Format: Question - How many times repeated

1. What are key tasks of ML [OR] what is ML [OR] Importance of ML - 4

Field of study that gives computers the capability to learn without being explicitly programmed. Machine Learning(ML) can be explained as automating and improving the learning process of computers based on their experiences without being actually programmed i.e. without any human assistance.

In Traditional Programming, Data is given as input with a program as logic but in Machine Learning the data is given as input along with the expected output, based on the data given and the expected output machine builds its own logic during the training phase which is evaluated at the testing phase. Machine learning is important because it gives enterprises a view of trends in customer behavior and operational business patterns, as well as supports the development of new products.

Advantages

- Saves Time and Resources
- Helps in decision making
- Adaptive Understanding
- Highly Precise
- Helps in forecasting/prediction

Use Cases:

- Healthcare
- Automation
- Banking and Finance
- Transportation and Traffic Prediction
- Speech and image recognition
- Recommendation System
- Virtual Assistant
- Email Spam Filtering

Key Tasks

CLASSIFICATION:

1. If we have data, say pictures of animals, we can classify them.
2. This animal is a cat, that animal is a dog and so on.
3. A computer can do the same task using a Machine Learning algorithm that's designed for the classification task.
4. In the real world, this is used for tasks like voice classification and object detection.
5. This is a supervised learning task, we give training data to teach the algorithm the classes they belong to

REGRESSION:

1. Sometimes you want to predict values.
2. What are the sales next month? And what is the salary for a job?
3. Those type of problems are regression problems.
4. The aim is to predict the value of a continuous response variable.
5. This is also a supervised learning task.

CLUSTERING:

1. Clustering is to create groups of data called clusters.
2. Observations are assigned to a group based on the algorithm.
3. This is an unsupervised learning task, clustering happens fully automatically.
4. Imagining have a bunch of documents on your computer, the computer will organize them in clusters based on their content automatically.

FEATURE SELECTION:

1. This task is important because selecting useful features would not only help to build models of higher accuracy
2. It also helps in achieving objectives related to building similar models
3. It also helps in reducing over fitting issue
4. Techniques used for feature selection: filter method, wrapper method

TESTING AND MATCHING:

1. This task related to comparing the data sets
2. Testing and matching methods: Minimum spanning trees, H point correlation

2. What is SVM [AND/OR] explain how margin is calculated and how optimal hyperplane is decided [OR] Components of SVM - 5

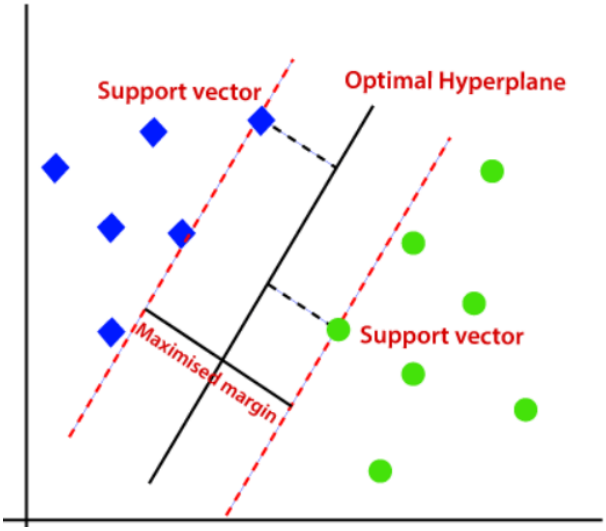
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms. It is used to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

Components of SVM:

1. Kernel: The function for converting a lower-dimensional data set to a higher-dimensional data set. A kernel aids in the search for a hyperplane in higher-dimensional space while reducing the computing cost.
2. Hyper Plane: This is the separating line between the data classes in SVM. Although, in SVR, we will describe it as a line that will assist us in predicting a continuous value or goal value.
3. Boundary line: Other than Hyper Plane, there are two lines in SVM that produce a margin. The support vectors might be within or outside the boundary lines. The two classes are separated by this line.
4. Support vectors: The data points closest to the border are listed here. The distance between the locations is little or negligible. Support vectors are locations that are outside the -tube in SVR.

Types:

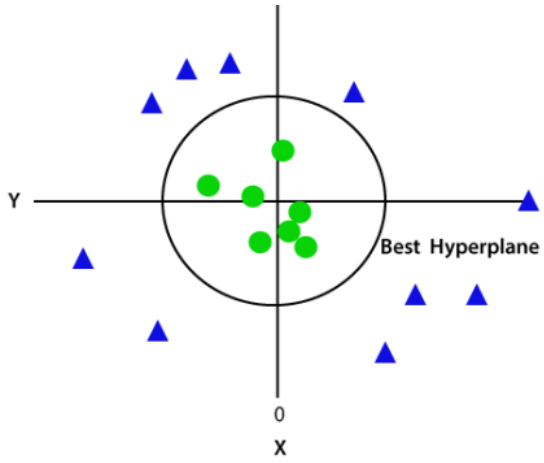
1. Linear:The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue Consider the below image:



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to

maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

2. Non Linear: If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.
   Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



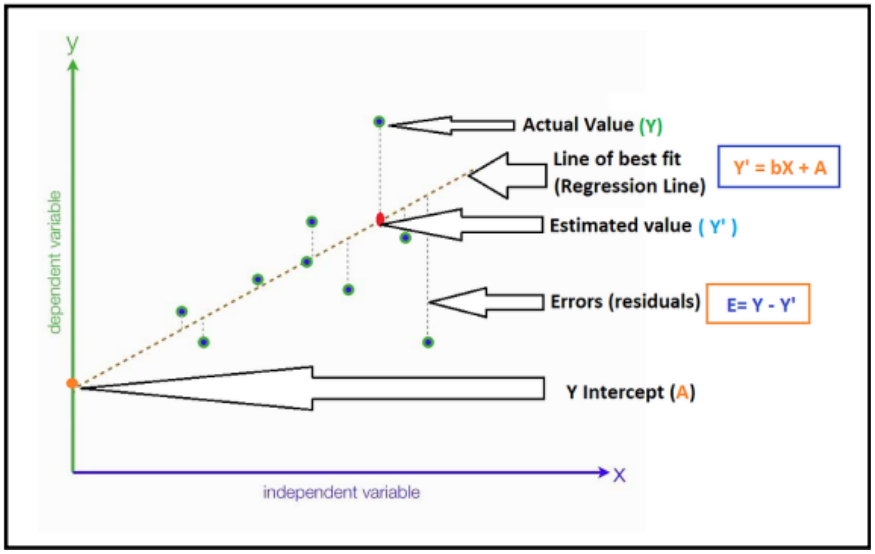
   And It can be calculated as: $\mathbf{z=x^2 +y^2}$

3. What is linear regression - 3

It is one of the most-used regression algorithms in Machine Learning. Linear regression algorithm is used if the labels are continuous, like the number of flights daily from an airport, etc. The representation of linear regression is y = b*x + c.

In the above representation, 'y' is the independent variable, whereas 'x' is the dependent variable. When you plot the linear regression, then the slope of the line that provides us the output variables is termed 'b', and 'c' is its intercept.

For Example: Medical researchers often use linear regression to understand the relationship between drug dosage and blood pressure of patients.



The Linear Regression model attempts to find the relationship between variables by finding the best fit line. Once we find the best b and c values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

Cost Function (J): By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

Gradient Descent: To update θ1 and θ2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively updating the values, reaching minimum cost.

4. Elements of Reinforcement Learning [AND/OR] What do you mean by partially observable state - 4

1) Policy: A policy can be defined as a way how an agent behaves at a given time. It maps the perceived states of the environment to the actions taken on those states. A policy is the core element of the RL as it alone can define the behavior of the agent. In some cases, it may be a simple function or a lookup table, whereas, for other cases, it may involve general computation as a search process. It could be deterministic or a stochastic policy:

For deterministic policy: $a = \pi(s)$
For stochastic policy: $\pi(a \mid s) = P[At = a \mid St = s]$

2) Reward Signal: The goal of reinforcement learning is defined by the reward signal. At each state, the environment sends an immediate signal to the learning agent, and this signal is known as a reward signal. These rewards are given according to the good and bad actions taken by the agent. The agent's main objective is to maximize the total number of rewards for good actions. The reward signal can change the policy, such as if an action selected by the agent leads to low reward, then the policy may change to select other actions in the future.

3) Value Function: The value function gives information about how good the situation and action are and how much reward an agent can expect. A reward indicates the immediate signal for each good and bad action, whereas a value function specifies the good state and action for the future. The value function depends on the reward as, without reward, there could be no value. The goal of estimating values is to achieve more rewards.

4) Model: The last element of reinforcement learning is the model, which mimics the behavior of the environment. With the help of the model, one can make inferences about how the environment will behave. Such as, if a state and an action are given, then a model can predict the next state and reward.

The model is used for planning, which means it provides a way to take a course of action by considering all future situations before actually experiencing those situations. The approaches for solving the RL problems with the help of the model are termed as the model-based approach. Comparatively, an approach without using a model is called a model-free approach.

A partially observable system is one in which the entire state of the system is not fully visible to an external sensor. In a partially observable system the observer may utilise a memory system in order to add information to the observer's understanding of the system.

5. Steps to select right ML Algorithm - 3

1) Categorize the problem: Categorize by the input: If it is a labeled data, it's a supervised learning problem. If it's unlabeled data with the purpose of finding structure, it's an unsupervised learning problem. Categorize by output: If the output of the model is a number, it's a regression problem.

2) Understand Your Data: The process of understanding the data plays a key role in the process of choosing the right algorithm for the right problem.

3) Analyze the Data: In this step, there are two important tasks which are understand data with descriptive statistics and understand data with visualization and plots.

4) Process the data: The components of data processing include pre-processing, profiling, cleansing, it often also involves pulling together data from different internal systems and external sources.

5) Transform the data: The traditional idea of transforming data from a raw state to a state suitable for modeling is where feature engineering fits in.

6) Find the available algorithms: After categorizing the problem and understand the data, the next milestone is identifying the algorithms that are applicable and practical to implement in a reasonable time.

6. For the given data determine the entropy after classification using each attribute for classification separately and find which attribute is best as decision attribute for the root by finding information gain with respect to entropy of Temperature as reference attribute. [OR] Consider following table for binary classification. Calculate the root of the decision tree using Gini index - Each Year

a)

Customer Income	Gender	Car Type	Class
High	M	Family	C1
High	M	Sports	C1
High	M	Family	C2
Low	M	Family	C2
Low	F	Family	C2
Low	F	Sports	C1
Low	F	Sports	C2
High	M	Family	C1
High	F	Family	C2
High	F	Family	C2
High	F	Sports	C2
Low	M	Sports	C2
Low	F	Family	C2
Low	M	Sports	C1

Similar to c

b)

Sr. No.	Temperature	Wind	Humidity
1	Hot	Weak	Normal
2	Hot	strong	High
3	Mild	Weak	Normal
4	Mild	Strong	High
5	Cool	Weak	Normal
6	Mild	Strong	Normal
7	Mild	Weak	High
8	Hot	Strong	Normal
9	Mild	Strong	Normal
10	Cool	Strong	Normal

1. **Temperature:**

There are three distinct values in Temperature which are Hot, Mild and Cool.

As there are three distinct values in reference attribute, Total information gain will be I(p, n, r).

Here, p = total count of Hot = 3

n = total count of Mild = 5

r = total count of cool = 2

s = p + n + r = 3 + 5 + 2 = 10

Therefore,

$I(p, n, r) = -\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s} - \frac{r}{s}\log_2\frac{r}{s}$

$= -\frac{3}{10}\log_2\frac{3}{10} - \frac{5}{10}\log_2\frac{5}{10} - \frac{2}{10}\log_2\frac{2}{10}$

I(p, n, r) = 1.486 ..... using calculator

2. **Wind:**

There are two distinct values in Wind which are Strong and Weak.

