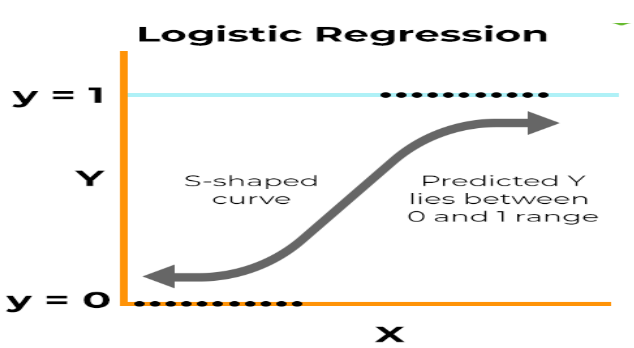
III/IV B.Tech (Regular/Supplementary) DEGREE EXAMINATION

May, 2025 **Common to CB, DS, CS, & IT**

Sixth Semester **Machine Learning (20CB602/20DS602/20CS602/20IT602)**

Scheme of Evaluation

PART-A

			CO	BL	M
1	a)	**Define Label in Machine Learning.** **Sol: -** Label is the thing we want to predict. Example Y=a+wx is the linear graph then Y is the Label. Other words label is the output also.	CO1	L1	1M
	b)	**Write the difference between the supervised and unsupervised learning.** **Sol: -** In supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels. Unsupervised learning includes the training data is unlabelled.	CO1	L1	1M
	c)	**Define the term Parameter tuning in machine learning.** **Sol: -** Parameters are the variables in the model that the programmer generally decides. At a particular value of your parameter, the accuracy will be the maximum. Parameter tuning refers to finding these values. accuracy can be improved in any way by tuning the parameters present in your model.	CO1	L1	1M
	d)	**Write the difference between the Batch and Stochastic gradient descent techniques.** **Sol:-** In Batch gradient descent the gradient can be computed using total training sample. In stochastic technique gradient computed by random training sample.	CO1	L1	1M
	e)	**State the Bayes rule.** **Sol:-** $$P(h\|D) = \frac{P(D\|h)P(h)}{P(D)}$$ P(h) : prior probability of h P(D) : prior probability that training data D will be observed P(D|h) : prior knowledge P(h|D) : posterior probability of h , given D	CO2	L1	1M
	f)	**State the MAP hypothesis in Bayes theorem** **Sol:-** $$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}} P(h\|D)$$ $$= \underset{h \in H}{\operatorname{argmax}} \frac{P(D\|h)P(h)}{P(D)}$$ $$= \underset{h \in H}{\operatorname{argmax}} P(D\|h)P(h)$$	CO2	L1	1M
	g)	**Define the decision tree.** **Sol:-** Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.	CO2	L1	1M
	h)	**Define the logistic regression with neat graph.** **Sol:-** Logistic regression (logit regression) is a type of regression analysis used for predicting the outcome of a categorical dependent variable.  	CO2	L1	1M
	i)	**Write the differences between the precision and recall.** **Sol:-** In precision consider both positive and negative classified samples. Precision $(\boldsymbol{P}) = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$ In Recall prefer only positive samples. Recall$(R) = \frac{TP}{TP+FN}$	CO3	L1	1M
	j)	**Mention any two types of cross-validation methods.**	CO3	L1	1M

		Sol:- K-Fold Cross-Validation Stratified K-Fold Cross-Validation Leave- One-Out Cross-Validation (LOOCV) Leave-p-Out Cross-Validation (LpOCV) Repeated K-Fold Cross-Validation Nested Cross-Validation and Time Series Cross-Validation.			
	k)	**Write the difference between the bagging and pasting techniques.** **Sol:-** sampling is performed with replacement is called bagging. Bagging allows training instances to be sampled several times for the same predictor. sampling is performed without replacement is called pasting. pasting allow training instances to be sampled several times across multiple predictors.	CO3	L1	1M
	l)	**Define the PAC learning model.** **Sol:-** PAC learning model, considers the questions of how many training examples and how much computation are required in order to learn various classes of target functions within this PAC model.	CO4	L1	1M
	m)	**Give the example for Instance based learning.** **Sol:-** KNN, kernel machines and RBF networks	CO4	L1	1M
	n)	**What is the principle of EM algorithm?** **Sol:-** To fit a Gaussian Mixture Model to the data, use the Expectation-Maximization (EM) algorithm, which is an iterative method that optimizes the parameters of the Gaussian distributions (mean, covariance, and mixing coefficients). It works in two main steps: ➢ Expectation Step (E-step): ➢ Maximization Step (M-step):	CO4	L1	1M

PART-B

Unit-I

2	a)	**Explain the different supervised and unsupervised types of machine learning algorithms and applications of each algorithm.** **Sol: -**	CO1	L2	7M

Supervised Algorithm	Description	Applications
Linear Regression	Predicts continuous values	house prices prediction, Stock market prediction
Logistic Regression	For binary classification	Email Spam Detection
Decision Trees	Tree-like models for classification/regression	Medical Diagnosis
Random Forest	Ensemble of decision trees to improve accuracy	Credit Scoring
Support Vector Machines (SVM)	Classifies data by finding the best hyperplane	Medical Diagnosis
K-Nearest Neighbors (KNN)	Classifies based on closest training examples	recommendation systems
Gradient Boosting (e.g., XGBoost, LightGBM)	Ensemble methods that build models sequentially to minimize errors	fraud detection, medical diagnosis

UnSupervised Algorithm	Description	Applications
K-Means Clustering	Groups data into K clusters based on similarity	Customer Segmentation
Hierarchical Clustering	Builds nested clusters in a tree-like structure	social network analysis, and bioinformatics
Gaussian Mixture Models (GMM)	Probabilistic model for soft clustering	Customer Segmentation

	b)	**Compare and explain the Lasso and Ridge regression models in detail.**	CO1	L4	7M

Sol:-

Characteristic	Ridge Regression	Lasso Regression
Regularization Type	Applies **L2 regularization**, adding a penalty term proportional to the **square of the coefficients**	Applies **L1 regularization**, adding a penalty term proportional to the **absolute value of the coefficients**.
Feature Selection	Does **not perform feature selection**. All predictors are retained, although their coefficients are reduced in size to minimize overfitting	Performs **automatic feature selection**. Less important predictors are completely excluded by setting their coefficients to zero.
When to use	Best suited for situations where **all predictors are potentially relevant**, and the goal is to reduce overfitting rather than eliminate features	Ideal when you suspect that only a **subset of predictors** is important, and the model should focus on those while ignoring the irrelevant ones.
Output model	Produces a model that includes **all features**, but their coefficients are smaller in magnitude to prevent overfitting	Produces a model that is **simpler**, retaining only the most significant features and ignoring the rest by setting their coefficients to zero.
Impact on Prediction	Reduces the magnitude of coefficients, shrinking them towards zero, but does not set any coefficients exactly to zero. All predictors remain in the model	Shrinks some coefficients to **exactly zero**, effectively removing their influence from the model. This leads to a simpler model with fewer features
Computation	Generally faster as it doesn't involve feature selection	May be slower due to the feature selection process
Example Use Case	Use when you have many predictors, all contributing to the outcome (e.g., predicting house prices where all features like size, location, etc., matter)	Use when you believe only some predictors are truly important (e.g., genetic studies where only a few genes out of thousands are relevant).

