RJS First Grade College

Koramangala, Bengaluru – 560034

Department of Computer Science and Applications

MACHINE LEARNING

Unit – 1 Introduction to Machine Learning

Two Marks Questions

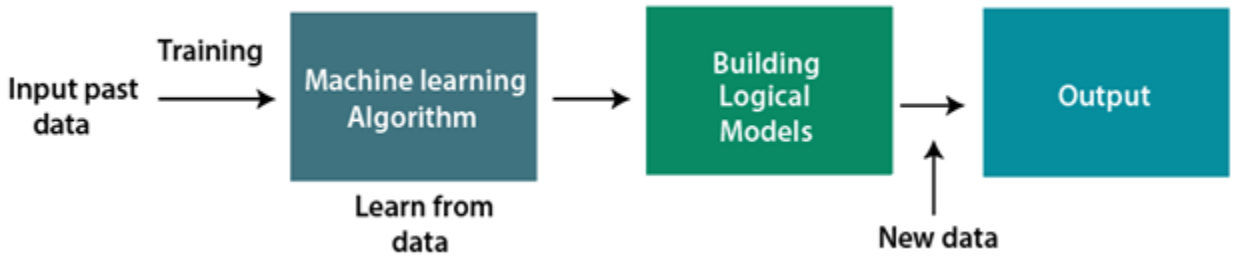
1) What is Machine Learning?

Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. A machine has the ability to learn if it can improve its performance by gaining more data.

2) How does Machine Learning work?

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.



3) What are the features of Machine Learning?

- Machine learning uses data to detect various patterns in a given dataset.

- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

4) What is the classification of Machine Learning?

Machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

5) Explain designing learning systems?

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E''.

6) What are the steps involved in Designing Learning Systems?

Designing a learning system is a five-step process. The steps are,

- Choosing the Training Experience
- Choosing the Target Function
- Choose a Representation for the Target Function
- Choosing a Function Approximation Algorithm
- The Final Design

7) What is concept of hypothesis?

A hypothesis is a specific, testable prediction. It describes in concrete terms what you expect will happen in a certain circumstance.

8) Explain Hypothesis in Machine Learning?

A hypothesis in machine learning:

1) Covers the available evidence: the training dataset.

2) Is falsifiable (kind-of): a test harness is devised beforehand and used to estimate performance and compare it to a baseline model to see if is skillful or not.

3) Can be used in new situations: make predictions on new data.

9) What is the purpose of Hypothesis?

A hypothesis is used in an experiment to define the relationship between two variables. The purpose of a hypothesis is to find the answer to a question. A formalized hypothesis will force us to think about what results we should look for in an experiment.

The first variable is called the independent variable. This is the part of the experiment that can be changed and tested.

The independent variable happens first and can be considered the cause of any changes in the outcome. The outcome is called the dependent variable.

10) Define Precision?

Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly).

Precision = True Positive/True Positive + False Positive
Precision = TP/TP+FP

11) Define Recall?

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

Recall = True Positive/True Positive + False Negative

1. Recall = TP/TP+FN

Five marks and Ten Marks Questions:

12) What are the perspective Issues in ML?

Common issues in Machine Learning

1. Inadequate Training Data

The major issue that comes while using machine learning algorithms is the lack of quality as well as quantity of data.
Data quality can be affected by some factors as follows:

- Noisy Data- It is responsible for an inaccurate prediction that affects the decision as well as accuracy in classification tasks.
- Incorrect data- It is also responsible for faulty programming and results obtained in machine learning models. Hence, incorrect data may affect the accuracy of the results also.
- Generalizing of output data- Sometimes, it is also found that generalizing output data becomes complex, which results in comparatively poor future actions.

2. Poor quality of data

Noisy data, incomplete data, inaccurate data, and unclean data lead to less accuracy in classification and low-quality results.

3. Non-representative training data

If we are using non-representative training data in the model, it results in less accurate predictions. A machine learning model is said to be ideal if it predicts well for generalized cases and provides accurate decisions. If there is less training data, then there will be a sampling noise in the model, called the non-representative training set.

13) Explain Over fitting and methods to reduce over fitting?

Over fitting

The main reason behind over fitting is using non-linear methods used in machine learning algorithms as they build non-realistic data models.

Methods to reduce over fitting:

- Increase training data in a dataset.
- Reduce model complexity by simplifying the model by selecting one with fewer parameters
- Ridge Regularization and Lasso Regularization
- Early stopping during the training phase
- Reduce the noise
- Reduce the number of attributes in training data.
- Constraining the model.

14) Explain Under fitting and methods to reduce under fitting?

Under fitting occurs when our model is too simple to understand the base structure of the data, just like an undersized pant. This generally happens when we have limited data into the data set, and we try to build a linear model with non-linear data.

Methods to reduce Under fitting:

- Increase model complexity
- Remove noise from the data
- Trained on increased and better features
- Reduce the constraints
- Increase the number of epochs to get better results.

15) What are the Basic Issues in ML?

1.Monitoring and maintenance

Generalized output data is mandatory for any machine learning model; hence, regular monitoring and maintenance become compulsory for the same. Different results for different actions require data change; hence editing of codes as well as resources for monitoring them also become necessary.

2. Getting bad recommendations

A machine learning model operates under a specific context which results in bad recommendations and concept drift in the model.It generally occurs when new data is introduced or interpretation of data changes.

3.Lack of skilled resources

The absence of skilled resources in the form of manpower is also an issue. Hence, we need manpower having in-depth knowledge of mathematics, science, and technologies for developing and managing scientific substances for machine learning.

4.Customer Segmentation

Customer segmentation is also an important issue while developing a machine learning algorithm. Hence, an algorithm is necessary to recognize the customer behavior and trigger a relevant recommendation for the user based on past experience.

16)Define Version Space in Machine Learning?

A version space is a hierarchical representation of knowledge that enables you to keep track of all the useful information supplied by a sequence of learning examples without remembering any of the examples.

The version space method is a concept learning process accomplished by managing multiple models within a version space.

17)Define Clustering?

Clustering is a data mining technique which groups unlabeled data based on their similarities or differences. Clustering algorithms are used to process raw, unclassified data objects into groups represented by structures or patterns in the information. Clustering algorithms can be categorized into a few types, specifically exclusive, overlapping, hierarchical, and probabilistic.

18) What are the characteristics of Version Space?

Version Space Characteristics

Tentative heuristics are represented using version spaces.

A version space represents all the alternative plausible descriptions of a heuristic.A plausible description is one that is applicable to all known positive examples and no known negative example.

A version space description consists of two complementary trees:

1. One that contains nodes connected to overly general models, and
2. One that contains nodes connected to overly specific models.

19) What are the basic guidelines in Version Space?

1) Generalization and 2) Specialization.

Each node is connected to a model.

Nodes in the generalization tree are connected to a model that matches everything in its subtree.

Nodes in the specialization tree are connected to a model that matches only one thing in its subtree.

Links between nodes and their models denote

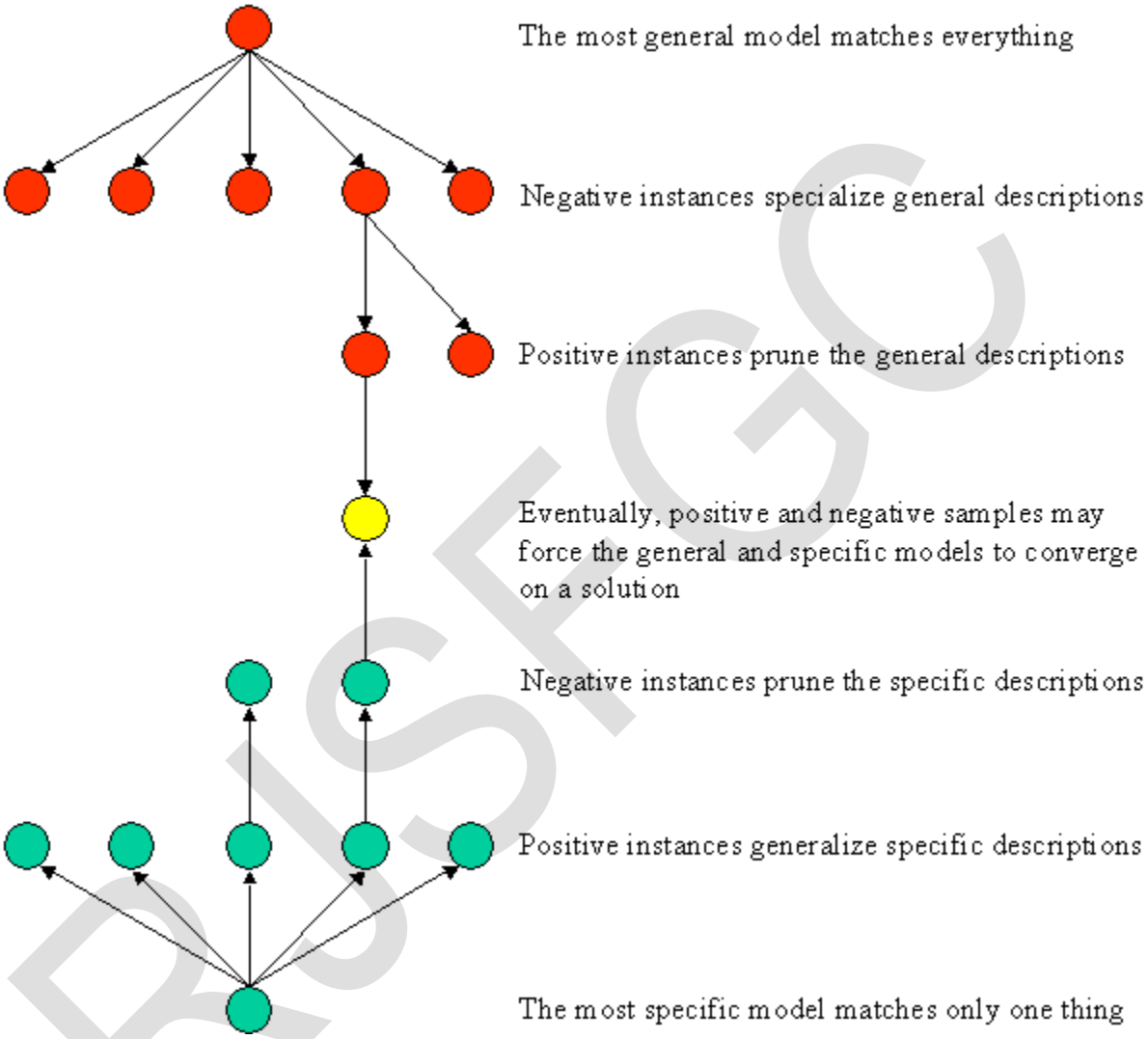
- generalization relations in a generalization tree, and
- specialization relations in a specialization tree.

20) What Are the Differences Between Sensitivity and Specificity?

While Sensitivity measure is used to determine the proportion of actual positive cases, which got predicted correctly, Specificity measure is used to determine the proportion of actual negative cases, which got predicted correctly.

TEN MARK QUESTIONS:

21) Explain the diagrammatic representation for Version Space?



The specialization tree is colored red, and the generalization tree is colored green.

22) What is Inductive bias?

Every machine learning model requires some type of architecture design and possibly some initial assumptions about the data we want to analyze.

Generally, every building block and every belief that we make about the data is a form of inductive bias.

A strong inductive bias can lead our model to converge to the global optimum.

On the other hand, a weak inductive bias can cause the model to find only the local optima and be greatly affected by random changes in the initial states.

We can categorize inductive biases into two different groups called relational and non-relational.

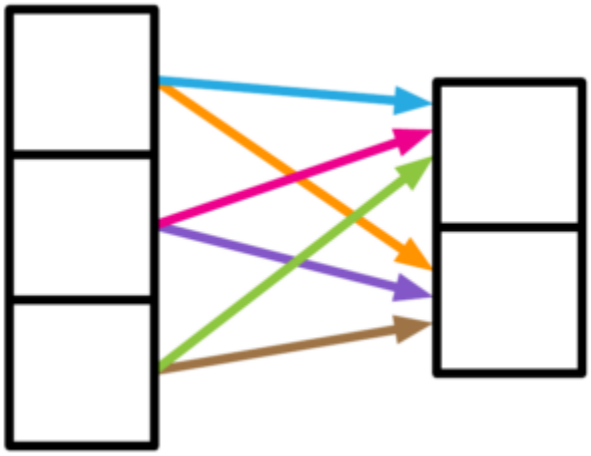
The former represents the relationship between entities in the network, while the latter is a set of techniques that further constrain the learning algorithm.

23) Explain Relational and Non-Relational Inductive Bias?

Relational inductive biases define the structure of the relationships between different entities or parts in our model. These relations can be arbitrary, sequential, local, and so on.

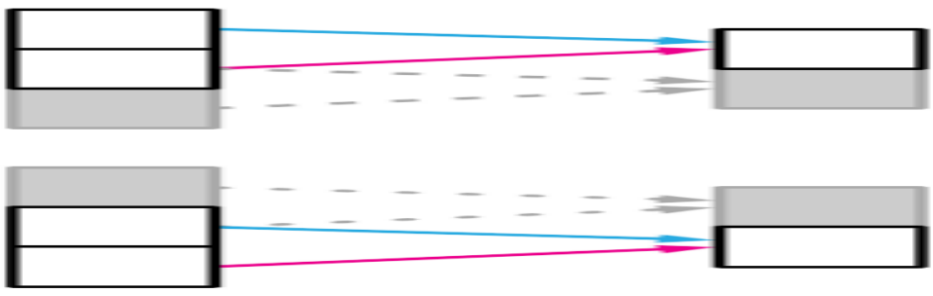
1. Weak Relation

Sometimes the relationship between the neural units is weak, meaning that they're somewhat independent of each other. The choice of including a fully connected layer in the net can represent this kind of relationship:



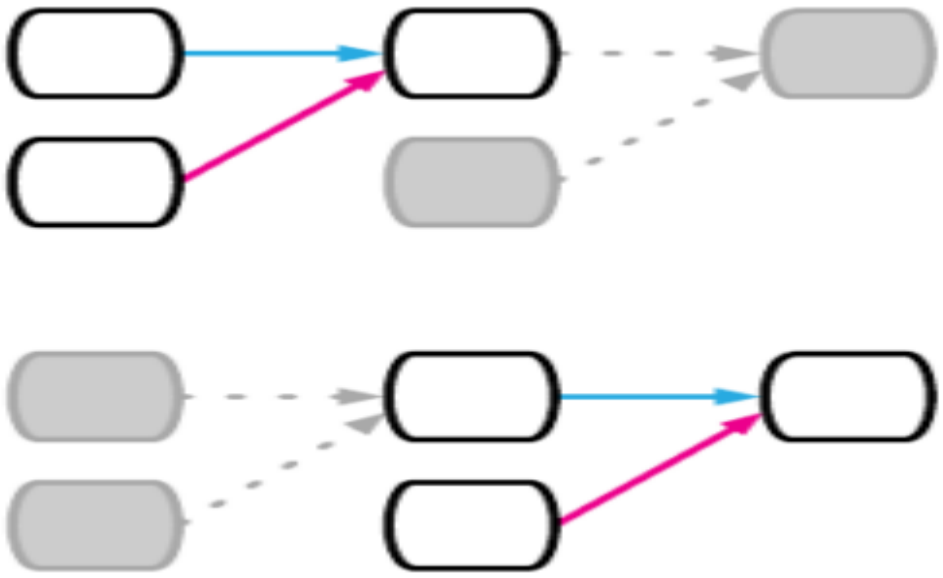
2. Locality

In order to process an image, we start by capturing the local information. One way to do that is the use of a convolutional layer. It can capture the local relationship between the pixels of an image. Then, as we go deeper in the model, the local feature extractors help to extract the global features:



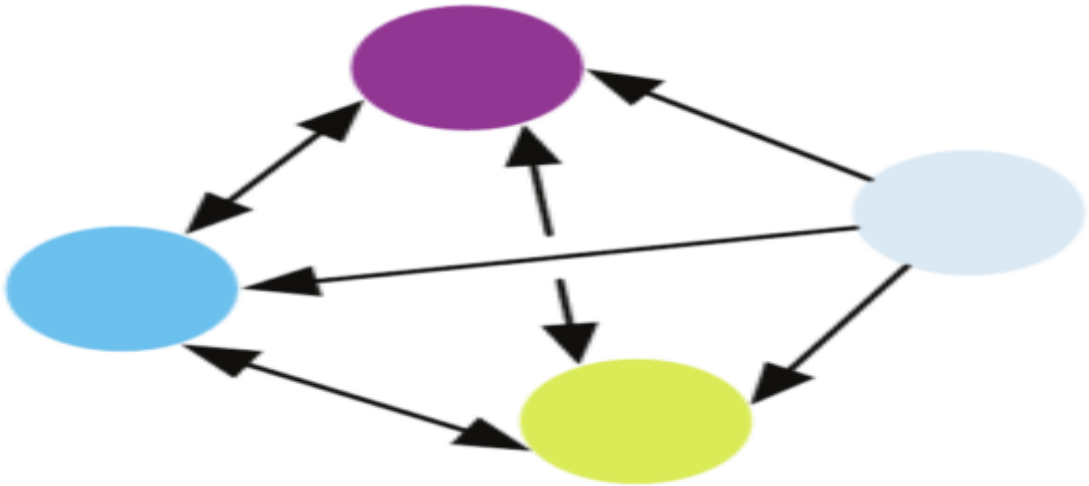
3. Sequential Relation

Sometimes our data has a sequential characteristic. For instance, time series and sentences consist of sequential elements that appear one after another. To model this pattern, we can introduce a recurrent layer to our network:



4. Arbitrary Relation

To solve problems related to a group of things or people, it might be more informative to see them as a graph. The graph structure imposes arbitrary relationships between the entities, which is ideal when there's no clear sequential or local relation in the model:



Non-Relational Inductive Biases in Deep Learning

Other than relational inductive biases, there are also some concepts that impose additional constraints on our model.