As there are two distinct values in reference attribute, Total information gain will be I(p, n).

Here, p = total count of Strong = 6

n = total count of Weak = 4

s = p + n = 6 + 4 = 10

Therefore,

$I(p, n) = -\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s}$

$= -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10}$

I(p, n) = 0.971 ..... as value of p and n are same, the answer will be 1.

3. **Humidity:**

There are two distinct values in Humidity which are High and Normal.

As there are two distinct values in reference attribute, Total information gain will be I(p, n).

Here, p = total count of High = 3

n = total count of Normal = 7

s = p + n = 3 + 7 = 10

Therefore,

$I(p, n) = -\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s}$

$= -\frac{3}{10}\log_2\frac{3}{10} - \frac{7}{10}\log_2\frac{7}{10}$

I(p, n) = 0.882 ..... as value of p and n are same, the answer will be 1.

Now we will find best root node using Temperature as reference attribute.

Here, reference attribute is Temperature.

There are three distinct values in Temperature which are Hot, Mild and Cool.

As there are three distinct values in reference attribute, Total information gain will be I(p, n, r).

Here, p = total count of Hot = 3

n = total count of Mild = 5

r = total count of cool = 2

s = p + n + r = 3 + 5 + 2 = 10

Therefore,

$I(p, n, r) = -\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s} - \frac{r}{s}\log_2\frac{r}{s}$

$= -\frac{3}{10}\log_2\frac{3}{10} - \frac{5}{10}\log_2\frac{5}{10} - \frac{2}{10}\log_2\frac{2}{10}$

I(p, n, r) = 1.486 ..... using calculator

Now we will find Information Gain, Entropy and Gain of other attributes except reference attribute

1. **Wind:**

Wind attribute have two distinct values which are weak and strong.

We will find information gain of these distinct values as following

I. **Weak =**

p_i = no of Hot values related to weak = 1

n_i = no of Mild values related to weak = 2

r_i = no of Cool values related to weak = 1

$s_i = p_i + n_i + r_i = 1 + 2 + 1 = 4$

Therefore,

$I(weak) = I(p, n, r) = -\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s} - \frac{r}{s}\log_2\frac{r}{s}$

$= -\frac{1}{4}\log_2\frac{1}{4} - \frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4}$

I(weak) = I(p, n, r) = 1.5 ..... using calculator

II. **Strong =**

p_i = no of Hot values related to strong = 2

n_i = no of Mild values related to strong = 3

r_i = no of Cool values related to strong = 1

$s_i = p_i + n_i + r_i = 2 + 3 + 1 = 6$

Therefore,

$I(weak) = I(p, n, r) = -\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s} - \frac{r}{s}\log_2\frac{r}{s}$

$= -\frac{2}{6}\log_2\frac{2}{6} - \frac{3}{6}\log_2\frac{3}{6} - \frac{1}{6}\log_2\frac{1}{6}$

I(weak) = I(p, n, r) = 1.460 ..... using calculator

Therefore,

Wind				
Distinct values from Wind	(total related values of Hot) p_i	(total related values of Mild) n_i	(total related values of Cool) r_i	Information Gain of value $I(p_i, n_i, r_i)$
Weak	1	2	1	1.5
Strong	2	3	1	1.460

Now we will find Entropy of Wind as following,

Entropy of Wind = $\sum_{i=1}^{k} \frac{p_i+n_i+r_i}{p+n+r} \times I(p_i, n_i, r_i)$

Here, p + n + r = total count of Hot, Mild and Cold from reference attribute = 10

$p_{i+n_i+r_i}$ = total count of related values from above table for distinct values in Wind attribute

$I(p_i, n_i, r_i)$ = Information gain of particular distinct value of attribute

Entropy of Wind $= \frac{p_{i+n_i+r_i\ for\ weak}}{p+n+r} \times I(p_i, n_i, r_i) + \frac{p_{i+n_i+r_i\ for\ strong}}{p+n+r} \times I(p_i, n_i, r_i)$

$= \frac{1+2+1}{10} \times 1.5 + \frac{2+3+1}{10} \times 1.460$

Entropy of wind = 1.476

Gain of wind = Entropy of Reference - Entropy of wind = 1.486 - 1.476 = **0.01**

2. Humidity :

Humidity attribute have two distinct values which are High and Normal.

We will find information gain of these distinct values as following

I. High =

p_i = no of Hot values related to High = 1

n_i = no of Mild values related to High = 2

r_i = no of Cool values related to High = 0

$s_i = p_i + n_i + r_i = 1 + 2 + 0 = 3$

Therefore,

I(High) = I(p, n, r) = $-\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s} - \frac{r}{s}\log_2\frac{r}{s}$

$= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} - \frac{0}{3}\log_2\frac{0}{3}$

I(High) = I(p, n, r) = 0.919 ..... using calculator

II. Normal =

p_i = no of Hot values related to Normal = 2

n_i = no of Mild values related to Normal = 3

r_i = no of Cool values related to Normal = 2

$s_i = p_i + n_i + r_i = 2 + 3 + 2 = 7$

Therefore,

I(Normal) = I(p, n, r) = $-\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s} - \frac{r}{s}\log_2\frac{r}{s}$

$= -\frac{2}{7}\log_2\frac{2}{7} - \frac{3}{7}\log_2\frac{3}{7} - \frac{2}{7}\log_2\frac{2}{7}$

I(weak) = I(p, n, r) = 1.557 ..... using calculator

Therefore,

Humidity				
Distinct values from Humidity	(total related values of Hot) p_i	(total related values of Mild) n_i	(total related values of Cool) r_i	Information Gain of va... $I(p_i, n_i, r_i)$
High	1	2	0	0.919
Normal	2	3	2	1.557

Now we will find Entropy by Humidity as following,

Entropy of Humidity = $\sum_{i=1}^{k} \frac{p_i+n_i+r_i}{p+n+r} \times I(p_i, n_i, r_i)$

Here, p + n + r = total count of Hot, Mild and Cold from reference attribute = 10

$p_{i+n_i+r_i}$ = total count of related values from above table for distinct values in Humidity attribute

$i(p_i, n_i, r_i)$ = Information gain of particular distinct value of attribute

Entropy of Humidity = $\frac{p_i+n_i+r_i \, for \, weak}{p+n+r} \times I(p_i, n_i, r_i) + \frac{p_i+n_i+r_i \, for \, strong}{p+n+r} \times I(p_i, n_i, r_i)$

Entropy of Humidity = $\frac{1+2+0}{10} \times 0.919 + \frac{2+3+2}{10} \times 1.557$

Entropy of Humidity = 1.366

Gain of wind = Entropy of Reference – Entropy of wind = 1.486 – 1.366 = **0.12**

Gain of wind = 0.01

Gain of humidity = 0.12

Here value of Gain(Humidity) is biggest so we will take Humidity attribute as root node.

c)

Sr. No.	Income	Defaulting	Credit Score	Location	Give Loan?
1	low	high	high	bad	no
2	low	high	high	good	no
3	high	high	high	bad	yes
4	medium	medium	high	bad	yes
5	medium	low	low	bad	no
6	medium	low	low	good	yes
7	high	low	low	good	yes
8	low	medium	high	bad	no
9	low	low	low	bad	no
10	medium	medium	low	bad	no
11	low	medium	low	good	yes
12	high	medium	high	good	yes
13	high	high	low	bad	no
14	medium	medium	high	good	yes

Solution c)

☑ **Solution :**

We will calculate Split for all attributes, i.e. Income, Defaulting, Creditscore and Location.

Income-> Split $= \frac{5}{14}$ gini (Low) $+ \frac{4}{14}$ gini (High) $+ \frac{5}{14}$ gini (Medium)

$$= \frac{5}{14}\left[1-\left(\left(\frac{1}{5}\right)^2+\left(\frac{4}{5}\right)^2\right)\right]+\frac{4}{14}\left[1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right]+\frac{5}{14}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right]=0.392$$

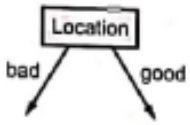
Defaulting-> Split $= \frac{4}{14}$ gini (High) $+ \frac{6}{14}$ gini (Medium) $+ \frac{4}{14}$ gini (Low) $= 0.438$

Creditscore-> Split $= \frac{7}{14}$ gini (High) $+ \frac{7}{14}$ gini (Low) $= 0.493$

Location-> Split $= \frac{8}{14}$ gini (bad) $+ \frac{6}{14}$ gini (good)

$$= \frac{5}{8}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right]+\frac{3}{8}\left[1-\left(\left(\frac{0}{3}\right)^2+\left(\frac{3}{3}\right)^2\right)\right]=0.336$$

Split value of Location is smallest, so we will select Location as root node.



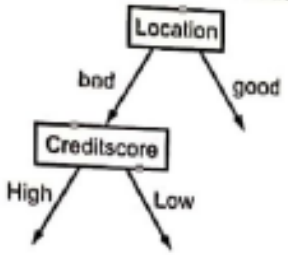
Now we will split the bad branch considering remaining attributes

Income-> Split $= \frac{3}{8}$ gini (Low) $+ \frac{2}{8}$ gini (High) $+ \frac{3}{8}$ gini (Medium) $= 0.295$

Defaulting-> Split $= \frac{3}{8}$ gini (High) $+ \frac{3}{8}$ gini (Medium) $+ \frac{2}{8}$ gini (Low) $= 0.34$

Creditscore-> Split $= \frac{4}{8}$ gini (High) $+ \frac{4}{8}$ gini (Low) $= 0.25$

Split value of Creditscore is smallest, so we will select Creditscore node below bad branch.



Now we will split the good branch considering remaining attributes

Income-> $\text{Split} = \frac{2}{6}\text{gini (Low)} + \frac{2}{6}\text{gini (High)} + \frac{2}{6}\text{gini (Medium)} = 0.295$

Defaulting-> $\text{Split} = \frac{1}{6}\text{gini (High)} + \frac{2}{6}\text{gini (Medium)} + \frac{3}{6}\text{gini (Low)} = 0$

Split value of Defaulting is smallest, so we will select Defaulting node below good branch

Since only one attribute is remaining, we can directly select Income below creditscore= High branch

For Location = bad and creditscore = High and Income = Low, Giveloan= No

For Location = bad and creditscore = High and Income = Medium, Giveloan= Yes

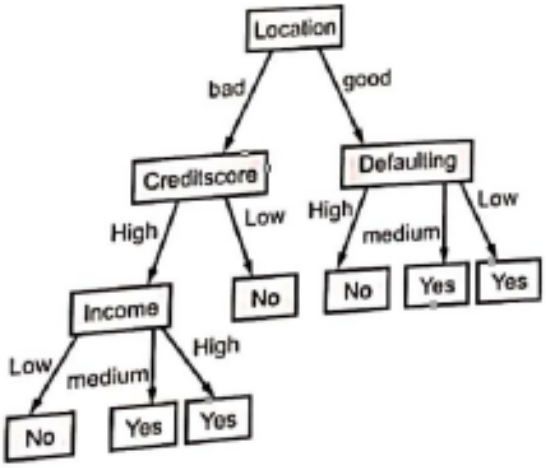
For Location = bad and creditscore = High and Income = High, Giveloan= Yes

For Location = bad and creditscore = Low, Giveloan= No

For Location = good and Defaulting = High, Giveloan= No

For Location = good and Defaulting = Low, Giveloan= Yes

For Location = good and Defaulting = Medium, Giveloan= yes



7. A) Apply K-means algorithm given data for k=3. Use Cl(2) , C2(16) and C3(38) as initial cluster centres. - Almost each year
Data: 2 4 6 3 38, 35, 14, 21, 23, 25, 30

formula for finding distance.

