SUPERVISED LEARNING

Multiple Choice Type Questions

1. What is supervised learning? [MODEL QUESTION]
   a) All data is unlabelled and the algorithms learn to inherent structure from the input data
   b) All data is labelled and the algorithms learn to predict the output from the input data
   c) It is a framework for learning where an agent interacts with an environment and receives a reward for each interaction
   d) Some data is labelled but most of it is unlabelled and a mixture of supervised and unsupervised techniques can be used

Answer: (b)

2. A regression model in which more than one independent variable is used to predict the dependent variable is called [MODEL QUESTION]
   a) a simple linear regression model
   b) a multiple regression models
   c) an independent model
   d) none of these

Answer: (c)

3. A nearest neighbour approach is best used [MODEL QUESTION]
   a) with large-sized datasets
   b) when irrelevant attributes have been removed from the data
   c) when a generalized model of the data is desirable
   d) when an explanation of what has been found is of primary importance

Answer: (b)

4. Logistic regression is a _____ regression technique that is used to model data having a _____outcome. [MODEL QUESTION]
   a) linear, numeric
   b) linear, binary
   c) non-linear, numeric
   d) non-linear, binary

Answer: (d)

5. This supervised learning technique can process both numeric and categorical input attributes. [MODEL QUESTION]
   a) linear regression
   b) Bayes classifier
   c) logistic regression
   d) backpropagation learning

Answer: (a)

6. Machine learning techniques differ from statistical techniques in that machine learning methods [MODEL QUESTION]
   a) typically assume an underlying distribution for the data
   b) are better able to deal with missing and noisy data
   c) are not able to explain their behaviour
   d) have trouble with large-sized datasets

Answer: (b)

Scanned by TapScanner

7. Which of the following methods do we use to best fit the data in Logistic Regression? [MODEL QUESTION]
   a) Least Square Error
   b) Maximum Likelihood
   c) Jaccard distance
   d) Both (a) & (b)

Answer: (b)

8. Choose which of the following options is true regarding One-vs-All method in Logistic Regression. [MODEL QUESTION]
   a) We need to fit n models in n-class classification problem
   b) We need to fit n-1 models to classify into n classes
   c) We need to fit only 1 model to classify into n classes
   d) None of these

Answer: (a)

9. Suppose, You applied a Logistic Regression model on a given data and got a training accuracy X and testing accuracy Y. Now, you want to add a few new features in the same data. Select the option(s) which is/are correct in such a case. Note: Consider remaining parameters are same. [MODEL QUESTION]
   a) Training accuracy increases
   b) Training accuracy increases or remains the same
   c) Testing accuracy decreases
   d) Testing accuracy increases or remains the same

Answer: (a) and (d)

10. Which of the following statement is true about outliers in Linear regression? [MODEL QUESTION]
   a) Linear regression is sensitive to outliers
   b) Linear regression is not sensitive to outliers
   c) Can't say
   d) None of these

Answer: (a)

11. Which of the following option is true? [MODEL QUESTION]
   a) Linear Regression errors values has to be normally distributed but in case of Logistic Regression it is not the case
   b) Logistic Regression errors values has to be normally distributed but in case of Linear Regression it is not the case
   c) Both Linear Regression and Logistic Regression error values have to be normally distributed
   d) Both Linear Regression and Logistic Regression error values have not to be normally distributed

Answer: (a)

12. The SVM's are less effective when: [MODEL QUESTION]
   a) The data is linearly separable
   b) The data is clean and ready to use
   c) The data is noisy and contains overlapping points
   d) None of these

Answer: (c)

Scanned by TapScanner

13. The cost parameter in the SVM means: [MODEL QUESTION]
   a) The number of cross-validations to be made
   b) The kernel to be used
   c) The trade-off between misclassification and simplicity of the model
   d) None of the above

Answer: (c)

14. Which of the following are real world applications of the SVM? [MODEL QUESTION]
   a) Text and Hypertext Categorization
   b) Image Classification
   c) Clustering of News Articles
   d) All of these

Answer: (d)

15. Which of the following is true about Naive Bayes? [MODEL QUESTION]
   a) Assumes that all the features in a dataset are equally important
   b) Assumes that all the features in a dataset are independent
   c) Both A and B – answer
   d) None of the above

Answer: (c)

16. What do you mean by generalization error in terms of the SVM? [MODEL QUESTION]
   a) How far the hyperplane is from the support vectors
   b) How accurately the SVM can predict outcomes for unseen data
   c) The threshold amount of error in an SVM
   d) None of the above

Answer: (b)

17. Which of the following is a disadvantage of decision trees? [MODEL QUESTION]
   a) Factor analysis
   b) Decision trees are robust to outliers
   c) Decision trees are prone to be overfit
   d) None of the above

Answer: (c)

18. You run gradient descent for 15 iterations with a=0.3 and compute J(theta) after each iteration. You find that the value of J (Theta) decreases quickly and then levels off. Based on this, which of the following conclusions seems most plausible? [MODEL QUESTION]
   a) Rather than using the current value of a, use a larger value of a (say a=1.0)
   b) Rather than using the current value of a, use a smaller value of a (say a=0.1)
   c) a=0.3 is an effective choice of learning rate
   d) None of the above

Answer: (c)

19. In linear regression, we try to _____ the least square errors of the model to identify the line of best fit. [MODEL QUESTION]
   a) Minimize
   b) Maximize
   c) Change
   d) none of these

Answer: (a)

Scanned by TapScanner

20. We can compute the coefficient of linear regression by using [MODEL QUESTION]

a) gradient descent
b) Normal Equation
c) both (a) & (b)
d) None of these

Answer: (c)

21. What's the objective of the support vector machine algorithm? [MODEL QUESTION]

a) to find an optimal hyperplane in an N-dimensional space that distinctly classifies the data points where N is the number of features
b) to find an optimal hyperplane in an N-dimensional space that distinctly classifies the data points where N is the number of samples
c) to find an optimal hyperplane in an N-dimensional space that distinctly classifies the data points where N is the number of target variables
d) None of these

Answer: (d)

22. Which options are true for SVM? [MODEL QUESTION]

a) The distance of the vectors from the margin is called the hyperplane
b) The loss function that helps minimize the margin is called hinge loss
c) SVM can solve the linearly separable data points
d) SVM can solve the data points that are not linearly separable

Answer: (d)

23. For SVM, which options are correct? [MODEL QUESTION]

a) Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane
b) Support vectors are data points that are far away from the hyperplane and influence the position and orientation of the hyperplane
c) Deleting the support vectors will change the position of the hyperplane
d) Deleting the support vectors won't change the position of the hyperplane

Answer: (d)

24. Decision tree is the most powerful for _____ [MODEL QUESTION]

a) classification b) prediction c) both (a) & (b) d) none of these

Answer: (c)

25. Which one of the following statements is TRUE for a Decision Tree? [MODEL QUESTION]

a) Decision tree is only suitable for the classification problem statement.
b) In a decision tree, the entropy of a node decreases as we go down a decision tree.
c) In a decision tree, entropy determines purity.
d) Decision tree can only be used for only numeric valued and continuous attributes.

Answer: (b)

Scanned by TapScanner

26. How do you choose the right node while constructing a decision tree? [MODEL QUESTION]

a) An attribute having high entropy
b) An attribute having high entropy and information gain
c) An attribute having the lowest information gain
d) An attribute having the highest information gain

Answer: (d)

27. Which of the following statement is False in the case of the KNN Algorithm? [MODEL QUESTION]

a) For a very large value of K, points from other classes may be included in the neighbourhood.
b) For the very small value of K, the algorithm is very sensitive to noise.
c) KNN is used only for classification problem statements.
d) KNN is a lazy learner.

Answer: (c)

28. In the Naive Bayes algorithm, suppose that prior for class w1 is greater than class w2, would the decision boundary shift towards the region R1(region for deciding w1) or towards region R2 (region for deciding w2)? [MODEL QUESTION]

a) towards region R1
b) towards region R2
c) No shift in decision boundary
d) It depends on the exact value of priors

Answer: (b)

Short Answer Type Questions

1. How machine learning comes into picture? [MODEL QUESTION]

Answer:

With the advancements in technology, data collection has become easy. When you turn on location in your mobile, upload your pictures on Facebook or Instagram, fill online forms, browse websites, or even order items from an e-commerce website, your data is collected. What do companies do with this huge data? They analyze it, find your preferences, and this helps them in marketing. The advertisements being shown to you, generally, depending on the above things. Marketing professionals must lure you into buying something that you need or are even remotely interested in. Your data helps them. Likewise, the dispensation may keep track of suspicious activities using this data, may track the source of transactions, or gather other important information using this data. However, this is easier said than done. It is a huge data and its analysis cannot be done using conventional methods.

Let us consider another example to understand this. Suppose Hari visit YouTube everyday and watches videos related to Indian Classical Music, Hindi Poetry and watch Lizzie McGuire. His friend Tarush goes to YouTube and watches Beer Biceps and other videos related to workouts. After some time YouTube starts suggesting different relevant videos to both of them. While Hari is shown a video related to Lizzie McGuire's reboot

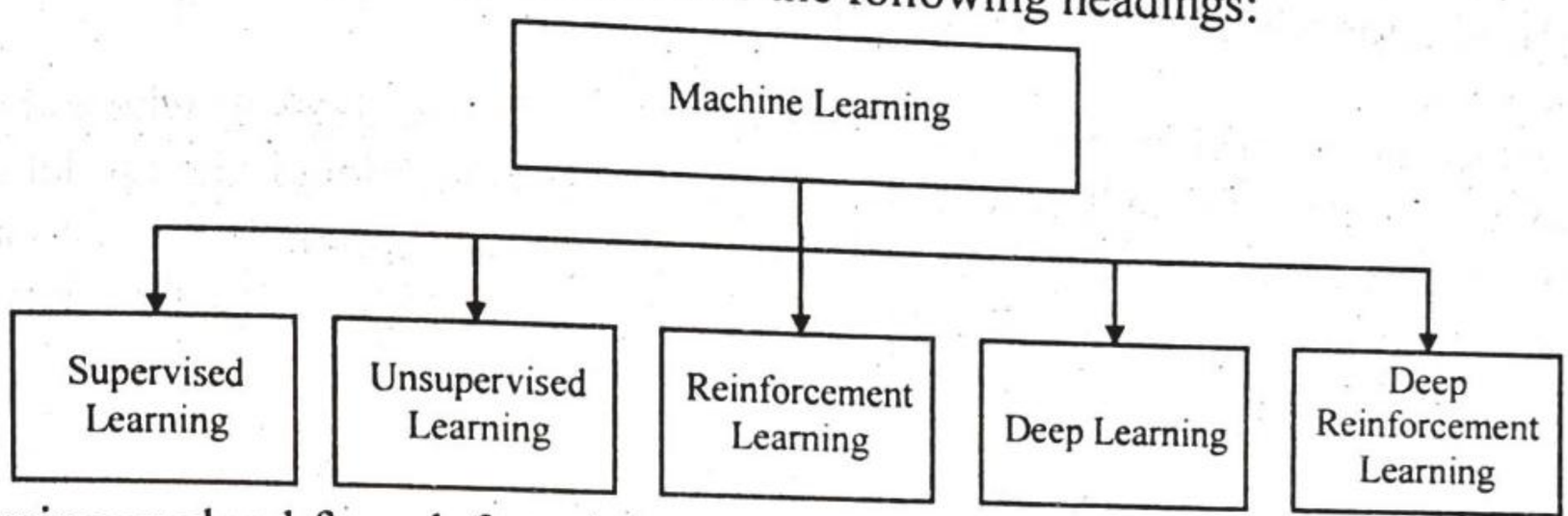
Scanned by TapScanner

or Dinkar, in the recommended videos' list, Tarush is not recommended any such video. On the other hand, Tarush is shown a recommendation for a workout video.
It may be stated that recommendation requires an in-depth analysis and cannot be done solely based on any conventional algorithms. Those using e-commerce websites or famous music streaming apps like YouTube must be knowing that the recommendations are mostly goods, if not excellent. Here the task is prediction. Your browsing history help in this task, and for sure, it cannot be accomplished by conventional algorithms. Moreover, the betterment in the output, with time, means there is a well-defined performance measure for the task.
Machine learning comes to the rescue of those wanting to analyze this huge data, predict trends, find patterns, and so on.

2. What are the categories of machine learning? **[MODEL QUESTION]**

Answer:

Machine Learning is broadly categorized under the following headings:



Machine learning evolved from left to right as shown in the above diagram.

- Initially, researchers started out with Supervised learning. This is the case of housing price prediction discussed earlier.
- This was followed by unsupervised learning, where the machine is made to learn on its own without any supervision.
- Scientists discovered further it may be a good idea to reward the machine when it does the job the expected way and there came the Reinforcement learning.
- Very soon, the data that is available these days has become so humongous that the conventional techniques developed so far failed to analyze the big data and provide us the predictions.
- Thus, came the deep learning where the human brain is simulated in the Artificial Neural Networks (ANN) created in our binary computers.
- The machine now learns on its own using the high computing power and huge memory resources that are available today.
- It is now observed the Deep Learning has solved many of the previously unsolvable problems.
- The technique is now further advanced by giving incentives to Deep Learning networks as awards and there finally comes Deep Reinforcement Learning.

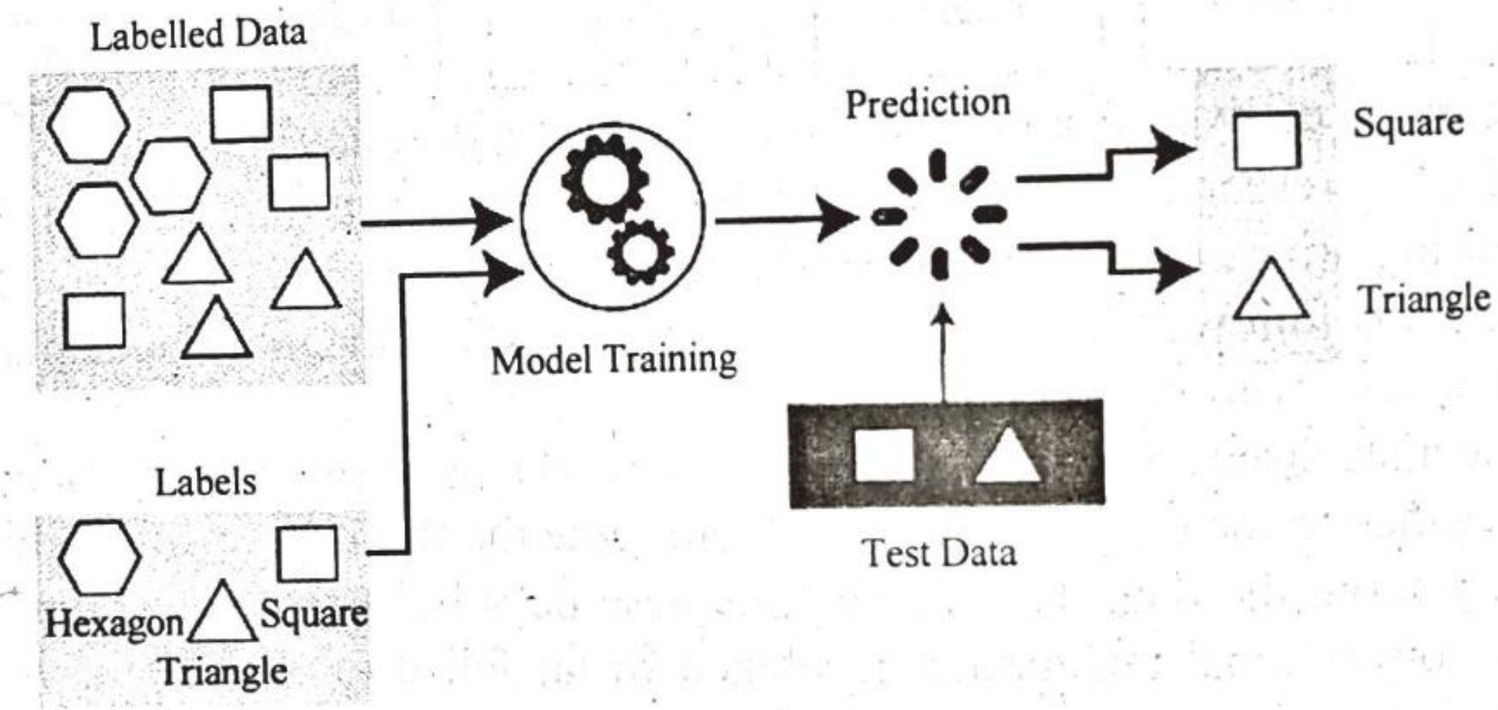
Scanned by TapScanner

3. What is supervised machine learning? **[MODEL QUESTION]**

Answer:
Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.
In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.
Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y).**
In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.

4. How supervised machine learning works? **[MODEL QUESTION]**

Answer:
In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.
The working of Supervised learning can be easily understood by the below example and diagram:



Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

- If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square.
- If the given shape has three sides, then it will be labelled as a triangle.
- If the given shape has six equal sides then it will be labelled as hexagon.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.
The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

Scanned by TapScanner

5. What are steps involved in supervised learning? **[MODEL QUESTION]**

Answer:

- First Determine the type of training dataset.
- Collect/Gather the labelled training data.
- Split the training dataset into training **dataset, test dataset, and validation dataset.**
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

6. What are advantages and disadvantages of supervised learning? **[MODEL QUESTION]**

Answer:

Advantages of supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

7. What are different types of supervised learning? **[MODEL QUESTION]**

Answer:

Types of Supervised Learning Algorithms

- **Classification:** In these types of problems, we predict the response as specific classes, such as "yes" or "no". When only 2 classes are present, then it is called a Binary Classification. For more than 2 class values, it is called a Multi-class Classification. The predicted response values are discrete values. **For example,** Is it the image of the sun or the moon? The classification algorithm separates the data into classes.
- **Regression:** Regression problems predict the response as continuous values such as predicting a value that ranges from -infinity to infinity. It may take many values. **For example,** the linear regression algorithm that is applied, predicts the cost of the house based on many parameters such as location, nearby airport, size of the house, etc.

Scanned by TapScanner

It is very important to understand which Machine Learning algorithm should be applied depending upon the type of problem we are dealing with. The below table consists of different types of Machine Learning problems and the possible algorithms for solving those problems.

ML Type	ML Algorithm
Classification	Naive Bayes
	Logistic Regression
	Support Vector Machines (SVMs)
Regression	Linear Regression
Both	Decision Trees/Random Forest
	k-Nearest Neighbours
	Gradient Boosting Algorithms

8. What are applications of supervised learning? **[MODEL QUESTION]**

Answer:

Supervised Learning Algorithms are used in a variety of applications. Let's go through some of the most well-known applications.

- **BioInformatics** – This is one of the most well-known applications of Supervised Learning because most of us use it in our day-to-day lives. BioInformatics is the storage of Biological Information of us humans such as fingerprints, iris texture, earlobe and so on. Cellphones of today are capable of learning our biological information and are then able to authenticate us bringing up the security of the system. Smartphones such as iPhones, Google Pixel are capable of facial recognition while OnePlus, Samsung is capable of In-display finger recognition.
- **Speech Recognition** – This is the kind of application where you teach the algorithm about your voice and it will be able to recognize you. The most well-known real-world applications are virtual assistants such as Google Assistant and Siri, which will wake up to the keyword with your voice only.
- **Spam Detection** – This application is used where the unreal or computer-based messages and E-Mails are to be blocked. G-Mail has an algorithm that learns the different keywords which could be fake such as "You are the winner of something" and so forth and blocks those messages directly. OnePlus Messages App gives the user the task of making the application learn which keywords need to be blocked and the app will block those messages with the keyword.
- **Object-Recognition for Vision** – This kind of application is used when you need to identify something. You have a huge dataset which you use to teach your algorithm and this can be used to recognize a new instance. Raspberry Pi algorithms which detect objects are the most well-known example.

9. What is a regression? **[MODEL QUESTION]**

Answer:

In Regression, we plot a graph between the variables which best fit the given data points. The machine learning model can deliver predictions regarding the data. In naïve

Scanned by TapScanner

words, "Regression shows a line or curve that passes through all the data points on a target-predictor graph in such a way that the vertical distance between the data points and the regression line is minimum." It is used principally for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

10. What do you understand by logistic regression? [MODEL QUESTION]

Answer:

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

11. What is Sigmoid Function? [MODEL QUESTION]

Answer:

The sigmoid function is represented by the symbol sigma. Its graphical behaviour has been described in the above figure. The mathematical equation for the sigmoid function is described as follows:

$$\hat{y} = \sigma\left(\theta^T \cdot X\right)$$

where, $\sigma\left(\theta^T \cdot X\right) = \dfrac{1}{1 + e^{-\theta^T \cdot X}}$

The sigmoid function gives out the probability of data belonging to a particular class which lies in the interval [0, 1]. It accepts the dot product of transpose of theta and feature vector X as the parameter. The resulting value is the probability.

Hence when $P\left(Y = 1 \mid X\right) = \text{sigmoid}\left(\text{theta} * X\right)$

$P\left(Y = 0 \mid X\right) = 1 - \text{sigmoid}\left(\text{theta} * X\right)$

Below are the behaviours of the sigmoid function:

- When theta(transpose)$*X$ becomes very bigger the value of the sigmoid becomes equal to 1
- When theta(transpose) $*X$ becomes very small the value of the sigmoid becomes equal to 0

12. How logical regression is different from linear regression? [MODEL QUESTION]

Answer:

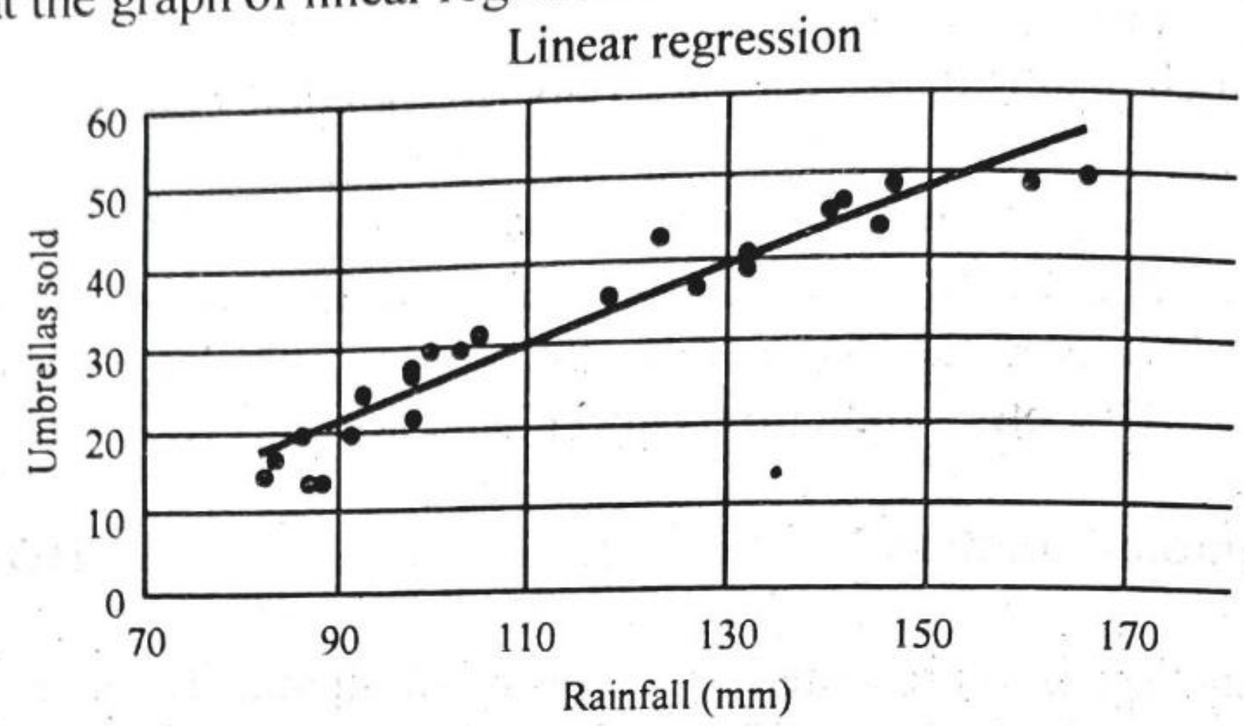
Let me tell you that there is a big difference between both linear regression and logistic regression. Linear regression is used for generating continuous values like the price of the house, income, population, etc. In logistic regression, we generally compute the probability which lies between the interval 0 and 1 (inclusive of both). Then probability can be used to classify the data. For example, if the computed probability comes out to

Scanned by TapScanner

be greater than 0.5, then the data belonged to class A and otherwise, for less than 0.5, the data belonged to class B.

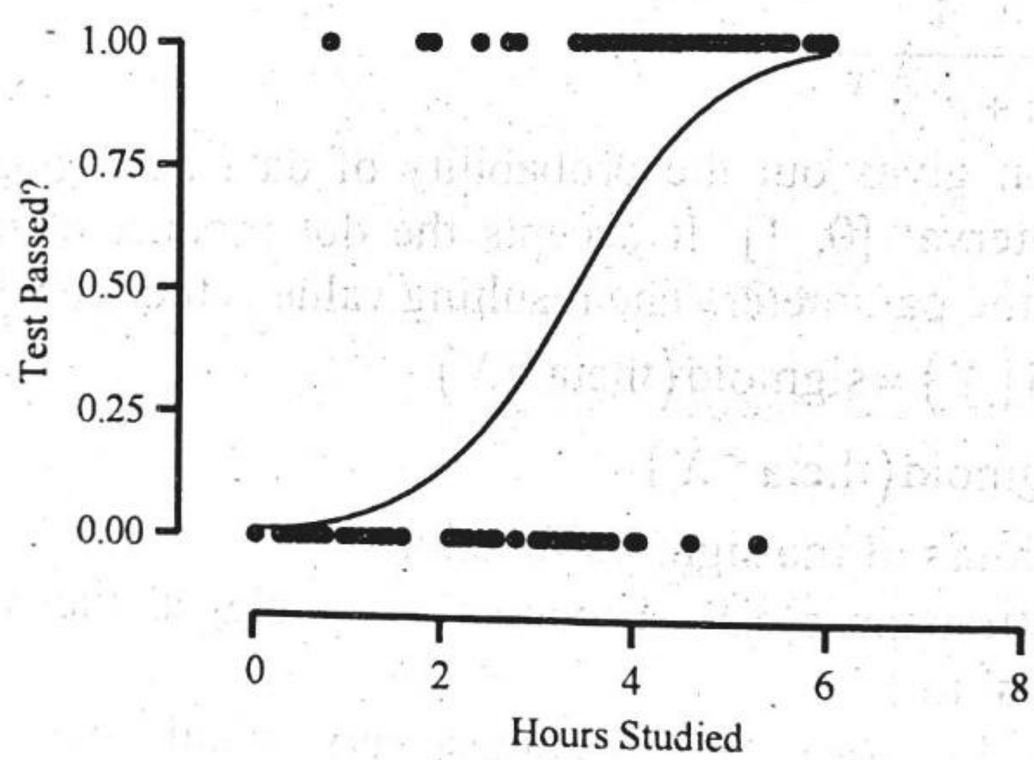
But question is that can we still use linear regression for classification? Answer will be "Yes! why not? But sure it is an absurd idea." The reason will be that we can assign a threshold value for linear regression, that is if the predicted value is greater than the threshold value, it belonged to class A otherwise class B. But it will give out a large error and a poor model with low accuracy, which we really don't want.

Now let's look at the graph of linear regression shown below



The graph is a straight line passing through some points since we always avoid underfitting and overfitting curves.

Now let's have a look at the logistic regression graph:



The graph is a curved line instead of the straight line, unlike linear regression.

This is a major difference between the two types of regression that we just talked about.

So my next question is.

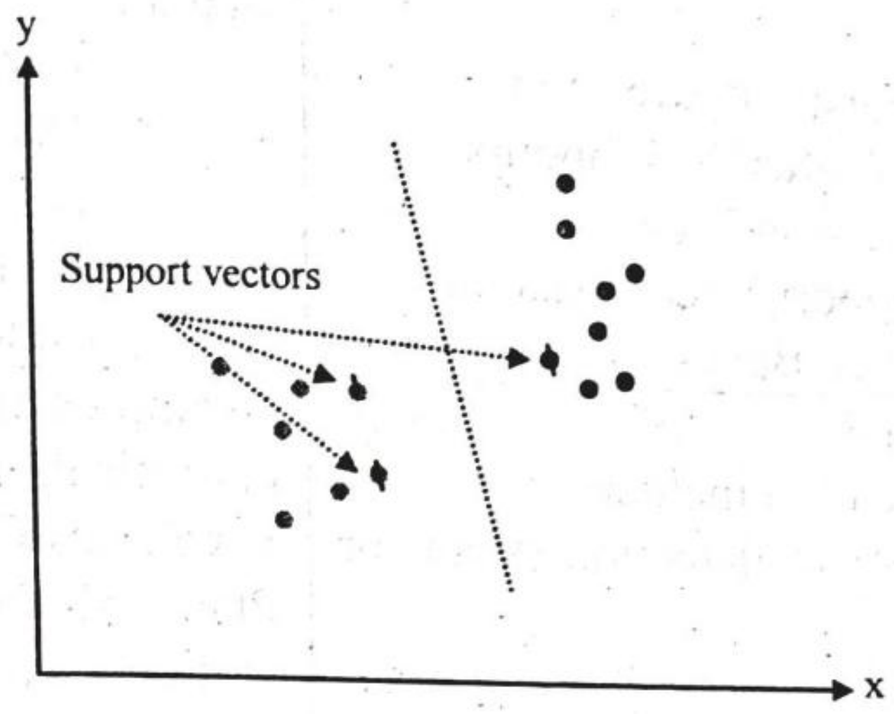
13. What is support vector machine algorithm? **[MODEL QUESTION]**

Answer:

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-

Scanned by TapScanner

dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line).