1. Non-linear Activation Functions

Non-linear activation functions allow the model to capture the non-linearity hidden in the data. Without them, a deep neural network wouldn't be able to work better than a single-layer network. The reason is that the combination of several linear layers would still be a linear layer.

2. Dropout

Dropout is a regularization technique that helps the network avoid memorizing the data by forcing random subsets of the network to each learn the data pattern. As a result, the obtained model, in the end, is able to generalize better and avoid overfitting.

3. Weight Decay

Weight decay is another regularization method that puts constraints on the model's weights. There are several versions of weight decay, but the common ones are and regularization techniques. Weight decay doesn't let the weights grow very large, which prevents the model from overfitting.

4. Normalization

Normalization techniques can help our model in several ways, such as making the training faster and regularizing. But most importantly, it reduces the change in the distribution of the net's activations which is called internal co-variate shift. There are different normalization techniques such as batch normalization, instance normalization, and layer normalization.

5. Data Augmentation

We can think of data augmentation as another regularization method. What it imposes on the model depends on its algorithm. For instance, adding noise or word substitution in sentences are two types of data augmentation. They assume that the addition of the noise or word substitution should not change the category of a sequence of words in a classification task.

6. Optimization Algorithm

The optimization algorithm has a key role in the model's outcome we want to learn. For example, different versions of the gradient descent algorithm can lead to different optima. Subsequently, the resulting models will have other generalization properties. Moreover, each optimization algorithm has its own parameters that can greatly influence the convergence and optimality of the model.

24) What is Precision?

Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly).

Precision = True Positive/True Positive + False Positive
Precision = TP/TP+FP

- TP- True Positive
- FP- False Positive

- The precision of a machine learning model will be low when the value of;

TP+FP (denominator) > TP (Numerator)

- The precision of the machine learning model will be high when Value of;

TP (Numerator) > TP+FP (denominator)

- Precision helps us to visualize the reliability of the machine learning model in classifying the model as positive.

25) What is Recall?

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples.

The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

Recall = True Positive/True Positive + False Negative
Recall = TP/TP+FN

- TP- True Positive
- FN- False Negative
- Recall of a machine learning model will be low when the value of;
  TP+FN (denominator) > TP (Numerator)

- Recall of machine learning model will be high when Value of;
  TP (Numerator) > TP+FN (denominator)

Unlike Precision, Recall is independent of the number of negative sample classifications. Further, if the model classifies all positive samples as positive, then Recall will be 1.

26) Define Sensitivity?

Sensitivity is a measure of the proportion of actual positive cases which got predicted as positive (or true positive). Sensitivity is also termed as Recall.
This implies that there will be another proportion of actual positive cases which would get predicted incorrectly as negative (and, thus, could also be termed as the false negative).
Sensitivity can also be represented in form of True Positive Rate (TPR). The sum of sensitivity (true positive rate) and false negative rate would be 1.
Higher the true positive rate, better the model is in identifying the positive cases in correct manner.
For example-the model used for predicting whether a person is suffering from the disease. Sensitivity or true positive rate is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease.
In other words, the person who is unhealthy (positive) actually got predicted as unhealthy.

sensitivity or true positive rate can be calculated as the following:

Sensitivity = (True Positive)/(True Positive + False Negative)

The following is the details in relation to True Positive and False Negative used in the above equation.

- True Positive:
- Persons predicted as suffering from the disease (or unhealthy) are actually suffering from the disease (unhealthy); In other words, the true positive represents the number of persons who are unhealthy and are predicted as unhealthy.
- False Negative: Persons who are actually suffering from the disease (or unhealthy) are actually predicted to be not suffering from the disease (healthy).
- In other words, the false negative represents the number of persons who are unhealthy and got predicted as healthy.
- Ideally, we would seek the model to have low false negatives as it might prove to be life-threatening or business threatening.

27) Define Specificity?

Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative).

This implies that there will be another proportion of actual negative, which got predicted as positive and could be termed as false positives. This proportion could also be called a false positive rate.
The sum of specificity and false positive rate would always be 1.
Let's try and understand this with the model used for predicting whether a person is suffering from the disease.
Specificity is a measure of the proportion of people not suffering from the disease who got predicted correctly as the ones who are not suffering from the disease.
In other words, the person who is healthy actually got predicted as healthy is specificity.

Mathematically, specificity can be calculated as the following:

Specificity = (True Negative)/(True Negative + False Positive)

The following is the details in relation to True Negative and False Positive used in the above equation.

- **True Negative =** Persons predicted as not suffering from the disease (or healthy) are actually found to be not suffering from the disease (healthy); In other words, the true negative represents the number of persons who are healthy and are predicted as healthy.
- **False Positive =** Persons predicted as suffering from the disease (or unhealthy) are actually found to be not suffering from the disease (healthy). In other words, the false positive represents the number of persons who are healthy and got predicted as unhealthy.

The higher value of specificity would mean higher value of true negative and lower false positive rate. The lower value of specificity would mean lower value of true negative and higher value of false positive.

28) What is the AUC – ROC Curve?

AUC – ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve, and AUC represents the degree or measure of separability.

It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s.

By analogy, Higher the AUC, better the model is at distinguishing between patients with the disease and no disease.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

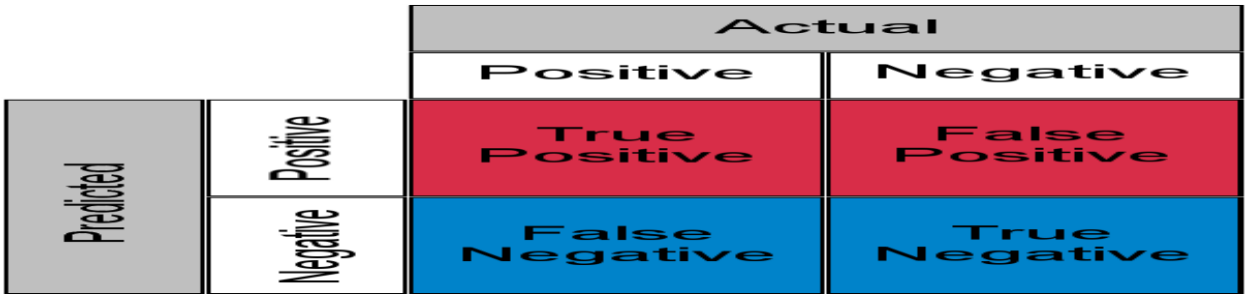
Consider a two-class prediction problem, in which the outcomes are labeled either as positive (*p*) or negative (*n*).

There are four possible outcomes from a binary classifier. If the outcome from a prediction is *p* and the actual value is also *p*, then it is called a *true positive* (TP);

however, if the actual value is *n* then it is said to be a *false positive* (FP).

Conversely, a *true negative* (TN) has occurred when both the prediction outcome and the actual value are *n*, and *false-negative* (FN) is when the prediction outcome is *n* while the actual value is *p*.

- True Positive: Actual Positive and Predicted as Positive
- True Negative: Actual Negative and Predicted as Negative
- False Positive(Type I Error): Actual Negative but predicted as Positive
- False Negative(Type II Error): Actual Positive but predicted as Negative



TPR(True Positive Rate) / Recall / Sensitivity

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

Specificity

$$\text{Specificity} = \frac{TN}{TN + FP}$$

FPR

$$\text{FPR} = 1 - \text{Specificity} = \frac{FP}{TN + FP}$$

UNIT- 2 Supervised Learning

TWO Marks Questions

1) Define Decision Trees?

A decision tree is a supervised learning technique that has a pre-defined target variable and is most often used in classification problems. This tree can be applied to either categorical or continuous input & output variables. The training process resembles a flow chart, with each internal (non-leaf) node a test of an attribute, each branch is the outcome of that test, and each leaf (terminal) node contains a class label. The uppermost node in the tree is called the root node.

2) What are the types of Decision Trees?

The two main families of decision trees are defined by function:

- **Classification trees** – Used to predict the class to which the data sample belongs.
- **Regression tree**

- **Boosted trees** – Used to train instances that were previously incorrectly modeled. For example, AdaBoost. This works for both regression and classification problems.

3) What are the practical issues in learning decision trees

- determining how deeply to grow the decision tree,
- handling continuous attributes,
- choosing an appropriate attribute selection measure,
- handling training data with missing attribute values,
- handling attributes with differing costs, and
- improving computational efficiency.

4) Define Instant Based Learning in ML?

It is called instance-based because it builds the hypotheses from the training instances. It is also known as **memory-based learning** or **lazy-learning.** The time complexity of this algorithm depends upon the size of training data. The worst-case time complexity of this algorithm is **O (n)**, where n is the number of training instances.

5)What are the advantages and disadvantages of Instance-based learning

Advantages:

1. Instead of estimating for the entire instance set, local approximations can be made to the target function.
2. This algorithm can adapt to new data easily, one which is collected as we go .

Disadvantages:

1. Classification costs are high
2. Large amount of memory required to store the data, and each query involves starting the identification of a local model from scratch.

6)What are the algorithms used in Instance Based Learning?

Some of the instance-based learning algorithms are :

1. K Nearest Neighbor (KNN)
2. Self-Organizing Map (SOM)
3. Learning Vector Quantization (LVQ)
4. Locally Weighted Learning (LWL)

7) Define Neural Network?

Neural networks reflect the behavior of the human brain, allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning.

8) Define perceptions in ML?

The objects that do the calculations are **perceptions**. They adjust themselves to minimize the loss function until the model is very accurate. For example, we can get handwriting analysis to be 99% accurate.

In the case of recognizing handwriting or facial recognition, the brain very quickly makes some decisions.

9) Define Multi layer networks?

A multi-layer neural network contains more than one layer of artificial neurons or nodes. They differ widely in design. the vast majority of networks used today have a multi-layer model.

10) What are the types of Multilayer Networks?

There are two main types of multilayer networks, multiplex networks and interconnected networks .In multiplex networks, interlayer edges can only connect nodes that represent the same actor in different layers.

In interconnected networks, interlayer edges can connect between different actors, and therefore different layers typically represent different entities.

11) Define Back Propogation?

The Backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent. The weights that minimize the error function is then considered to be a solution to the learning problem.

FIVE Mark Questions:

12) Explain the Procedures involved in ID3 algorithm?

ID3 Steps

1. Calculate the Information Gain of each feature.
2. Considering that all rows don't belong to the same class, split the dataset **S** into subsets using the feature for which the Information Gain is maximum.
3. Make a decision tree node using the feature with the maximum Information gain.
4. If all rows belong to the same class, make the current node as a leaf node with the class as its label.
5. Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

13) Explain K-Nearest Algorithm?

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.

It is also called a **lazy learner algorithm**

14)What are the advantages and disadvantages of K-NN algorithm?

Advantages of KNN Algorithm:

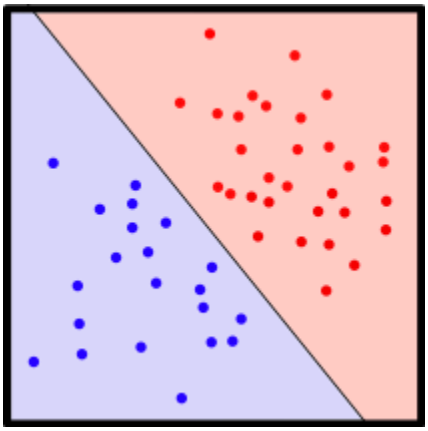
- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

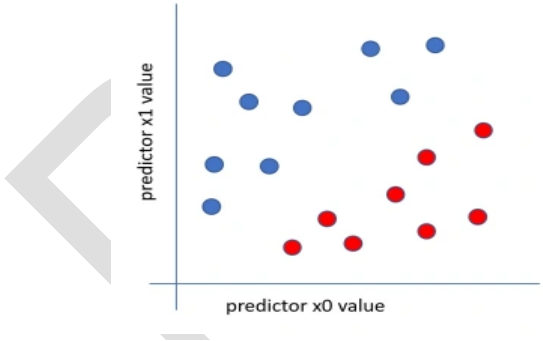
Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

15) Explain the concept of handling the data?

Linear separability is a property of two sets of points. This is most easily visualized in two dimensions (the Euclidean plane) by thinking of one set of points as being colored blue and the other set of points as being colored red. These two sets are *linearly separable* if there exists at least one line in the plane with all of the blue points on one side of the line and all the red points on the other side.

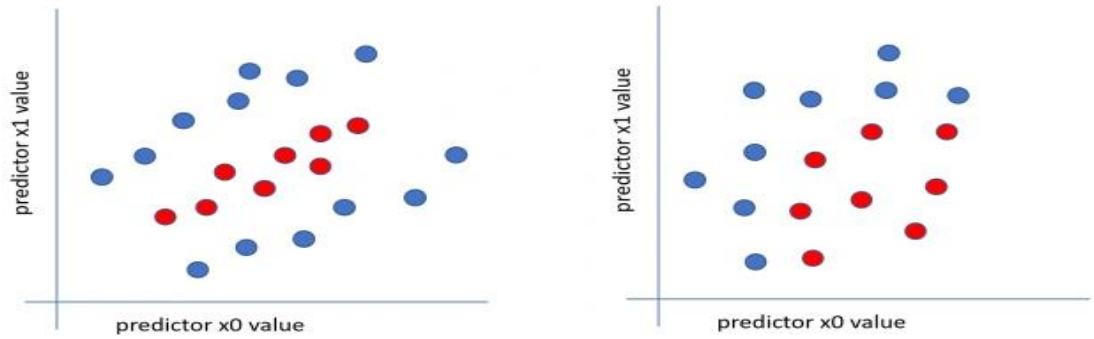




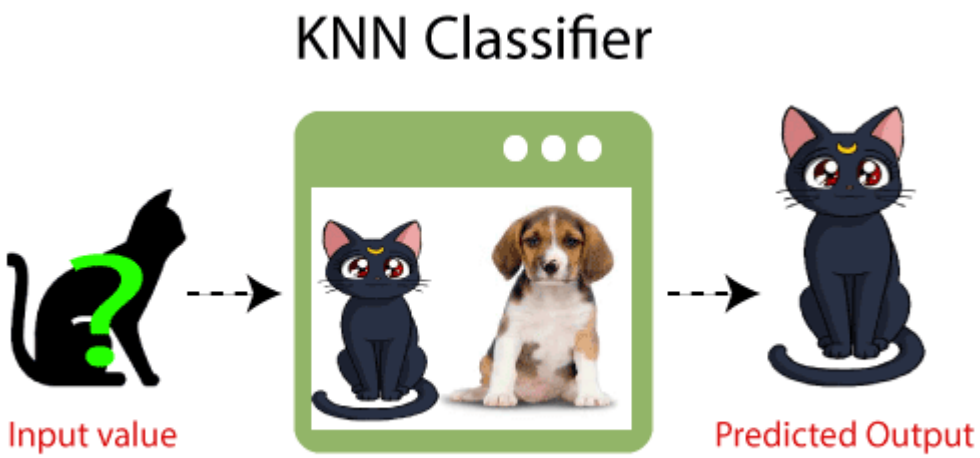
This data is linearly separable because there is a line (actually many lines) from lower left to upper right that separates the red and blue classes.
You can imagine the data represents people and the goal is to predict political conservative (red) or liberal (blue) based on age (predictor x0) and income (predictor x1).