Distance [x, a] = $\sqrt{(x-a)^2}$

OR

Distance [(x, y), (a, b)] = $\sqrt{(x-a)^2 + (y-b)^2}$

As given data is not in pair, we will use first formula of Euclidean

Finding Distance between data points and cluster centres.

We will use following notations for calculating Distance:

D_1 = Distance from cluster C_1 centre

D_2 = Distance from cluster C_2 centre

D_3 = Distance from cluster C_3 centre

2:

$D_1(2, 2) = \sqrt{(x-a)^2} = \sqrt{(2-2)^2} = 0$

$D_2(2, 16) = \sqrt{(x-a)^2} = \sqrt{(2-16)^2} = 14$

$D_3(2, 38) = \sqrt{(x-a)^2} = \sqrt{(2-38)^2} = 34$

Here 0 is smallest distance so Data point 2 belongs to C_1

4:

$D_1(4, 2) = \sqrt{(x-a)^2} = \sqrt{(4-2)^2} = 2$

$D_2(4, 16) = \sqrt{(x-a)^2} = \sqrt{(4-16)^2} = 12$

$D_3(4, 38) = \sqrt{(x-a)^2} = \sqrt{(4-38)^2} = 34$

Here 2 is smallest distance so Data point 4 belongs to C_1

6:

$D_1(6, 2) = \sqrt{(x-a)^2} = \sqrt{(6-2)^2} = 4$

$D_2(6, 16) = \sqrt{(x-a)^2} = \sqrt{(6-16)^2} = 10$

$D_3(6, 38) = \sqrt{(x-a)^2} = \sqrt{(6-38)^2} = 32$

Here 4 is smallest distance so Data point 6 belongs to C_1

3:

$D_1(3, 2) = \sqrt{(x-a)^2} = \sqrt{(3-2)^2} = 1$

$D_2(3, 16) = \sqrt{(x-a)^2} = \sqrt{(3-16)^2} = 13$

$D_3(3, 38) = \sqrt{(x-a)^2} = \sqrt{(3-38)^2} = 35$

Here 1 is smallest distance so Data point 3 belongs to C_1

31:

$D_1(31, 2) = \sqrt{(x-a)^2} = \sqrt{(31-2)^2} = 29$

$D_2(31, 16) = \sqrt{(x-a)^2} = \sqrt{(31-16)^2} = 15$

$D_3(31, 38) = \sqrt{(x-a)^2} = \sqrt{(31-38)^2} = 7$

Here 0 is smallest distance so Data point 31 belongs to C_3

12:

$D_1(12, 2) = \sqrt{(x-a)^2} = \sqrt{(12-2)^2} = 10$

$D_2(12, 16) = \sqrt{(x-a)^2} = \sqrt{(12-16)^2} = 4$

$D_3(12, 38) = \sqrt{(x-a)^2} = \sqrt{(12-38)^2} = 26$

Here 0 is smallest distance so Data point 12 belongs to C_2

15:

$D_1(15, 2) = \sqrt{(x-a)^2} = \sqrt{(15-2)^2} = 13$

$D_2(15, 16) = \sqrt{(x-a)^2} = \sqrt{(15-16)^2} = 1$

$D_3(15, 38) = \sqrt{(x-a)^2} = \sqrt{(15-38)^2} = 23$

Here 1 is smallest distance so Data point 15 belongs to C_2

16:

$D_1(16, 2) = \sqrt{(x-a)^2} = \sqrt{(16-2)^2} = 14$

$D_2(16, 16) = \sqrt{(x-a)^2} = \sqrt{(16-16)^2} = 0$

$D_3(16, 38) = \sqrt{(x-a)^2} = \sqrt{(16-38)^2} = 22$

Here 0 is smallest distance so Data point 4 belongs to C_2

38:

$D_1(38, 2) = \sqrt{(x-a)^2} = \sqrt{(38-2)^2} = 36$

$D_2(38, 16) = \sqrt{(x-a)^2} = \sqrt{(38-16)^2} = 22$

$D_3(38, 38) = \sqrt{(x-a)^2} = \sqrt{(38-38)^2} = 0$

Here 0 is smallest distance so Data point 38 belongs to C_3

35:

$D_1(35, 2) = \sqrt{(x-a)^2} = \sqrt{(35-2)^2} = 33$

$D_2(35, 16) = \sqrt{(x-a)^2} = \sqrt{(35-16)^2} = 21$

$D_3(35, 38) = \sqrt{(x-a)^2} = \sqrt{(35-38)^2} = 3$

Here 3 is smallest distance so Data point 35 belongs to C_3

14:

$D_1(14, 2) = \sqrt{(x-a)^2} = \sqrt{(14-2)^2} = 12$

$D_2(14, 16) = \sqrt{(x-a)^2} = \sqrt{(14-16)^2} = 2$

$D_3(14, 38) = \sqrt{(x-a)^2} = \sqrt{(14-38)^2} = 24$

Here 2 is smallest distance so Data point 14 belongs to C_2

21:

$D_1(21, 2) = \sqrt{(x-a)^2} = \sqrt{(21-2)^2} = 19$

$D_2(21, 16) = \sqrt{(x-a)^2} = \sqrt{(21-16)^2} = 5$

$D_3(21, 38) = \sqrt{(x-a)^2} = \sqrt{(21-38)^2} = 17$

Here 5 is smallest distance so Data point 21 belongs to C_2

23:

$D_1(23, 2) = \sqrt{(x-a)^2} = \sqrt{(23-2)^2} = 21$

$D_2(23, 16) = \sqrt{(x-a)^2} = \sqrt{(23-16)^2} = 7$

$D_3(23, 38) = \sqrt{(x-a)^2} = \sqrt{(23-38)^2} = 15$

Here 7 is smallest distance so Data point 23 belongs to C_2

25:

$D_1(25, 2) = \sqrt{(x-a)^2} = \sqrt{(25-2)^2} = 23$

$D_2(25, 16) = \sqrt{(x-a)^2} = \sqrt{(25-16)^2} = 9$

$D_3(25, 38) = \sqrt{(x-a)^2} = \sqrt{(25-38)^2} = 13$

Here 9 is smallest distance so Data point 25 belongs to C_2

30:

$D_1(30, 2) = \sqrt{(x-a)^2} = \sqrt{(30-2)^2} = 28$

$D_2(30, 16) = \sqrt{(x-a)^2} = \sqrt{(30-16)^2} = 14$

$D_3(30, 38) = \sqrt{(x-a)^2} = \sqrt{(30-38)^2} = 8$

Here 8 is smallest distance so Data point 30 belongs to C_3

The clusters will be,

$C_1 = \{2, 4, 6, 3\}$,

$C_2 = \{12, 15, 16, 14, 21, 23, 25\}$,

$C_3 = \{31, 38, 35, 30\}$

Now we have to recalculate the centre of these clusters as following

$C_1 = \frac{2+4+6+3}{4} = \frac{15}{4} = 3.75$ (we can round off this value to 4 also)

$C_2 = \frac{12+15+16+14+21+23+25}{7} = \frac{126}{7} = 18$

$C_3 = \frac{31+38+35+30}{4} = \frac{134}{4} = 33.5$ (we can round of this value to 34 also)

Now we will again calculate distance from each data point to all new cluster centres,

2:

$D_1(2, 4) = \sqrt{(x-a)^2} = \sqrt{(2-4)^2} = 2$

$D_2(2, 18) = \sqrt{(x-a)^2} = \sqrt{(2-18)^2} = 16$

$D_3(2, 34) = \sqrt{(x-a)^2} = \sqrt{(2-34)^2} = 32$

Here 2 is smallest distance so Data point 2 belongs to C_1

4:

$D_1(4, 4) = \sqrt{(x-a)^2} = \sqrt{(4-4)^2} = 0$

$D_2(4, 18) = \sqrt{(x-a)^2} = \sqrt{(4-18)^2} = 14$

$D_3(4, 34) = \sqrt{(x-a)^2} = \sqrt{(4-34)^2} = 30$

Here 0 is smallest distance so Data point 4 belongs to C_1

6:

$D_1(6, 4) = \sqrt{(x-a)^2} = \sqrt{(6-4)^2} = 2$

$D_2(6, 18) = \sqrt{(x-a)^2} = \sqrt{(6-18)^2} = 12$

$D_3(6, 34) = \sqrt{(x-a)^2} = \sqrt{(6-34)^2} = 28$

Here 2 is smallest distance so Data point 6 belongs to C_1

3:

$D_1(3, 4) = \sqrt{(x-a)^2} = \sqrt{(3-4)^2} = 1$

$D_2(3, 18) = \sqrt{(x-a)^2} = \sqrt{(3-18)^2} = 15$

$D_3(3, 34) = \sqrt{(x-a)^2} = \sqrt{(3-34)^2} = 31$

Here 1 is smallest distance so Data point 3 belongs to C_1

31:

$D_1(31, 4) = \sqrt{(x-a)^2} = \sqrt{(31-4)^2} = 27$

$D_2(31, 18) = \sqrt{(x-a)^2} = \sqrt{(31-18)^2} = 13$

$D_3(31, 34) = \sqrt{(x-a)^2} = \sqrt{(31-34)^2} = 3$

Here 3 is smallest distance so Data point 31 belongs to C_3

12:

$D_1(12, 4) = \sqrt{(x-a)^2} = \sqrt{(12-4)^2} = 8$

$D_2(12, 18) = \sqrt{(x-a)^2} = \sqrt{(12-18)^2} = 6$

$D_3(12, 34) = \sqrt{(x-a)^2} = \sqrt{(12-34)^2} = 22$

Here 6 is smallest distance so Data point 12 belongs to C_2

15:

$D_1(15, 4) = \sqrt{(x-a)^2} = \sqrt{(15-4)^2} = 11$

$D_2(15, 18) = \sqrt{(x-a)^2} = \sqrt{(15-18)^2} = 3$

$D_3(15, 34) = \sqrt{(x-a)^2} = \sqrt{(15-34)^2} = 19$

Here 3 is smallest distance so Data point 15 belongs to C_2

16:

$D_1(16, 4) = \sqrt{(x-a)^2} = \sqrt{(16-4)^2} = 12$

$D_2(16, 18) = \sqrt{(x-a)^2} = \sqrt{(16-18)^2} = 2$

$D_3(16, 34) = \sqrt{(x-a)^2} = \sqrt{(16-34)^2} = 18$

Here 2 is smallest distance so Data point 4 belongs to C_2

38:

$D_1(38, 4) = \sqrt{(x-a)^2} = \sqrt{(38-4)^2} = 34$

$D_2(38, 18) = \sqrt{(x-a)^2} = \sqrt{(38-18)^2} = 20$

$D_3(38, 34) = \sqrt{(x-a)^2} = \sqrt{(38-34)^2} = 4$

Here 4 is smallest distance so Data point 38 belongs to C_3

35:

$D_1(35, 4) = \sqrt{(x-a)^2} = \sqrt{(35-4)^2} = 31$

$D_2(35, 18) = \sqrt{(x-a)^2} = \sqrt{(35-18)^2} = 17$

$D_3(35, 34) = \sqrt{(x-a)^2} = \sqrt{(35-34)^2} = 1$

Here 1 is smallest distance so Data point 35 belongs to C_3

14:

$D_1(14, 4) = \sqrt{(x-a)^2} = \sqrt{(14-4)^2} = 10$

$D_2(14, 18) = \sqrt{(x-a)^2} = \sqrt{(14-18)^2} = 4$

$D_3(14, 34) = \sqrt{(x-a)^2} = \sqrt{(14-34)^2} = 20$

Here 4 is smallest distance so Data point 14 belongs to C_2

21:

$D_1(21, 4) = \sqrt{(x-a)^2} = \sqrt{(21-4)^2} = 17$

$D_2(21, 18) = \sqrt{(x-a)^2} = \sqrt{(21-18)^2} = 3$

$D_3(21, 34) = \sqrt{(x-a)^2} = \sqrt{(21-34)^2} = 13$

Here 3 is smallest distance so Data point 21 belongs to C_2

23:

$D_1(23, 4) = \sqrt{(x-a)^2} = \sqrt{(23-4)^2} = 19$

$D_2(23, 18) = \sqrt{(x-a)^2} = \sqrt{(23-18)^2} = 5$

$D_3(23, 34) = \sqrt{(x-a)^2} = \sqrt{(23-34)^2} = 11$

Here 5 is smallest distance so Data point 23 belongs to C_2

25:

$D_1(25, 4) = \sqrt{(x-a)^2} = \sqrt{(25-4)^2} = 21$

$D_2(25, 18) = \sqrt{(x-a)^2} = \sqrt{(25-18)^2} = 7$

$D_3(25, 34) = \sqrt{(x-a)^2} = \sqrt{(25-34)^2} = 9$

Here 7 is smallest distance so Data point 25 belongs to C_2

30:

$D_1(30, 4) = \sqrt{(x-a)^2} = \sqrt{(30-4)^2} = 26$

$D_2(30, 18) = \sqrt{(x-a)^2} = \sqrt{(30-18)^2} = 12$

$D_3(30, 34) = \sqrt{(x-a)^2} = \sqrt{(30-34)^2} = 4$

Here 4 is smallest distance so Data point 30 belongs to C_3

The updated clusters will be,

$C_1 = \{2, 4, 6, 3\}$,

$C_2 = \{12, 15, 16, 14, 21, 23, 25\}$,

$C_3 = \{31, 38, 35, 30\}$

We can see that there is no difference between previous clusters and these updated clusters, so we will stop the process here.