14. Define multiclass classification? **[MODEL QUESTION]**

Answer:

Multi-class classification is the task of classifying elements into different classes. Unlike binary, it doesn't restrict itself to any number of classes.

Examples of multi-class classification are

- classification of news in different categories,
- classifying books according to the subject,
- classifying students according to their streams etc.

In these, there are different classes for the response variable to be classified in and thus according to the name, it is a Multi-class classification.

15. What isn't Multiclass Classification? **[MODEL QUESTION]**

Answer:

There are many scenarios in which there are multiple categories to which points belong, but a given point can belong to multiple categories. In its most basic form, this problem decomposes trivially into a set of unlinked binary problems, which can be solved naturally using our techniques for binary classification.

16. Differentiate between binary and multiclass classification. **[MODEL QUESTION]**

Answer:

Parameters	Binary classification	Multi-class classification
No. of classes	It is a classification of two groups, i.e., classifies objects in at most two classes.	There can be any number of classes in it, i.e., classifies the object into more than two classes.

Scanned by TapScanner

Parameters	Binary classification	Multi-class classification
Algorithms used	The most popular algorithms used by the binary classification are- Logistic Regression k-Nearest Neighbours Decision Trees Support Vector Machine Naive Bayes	Popular algorithms that can be used for multi-class classification include: k-Nearest Neighbours Decision Trees Naive Bayes Random Forest. Gradient Boosting
Examples	Examples of binary classification include- Email spam detection (spam or not). Churn prediction (churn or not). Conversion prediction (buy or not).	Examples of multi-class classification include: Face classification. Plant species classification. Optical character recognition.

17. What machine learning algorithms are used for multiclass classification? **[MODEL QUESTION]**

Answer:

Some of the most popular algorithms for multi-class classifications are:

- **Decision Trees** – The classification model is built using the decision tree method in the form of a tree structure. It employs if-then principles, which are both exhaustive and mutually exclusive when it comes to categorization. The process continues with the data being broken down into smaller structures and finally being linked to an incremental decision tree. The finished product resembles a tree with nodes and leaves. The rules are learnt one by one, one by one, utilizing the training data. The tuples that cover the rules are eliminated each time a rule is learnt. On the training set, the procedure continues until the termination point is reached.
- **k-Nearest Neighbours** – It's an n-dimensional space-based lazy learning method that saves all sections corresponding to training data. It is lazy since it focuses on keeping sections of training data rather than building a broad internal model. A majority vote of each point's k closest neighbours is used to classify it. It's supervised and utilizes a collection of identified points to class/label other points. It looks at the labeled points closest to the new point, usually known as its nearest neighbours, to label it. It polls those neighbours, and whichever label receives the most votes becomes the new point's label.

Scanned by TapScanner

Long Answer Type Questions

1. Explain conventional algorithm in machine learning. **[MODEL QUESTION]**

Answer:

The algorithmic solution of a problem requires the input data and a program to produce an output. Here, a program is a set of instructions, and output is generated by applying those instructions to the input data. In a machine learning algorithm, the system takes the Input data. In a machine learning algorithm, the system takes the Input Data along with the examples of output (in the case of supervised learning). It creates a model, which establishes (or tries to establish) some relation between the input and the output. Learning, in general, is improving the outcome using experience (E). How do we know that we have improved? The performance measure tells the performance of our model. As per Tom Michel, machine learning can be defined as follows.

If the performance measure (P) improves with experience (E) on task (T), then the system is said to have learned.

Here the Task (T) can be classification, Regression, Clustering and so on. The data constitutes experience (E). The performance measure (P) can be any accuracy, specificity, sensitivity, F measure, Sum of squared errors, and so on. These terms will be defined as we proceed. To understand these, let us consider an example of disease classification using Magnetic Resonance Imagining. If the number of patients correctly classified (accuracy) as diseased is considered as a performance measure, then this problem can be defined as follows:

- T: Classify given patients as diseased or not-diseased
- P: Accuracy
- E: The MRI images of a patient

The tusk will be accomplished by pre-processing the given data, extracting relevant features from the pre-processed data, selecting the most important features, applying a classification algorithm followed by post-processing. In general, a machine learning pipeline constitutes the following steps (Figure 1):

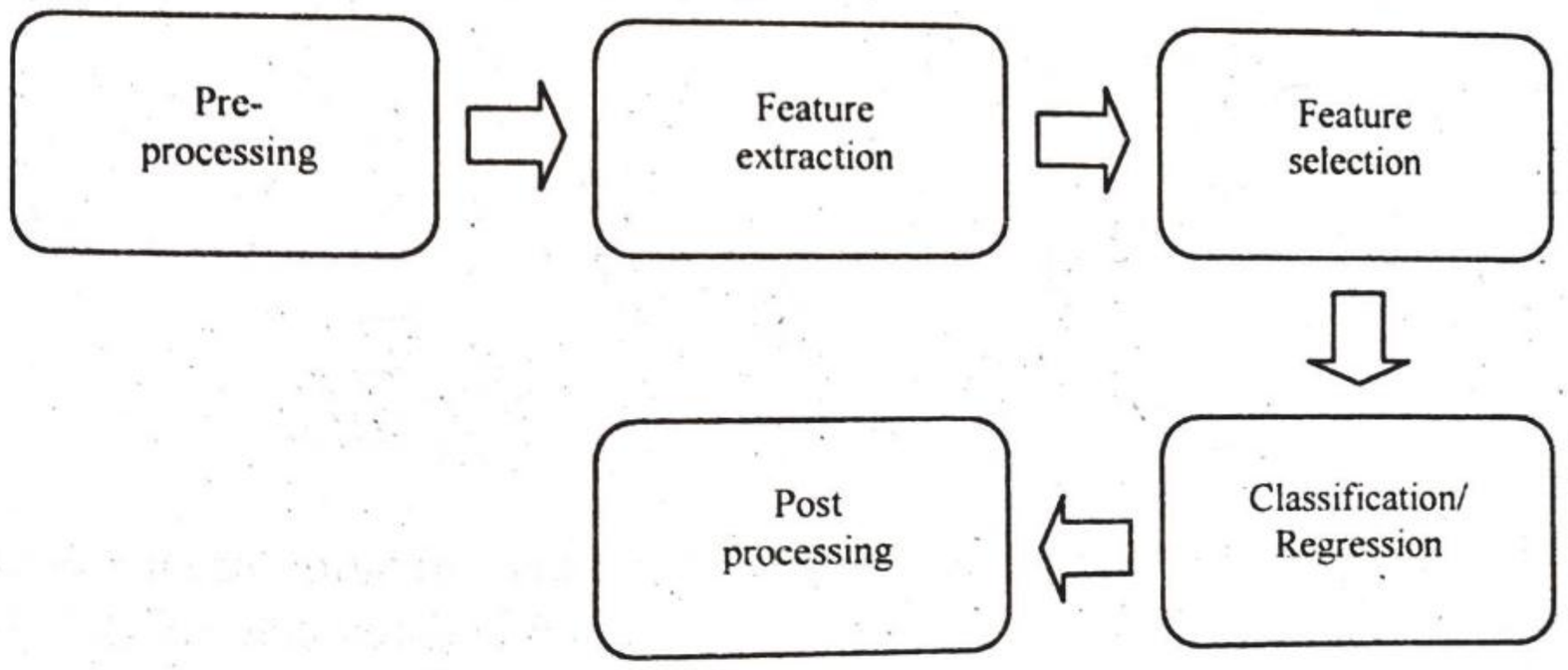
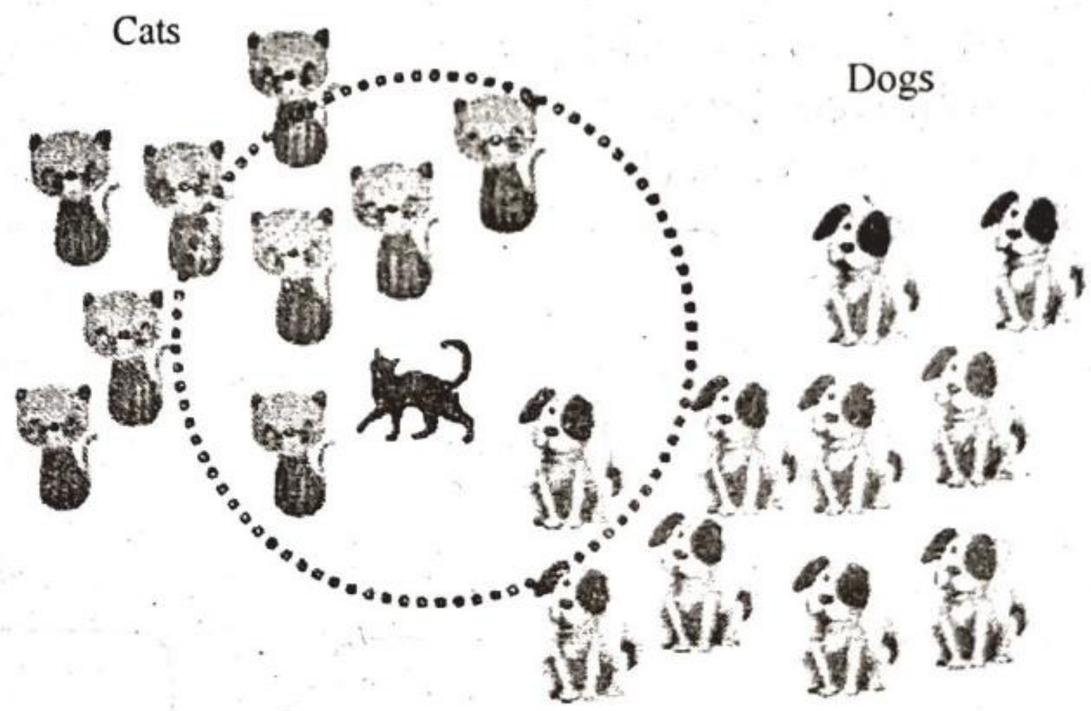


Fig: 1 Machine learning pipeline

Scanned by TapScanner

2. Explain K-Nearest Neighbour – The Distance-Based machine learning algorithm. [MODEL QUESTION]

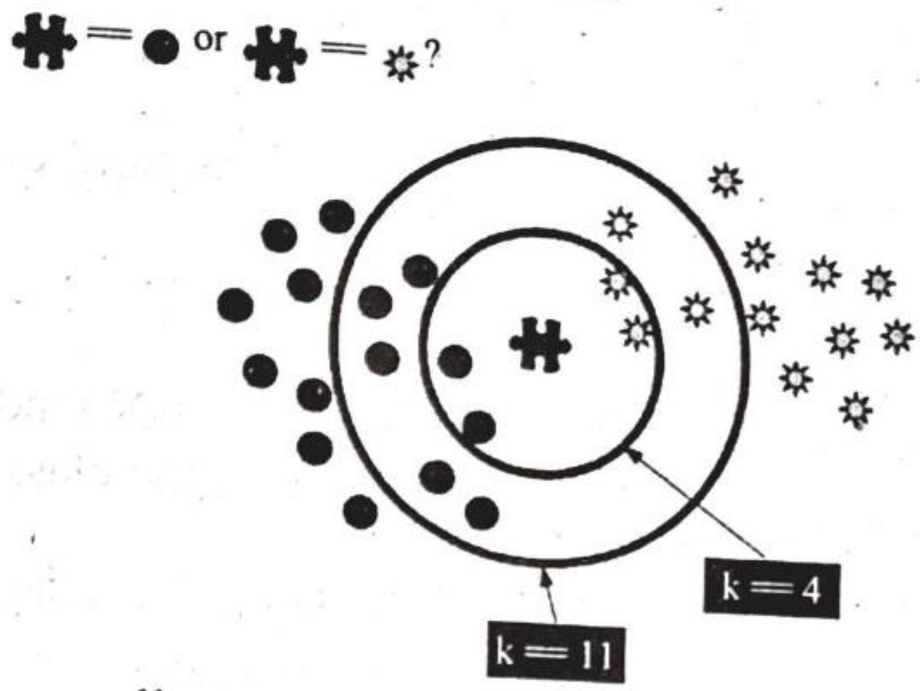
Answer:
The abbreviation KNN stands for "K-Nearest Neighbour". It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements.
The number of nearest neighbours to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'. Let's take a good look at a related real-world scenario before we get started with this awesome algorithm.
We are often notified that you share many characteristics with your nearest peers, whether it be your thinking process, working etiquettes, philosophies, or other factors. As a result, we build friendships with people we deem similar to us.
The KNN algorithm employs the same principle. Its aim is to locate all of the closest neighbours around a new unknown data point in order to figure out what class it belongs to. It's a distance-based approach.
Consider the diagram below; it is straightforward and easy for humans to identify it as a "Cat" based on its closest allies. This operation, however, cannot be performed directly by the algorithm.
KNN calculates the distance from all points in the proximity of the unknown data and filters out the ones with the shortest distances to it. As a result, it's often referred to as a distance-based algorithm.
In order to correctly classify the results, we must first determine the value of K (Number of Nearest Neighbours).
In the following diagram, the value of K is 5. Since there are four cats and just one dog in the proximity of the five closest neighbours, the algorithm would predict that it is a cat based on the proximity of the five closest neighbours in the red circle's boundaries.



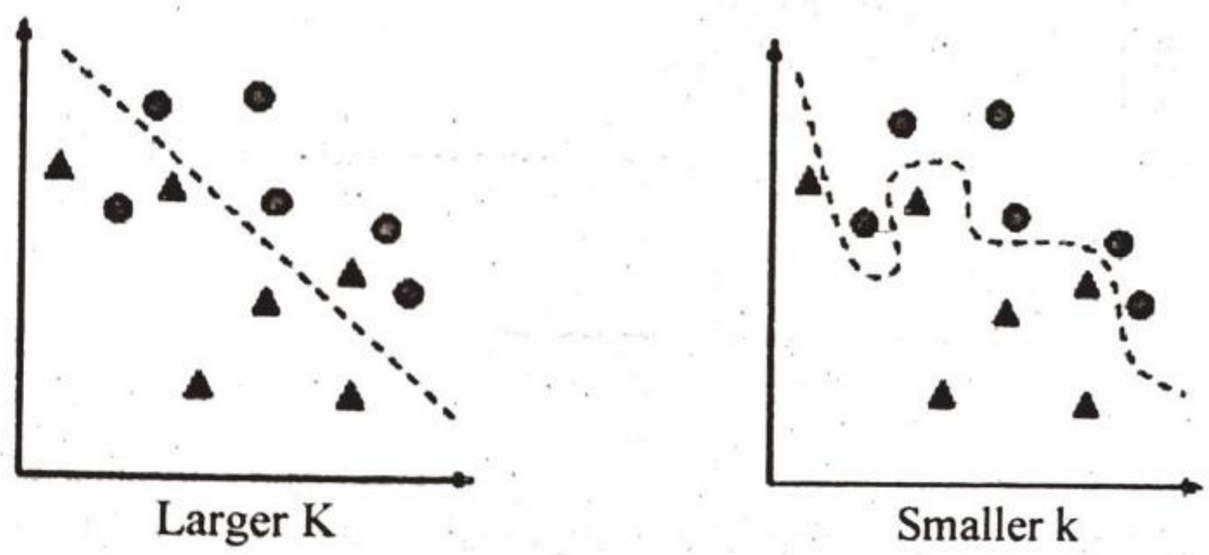
Here, 'K' is the hyperparameter for KNN. For proper classification/prediction, the value of K must be fine-tuned. We don't have a particular method for determining the correct value of K. Here, we'll try to test the model's accuracy for different K values. The value of K that delivers the best accuracy for both training and testing data is selected.

Scanned by TapScanner

It is recommended to always select an odd value of K ~
When the value of K is set to even, a situation may arise in which the elements from both groups are equal. In the diagram below, elements from both groups are equal in the internal "Red" circle (k == 4).
In this condition, the model would be unable to do the correct classification for you. Here the model will randomly assign any of the two classes to this new unknown data.
Choosing an odd value for K is preferred because such a state of equality between the two classes would never occur here. Due to the fact that one of the two groups would still be in the majority, the value of K is selected as odd.



The impact of selecting a smaller or larger K value on the model
Larger K value: The case of underfitting occurs when the value of k is increased. In this case, the model would be unable to correctly learn on the training data.
Smaller k value: The condition of overfitting occurs when the value of k is smaller. The model will capture all of the training data, including noise. The model will perform poorly for the test data in this scenario.



kNN algorithm
Input: Training data set, test data set (or data points), value of 'k' (i.e., number of nearest neighbours to be considered)
Steps:
Do for all test data points
Calculate the distance (usually Euclidian distance) of the test data point from the different training data points.

Scanned by TapScanner

Find the closest 'k' training data points, i.e., training data points whose distance are least from the test data point.

If k=1

Then assign class label of the training data point to the test data point

Else

Whichever class label is predominantly present in the training data points, assign that class label to the test data points , assign that class label to the test data point

End do

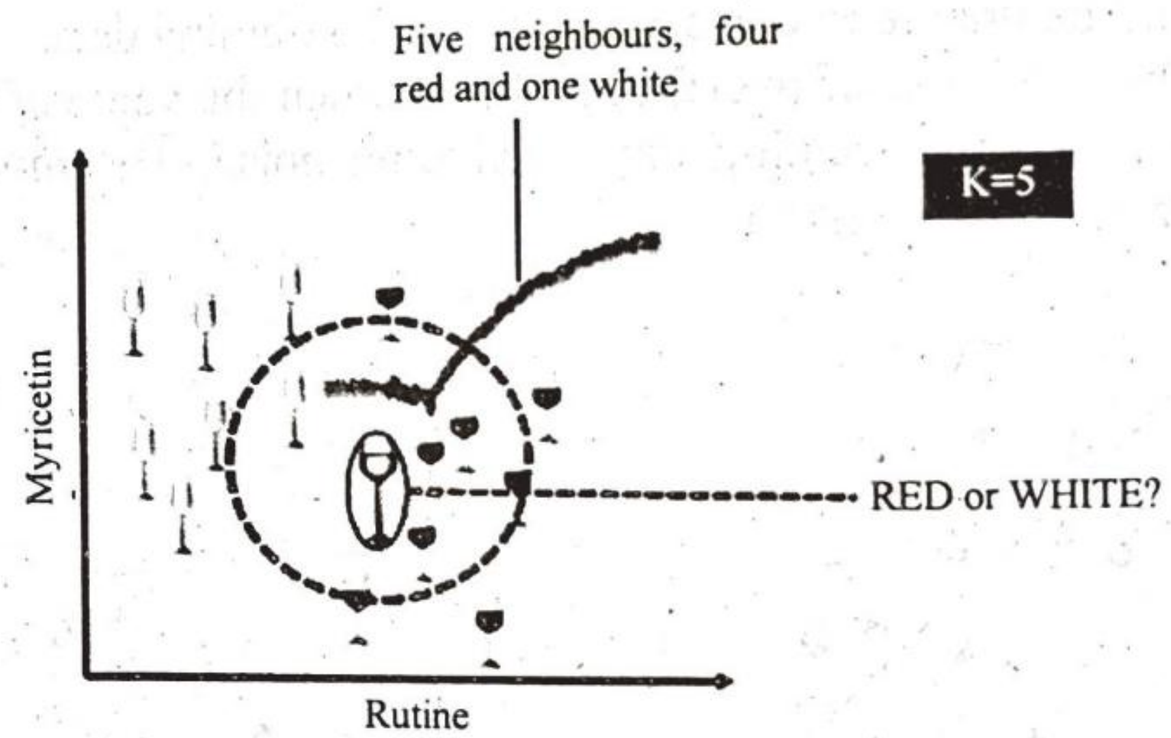
3. How does KNN work for 'Classification' and 'Regression' problem statements? [MODEL QUESTION]

Answer:

Classification:

When the problem statement is of 'classification' type, KNN tends to use the concept of "Majority Voting". Within the given range of K values, the class with the most votes is chosen.

Consider the following diagram, in which a circle is drawn within the radius of the five closest neighbours. Four of the five neighbours in this neighbourhood voted for 'RED,' while one voted for 'WHITE.' It will be classified as a 'RED' wine based on the majority votes.



Real-world example:

Several parties compete in an election in a democratic country like India. Parties compete for voter support during election campaigns. The public votes for the candidate with whom they feel more connected.

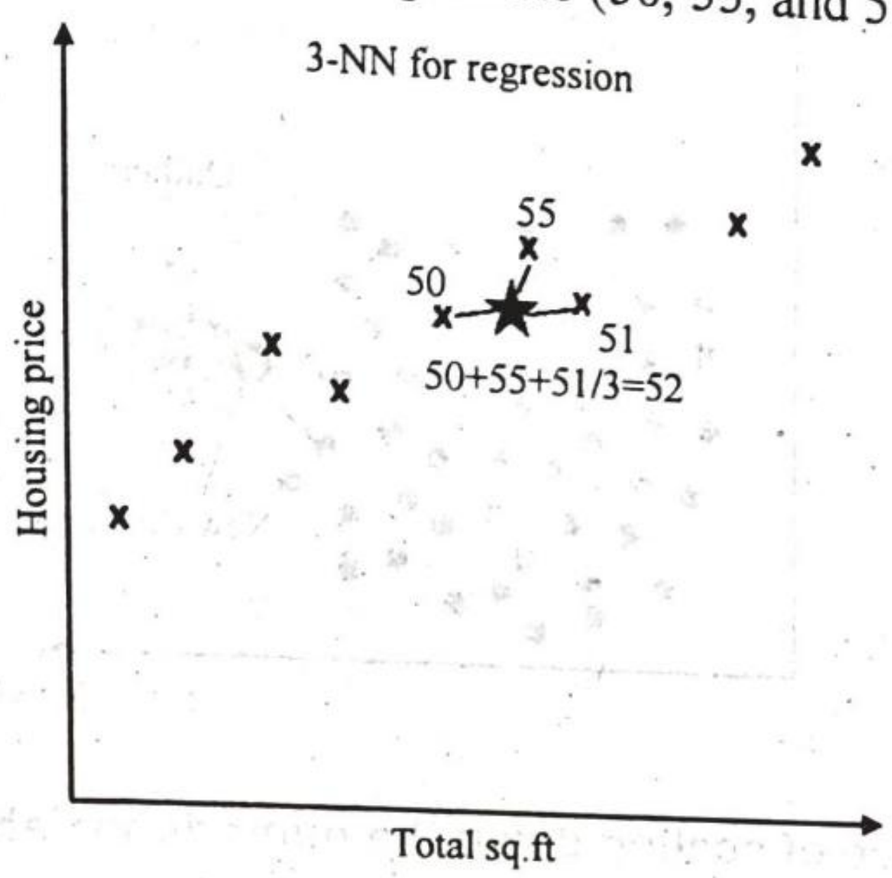
When the votes for all of the candidates have been recorded, the candidate with the most votes is declared as the election's winner.

Regression:

KNN employs a mean/average method for predicting the value of new data. Based on the value of K, it would consider all of the nearest neighbours.

Scanned by TapScanner

The algorithm attempts to calculate the mean for all the nearest neighbours' values until it has identified all the nearest neighbours within a certain range of the K value.
Consider the diagram below, where the value of k is set to 3. It will now calculate the mean (52) based on the values of these neighbours (50, 55, and 51) and allocate this value to the unknown data.

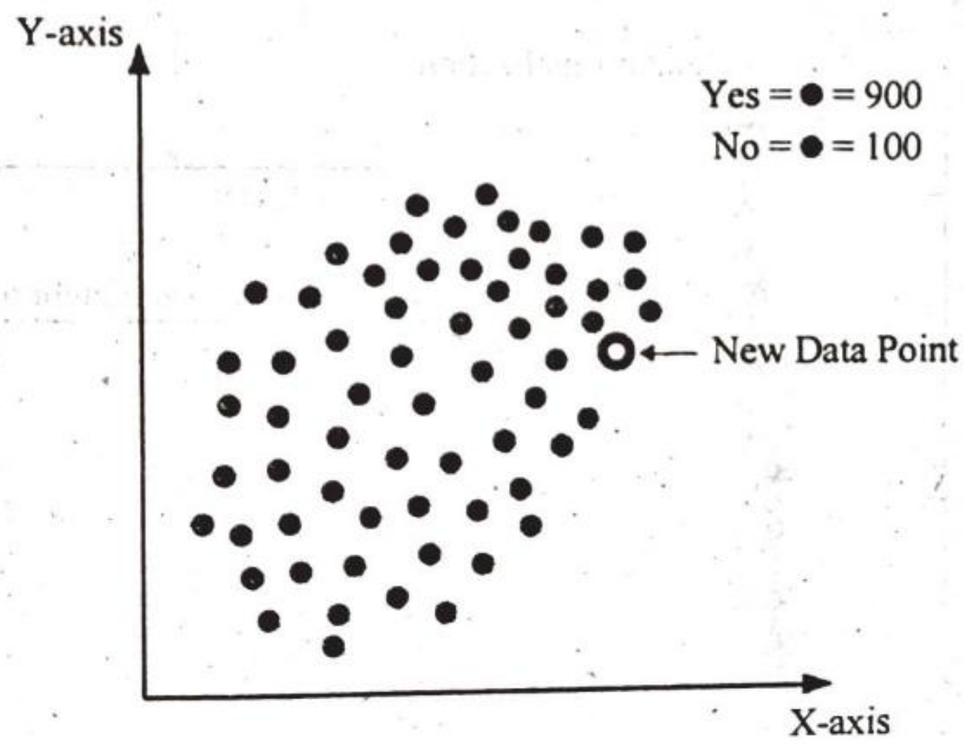


4. What is impact of imbalanced dataset and outliers on KNN? [MODEL QUESTION]

Answer:

- **Imbalanced dataset**

When dealing with an imbalanced data set, the model will become biased. Consider the example shown in the diagram below, where the "Yes" class is more prominent.
As a consequence, the bulk of the closest neighbours to this new point will be from the dominant class. Because of this, we must balance our data set using either an "Upscaling" or "Downscaling" strategy.

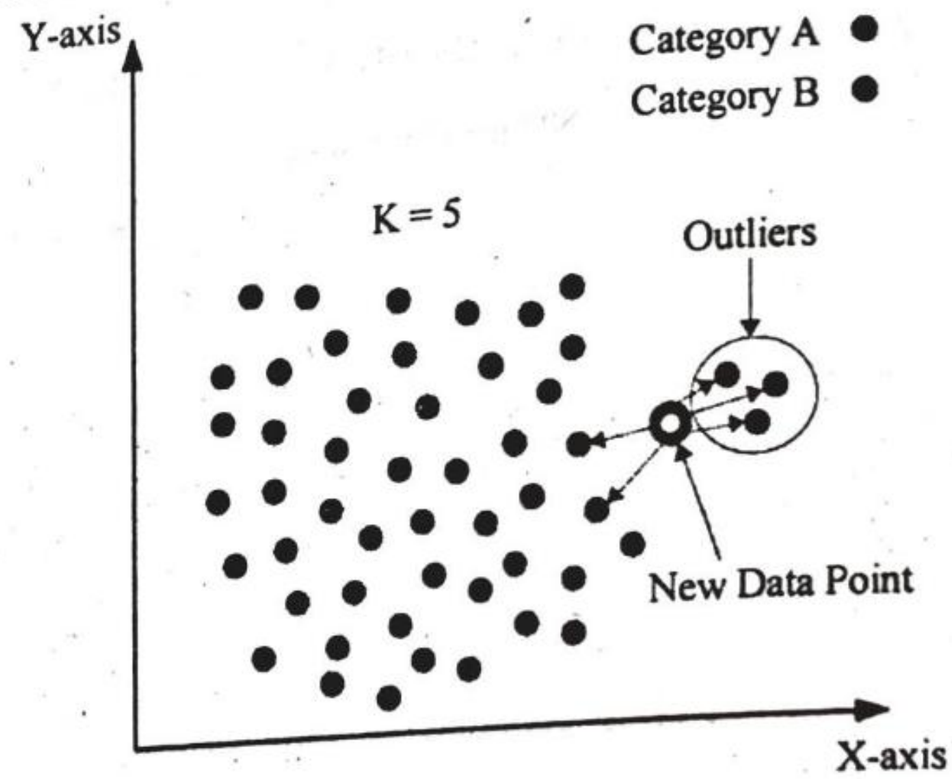


- **Outliers**

Outliers are the points that differ significantly from the rest of the data points.
The outliers will impact the classification/prediction of the model. The appropriate class for the new data point, according to the following diagram, should be "Category B" in green.

Scanned by TapScanner

The model, however, would be unable to have the appropriate classification due to the existence of outliers. As a result, removing outliers before using KNN is recommended.