This data is linearly separable because there is a straight line from lower left to upper right that separates the red and blue data.

Neither of these two datasets is linearly separable. The data on the left needs two straight lines. The data on the right needs a curved line.17



16) What are the basic functions of ANN?



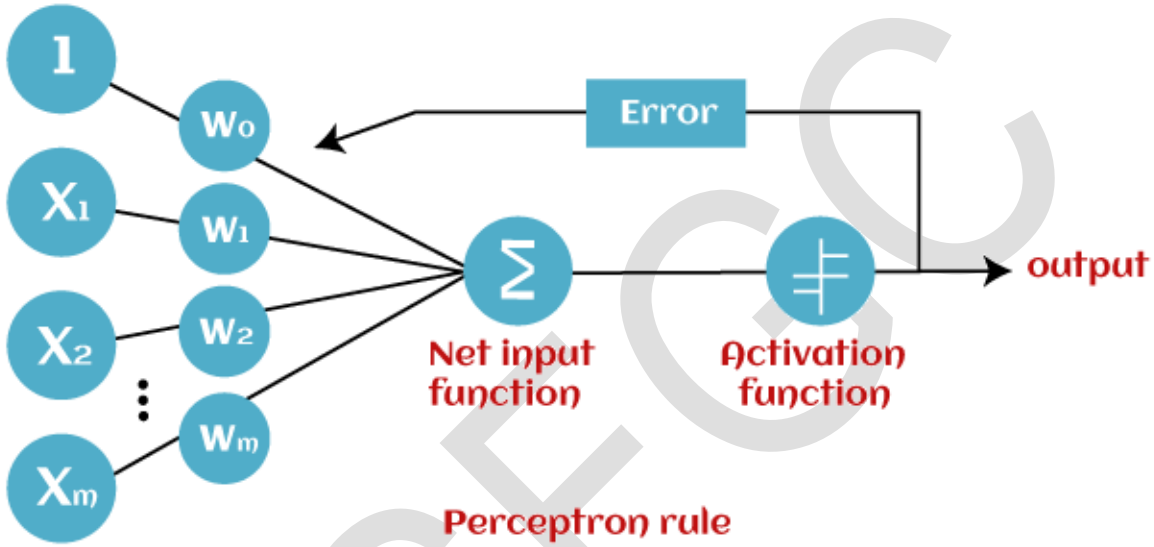
How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

17) How does Perceptron work?

In Machine Learning, Perceptron is considered as a single-layer neural network that consists of four main parameters named input values (Input nodes), weights and Bias, net sum, and an activation function.
The perceptron model begins with the multiplication of all input values and their weights, then adds these values together to create the weighted sum.
Then this weighted sum is applied to the activation function 'f' to obtain the desired output. This activation function is also known as the **step function** and is represented by '**f**'.



This step function or Activation function plays a vital role in ensuring that output is mapped between required values (0,1) or (-1,1). It is important to note that the weight of input is indicative of the strength of a node. Similarly, an input's bias value gives the ability to shift the activation function curve up or down.

Perceptron model works in two important steps as follows:

Step-1

In the first step first, multiply all input values with corresponding weight values and then add them to determine the weighted sum. Mathematically, we can calculate the weighted sum as follows:

$\sum wi*xi = x1*w1 + x2*w2 + \ldots wn*xn$

Add a special term called **bias 'b'** to this weighted sum to improve the model's performance.

$\sum \mathbf{wi*xi + b}$

Step-2

In the second step, an activation function is applied with the above-mentioned weighted sum, which gives us output either in binary form or a continuous value as follows:

Y = f(∑wi*xi + b)

18) What are the characteristics of Perceptrons?

The perceptron model has the following characteristics.

1. Perceptron is a machine learning algorithm for supervised learning of binary classifiers.
2. In Perceptron, the weight coefficient is automatically learned.
3. Initially, weights are multiplied with input features, and the decision is made whether the neuron is fired or not.
4. The activation function applies a step rule to check whether the weight function is greater than zero.
5. The linear decision boundary is drawn, enabling the distinction between the two linearly separable classes +1 and -1.
6. If the added sum of all input values is more than the threshold value, it must have an output signal; otherwise, no output will be shown.

TEN Mark Questions

19) Explain ID3 algorithm in detail?

ID3 stands for Iterative Dichotomiser 3 and is named such because the algorithm iteratively (repeatedly) dichotomizes(divides) features into two or more groups at each step.

Invented by Ross Quinlan, ID3 uses a **top-down greedy** approach to build a decision tree. In simple words, the **top-down** approach means that we start building the tree from the top and the **greedy** approach means that at each iteration we select the best feature at the present moment to create a node.

Dataset description

Example using a sample dataset of COVID-19 infection. A preview of the entire dataset is shown below.

ID	Fever	Cough	Breathing issues	Infected
1	NO	NO	NO	NO
2	YES	YES	YES	YES
3	YES	YES	NO	NO
4	YES	NO	YES	YES
5	YES	YES	YES	YES

Y and N stand for Yes and No respectively. The values or **classes** in Infected column Y and N represent Infected and Not Infected respectively.

The columns used to make decision nodes viz. 'Breathing Issues', 'Cough' and 'Fever' are called feature columns or just features and the column used for leaf nodes i.e. 'Infected' is called the target column.

Metrics in ID3

How does ID3 select the best feature?' is that ID3 uses **Information Gain** or just **Gain** to find the best feature.

Information Gain calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes. The feature with the **highest Information Gain** is selected as the **best** one.

Entropy is the measure of disorder and the Entropy of a dataset is the measure of disorder in the target feature of the dataset.

In the case of binary classification (where the target column has only two types of classes) entropy is **0** if all values in the target column are homogenous(similar) and will be **1** if the target column has equal number values for both the classes.

We denote our dataset as **S,** entropy is calculated as:

$$\textbf{Entropy}(S) = -\sum p_i * \log_2(p_i) \; ; \; i = 1 \text{ to } n$$

where,

n is the total number of classes in the target column (in our case n = 2 i.e YES and NO)

p_i is the **probability of class 'i'** or the ratio of "*number of rows with class i in the target column"* to the "*total number of rows"* in the dataset.

Information Gain for a feature column **A** is calculated as:

$$\mathbf{IG(S, A) = Entropy(S) - \sum((|S_v| / |S|) * Entropy(S_v))}$$

where $\mathbf{S_v}$ is the set of rows in **S** for which the feature column **A** has value **v**, $|\mathbf{S_v}|$ is the number of rows in $\mathbf{S_v}$ and likewise **|S|** is the number of rows in **S.**

ID3 Steps

1. Calculate the Information Gain of each feature.
2. Considering that all rows don't belong to the same class, split the dataset **S** into subsets using the feature for which the Information Gain is maximum.
3. Make a decision tree node using the feature with the maximum Information gain.
4. If all rows belong to the same class, make the current node as a leaf node with the class as its label.
5. Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

20)Explain the working of Back Propagation ?

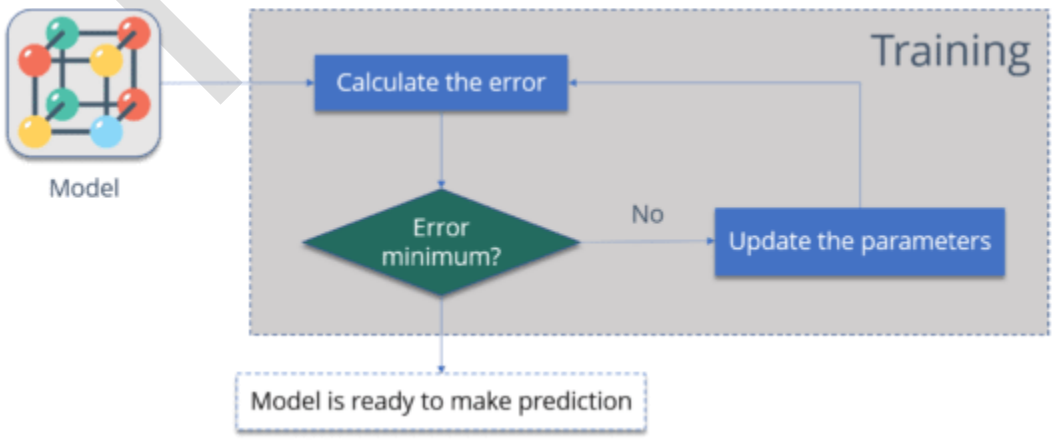
Backpropagation is a supervised learning algorithm, for training Multi-layer Perceptrons (Artificial Neural Networks).

Why We Need Backpropagation?

While designing a Neural Network, in the beginning, we initialize weights with some random values or any variable for that fact.

whatever weight values we have selected will be correct, or it fits our model the best. To reduce the error in our designed model we need to somehow explain the model to change the parameters (weights), such that error becomes minimum.

One way to train our model is called as Backpropagation. Consider the diagram below:



Steps involved in this is

- **Calculate the error** – How far is your model output from the actual output.
- **Minimum Error** – Check whether the error is minimized or not.
- **Update the parameters** – If the error is huge then, update the parameters (weights and biases). After that again check the error. Repeat the process until the error becomes minimum.
- **Model is ready to make a prediction** – Once the error becomes minimum, you can feed some inputs to your model and it will produce the output.
- **What is Backpropagation?**
- The Backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent. The weights that minimize the error function is then considered to be a solution to the learning problem.
- Consider the dataset, which has labels.
- Consider the below table:

Input	Desired Output
0	0
1	2
2	4

- Now the output of your model when 'W" value is 3:

Input	Desired Output	Model Output(W=3)
0	0	0
1	2	3
2	4	6

- Notice the difference between the actual output and the desired output:

Input	Desired Output	Model Output(W=3)	Absolute Error	Square Error
0	0	0	0	0
1	2	3	1	1
2	4	6	2	4

- Let's change the value of 'W'. Notice the error when 'W' = '4'

Input	Desired Output	Model Output(W=3)	Absolute Error	Square Error	Model Output(W=4)	Square Error
0	0	0	0	0	0	0
1	2	3	1	1	4	4
2	4	6	2	4	8	16

- Now if you notice, when we increase the value of ‘W’ the error has increased. So, obviously there is no point in increasing the value of ‘W’ further. But, what happens if I decrease the value of ‘W’? Consider the table below:

Input	Desired Output	Model Output(W=3)	Absolute Error	Square Error	Model Output(W=2)	Square Error
0	0	0	0	0	0	0
1	2	3	2	4	3	0
2	4	6	2	4	4	0
The Procedure Followed was						

- first initialized some random value to ‘W’ and propagated forward.
- Then, there is some error. To reduce that error, propagated backwards and increased the value of ‘W’.
- After that, also the error has increased. we can’t increase the ‘W’ value.
- So, again propagated backwards and decreased ‘W’ value.
- Now, noticed that the error has reduced.

we are trying to get the value of weight such that the error becomes minimum. Basically, we need to figure out whether we need to increase or decrease the weight value.

Once we know that, we keep on updating the weight value in that direction until error becomes minimum. You might reach a point, where if you further update the weight, the error will increase. At that time you need to stop, and that is your final weight value.

Consider the graph below:

We need to reach the ‘Global Loss Minimum’.

This is nothing but Backpropagation.

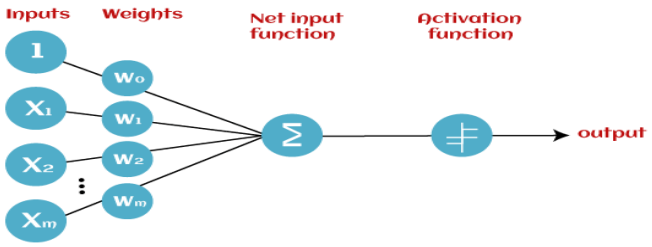
21)What is the Perceptron model in Machine Learning?

Perceptron is Machine Learning algorithm for supervised learning of various binary classification tasks. Further, ***Perceptron is also understood as an Artificial Neuron or neural network unit that helps to detect certain input data computations in business intelligence***.

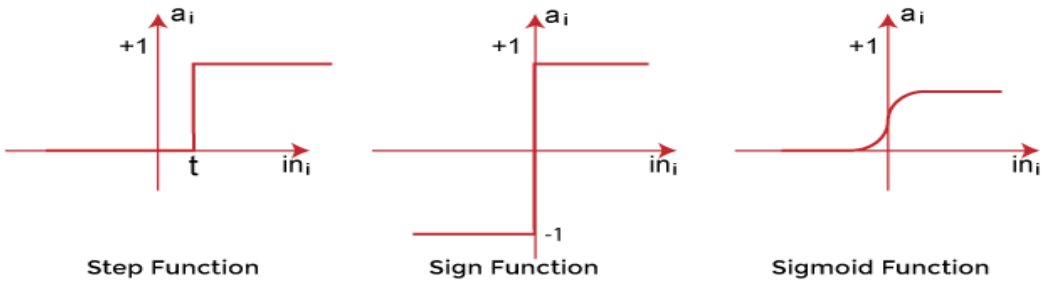
Perceptron model is also treated as one of the best and simplest types of Artificial Neural networks. However, it is a supervised learning algorithm of binary classifiers. Hence, we can consider it as a single-layer neural network with four main parameters, i.e., **input values, weights and Bias, net sum, and an activation function.**

Basic Components of Perceptron

Mr. Frank Rosenblatt invented the perceptron model as a binary classifier which contains three main components. These are as follows:



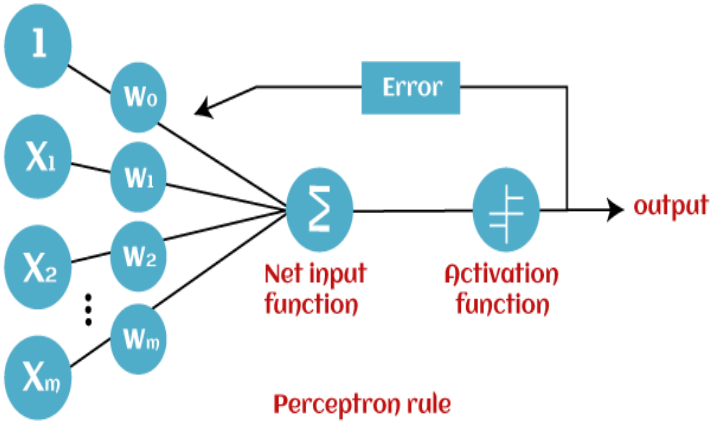
- **Input Nodes or Input Layer:**
  This is the primary component of Perceptron which accepts the initial data into the system for further processing. Each input node contains a real numerical value.
- **Wight and Bias:**
  Weight parameter represents the strength of the connection between units. This is another most important parameter of Perceptron components. Weight is directly proportional to the strength of the associated input neuron in deciding the output. Further, Bias can be considered as the line of intercept in a linear equation.
- **Activation Function:**
  These are the final and important components that help to determine whether the neuron will fire or not. Activation Function can be considered primarily as a step function.
  Types of Activation functions:
- Sign function
- Step function, and
- Sigmoid function



The data scientist uses the activation function to take a subjective decision based on various problem statements and forms the desired outputs. Activation function may differ (e.g., Sign, Step, and Sigmoid) in perceptron models by checking whether the learning process is slow or has vanishing or exploding gradients.