(OR)

3	a)	**Analyze the challenges faced in machine leaning environment in detail.**	CO1	L4	7M

Sol:-

1. Insufficient Quantity of Training Data
2. Nonrepresentative Training Data
3. Poor-Quality Data
4. Irrelevant Features
5. Overfitting the Training Data
6. Underfitting the Training Data
7. Stepping Back
8. Testing and Validating
9. Hyperparameter Tuning and Model Selection
10. Data Mismatch

Insufficient Quantity of Training Data

- Machine Learning takes a lot of data for most Machine Learning algorithms to work properly.
- Even for very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples.

Nonrepresentative Training Data

- Define Nonrepresentative Training Data.
- If there is less training data, then there will be a sampling noise (sample is not perfectly representative) in the model, called the non-representative training set. It won't be accurate in predictions.

Poor-Quality Data

- if your training data is full of errors, outliers (different from other), and noise (e.g., due to poor quality measurements), it will make it harder for the system to detect the underlying patterns, so your system is less likely to perform well.

Irrelevant Features

- training data contains enough relevant features and not too many irrelevant ones.
- **feature engineering involves:**
- *a)* ***Feature selection*****:** selecting the most useful features to train on among existing features.
- b) ***Feature extraction*****:** combining existing features to produce a more useful.
- c) ***Creating new features*** by gathering new data.

Overfitting the Training Data

- Overfitting - model performs well on the training data, but it does not generalize well (performs poorly on unseen data).

  The **possible solutions for solving overfitting are:**

- To simplify the model by selecting one with fewer parameters, by **reducing the number of attributes in the training data** or by constraining (control and limit something) the model.
- To gather more training data.
- To reduce the **noise in the training data** (e.g., fix data errors and remove outliers)

Underfitting the Training Data

- underfitting is the opposite of overfitting: it occurs when your model is too simple to learn the underlying structure of the data.

  The **possible solutions for solving underfitting are** :

- Selecting a more powerful model, with more parameters
- Feeding better features to the learning algorithm (**feature engineering)**
- **Reducing the constraints** on the model.

Stepping Back

- considering so many concepts may be feeling a little lost, so let's step back and look
- There are many different types of ML systems: supervised or not, batch or online, instance-based or model-based, and so on.
- • In a ML project you gather data in a training set, and you feed the training set to a learning algorithm.
- If the algorithm is model-based it tunes some parameters to fit the model to the training set (i.e., to make good predictions on the training set itself) and then hopefully it will be able to make good predictions on new cases as well.
- If the algorithm is instance-based, it just learns the examples by heart and generalizes to new instances by comparing them to the learned instances using a similarity measure.

Testing and Validating

- split your data into two sets: the training set and the test set.
- As these names imply, train your model using the training set, and you test it using the test set.
- The error rate on new cases is called the generalization error (or out of sample error), and by evaluating your model on the test set, you get an estimate of this error. This value tells you how well your model will perform on instances it has never seen before.
- If the training error is low (i.e., your model makes few mistakes on the training set) but the generalization error is high, it means that your model is overfitting the training data.

Hyperparameter Tuning and Model Selection

- hyperparameter is a parameter whose value is used to control the learning process.
- Model architecture.
- Learning rate.
- Number of **epochs**.
- Number of clusters in a clustering algorithm.
- Number of branches in a decision tree.

Data Mismatch

- One solution for Data Mismatch → is to hold out part of the training pictures (from the web) in another set the train-dev set.

- After the model is trained (on the training set, not on the train-dev set), you can evaluate it on the train-dev set: if it performs well, then the model is not overfitting the training set, so if performs poorly on the validation set, the problem must come from the data mismatch.
- try to tackle this problem by pre-processing the web images to make them look more like the pictures that will be taken by the mobile app, and then retraining the model.

b) **Code and explain the machine learning pipeline for classifying the iris species using Sci-kit learn.** CO1 L2 7M

Sol:-

```
from sklearn.datasets import load_iris
```

load iris dataset

```
iris_dataset = load_iris()
```

Print the keys of iris dataset

```
print("Keys of iris_dataset: \n{}".format(iris_dataset.keys()))
```

```
Keys of iris_dataset:
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])
```

```
iris_dataset
```

```
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 5. , 1.9],
       [6.5, 3. , 5.2, 2. ],
       [6.2, 3.4, 5.4, 2.3],
       [5.9, 3. , 5.1, 1.8]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
 'frame': None,
 'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
 'DESCR': '.. iris dataset:\n\nIris plants dataset\n--------------------\n\n**Data Set Characteristics:**\n\n    :Number of I
```

```
print(iris_dataset['DESCR'][:193] + "\n...")
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, pre
...
```

Target names of Iris dataset (outputs)

```
print("Target names: {}".format(iris_dataset['target_names']))
```

```
Target names: ['setosa' 'versicolor' 'virginica']
```

Feature names of Iris dataset (Inputs)

```
print("Feature names: \n{}".format(iris_dataset['feature_names']))
```

```
Feature names:
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

shape of iris_dataset (Number of Instances and attributes)

```
print("Shape of data: {}".format(iris_dataset['data'].shape))
```

```
Shape of data: (150, 4)
```

```
print("First five rows of data:\n{}".format(iris_dataset['data'][:5]))
```

```
First five columns of data:
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]]
```

Print target values of datset

```
print("Target:\n{}".format(iris_dataset['target']))
```

```
Target:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

Unit-II

4 a) **Train the Naïve Bayes model and classify (today(X)= (Rainy, Hot, High, False)) by using the dataset given below.** — CO2 — L3 — 7M

Sol: -

- Learning Phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
True	3/9	3/5
False	6/9	2/5