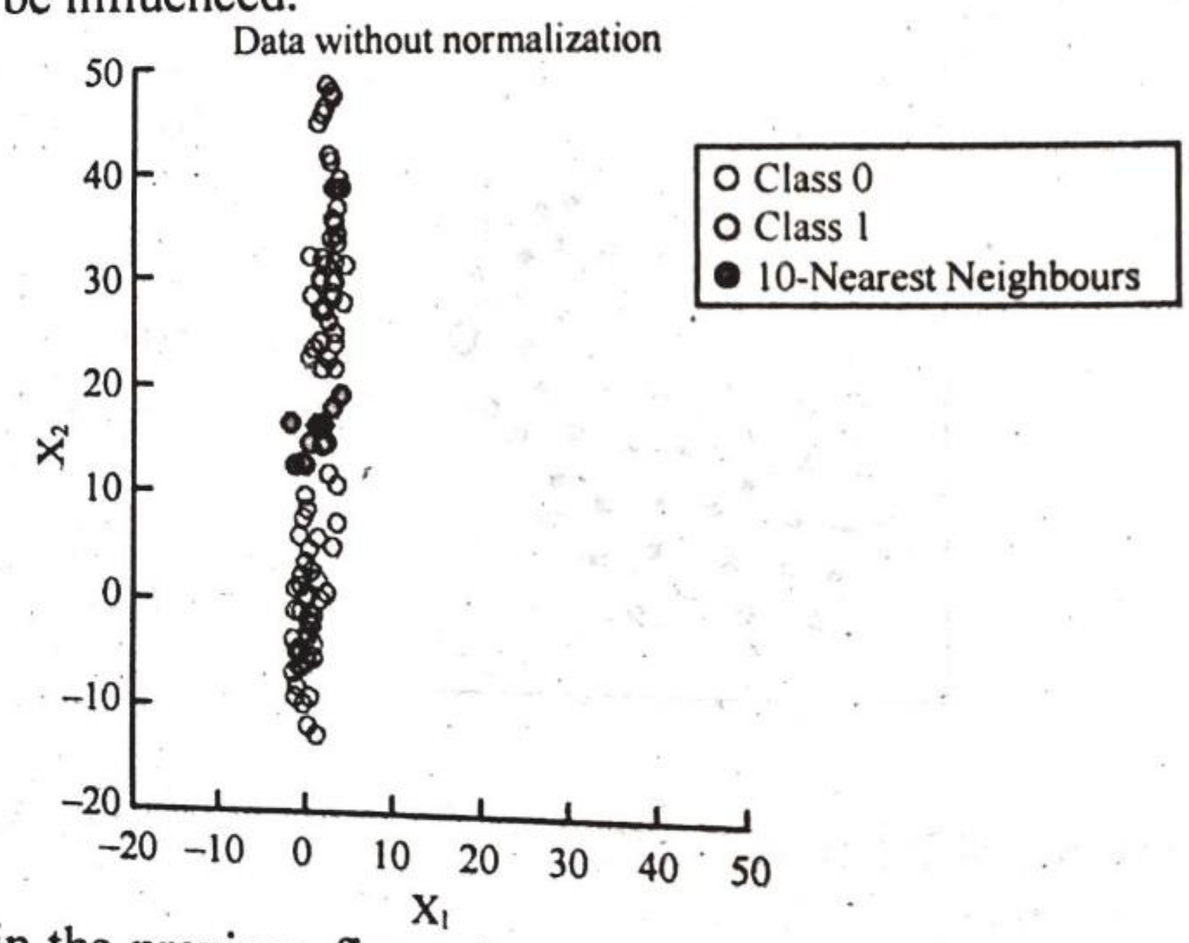
5. What is the importance of scaling down the numeric variables to the same level? **[MODEL QUESTION]**

Answer:

Data has 2 parts:

1) Magnitude
2) Unit

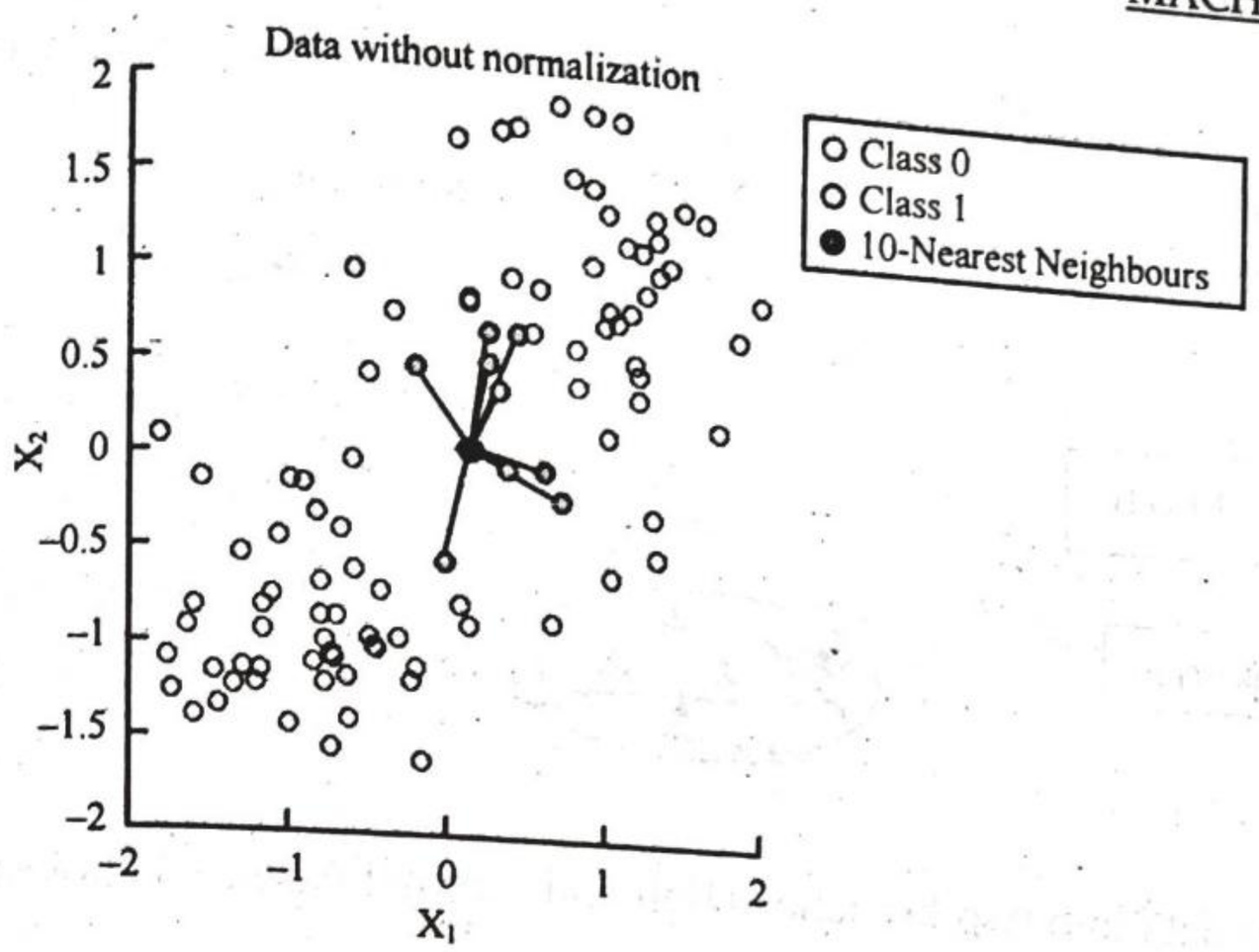
For instance; if we say 20 years then "20" is the magnitude here and "years" is its unit. Since it is a distance-dependent algorithm, KNN selects the neighbours in the closest vicinity based solely on the magnitude of the data. Have a look at the diagram below; the data is not scaled, so it can not find the closest neighbours correctly. As a consequence, the outcome will be influenced.



The data values in the previous figure have now been scaled down to the same level in the following example. Based on the scaled distance, all of the closest neighbours would be accurately identified.

Scanned by TapScanner

Scanned by TapScanner



6. Explain Decision tree algorithm. **[MODEL QUESTION]**

Answer:

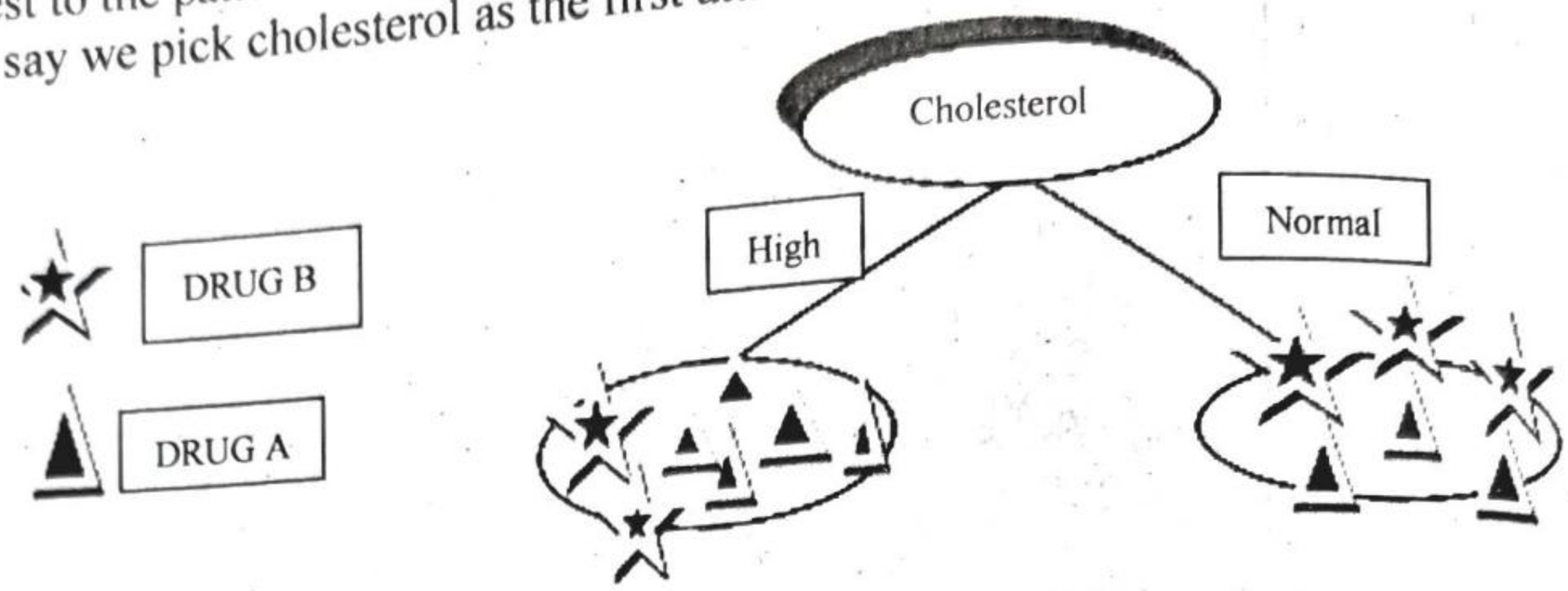
The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

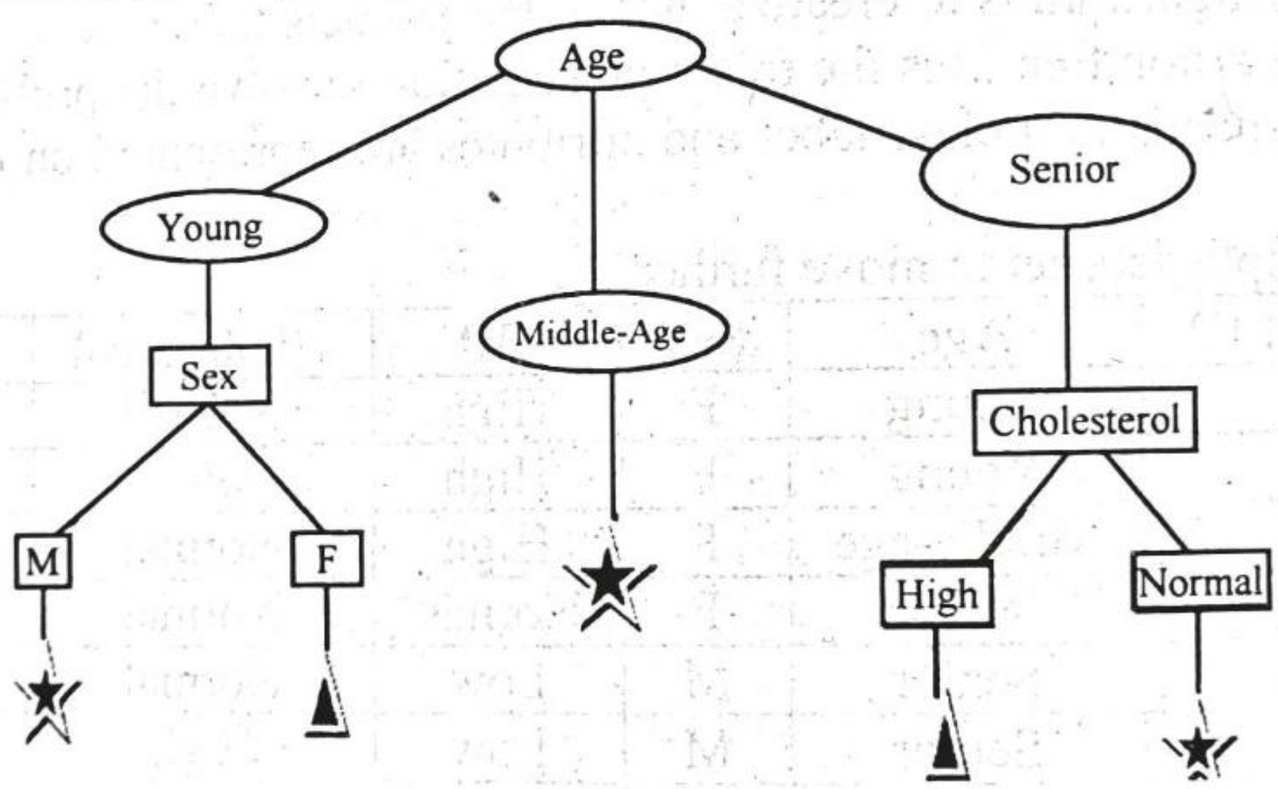
Let's take a sample data set to move further ....

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	High	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A
p15	Middle-age	F	Low	Normal	?

Suppose we have a sample of 14 patient data set and we have to predict which drug to suggest to the patient A or B.
Let's say we pick cholesterol as the first attribute to split data



It will split our data into two branches High and Normal based on cholesterol, as you can see in the above figure.
Let's suppose our new patient has high cholesterol by the above split of our data we cannot say whether Drug B or Drug A will be suitable for the patient.
Also, If the patient cholesterol is normal we still do not have an idea or information to determine that either Drug A or Drug B is Suitable for the patient.
Let us take Another Attribute Age, as we can see age has three categories in it Young, middle age and senior let's try to split.



From the above figure, now we can say that we can easily predict which Drug to give to a patient based on his or her reports.

Assumptions that we make while using the Decision tree:

- In the beginning, we consider the whole training set as the root.
- Feature values are preferred to be categorical; if the values continue then they are converted to discrete before building the model.
- Based on attribute values records are distributed recursively.
- We use a statistical method for ordering attributes as a root node or the internal node.

Scanned by TapScanner

Mathematics behind Decision tree algorithm: Before going to the Information Gain first we have to understand entropy

Entropy: Entropy is the measures of impurity, disorder, or uncertainty in a bunch of examples.

Purpose of Entropy:

Entropy controls how a Decision Tree decides to split the data. It affects how a Decision Tree draws its boundaries.

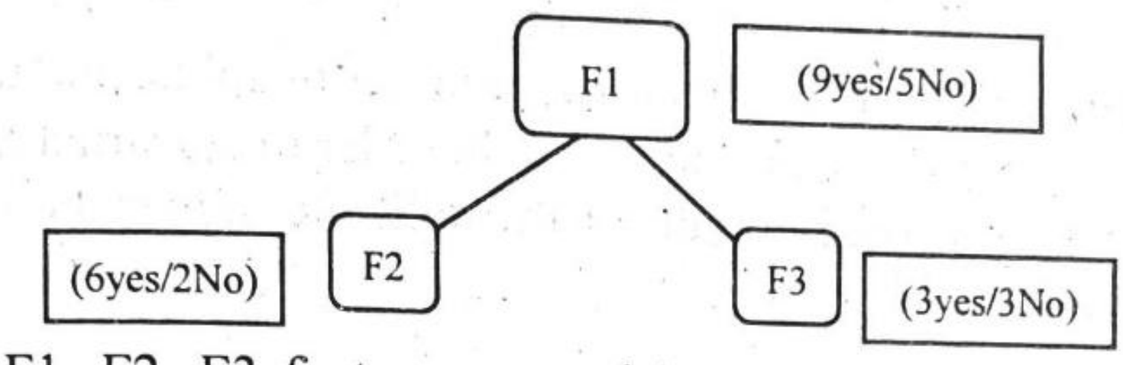
"Entropy values range from 0 to 1", Less the value of entropy more it is trusting able.

Formula for Entropy:

$$H(s) = -\text{probability of } \log_2(p+) - -\text{probability of } \log_2(p-)$$

where $(p+) \rightarrow \%$ of positive class

$(p-) \rightarrow \%$ of negative class



Suppose we have F1, F2, F3 features we selected the F1 feature as our root node F1 contains 9 yes label and 5 no label in it, after splitting the F1 we get F2 which have 6 yes/2 No and F3 which have 3yes/3 no.

Now if we try to calculate the Entropy of both F2 by using the Entropy formula…

Putting the values in the formula:

$$H(s) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} = 0.81\,\text{bits}$$

Here, 6 is the number of yes taken as positive as we are calculating probability divided by 8 is the total rows present in the F2.

Similarly, if we perform Entropy for F3 we will get 1 bit which is a case of an attribute as in it there is 50%, yes and 50% no.

This splitting will be going on unless and until we get a pure subset.

The **pure subset** is a situation where we will get either all yes or all no in this case.

We have performed this concerning one node what if after splitting F2 we may also require some other attribute to reach the leaf node and we also have to take the entropy of those values and add it up to do the submission of all those entropy values for that we have the concept of information gain.

Information Gain: Information gain is used to decide which feature to split on at each step in building the tree. Simplicity is best, so we want to keep our tree small. To do so, at each step we should choose the split that results in the purest daughter nodes. A commonly used measure of purity is called information.

For each node of the tree, the information value measures how much information a feature gives us about the class. The split with the highest information gain will be taken as the first split and the process will continue until all children nodes are pure, or until the information gain is 0.

Scanned by TapScanner

Formula for Information Gain:

$$\text{Gain (S.A)} = H(s)\sum\frac{|Sv|}{|s|}H(Sv)$$

The algorithm calculates the information gain for each split and the split which is giving the highest value of information gain is selected.
We can say that in Information gain we are going to compute the average of all the entropy-based on the specific split.
Sv = Total sample after the split as in F2 there are 6 yes
s = Total Sample as in $F1 = 9 + 5 = 14$
Now calculating the Information Gain:

$$\text{Gain (S.A)} = H(F1) - \frac{8}{14} * H(F2) - \frac{6}{14} * H(F3) = 0.91 - \frac{8}{14} * 0.81 - \frac{6}{14} * 1 = 0.05$$

Like this, the algorithm will perform this for n number of splits, and the information gain for whichever split is higher it is going to take it in order to construct the decision tree.
The higher the value of information gain of the split the higher the chance of it getting selected for the particular split.

Gini Impurity:

Gini Impurity is a measurement used to build Decision Trees to determine how the features of a data set should split nodes to form the tree. More precisely, the Gini Impurity of a data set is a number between 0-0.5, which indicates the likelihood of new, random data being miss classified if it were given a random class label according to the class distribution in the data set.

Entropy vs. Gini Impurity

The maximum value for entropy is 1 whereas the maximum value for Gini impurity is 0.5.
As the Gini Impurity does not contain any logarithmic function to calculate it takes less computational time as compared to entropy.

Algorithm for decision tree

Input: Training data set, test data set (or data points)

Steps:

Do for all attributes
Calculate the entropy E_i of the attribute F_i
    if $E_i < E_{min}$
        then $E_{min} = E_i$ and $F_{min} = F_i$
    end if
End do
Split the data set into subsets using the attribute F_{min}
Draw a decision tree node containing the attribute F_{min} and split the data set into subsets
Repeat the above steps until the full tree is drawn converting all the attributes of the original table

Scanned by TapScanner

7. What is Naive Bayes algorithm? **[MODEL QUESTION]**

Answer:

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

Likelihood Class prior probability

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Posterior probability Predictor prior probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

Above,

- $P(c|x)$ is the posterior probability of class (c, target) given predictor (x, attributes).
- $P(c)$ is the prior probability of class.
- $P(x|c)$ is the likelihood which is the probability of predictor given class.
- $P(x)$ is the prior probability of predictor.

8. How Naive Bayes algorithm work? **[MODEL QUESTION]**

Answer:

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Scanned by TapScanner

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.
P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) $= \frac{3}{9} = 0.33$, P(Sunny) $= \frac{5}{14} = 0.36$, P(Yes) $= \frac{9}{14} = 0.64$

Now, P (Yes | Sunny) $= 0.33 \times \frac{0.64}{0.36} = 0.60$, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

Scanned by TapScanner

9. What are the Pros and Cons of Naive Bayes? **[MODEL QUESTION]**

Answer:

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

10. What are applications of Naive Bayes algorithms? **[MODEL QUESTION]**

Answer:

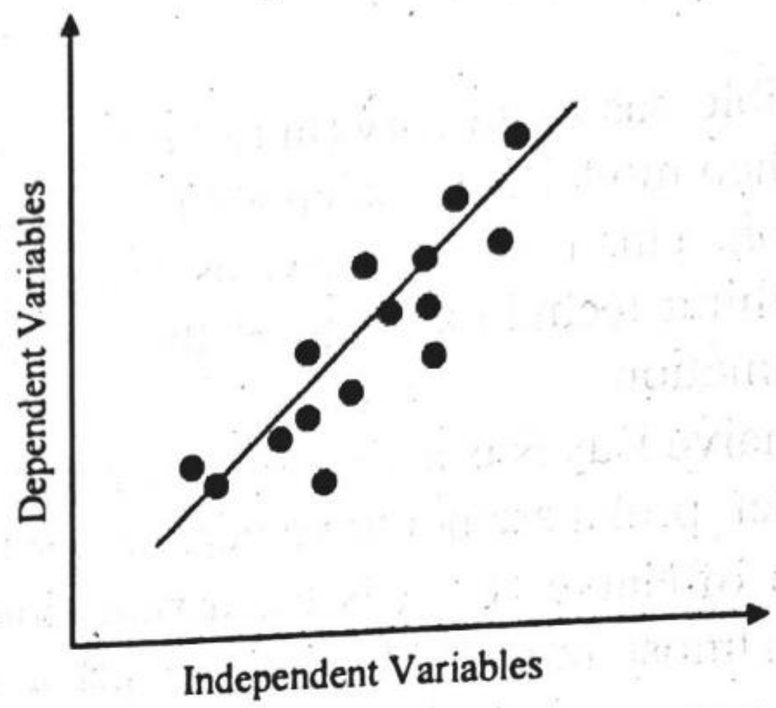
- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction**: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification / Spam Filtering / Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail)' and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments).
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

Scanned by TapScanner

11. Explain linear regression technique.

Answer:

Linear regression is a quiet and simple statistical regression method used for predictive analysis and shows the relationship between the continuous variables. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression. If there is a single input variable (x), such linear regression is called simple linear regression. And if there is more than one input variable, such linear regression is called multiple linear regression. The linear regression model gives a sloped straight line describing the relationship within the variables.



The above graph presents the linear relationship between the dependent variable and independent variables. When the value of x (independent variable) increases, the value of y (dependent variable) is likewise increasing. The red line is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best.

To calculate best-fit line linear regression uses a traditional slope-intercept form.

$$y = mx + b \rightarrow y = a_0 + a_1 x$$

$y =$ Dependent Variable

$x =$ Independent Variable

$a_0 =$ Intercept of the line.

$a_1 =$ Linear regression coefficient.

Need of a Linear regression

As mentioned above, Linear regression estimates the relationship between a dependent variable and an independent variable. Let's understand this with an easy example:

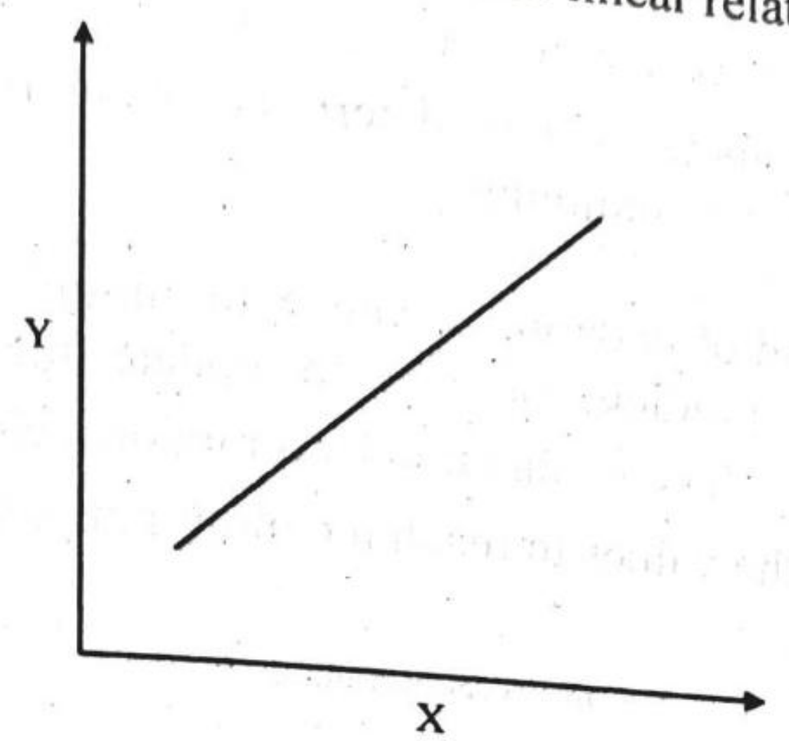
Let's say we want to estimate the salary of an employee based on year of experience. You have the recent company data, which indicates that the relationship between experience and salary. Here year of experience is an independent variable, and the salary of an employee is a dependent variable, as the salary of an employee is dependent on the experience of an employee. Using this insight, we can predict the future salary of the employee based on current & past information.

Scanned by TapScanner

A regression line can be a Positive Linear Relationship or a Negative Linear Relationship.

Positive Linear Relationship

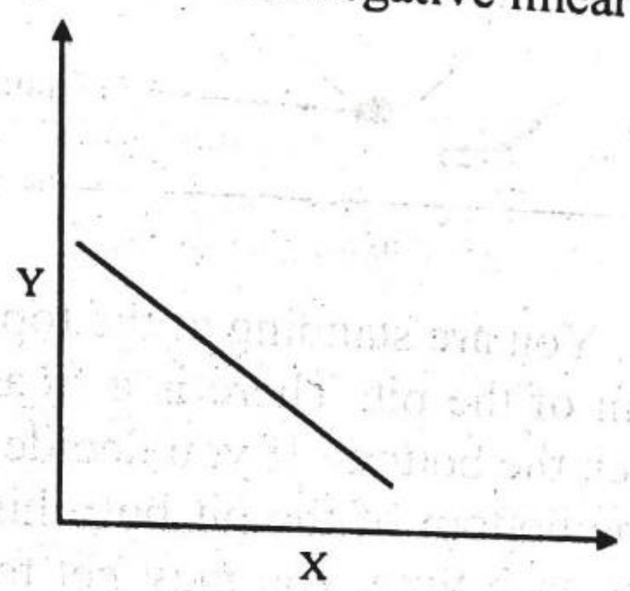
If the dependent variable expands on the Y-axis and the independent variable progress on X-axis, then such a relationship is termed a Positive linear relationship.



$y = a_0 + a_1 x$

Negative Linear Relationship

If the dependent variable decreases on the Y-axis and the independent variable increases on the X-axis, such a relationship is called a negative linear relationship.



$y = -a_0 + a_1 x$

The goal of the linear regression algorithm is to get the best values for a_0 and a_1 to find the best fit line. The best fit line should have the least error means the error between predicted values and actual values should be minimized.

Cost function

The cost function helps to figure out the best possible values for a_0 and a_1, which provides the best fit line for the data points.

Cost function optimizes the regression coefficients or weights and measures how a linear regression model is performing. The cost function is used to find the accuracy of the **mapping function** that maps the input variable to the output variable. This mapping function is also known as **the Hypothesis function.**

In Linear Regression, **Mean Squared Error (MSE)** cost function is used, which is the average of squared error that occurred between the predicted values and actual values.