How does Perceptron work?

In Machine Learning, Perceptron is considered as a single-layer neural network that consists of four main parameters named input values (Input nodes), weights and Bias, net sum, and an activation function.
The perceptron model begins with the multiplication of all input values and their weights, then adds these values together to create the weighted sum.
Then this weighted sum is applied to the activation function 'f' to obtain the desired output. This activation function is also known as the **step function** and is represented by **'f'**.



This step function or Activation function plays a vital role in ensuring that output is mapped between required values (0,1) or (-1,1). It is important to note that the weight of input is indicative of the strength of a node. Similarly, an input's bias value gives the ability to shift the activation function curve up or down.
Perceptron model works in two important steps as follows:

Step-1
In the first step first, multiply all input values with corresponding weight values and then add them to determine the weighted sum. Mathematically, we can calculate the weighted sum as follows:
$\sum wi*xi = x1*w1 + x2*w2 + \ldots wn*xn$

Add a special term called **bias 'b'** to this weighted sum to improve the model's performance.
∑wi*xi + b
Step-2
In the second step, an activation function is applied with the above-mentioned weighted sum, which gives us output either in binary form or a continuous value as follows:
Y = f(∑wi*xi + b).

Perceptron Function

Perceptron function "f(x)" can be achieved as output by multiplying the input 'x' with the learned weight coefficient 'w'.
Mathematically, we can express it as follows:
f(x)=1; if w.x+b>0
otherwise, f(x)=0

- 'w' represents real-valued weights vector
- 'b' represents the bias
- 'x' represents a vector of input x values.

Unit – III Probabilistic and Stochastic Models

TWO Mark Questions:

1)Define Bayes Theorem?

Bayes Theorem is named for English mathematician Thomas Bayes, who worked extensively in decision theory, the field of mathematics that involves probabilities. The Bayesian method of calculating conditional probabilities is used in machine learning applications that involve classification tasks.

For example: if we have to calculate the probability of taking a blue ball from the second bag out of three different bags of balls, where each bag contains three different colour balls viz. red, blue, black. In this case, the probability of occurrence of an event is calculated depending on other conditions is known as conditional probability.

2)What is the formula for Bayes theorem?

The formula for Bayes theorem is:
P(A|B)= [P(B|A). P(A)]/P(B)
Where P(A) and P(B) are the probabilities of events A and B.
P(A|B) is the probability of event A given B
P(B|A) is the probability of event B given A.

3) Define Concept Learning?

The concept learning can be formulated as ***"Problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples"-***
The special Set of Features differentiated from others is called a concept.
Similarly, machines can also learn from the concepts to identify whether an object belongs to a specific category or not by processing past/training data to find a hypothesis that best fits the training examples.

4) Define Maximuum Likelihood?

The maximum likelihood estimation is a method that determines values for parameters of the model. It is the statistical method of estimating the parameters of the probability distribution by maximizing the likelihood function. The point in which the parameter value that maximizes the likelihood function is called the maximum likelihood estimate.

5) What are the areas which supports concept learning?

1. Training data (Past experiences to train our models)
2. Target Concept (Hypothesis to identify data objects)
3. Actual data objects (For testing the models)

6) What are the types of Bayes Classifiers?

There are three types of Naive Bayes Model, which are given below:

- **Gaussian**: The Gaussian model assumes that features follow a normal distribution.
- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed.
- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables.

7) Define hidden markov models?

The Hidden Markov model is a probabilistic model which is used to explain or derive the probabilistic characteristic of any random process. It basically says that an observed event will not be corresponding to its step-by-step status but related to a set of probability distributions.

FIVE & TEN Marks Questions

8) Explain the working of Bayes Theorem ?

Bayes' theorem can be derived using product rule and conditional probability of event X with known event Y:

- According to the product rule we can express as the probability of event X with known event Y as follows;

  P(X ? Y)= P(X|Y) P(Y) {equation 1}

- Further, the probability of event Y with known event X:

  P(X ? Y)= P(Y|X) P(X) {equation 2}

Mathematically, Bayes theorem can be expressed by combining both equations on right hand side. We will get:

$$P(X|Y) = \frac{P(Y|X).P(X)}{P(Y)}$$

Here, both events X and Y are independent events which means probability of outcome of both events does not depends one another.

The above equation is called as Bayes Rule or Bayes Theorem.

- P(X|Y) is called as **posterior**, which we need to calculate. It is defined as updated probability after considering the evidence.
- P(Y|X) is called the likelihood. It is the probability of evidence when hypothesis is true.
- P(X) is called the **prior probability**, probability of hypothesis before considering the evidence
- P(Y) is called marginal probability. It is defined as the probability of evidence under any consideration.
- Hence, Bayes Theorem can be written as:
- **posterior = likelihood * prior / evidence**
- Let's understand the use of Bayes theorem in machine learning with below example.
- Suppose, we have a vector A with I attributes. It means
- A = A1, A2, A3, A4……Ai
- Further, we have n classes represented as C1, C2, C3, C4……Cn.
- These are two conditions given to us, and our classifier that works on Machine Language has to predict A and the first thing that our classifier has to choose will be the best possible class. So, with the help of Bayes theorem, we can write it as:
- P(Ci/A)= [P(A/Ci) * P(Ci)] / P(A)
- Here;
- P(A) is the condition-independent entity.
- P(A) will remain constant throughout the class means it does not change its value with respect to change in class. To maximize the P(Ci/A), we have to maximize the value of term P(A/Ci) * P(Ci).
- P(Ai/C)= P(A1/C)* P(A2/C)* P(A3/C)*……*P(An/C)
- Hence, by using Bayes theorem in Machine Learning we can easily describe the possibilities of smaller events.
- With n number classes on the probability list let's assume that the possibility of any class being the right answer is equally likely. Considering this factor, we can say that:
- P(C1)=P(C2)-P(C3)=P(C4)=…..=P(Cn).
- This process helps us to reduce the computation cost as well as time. This is how Bayes theorem plays a significant role in Machine Learning and Naïve Bayes theorem has simplified the conditional probability tasks without affecting the precision.

- P(Ai/C)= P(A1/C)* P(A2/C)* P(A3/C)*……*P(An/C)
- Hence, by using Bayes theorem in Machine Learning we can easily describe the possibilities of smaller events.

9) What are the advantages of Naïve Bayes Classifier?

Advantages of Naïve Bayes Classifier in Machine Learning:

- It is one of the simplest and effective methods for calculating the conditional probability and text classification problems.
- A Naïve-Bayes classifier algorithm is better than all other models where assumption of independent predictors holds true.
- It is easy to implement than other models.
- It requires small amount of training data to estimate the test data which minimize the training time period.
- It can be used for Binary as well as Multi-class Classifications.

10) Explain Hidden Markov Model With an Example?

A Hidden Markov Model (HMM) is a statistical model which is also used in machine learning. It can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable.

To explain it more we can take the example of two friends, Rahul and Ashok. Now Rahul completes his daily life works according to the weather conditions. Major three activities completed by Rahul are- go jogging, go to the office, and cleaning his residence. What Rahul is doing today depends on whether and whatever Rahul does he tells Ashok and Ashok has no proper information about the weather But Ashok can assume the weather condition according to Rahul work.

Ashok believes that the weather operates as a discrete Markov chain, wherein the chain there are only two states whether the weather is Rainy or it is sunny. The condition of the weather cannot be observed by Ashok, here the conditions of the weather are hidden from Ashok.

On each day, there is a certain chance that Bob will perform one activity from the set of the following activities {“jog”, “work”,” clean”}, which are depending on the weather. Since Rahul tells Ashok that what he has done, those are the observations. The entire system is that of a hidden Markov model (HMM).

Here we can say that the parameter of HMM is known to Ashok because he has general information about the weather and he also knows what Rahul likes to do on average.

So let's consider a day where Rahul called Ashok and told him that he has cleaned his residence. In that scenario, Ashok will have a belief that there are more chances of a rainy day and we can say that belief Ashok has is the start probability of HMM let's say which is like the following.

The states and observation are:

states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

And the start probability is:

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

Now the distribution of the probability has the weightage more on the rainy day stateside so we can say there will be more chances for a day to being rainy again and the probabilities for next day weather states are as following

```
transition_probability = {

  'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
  'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},

  }
```

From the above we can say the changes in the probability for a day is transition probabilities and according to the transition probability the emitted results for the probability of work that Rahul will perform is

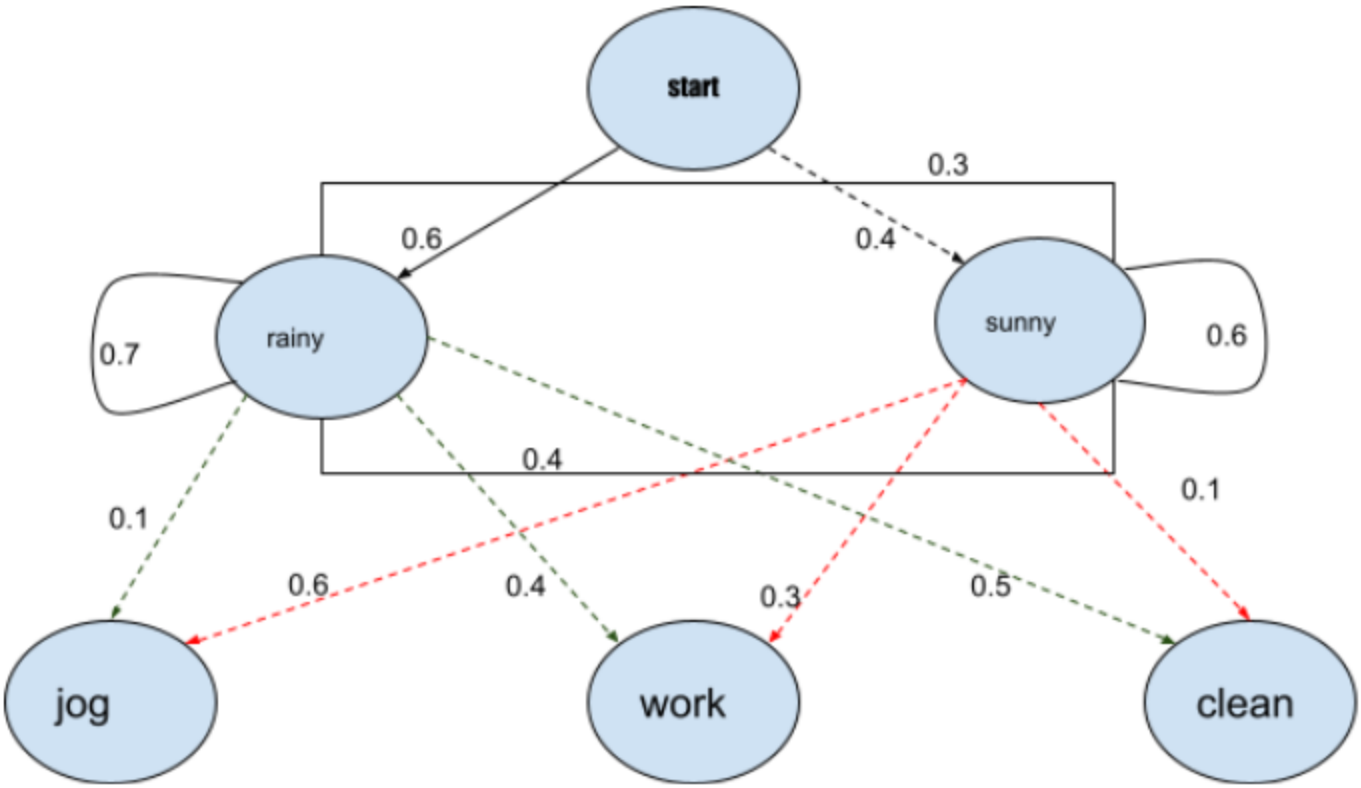
```
emission_probability = {

  'Rainy' : {'jog': 0.1, 'work': 0.4, 'clean': 0.5},
  'Sunny' : {'jog': 0.6, 'work: 0.3, 'clean': 0.1},

  }
```

This probability can be considered as the emission probability. Using the emission probability Ashok can predict the states of the weather or using the transition probabilities Ashok can predict the work which Rahul is going to perform the next day.

Below image shown the HMM process for making probabilities



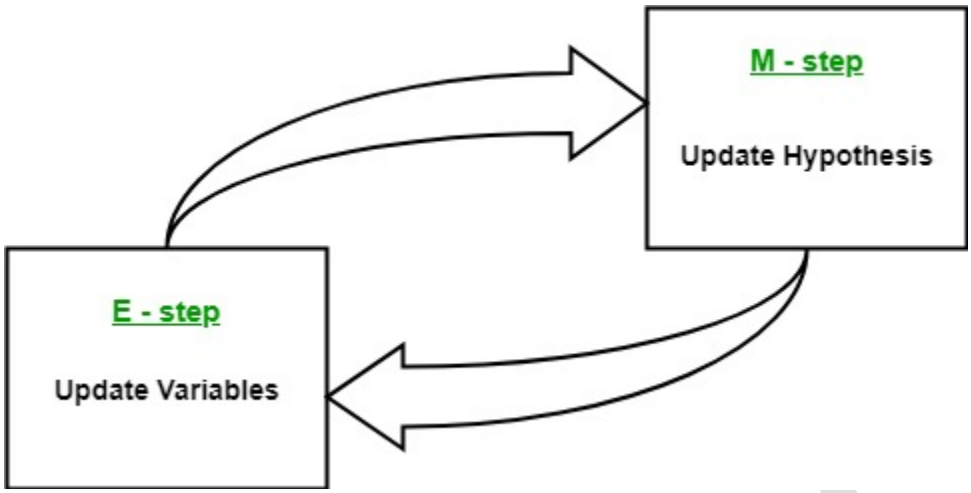
So here from the above intuition and the example we can understand how we can use this probabilistic model to make a prediction.

11) Explain Expectation Maximization in ML?

A latent variable model consists of **observable** variables along with **unobservable** variables. Observed variables are those variables in the dataset that can be measured whereas unobserved (latent/hidden) variables are inferred from the observed variables.