P(Play=*Yes*) = 9/14   *P*(Play=*No*) = 5/14

- Classify (today(X) = (Rainy, Hot, High, False))
- P(Yes|today)=P(RainyOutlook|Yes)P(HotTemperature|Yes)P(High Humidity|Yes)P(FalseWind|Yes)*P(Yes)/ P(today)
- P(Yes|today)=3/9.2/9.3/9.6/9.9/14≈0.0106
-
- P(No|today)=P(RainyOutlook|No)P(HotTemperature|No)P(HotHumidity|No)P(FalseWind|No)*P(No)/ P(today)
- P(No|today)=2/5.2/5.4/5.2/5.5/14≈0.0183
- Since, P(today) is common in both probabilities, we can ignore P(today) and find proportional probabilities as:
- P(No|today)>P(Yes|today)
- So, **prediction that Tennis would be played is 'NO'.**

b) **Explain the Logistic Regression algorithm in detail.** — CO2 — L2 — 7M

Sol:-

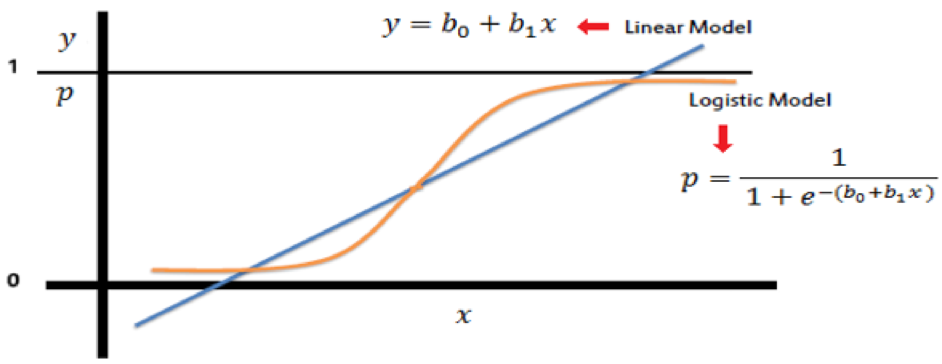
Logistic Regression: Logistic regression is a type of regression used for classification problems, where the output variable is categorical in nature. Logistic regression uses a **logistic function to predict the probability** of the input belonging to a particular category.

In logistic regression, dependent variable (Y) is binary (0,1) and independent variables (X) are continuous in nature.

Sigmoid Function

It is a mathematical function having a characteristic that can take any real value and map it to between 0 to 1 shaped like the letter "S". The sigmoid function also called a logistic function. $Y = 1 / 1+e^{-z}$.

- Logistic regression finds a linear decision boundary$w_0x_0 + w_1x_1 + \cdots + w_dx_d = 0$ such that
- $P(Y=0|\bar{X}) = \frac{1}{1+e^{\sum_{j=0}^{d} w_jx_j}}$ is maximum for class 0 samples and minimum for class 1 samples. $\bar{X} = [x_0, x_1, \ldots x_d]$
- $P(Y=1|X) = \frac{e^{\sum_{j=0}^{d} w_jx_j}}{1+e^{\sum_{j=0}^{d} w_jx_j}}$ is maximum for class 1 samples and minimum for class 0 samples.
- The sum of the probabilities is equal to 1.
- $P(Y=0|\bar{X}) + P(Y=1|X) = \frac{1}{1+e^{\sum_{j=0}^{d} w_jx_j}} + \frac{e^{\sum_{j=0}^{d} w_jx_j}}{1+e^{\sum_{j=0}^{d} w_jx_j}} = 1$
- $z = \sum_{j=0}^{d} w_jx_j$
- $P(Y=0|X) = \frac{1}{1+e^z}$
- $P(Y=1|X) = \frac{e^z}{1+e^z}$
- examples are
- An email being spam (1) or ham (0)
- A tumor being malignant (1) or benign (0).



(OR)

5	a)	**Construct Decision Tree using Gini Impurity for the below dataset of root and next level.**	CO2	L3	7M

Sol:-

<u>For Root level</u>

Outlook	Yes	No
Sunny	2	3
Overcast	4	0
Rain	3	2

$$GI(Outlook = Sunny) = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = \frac{12}{25}$$

$$GI(Outlook = Overcast) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$$

$$GI(Outlook = Rain) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = \frac{12}{25}$$

$$GI(Outlook) = \frac{5}{14} * \frac{12}{25} + \frac{4}{14} * 0 + \frac{5}{14} * \frac{12}{25} = 0.342$$

Temperature	Yes	No
Hot	2	2
Mild	4	2
Cool	3	1

$$GI(Temperature = Hot) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = \frac{1}{2}$$

$$GI(Temperature = Mild) = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = \frac{4}{9}$$

$$GI(Temperature = Cool) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = \frac{6}{16}$$

$$GI(Temperature) = \frac{4}{14} * \frac{1}{2} + \frac{6}{14} * \frac{4}{9} + \frac{4}{14} * \frac{6}{16} = 0.439$$

Humidity	Yes	No
High	3	4
Normal	6	1

$$GI(Humidity = High) = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = \frac{24}{49}$$

$$GI(Humidity = Normal) = 1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2 = \frac{12}{49}$$

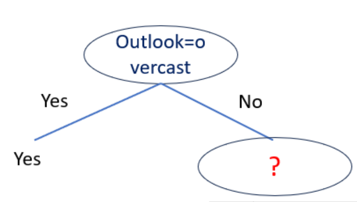
$$GI(Humidity) = \frac{7}{14} * \frac{24}{49} + \frac{7}{14} * \frac{12}{49} = 0.367$$

Wind	Yes	No
False	6	2
True	3	3

$$GI(Wind = False) = 1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2 = \frac{6}{16}$$

$$GI(Wind = True) = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = \frac{1}{2}$$

$$GI(Wind) = \frac{8}{14} * \frac{6}{16} + \frac{6}{14} * \frac{1}{2} = 0.428$$



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D3	Overcast	Hot	High	False	Yes
D7	Overcast	Cool	Low	True	Yes
D12	Overcast	Mild	High	True	Yes
D13	Overcast	Hot	Low	False	Yes

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	False	No
D2	Sunny	Hot	High	True	No
D4	Rain	Mild	High	False	Yes
D5	Rain	Cool	Low	False	Yes
D6	Rain	Cool	Low	True	No
D8	Sunny	Mild	High	False	No
D9	Sunny	Cool	Low	False	Yes
D10	Rain	Mild	Low	False	Yes
D11	Sunny	Mild	Low	True	Yes
D14	Rain	Mild	High	True	No

For Next Level

outlook	Yes	No
Sunny	2	3
Rain	3	2

$$GI(Outlook = Sunny) = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = \frac{12}{25}$$

$$GI(Outlook = Rain) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = \frac{12}{25}$$

$$GI(Outlook) = \frac{5}{10} * \frac{12}{25} + \frac{5}{10} * \frac{12}{25} = \frac{12}{25} = 0.48$$

Temperature	Yes	No
Hot	0	2
Mild	3	2
Cool	2	1

$$GI(Temperature = Hot) = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$GI(Temperature = Mild) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = \frac{12}{25}$$

$$GI(Temperature = Cool) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$GI(Temperature) = \frac{2}{10} * 0 + \frac{5}{10} * \frac{12}{25} + \frac{3}{10} * \frac{4}{9} = 0.373$$

Humidity	Yes	No
High	1	4
Normal	4	1

$$GI(Humidity = High) = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2 = \frac{8}{25}$$

$$GI(Humidity = Normal) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2 = \frac{8}{25}$$