Finalised clusters –

$C_1 = \{2, 4, 6, 3\}$,

$C_2 = \{12, 15, 16, 14, 21, 23, 25\}$,

$C_3 = \{31, 38, 35, 30\}$

[OR] Apply kmeans algorithm on given data for k=2. Use Cl(2, 4) & C2(6, 3) as initial cluster centres.

Data: a(2,4) b(3, 3). c(5s5), d(6, 3), e(4, 3), f(6, 6)

Number of clusters k = 2

Initial cluster centre for $C_1 = (2, 4)$, $C_2 = (6, 3)$

We will check distance between data points and all cluster centres. We will use Euclidean Distance formula for finding distance.

Distance [x, a] = $\sqrt{(x-a)^2}$

OR

Distance [(x, y), (a, b)] = $\sqrt{(x-a)^2 + (y-b)^2}$

As given data is in pair, we will use second formula of Euclidean Distance.

Finding Distance between data points and cluster centres.

We will use following notations for calculating Distance:

D_1 = Distance from cluster C_1 centre

D_2 = Distance from cluster C_2 centre

(2, 4):

$D_1[(2, 4), (2, 4)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(2-2)^2 + (4-4)^2} = 0$

$D_2[(2, 4), (6, 3)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(2-6)^2 + (4-3)^2} = 4.13$

Here 0 is smallest distance so Data point (2, 4) belongs to cluster C_1.

As Data point belongs to cluster C_1, we will recalculate the centre of cluster C_1 as following-

Using following formula for finding new centres of cluster =

Centre [(x, y), (a, b)] = $\left(\frac{x+a}{2}, \frac{y+b}{2}\right)$

Here, (x, y) = current data point

(a, b) = old centre of cluster

Updated Centre of cluster $C_1 = \left(\frac{x+a}{2}, \frac{y+b}{2}\right) = \left(\frac{2+2}{2}, \frac{4+4}{2}\right) = (2, 4)$

(3, 3):

$D_1[(3, 3), (2, 4)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(3-2)^2 + (3-4)^2} = 1.42$

$D_2[(3, 3), (6, 3)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(3-6)^2 + (3-3)^2} = 3$

Here 1.42 is smallest distance so Data point (3, 3) belongs to cluster C_1.

As Data point belongs to cluster C_1, we will recalculate the centre of cluster C_1 as following-

Updated Centre of cluster $C_1 = \left(\frac{x+a}{2}, \frac{y+b}{2}\right) = \left(\frac{3+2}{2}, \frac{3+4}{2}\right) = (2.5, 3.5)$

(5, 5):

$D_1[(5, 5), (2.5, 3.5)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(5-2.5)^2 + (5-3.5)^2} = 2.92$

$D_2[(5, 5), (6, 3)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(5-6)^2 + (5-3)^2} = 2.45$

Here 2.45 is smallest distance so Data point (5, 5) belongs to cluster C_2

As Data point belongs to cluster C_2, we will recalculate the centre of cluster C_2 as following-

Updated Centre of cluster $C_2 = \left(\frac{x+a}{2}, \frac{y+b}{2}\right) = \left(\frac{5+6}{2}, \frac{5+3}{2}\right) = (5.5, 4)$

(6, 3):

$D_1[(6, 3), (2.5, 3.5)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(6-2.5)^2 + (3-3.5)^2} = 3.54$

$D_2[(6, 3), (5.5, 4)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(6-5.5)^2 + (3-4)^2} = 1.12$

Here 1.12 is smallest distance so Data point (6, 3) belongs to cluster C_2

As Data point belongs to cluster C_2, we will recalculate the centre of cluster C_2 as following-

Updated Centre of cluster $C_2 = \left(\frac{x+a}{2}, \frac{y+b}{2}\right) = \left(\frac{6+5.5}{2}, \frac{3+4}{2}\right) = (5.75, 3.5)$

(4, 3):

$D_1[(4, 3), (2.5, 3.5)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(4-2.5)^2 + (3-3.5)^2} = 1.59$

$D_2[(4, 3), (5.75, 3.5)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(4-5.75)^2 + (3-3.5)^2} = 1.83$

Here 1.59 is smallest distance so Data point (4, 3) belongs to cluster C_1

As Data point belongs to cluster C_1, we will recalculate the centre of cluster C_1 as following-

Updated Centre of cluster $C_1 = \left(\frac{x+a}{2}, \frac{y+b}{2}\right) = \left(\frac{4+2.5}{2}, \frac{3+3.5}{2}\right) = (3.25, 3.25)$

(6, 6):

$D_1[(6, 6), (3.25, 3.25)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(6-3.25)^2 + (6-3.25)^2} = 3.89$

$D_2[(6, 6), (5.75, 3.5)] = \sqrt{(x-a)^2 + (y-b)^2} = \sqrt{(6-5.75)^2 + (6-3.5)^2} = 2.52$

Here 2.52 is smallest distance so Data point (6, 6) belongs to cluster C_1

The final clusters will be,

$C_1 = \{(2, 4), (3, 3), (4, 3), (6, 6)\}$,

$C_2 = \{(5, 5), (6, 3)\}$

8. Write Short Note on

a. **PCA:** Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components.

The PCA algorithm is based on some mathematical concepts such as:

- Variance and Covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Steps:

1. **Getting the dataset:** Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.
2. **Representing data into a structure:** Now we will represent our dataset into a structure.
3. **Standardizing the data:**In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance.

$$z = \frac{value - mean}{standard\ deviation}$$

4. **Calculating the Covariance of Z:** To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

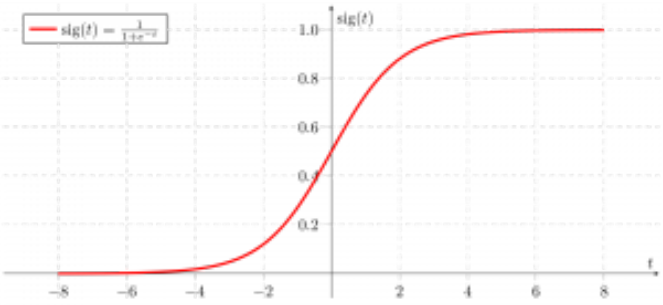
5. **Calculating the Eigen Values and Eigen Vectors:** Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

b. Linear regression -2 refers to the question 3

c. Logistic Regression: Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent Variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function[g(z)].

$$g(z) = \frac{1}{1+e^{-z}}$$



Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself. The decision for the value of the threshold value is majorly affected by the values of precision and recall. Ideally, we want both precision and recall to be 1, but this seldom is the case.

For Example: A credit card company wants to know whether transaction amount and credit score impact the probability of a given transaction being fraudulent. To understand the relationship between these two predictor variables and the probability of a transaction being fraudulent, the company can perform logistic regression.

In the case of a Precision-Recall tradeoff, we use the following arguments to decide upon the threshold:-

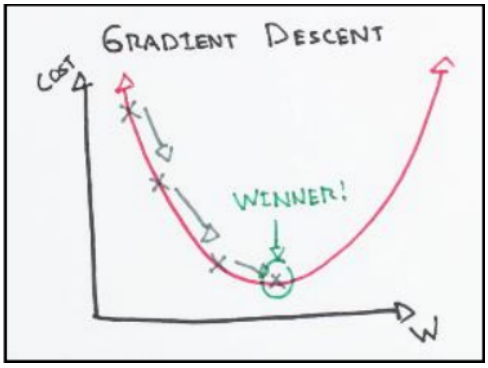
1. Low Precision/High Recall: In applications where we want to reduce the number of false negatives without necessarily reducing the number of false positives, we choose a decision value that has a low value of Precision or a high value of Recall. For example, in a cancer diagnosis application, we do not want any affected patient to be classified as not affected without giving much heed to if the patient is being wrongfully diagnosed with cancer. This is because the absence of cancer can be detected by further medical

diseases but the presence of the disease cannot be detected in an already rejected candidate.

2. High Precision/Low Recall: In applications where we want to reduce the number of false positives without necessarily reducing the number of false negatives, we choose a decision value that has a high value of Precision or a low value of Recall. For example, if we are classifying customers whether they will react positively or negatively to a personalized advertisement, we want to be absolutely sure that the customer will react positively to the advertisement because otherwise, a negative reaction can cause a loss of potential sales from the customer.

d. **SVM** Refer to the question 2

e. **Steepest Descent with steps:** Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model. Parameters refer to coefficients in Linear Regression and weights in neural networks. It also findsthe best-fit line for a given training dataset in a smaller number of iterations.



Starting at the top of the mountain, we take our first step downhill in the direction specified by the negative gradient. Next we recalculate the negative gradient (passing in the coordinates of our new point) and take another step in the direction it specifies. We continue this process iteratively until we get to the bottom of our graph, or to a point where we can no longer move downhill–a local minimum.

Cost Function (J): By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

The idea is to start with random θ1 and θ2 values and then iteratively updating the values, reaching minimum cost.

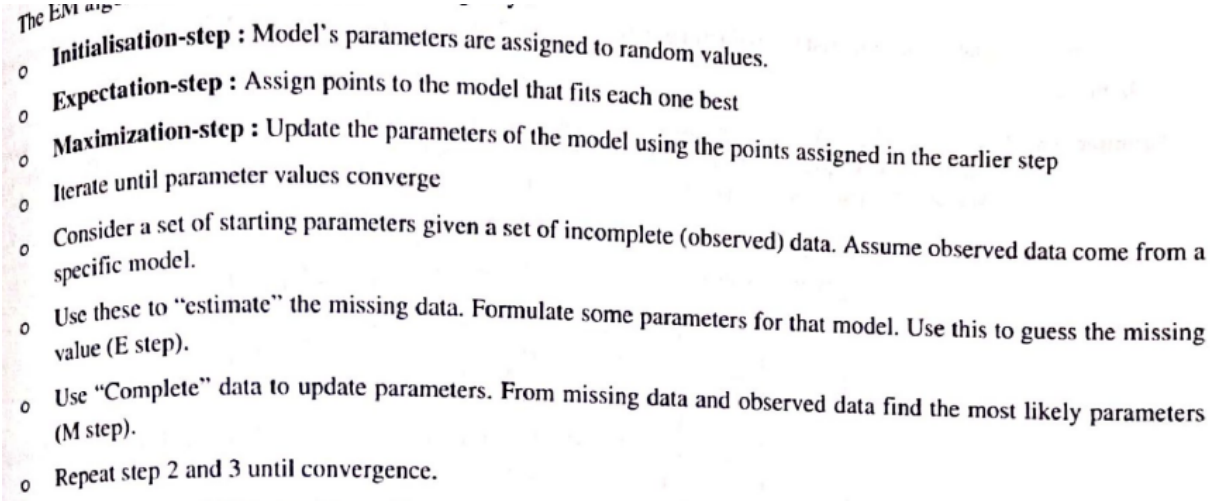
f. **ML Applications -3** refer Question 1

g. Issues in ML -3:

1. Inadequate Training Data
2. Poor quality of data
3. Non-representative training data
4. Over fitting and Under fitting
5. Monitoring and maintenance
6. Getting bad recommendations
7. Lack of skilled resources
8. Slow implementations and results

h. EM Algorithm -2:

EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely. Each iteration of the EM algorithm consists of two processes: The E-step. and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimates of the missing data from the E- step are used in lieu of the actual missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration.



