids.	[10]	CO2	L4

- K-Means Clustering is an unsupervised machine learning algorithm used for grouping data points into clusters.
- K-Means identifies clusters by minimizing the sum of squared distances between data points and their respective cluster centroids. It iteratively assigns data points to clusters and updates cluster centroids until convergence is achieved.
- Use-Cases:
  K-Means Clustering: K-Means clustering is used for tasks like customer segmentation, image compression, and anomaly detection. It's ideal for situations where you want to discover underlying patterns or group data points based on their intrinsic similarities.

3(a) In what aspects polynomial regression is different from linear regression? [03] CO1 L1

Aspect	Linear Regression	Polynomial Regression
Nature of Model	Models a straight-line relationship between input (X) and output (Y).	Models a curved / non-linear relationship by adding higher powers of X (e.g., X2,X3X^2, X^3).
Equation	Y=β0+β1X+ϵY = \beta_0 + \beta_1 X + \epsilon	Y=β0+β1X+β2X2+β3X3+⋯+ϵY = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \cdots + \epsilon
Data Fit	Fits well only if the relationship between variables is roughly linear.	Can capture non-linear patterns in the data.
Flexibility	Less flexible, may underfit if relationship is complex.	More flexible, can fit complex curves, but risks overfitting.
Interpretability	Easy to interpret (slope = effect of X).	Harder to interpret as degree increases (hard to explain meaning of X3,X4X^3, X^4).
Visualization	Straight line through the data points.	Smooth curve through the data points (depending on degree).
Use Cases	Predicting salary vs. years of experience, predicting house price vs. size.	Modeling growth trends, disease progression curves, or any non-linear relationship.

(b) Analyze how the architecture of a neural network influences its ability to accurately classify handwritten digits in the MNIST dataset. [07] CO1 L4

The MNIST dataset contains 70,000 images of handwritten digits (0–9), each of size 28×28 pixels (784 features). The neural network's architecture—its layers, neurons, and connections—directly affects how well it learns to classify these digits.

1. Input Layer

- Structure: 784 input nodes (one for each pixel).
- Impact: If input normalization (e.g., scaling pixels 0–1) is applied, learning is faster and more accurate. Poorly preprocessed input slows convergence.

2. Hidden Layers

- Number of Layers:

Shallow networks (1 hidden layer) can capture simple patterns but may struggle with complex digit shapes. Deep networks (multiple hidden layers) can extract hierarchical features (edges → shapes → digit identity).

    - Number of Neurons per Layer: Too few → underfitting (network cannot capture complexity). Too many → risk of overfitting (memorizes training data instead of generalizing).

Example:

- 1 hidden layer with 64 neurons → decent performance (~92–94%).
- 3–5 hidden layers with hundreds of neurons → higher accuracy (~97–98%).

3. Activation Functions

- Common choices: ReLU, Sigmoid, Tanh.
- ReLU is preferred in hidden layers because it helps gradient flow and speeds up training.
- Incorrect activation choice (e.g., only sigmoid) may cause vanishing gradient, hurting accuracy.

4. Output Layer

- 10 output neurons (one per digit, 0–9). Uses Softmax activation to produce probability distribution. Essential for multi-class classification.

5. Regularization Techniques (Part of Architecture)

- Dropout layers reduce overfitting by randomly turning off neurons during training. Batch normalization stabilizes learning by normalizing activations.

6. Convolutional Neural Networks (CNNs) vs Fully Connected Networks

- Fully connected networks (dense layers only): Can classify MNIST, but require more parameters and risk overfitting.
- CNNs:Use convolution + pooling layers to extract spatial features (edges, curves). More efficient and accurate because they exploit the 2D structure of images. Typically achieve >99% accuracy on MNIST.

7. Depth vs Generalization

- Shallow → faster but less powerful.
- Deep → better accuracy but risk of overfitting + computational cost.
- Optimal balance is crucial.

Part B - Compulsory (01*10=10 marks) – CASE STUDY

Given below is the data of weather conditions in which play/no play decisions were taken.

4.

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

(a) Evaluate the suitability of decision tree for the above dataset, if it is to be predicted whether playing happens or not. [5] CO2 L5

Reason	Explanation
Handles categorical data	Decision trees can directly work with categorica without encoding.
Interpretable	Each decision path can be expressed as a simple rule.
Captures non-linear relationships	For example, "Play = No if Outlook = Sunny AN = High."
No need for feature scaling	Unlike algorithms like SVM or KNN, decision t require normalization.
Can handle small datasets effectively	Works well even if the dataset has fewer rows (l Tennis dataset).
Flexible	Can model complex interactions between attribu Outlook + Humidity).

(b)Explain the process of computing entropy and constructing the decision tree accordingly [5] CO2 L3

1. Understanding Entropy

Entropy (E) measures the **impurity or disorder** in a dataset:

$$E(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

Where:

- S = set of examples
- c = number of classes (Yes/No)
- p_i = proportion of examples in class i

Interpretation:

- Entropy = 0 → all examples belong to one class (pure)
- Entropy = 1 → examples are evenly split between classes

2. Compute Entropy for the Entire Dataset

- Dataset: 14 records
- Target: Play Tennis = Yes (9), No (5)

$$p_{Yes} = \frac{9}{14}, \quad p_{No} = \frac{5}{14}$$

$$E(S) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right)$$

Step-by-step calculation:

1. $\frac{9}{14} \approx 0.643, \frac{5}{14} \approx 0.357$
2. $-0.643 \times \log_2 0.643 \approx -0.643 \times (-0.643) \approx 0.413$
3. $-0.357 \times \log_2 0.357 \approx -0.357 \times (-1.485) \approx 0.530$
4. Total Entropy $E(S) \approx 0.413 + 0.530 = 0.943$

So, **entropy of the full dataset** = **0.94 bits**

3. Compute Entropy for Each Attribute (Example: Outlook)

Attribute **Outlook** has 3 values: Sunny, Overcast, Rain

Outlook	Yes	No	Total	Entropy
Sunny	2	3	5	$E = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} \approx 0.971$
Overcast	4	0	4	$E = -1\log_2 1 - 0\log_2 0 = 0$
Rain	3	2	5	$E = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} \approx 0.971$

4. Compute Information Gain (IG)

$$IG(S, \text{Outlook}) = E(S) - \sum_{v \in Values} \frac{|S_v|}{|S|} E(S_v)$$

Step-by-step:

$$IG = 0.94 - \left(\frac{5}{14} \cdot 0.971 + \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0.971\right)$$

$$IG = 0.94 - (0.347 + 0 + 0.347) = 0.94 - 0.694 = 0.246$$

- Repeat this **for all other attributes** (Temperature, Humidity, Wind)
- **Choose attribute with highest IG** as the root node

5. Split the Dataset & Recurse

1. Root node → attribute with highest IG (say, Outlook)
2. For each branch (Sunny / Overcast / Rain), create a **subset**
3. Compute **entropy and IG** for remaining attributes in that subset
4. Continue splitting **until stopping conditions**:
   - Subset is pure (all Yes or No)
   - No attributes left

6. Assign Class Labels at Leaf Nodes

- Leaf nodes = final decision (Yes / No)
- Example path:
  - Outlook = Overcast → Leaf = Yes
  - Outlook = Sunny & Humidity = High → Leaf = No

7. Resulting Decision Tree (Simplified)

```
              [Outlook]
           /      |      \
        Sunny  Overcast   Rain
       /   \       |      /   \
  Humidity   Yes  Wind    Humidity
 High/Normal          Weak/Strong
```