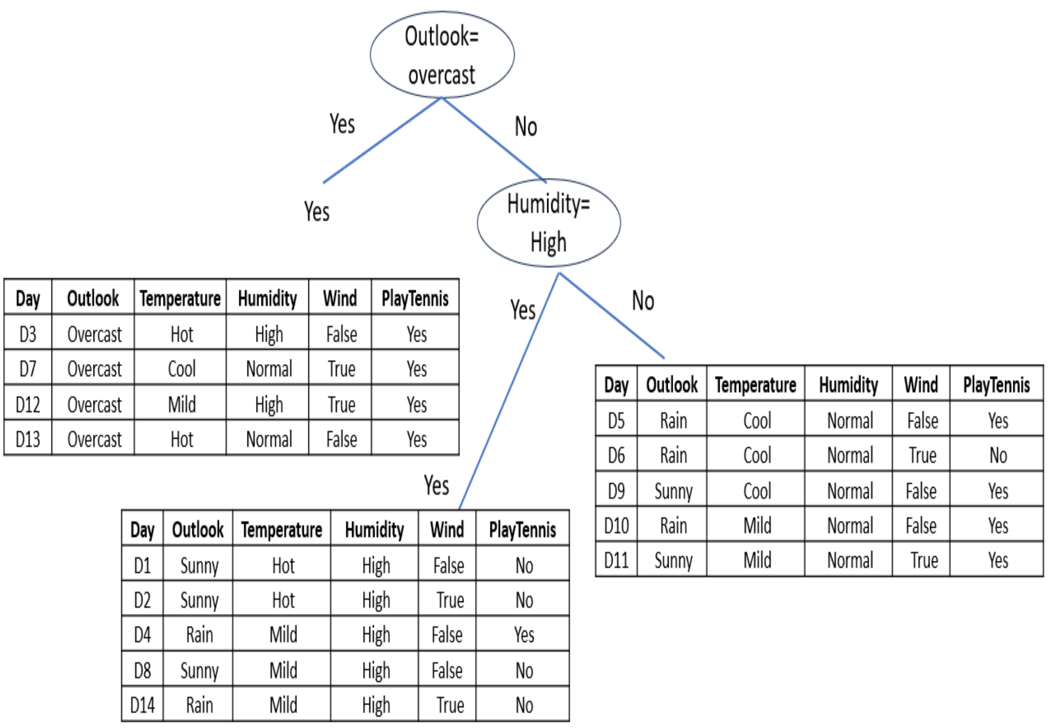
$$GI(Humidity) = \frac{5}{10} * \frac{8}{25} + \frac{5}{10} * \frac{8}{25} = 0.32$$

Wind	Yes	No
False	4	2
True	1	3

$$GI(Wind = Weak) = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = \frac{16}{36}$$

$$GI(Wind = Strong) = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = \frac{6}{16}$$

$$GI(Wind) = \frac{6}{10} * \frac{4}{9} + \frac{4}{10} * \frac{3}{8} = 0.414$$



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D3	Overcast	Hot	High	False	Yes
D7	Overcast	Cool	Normal	True	Yes
D12	Overcast	Mild	High	True	Yes
D13	Overcast	Hot	Normal	False	Yes

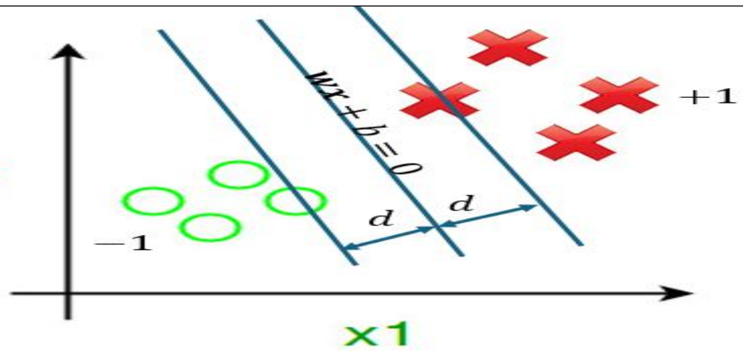
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D5	Rain	Cool	Normal	False	Yes
D6	Rain	Cool	Normal	True	No
D9	Sunny	Cool	Normal	False	Yes
D10	Rain	Mild	Normal	False	Yes
D11	Sunny	Mild	Normal	True	Yes

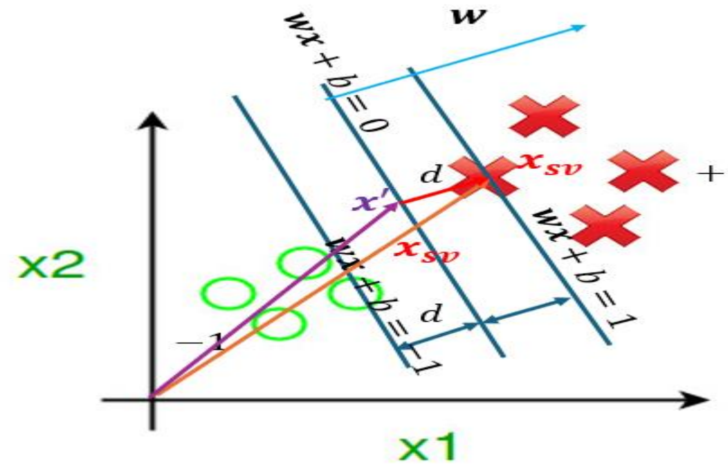
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	False	No
D2	Sunny	Hot	High	True	No
D4	Rain	Mild	High	False	Yes
D8	Sunny	Mild	High	False	No
D14	Rain	Mild	High	True	No

b) **Describe the linear SVM technique in detail with necessary diagrams.** CO2 L2 7M

Sol:-

- Support Vector Machine (SVM) is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection.
- SVM aims to find the optimal hyperplane in an N-dimensional space to separate data points into different classes. The algorithm maximizes the margin between the closest points of different classes.
- $\boldsymbol{wx} + b = 0$ such that the decision boundary/hyperplane correctly classifies the samples and the distance/margin from the closest points of either class is maximum. i.e d is maximum.
- The closest points are called Support vectors.
- $\boldsymbol{w}$ is the vector perpendicular to the hyperplane
- Let $\boldsymbol{x_{sv}}$ be the support vector.
- Drop a perpendicular from $\boldsymbol{x_{sv}}$ to the decision boundary. It represents the distance d.
- Let it cut the hyper plane at $\boldsymbol{x'}$
- Vectorially, $\boldsymbol{x'} + d\frac{\boldsymbol{w}}{||\boldsymbol{w}||} = \boldsymbol{x_{sv}}$
- (||w|| represents the Euclidean norm of the weight vector w)
- $\boldsymbol{x'} = \boldsymbol{x_{sv}} - d\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$
- But $\boldsymbol{x'}$ lies on the hyperplane. So, $\boldsymbol{wx'} + b = 0$
- To maximize d we can minimize $||\boldsymbol{w}||$
- To be more stable, minimize $||\boldsymbol{w}||^2$
- Find $\boldsymbol{w}$ (Coefficient) and b (Intercept value)
- For all the samples with class +1 hyper plane is $\boldsymbol{wx} + b \geq 1$
- For all the samples with class -1 hyper plane is $\boldsymbol{wx} + b \leq -1$
- Thus, the optimised cost function is,
- $minimize\ \frac{1}{2}||\boldsymbol{w}||^2$
- such that $y(wx + b) \geq 1$