The EM algo-

- **Initialisation-step :** Model's parameters are assigned to random values.
- **Expectation-step :** Assign points to the model that fits each one best
- **Maximization-step :** Update the parameters of the model using the points assigned in the earlier step
- Iterate until parameter values converge
- Consider a set of starting parameters given a set of incomplete (observed) data. Assume observed data come from a specific model.
- Use these to "estimate" the missing data. Formulate some parameters for that model. Use this to guess the missing value (E step).
- Use "Complete" data to update parameters. From missing data and observed data find the most likely parameters (M step).
- Repeat step 2 and 3 until convergence.

Applications of EM algorithm

The EM algorithm is applicable in data clustering in machine learning.
It is often used in computer vision and NLP (Natural language processing).
It is used to estimate the value of the parameter in mixed models such as the Gaussian Mixture Model and quantitative genetics.

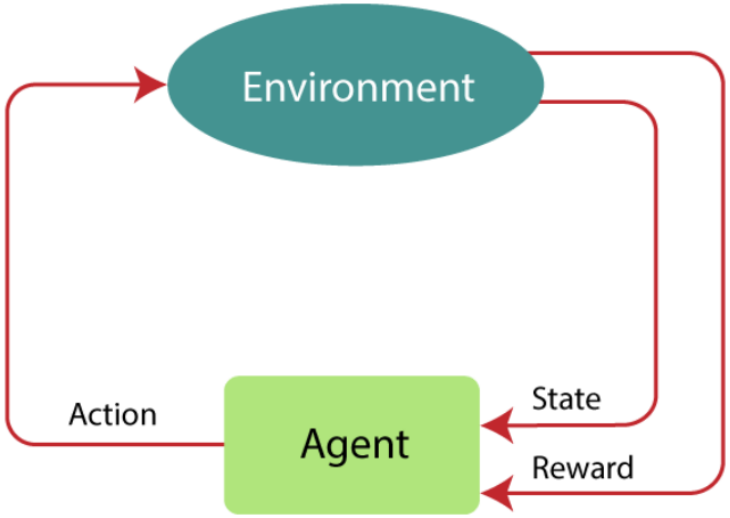
i. **Reinforcement Learning with example -2**

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty. In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike supervised learning. Since there is no labeled data, so the agent is bound to learn by its experience only. RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as game-playing, robotics, etc. The agent interacts with the environment and explores it by itself. The primary goal of an agent in reinforcement learning is to improve the performance by getting the maximum positive rewards. The agent learns with the process of hit and trial, and based on the experience, it learns to perform the task in a better way. Hence, we can say that "Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that." How a Robotic dog learns the movement of his arms is an example of Reinforcement learning. It is a core part of Artificial intelligence, and all AI agent works on the concept of reinforcement learning. Here we do not need to pre-program the agent, as it learns from its own experience without any human intervention.

Terms:

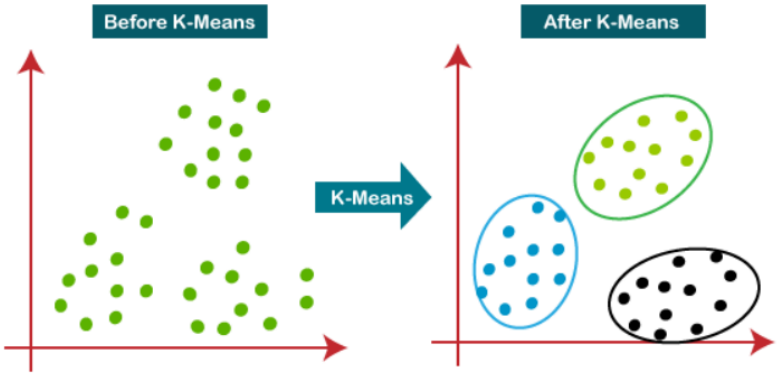
Agent(): An entity that can perceive/explore the environment and act upon it.

1. Environment(): A situation in which an agent is present or surrounded by. In RL, we assume the stochastic environment, which means it is random in nature.
2. Action(): Actions are the moves taken by an agent within the environment.
3. State(): State is a situation returned by the environment after each action taken by the agent.
4. Reward(): A feedback returned to the agent from the environment to evaluate the action of the agent.
5. Policy(): Policy is a strategy applied by the agent for the next action based on the current state.
6. Value(): It is expected long-term retuned with the discount factor and opposite to the short-term reward.
7. Q-value(): It is mostly similar to the value, but it takes one additional parameter as a current action (a).



9. Explain steps in K-Means algorithm for clustering analysis

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.



The k-means clustering algorithm mainly performs two tasks:

1. Determines the best value for K center points or centroids by an iterative process.
2. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Steps:

Step-1: Select the number K to decide the number of clusters.
Step-2: Select random K points or centroids. (It can be other from the input dataset).
Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
Step-7: The model is ready.

10.

The values of independent variable x and dependent value y are given below:

X	Y
0	2
1	3
2	5
3	4
4	6

Find the least square regression line y=ax+b. Estimate the value of y when x is 10.

$y' = aX + b$

X	Y	XY	X^2
0	2	0	0
1	3	3	1
2	5	10	4
3	4	12	9
4	6	24	16
10	20	49	30

$$a = \frac{n\Sigma XY - \Sigma X \Sigma Y}{n\Sigma x^2 - (\Sigma x)^2} = \frac{5 \times 49 - 10 \times 20}{5 \times 30 - 10^2} = 0.9$$

$$b = \frac{1}{n}(\Sigma Y - a\Sigma X) = \frac{1}{5}(20 - 0.9 \times 10) = 2.2$$

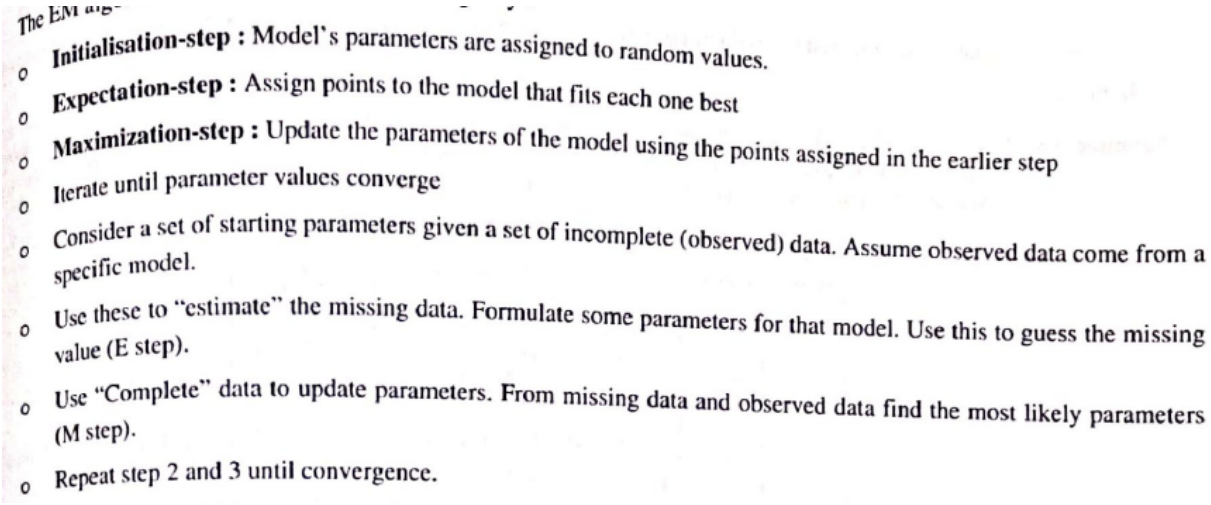
$y = 0.9x + 2.2$

$= 9 + 2.2 = 11.2$

11. What is EM Algorithm also explain what is initial hypothesis and algorithm. How Initial hypothesis converges to optimal solution?

EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely. Each iteration of the EM algorithm consists of two processes: The E-step. and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimates of the missing data from the E- step are used in lieu of the actual missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration.

Initial Hypothesis: As having one set of missing or incomplete data and another set of starting parameters, we assume that observed data or initial values of the parameters are produced from the specific model. Therefore, an entire set of incomplete observed data is provided to the system, assuming that an observed data comes from a specific model.



The EM algorithm

- **Initialisation-step :** Model's parameters are assigned to random values.
- **Expectation-step :** Assign points to the model that fits each one best
- **Maximization-step :** Update the parameters of the model using the points assigned in the earlier step
- Iterate until parameter values converge
- Consider a set of starting parameters given a set of incomplete (observed) data. Assume observed data come from a specific model.
- Use these to "estimate" the missing data. Formulate some parameters for that model. Use this to guess the missing value (E step).
- Use "Complete" data to update parameters. From missing data and observed data find the most likely parameters (M step).
- Repeat step 2 and 3 until convergence.

Applications of EM algorithm

- The EM algorithm is applicable in data clustering in machine learning.
- It is often used in computer vision and NLP (Natural language processing).
- It is used to estimate the value of the parameter in mixed models such as the Gaussian Mixture Model and quantitative genetics.

Example 5.6.12 : Suppose Coin A and B is used for tossing. Each coin is tossed 10 times. Following table shows the observation sequence of getting H and T for each round. But which coin is used for which round is not known. Then how to calculate the probability of getting H for coin A and B?

	Number of Toss									
Round number	1	2	3	4	5	6	7	8	9	10
0	H	H	H	H	H	T	T	T	T	T
1	H	H	H	H	H	H	H	H	T	T
2	H	H	H	H	H	H	H	H	H	T
3	H	H	H	H	T	T	T	T	T	T
4	H	H	H	H	H	H	H	T	T	T

Solution

Round 0 : In round 0 there are 5 H, 5 T and total tosses are 10.

Now we will calculate probability of using A and B coin as,

$$A = (P_A)^H (1-P_A)^{N-H} = (0.6)^5 (1-0.6)^{10-5} = 0.00079626$$

$$B = (P_B)^H (1-P_B)^{N-H} = (0.5)^5 (1-0.5)^{10-5} = 0.0009765$$

Now we will apply Normalization,

$$A_N = \frac{A}{A+B} = 0.45$$

$$B_N = \frac{B}{A+B} = 0.55$$

Now we will calculate probability of getting H and T for Coin A and B for round 0 as,

A_H = A_N * Number of H = 0.45* 5 = 2.25

A_T = A_N * Number of T = 0.45* 5 = 2.25

B_H = B_N * Number of H = 0.55* 5 = 2.75

A_H = B_N * Number of T = 0.55* 5 = 2.75

Round 1 : In round 1 there are 8 H, 2 T and total tosses are 10.