By simple linear equation $y = mx + b$ we can calculate MSE as:

Scanned by TapScanner

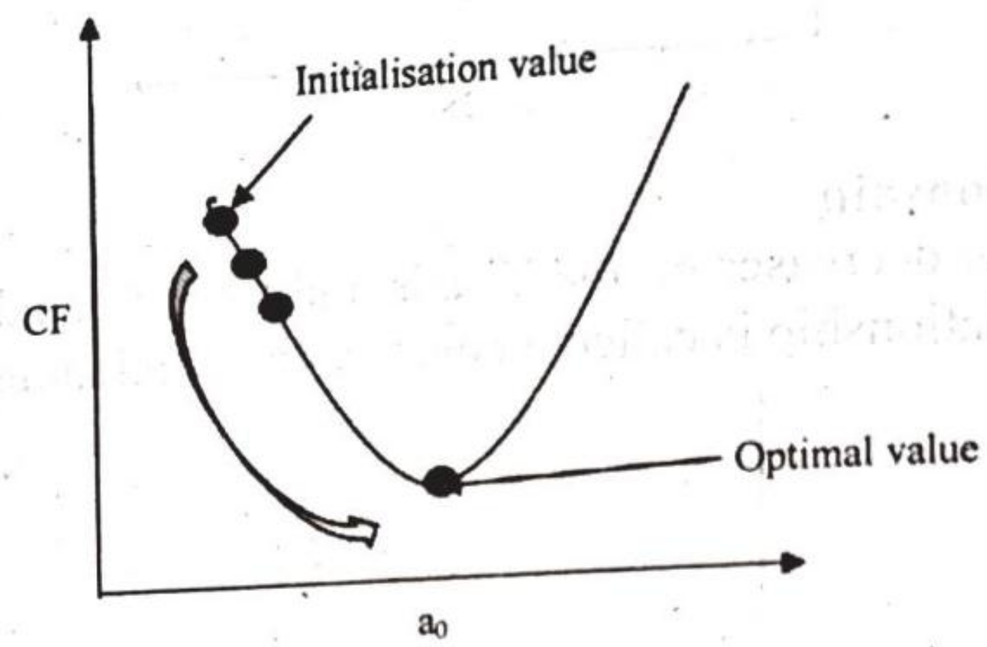
Let's y = actual values, y_i = predicted values.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{n}\left(y_i - (mx_i + b)\right)^2$$

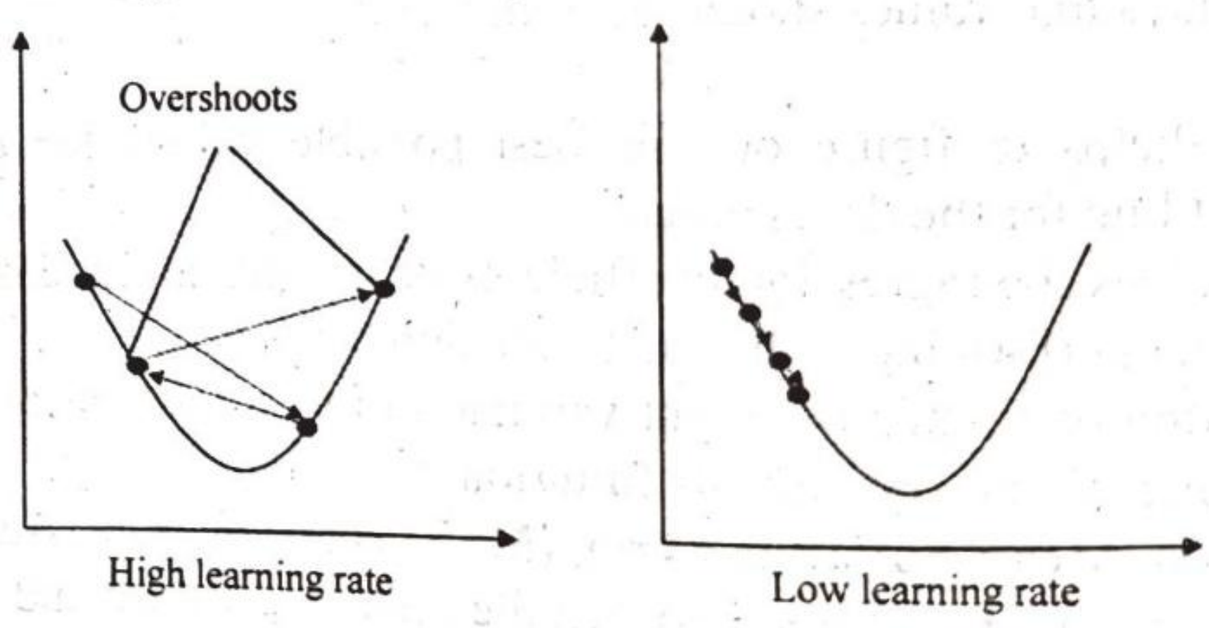
Using the MSE function, we will change the values of a_0 and a_1 such that the MSE value settles at the minima. Model parameters $x_i, b(a_0, a_1)$ can be manipulated to minimize the cost function. These parameters can be determined using the gradient descent method so that the cost function value is minimum.

Gradient descent

Gradient descent is a method of updating a_0 and a_1 to minimize the cost function (MSE). A regression model uses gradient descent to update the coefficients of the line $(a_0, a_1 \Rightarrow x_i, b)$ by reducing the cost function by a random selection of coefficient values and then iteratively update the values to reach the minimum cost function.



Imagine a pit in the shape of U. You are standing at the topmost point in the pit, and your objective is to reach the bottom of the pit. There is a treasure, and you can only take a discrete number of steps to reach the bottom. If you decide to take one footstep at a time, you would eventually get to the bottom of the pit but, this would take a longer time. If you choose to take longer steps each time, you may get to sooner but, there is a chance that you could overshoot the bottom of the pit and not near the bottom. In the gradient descent algorithm, the number of steps you take is the learning rate, and this decides how fast the algorithm converges to the minima.



Scanned by TapScanner

To update a_0 and a_1, we take gradients from the cost function. To find these gradients, we take partial derivatives for a_0 and a_1.

$$J = \frac{1}{n}\sum_{i=1}^{n}\left(a_0 + a_1 \cdot x_i - y_i\right)^2$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}\left(a_0 + a_1 \cdot x_i - y_i\right)$$

$$\frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}\left(a_0 + a_1 \cdot x_i - y_i\right)\cdot x_i$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}\left(pred_i - y_i\right)$$

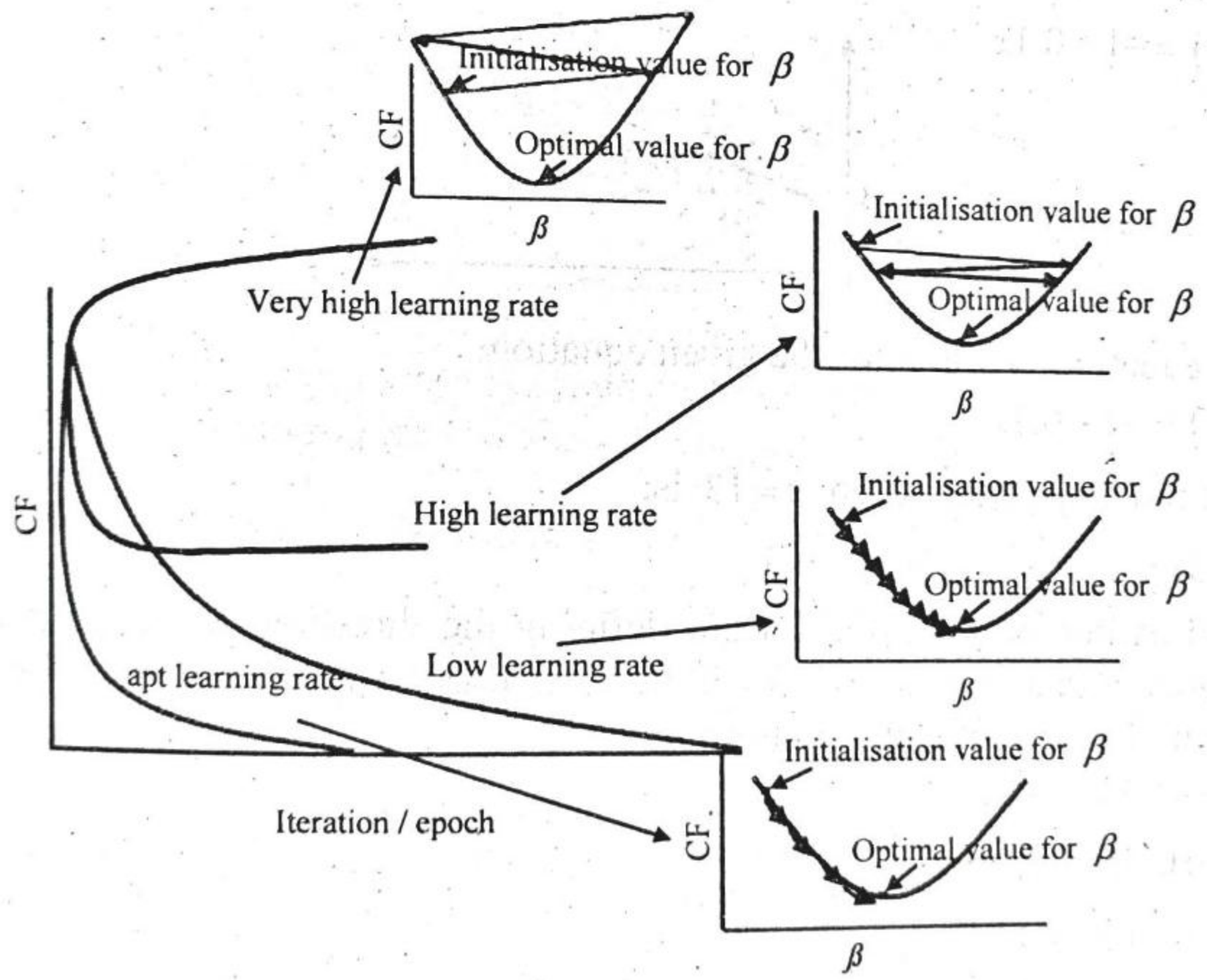
$$\frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}\left(pred_i - y_i\right)\cdot x_i$$

$$a_0 = a_0 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}\left(pred_i - y_i\right)$$

$$a_1 = a_1 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}\left(pred_i - y_i\right)\cdot x_i$$

The partial derivates are the gradients, and they are used to update the values of a_0 and a_1. Alpha is the learning rate.

Impact of different values for learning rate



Scanned by TapScanner

The blue line represents the optimal value of the learning rate, and the cost function value is minimized in a few iterations. The green line represents if the learning rate is lower than the optimal value, then the number of iterations required high to minimize the cost function. If the learning rate selected is very high, the cost function could continue to increase with iterations and saturate at a value higher than the minimum value, that represented by a red and black line.

12. Why do we have a curved line for logistic regression instead of a straight line? [MODEL QUESTION]

Answer:

Equation for a linear classifier in Linear regression:

$$\theta^T X + \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots\cdots\cdots + \theta_n X_n$$

where $\theta^T = [\theta_0, \theta_1, \theta_2, \cdots\cdots\cdots, \theta_n]$

(is a weight factor which is transpose of θ)

$X = [X_1, X_2, \cdots\cdots\cdots, X_n]$, which is a feature vector.

For now, the equation for the linear classifier is:

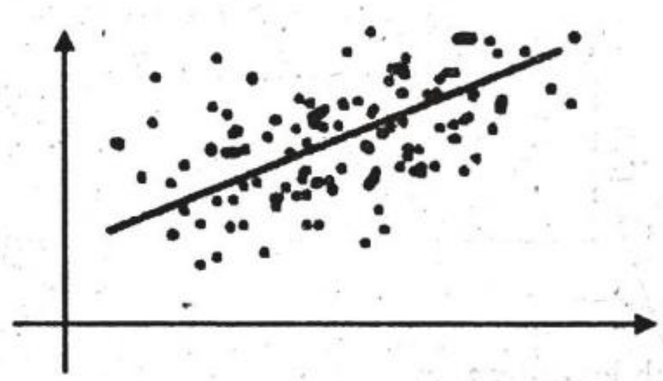
$$h(x) = \theta_0 + \theta_1 X_1$$

We will now define the values of the variable weights:

Theta $0 = -1$ and theta $= 0.1$

So our equation looks like this and the following is the graph representing the equation in the 2-D plane:

$$h(x) = -1 + 0.1x$$



Above is an equation of a line for the given equation:

$$h(x) = -1 + 0.1x$$

The value of function $h(x)$ when $x = 13$ is:

$$h(13) = -1 + (0.1) * (13) = 0.3$$

As described earlier in this article, I am defining the threshold to be 0.5 which is any value of h greater than (equal to) 0.5 will be labeled as 1 and otherwise 0. We can define it as follows in the form of step function:

Let $\hat{y}$ be the label of class

$$\hat{y} = \begin{cases} 0, \text{if } h(x) < 0.5 \\ 1, \text{if } h(x) \geq 0.5 \end{cases}$$

Here 0.5 is the threshold.

Scanned by TapScanner

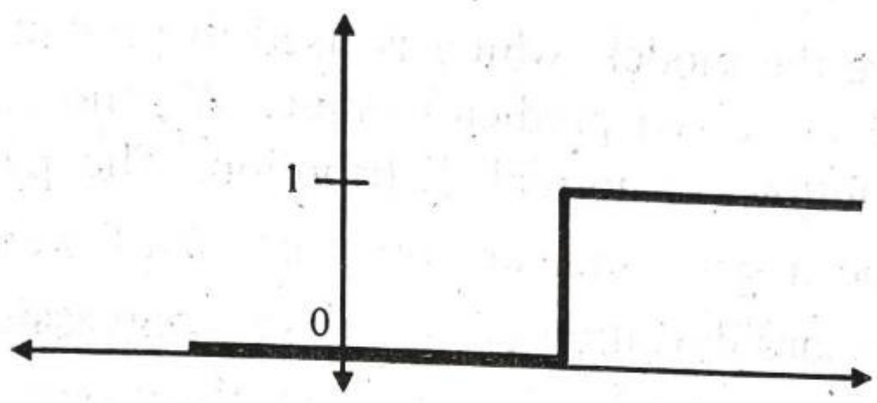
Now according to this h has a value of 0.3, hence the value of $\hat{y}=0$ as per the function defined above.

Now one thing you must note here that every value greater than 0.5, suppose I say the value of 'h' is 1000 for some value of x, then it will be labeled as 1 only, there is no difference between the value 1 and 1000 as both are classified as 1 only. Is it right? Can we accept this solution? Well, No! I would not accept it!!!

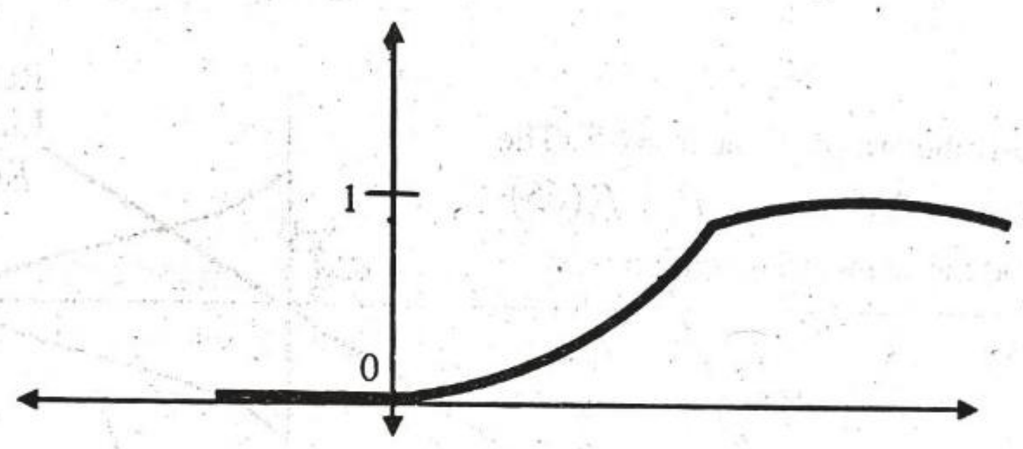
One more thing, what is the probability that the h has a value of 0.3? All these questions are left unanswered. For these reasons, data scientists do not prefer to use linear regression for classification purposes.

Before we move on I want to show you how the $\hat{y}$ function behaves graphically:

$$\hat{y}=\begin{cases}0, h(x)<0.5\\ 1, h(x)\geq 0.5\end{cases}$$



It will be better if we have a smoother curve instead of on given above. Let's see:



The above curve is known as the Sigmoid function which we will be using throughout this article. Here I will introduce the Sigmoid function.

13. What are generalized linear models? **[MODEL QUESTION]**

Answer:

Generalized linear models represent the class of regression models which models the response variable, Y and the random **error** term (e) based on **exponential family of distributions** such as normal, Poisson, Gamma, Binomial, inverse Gaussian etc. GLM assumes that the distribution of the response variable is a member of the exponential family of distribution. This is different from the **general linear models (linear regression / ANOVA)** where response variable, Y and the random **error** term (e) have to be based solely on the **normal distribution**. Linear models can be expressed in terms of expected value (mean) of response variable as the following:

Scanned by TapScanner

$$g(\mu)=\sum_{i=1}^{n}\beta_i X_i \cdots$$

where μ can be expressed as $E(Y)$ aka expected value of response variable Y. Based on the probability distribution of the response variable, different link functions get used which transforms $g(\mu)$ appropriately to the output value which gets modelled using different types of regression models. If the response variable is normally distributed, the link function is identifying function and the model looks like the following. Y, in the equation below, represents the expected value or $E(Y)$.

$$Y=\sum_{i=1}^{n}\beta_i X_i$$

Before getting into generalized linear models, lets quickly understand the concepts of general linear models.

General linear models are the models which is used to predict the value of continuous response variable and at any given predictor value, X, the response variable, Y and random error term (e) follows a normal distribution. The parameter of such normal distribution represents the mean as linear combination of weights (W) and predictor variable (X) and the standard deviation of σ. **Linear regression and ANOVA models represent the general linear models**. The diagram given below represents the same in form of simple linear regression model where there is just one coefficient.

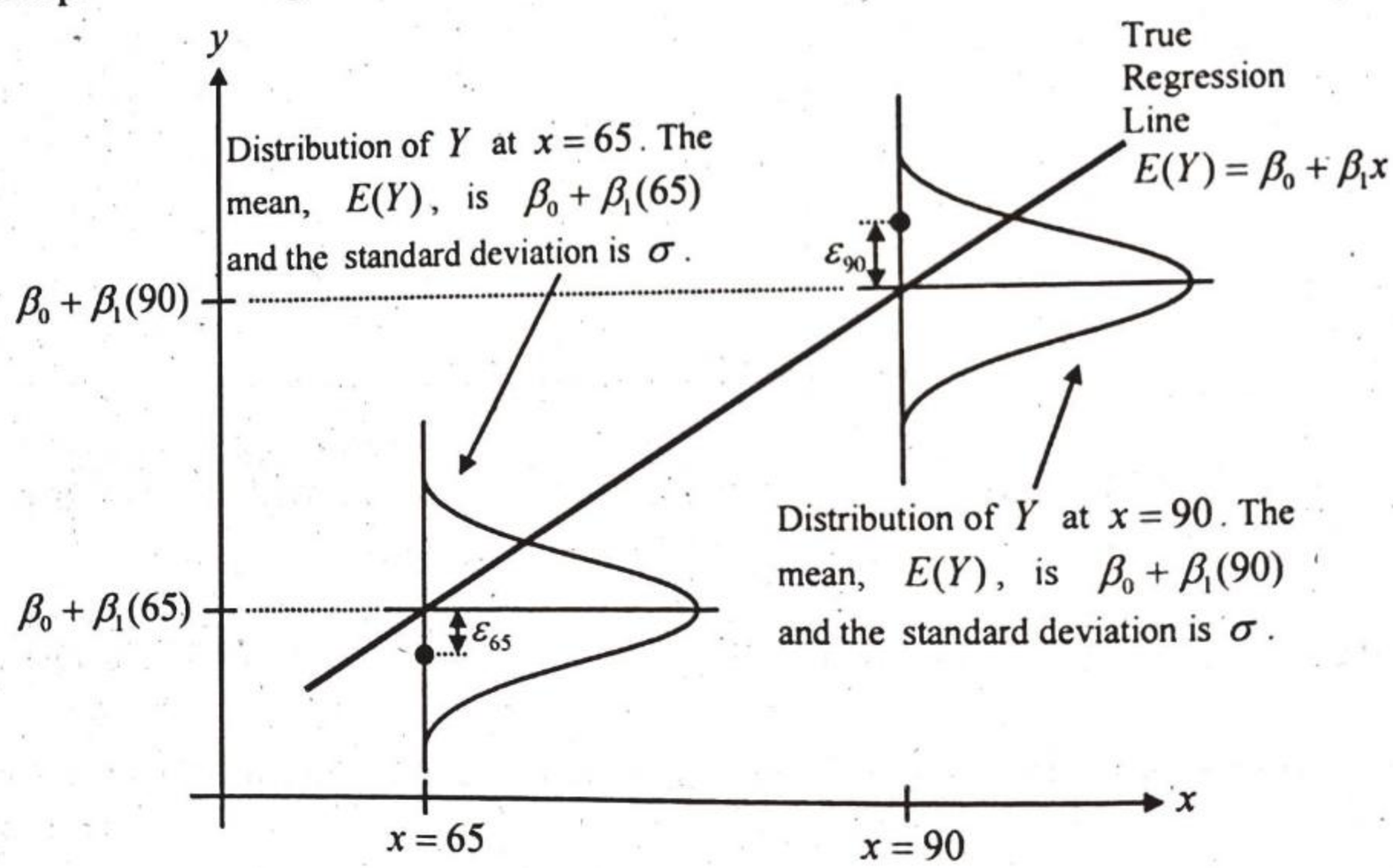


Fig: 1 Response variable and error term follows normal distribution

For linear regression models, the link function is identity function. Recall that a link function transforms the probabilities of the levels of a categorical response variable to a continuous scale that is unbounded. Once the transformation is complete, the relationship between the predictors and the response can be modelled with linear regression. Thus, the linear combination of weights and predictor variable is modelled as output. The linear

Scanned by TapScanner

regression models using identity function as link function can be understood as the following:

$$Y_{\text{actual}} = Y_{\text{predicted}} + \varepsilon ....$$

The actual value is the sum of predicted value and the random error term (which can be on either side of the mean value – response value)

$$Y_{\text{actual}} = \sum_{i=1}^{n} \beta_i X_i + \varepsilon$$

As part of training regression models, one must understand that what is actually modelled is the mean of the response variable values and not the actual values. As the response variable Y follows normal distribution, the summation of weights and predictor variable can be equated as the **expected value of** Y.

$$E(Y) = \sum_{i=1}^{n} \beta_i X_i$$

The actual value of Y can be represented as the following in terms of outcome from regression model and the random error term:

$$Y_{\text{actual}} = E(Y) + \varepsilon$$

For the linear regression model, the **identity function** is **link function** used to link the mean of expected value of response variable, Y and the summation of weights and predictor variable thus, the $g(E(Y))$ **becomes** $E(Y)$ **which is represented as** $Y_{\text{predicted}}$. Thus, linear regression model (also, at times termed as **general linear models**) is represented as the following:

$$Y_{\text{predicted}} = \sum_{i=1}^{n} \beta_i X_i$$

Given above, lets understand what are generalized linear models.
In generalized linear models, the link function used to model the response variable as a function of the predictor variables are the following. Note that the Y represents the mean or expected value of the response variable.

- **Log Link:** $\log(Y)$ – Models the logarithm of mean Y. Thus, the regression model can be represented as the following:

  $$\log(Y) = \sum_{i=1}^{n} \beta_i x_i ...$$

  This type of regression model is called as **Poisson regression**. It is used to model the non-negative count value as in Poisson probability distribution.

- **Logit Link:** $\log\left(\frac{Y}{1-Y}\right)$ – Models the logarithm of odds of the probability of binary outcome. Thus, the regression model can be represented as the following:

  $$\log\left(\frac{Y}{1-Y}\right) = \sum_{i=1}^{n} \beta_i x_i ...$$

Scanned by TapScanner

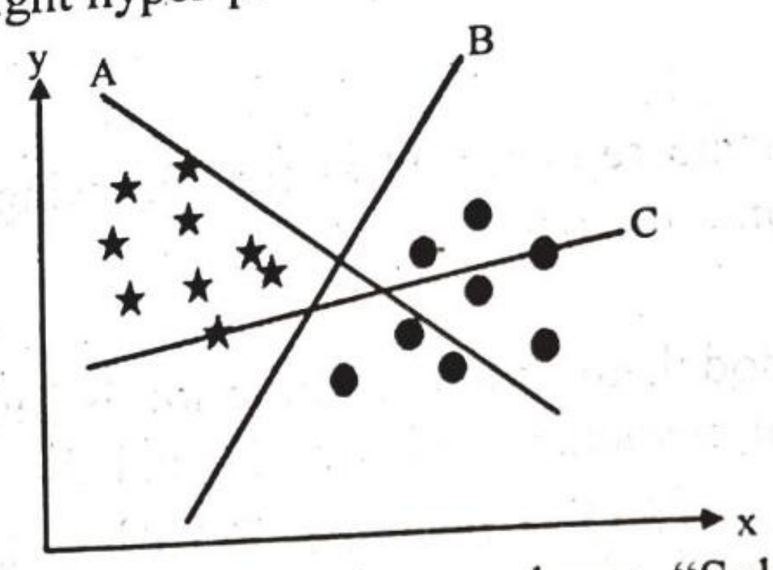
- This type of regression model is called as **Logistic regression**. it is used to model the probability of the binary outcome.
- There are other link functions which result in modelling response variable based on **gamma distribution** (GammaRegressor). **Tweedie distribution** (TweedieRegressor) etc.

The above regression models used for modelling response variable with Poisson, Gamma, Tweedie distribution etc. are called as **Generalized Linear Models (GLM).**

14. How does support vector machine work? **[MODEL QUESTION]**

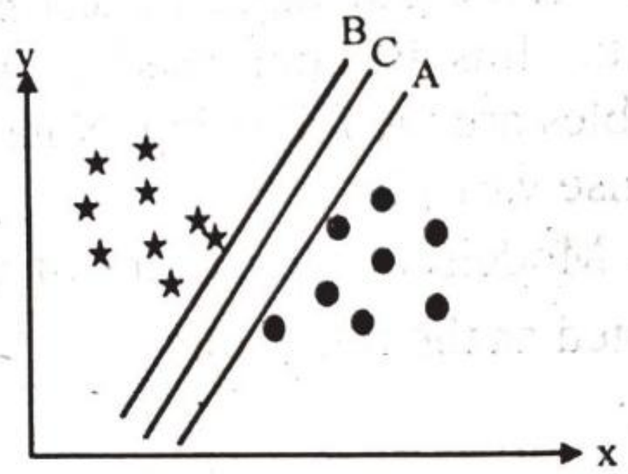
Answer:

Identify the right hyper-plane (Scenario-1): Here, we have three hyper-planes (A, B, and C). Now, identify the right hyper-plane to classify stars and circles.

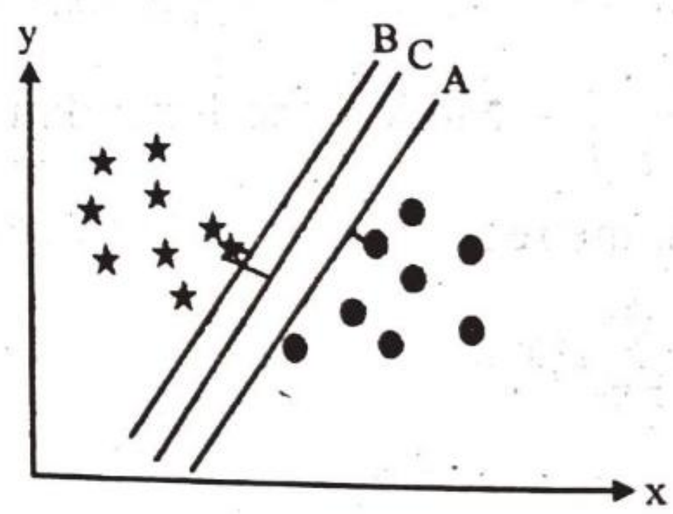


Here a thumb rule to identify the right hyper-plane: "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.

Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B, and C) and all are segregating the classes well. Now, how can we identify the right hyper-plane?