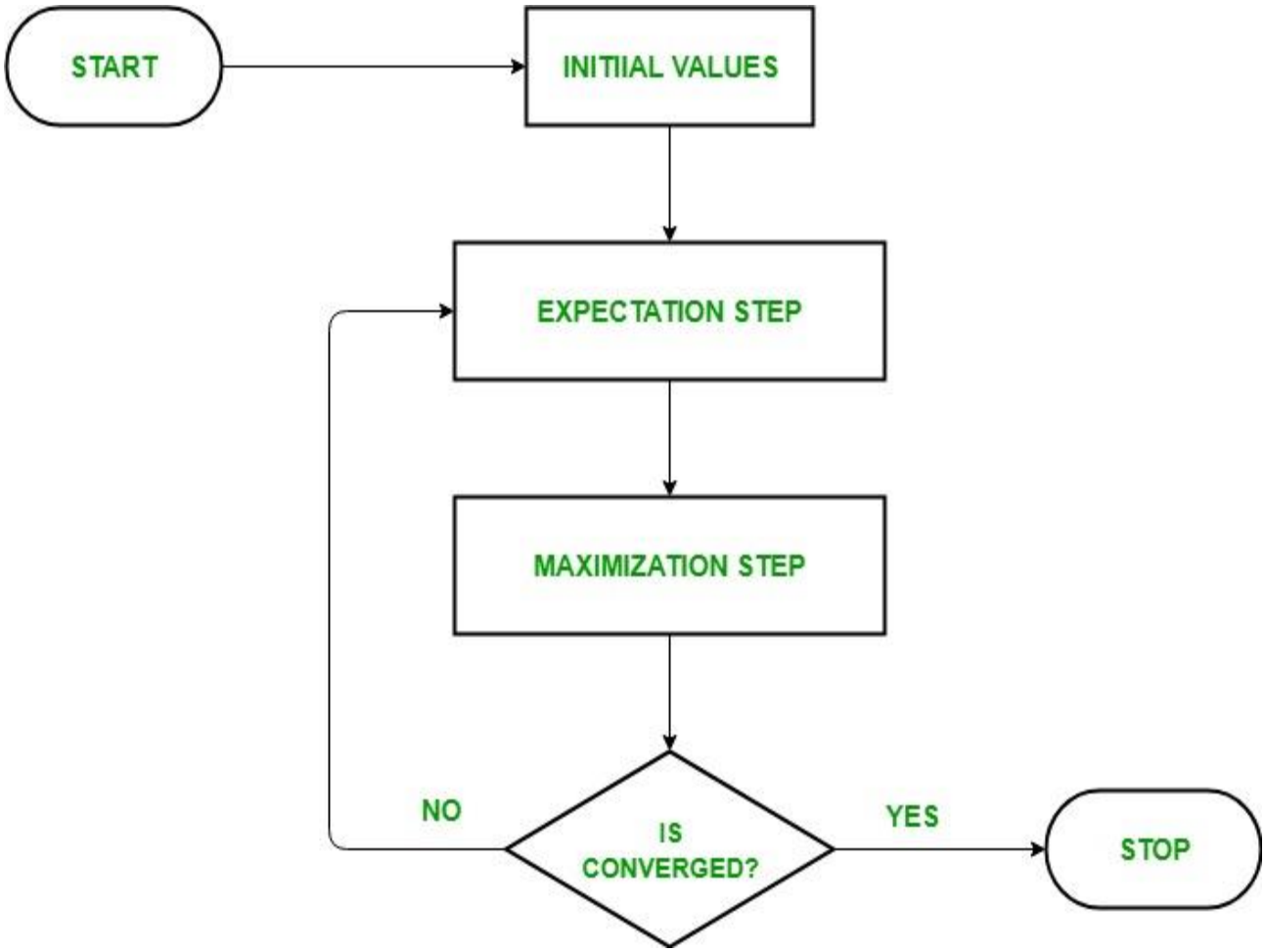
Detailed Explanation of the EM Algorithm

- Given a set of incomplete data, start with a set of initialized parameters.
- **Expectation step (E – step):** In this expectation step, by using the observed available data of the dataset, we can try to estimate or guess the values of the missing data. Finally, after this step, we get complete data having no missing values.
- **Maximization step (M – step):** Now, we have to use the complete data, which is prepared in the expectation step, and update the parameters.
- Repeat step 2 and step 3 until we converge to our solution.



The Expectation-Maximization algorithm aims to use the available observed data of the dataset to estimate the missing data of the latent variables and then using that data to update the values of the parameters in the maximization step.

- **Initialization Step:** In this step, we initialized the parameter values with a set of initial values, then give the set of incomplete observed data to the system with the assumption that the observed data comes from a specific model **i.e, probability distribution.**
- **Expectation Step:** In this step, by using the observed data to estimate or guess the values of the missing or incomplete data. It is used to update the variables.
- **Maximization Step:** In this step, we use the complete data generated in the **"Expectation"** step to update the values of the parameters i.e, **update the hypothesis**.
- **Checking of convergence Step:** Now, in this step, we checked whether the values are converging or not, if yes, then stop otherwise repeat these two steps i.e, the **"Expectation"** step and **"Maximization"** step until the convergence occurs.


START
INITIIAL VALUES
EXPECTATION STEP
MAXIMIZATION STEP
IS CONVERGED?
NO
YES
STOP

UNIT 4- ASSOCIATION MINING AND UNSUPERVISED LEARNING

TWO Marks Questions

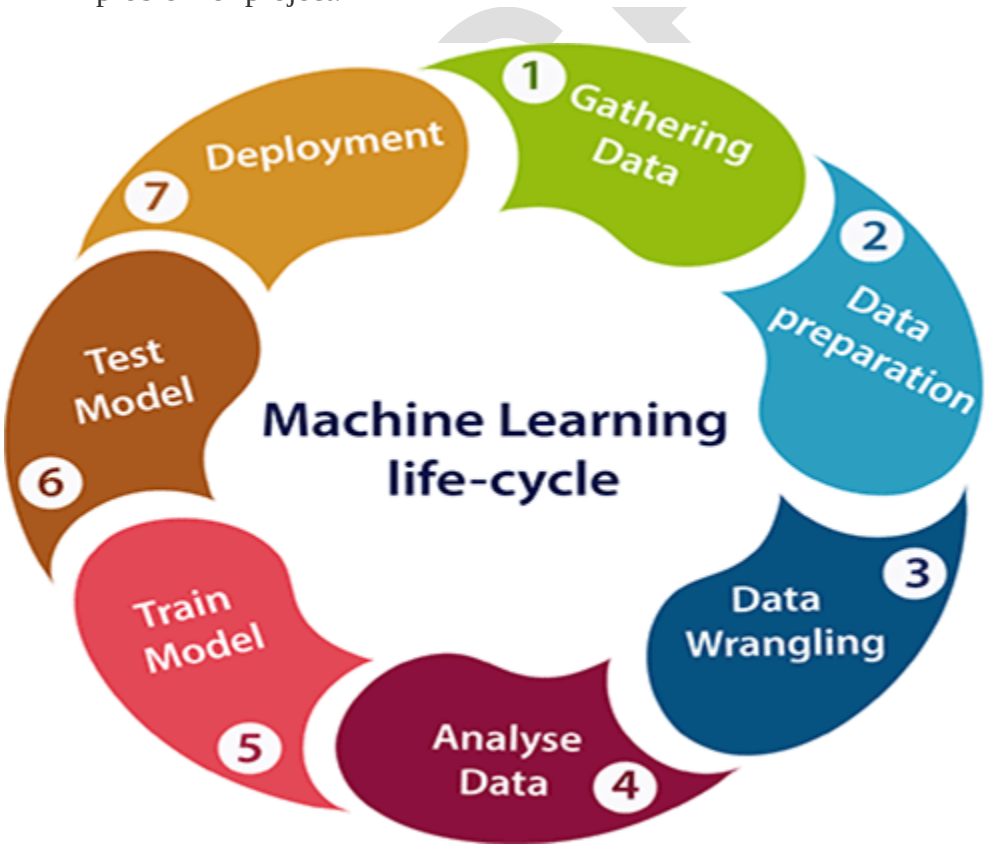
1)Define unsupervised learning?

Unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention.

Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

2)Define of Machine Learning?

Machine learning has given the computer systems the abilities to automatically learn without being explicitly programmed. Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project.



3)Define Association Mining?

Association Rule Mining, as the name suggests, association rules are simple If/Then statements that help discover relationships between seemingly independent relational databases or other data repositories.

Association rule mining is a procedure which aims to observe frequently occurring patterns, correlations, or associations from datasets found in various kinds of databases such as relational databases, transactional databases, and other forms of repositories.

4)What are the two parts of Association rule?

An association rule has 2 parts:
an antecedent (if) and
a consequent (then)

5)Define Apriori Algorithm?

This algorithm was given by the **R. Agrawal** and **Srikant** in the year **1994**. It is mainly used for *market basket analysis* and helps to find those products that can be bought together.

The Apriori algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a **breadth-first search** and **Hash Tree** to calculate the itemset associations efficiently.

6) What is Frequent Itemset?

Frequent itemsets are those items whose support is greater than the threshold value or user-specified minimum support. It means if A & B are the frequent itemsets together, then individually A and B should also be the frequent itemset.

Suppose there are the two transactions: A= {1,2,3,4,5}, and B= {2,3,7}, in these two transactions, 2 and 3 are the frequent itemsets.

7) How does Association Rule Learning work?

Association rule learning works on the concept of If and Else Statement, such as if A then B.

Here the If element is called **antecedent**, and then statement is called as **Consequent**.

These types of relationships where we can find out some association or relation between two items is known *as single cardinality*.

It is all about creating rules, and if the number of items increases, then cardinality also increases accordingly. So, to measure the associations between thousands of data items, there are several metrics. These metrics are given below:

- **Support**
- **Confidence**
- **Lift**

8)Define Support?

Support

Support is the frequency of A or how frequently an item appears in the dataset. It is defined as the fraction of the transaction T that contains the itemset X. If there are X datasets, then for transactions T, it can be written as:

$$\text{Supp(X)} = \frac{Freq(X)}{T}$$

9)Define Confidence?

Confidence

Confidence indicates how often the rule has been found to be true. Or how often the items X and Y occur together in the dataset when the occurrence of X is already given. It is the ratio of the transaction that contains X and Y to the number of records that contain X.

$$\text{Confidence} = \frac{Freq(X,Y)}{Freq(X)}$$

10)Define TP and FP Trees?

- True positive (*TP*): Prediction is +ve
- True negative *(TN*): Prediction is -ve
- False positive (*FP*): Prediction is +ve
- False negative (*FN*): Prediction is -ve

Five Mark Questions

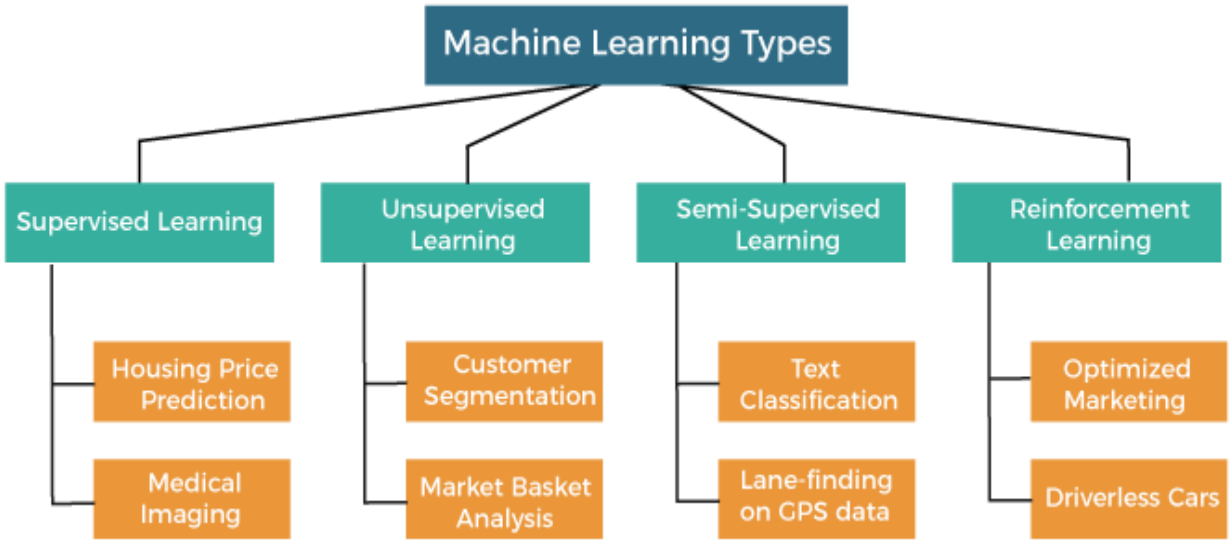
11) Explain the types of ML?

Machine learning is a subset of AI, which enables the machine to automatically learn from data, improve performance from past experiences, and make predictions. Machine learning contains a set of algorithms that work on a huge amount of data. Data is fed to these algorithms to train them, and on the basis of training, they build the model & perform a specific task.

ML algorithms help to solve different business problems like Regression, Classification, Forecasting, Clustering, and Associations, etc.

Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning



12)Explain Clustering?

Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. ***A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."***

It does it by finding some similar patterns in the unlabelled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.

After applying this clustering technique, each cluster or group is provided with a cluster-ID. ML system can use this id to simplify the processing of large and complex datasets.

The clustering technique is commonly used for **statistical data analysis.**

13)What are the types of Clustering?

Types of Clustering Methods

The clustering methods are broadly divided into **Hard clustering** (datapoint belongs to only one group) and **Soft Clustering** (data points can belong to another group also).

But there are also other various approaches of Clustering exist. Below are the main clustering methods used in Machine learning:

1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

14)Explain Hierarchical Clustering?

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as **hierarchical cluster analysis** or HCA.

In this algorithm, the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

The hierarchical clustering technique has two approaches:

1. **Agglomerative:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.

2. **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

15)Explain K-Means Clusturing?

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.
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, and for K=3, there will be three clusters, and so on.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

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 working of the K-Means algorithm is explained in the below 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.

16)What is the need for confusion Matrix?

- It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
- It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
- With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

17)Explain Apriori Algorithm?

The Apriori algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a **breadth-first search** and **Hash Tree** to calculate the itemset associations efficiently. It is the iterative process for finding the frequent itemsets from the large dataset.

Steps for Apriori Algorithm

Below are the steps for the apriori algorithm:

Step-1: Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

Step-2: Take all supports in the transaction with higher support value than the minimum or selected support value.

Step-3: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

Step-4: Sort the rules as the decreasing order of lift.

Apriori Algorithm Working

Example: Suppose we have the following dataset that has various transactions, and from this dataset, we need to find the frequent itemsets and generate the association rules using the Apriori algorithm:

TID	ITEMSETS
T1	A, B
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

Given: Minimum Support= 2, Minimum Confidence= 50%

Solution:

Step-1: Calculating C1 and L1:

- In the first step, we will create a table that contains support count (The frequency of each itemset individually in the dataset) of each itemset in the given dataset. This table is called the **Candidate set or C1.**

Itemset	Support_Count
A	6
B	7
C	5
D	2
E	1

- Now, we will take out all the itemsets that have the greater support count that the Minimum Support (2). It will give us the table for the **frequent itemset L1.** Since all the itemsets have greater or equal support count than the minimum support, except the E, so E itemset will be removed.

Itemset	Support_Count
A	6
B	7
C	5
D	2

Step-2: Candidate Generation C2, and L2:

- In this step, we will generate C2 with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets.
- After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e., how many times these pairs have occurred together in the given dataset. So, we will get the below table for C2:

Itemset	Support_Count
{A, B}	4
{A,C}	4
{A, D}	1
{B, C}	4
{B, D}	2
{C, D}	0

- Again, we need to compare the C2 Support count with the minimum support count, and after comparing, the itemset with less support count will be eliminated from the table C2. It will give us the below table for L2

Itemset	Support_Count
{A, B}	4
{A, C}	4
{B, C}	4
{B, D}	2

A, B, C, D

Step-3: Candidate generation C3, and L3:

- For C3, we will repeat the same two processes, but now we will form the C3 table with subsets of three itemsets together, and will calculate the support count from the dataset. It will give the below table:

Itemset	Support_Count
{A, B, C}	2
{B, C, D}	1
{A, C, D}	0
{A, B, D}	0

- Now we will create the L3 table. As we can see from the above C3 table, there is only one combination of itemset that has support count equal to the minimum support count. So, the L3 will have only one combination, i.e., **{A, B, C}.**

Step-4: Finding the association rules for the subsets:

To generate the association rules, first, we will create a new table with the possible rules from the occurred combination {A, B.C}. For all the rules, we will calculate the Confidence using formula **sup(A ^B)/A.** After calculating the confidence value for all rules, we will exclude the rules that have less confidence than the minimum threshold(50%).

Consider the below table:

Rules	**Support**	**Confidence**
A ^B → C	2	Sup{(A ^B) ^C}/sup(A ^B)= 2/4=0.5=50%
B^C → A	2	Sup{(B^C) ^A}/sup(B ^C)= 2/4=0.5=50%
A^C → B	2	Sup{(A ^C) ^B}/sup(A ^C)= 2/4=0.5=50%
C→ A ^B	2	Sup{(C^(A ^B)}/sup(C)= 2/5=0.4=40%
A→ B^C	2	Sup{(A^(B ^C)}/sup(A)= 2/6=0.33=33.33%
B→ B^C	2	Sup{(B^(B ^C)}/sup(B)= 2/7=0.28=28%

As the given threshold or minimum confidence is 50%, so the first three rules **A ^B → C, B^C → A, and A^C → B** can be considered as the strong association rules for the given problem.

18)Explain Agglomerative and divise clustering?