Now we will calculate probability of using A and B coin as,

$$A = (P_A)^H (1-P_A)^{N-H} = (0.6)^8 (1-0.6)^{10-8} = 0.002687$$

$$B = (P_B)^H (1-P_B)^{N-H} = (0.5)^8 (1-0.5)^{10-8} = 0.0009755$$

Now we will apply Normalization,

$$A_N = \frac{A}{A+B} = 0.73$$

$$B_N = \frac{B}{A+B} = 0.27$$

Now we will calculate probability of getting H and T for Coin A and B for round 1 as,

A_H = A_N * Number of H = 0.73* 8 = 5.84

A_T = A_N * Number of T = 0.73* 2 = 1.46

B_H = B_N * Number of H = 0.27* 8 = 2.16

A_H = B_N * Number of T = 0.27* 2 = 0.54

Round 2 : In round 2 there are 9 H, 1 T and total tosses are 10.

Now we will calculate probability of using A and B coin as,

$$A = (P_A)^H (1-P_A)^{N-H} = (0.6)^9 (1-0.6)^{10-9} = 0.004031$$

$$B = (P_B)^H (1-P_B)^{N-H} = (0.5)^9 (1-0.5)^{10-9} = 0.0009765$$

Now we will apply Normalization,

$$A_N = \frac{A}{A+B} = 0.80$$

$$B_N = \frac{B}{A+B} = 0.20$$

Now we will calculate probability of getting H and T for Coin A and B for round 2 as,

A_H = A_N * Number of H = 0.80 * 9 = 7.2

A_T = A_N * Number of T = 0.80 * 1 = 0.8

B_H = B_N * Number of H = 0.2 * 9 = 1.8

A_H = B_N * Number of T = 0.2 * 1 = 0.2

	Coin A		Coin B	
	A_H	A_T	B_H	B_T
0	2.25	2.25	2.75	2.75
1	5.84	1.46	2.16	0.54
2	7.2	0.8	1.8	0.2
3	1.4	2.1	2.6	3.9
4	4.55	1.95	2.45	1.05
Total	21.24	8.56	11.76	8.44

$$P_A = \frac{\Sigma H}{\Sigma H + \Sigma T} = 0.71$$

$$P_B = \frac{\Sigma H}{\Sigma H + \Sigma T} = 0.58$$

13

For a unknown tuple t =<Outllook =Sunny,Temperature =Cool, Wind= Strong> use naïve Bayes classifier to find whether the class for PlayTennis is yes or no. The dataset is given below

Outlook	Temperature	Wind	PlayTennis
Sunny	Hot	Weak	No
Sunny	Hot	Strong	No
Overcast	Hot	Weak	Yes
Rain	Mild	Weak	Yes
Rain	Cool	Weak	Yes
Rain	Cool	Strong	No
Overcast	Cool	Strong	Yes
Sunny	Mild	Weak	No
Sunny	Cool	Weak	Yes
Rain	Mild	Weak	Yes
Sunny	Mild	Strong	Yes
Overcast	Mild	Strong	Yes
Overcast	Hot	Weak	Yes
Rain	Mild	Strong	No

P (Sunny, Cool, Strong /Yes)*P(Yes)

= P (Sunny/yes)* P (Cool/yes)* P (Strong/yes)*P(Yes) = 3/5*1/5*3/5*5/14

= 0.0257

P (Sunny, Cool, Strong /No)*P(No)

= P (Sunny/no)* P (Cool/no)* P (Strong/no)*P(no) = 2/9*3/9*3/9*9/14

= 0.0158

P (Sunny, Cool, Strong /Y)*P (Y) > P (Sunny, Cool, Strong /N)*P(N)

So record < Sunny, Cool, Strong > is classified as Play Tennis = Yes

14. Explain the steps of developing Machine Learning applications

1. **Collection of Data**

   You could collect the samples from a website and extracting data.

   - From RSS feed or an API
   - From device to collect wind speed measurement
   - Publicly available data.

2. **Preparation of the input data**

   - Once you have the input data, you need to check whether it's in a useable format or not.
   - Some algorithm can accept target variables and features as string; some need them to be integers.
   - Some algorithm accepts features in a special format.

3. **Analyse the input data**

   - Looking at the data you have passed in a text editor to checkcollection and preparation of input data steps are properly working and you don't have a bunch of empty values.
   - You can also check at the data to find out if you can see any patterns or if there is anything obvious, such as a few data points greatly differ from remaining set of the data.
   - Plotting data in 1, 2 or 3 dimensions can also help.
   - Distil multiple dimensions down to 2/3 so that you can visualize the data.

4. The importance of this step is that it makes you understand that you don't have any garbage value coming in.

5. **Train the algorithm**

   - Good clean data from the first two steps is given as input to the algorithm. The algorithm extracts information or knowledge. This knowledge is mostly stored in a format that is readily useable by machine for next 2 steps.
   - In case of unsupervised learning, training step is not there because target value is not present. Complete data is used in the next step.

6. **Test the algorithm**

   - In this step the information learned in the previous step is used. When you are checking an algorithm, you will test it to find out whether it works properly or not. In supervised case, you have some known values that can be used to evaluate the algorithm.
   - In case of unsupervised, you may have to use some other matrices to evaluate the success. In either case, if you are not satisfied, you can again go back to step 4, change some things and test again.
   - Mostly problem occurs in collection or preparation of data and you will have to go back to step 1.

7. **Use it**

   In this step a real program is developed to do some task, and once again it is checked if all the previous steps worked as you expected. You might encounter some new data and have to revisit step 1-5.

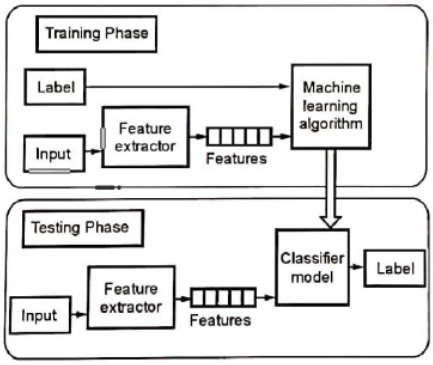
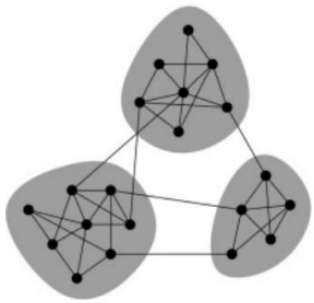


Fig. 1.6.1 : Typical example of Machine Learning Application

Topics not included in the question papers but in syllabus

1. Introduction to clustering with overview of distance metrics and major clustering approaches.
   - Graph Based Clustering: Clustering with minimal spanning tree
     A spanning tree in an undirected graph is a set of edges with no cycles that connects all nodes.A minimum spanning tree (or MST) is a spanning tree with the least total cost. Note that a shortest-path tree might not be an MST and vice-versa

     Graph based clustering is the task of clustering the graph's vertices to evaluate the edge of the chart, involving multiple edges and a considerably small number of clusters in each cluster. The graphic clustering of the input graph vertices in the cluster predicates an unsupervised learning method, which does not include the classes before Clustering. The graph clusters are built based on certain similarities in the graph. In graph classification (graph categorization), the primary objective is to graph distinct graphs into two possible classes throughout the data source.

     The data of a clustering problem can be represented as a graph where each element to be clustered is represented as a node and the distance between two elements is modeled by a certain weight on the edge linking the nodes. Thus in graph clustering, elements within a cluster are connected to each other but have no connection to elements outside that cluster. Also, some recently proposed approaches perform clustering directly on graph-based data. Some important approaches toward graph-based clusters are contiguity-based clusters and clique.

   - Model based Clustering:Density Based Clustering: DBSCAN
     Model-based clustering is a statistical approach to data clustering. The observed (multivariate) data is considered to have been created from a finite combination of component models. Each component model is a probability distribution, generally a parametric multivariate distribution.

     DBSCAN stands for density-based spatial clustering of applications with noise. It is able to find arbitrary shaped clusters and clusters with noise (i.e. outliers). The main idea behind DBSCAN is that a point belongs to a cluster if it is close to many points from that cluster.

There are two key parameters of DBSCAN:

- eps: The distance that specifies the neighborhoods. Two points are considered to be neighbors if the distance between them are less than or equal to eps.
- minPts: Minimum number of data points to define a cluster.

Based on these two parameters, points are classified as core point, border point, or outlier:

1. Core point: A point is a core point if there are at least minPts number of points (including the point itself) in its surrounding area with radius eps.
2. Border point: A point is a border point if it is reachable from a core point and there are less than minPts number of points within its surrounding area.
3. Noise point/Outlier: A point is an outlier if it is not a core point and not reachable from any core points.

2. Linear Discriminant Analysis == (PCA)

PCA: Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components.

The PCA algorithm is based on some mathematical concepts such as:

- Variance and Covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Steps:

1. **Getting the dataset:** Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

2. **Representing data into a structure:** Now we will represent our dataset into a structure.
3. **Standardizing the data:**In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance.

$$z = \frac{value - mean}{standard\ deviation}$$

4. **Calculating the Covariance of Z:** To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

5. **Calculating the Eigen Values and Eigen Vectors:** Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

Singular Valued Decomposition.

- In singular value decomposition method a matrix is decomposed into three other matrices:

$$A = USV^T$$

- Here, A represents m × n matrix. U represents m × n orthogonal matrix. S is a n × n diagonal matrix and V is a n × n orthogonal matrix.
- Matrix U has the left singular vectors as columns; S is a diagonal matrix which contains singular values; and V^T has right singular vectors as rows. In singular value decomposition original data present in a coordinate system is expanded. Here the covariance matrix is diagonal.
- To calculate singular value decomposition we need to find the eigenvalues and eigenvectors of AA^T and A^TA. The columns of V consists of eigenvectors of A^T. The columns of U consists of the eigenvectors of AA^T.

- The square roots of eigenvalues from AA^T or A^TA represents the singular values in S.
- The singular values are arranged in descending order and stored as the diagonal entries of the S matrix. The singular values are always real numbers. If the matrix A is a real matrix, then U and V are also real.

Example 1 : Find SVD for $A = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$

First we will calculate $A^TA = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$

Now we will calculate eigen vectors V_1 and V_2 using the method that we have seen in PCA

$$V_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \qquad V_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Next we will calculate AV_1 and AV_2

$$AV_1 = \begin{bmatrix} 2\sqrt{2} \\ 0 \end{bmatrix} \qquad AV_2 = \begin{bmatrix} 0 \\ \sqrt{2} \end{bmatrix}$$

Next we will calculate U_1 and U_2

$$U_1 = \frac{AV_1}{AV_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad U_2 = \frac{AV_2}{|AV_2|} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

SVD is written as,

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

3. **SVM as constrained optimization problem,Quadratic Programming**

- A hyperplane is formally defined by the following notation as,

$$F(x) = w^t x + b$$

In above equation, w represents the weight vector and b represents the bias.