Dataset for 4(a) & 5(a)

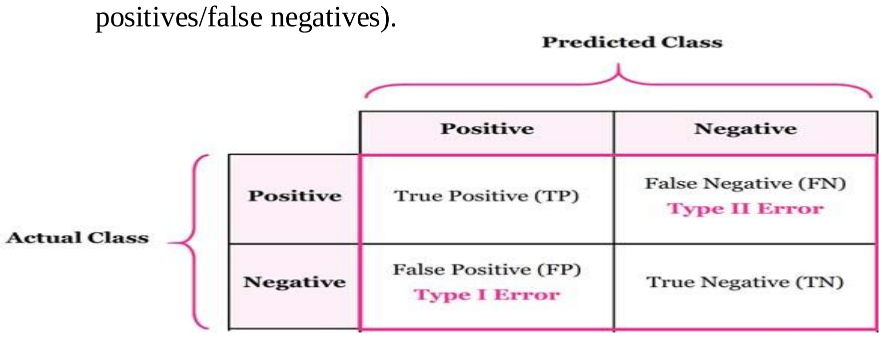
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	False	No
D2	Sunny	Hot	High	True	No
D3	Overcast	Hot	High	False	Yes
D4	Rain	Mild	High	False	Yes
D5	Rain	Cool	Normal	False	Yes
D6	Rain	Cool	Normal	True	No
D7	Overcast	Cool	Normal	True	Yes
D8	Sunny	Mild	High	False	No
D9	Sunny	Cool	Normal	False	Yes
D10	Rain	Mild	Normal	False	Yes
D11	Sunny	Mild	Normal	True	Yes
D12	Overcast	Mild	High	True	Yes
D13	Overcast	Hot	Normal	True	Yes
D14	Rain	Mild	High	True	No

Unit-III

6 a) **There are 18 instances of Class A and 7 instances of Class B. After training, the model predicted 12 instances of Class A and 4 instances of Class B correctly. Construct Confusion matrix and find the accuracy, precision, recall, F1-score, specificity and False positive rate.** CO3 L3 7M

Sol:-

- In a confusion matrix, the rows represent the actual classes, while the columns represent the predicted classes.
- Each cell in the matrix represents the number of instances where the actual class was predicted correctly (true positives/true negatives), incorrectly (false positives/false negatives).

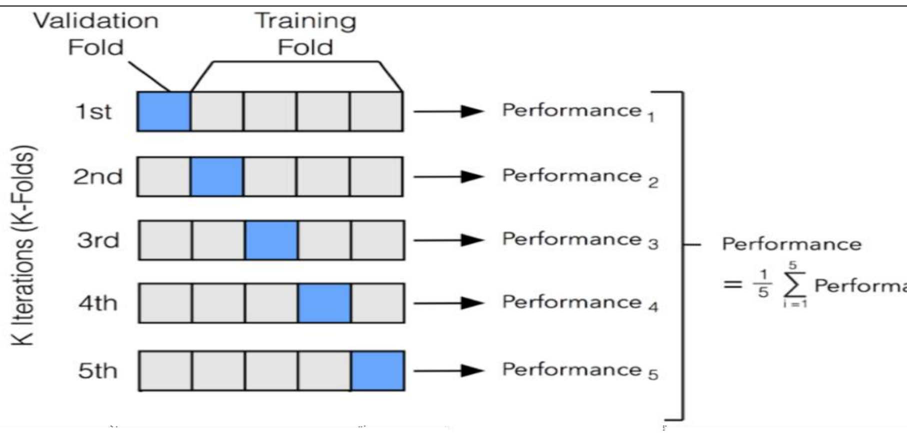


Confusion matrix for the given problem:

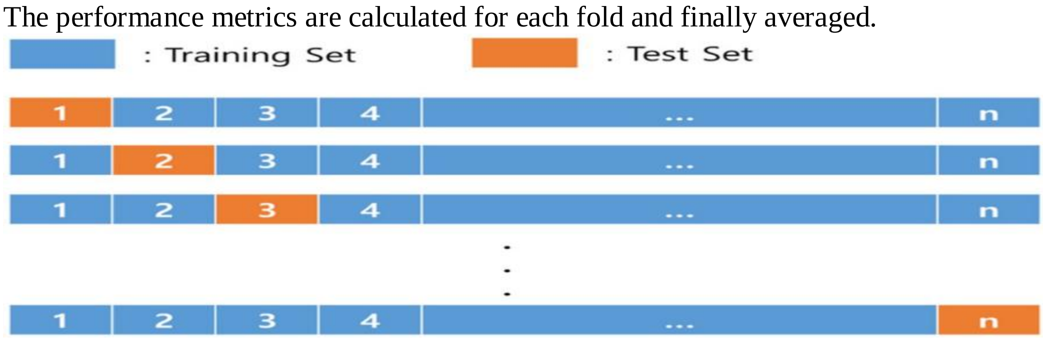
	Predicted Positive	Predicted Negative
Actual Positive	12	6
Actual Negative	3	4