1. ***Agglomerative clustering:***

   Agglomerative Clustering is a bottom-up strategy in which each data point is originally a cluster of its own, and as one travels up the hierarchy, more pairs of clusters are combined. In it, two nearest clusters are taken and joined to form one single cluster.

2.Divisive clustering:

The divisive clustering algorithm is a top-down clustering strategy in which all points in the dataset are initially assigned to one cluster and then divided iteratively as one progresses down the hierarchy.

It partitions data points that are clustered together into one cluster based on the slightest difference. This process continues till the desired number of clusters is obtained.

How does it work?

Each observation is treated as a separate cluster in hierarchical clustering. After that, it repeats the next two steps:

1. Finds the two clusters that are the closest together
2. Combines the two clusters that are the most similar. This iterative process is repeated until all of the clusters have been integrated.

In it, there is one cluster that after combining with data points closest to it, starts getting bigger. The same cluster gets bigger as long as all the data points are inside a single cluster.

19) Explain FP Growth?

Association rule mining is a two-step process:

1. Finding frequent Itemsets
2. Generation of strong association rules from frequent itemsets

Finding Frequent Itemsets

Frequent itemsets can be found using two methods, viz Apriori Algorithm and FP growth algorithm.

Apriori algorithm generates all itemsets by scanning the full transactional database. Whereas the FP growth algorithm only generates the frequent itemsets according to the minimum support defined by the user. Since Apriori scans the whole database multiple times, it Is more resource-hungry and the time to generate the association rules increases exponentially with the increase in the database size. On the other hand, the FP growth algorithm doesn't scan the whole database multiple times and the scanning time increases linearly. Hence, the FP growth algorithm is much faster than the Apriori algorithm.

1. FP Tree construction by compressing the DB representing frequent items

Compressing the transactional database to mine association rules by finding frequent itemsets into a frequent pattern tree or FP-tree. This also retains the itemset association information.

So let's start with a small transaction data to understand the construction of the FP tree. The transaction which we consider here suppose consists of 5 items such as-

Asparagus (A), Corn (C), Beans (B), Tomatoes (T) & Squash (S)

Table 1

Transaction ID	**List of items in the transaction**
T1	B , A , T
T2	A , C
T3	A , S
T4	B , A , C
T5	B , S
T6	A , S
T7	B , S

T8	B , A , S , T
T9	B , A , S

So for example, for the first transaction T1 consists of three items such as Beans (B), Asparagus (A), and Tomatoes (T). Similarly, the transaction T6 contains the items Asparagus (A) and Squash (S). Let us also consider the minimum support for this small transaction data to be 2. Hence, min_support = 2.

First of all, we need to create a table of item counts in the whole transactional database as below:

Table 2

Item	**Support Count**
Beans (B)	6
Asparagus (A)	7
Squash (S)	6
Corn (C)	2
Tomatoes (T)	2

This is simply the count of each item, such as if we see Squash (S) has been bought in 6 transactions viz, T3, T5, T6, T7, T8 & T9, so the support count is 6 for Squash.

Next, we just need to sort the item list in descending order for their support count. Hence the table of support count may now be as represented in table 3 below:

Table 3

tem	**Support Count**
Asparagus (A)	7

Beans (B)	6
Squash (S)	6
Corn (C)	2
Tomatoes (T)	2

Beans (B) & Squash (S) have the same support count of 6 and any of them can be written first. Here, we have written Beans (B) first. Similarly, Corn(C) & tomatoes can also be listed in the same fashion.

Now we are ready to start with the construction of the FP Tree. The FP tree root node is usually represented with a NULL root node. Now consider the Transaction T1. T1 consists of Beans (B), Asparagus (A) & tomatoes (T). Now out of these three items, we need to look for the item which has the maximum support count. Since Asparagus (A) has the highest support count of 7, we will extend the tree from its root node to A as Asparagus. Since this is the first transaction, the count is denoted by A:1. Let's look further, out of Beans(B) and Tomatoes (T) the support count of Beans is 6, and the support count for Tomatoes is 2. From Asparagus, we can extend the tree to Beans (B:1 for the first transaction consisting of beans) and after that T:1 for Tomatoes. The tree structure is as below in Figure 1.

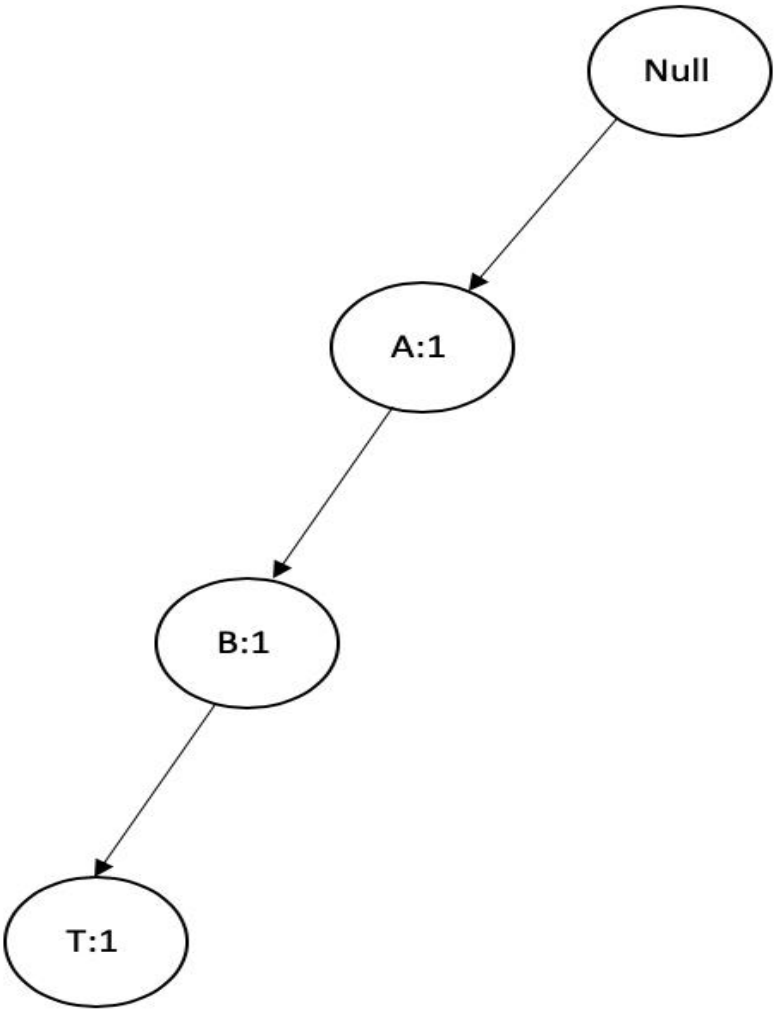


Figure 1 : Transaction ABT

Going further in the transaction T2, there are two items viz Asparagus(A) and Corn (C). Since the support count for Asparagus is 7 and Corn (C) is 2, we need to consider Asparagus (A) first and going from the root node we need to see if there is any branch that is extended to Asparagus (A) which is true in this case, and we can increase the count as A:2 (Figure 2). But after that, there is no node connected to Asparagus (A) to corn (C) we need to create another branch for Corn as C:1.

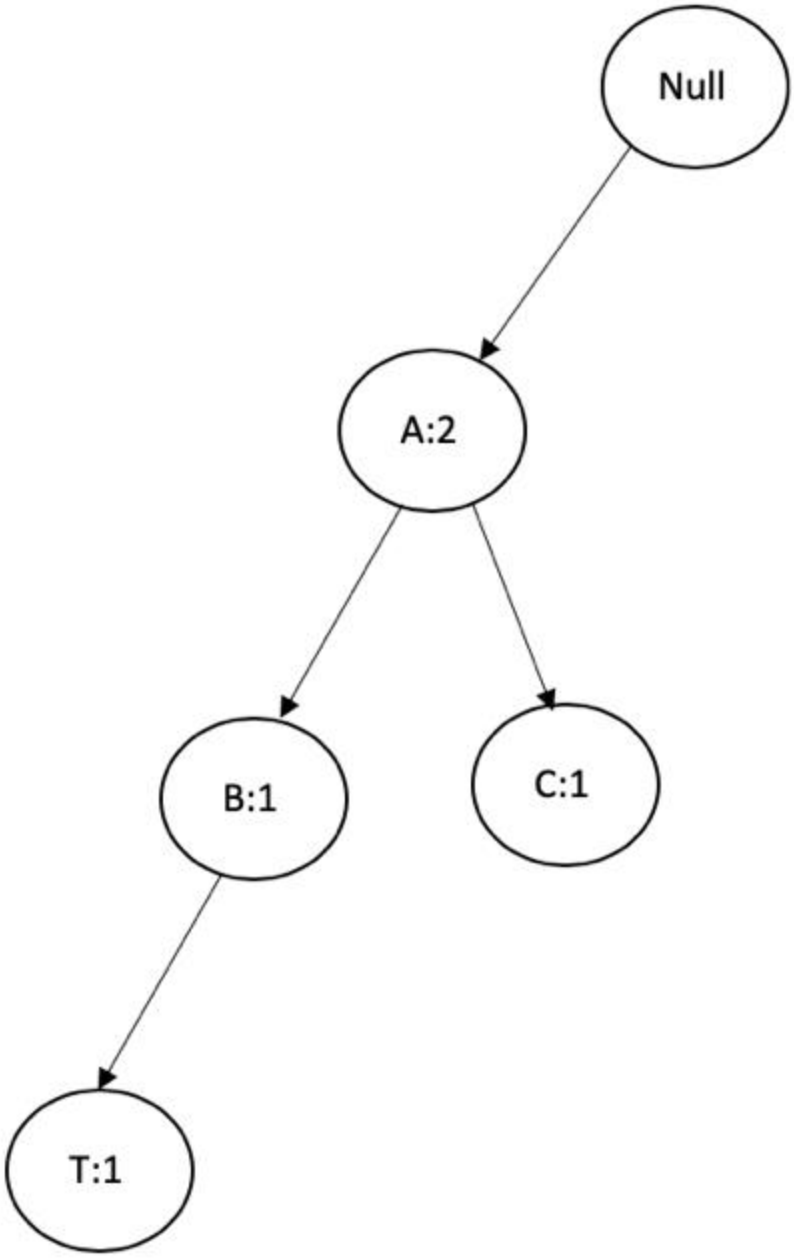


Figure 2 : Transaction AC

Similarly, for transaction T3 we have Asparagus (A) then Squash (S) in the descending order of their support count. So Asparagus(A) count has been increased from A:2 to A:3, and further, we can see that there aren't any nodes for Squash from Asparagus, so we need to create another branch going for a Squash node S:1, as described in Figure 3.

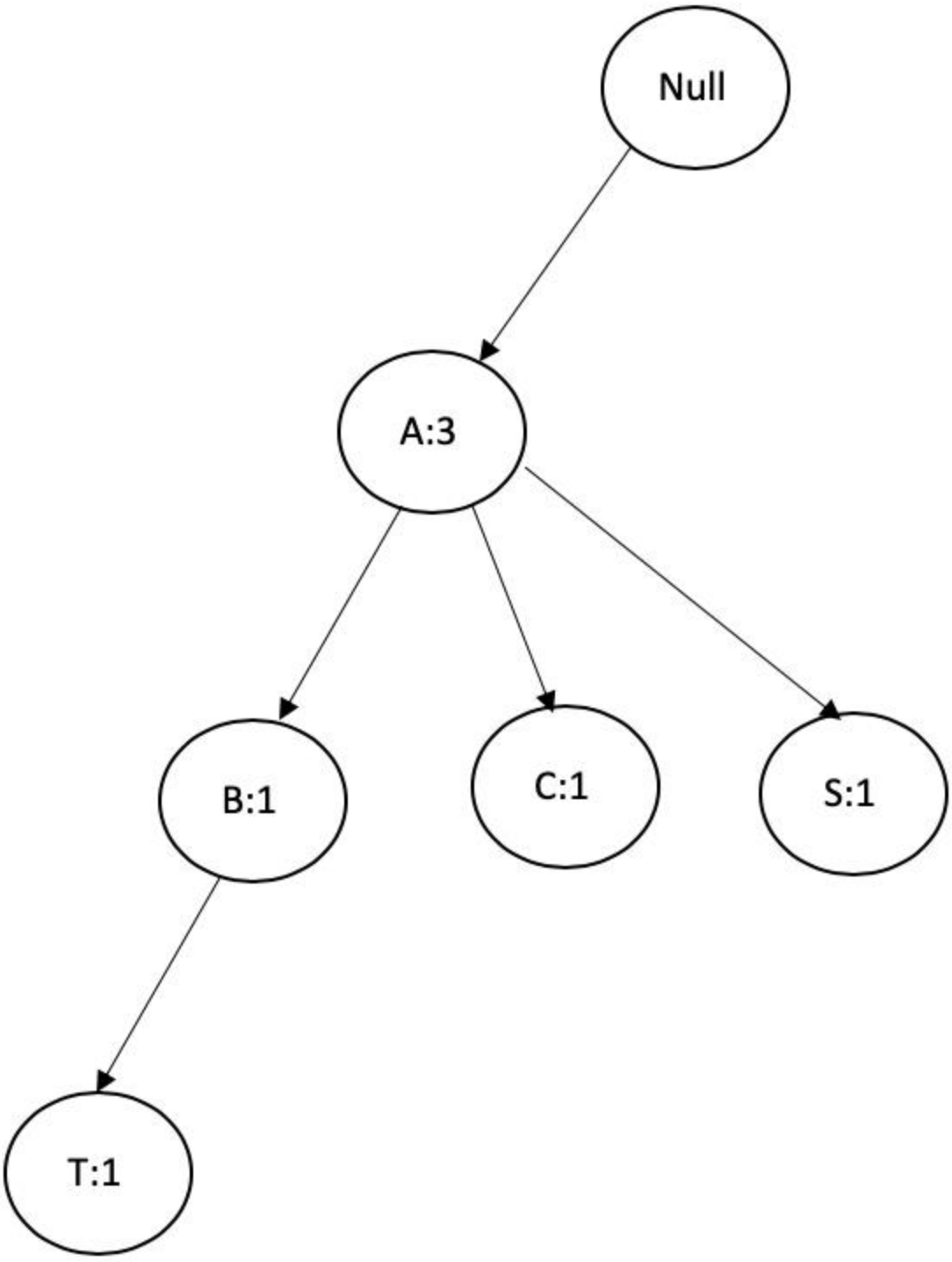


Figure 3 : Transaction AS

For transaction 4, we can draw the node as below shown in Figure 4. Here the count for Asparagus and Beans has been increased to A:4 and B:2, but after that, we can see there isn't any branch that extends to corn (C), so we need to create a node for C:1.