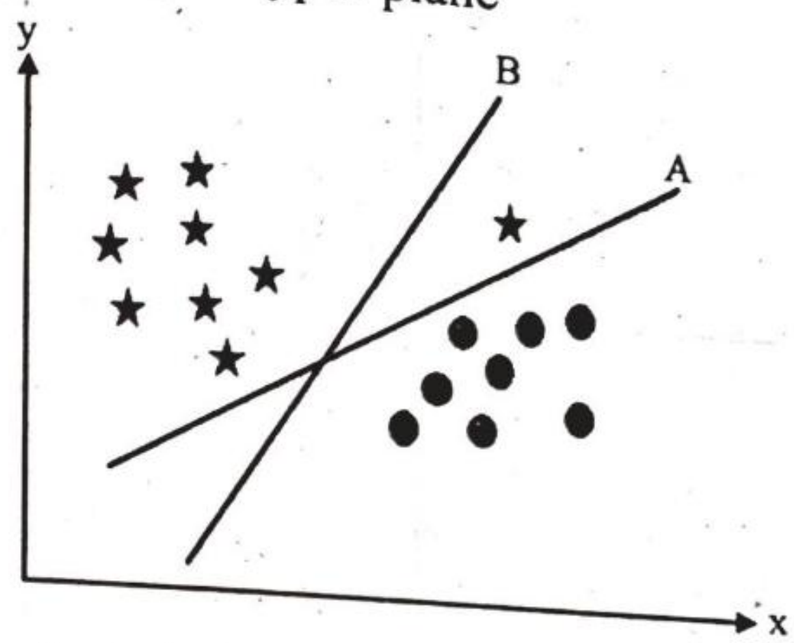
Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let's look at the below snapshot:



Scanned by TapScanner

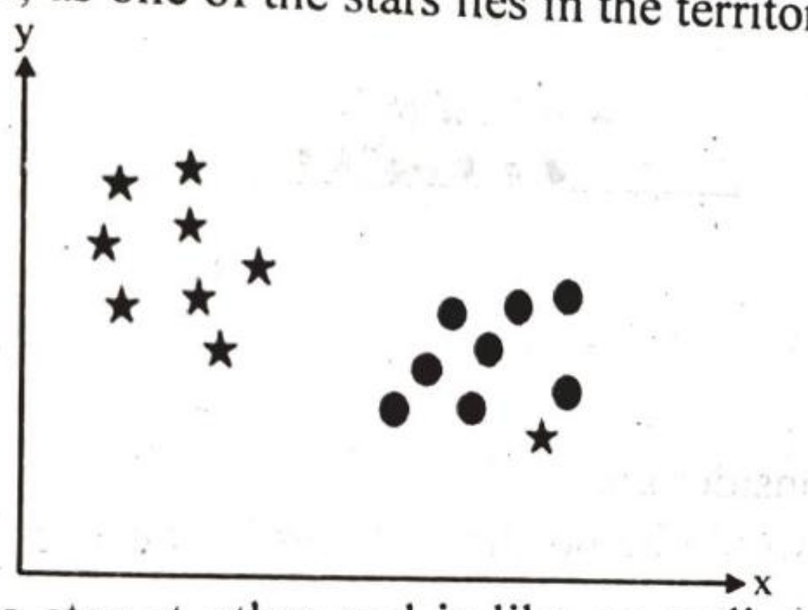
Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

Identify the right hyper-plane (Scenario-3): Hint: Use the rules as discussed in previous section to identify the right hyper-plane

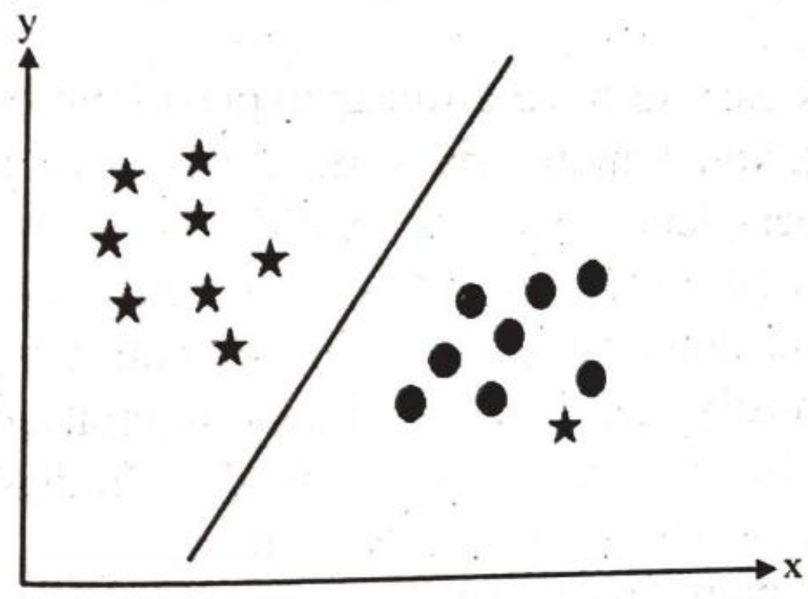


Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

Can we classify two classes (Scenario-4)?: Below, we are unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.



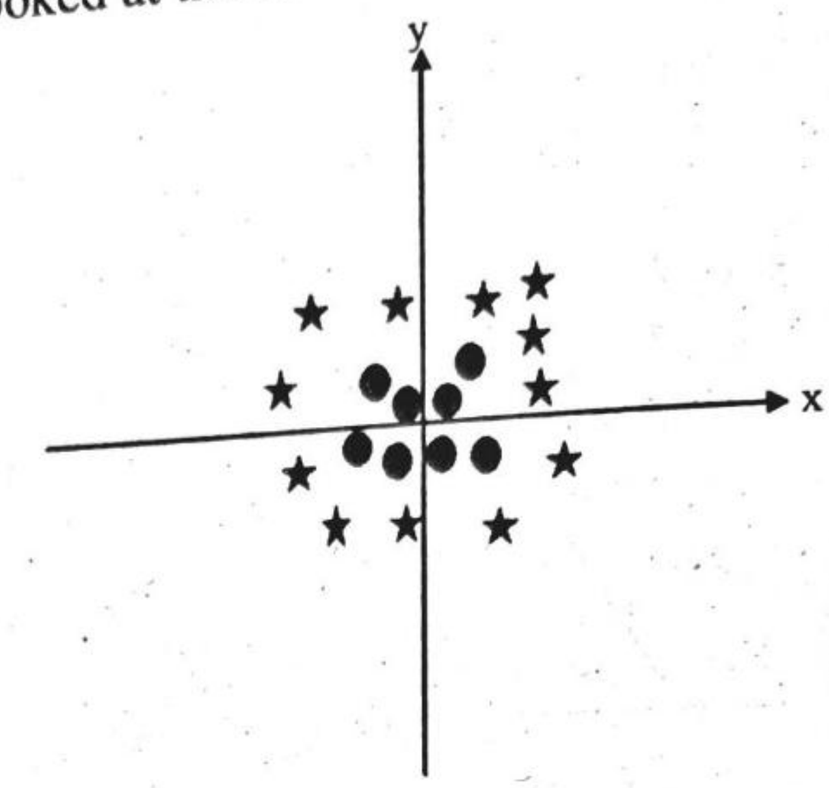
As already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



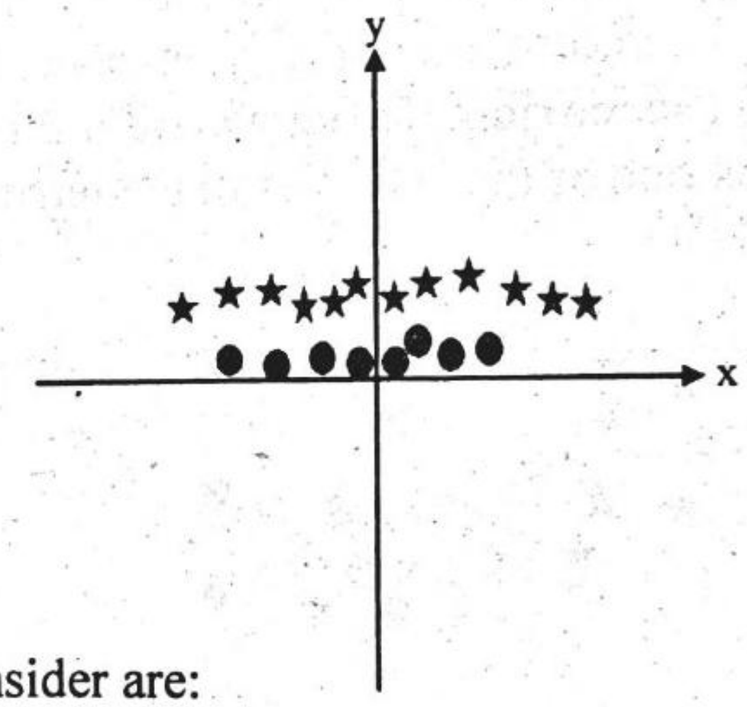
Scanned by TapScanner

Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes?

Till now, we have only looked at the linear hyper-plane.



SVM can solve this problem. It solves this problem by introducing additional feature. Here, we will add a new feature $z = x^2 + y^2$. Now, let's plot the data points on axis x and z:



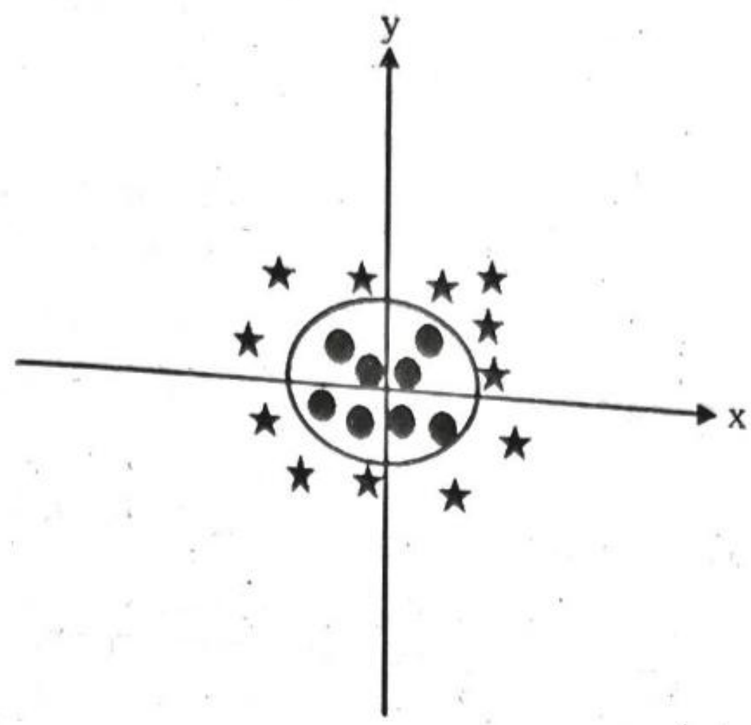
In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y.
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the kernel **trick**. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e., it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs we've defined.

When we look at the hyper-plane in original input space it looks like a circle:

Scanned by TapScanner



15. Give some examples of Generalized Linear Model (GLM). **[MODEL QUESTION]**

Answer:

Here is some real-world examples where generalized linear models can be used to predict continuous response variables based on their probability distribution. The table consists of reference to the SKlearn class which can be used to model the response variables.

Problem Type	Example	Regression Model (Sklearn)
Agriculture / weather modeling	Number of rain events per year	PoissonRegressor
Agriculture / weather modeling	Amount of rainfall per rainfall event	GammaRegressor
Agriculture / weather modeling	Total rainfall per year	TweedieRegressor
Risk modeling / insurance policy pricing	No of claim events / policyholder per year	PoissonRegressor
Risk modeling / insurance policy pricing	Cost per claim event	GammaRegressor
Risk modeling / insurance policy pricing	Total cost per policyholder	TweedieRegressor
Predictive maintenance	Number of production interruption events per year	PoissonRegressor
Predictive maintenance	Duration of interruption	GammaRegressor
Predictive maintenance	Total interruption time per year	TweedieRegressor

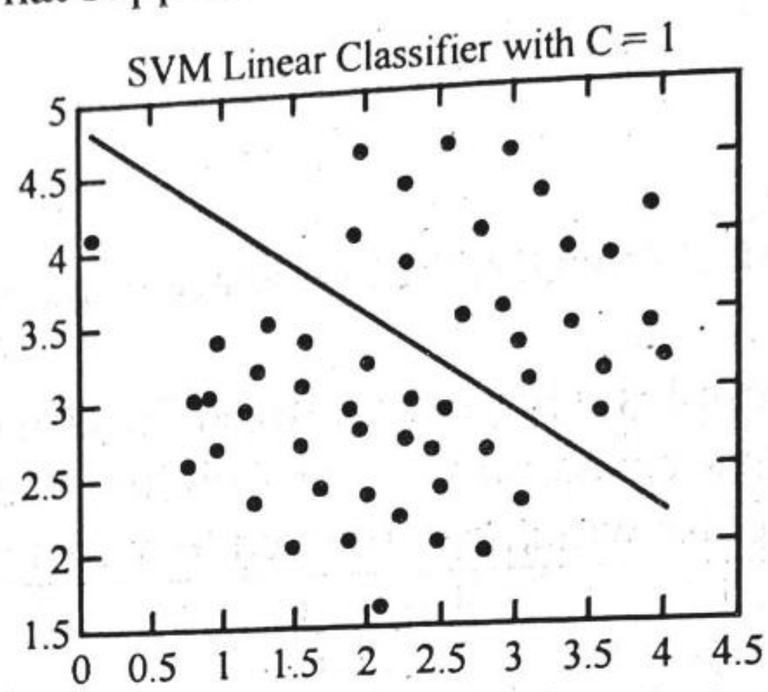
16. Explain the need for kernel method and its working. **[MODEL QUESTION]**

Answer:

Kernels or kernel methods (also called Kernel functions) are sets of different types of algorithms that are being used for pattern analysis. They are used to solve a non-linear problem by using a linear classifier. Kernels Methods are employed in SVM (Support

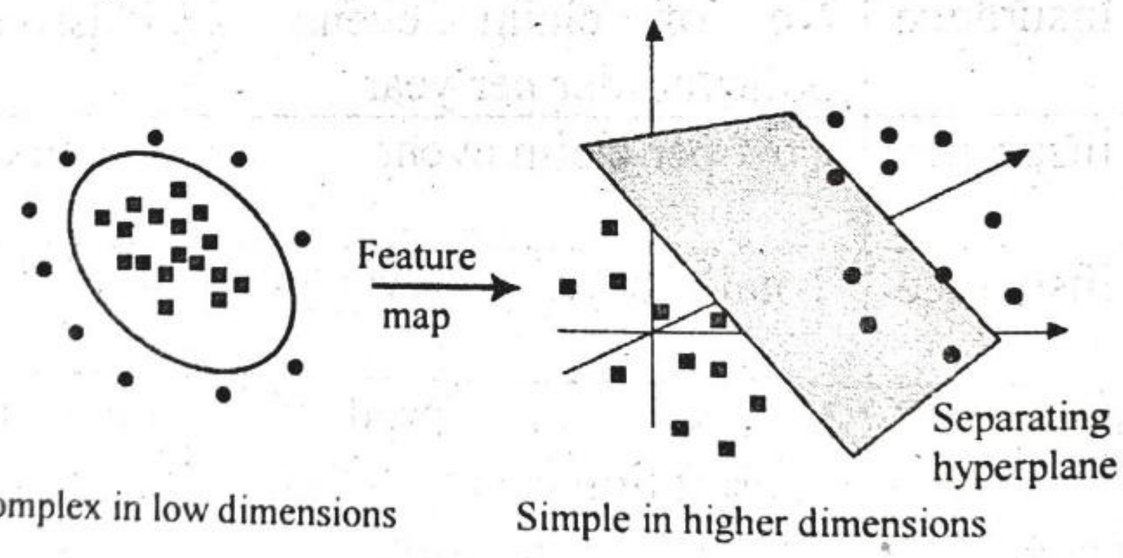
Scanned by TapScanner

Vector Machines) which are used in classification and regression problems. The SVM uses what is called a "Kernel Trick" where the data is transformed and an optimal boundary is found for the possible outputs.
Before we get into the working of the Kernel Methods, it is more important to understand support vector machines or the SVMs because kernels are implemented in SVM models. So, Support Vector Machines are supervised machine learning algorithms that are used in classification and regression problems such as classifying an apple to class fruit while classifying a Lion to the class animal.
To demonstrate, below is what support vector machines look like:



Here we can see a hyperplane which is separating green dots from the blue ones. A hyperplane is one dimension less than the ambient plane. E.g. in the above figure, we have 2 dimension which represents the ambient space but the lone which divides or classifies the space is one dimension less than the ambient space and is called hyperplane.
But what if we have input like this:

Separation may be easier in higher dimensions



It is very difficult to solve this classification using a linear classifier as there is no good linear line that should be able to classify the red and the green dots as the points are randomly distributed. Here comes the use of kernel function which takes the points to higher dimensions, solves the problem over there and returns the output. Think of this in this way, we can see that the green dots are enclosed in some perimeter area while the red one lies outside it, likewise, there could be other scenarios where green dots might be distributed in a trapezoid-shaped area.
So what we do is to convert the two-dimensional plane which was first classified by one-dimensional hyperplane ("or a straight line") to the three-dimensional area and here our

Scanned by TapScanner

classifier i.e., hyperplane will not be a straight line but a two-dimensional plane which will cut the area.

In order to get a mathematical understanding of kernel, let us understand the Lili Jiang's equation of kernel which is:

$$K(x, y) = \langle f(x), f(y) \rangle$$

where, K is the kernel function, x and y are the dimensional inputs, f is the map from n-dimensional to m-dimensional space and, $\langle x, y \rangle$ is the dot product.

Let us say that we have two points, $x = (2, 3, 4)$ and $y = (3, 4, 5)$

As we have seen, $K(x, y) = \langle f(x), f(y) \rangle$.

Let us first calculate $\langle f(x), f(y) \rangle$

$$f(x) = (x_1x_1, x_1x_2, x_1x_3, x_2x_1, x_2x_2, x_2x_3, x_3x_1, x_3x_2, x_3x_3)$$

$$f(y) = (y_1y_1, y_1y_2, y_1y_3, y_2y_1, y_2y_2, y_2y_3, x_3x_1, x_3x_2, x_3x_3)$$

So, $f(2, 3, 4) = (4, 6, 8, 6, 9, 12, 8, 12, 16)$ and

$$f(3, 4, 5) = (9, 12, 15, 12, 16, 20, 15, 20, 25)$$

So the dot product,

$$f(x) \cdot f(y) = f(2, 3, 4) \cdot f(3, 4, 5)$$

$$= (36 + 72 + 120 + 72 + 144 + 240 + 120 + 240 + 400) = 1444$$

And, $K(x, y) = (2 \times 3 + 3 \times 4 + 4 \times 5)^2 = (6 + 12 + 20)^2 = 38 \times 38 = 1444$

This as we find out, $f(x) \cdot f(y)$ and $K(x, y)$ give us the same result, but the former method required a lot of calculations (because of projecting 3 dimensions into 9 dimensions) while using the kernel, it was much easier.

17. What are types of kernel and methods in SVM? **[MODEL QUESTION]**

Answer:

Let us see some of the kernel function or the types that are being used in SVM:

1. Liner Kernel

Let us say that we have two vectors with name x_1 and y_1, then the linear kernel is defined by the dot product of these two vectors:

$$K(x_1, x_2) = x_1 \cdot x_2$$

2. Polynomial Kernel

A polynomial kernel is defined by the following equation:

$$K(x_1, x_2) = (x_1 \cdot x_2 + 1)d$$

where, d is the degree of the polynomial and x_1 and x_2 are vectors

3. Gaussian Kernel

This kernel is an example of a radial basis function kernel. Below is the equation for this:

$$k(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2)$$

Scanned by TapScanner

The given sigma plays a very important role in the performance of the Gaussian kernel and should neither be overestimated and nor be underestimated, it should be carefully tuned according to the problem.

4. Exponential Kernel

This is in close relation with the previous kernel i.e., the Gaussian kernel with the only difference is – the square of the norm is removed.

The function of the exponential function is:

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{2\sigma^2}\right)$$

This is also a radial basis kernel function.

5. Laplacian Kernel

This type of kernel is less prone for changes and is totally equal to previously discussed exponential function kernel, the equation of Laplacian kernel is given as:

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma}\right)$$

6. Hyperbolic or the Sigmoid Kernel

This kernel is used in neural network areas of machine learning. The activation function for the sigmoid kernel is the bipolar sigmoid function. The equation for the hyperbolic kernel function is:

$$k(x, y) = \tanh\left(\alpha x^T y + c\right)$$

This kernel is very much used and popular among support vector machines.

7. Anova radial basis kernel

This kernel is known to perform very well in multidimensional regression problems just like the Gaussian and Laplacian kernels. This also comes under the category of radial basis kernel.

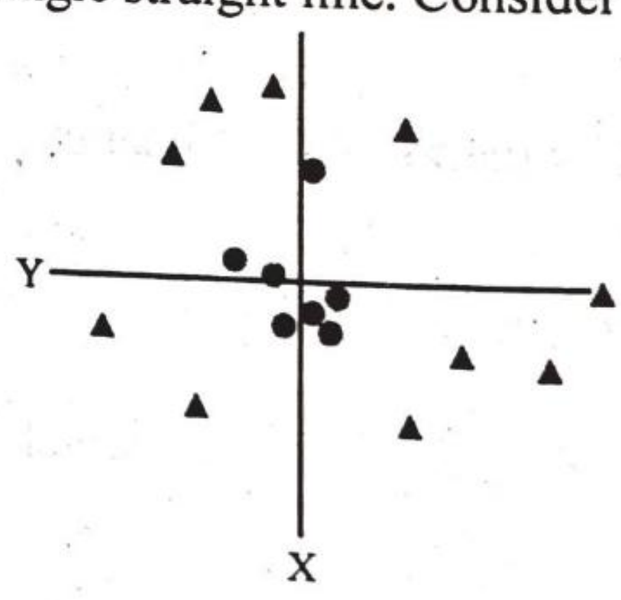
The equation for Anova kernel is

$$k(x, y) = \sum_{k=1}^{n} \exp\left(-\sigma\left(x^k - y^k\right)^2\right)^d$$

18. Explain non linear SVM in short. **[MODEL QUESTION]**

Answer:

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

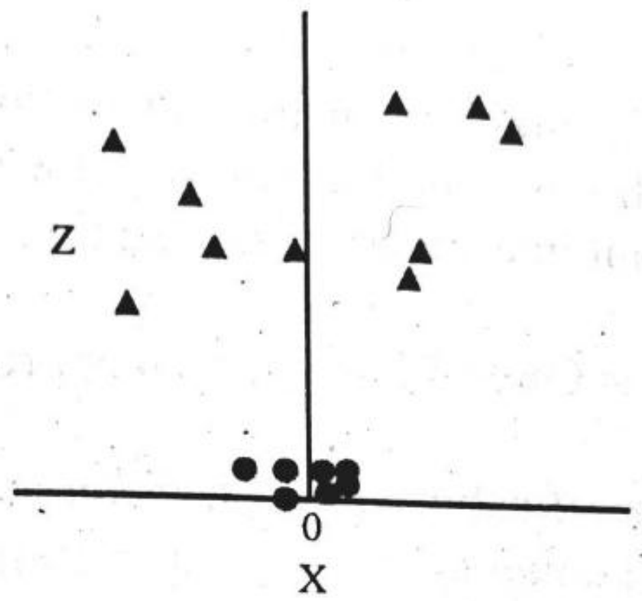


Scanned by TapScanner

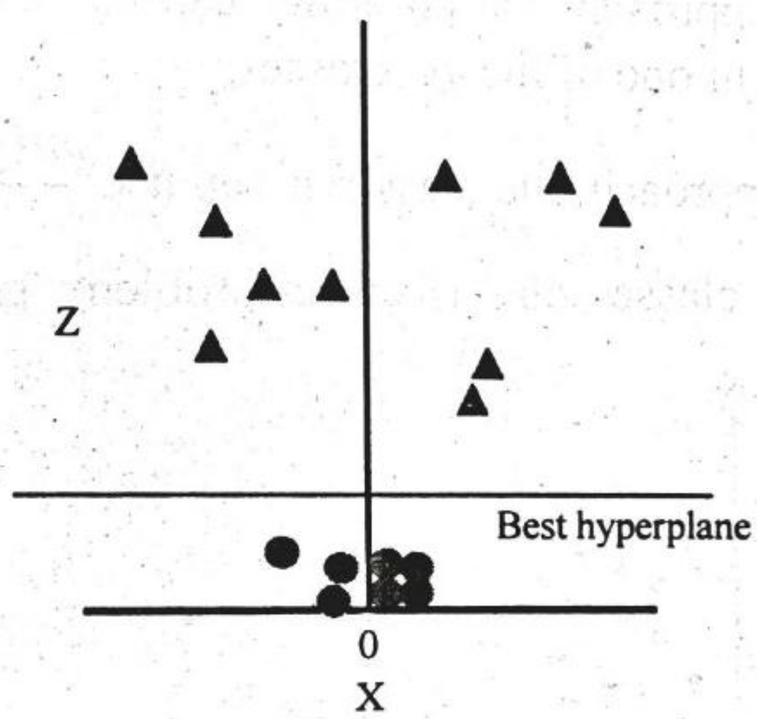
So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

$$z = x_2 + y_2$$

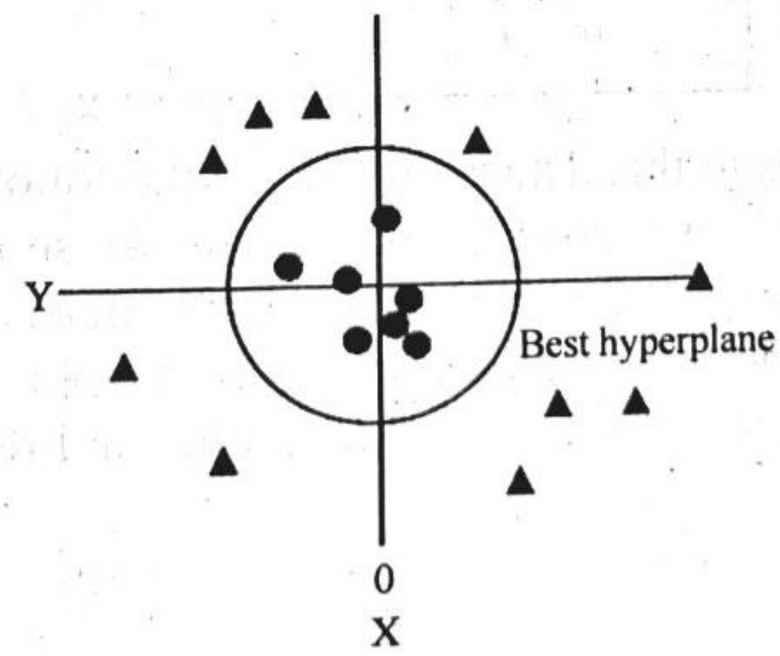
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence we get a circumference of radius 1 in case of non-linear data.

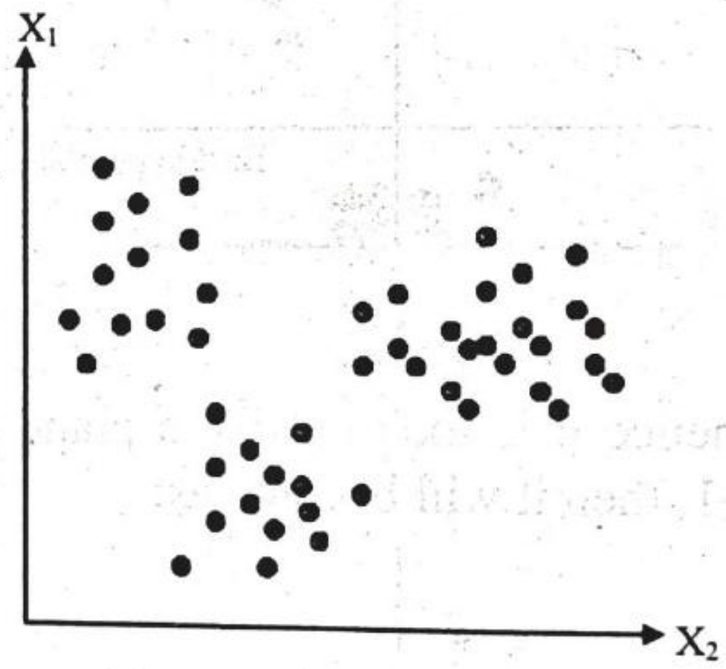
Scanned by TapScanner

19. How we can implement multiclass classification using SVM? **[MODEL QUESTION]**

Answer:
In its most simple type, SVM doesn't support multiclass classification natively. It supports binary classification and separating data points into two classes. For multiclass classification, the same principle is utilized after breaking down the multi-classification problem into multiple binary classification problems.
The idea is to map data points to high dimensional space to gain mutual linear separation between every two classes. This is called a **One-to-One** approach, which breaks down the multiclass problem into multiple binary classification problems. A binary classifier per each pair of classes.
Another approach one can use is **One-to-Rest**. In that approach, the breakdown is set to a binary classifier per each class.
A single SVM does binary classification and can differentiate between two classes. So that, according to the two breakdown approaches, to classify data points from m classes data set:

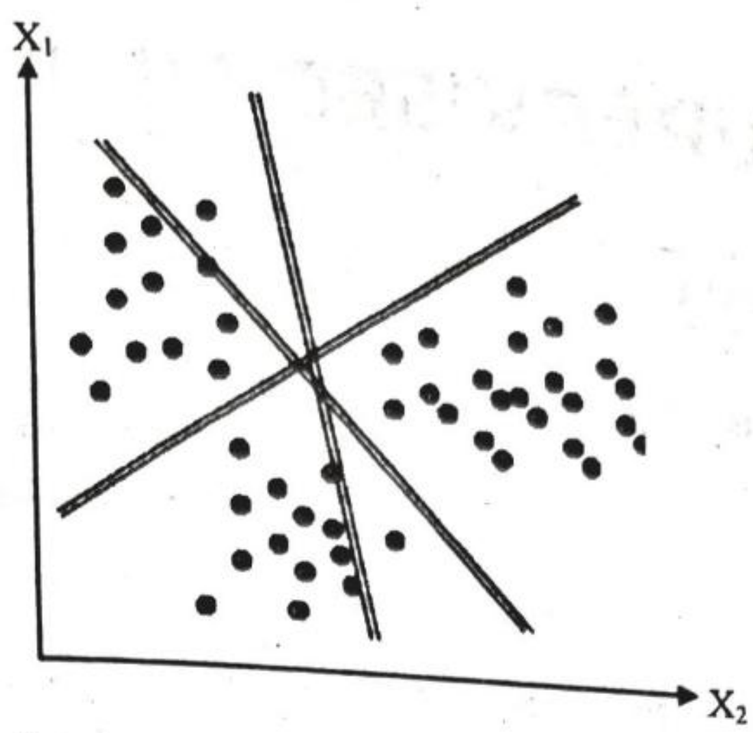
- In the One-to-Rest approach, the classifier can use m SVMs. Each SVM would predict membership in one of the m classes.
- In the One-to-One approach, the classifier can use $\frac{m(m-1)}{2}$ SVMs.