$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$ =12+4/25=16/25=0.64

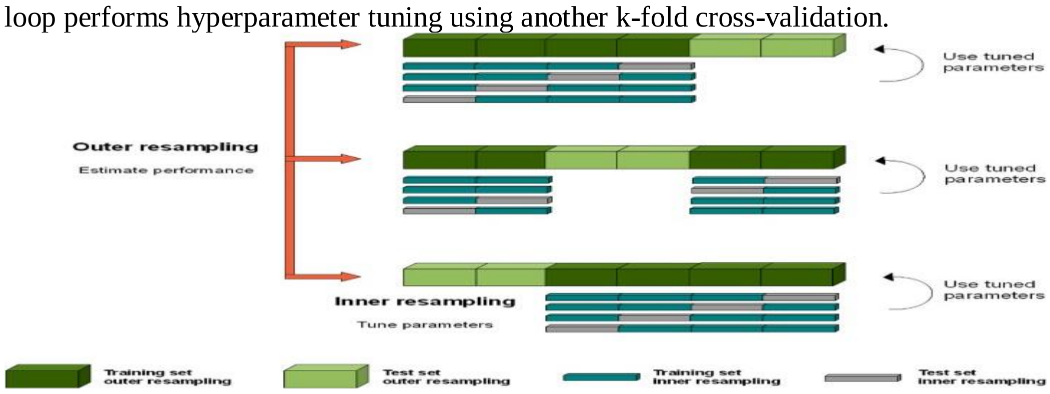
		Precision $(P)=\frac{TP}{TP+FP}$= 12/15=0.80 Recall$(R)=\frac{TP}{TP+FN}$ =12/18=0.66 F1 Score= $\frac{2PR}{P+R}$ = (2*0.80*0.66)/(0.80+0.66)=1.056/1.46=0.723 Specificity$(S)=\frac{TN}{TN+FP}$ =4/7=0.57 $FPR=\frac{FP}{TN+FP}$ = 3/7=0.429			
	b)	**Explain the Random Forest technique in detail and mention the applications, advantages and disadvantages of it.** **Sol:-** • Random forests are a specific implementation of bagging that is applied to decision trees. • In addition to sampling from the training data, random forests also introduce randomness in the feature selection process. • Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset • Instead of considering all features at each split, random forests randomly select a subset of features to consider for splitting at each node of the tree. • This randomness further diversifies the individual trees and prevents them from being highly correlated, which can lead to improved generalization performance. • When making predictions, random forests aggregate the predictions of all the trees, typically using a majority vote for classification or averaging for regression. • The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. • The Working process can be explained in the below steps • Step-1: Select random K data points from the training set. • Step-2: Build the decision trees associated with the selected data points (Subsets). • Step-3: Choose the number N for decision trees that you want to build. • Step-4: Repeat Step 1 & 2. • Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes. **Applications of Random Forest** • **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk. • **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified. • **Land Use:** We can identify the areas of similar land use by this algorithm. • **Marketing:** Marketing trends can be identified using this algorithm. **Advantages of Random Forest** • Random Forest is capable of performing both Classification and Regression tasks. • It is capable of handling large datasets with high dimensionality. • It enhances the accuracy of the model and prevents the overfitting issue. **Disadvantages of Random Forest** • Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks. •	CO3	L2	7M
		(OR)			
7	a)	**Describe in details the different types of cross-validation methods.** **Sol:-** • Cross-validation is a statistical technique used in machine learning to evaluate how well a predictive model will generalize to new, unseen data. • **cross_val_score()** function to evaluate your SGD Classifier model using K-fold cross-validation, with three folds. Remember that K-fold cross-validation means splitting the training set into K-folds (in this case, three), then Training the model k times, holding out a different fold each time for evaluation. • The most common variants are : **K-Fold Cross-Validation**, **Stratified K-Fold Cross-Validation**, **Leave- One-Out Cross-Validation** (LOOCV), **Leave-p-Out Cross-Validation** (LpOCV), **Repeated K-Fold Cross-Validation**, **Nested Cross-Validation and Time Series Cross-Validation.** **1. K-Fold Cross-Validation** The dataset is divided into k subsets (folds) of approximately equal size. The model is trained on k-1 folds and tested on the remaining fold. The performance metrics are calculated based on the test results. This process is repeated k times, with each fold used as the testing set exactly once. The final performance metrics are averaged across all the folds.	CO3	L2	7M



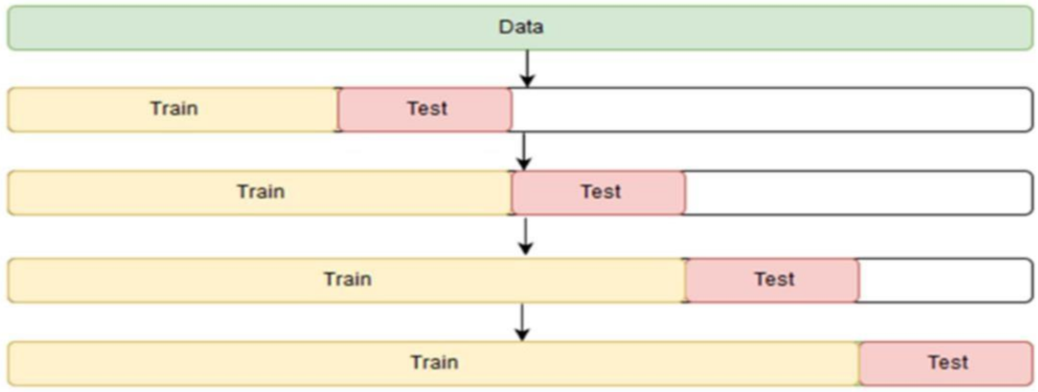
2. **Stratified K-Fold Cross-Validation**
Similar to k-fold cross-validation, but the class distribution in each fold is preserved to mitigate the impact of class imbalance.
Stratification: Before splitting, the dataset is divided into two strata based on the class labels. Each stratum contains data points belonging to a particular class.
Folding: The dataset is divided into n number of folds while maintaining the same class distribution in each fold.
3. Leave- One-Out Cross-Validation (LOOCV)
Each data point is used as a validation set, and the model is trained on all other data points. This process is repeated for each data point.
LOOCV is computationally expensive but provides a low bias estimate of the model's performance, especially for small datasets.
The performance metrics are calculated for each fold and finally averaged.



4. **Leave-p-Out Cross-Validation** (LpOCV)
Each time p data points are used as a validation set, and the model is trained on all other data points.
This process is repeated for different sets of p data points. LpOCV is computationally expensive but provides a low bias estimate of the model's performance, especially for small datasets.
The performance metrics are calculated for each fold and finally averaged
5. Repeated K-Fold Cross-Validation
The k-fold cross-validation process is repeated multiple times with different random splits of the data.
This helps in obtaining more reliable estimates of the model's performance with small datasets or when the performance is sensitive to the data split.
6. **Nested Cross-Validation**
It involves using multiple cross-validation loops to evaluate the performance of nested models, such as models with hyperparameters.
The outer loop performs model evaluation using k-fold cross-validation, while the inner loop performs hyperparameter tuning using another k-fold cross-validation.



7. Time Series Cross-Validation.
Specifically designed for time series data, where the order of data points is important.
It involves splitting the dataset into consecutive folds, ensuring that each fold contains a continuous segment of data.
This helps in assessing the model's ability to generalize to future unseen data.



b) **Describe the Ada and Gradient Boosting techniques in detail?**

CO3 | L2 | 7M

Sol:-

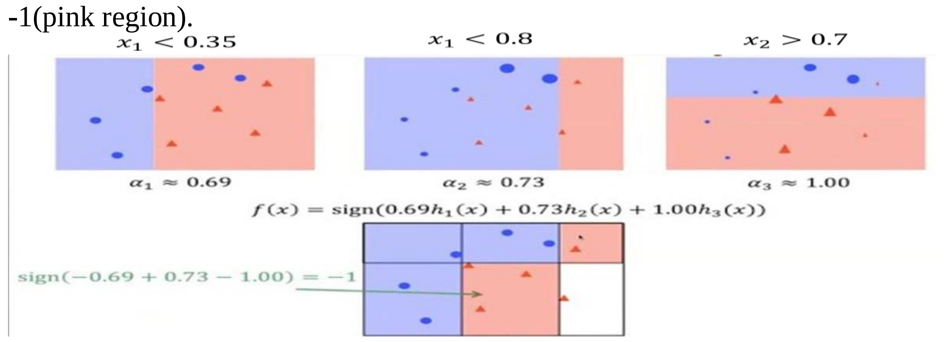
- **Boosting** (originally called hypothesis boosting) refers to any Ensemble method that can combine several weak base learners into a strong learner.
- The general idea of most boosting methods is to **train predictors sequentially**, each trying to **correct its predecessor**.
- The two popular algorithms are **AdaBoost and Gradient Boosting.**

AdaBoost

- which uses Decision Tree as a base classifier.
- Each sample is given a weight.
- These weighted samples are used to train a Decision tree stump.
- The resulting Decision Tree is used to make predictions on the training set.
- Depending on the no. of training samples correctly/incorrectly classified and their weights, the weighted error of this weak classifier is calculated.