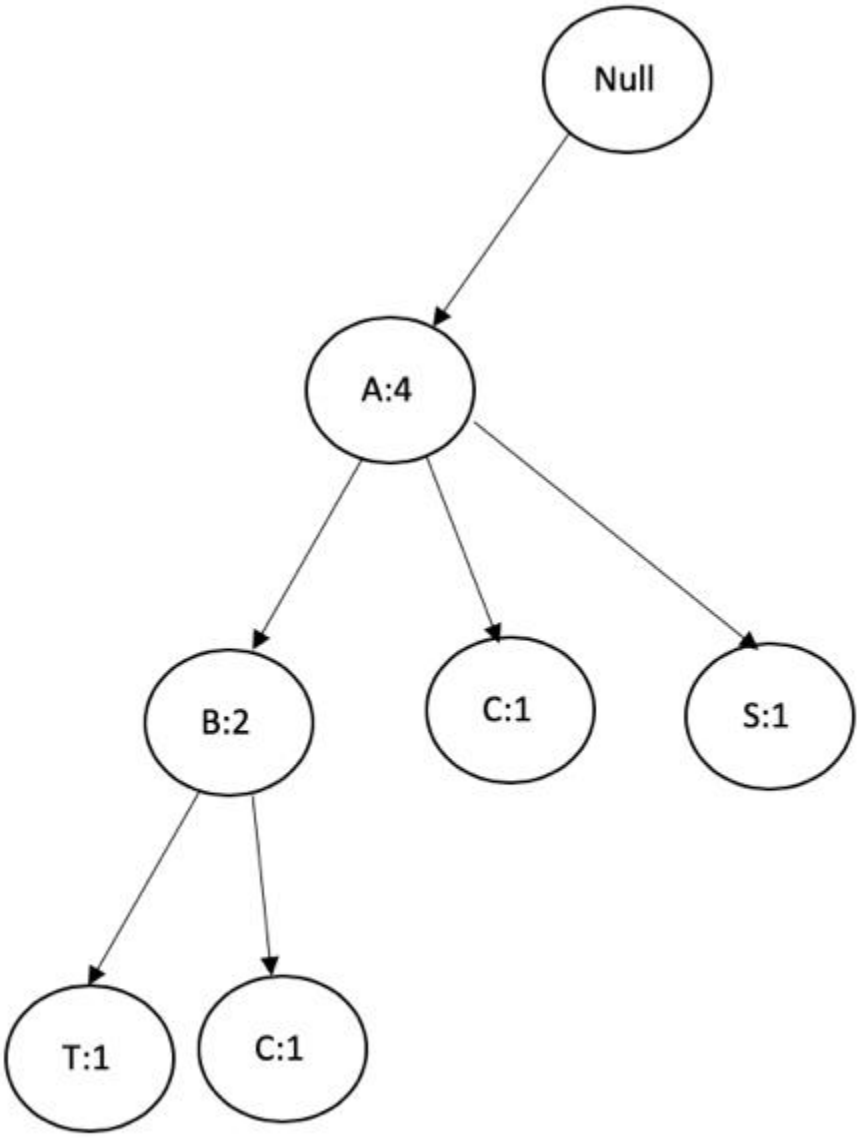


Figure 4 : Transaction ABC

For transaction 5, we can see that it contains Beans (B) and Squash (S), but there aren't any direct node linked to Beans from the Null root node, we need to create a new branch from the Null node as depicted in Figure 5 with B:1 and S:1 counts.

Transaction 6 to transaction 10 are self-explanatory. For your reference, the figures have been provided for each transaction. So for transaction 6 check figure 6, for transaction 7 check Figure 7, and so on.

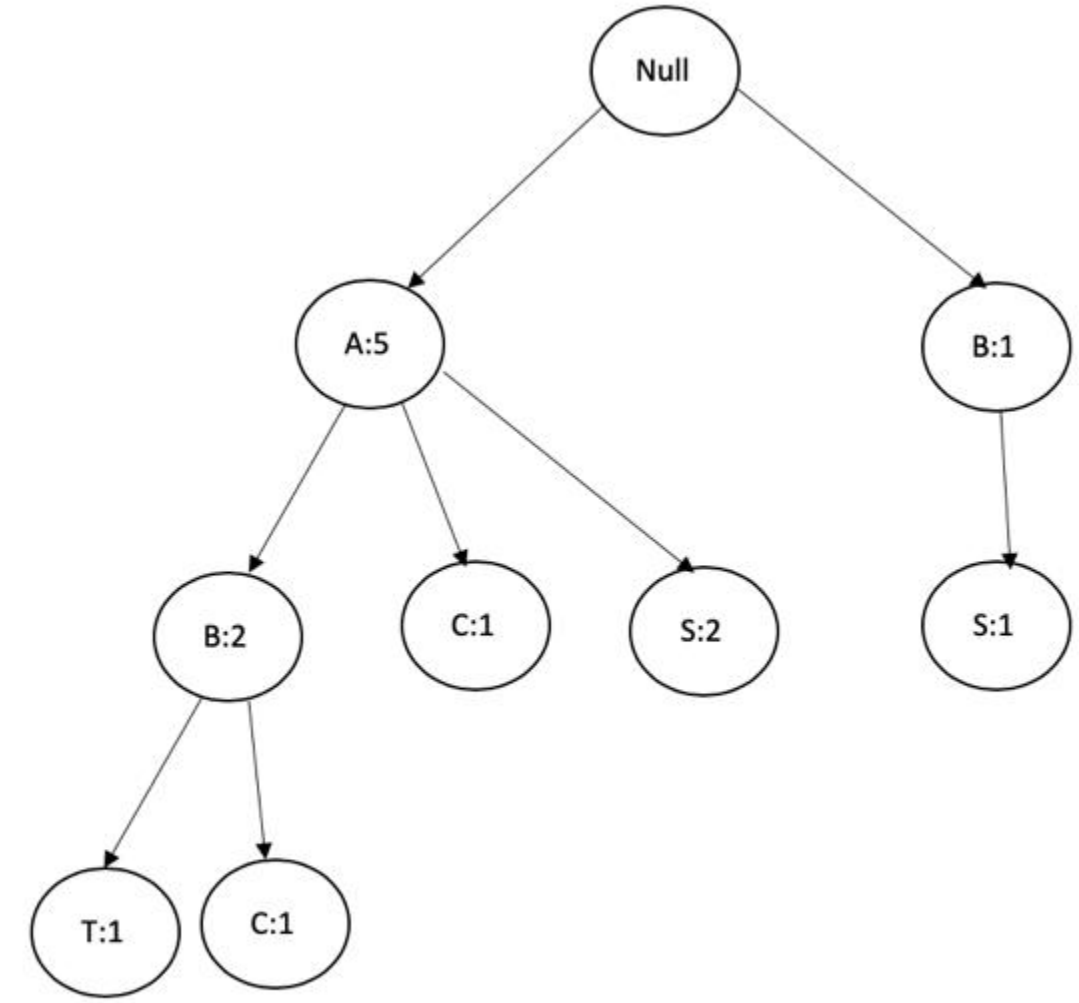


Figure 6 : Transaction AS

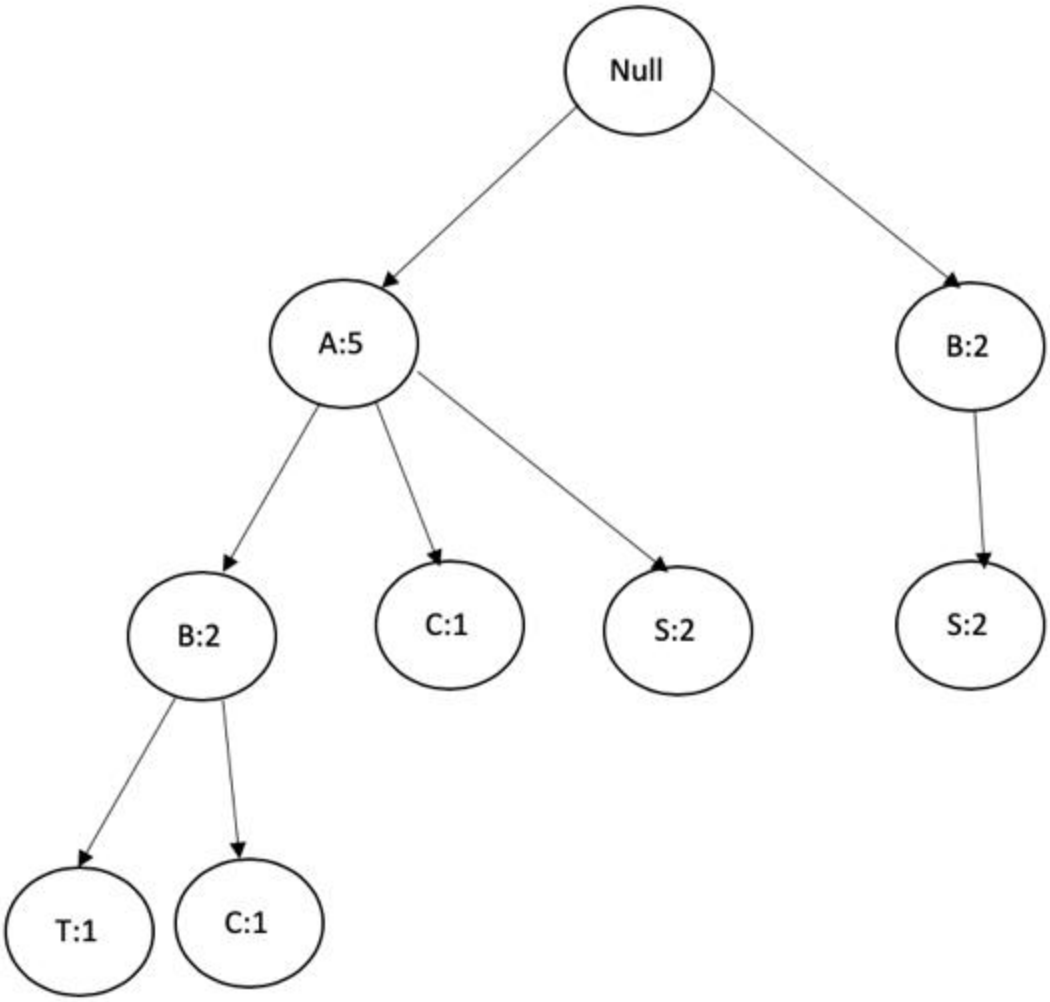


Figure 7 : Transaction BS

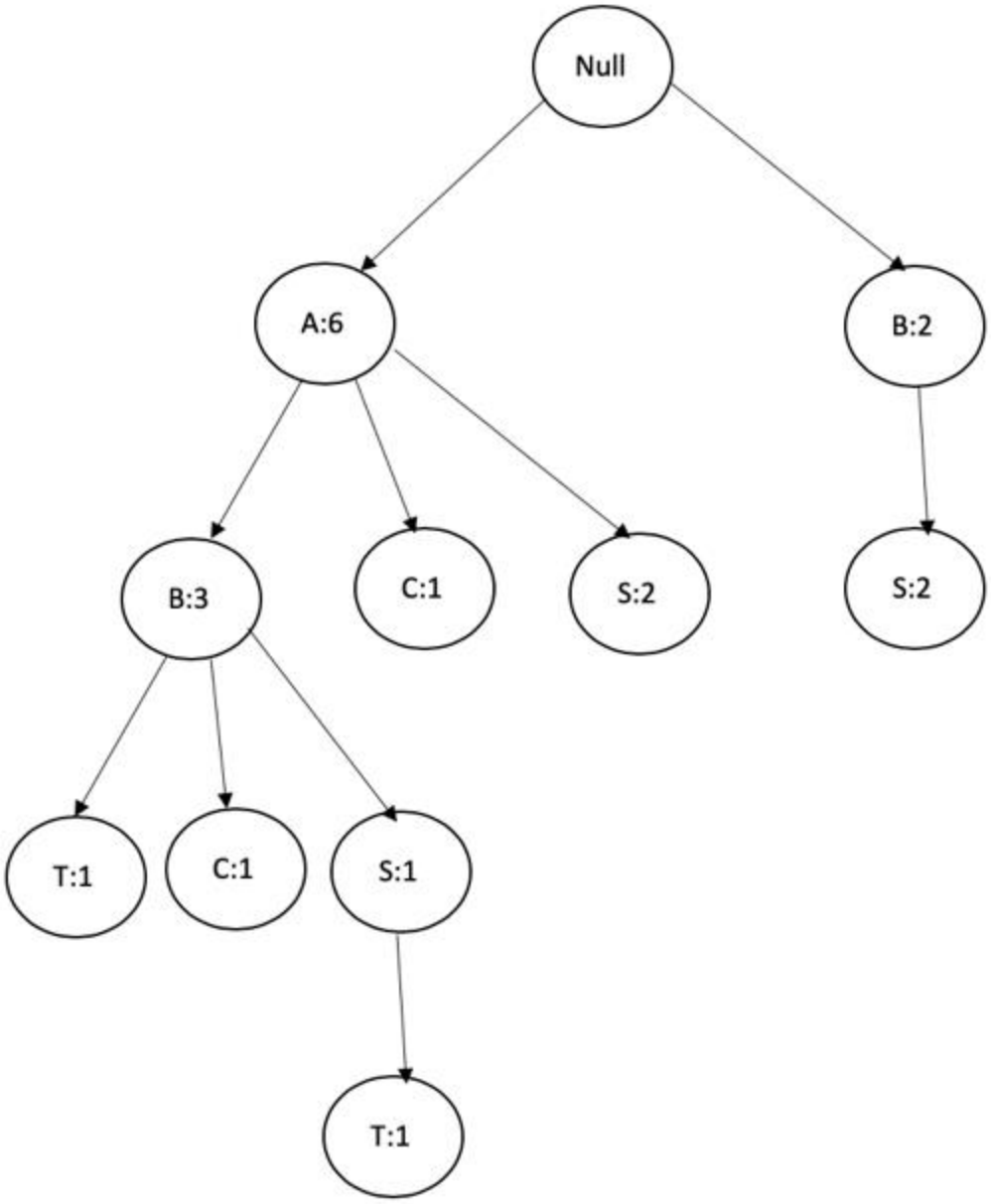


Figure 8 : Transaction ABST

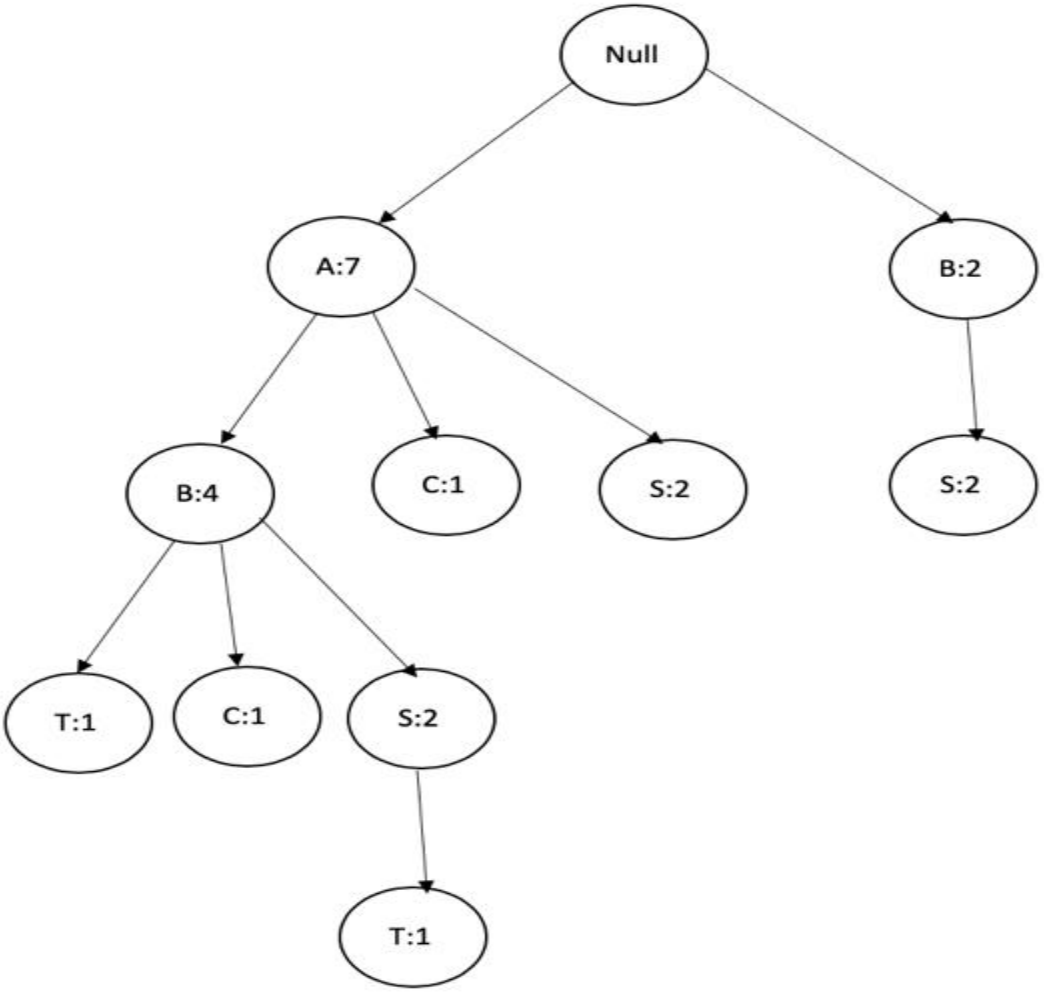


Figure 9 : Transaction ABS

20)Write the differences between Hierarchical and Non-Hierarchical Clustering?