Let's take an example of 3 classes classification problem; green, red, and blue, as the following image:

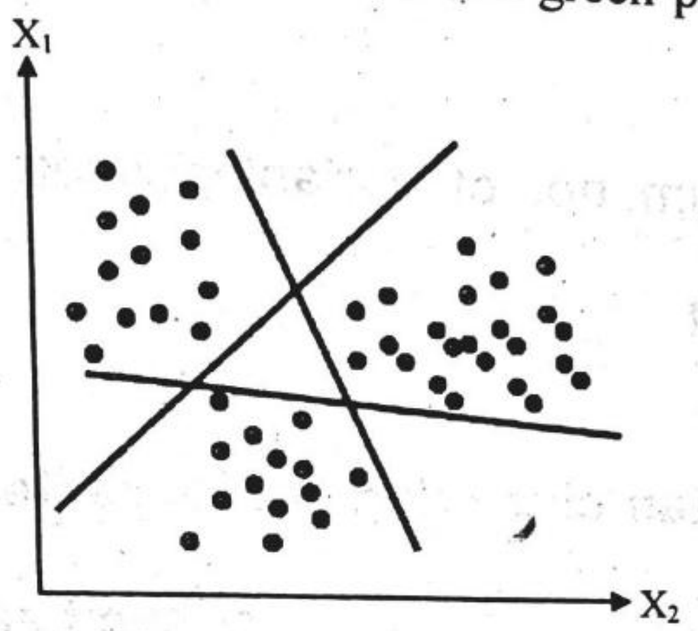


Applying the two approaches to this data set results in the followings:
In the **One-to-One** approach, we need a hyperplane to separate between every two classes, neglecting the points of the third class. This means the separation takes into account only the points of the two classes in the current split. For example, the red-blue line tries to maximize the separation only between blue and red points. It has nothing to do with green points:

Scanned by TapScanner



In the **One-to-Rest** approach, we need a hyperplane to separate between a class and all others at once. This means the separation takes all points into account, dividing them into two groups; a group for the class points and a group for all other points. For example, the green line tries to maximize the separation between green points and all other points at once:



One of the most common real-world problems for multiclass classification using SVM is text classification. For example, classifying news articles, tweets, or scientific papers.

Scanned by TapScanner

UNSUPERVISED LEARNING

Multiple Choice Type Questions

1. Which of the following is NOT supervised learning? [MODEL QUESTION]
   a) PCA
   b) Decision Tree
   c) Linear Regression
   d) Naive Bayesian

Answer: (a)

2. Suppose we would like to perform clustering on spatial data such as the geometrical locations of houses. We wish to produce clusters of many different sizes and shapes. Which of the following methods is the most appropriate? [MODEL QUESTION]
   a) Decision Trees
   b) Density-based clustering
   c) Model-based clustering
   d) K-means clustering

Answer: (b)

3. What is the minimum no. of variables/ features required to perform clustering? [MODEL QUESTION]
   a) 0
   b) 1
   c) 2
   d) 3

Answer: (b)

4. For two runs of K-Mean clustering is it expected to get same clustering results? [MODEL QUESTION]
   a) yes
   b) no
   c) can't say
   d) sometimes yes

Answer: (b)

5. Is it possible that Assignment of observations to clusters does not change between successive iterations in K-Means [MODEL QUESTION]
   a) yes
   b) no
   c) can't say
   d) sometimes yes

Answer: (a)

6. Which of the following can act as possible termination conditions in K-Means? [MODEL QUESTION]
   1. For a fixed number of iterations.
   2. Assignment of observations to clusters does not change between iterations. Except for cases with a bad local minimum.
   3. Centroids do not change between successive iterations.
   4. Terminate when RSS falls below a threshold.

   a) 1, 3 and 4
   b) 1, 2 and 3
   c) 1, 2 and 4
   d) All of these

Answer: (d)

Scanned by TapScanner

7. Which of the following algorithm is most sensitive to outliers? [MODEL QUESTION]

a) K-means clustering algorithm
b) K-medians clustering algorithm
c) K-modes clustering algorithm
d) K-medoids clustering algorithm

Answer: (a)

8. In which of the following cases will K-Means clustering fail to give good results? [MODEL QUESTION]

1. Data points with outliers
2. Data points with different densities
3. Data points with round shapes
4. Data points with non-convex shapes

a) 1 and 2
b) 2 and 3
c) 2 and 4
d) 1, 2 and 4

Answer: (d)

9. PCA reduces the dimension by finding a few_____. [MODEL QUESTION]

a) Hexagonal linear combination
b) Orthogonal linear combinations
c) Octagonal linear combination
d) Pentagonal Linear Combination

Answer: (b)

10. PCA is a _____. [MODEL QUESTION]

a) Non linear method
b) Linear method
c) Continuous method
d) Repeated method

Answer: (b)

11. PCA is used to find _____. [MODEL QUESTION]

a) Relationship between components
b) Linear regression
c) Linear relation
d) Inter relation

Answer: (d)

12. _____ basically known as characteristic roots. It basically measures the variance in all variables which is accounted for by that factor [MODEL QUESTION]

a) Eigen value
b) Eigen vector
c) Linear value
d) None of these

Answer: (a)

13. _____is a dimensionality reduction technique which is commonly used for the supervised classification problems. [MODEL QUESTION]

a) Value analysis
b) Function Analysis
c) Pure analysis
d) None of these

Answer: (b)

14. Which of the following is a reasonable way to select the number of principal components "k"? [MODEL QUESTION]

a) Choose k to be the smallest value so that at least 99% of the variance is retained
b) Choose k to be 99% of m (k = 0.99*m, rounded to the nearest integer)
c) Choose k to be the largest value so that 99% of the variance is retained
d) Use the elbow method

Answer: (a)

Scanned by TapScanner

15. In which of the following cases will K-means clustering fail to give good results? 1) Data points with outliers 2) Data points with different densities 3) Data points with nonconvex shapes [MODEL QUESTION]
a) 1 and 2 b) 2 and 3 c) 1, 2, and 3 d) 1 and 3

Answer: (c)

16. How can you prevent a clustering algorithm from getting stuck in bad local optima? [MODEL QUESTION]
a) Set the same seed value for each run b) Use multiple random initializations
c) Both A and B d) None of these

Answer: (b)

17. In Which Machine Learning Algorithm, No labels are given to the learning algorithm, leaving it on its own [MODEL QUESTION]
a) Supervised b) Reinforcement
c) Semi Supervised d) Unsupervised

Answer: (d)

18. Which of the following is required by K-means clustering? [MODEL QUESTION]
a) defined distance metric b) number of clusters
c) initial guess as to cluster centroids d) all of the mentioned

Answer: (d)

19. Point out the wrong statement. [MODEL QUESTION]
a) k-means clustering is a method of vector quantization
b) k-means clustering aims to partition n observations into k clusters
c) k-nearest neighbor is same as k-means
d) none of the mentioned

Answer: (c)

20. PCA is [MODEL QUESTION]
a) forward feature selection b) backword feature selection
c) feature extraction d) all of these

Answer: (c)

Short Answer Type Questions

1. What is unsupervised learning? [MODEL QUESTION]

Answer:

Unsupervised learning, also known as 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.

Scanned by TapScanner

2. What do you understand by dimensionality reduction? **[MODEL QUESTION]**

Answer:

The number of input variables or features for a dataset is referred to as its dimensionality. Dimensionality reduction refers to techniques that reduce the number of input variables in a dataset.

More input features often make a predictive modeling task more challenging to model, more generally referred to as the curse of dimensionality.

High-dimensionality statistics and dimensionality reduction techniques are often used for data visualization. Nevertheless these techniques can be used in applied machine learning to simplify a classification or regression dataset in order to better fit a predictive model.

In this post, you will discover a gentle introduction to dimensionality reduction for machine learning

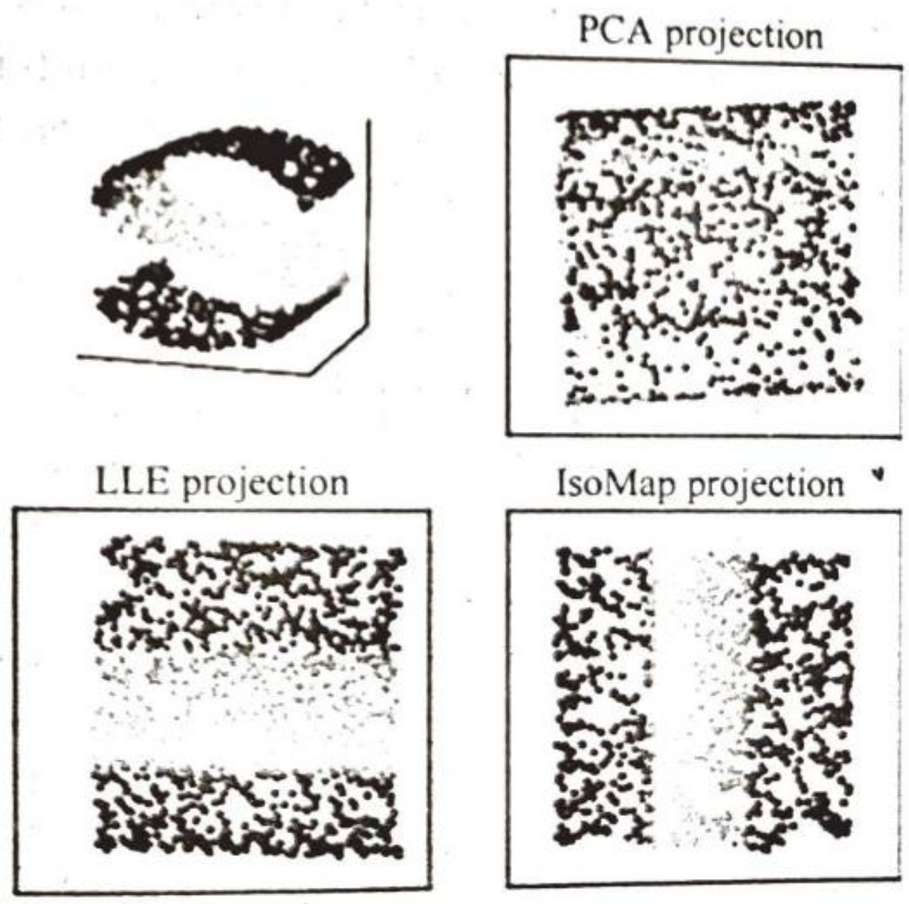
After reading this post, you will know:

- Large numbers of input features can cause poor performance for machine learning algorithms.
- Dimensionality reduction is a general field of study concerned with reducing the number of input features.
- Dimensionality reduction methods include feature selection, linear algebra methods, projection methods, and auto encoders.

3. What is non-linear dimensionality reduction? **[MODEL QUESTION]**

Answer:

Sometimes the structure of the data is non-linear, and the principle components will not give us the optimal dimensionality reduction, so we use non-linear methods like KPCA. For example in the figure below, PCA will look for the surface that will cross the S shaped data with minimal reconstruction error, but no matter what surface we choose, there will be some loss in information (Some points from different colors are mapped to one point. Unlike PCA, non-linear methods will give us more optimal projections (last two figures).



Scanned by TapScanner

Kernel Principal Component Analysis (KPCA) is a non-linear dimensionality reduction technique. It is an extension of Principal Component Analysis (PCA) - which is a linear dimensionality reduction technique - using kernel methods.

Long Answer Type Questions

1. Explain clustering technique of unsupervised learning. **[MODEL QUESTION]**

Answer:
Unsupervised learning models are utilized for three main tasks—clustering, association, and dimensionality reduction. Below we'll define each learning method and highlight common algorithms and approaches to conduct them effectively.

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.

Exclusive and Overlapping Clustering:
Exclusive clustering is a form of grouping that stipulates a data point can exist only in one cluster. This can also be referred to as "hard" clustering. The K-means clustering algorithm is an example of exclusive clustering.
– **K-means clustering** is a common example of an exclusive clustering method where data points are assigned into K groups, where K represents the number of clusters based on the distance from each group's centroid. The data points closest to a given centroid will be clustered under the same category. A larger K value will be indicative of smaller groupings with more granularity whereas a smaller K value will have larger groupings and less granularity. K-means clustering is commonly used in market segmentation, document clustering, image segmentation and image compression.
Overlapping clusters differs from exclusive clustering in that it allows data points to belong to multiple clusters with separate degrees of membership. "Soft" or fuzzy k-means clustering is an example of overlapping clustering.

Hierarchical clustering:
Hierarchical clustering, also known as hierarchical cluster analysis (HCA), is an unsupervised clustering algorithm that can be categorized in two ways; they can be agglomerative or divisive. Agglomerative clustering is considered a "bottoms-up approach". Its data points are isolated as separate groupings initially and then they are merged together iteratively on the basis of similarity until one cluster has been achieved. Four different methods are commonly used to measure similarity:

1. **Ward's linkage:** This method states that the distance between two clusters is defined by the increase in the sum of squared after the clusters are merged.

Scanned by TapScanner

2. **Average linkage:** This method is defined by the mean distance between two points in each cluster
3. **Complete (or maximum) linkage:** This method is defined by the maximum distance between two points in each cluster
4. **Single (or minimum) linkage:** This method is defined by the minimum distance between two points in each cluster

Euclidean distance is the most common metric used to calculate these distances; however, other metrics, such as Manhattan distance, are also cited in clustering literature. Divisive clustering can be defined as the opposite of agglomerative clustering; instead it takes a "top-down" approach. In this case, a single data cluster is divided based on the differences between data points. Divisive clustering is not commonly used, but it is still worth noting in the context of hierarchical clustering. These clustering processes are usually visualized using a dendrogram, a tree-like diagram that documents the merging or splitting of data points at each iteration.

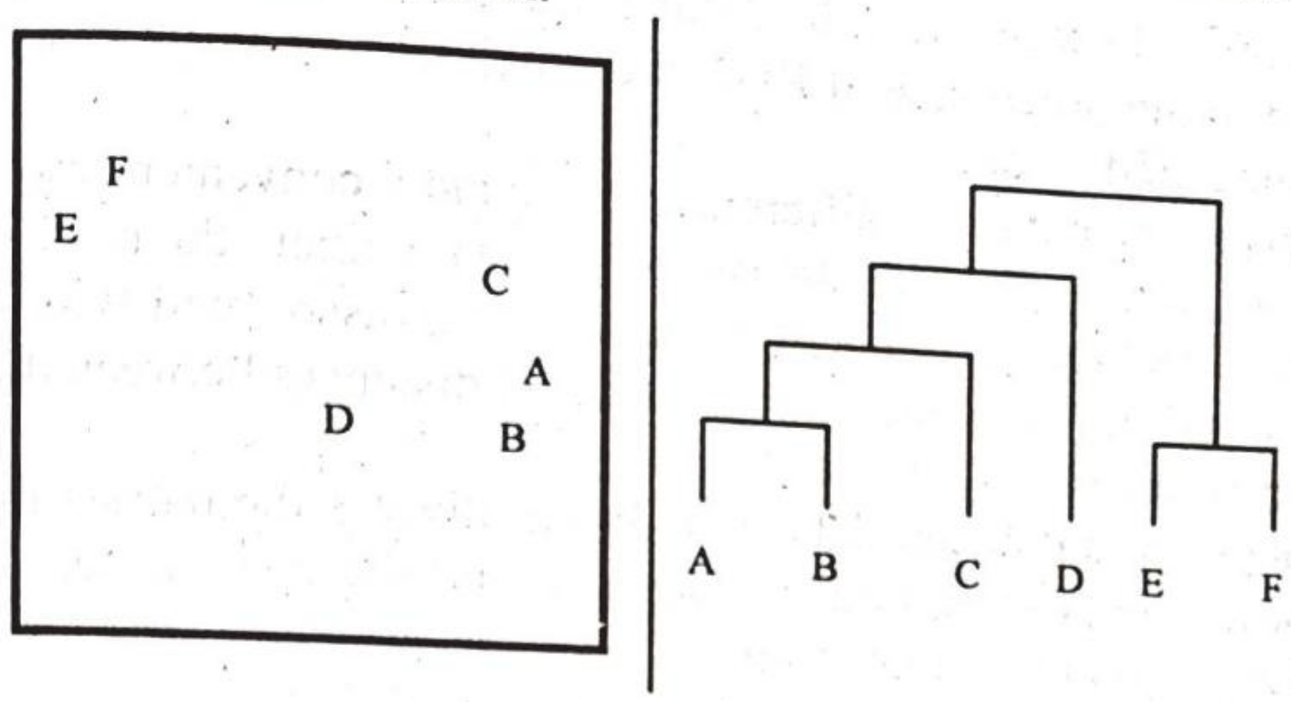


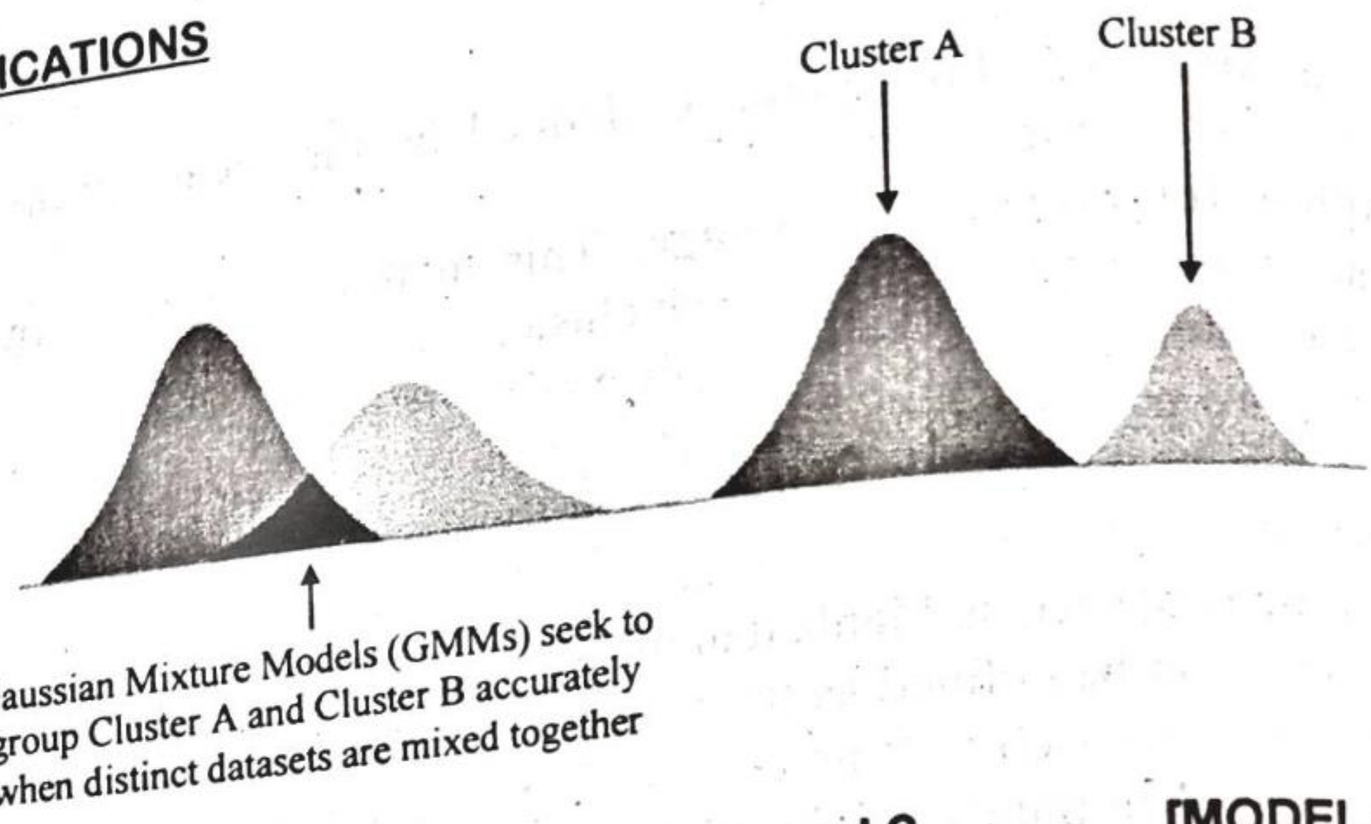
Diagram of a Dendrogram; reading the chart "botton-up" demonstrates agglomerative clustering while "top-down" is indicative of divisive clustering

Probabilistic clustering:

A probabilistic model is an unsupervised technique that helps us solve density estimation or "soft" clustering problems. In probabilistic clustering, data points are clustered based on the likelihood that they belong to a particular distribution. The Gaussian Mixture Model (GMM) is the one of the most commonly used probabilistic clustering methods.

- **Gaussian Mixture Models** are classified as mixture models, which means that they are made up of an unspecified number of probability distribution functions. GMMs are primarily leveraged to determine which Gaussian, or normal, probability distribution a given data point belongs to. If the mean or variance are known, then we can determine which distribution a given data point belongs to. However, in GMMs, these variables are not known, so we assume that a latent, or hidden, variable exists to cluster data points appropriately. Which it is not required to use the Expectation-Maximization (EM) algorithm, it is a commonly used to estimate the assignment probabilities for a given data point to a particular data cluster.

Scanned by TapScanner



2. What is K-Means algorithm and how does it work? **[MODEL QUESTION]**

Answer:

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 algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

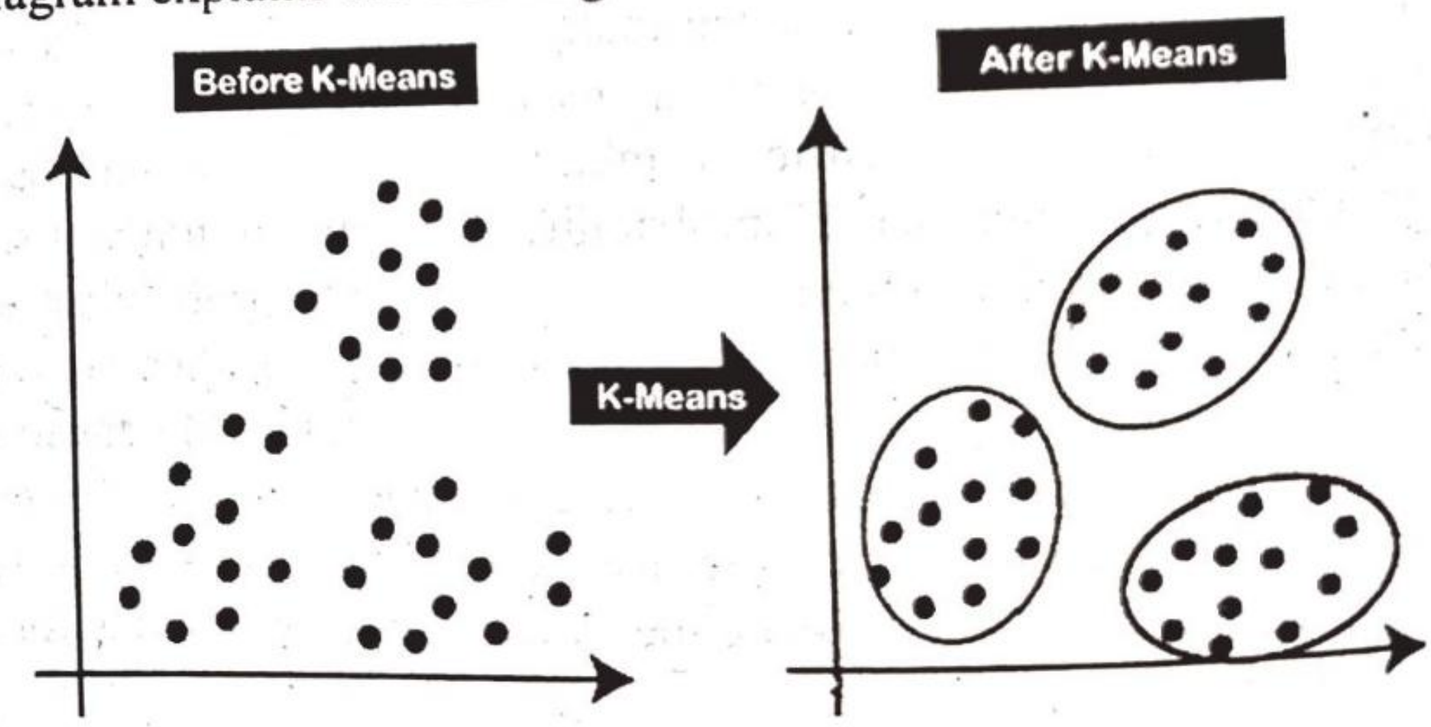
The k-means clustering algorithm mainly performs two tasks:

Determines the best value for K center points or centroids by an iterative process.

Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



Scanned by TapScanner

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.

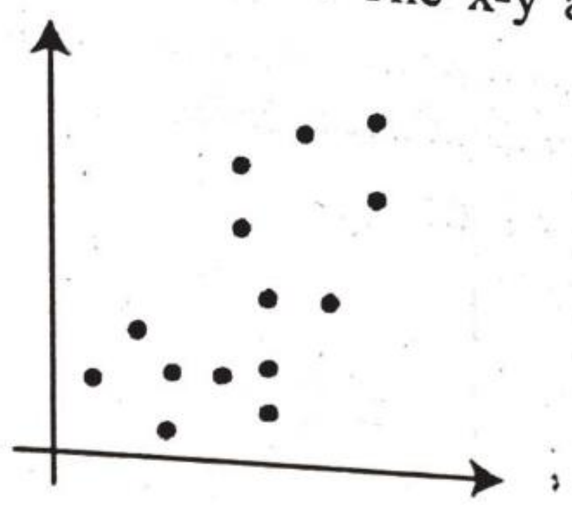
Step-4: Calculate the variance and place a new centroid of each cluster.

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.

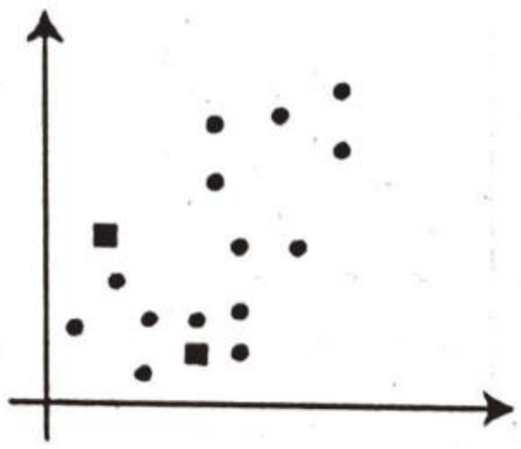
Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:



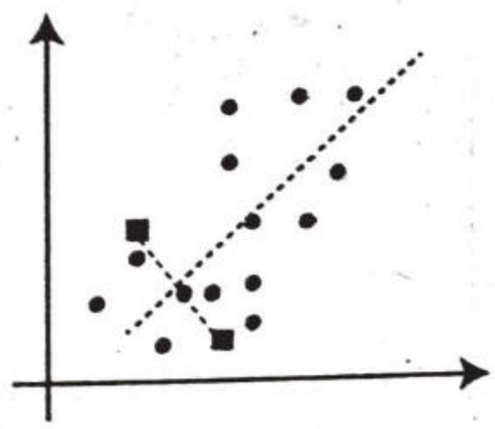
Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.

We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset.

Consider the below image:

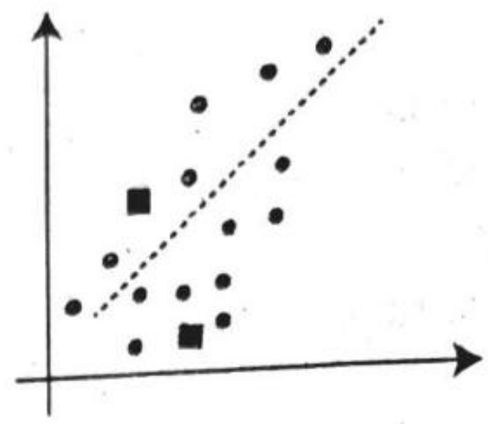


Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:

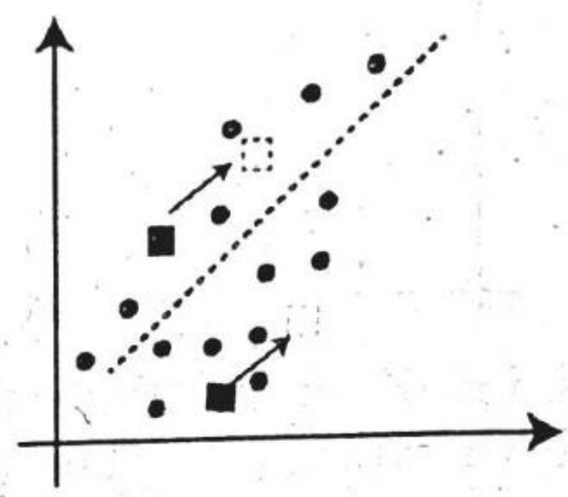


Scanned by TapScanner

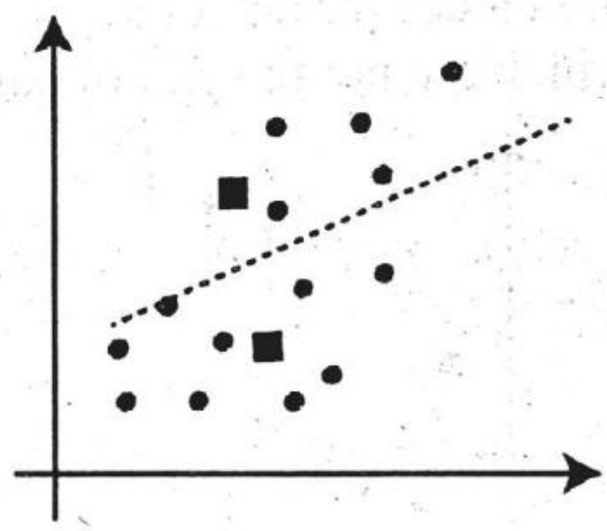
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



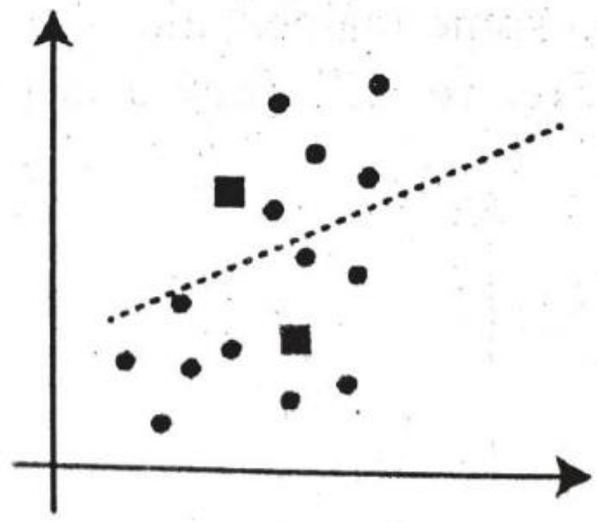
As we need to find the closest cluster, so we will repeat the process by choosing a new centroid. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:



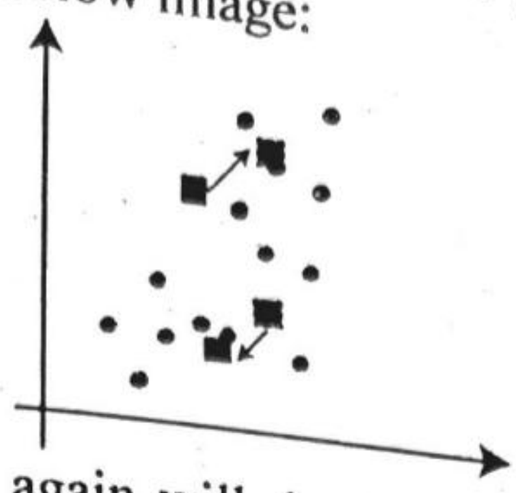
From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.