Example- where three Decision Tree stumps are trained. The dataset consists of positive (circles) and negative(triangle) samples. Each has its own classification errors. The first two trees used the feature $x1$ at the root node. Each tree has it classification error α.

- The algorithm then increases the relative weight of misclassified training instances.
- Then it trains a second classifier, using the updated weights, and again makes predictions on the training set, calculates the weighted error of the second classifier and updates the instance weights, and so on.
- A new instance is classified based on the sum of predictions of each weak classifier multiplied by the weighted error of the respective weak leaner.
- Consider any sample and input it to each of the classifiers. The prediction is based on the sign of the following **sum**. αc indicates the **error of a particular classifier** and $hc(x)$ is the **prediction** of the corresponding classifier as +1(blue region) or -1(pink region).



Unit-IV

8 a) **Apply the K-NN classifier technique on the following dataset with k=5. Find the class of the sample in the last row.**

CO4 | L3 | 7M

Brightness	Saturation	Class
40	20	Red
50	50	Blue
60	90	Blue
10	35	Red
70	70	Blue
60	10	Red
25	80	Blue
15	40	?

Sol:-

- **$d1=\sqrt{(15-40)^2+(40-20)^2}$**
  $=\sqrt{(-25)^2+(-20)^2}=\sqrt{625+400}=\sqrt{1025}= 32.01$
- **$d2=\sqrt{(15-50)^2+(40-50)^2}$**
  $=\sqrt{(-35)^2+(-10)^2}=\sqrt{1225+100}=\sqrt{1325}= 36.4$
- **$d3=\sqrt{(15-60)^2+(40-90)^2}$**
  $=\sqrt{(-45)^2+(-50)^2}=\sqrt{2025+2500}=\sqrt{4525}= 67.26$
- **$d4=\sqrt{(15-10)^2+(40-35)^2}$**
  $=\sqrt{(5)^2+(5)^2}=\sqrt{25+25}=\sqrt{50}= 7.07$

- d5=√(15-70)²+(40-75)²
  =√(-55)²+(-25)²=√3025+625=√3650= 60.42
- d6=√(15-60)²+(40-10)²
  =√(-45)²+(-30)²=√2025+900=√2925= 54.08
- d7=√(15-25)²+(40-80)²
  =√(-10)²+(-40)²=√100+1600=√1700= 41.23
-

S.No.	Brightness	Saturation	Class	Distance(d)
1	40	20	Red	32.01
2	50	50	Blue	36.4
3	60	90	Blue	67.26
4	10	35	Red	7.071
5	70	70	Blue	60.42
6	60	10	Red	54.08
7	25	80	Blue	41.23
	15	40	?	

K=5 means top 5 samples near to the last problematic sample

S.No.	Brightness	Saturation	Class	Distance(d)
1	10	35	Red	7.071
2	40	20	Red	32.01
3	50	50	Blue	36.4
4	25	80	Blue	41.23
5	60	10	Red	54.08

the majority class within the 5 nearest neighbors to the new entry is Red. Therefore, we'll classify the new entry as Red.

b) **Apply the K-means clustering technique (Number of clusters are two) on the following dataset for two iterations.** CO4 L3 7M

X1	X2
1	2
5	1
8	1
2	1
6	1
8	2
2	2
7	1
9	1
2	3

Sol:- Randomly Choose k centers/centroid $\mu_1 = (5, 1)$ $\mu_2 = (8, 2)$

X1	X2	Distance form (µ1=(5,1)) (Cluster-1)	Distance form (µ2=(8,2))(Cluster-2)	Cluster
1	2	Sqrtof(16+1)= 4.12	Sqrtof(49+0)= 7	1
5	1	Sqrtof(0+0)= 0	Sqrtof(9+1)= 3.16	1
8	1	Sqrtof(9+0)= 3	Sqrtof(0+1)= 1	2
2	1	Sqrtof(9+0)= 3	Sqrtof(36+1)= 6.08	1
6	1	Sqrtof(1+0)= 1	Sqrtof(4+1)= 2.23	1
8	2	Sqrtof(9+1)= 3.16	Sqrtof(0+0)= 0	2
2	2	Sqrtof(9+1)= 3.16	Sqrtof(36+0)= 6	2
7	1	Sqrtof(4+0)= 2	Sqrtof(1+1)= 1.414	2
9	1	Sqrtof(16+0)= 4	Sqrtof(1+1)= 1.414	2
2	3	Sqrtof(9+4)= 3.60	Sqrtof(36+1)= 6.08	1

Find the Second level k centers/centroids

$$\mu_1 = \left(\frac{1+5+2+6+2}{5}, \frac{2+1+1+1+3}{5}\right) = \left(\frac{16}{5}, \frac{8}{5}\right) = (3.2, 1.6)$$

$$\mu_2 = \left(\frac{8+8+2+7+9}{5}, \frac{1+2+2+1+1}{5}\right) = \left(\frac{34}{5}, \frac{7}{5}\right) = (6.8, 1.4)$$