Difference between Hierarchical Clustering and Non Hierarchical Clustering:

S.NO.	Hierarchical Clustering:	Non Hierarchical Clustering:
1.	Hierarchical Clustering involves creating clusters in a predefined order from top to bottom .	Non Hierarchical Clustering involves formation of new clusters by merging or splitting the clusters instead of following a hierarchical order.
2.	It is considered less reliable than Non Hierarchical Clustering.	It is comparatively more reliable than Hierarchical Clustering.
3.	It is considered slower than Non Hierarchical Clustering.	It is comparatavely more faster than Hierarchical Clustering.
4.	It is very problematic to apply this technique when we have data with high level of error.	It can work better then Hierarchical clustering even when error is there.
5.	It is comparatively easier to read and understand.	The clusters are difficult to read and understand as compared to Hierarchical clustering.
	It is relatively unstable	It is a relatively stable technique.

UNIT-V GENETIC ALGORITHM

TWO Marks Questions

1)Define Genetic Algorithm in ML?

A genetic algorithm is an adaptive heuristic search algorithm inspired by "Darwin's theory of evolution in Nature." It is used to solve optimization problems in machine learning. Genetic Algorithms are being widely used in different real-world applications, for example, **Designing**

It is a type of reinforcement learning where the feedback is necessary without telling the correct path to follow. The feedback can either be positive or negative.

2)Why Use Genetic Algorithms?

GAs are more robust algorithms that can be used for various optimization problems. These algorithms do not deviate easily in the presence of noise, unlike other AI algorithms. GAs can be used in the search for large space or multimodal space.

3) How Genetic Algorithm Work?

The genetic algorithm works on the evolutionary generational cycle to generate high-quality solutions. These algorithms use different operations that either enhance or replace the population to give an improved fit solution.

It basically involves five phases to solve the complex optimization problems, which are given as below:

- Initialization
- Fitness Assignment
- Selection
- Reproduction
- Termination

4)Write any 3 advantages of Genetic algorithm?

- The parallel capabilities of genetic algorithms are best.
- It helps in optimizing various problems such as discrete functions, multi-objective problems, and continuous functions.

- It provides a solution for a problem that improves over time.
-

5)Define Hypothesis?

A Hypothesis is an assumption of a result that is falsifiable, meaning it can be proven wrong by some evidence. A Hypothesis can be either rejected or failed to be rejected. We never accept any hypothesis in statistics because it is all about probabilities and we are never 100% certain.

6) Define two hypotheses?

1.Null Hypothesis: says that there is no significant effect.
2. Alternative Hypothesis: says that there is some significant effect.

7) What is Hypothesis in ML?

Data scientists and ML professionals conduct experiments that aim to solve a problem. These ML professionals and data scientists make an initial assumption for the solution of the problem.

This assumption in Machine learning is known as Hypothesis.

It is specifically used in Supervised Machine learning, where an ML model learns a function that best maps the input to corresponding outputs with the help of an available dataset.

8) Define Genetic Operators?

A genetic operator is an operator used in genetic algorithms to guide the algorithm towards a solution to a given problem. There are three main types of operators (mutation, crossover and selection), which must work in conjunction with one another in order for the algorithm to be successful.

FIVE Marks and Ten Marks Questions

9)What are the Applications of genetic algorithms?

Applications Of Genetic Algorithms

GA is effective to solve high dimensional problems. A GA is effectively used when the search space is very large, there are no mathematical problem-solving techniques available and other traditional search algorithms do not work.

Some applications where GA is used:

- **Optimization Problem:** One of the best examples of the optimization problems is the travel salesman problem which uses GA. Other optimization problems such as job scheduling, sound quality optimization GAs are widely used.
- **Immune system model:** GAs are used to model various aspects of the immune system for individual gene and multi-gene families during evolutionary time.
- **Machine Learning:** GAs have been used to solve problem-related to classification, prediction, create rules for learning and classification**.**

10) How Genetic Algorithm Work?

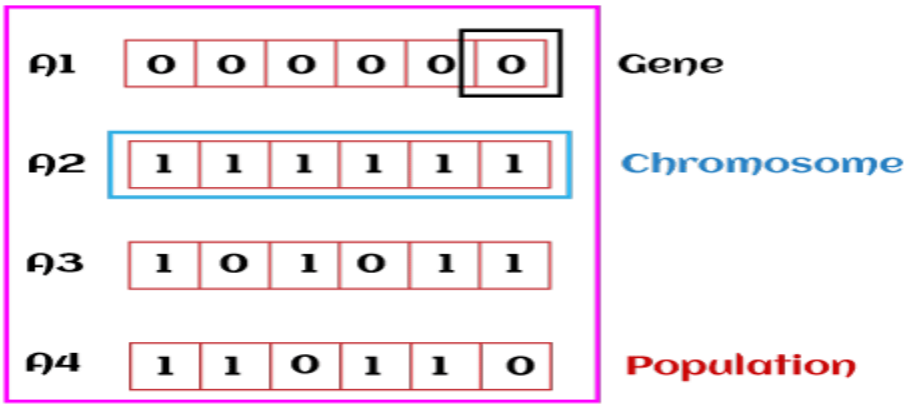
The genetic algorithm works on the evolutionary generational cycle to generate high-quality solutions. These algorithms use different operations that either enhance or replace the population to give an improved fit solution.

It basically involves five phases to solve the complex optimization problems, which are given as below:

- **Initialization**
- **Fitness Assignment**
- **Selection**
- **Reproduction**
- **Termination**

1. Initialization

The process of a genetic algorithm starts by generating the set of individuals, which is called population. Here each individual is the solution for the given problem. An individual contains or is characterized by a set of parameters called Genes. Genes are combined into a string and generate chromosomes, which is the solution to the problem. One of the most popular techniques for initialization is the use of random binary strings.



2. Fitness Assignment

Fitness function is used to determine how fit an individual is? It means the ability of an individual to compete with other individuals. In every iteration, individuals are evaluated based on their fitness function. The fitness function provides a fitness score to each individual. This score further determines the probability of being selected for reproduction. The high the fitness score, the more chances of getting selected for reproduction.

3. Selection

The selection phase involves the selection of individuals for the reproduction of offspring. All the selected individuals are then arranged in a pair of two to increase reproduction. Then these individuals transfer their genes to the next generation.

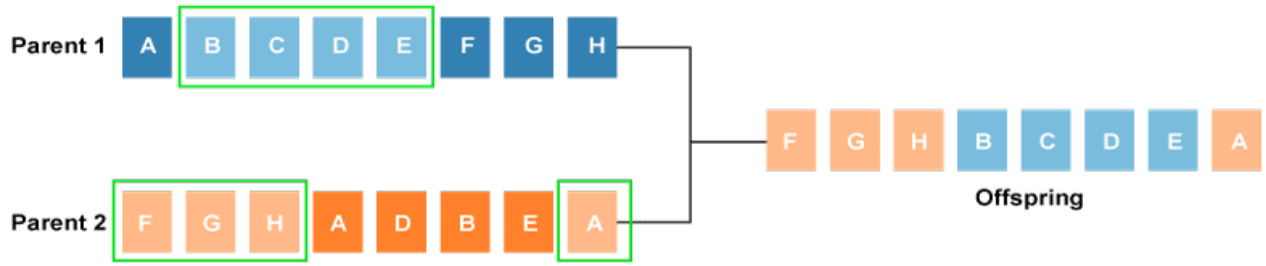
There are three types of Selection methods available, which are:

- Roulette wheel selection
- Tournament selection
- Rank-based selection

4. Reproduction

After the selection process, the creation of a child occurs in the reproduction step. In this step, the genetic algorithm uses two variation operators that are applied to the parent population. The two operators involved in the reproduction phase are given below:

- **Crossover:** The crossover plays a most significant role in the reproduction phase of the genetic algorithm. In this process, a crossover point is selected at random within the genes. Then the crossover operator swaps genetic information of two parents from the current generation to produce a new individual representing the offspring.



- The genes of parents are exchanged among themselves until the crossover point is met. These newly generated offspring are added to the population. This process is also called or crossover. Types of crossover styles available:
    - One point crossover
    - Two-point crossover
    - Livery crossover
    - Inheritable Algorithms crossover
- **Mutation**

  The mutation operator inserts random genes in the offspring (new child) to maintain the diversity in the population. It can be done by flipping some bits in the chromosomes. Mutation helps in solving the issue of premature convergence and enhances diversification. The below image shows the mutation process:
  Types of mutation styles available,
    - **Flip bit mutation**
    - **Gaussian mutation**
    - **Exchange/Swap mutation**

5. Termination

After the reproduction phase, a stopping criterion is applied as a base for termination. The algorithm terminates after the threshold fitness solution is reached. It will identify the final solution as the best solution in the population.

11)**Explain Clustering and its types?**

In machine learning clustering is the process by which we create groups in a data, like customers, products, employees, text documents, in such a way that objects falling into one group exhibit many similar properties with each other and are different from objects that fall in the other groups that got created during the process.

The various types of clustering are:

1. **Connectivity-based Clustering (Hierarchical clustering)**
2. **Centroids-based Clustering (Partitioning methods)**
3. **Distribution-based Clustering**
4. **Density-based Clustering (Model-based methods)**
5. **Fuzzy Clustering**
6. **Constraint-based (Supervised Clustering)**

1. Connectivity-Based Clustering (Hierarchical Clustering)

Hierarchical Clustering is a method of unsupervised machine learning clustering where it begins with a pre-defined top to bottom hierarchy of clusters. It then proceeds to perform a decomposition of the data objects based on this hierarchy, hence obtaining the clusters.

These are Divisive Approach and the Agglomerative Approach

2. Centroid Based Clustering

Centroid based clustering is considered as one of the most simplest clustering algorithms, yet the most effective way of creating clusters and assigning data points to it. The intuition behind centroid based clustering is that a cluster is characterized and represented by a central vector and data points that are in close proximity to these vectors are assigned to the respective clusters.

3. Density-based Clustering (Model-based Methods)

Density-based clustering methods take density into consideration instead of distances. Clusters are considered as the densest region in a data space, which is separated by regions of lower object density and it is defined as a maximal-set of connected points.

4. Distribution-Based Clustering

Distribution-based clustering creates and groups data points based on their likely hood of belonging to the same probability distribution (Gaussian, Binomial etc.) in the data.

The distribution models of clustering are most closely related to statistics as it very closely relates to the way how datasets are generated and arranged using random sampling principles i.e., to fetch data points from one form of distribution.

5. Fuzzy Clustering

Fuzzy clustering methods assign a data-point to multiple clusters with a quantified degree of belongingness metric. The data-points that are in proximity to the center of a cluster, may also belong in the cluster that is at a higher degree than points in the edge of a cluster. The possibility of which an element belongs to a given cluster is measured by membership coefficient that vary from 0 to 1.

Fuzzy clustering can be used with datasets where the variables have a high level of overlap. It is a strongly preferred algorithm for Image Segmentation, especially in bioinformatics where identifying overlapping gene codes makes it difficult for generic clustering algorithms to differentiate between the image’s pixels and they fail to perform a proper clustering.

6. Constraint-based (Supervised Clustering)

A constraint is defined as the desired properties of the clustering results, or a user's expectation on the clusters so formed – this can be in terms of a fixed number of clusters, or, the cluster size, or, important dimensions (variables) that are required for the clustering process.

12)What is a Decision Tree and its types?

A decision tree is a support tool with a tree-like structure that models probable outcomes, cost of resources, utilities, and possible consequences. Decision trees provide a way to present algorithmsAlgorithms (Algos)Algorithms (Algos) are a set of instructions that are introduced to perform a task. They automate trading to generate profits at a frequency impossible to a human trader. with conditional control statements. They include branches that represent decision-making steps that can lead to a favorable result.

Types of Decisions

There are two main types of decision trees that are based on the target variable, i.e., categorical variable decision trees and continuous variable decision trees.

1. Categorical variable decision tree

A categorical variable decision tree includes categorical target variables that are divided into categories. For example, the categories can be yes or no. The categories mean that every stage of the decision process falls into one category, and there are no in-betweens.

2. Continuous variable decision tree

A continuous variable decision tree is a decision tree with a continuous target variable. For example, the income of an individual whose income is unknown can be predicted based on available information such as their occupation, age, and other continuous variables.

13)What are the applications of Decision Trees

1. Assessing prospective growth opportunities

One of the applications of decision trees involves evaluating prospective growth opportunities for businesses based on historical data. Historical data on sales can be used in decision trees that may lead to making radical changes in the strategy of a business to help aid expansion and growth.

2. Using demographic data to find prospective clients

Another application of decision trees is in the use of demographic data.Demographics refer to the socio-economic characteristics of a population that businesses use to identify the product preferences and to find prospective clients. They can help streamline a marketing budget and make informed decisions on the

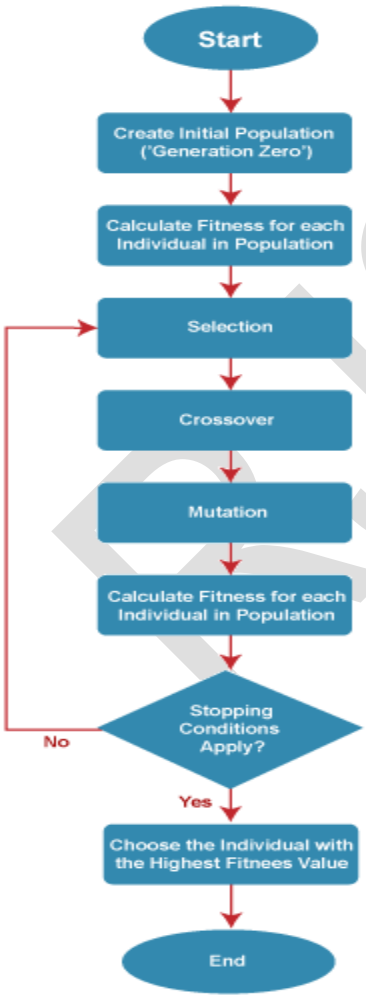
target market that the business is focused on. In the absence of decision trees, the business may spend its marketing market without a specific demographic in mind, which will affect its overall revenues.

3. Serving as a support tool in several fields

Lenders also use decision trees to predict the probability of a customer defaulting on a loan by applying predictive model generation using the client's past data. The use of a decision tree support tool can help lenders evaluate a customer's creditworthiness to prevent losses.

Decision trees can also be used in operations research in planning logistics and strategic managementStrategic ManagementStrategic management is the formulation and implementation of major goals and initiatives taken by an organization's top management on behalf of its. They can help in determining appropriate strategies that will help a company achieve its intended goals. Other fields where decision trees can be applied include engineering, education, law, business, healthcare, and finance.

14) Explain the General Workflow of GA?



15) Write the differences between Genetic and Traditional Algorithms

Difference between Genetic Algorithms and Traditional Algorithms

- A search space is the set of all possible solutions to the problem. In the traditional algorithm, only one set of solutions is maintained, whereas, in a genetic algorithm, several sets of solutions in search space can be used.
- Traditional algorithms need more information in order to perform a search, whereas genetic algorithms need only one objective function to calculate the fitness of an individual.
- Traditional Algorithms cannot work parallelly, whereas genetic Algorithms can work parallelly (calculating the fitness of the individualities are independent).
- One big difference in genetic Algorithms is that rather of operating directly on seeker results, inheritable algorithms operate on their representations (or rendering), frequently appertained to as chromosomes.
- One of the big differences between traditional algorithm and genetic algorithm is that it does not directly operate on candidate solutions.
- Traditional Algorithms can only generate one result in the end, whereas Genetic Algorithms can generate multiple optimal results from different generations.
- The traditional algorithm is not more likely to generate optimal results, whereas Genetic algorithms do not guarantee to generate optimal global results, but also there is a great possibility of getting the optimal result for a problem as it uses genetic operators such as Crossover and Mutation.
- Traditional algorithms are deterministic in nature, whereas Genetic algorithms are probabilistic and stochastic in nature.