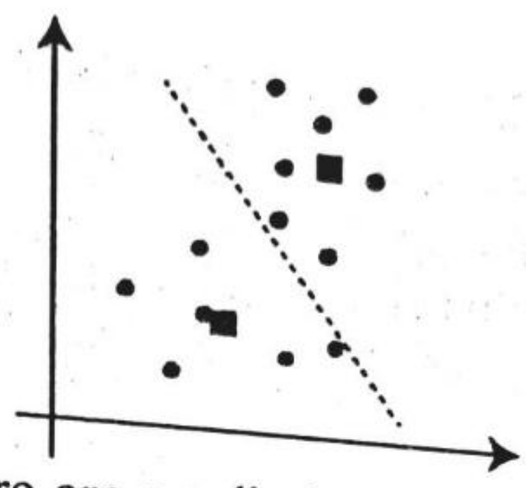
As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

Scanned by TapScanner

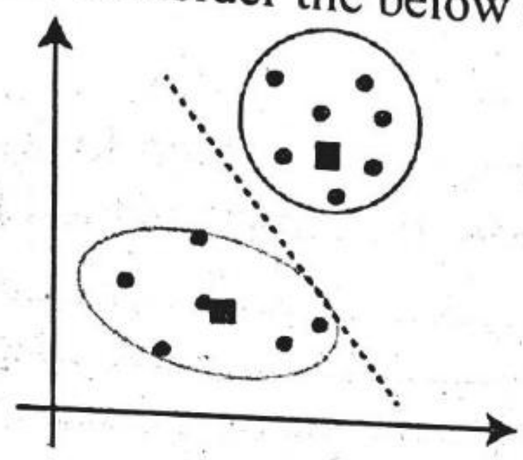
We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



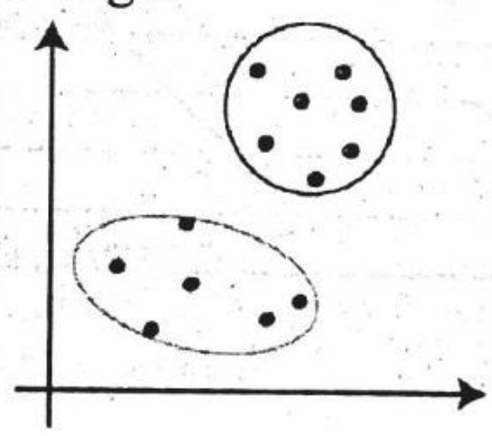
As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



3. How to choose the value of "K number of clusters" in K-means Clustering?

[MODEL QUESTION]

Answer:

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Scanned by TapScanner

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

WCSS = $\sum P_i$ in Cluster1 distance $(P_iC_1)^2 + \sum P_i$ in Cluster2 distance $(P_iC_2)^2 + \sum P_i$ in CLuster3 distance $(P_iC_3)^2$

In the above formula of WCSS,

$\sum P_i$ in Cluster1 distance $(P_iC_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

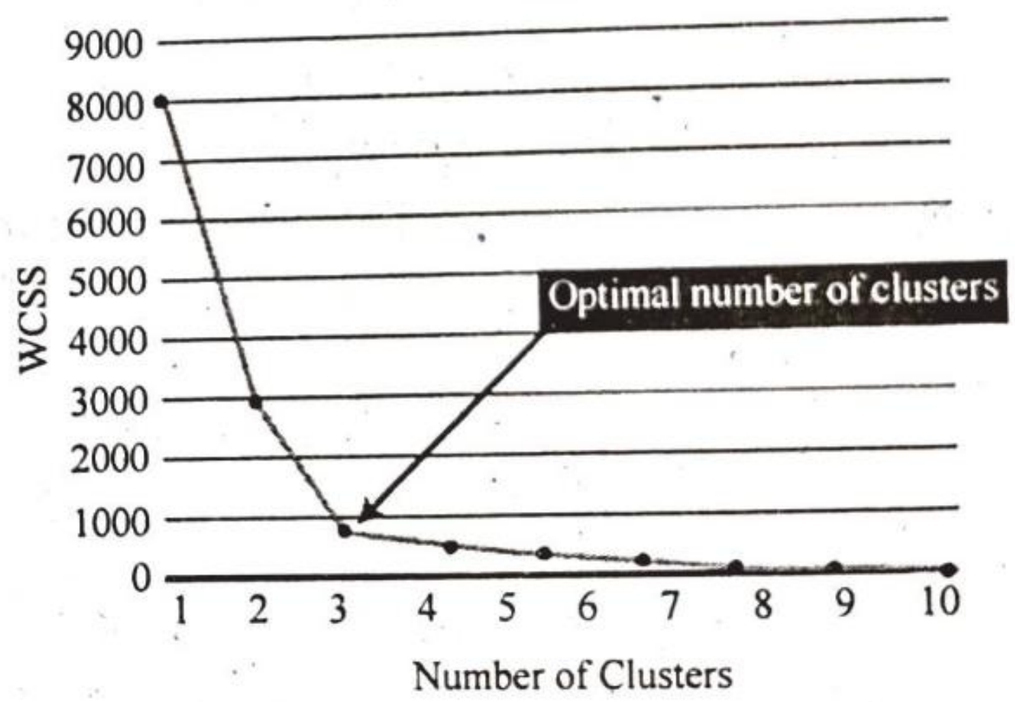
It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).

For each value of K, calculates the WCSS value.

Plots a curve between calculated WCSS values and the number of clusters K.

The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow; hence it is known as the elbow method. The graph for the elbow method looks like the below image:



4. How does Dimensionality Reduction works (using PCA)? **[MODEL QUESTION]**

Answer:

Dimensionality Reduction is simply reducing the number of features (columns) while retaining maximum information. Following are reasons for Dimensionality Reduction:

- Dimensionality Reduction helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.
- Removes Correlated Features.

Scanned by TapScanner

- Reducing the dimensions of data to 2D or 3D may allow us to plot and visualize it precisely. You can then observe patterns more clearly.
- It is helpful in noise removal also and as a result of that, we can improve the performance of models.

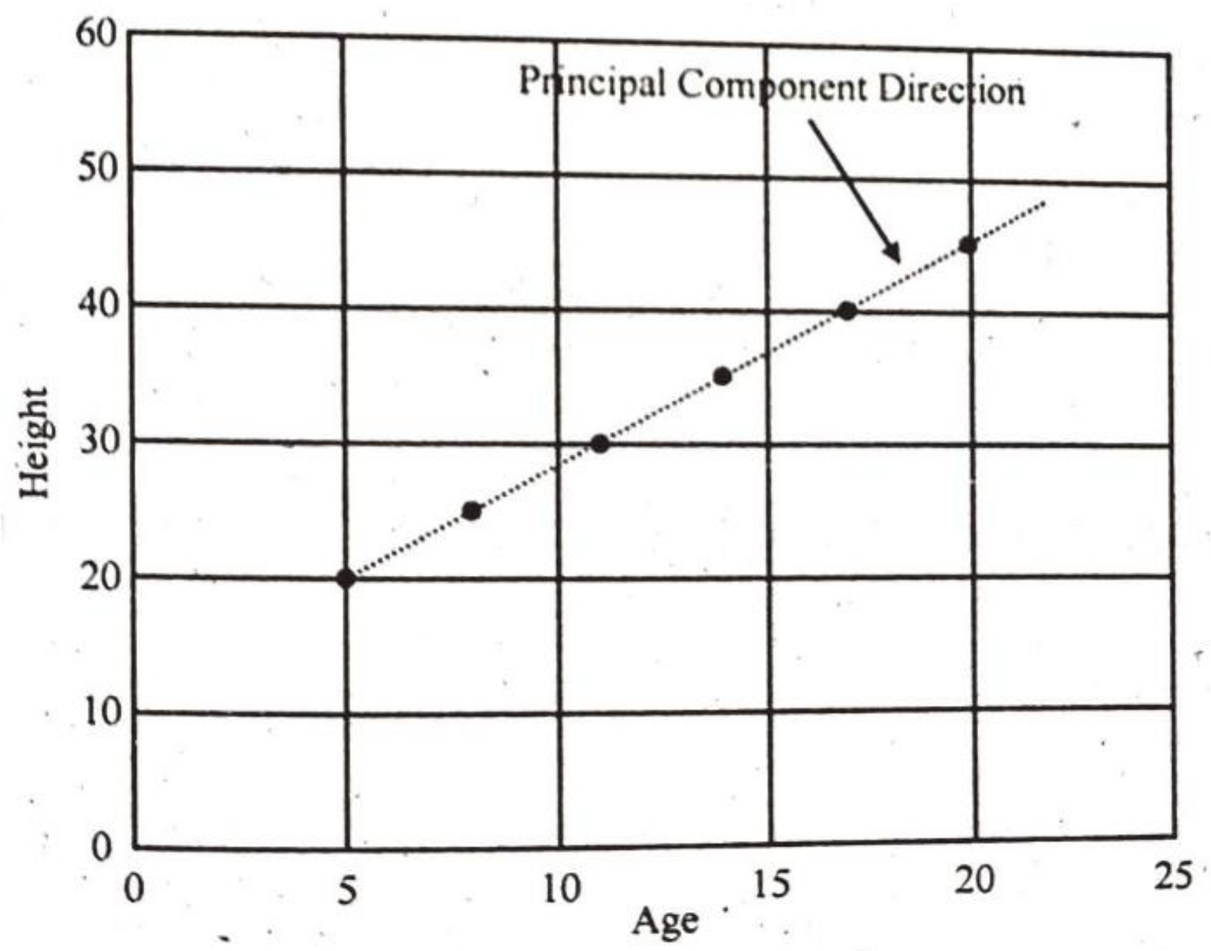
There are many methods for Dimensionality Reduction like PCA, ICA, t-SNE, etc., we shall see PCA (Principal Component Analysis).

Let's first understand what is **information** in data. Consider the following imaginary data, which has Age, Weight, and Height of people.

Age	Weight	Height
5	20	4
8	25	4
11	30	4
14	35	4
17	40	4
20	45	4

As the 'Height' of all the people is the same i.e., the variance is 0 thus it's not adding any information, so we can remove the 'Height' column without losing any information.

Now we know that information is variance, let's understand the working of PCA. In PCA, we find new dimensions (features) that capture maximum variance (i.e., information). To understand this we shall use the previous example. After removing 'Height', we are left with 'Age' and 'Weight'. These two features are caring all the information. In other words, we can say that we require 2 features (Age and Height) to hold the information, and if we can find a new feature that alone can hold all the information, we can replace origination 2 features with a new single feature, achieving dimensionality reduction!



Scanned by TapScanner

Now consider a line (blue dotted line) that is passing through all the points. The blue dotted line is capturing all the information so, we can replace 'Age' and 'Weight' (2 Dimensions) with the blue dotted line (1 Dimension) without losing any information, and in this way, we have done dimensionality reduction (2 dimensions to 1 dimension). The blue dotted line is called **Principal Component**.

5. What is the intuition behind Kernel Principal Component Analysis (KPCA)? What are the steps of KPCA? **[MODEL QUESTION]**

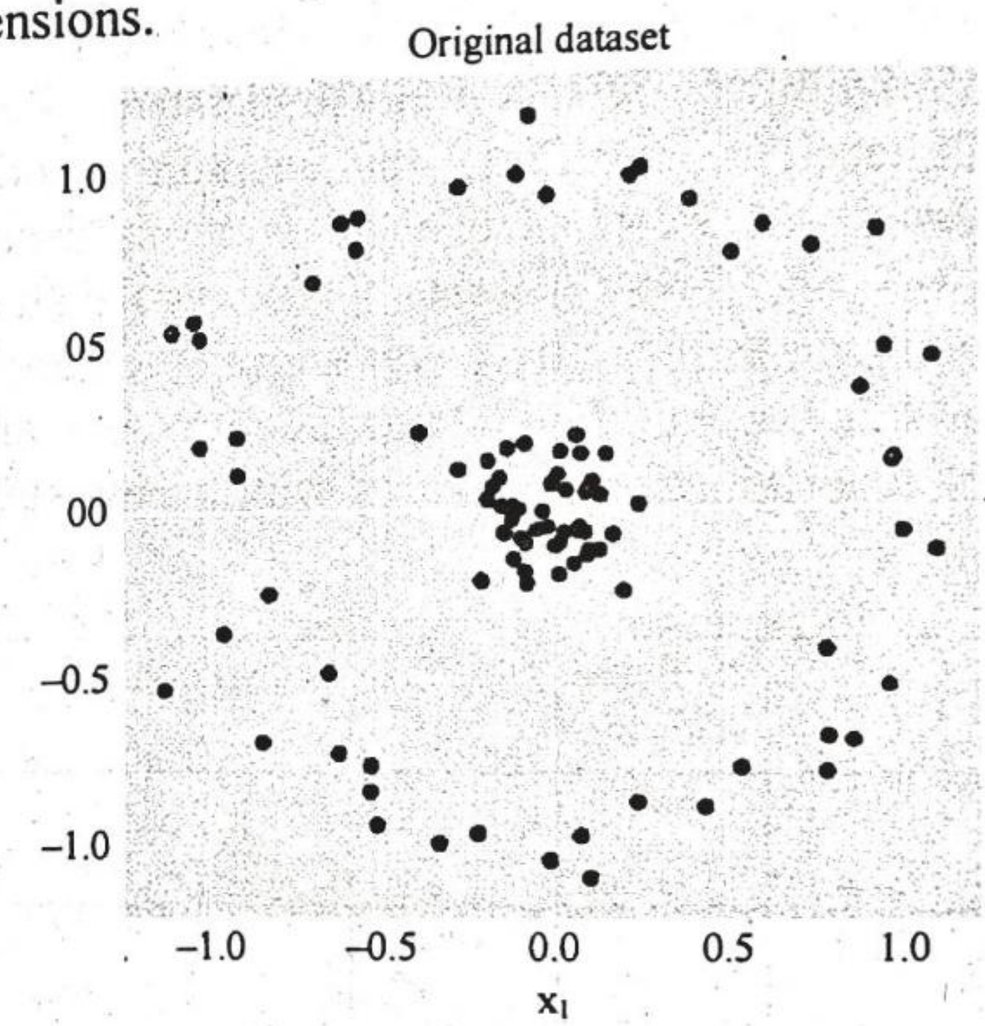
Answer:

The idea of KPCA relies on the intuition that many datasets, which are not linearly separable in their space, can be made linearly separable by projecting them into a higher dimensional space. The added dimensions are just simple arithmetic operations performed on the original data dimensions.

So we project our dataset into a higher dimensional feature space, and because they become linearly separable, then we can apply PCA on this new dataset.

Performing this linear dimensionality reduction in that space will be equivalent to a non-linear dimensionality reduction in the original space.

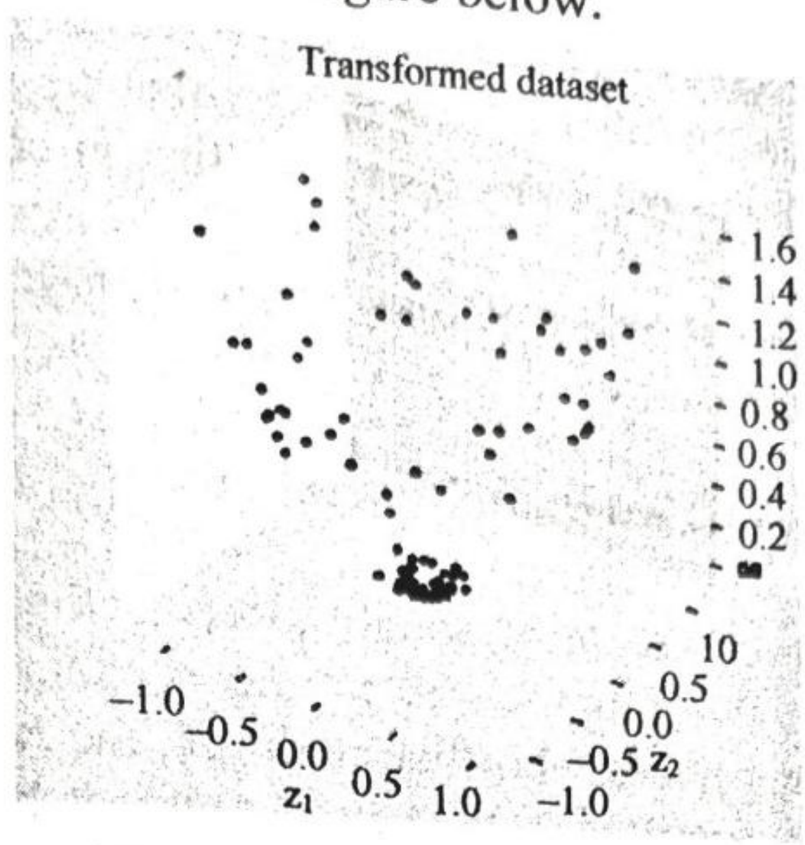
For example consider this dataset where the brown and blue points are not linearly separable in two dimensions.



We define the linear mapping $T(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$

Scanned by TapScanner

This mapping function will project the data from a lower dimensional (2D) to a higher dimensional (3D) space as shown in the figure below.



We can see from the above figure that the dataset becomes linearly separable. The optimal surface that PCA will find will be in the middle between blue and brown points. And if we inverse this transform then this surface will correspond to a non-linear curve in the original 2D space. And by projecting on that curve, we will find the optimal dimensionality reduction.

But doing PCA in high dimensional space will need a lot of computations, so in order to solve this problem we use kernel methods.

Steps of KPCA:

1. First we will choose a kernel functions $k(x_i, x_j)$ and let T be any transformation to a higher dimension.
2. And like PCA, we will find the covariance matrix of our data. But here, we will use kernel function to calculate this matrix. So will compute kernel matrix, which is the matrix that results from applying kernel function to all pairs of data.

$$K = T(X)T(X)^T$$

$$K = \begin{matrix} k(x_1, x_1) & k(x_1, x_2) & \dots\dots\dots & k(x_1, x_d) \\ k(x_2, x_1) & k(x_2, x_2) & \dots\dots\dots & k(x_2, x_d) \\ \bullet & \bullet & & \bullet \\ \bullet & \bullet & & \bullet \\ k(x_d, x_1) & k(x_d, x_2) & \dots\dots\dots & k(x_d, x_d) \end{matrix}$$

Note: KPCA scales with the size of our data.

1. Center our kerenl matrix (this equivalent to substract the mean of the transformed data and dividing by standard deviations) :

$$K_{new} = K - 2(I)K + (I)K(I)$$

where I is a matrix that its all elements are equal to i/d.

Scanned by TapScanner

2. Then, we will find eigenvectors and eigenvalues of this matrix.
3. Sort our eigenvectors based on their corresponding eigen values in a decreasing order.
4. We will choose what number of dimensions that we want our reduced dataset to be, let's call it m. Then we will choose our first m eigenvectors and concatenate them in one matrix.
5. Finally, calculate the product of that matrix with your data. The result will be your new reduced dataset.

6. What do you understand by matrix factorization? **[MODEL QUESTION]**

Answer:

Matrix factorization is one of the most sought-after machine learning recommendation models. It acts as a catalyst, enabling the system to gauge the customer's exact purpose of the purchase, scan numerous pages, shortlist, and rank the right product or service, and recommend multiple options available. Once the output matches the requirement, the lead translates into a transaction and the deal clicks.

This mathematical model helps the system split an entity into multiple smaller entries, through an ordered rectangular array of numbers or functions, to discover the features or information underlying the interactions between users and items.

Once an individual raises a query on a search engine, the machine deploys uses matrix factorization to generate an output in the form of recommendations. The system uses two approaches – content-based filtering and collaborative filtering – to make recommendations.

Collaborative Filtering: Collaborative filtering is to discover the similarities on the user's past behaviour and make predictions to the user based on a similar preferecne with other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.

Content-Based Filtering: Content-Based Filtering is used to produce items recommendation based on items' characteristics.

Matrix factorization is a way to generate latent features when multiplying two different kinds of entities. Collaborative filtering is the application of matrix factorisation to identify the relationship between items' and users' entities. With the input of users' ratings on the shop items, we would like to predict how the users would rate the items so the users can get the recommendation based on the prediction.

Assume we have the customers' ranking table of 5 users and 5 movies, and the ratings are integers ranging from 1 to 5, the matrix is provided by the table below.

	Movie1	Movie2	Movie3	Movie4	Movie5
U1		5	4	2	1
U2	1			5	3
U3	1	4	4	1	
U4			2		2
U5	3	1	1		

Scanned by TapScanner

Since not every user gives ratings to all the movies, there are many missing values in the matrix and it results in a sparse matrix. Hence, the null values not given by the users would be filled with 0 such that the filled values are provided for the multiplication. For example, two users give high ratings to a certain move when the movie is acted by their favourite actor and actress or the movie genre is an action one, etc. From the table above, we can find that the user1 and user3 both give high ratings to move2 and movie3. Hence, from the matrix factorization, we are able to discover these latent features to give a prediction on a rating with respect to the similarity in user's preferences and interactions.

Given a scenario, user 4 didn't give a rating to the movie 4. We'd like to know if user 4 would like movie 4. The method is to discover other users with similar preferences of user 4 by taking the ratings given by users of similar preferences to the movie 4 and predict whether the user 4 would like the movie 4 or not.

7. What is mathematic concept of matrix factorization? **[MODEL QUESTION]**

Answer:

Define a set of Users (U), items (D), R size of $|U|$, and $|D|$. The matrix $|U|*|D|$ includes all the ratings given by users. The goal is to discover K latent features. Given with the input of two matrices $P(|U|*k)$ and $Q(|D|*k)$, it would generate the product result R.

$$R \approx P \times Q^T = \hat{R}$$

Matrix P represents the association between a user and the features while matrix Q represents the association between an item and the features. We can get the prediction of a rating of an item by the calculation of the dot product of the two vectors corresponding to u_i and d_j.

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^{k} p_{ik} q_{kj}$$

To get two entities of both P and Q, we need to initialize the two matrices and calculate the difference of the product named as matrix M. Next, we minimize the difference through the iterations. The method is called **gradient descent**, aiming at finding a local minimum of the difference.

$$e_{ij}^2 = \left(r_{ij} - \hat{r}_{ij}\right)^2 = \left(r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj}\right)^2$$

To minimize the error, the gradient is able to minimize the error, and therefore we differentiate the above equation with respect to these two variables separately.

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2\left(r_{ij} - \hat{r}_{ij}\right)\left(q_{kj}\right) = -2e_{ij} q_{kj}$$

$$\frac{\partial}{\partial q_{ik}} e_{ij}^2 = -2\left(r_{ij} - \hat{r}_{ij}\right)\left(p_{ik}\right) = -2e_{ij} p_{ik}$$

Scanned by TapScanner

From the gradient, the mathematic formula can be updated for both p_{ik} and q_{kj}. α is the step to reach the minimum while the gradient is calculated, and α is usually set with a small value.

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + 2\alpha e_{ij} q_{kj}$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + 2\alpha e_{ij} p_{ik}$$

From the above equation, p'_{ik} and q'_{kj} can both be updated through iterations until the error converges to its minimum.

$$E = \sum_{(u_i, d_j, r_{ij}) \in T} e_{ij} = \sum_{(u_i, d_j, r_{ij}) \in T} \left(r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj} \right)^2$$

8. Explain generative model. **[MODEL QUESTION]**

Answer:

A **Generative Model** is a way of learning any kind of data distribution. It is used in **unsupervised machine learning** as a means to describe phenomena in data, enabling the computers to understand the real world.

In unsupervised machine learning, generative modeling algorithms process the training data and make reductions in the data. These models generally are run on neural networks and can come to naturally recognize the distinctive features of the data. The neural networks take these reduced fundamental understandings of real world data and then use them to model data that is similar or indistinguishable from real world data.

The **main aim** of all types of generative models is to **learn the true data distribution of the training set so that the new data points are generated with some variations.** But it is not possible for the model to learn the exact distribution of our data and so we model a distribution which is as similar to the true data distribution. For this, we use the knowledge of neural networks to learn a function which can approximate the model distribution to the true distribution.

Different types of generative modeling:

The two types of generative models are:

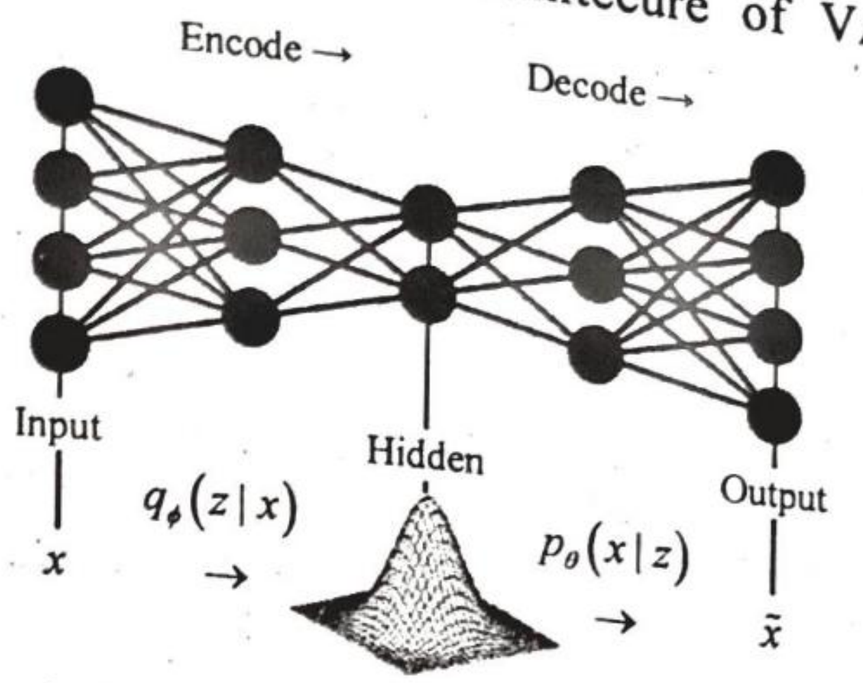
1. **Variational Autoencoder** (VAE)
2. **Generative Adversarial Networks** (GAN)

Variational Autoencoder (VAE):

The definition of a VAE is that it "**provides probabilistic descriptions of observations in latent spaces**". This means VAEs store latent attributes as probability distributions.