- By scaling the values of w and b we can represent the optimal hyperplane in many ways. As a matter of convention among all the possible notations of the hyperplane the one selected is

$$|w^t x + b| = 1$$

- Here x represents the training records closest to the hyperplane. In general the training records that are closest to the hyperplane are called as support vectors. This notation is called as the canonical hyperplane.
- The distance between a point x and a hyperplane (w, b) is given by the result of geometry as follows,

$$\text{Distance} = \frac{|w^t x + b|}{\|w\|}$$

- In general, the numerator is equal to one for the canonical hyperplane and distance to the support vector is given as,

$$\text{Distance}_{sv} = \frac{|w^t x + b|}{\|w\|} = \frac{1}{\|w\|}$$

- Margin is twice the distance to nearest samples

$$M = \frac{2}{\|w\|}$$

- Ultimately, the task of maximizing M is same as compared to the task of minimizing a function L (w) subject to some conditions. The conditions used to model the requirements for correct classification of all training samples x_i by the hyperplane are formally stated as,

$$\min_{w, b} L(w) = \frac{1}{2}\|w\|^2 \text{ subject to } y_i (w^t x + b) \geq 1 \text{ for all i.}$$

Where y_i represents the labels of training

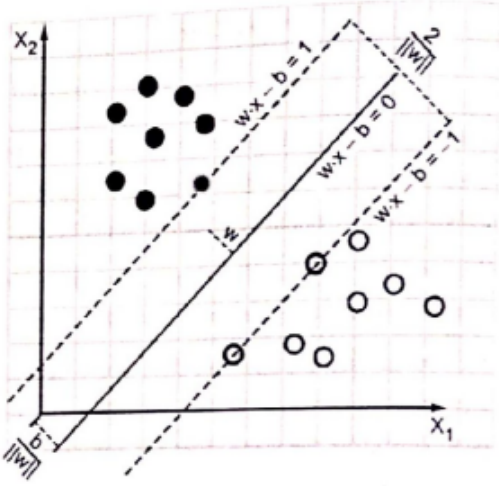


Fig. 5.5.3 : Solution to find maximum margin

This is a problem of Lagrangian optimization that can be solved using Lagrange multiplier to calculate weight vector 'w' and the bias 'b' of the optimal hyperplane.

Let's assume that we have 2 classes of 2 dimensional data to separate. Let's also assume that each class consist of only one point

These points are

$$X_1 = A_1 = (3, 3)$$

$$X_2 = B_1 = (6, 6)$$

Find the hyper plane that separates theses 2 classes

$$f(w) = \frac{1}{2}\|w\|^2$$

The constraints are,

$$c_1(w, b) = y_i\,|\,wx_i + b\,|\,-1 \geq 0$$

$$c_1(w, b) = 1\,|\,wx_1 + b\,|\,-1 \geq 0$$

$$c_2(w, b) = -1\,|\,wx_2 + b\,|\,-1 \geq 0$$

Next, we put equation into form of Lagrangian

$$L(w, b, m) = f(w) - m_1 c_1(w, b) - m_2 c_2(w, b)$$

$$= \frac{1}{2}\|w\|^2 - m_1((wx_1 + b) - 1) - m_2(-(wx_2 + b) - 1)$$

$$= \frac{1}{2}\|w\|^2 - m_1((wx_1 + b) - 1) + m_2((wx_2 + b) + 1)$$

We solve for the gradient of Lagrangian

$$\nabla L(w, b, m) = \nabla f(w) - m_1 \nabla c_1(w, b) + m_2 \nabla c_2(w, b) = 0$$

$$\frac{\partial}{\partial w} L(w, b, m) = w - m_1 x_1 + m_2 x_2 = 0 \quad \ldots(5.5.1)$$

$$\frac{\partial}{\partial b} L(w, b, m) = -m_1 + m_2 = 0 \quad \ldots(5.5.2)$$

$$\frac{\partial}{\partial \lambda_1} L(w, b, m) = (wx_1 + b) - 1 = 0 \quad \ldots(5.5.3)$$

$$\frac{\partial}{\partial \lambda_2} L(w, b, m) = (wx_2 + b) + 1 = 0 \quad \ldots(5.5.4)$$

Equating Equation (5.5.3) and (5.5.4), we get

$$(wx_1 + b) - 1 = (wx_2 + b) + 1$$

$$(wx_1) - 1 = (wx_2) + 1$$

$$(wx_1) - (wx_2) = 2$$

$$w(x_1 - x_2) = 2$$

w is divided into parts as,

$$w = (w_1, w_2)$$

$$W(x_1 - x_2) = 2$$

$$(w_1, w_2)[(3, 3) - (6, 6)] = 2$$

$$(w_1, w_2)[(-3, -3)] = 2$$

$$-3w_1 - 3w_2 = 2$$

$$w_1 = -(0.67 + w_2) \quad \ldots(5.5.5)$$

Adding values to Equation (5.5.1) and combining with Equation (5.5.2)

$(w_1, w_2) - m_1 (1, 1) + m_2 (2, 2) = 0$

From Equation (5.5.2)

$$m_1 = m_2$$

$(w_1, w_2) - m_1 (3, 3) + m\lambda_1 (6, 6) = 0$

$$(w_1, w_2) + m_1 (3, 3) = 0 \qquad \ldots(5.5.6)$$

$$w_1 + 3 m_1 = 0 \qquad \ldots(5.5.7)$$

$$w_2 + 3 m_1 = 0$$

Equating these we get,

$$w_1 = w_2$$

Putting this in Equation (5.5.5)

$$w_1 = w_2 = -0.34$$

Putting this in either Equation (5.5.6) or Equation (5.5.7) will give

$$m_1 = m_2 = 0.11$$

And finally, using this in Equation (5.5.3) and Equation (5.5.4)

$$b = 1 - (wx_1) \text{ or } = -1 - (w x_2)$$

$$= 1 - ((-0.34, -0.34), (3, 3)) \text{ or } = 1 - ((-0.34, -0.34), (6, 6)) = 3.04$$

classification

EXAMPLE ON EXPLAINING HOW KERNEL CAN BE USED FOR CLASSIFYING NON-LINEARLY SEPARABLE DATA:

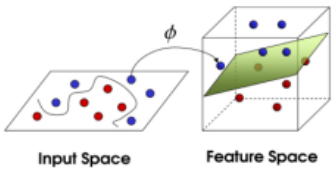
1. To predict if a dog is a particular breed, we load in millions of dog information/properties like type, height, skin colour, body hair length etc.

In ML language, these properties are referred to as 'features'.

A singie entry of these list of features is a data instance while the collection of everything is the Training Data which forms the basis of your prediction

I.e. if you know the skin colour, body hair length, height and so on of a particular dog, then you can predict the breed it will probably belong to.

In support vector machines, it looks somewhat like shown in figure 4.5 which separates the blue balls from red.



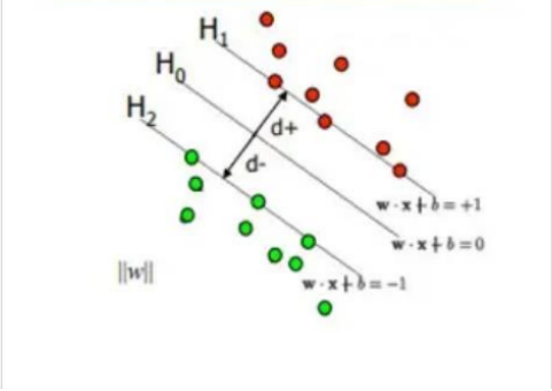
Therefore the hyperplane of a two dimensional space below is a one dimensional line dividing the red and blue dots.

From the example above of trying to predict the breed of a particular dog, it goes like this:

Data (all breeds of dog) → Features (skin colour, hair etc.) → Learning algorithm

Basics of Kernel trick.

A Kernel Trick is a simple method where a Non Linear data is projected onto a higher dimension space so as to make it easier to classify the data where it could be linearly divided by a plane. This is mathematically achieved by Lagrangian formula using Lagrangian multipliers.



- The Lagrangian is

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

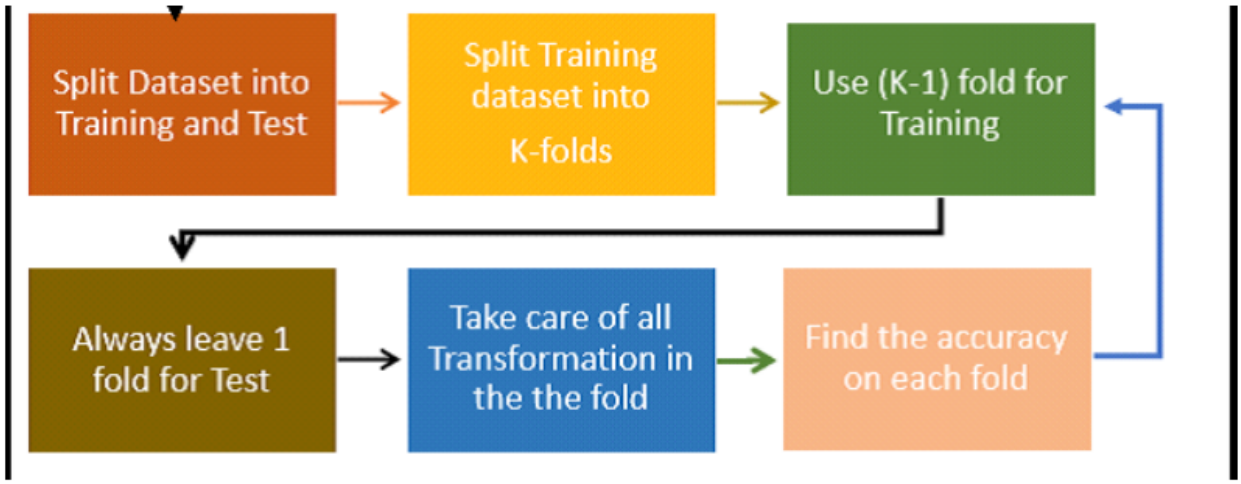
4. Understanding Ensembles,

K-fold cross validation

The k-fold cross-validation method is widely used for calculating how well a machine learning model performs on a validation dataset.

The general process of k-fold cross-validation for evaluating a model's performance is:

1. The whole dataset is randomly split into independent k-folds without replacement.
2. k-1 folds are used for the model training and one fold is used for performance evaluation.
3. This procedure is repeated k times (iterations) so that we obtain k number of performance estimates (e.g.MSE) for each iteration.
4. Then we get the mean of k number of performance estimates (e.g. MSE).
5. Once a k-value is determined, we can use it to assess various models on the dataset



Issues in K fold Cross Validation

1. To keep the training set large, we allow validation sets that are small.
2. The training sets overlap considerably, namely, any two training sets share K − 2 parts.
3. K is typically 10 or 30. As K increases, the percentage of training instances increases and we get more robust estimators, but the validation set becomes smaller. Furthermore, there is the cost of training the classifier K times, which increases as K is increased.

Boosting
It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

Stumping,

XGBoost
XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now. Please see the chart below for the evolution of tree-based algorithms over the years.

These are some key members of XGBoost models, each plays an important role.

1. RMSE: It is the square root of mean squared error (MSE).
2. MAE: It is an absolute sum of actual and predicted differences, but it lacks mathematically, that's why it is rarely used, as compared to other metrics.

XGBoost is a powerful approach for building supervised regression models. The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners. The objective function contains loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values. The most common loss functions in XGBoost for regression problems is reg:linear, and that for binary classification is reg:logistics. Ensemble learning involves training and combining individual models (known as base learners) to get a single prediction, and XGBoost is one of the ensemble learning methods.

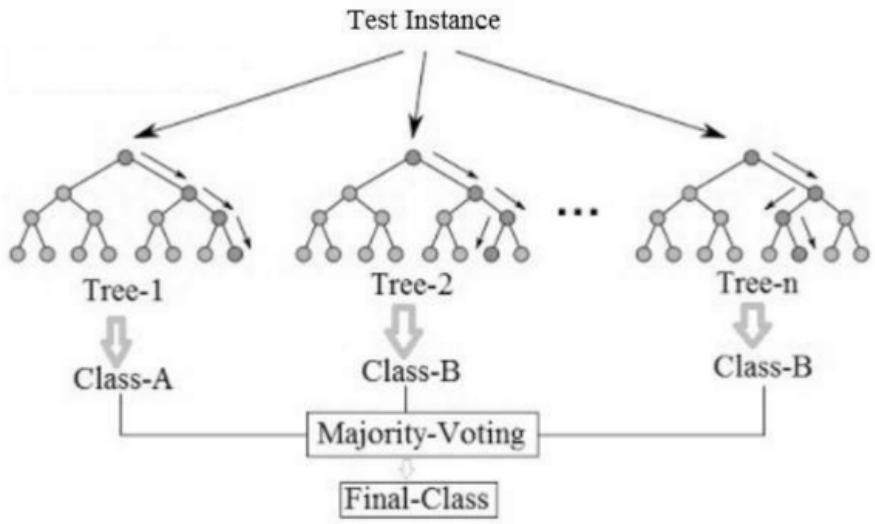
Bagging,
Bootstrap Aggregating, also known as bagging, creates a different training subset from sample training data with replacement & the final output is based on majority voting. It decreases the variance and helps to avoid over fitting. It is usually applied to decision tree methods. Bagging is a special case of the model averaging approach. The base-learners are trained with L number of random samples. A learning algorithm is an unstable algorithm if small changes in the training set causes a large difference in the generated learner.