X1	X2	Distance form (µ1=(3.2,1.6)) (Cluster-1)	Distance form (µ2=(6.8,1.4))(Cluster-2)	Cluster
1	2	Sqrtof(sqr(2.2)+ sqr(-0.4))= sqrtof((4.84)+(0.16)= sqrt(5)=2.24	Sqrtof(sqr(5.8)+ sqr(-0.6))= sqrtof((33.64)+(0.36)= Sqrt(34)=5.83	1
5	1	Sqrtof(sqr(-1.8)+ sqr(-0.6))= sqrtof((3.24)+(0.36)=	Sqrtof(sqr(1.8)+ sqr(-0.4))= sqrtof((3.24)+(0.16)=	2

		sqrt(3.6)=1.90	sqrt(3.4)=1.84	
8	1	Sqrtof(sqr(-4.8)+ sqr(0.6))= sqrtof((23.04)+(0.36)= sqrt(23.4)=4.84	Sqrtof(sqr(-1.2)+ sqr(-0.4))= sqrtof((1.44)+(0.16)= sqrt(1.6)=1.26	2
2	1	Sqrtof(sqr(-1.2)+ sqr(0.6))= sqrtof((1.44)+(0.36)= sqrt(1.8)=1.34	Sqrtof(sqr(4.8)+ sqr(0.4))= sqrtof((23.04)+(0.16)= sqrt(23.2)=4.82	1
6	1	Sqrtof(sqr(-3.8)+ sqr(0.6))= sqrtof((14.44)+(0.36)= sqrt(1.8)=3.85	Sqrtof(sqr(0.8)+ sqr(0.4))= sqrtof((0.64)+(0.16)= sqrt(0.8)=0.90	2
8	2	Sqrtof(sqr(-4.8)+ sqr(-0.4))= sqrtof((23.04)+(0.16)= sqrt(23.2)=4.82	Sqrtof(sqr(-1.2)+ sqr(-0.6))= sqrtof((1.44)+(0.36)= sqrt(1.8)=1.34	2
2	2	Sqrtof(sqr(0.8)+ sqr(-0.4))= sqrtof((0.64)+(0.16)= sqrt(0.8)=0.9	Sqrtof(sqr(-4.2)+ sqr(-0.6))= sqrtof((17.64)+(0.36)= sqrt(18)=4.24	1
7	1	Sqrtof(sqr(-3.8)+ sqr(0.6))= sqrtof((14.44)+(0.36)= sqrt(14.8)=3.85	Sqrtof(sqr(-0.2)+ sqr(0.4))= sqrtof((0.04)+(0.16)= sqrt(0.2)=0.45	2
9	1	Sqrtof(sqr(-5.8)+ sqr(0.6))= sqrtof((33.64)+(0.36)= sqrt(34)=5.83	Sqrtof(sqr(-2.2)+ sqr(0.4))= sqrtof((4.84)+(0.16)= sqrt(5)=2.24	2
2	3	Sqrtof(sqr(1.2)+ sqr(-1.4))= sqrtof((1.44)+(1.96)= sqrt(3.4)=1.84	Sqrtof(sqr(-4.8)+ sqr(-1.6))= sqrtof((23.04)+(2.56)= sqrt(25.60)=5.06	1

Find the third level k centers/centroids

$$\mu_1 = \left(\frac{1+2+2+2}{4}, \frac{2+1+2+3}{4}\right) = \left(\frac{9}{4}, \frac{8}{5}\right) = (2.25, 1.6)$$

$$\mu_2 = \left(\frac{5+8+6+8+7+9}{6}, \frac{1+1+1+2+1+1}{6}\right) = \left(\frac{45}{6}, \frac{7}{6}\right) = (7.5, 1.16)$$

(OR)

9 a) **Describe Probably Approximately Correct learning theory and derive the expression for sample complexity in finite hypothesis spaces** CO4 L2 7M

Sol:-

- PAC learning model, considers the questions of how many training examples and how much computation are required in order to learn various classes of target functions within this PAC model.
- The framework for analysis is
  - c is a concept/target function that belongs to C. (PlayTennis)
  - set of training examples X, from the distribution D.
  - The learning algorithm L outputs a hypothesis h that approximates c.
  - If x is a positive c, then we will write c(x) = 1; if x is a negative c, c(x) = 0
  - Each time a set of examples X is given, decision tree algorithm L outputs a decision tree h.
  - H might be the set of all hypotheses describable by conjunctions of the attributes (ex:- age and height).
- **Hypothesis Space :** The set of all possible models (or functions) that could be used to map the relationship between input features and target labels. This space includes all the functions that the learning algorithm can choose from during training.
- A hypothesis is a specific model chosen from the hypothesis space that approximates the true relationship (target function) features (x) and predicted label (y).

sample complexity in finite hypothesis spaces

The growth in the number of required training examples with problem size, called the sample complexity of the learning problem.

general bound on the sample complexity for a very broad class of learners, called consistent learners.

A learner is consistent if it outputs hypotheses that perfectly fit the training data, whenever possible.

It is quite reasonable to ask that a learning algorithm be consistent, given that we typically prefer a hypothesis that fits the training data over one that does not.

- ➢ In other words, our learner can output h such that its true error is $\leq \varepsilon$, with a probability of $1-\delta$.

- The probability that any single hypothesis h having true error greater than ε would be consistent with one randomly drawn training example is at most $(1 - \epsilon)$.
- m randomly drawn training example is at most $(1 - \epsilon)^m$.
- Let H be the hypothesis space and $|H|$ is the number of hypotheses in H.
- The probability is atmost $|H|(1 - \epsilon)^m$
- The probability is at most $|H|e^{-\varepsilon m}$

Using the inequality $(1 - \epsilon) \leq e^{-\varepsilon}$

b) **Explain the Agglomerative clustering in detail and find the first level distance matrix for the following dataset.** CO4 L3 7M

	X1	X2
A	1	2
B	3	4
C	4	5
D	2	8
E	6	3

Sol: - Initial distance matrix

	A	B	C	D	E
A	0.00	2.89	4.24	6.08	5.1
B	2.89	0.00	1.414	4.12	3.16
C	4.24	1.414	0.00	3.6	4.12
D	6.08	4.12	3.6	0.00	6.4
E	5.1	3.16	4.12	6.4	0.00

Dist(A,B)= Sqrtof(4+4)= sqrt(8)=2.24
Dist(A,C)= Sqrtof(9+9)= sqrt(18)=4.24
Dist(C,B)= Sqrtof(1+1)= sqrt(2)=1.414
Dist(A,D)= Sqrtof(1+36)= sqrt(37)=6.08
Dist(B,D)= Sqrtof(1+16)= sqrt(17)=4.12
Dist(C,D)= Sqrtof(4+9)= sqrt(13)=3.6
Dist(A,E)= Sqrtof(25+1)= sqrt(26)=5.1
Dist(B,E)= Sqrtof(9+1)= sqrt(10)=3.16
Dist(C,E)= Sqrtof(16+1)= sqrt(17)=4.12
Dist(D,E)= Sqrtof(16+25)= sqrt(41)=6.4

	A,B	C	D	E
A,B	0.00	?	?	?
C	?	0.00	3.6	4.12
D	?	3.6	0.00	6.4
E	?	4.12	6.4	0.00

Dist((A,B)->C)=min(distAC,distBC)=min(4.24,1.414)=1.414
Dist((A,B)->D)=min(distAD,distBD)=min(6.08,4.12)=4.12
Dist((A,B)->E)=min(distAE,distBE)=min(5.1,3.16)=3.16

First level distance matrix

	A,B	C	D	E
A,B	0.00	1.414	4.12	3.16
C	1.414	0.00	3.6	4.12
D	4.12	3.6	0.00	6.4
E	3.16	4.12	6.4	0.00