Bagging uses bootstrap to generate L training sets, trains L base- learners using an unstable learning procedure and then during testing, takes an average. Bagging can be used both for classification and regression. In the case of regression, to be more robust, one can take the median instead of the average when combining predictions.

Random Forest,
A random forest is an ensemble learning method where multiple decision trees are constructed and then they are merged to get a more accurate prediction.

Algorithm:
1. The random forests algorithm generates many classification trees. Each tree is generated as follows:
   (a) If the number of examples in the training set is N, take a sample of N examples at random - but with replacement, from the original data. This sample will be the training set for generating the tree.
   (b) If there are M input variables, a number m is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the generation of the various trees in the forest.
   (c) Each tree is grown to the largest extent possible.

2. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree “votes” for that class. The forest chooses the classification



5. Performance Metrics:

Confusion Matrix,

A confusion matrix is used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. A confusion matrix is a table that categorizes predictions according to whether they match the actual value.

Two-class datasets

For a two-class dataset, a confusion matrix is a table with two rows and two columns that reports the number of false positives, false negatives, true positives, and true negatives. Assume that a classifier is applied to a two-class test dataset for which the true values are known. Let TP denote the number of true positives in the predicted values, TN the number of true negatives, etc. Then the confusion matrix of the predicted values can be represented as follows:

	Actual condition is true	Actual condition is false
Predicted condition is true	TP	FP
Predicted condition is false	FN	FN

Table 5.1: Confusion matrix for two-class dataset

Multiclass datasets

Confusion matrices can be constructed for multiclass datasets also.

Example If a classification system has been trained to distinguish between cats, dogs and rabbits, a confusion matrix will summarize the results of testing the algorithm for further inspection. Assuming a sample of 27 animals - 8 cats, 6 dogs, and 13 rabbits, the resulting confusion matrix could look like the table below: This confusion matrix shows that, for example, of the 8 actual cats, the system predicted that

	Actual “cat”	Actual “dog”	Actual “rabbit”
Predicted “cat”	5	2	0
Predicted “dog”	3	3	2
Predicted “ rabbit”	0	1	11

three were dogs, and of the six dogs, it predicted that one was a rabbit and two were cats.

[Kappa Statistics],

1. Precision and Recall
   In machine learning, precision and recall are two measures used to assess the quality of results produced by a binary classifier. They are formally defined as follows.
   Definitions
   Let a binary classifier classify a collection of test data. Let
   TP = Number of true positives
   TN = Number of true negatives
   FP = Number of false positives
   FN = Number of false negatives

   The *precision* P is defined as

   $$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

   The *recall* R is defined as

   $$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

2. **ROC:** The acronym ROC stands for Receiver Operating Characteristic, a terminology coming from signal detection theory. The ROC curve was first developed by electrical engineers and radar engineers during World War II for detecting enemy objects in battlefields. They are now increasingly used in machine learning and data mining research.

   $$\begin{aligned}
   \mathrm{TPR} &= \text{True Positive Rate} \\
   &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \\
   &= \text{Fraction of positive examples correctly classified} \\
   &= \text{Sensitivity} \\
   \mathrm{FPR} &= \text{False Positive Rate} \\
   &= \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \\
   &= \text{Fraction of negative examples incorrectly classified} \\
   &= 1 - \text{Specificity}
   \end{aligned}$$

   ROC Curve: ROC curve
   In the case of certain classification algorithms, the classifier may depend on a parameter. Different values of the parameter will give different classifiers and these in turn give different values to TPR and FPR. The ROC curve is the curve obtained by plotting in the ROC space the points (TPR , FPR) obtained by assigning all possible values to the parameter in the classifier.

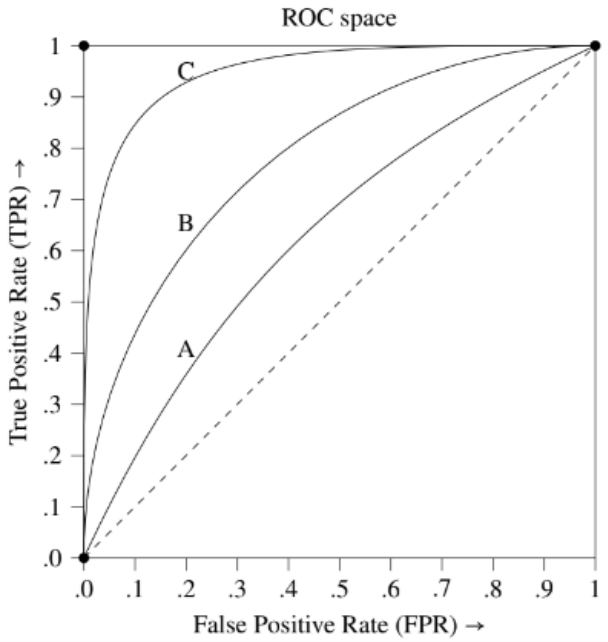


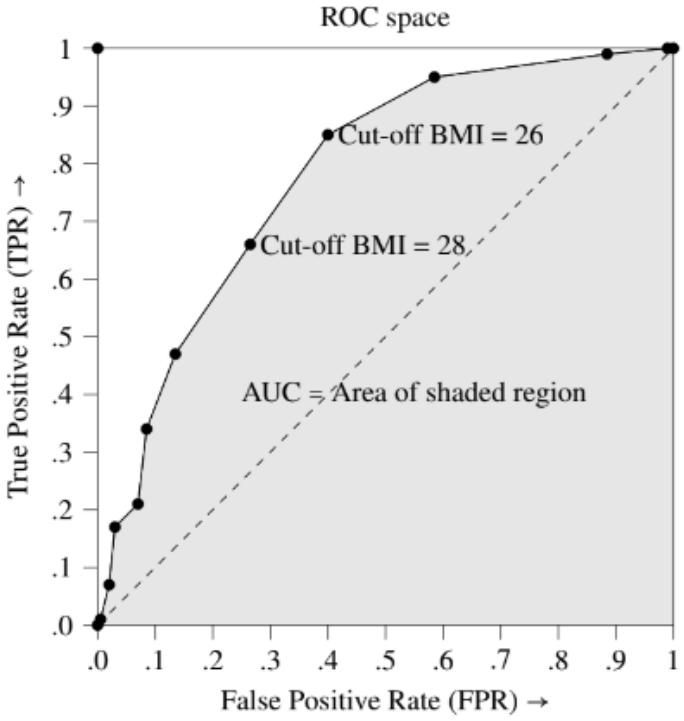
Figure 5.3: ROC curves of three different classifiers A, B, C

The closer the ROC curve is to the top left corner (0, 1) of the ROC space, the better the accuracy of the classifier. Among the three classifiers A, B, C with ROC curves as shown in Figure, the classifier C is closest to the top left corner of the ROC space. Hence, among the three, it gives the best accuracy in predictions.

Example:

Cut-off value of BMI	Breast cancer		Normal persons		TPR	FPR
	TP	FN	FP	TN		
18	100	0	200	0	1.00	1.000
20	100	0	198	2	1.00	0.990
22	99	1	177	23	0.99	0.885
24	95	5	117	83	0.95	0.585
26	85	15	80	120	0.85	0.400
28	66	34	53	147	0.66	0.265
30	47	53	27	173	0.47	0.135
32	34	66	17	183	0.34	0.085
34	21	79	14	186	0.21	0.070
36	17	83	6	194	0.17	0.030
38	7	93	4	196	0.07	0.020
40	1	99	1	199	0.01	0.005

Table 5.3: Data on breast cancer for various values of BMI

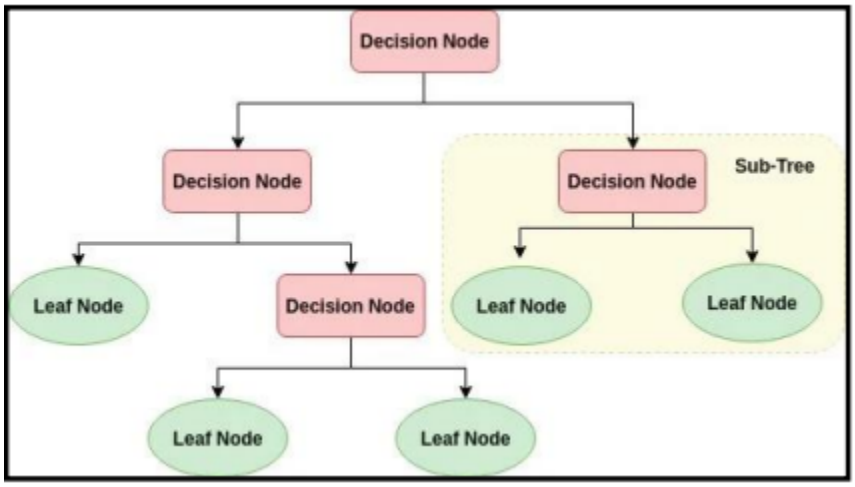


1. Accuracy = $\frac{TP + TN}{TP + TN + FP + FN}$
2. Error rate = 1− Accuracy
3. Sensitivity = $\frac{TP}{TP + FN}$
4. Specificity = $\frac{TN}{TN + FP}$
5. F-measure = $\frac{2 \times TP}{2 \times TP + FP + FN}$

6. Classification and Regression Trees (CART)

In the decision tree, the nodes are split into subnodes on the basis of a threshold value of an attribute.

Classification And Regression Trees (CART) algorithm is a classification algorithm for building a decision tree based on Gini's impurity index as splitting criterion. CART is a binary tree build by splitting node into two child nodes repeatedly. The root node is taken as the training set and is split into two by considering the best attribute and threshold value. Further, the subsets are also split using the same logic. This continues till the last pure sub-set is found in the tree or the maximum number of leaves possible in that growing tree. This is also known as Tree Pruning.



CART is an umbrella word that refers to the following types of decision trees:

Classification Trees: When the target variable is continuous, the tree is used to find the "class" into which the target variable is most likely to fall.

Regression trees: These are used to forecast the value of a continuous variable.

The main elements of CART are:

- Rules for splitting data at a node based on the value of one variable
- Stopping rules for deciding when a branch is terminal and can be split no more
- A prediction for the target variable in each terminal node

Regression Tree

A regression problem is the problem of determining a relation between one or more independent variables and an output variable which is a real continuous variable and then using the relation to predict the values of the dependent variables. Regression problems are in general related to prediction of numerical values of variables. Trees can also be used to make such predictions. A tree used for making predictions of numerical variables is called a prediction tree or a regression tree.

$$m_c = \frac{1}{n_c} \sum_{i \epsilon C} y_i$$

$$S_T = \sum_{c \epsilon \text{leaves}(T)} \sum_{i \epsilon C} (y_i - m_c)^2$$

Algorithm

Step 1. Start with a single node containing all data points. Calculate mc and ST .

Step 2. If all the points in the node have the same value for all the independent variables, stop.

Step 3. Otherwise, search over all binary splits of all variables for the one which will reduce ST as much as possible.

(a) If the largest decrease in ST would be less than some threshold δ, or one of the resulting nodes would contain less than q points, stop and if c is a node where we have stopped, then assign the value mc to the node.

(b) Otherwise, take that split, creating two new nodes.

Step 4. In each new node, go back to Step 1.