Scanned by TapScanner

Let's first understand about the general architecure of VAE through the below diagram:

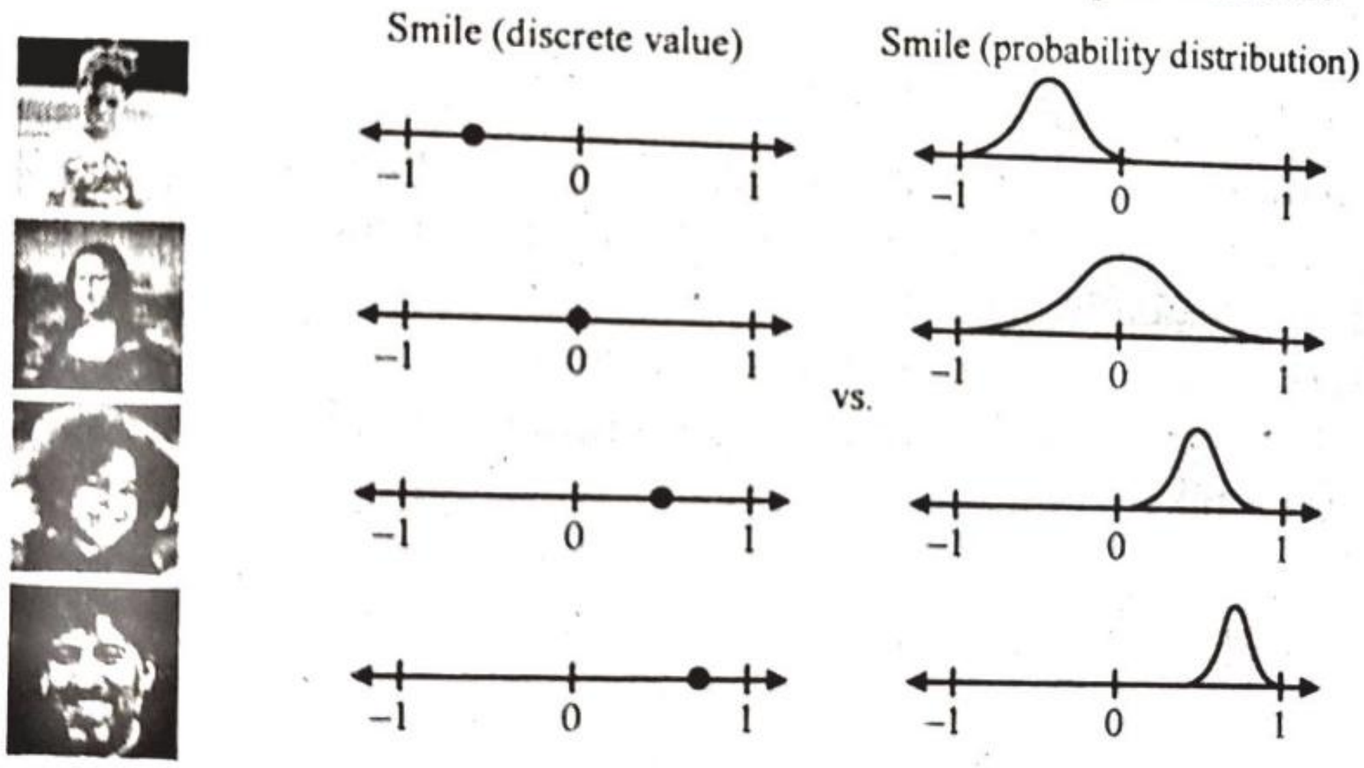


The setup looks like the design of deep neural network and it consists of a pair of networks:

- the encoder
- the decoder

The **encoder** is responsible for the **mapping of input x to posteriors distributions** $q_\theta(z|x)$. Similarly $p(x|z)$ **is then parametrized by the decoder,** Combinely they form a generative network which takes latent variables z and parameters as inputs and projects them to data distributions $p_\phi(x|z)$.

Each input image has features that can normally be described as single, discrete values. Variational autoencoders describe these values as probability distributions. Decoders can then sample randomly from the probability distributions for input vectors.



A major drawback of VAEs is **the blurry outputs that they generate.** This is because data distributions are not recovered completely.

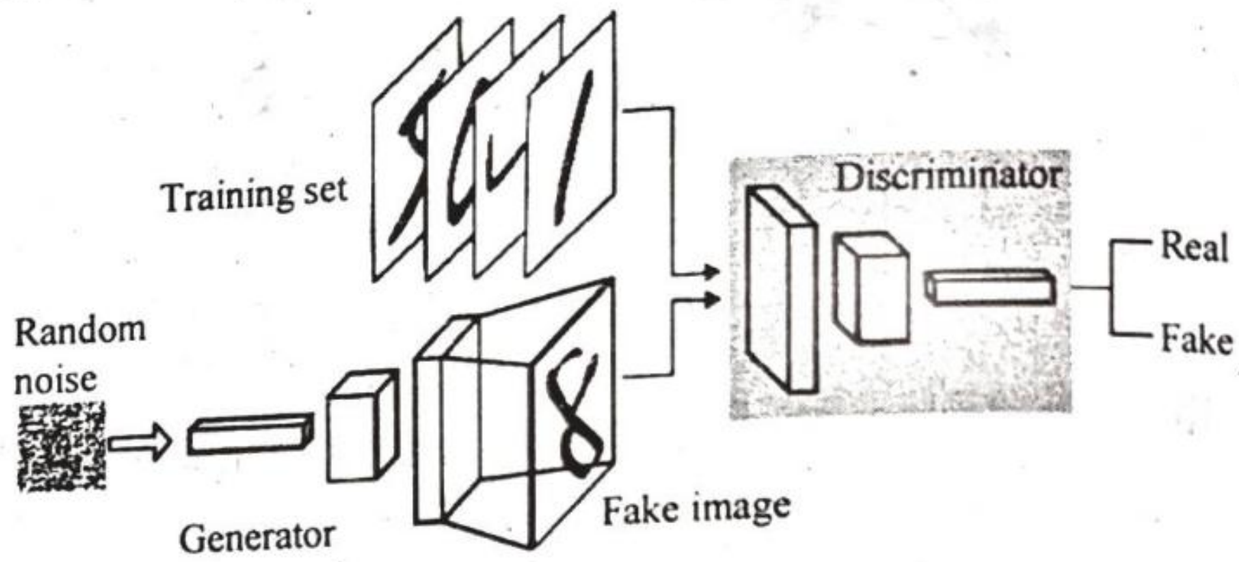
Generative Adversarial Networks (GAN)

Generative adversarial networks (GANs) are **deep neural net architectures comprised of two nets, pitting one against the other** (thus the "adversarial").

Scanned by TapScanner

GAN is having a huge scope or potential because they can **learn to mimic any distribution of data**. That is, GANs can be taught to learn anything in any domain: images, music, speech, prose. They are **robot artists** in a sense, and their output is impressive.

The general architecture of GAN is described below:



One neural network, called the generator, generates new data instances, while the other, the discriminator, evaluates them for authenticity; i.e., the discriminator decides whether each instance of data it reviews belongs to the actual training dataset or not.

Let us assume we are going to generate hand-written numerals like those found in the MNIST dataset, which is taken from the real world. The goal of the discriminator, when shown an instance from the true MNIST dataset, is to recognize them as authentic. Meanwhile, the generator creates images and passes it to the discriminator. The goal of the generator is to generate passable hand-written digits, to lie without being caught. The goal of the discriminator is to identify images coming from the generator as fake.

Here are the steps a GAN takes:

- The generator takes random numbers and returns an image.
- This generated image is fed into the discriminator alongside a stream of images taken from the actual dataset.
- The discriminator takes in both real and fake images and returns probabilities, a number between 0 and 1, with 1 representing a prediction of authenticity and 0 representing fake.

Scanned by TapScanner

EVALUATING MACHINE LEARNING ALGORITHM AND MODEL SELECTION

Multiple Choice Type Questions

1. Which of the following is a widely used and effective machine learning algorithm based on the idea of bagging? [MODEL QUESTION]
   a) Decision Tree
   b) Regression
   c) Classification
   d) Random Forest

Answer: (d)

2. To find the minimum or the maximum of a function, we set the gradient to zero because: [MODEL QUESTION]
   a) The value of the gradient at extrema of a function is always zero
   b) Depends on the type of problem
   c) Both (a) & (b)
   d) None of these

Answer: (a)

3. The most widely used metrics and tools to assess a classification model are: [MODEL QUESTION]
   a) Confusion matrix
   b) Cost-sensitive accuracy
   c) Area under the ROC curve
   d) All of these

Answer: (d)

4. Which of the following is a good test dataset characteristic? [MODEL QUESTION]
   a) Large enough to yield meaningful results
   b) Is representative of the dataset as a whole
   c) Both (a) and (b)
   d) None of the above

Answer: (c)

5. What is the purpose of performing cross-validation? [MODEL QUESTION]
   a) To assess the predictive performance of the models
   b) To judge how the trained model performs outside the sample on test data
   c) Both (a) and (b)
   d) none of the above

Answer: (c)

6. Which of the following is correct use of cross validation? [MODEL QUESTION]
   a) Selecting variables to include in a model
   b) Comparing predictors
   c) Selecting parameters in prediction function
   d) All of the mentioned

Answer: (d)

Scanned by TapScanner

7. Point out the wrong combination. [MODEL QUESTION]
   a) True negative = correctly rejected
   b) False negative = correctly rejected
   c) False positive = correctly identified
   d) All of the mentioned

Answer: (c)

8. Which of the following is an example of feature extraction? [MODEL QUESTION]
   a) Constructing bag of words vector from an email
   b) Applying PCA projects to a large high-dimensional data
   c) Removing stopwords in a sentence
   d) All of the above

Answer: (d)

9. Random Forest has _____ as base learning models [MODEL QUESTION]
   a) multiple decision trees
   b) bagging
   c) Entropy
   d) none of these

Answer: (a)

10. _____helps improve machine learning results by combining several models. [MODEL QUESTION]
   a) Machine Learning
   b) bagging
   c) Entropy
   d) Ensemble learning

Answer: (d)

11. What is overfit model? [MODEL QUESTION]
   a) A model that learns the training dataset too well, performing well on the training dataset but does not perform well on a testing dataset
   b) A model that fails to sufficiently learn the problem and performs poorly on a training dataset and does not perform well on a testing dataset
   c) All of above
   d) None of above

Answer: (a)

Short Answer Type Questions

1. What is Statistical Learning Theory? [MODEL QUESTION]

Answer:

Statistical learning theory is a framework for machine learning that draws from statistics and functional analysis. It deals with finding a predictive function based on the data presented. The main idea in statistical learning theory is to build a model that can draw conclusions from data and make predictions.

2. What are types of Statistical Learning? [MODEL QUESTION]

Answer:

With statistical learning theory, there are two main types of data:

Dependent Variable — a variable (y) whose values depend on the values of other variables (a dependent variable is sometimes also referred to as a target variable)

Scanned by TapScanner

Independent Variables — a variable (x) whose value does not depend on the values of other variables (independent variables are sometimes also referred to as predictor variables, input variables, explanatory variables, or features)

In statistical learning, the independent variable(s) are the variable that will affect the dependent variable.

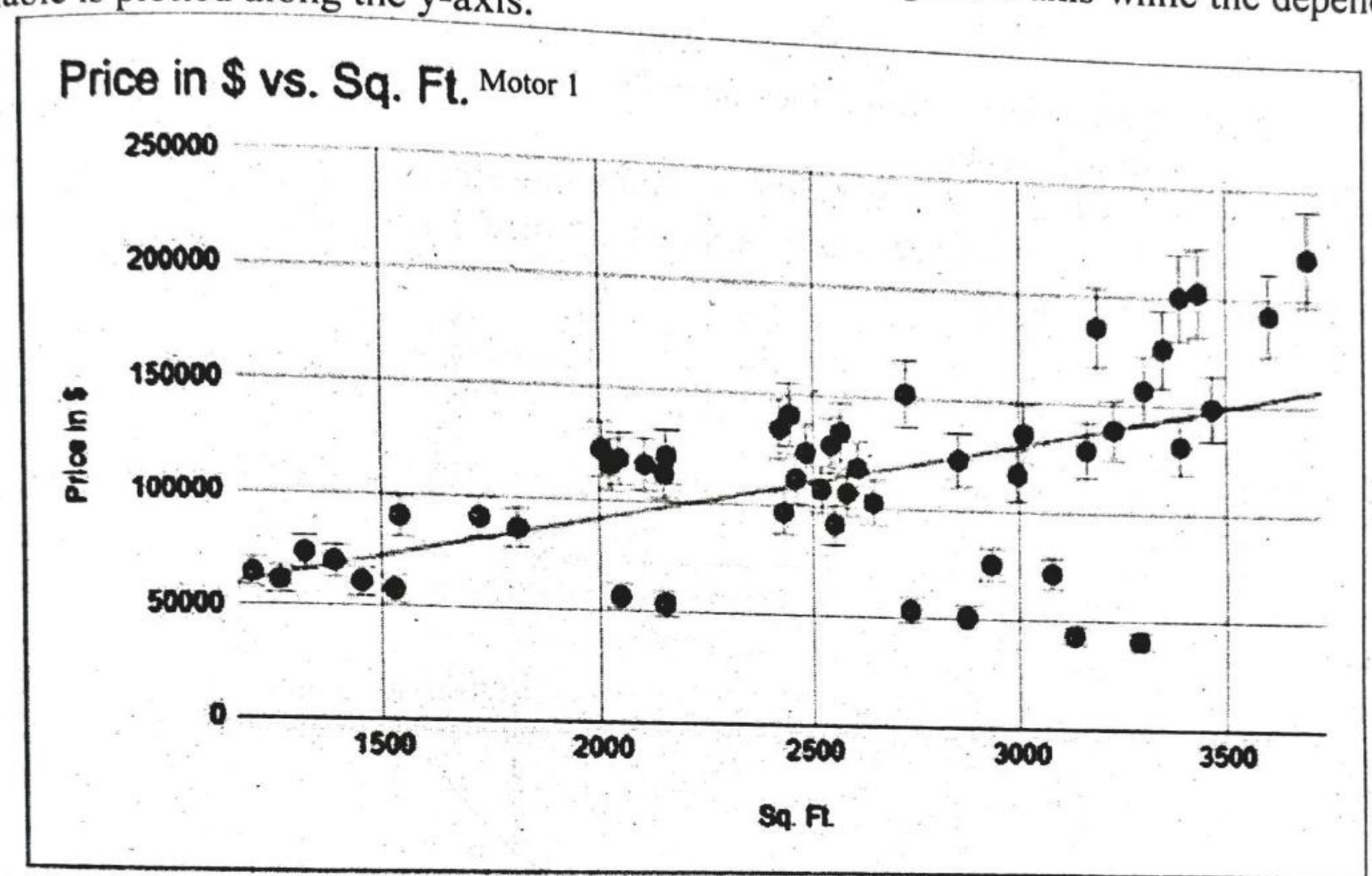
A common examples of an Independent Variable is Age. There is nothing that one can do to increase or decrease age. This variable is independent.

Some common examples of Dependent Variables are:

Weight — a person's weight is dependent on his or her age, diet, and activity levels (as well as other factors)

Temperature — temperature is impacted by altitude, distance from equator (latitude) and distance from the sea

In graphs, the independent variable is often plotted along the x-axis while the dependent variable is plotted along the y-axis.



In this example, which shows how the price of a home is affected by the size of the home, sq. ft is the independent variable while price of the home is the dependent variable.

3. What is Statistical Model? **[MODEL QUESTION]**

Answer:

A statistical model defines the relationships between a dependent and independent variable. In the above graph, the relationships between the size of the home and the price of the home is illustrated by the straight line.

We can define this relationship by using $y = mx + c$ where m represents the gradient and c is the intercept. Another way that this equation can be expressed is with roman numerals which would look something like:

$$\text{Price of Home} = \beta_0 + \beta_1 * \text{sq.ft.}$$

Scanned by TapScanner

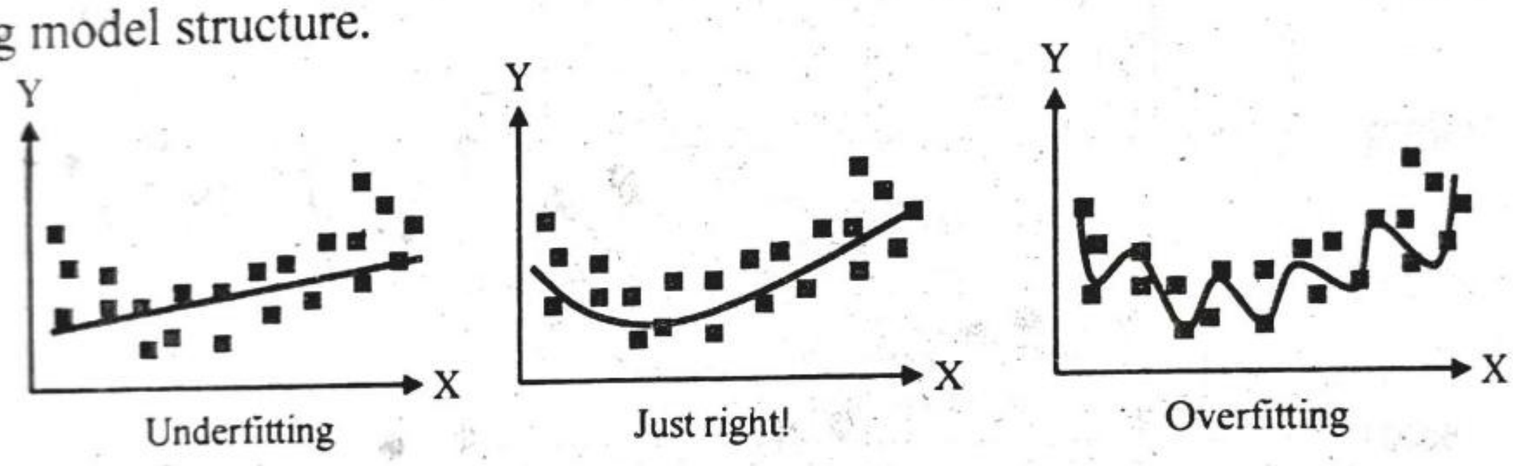
This model would describe the price of a home as having a linear relationship with the size of a home. This would represent a simple model for the relationship.
If we suppose that the size of the home is not the only independent variable when determining the price and that the number of bathrooms is also an independent variable, the equation would look like:

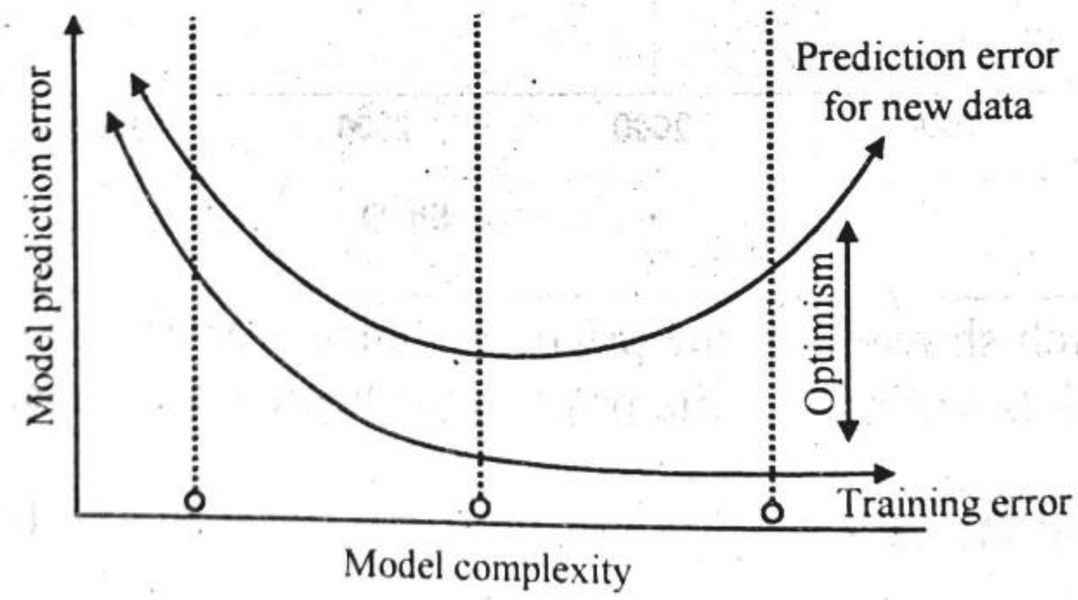
$$\text{Price of Home} = \beta_0 + \beta_1 * \text{sq.ft.} + \beta_2 * \#\text{ of bathrooms}$$

4. How we can generalize and validate statistical model? **[MODEL QUESTION]**

Answer:
In order to build an effective model, the available data needs to be used in a way that would make the model generalizable for unseen situations. Some problems that occur when building models is that the model under-fits or over-fits to the data.

Under-fitting — when a statistical model does not adequately capture the underlying structure of the data and, therefore, does not include some parameters that would appear in a correctly specified model.
Over-fitting — when a statistical model contains more parameters that can be justified by the data and includes the residual variation ("noise") as if the variation represents underlying model structure.





As you can see, it is important to create a model that can generalize to the data that it is given so that it can make the most accurate predictions when given new data.
Model Validation is used to assess over-fitting and under-fitting of the data. The steps to perform model validation are:

1. Split the data into two parts, training data and testing data (anywhere between 80/20 and 70/30 is ideal)
2. Use the larger portion (training data) to train the model

Scanned by TapScanner

3. Use the smaller portion (testing data) to test the model. This data is not being used to train the model, so it will be new data for the model to build predictions from.
4. If the model has learned well from the training data, it will perform well with both the training data and testing data. To determine how well the model is performing on both sets of data, you can calculate the accuracy score for each. The model is over-fitting if the training data has a significantly higher accuracy score than the testing data.

5. How to select a model in machine learning? **[MODEL QUESTION]**

Answer:

In machine learning paradigm, the potential causes of disturbance, e.g. average income of the local population, weapon sales, the inflow of immigrants, etc. are input variables. They are also called predictors, attributes, features, independent variables. The number of criminal incidents is an output variable (also called response or dependent variable). Input variables can be denoted by X, while individual input variables are represented as $X_1, X_2, \ldots\ldots, X_n$ and output variable by symbol Y. The relationship between X and Y is represented in the general form: $Y = f(X) + e$, where 'f' is the **target function** and 'e' is a random term.

A **cost function** (also called error function) helps to measure the extent to which the model is going wrong in estimating the relationship between X and Y. In that sense, cost function can tell how bad the model is performing. For example, R-squared is a cost function of regression model.

Loss function is almost synonymous to cost function – only difference being loss function is usually a function defined on a data point, while cost function is for the entire training data set.

Machine learning is an optimization problem, we try to define a model and tune the parameters to find the most suitable solution to a problem. However, we need to have a way to evaluate the quality or optimality of a solution. This is done using **objective function.**

Objective function takes in data and model (along with parameters) as input and returns a value. Target is to find values of model parameter to maximize or minimize the return value. When the objective is to minimize the value, it becomes synonymous to cost function.

6. What is a confusion matrix? **[MODEL QUESTION]**

Answer:

A Confusion matrix is an $N \times N$ matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

Scanned by TapScanner

For a binary classification problem, we would have a 2×2 matrix as shown below with 4 values:

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Let's decipher the matrix:

- The target variable has two values: Positive or Negative
- The columns represent the actual values of the target variable
- The rows represent the predicted values of the target variable

7. Explain True Positive, True Negative, False Positive and False Negative in a Confusion Matrix. **[MODEL QUESTION]**

Answer:

True Positive (TP)

- The predicted value matches the actual value
- The actual value was positive and the model predicted a positive value

True Negative (TN)

- The predicted value matches the actual value
- The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error

- The predicted value was falsely predicted
- The actual value was negative but the model predicted a positive value
- Also known as the **Type 1 error**

False Negative (FN) – Type 2 error

- The predicted value was falsely predicted
- The actual value was positive but the model predicted a negative value
- Also known as the **Type 2 error**

Suppose we had a classification dataset with 1000 data points. We fit a classifier on it and get the below confusion matrix:

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	560	
	NEGATIVE		330

Scanned by TapScanner

The different values of the Confusion matrix would be as follows:

- True Positive (TP) = 560; meaning 560 positive class data points were correctly classified by the model
- True Negative (TN) = 330; meaning 330 negative class data points were correctly classified by the model
- False Positive (FP) = 60; meaning 60 negative class data points were incorrectly classified as belonging to the positive class by the model
- False Negative (FN) = 50; meaning 50 positive class data points were incorrectly classified as belonging to the negative class by the model.

8. Why Do We Need a Confusion Matrix?

[MODEL QUESTION]

Answer:

Let's say you want to predict how many people are infected with a contagious virus in times before they show the symptoms, and isolate them from the healthy population. The two values for our target variable would be: Sick and Not Sick.

Now, you must be wondering – why do we need a confusion matrix when we have our all-weather friend – Accuracy? Well, let's see where accuracy falters.

Our dataset is an example of an imbalanced dataset. There are 947 data points for the negative class and 3 data points for the positive class. This is how we'll calculate the accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Let's see how our model performed:

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

The total outcome values are:

$$TP = 30, TN = 930, FP = 30, FN = 10$$

So, the accuracy for our model turns out to be:

$$\text{Accuracy} = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

Scanned by TapScanner

Our model is saying "I can predict sick people 96% of the time". However, it is doing the opposite. It is predicting the people who will not get sick with 96% accuracy while the sick are spreading the virus!

9. Explain precision and recall. **[MODEL QUESTION]**

Answer:

Precision tells us how many of the correctly predicted cases actually turned out to be positive.

Here's how to calculate Precision:

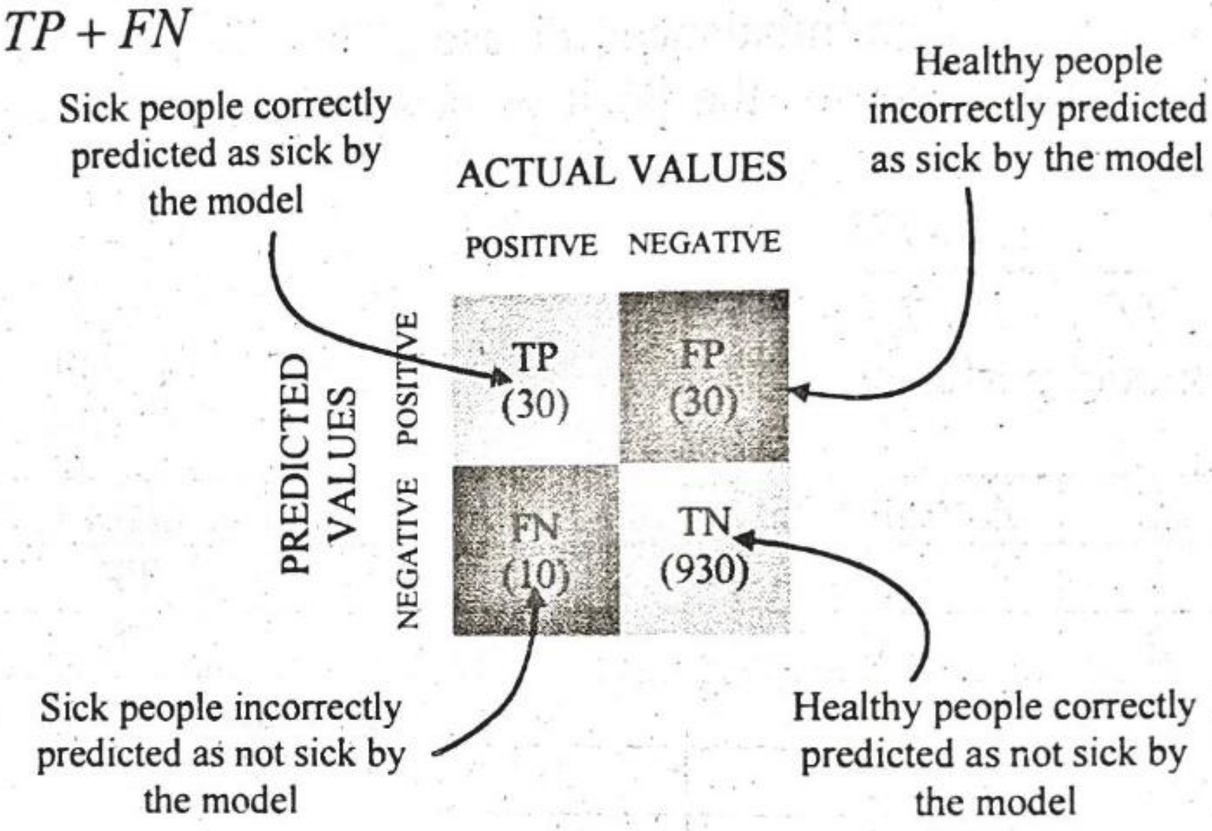
$$\text{Precision} = \frac{TP}{TP + FP}$$

This would determine whether our model is reliable or not.

Recall tells us how many of the actual positive cases we were able to predict correctly with our model.

And here's how we can calculate Recall:

$$\text{Recall} = \frac{TP}{TP + FN}$$



We can easily calculate Precision and Recall for our model by plugging in the values into the above questions:

$$\text{Precision} = \frac{30}{30 + 30} = 0.5$$

$$\text{Recall} = \frac{30}{30 + 10} = 0.75$$

50% percent of the correctly predicted cases turned out to be positive cases. Whereas 75% of the positives were successfully predicted by our model.

Precision is a useful metric in cases where False Positive is a higher concern than False Negatives.

Precision is important in music or video recommendation systems, e-commerce websites, etc. Wrong results could lead to customer churn and be harmful to the business.

Recall is a useful metric in cases where False Negative trumps False Positive.

Scanned by TapScanner

Recall is important in medical cases where it doesn't matter whether we raise a false alarm but the actual positive cases should not go undetected!
In our example, Recall would be a better metric because we don't want to accidentally discharge an infected person and let them mix with the healthy population thereby spreading the contagious virus. Now you can understand why accuracy was a bad metric for our model.

10. What is F1-Score? **[MODEL QUESTION]**

Answer:

In practice, when we try to increase the precision of our model, the recall goes down, and vice-versa. The F1-score captures both the trends in a single value:

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

F1-score is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall.
But there is a catch here. The interpretability of the F1-score is poor. This means that we don't know what our classifier is maximizing – precision or recall? So, we use it in combination with other evaluation metrics which gives us a complete picture of the result.

11. How we can improve performance of a learning model? **[MODEL QUESTION]**

Answer:

One effective way to improve model performance is by tuning model parameter. Model parameter tuning is the process of adjusting the model fitting options. For example, in the popular classification model k-Nearest Neighbour (kNN), using different values of 'k' or the number of nearest neighbours to be considered, the model can be tuned. In the same way, a number of hidden layers can be adjusted to tune the performance in neutral networks model. Most machine learning models have at least one parameter which can be tuned.
As an alternate approach of increasing the performance of one model, several models may be combined together. The models in such combination are complimentary to each other, i.e., one model may learn one type data sets well while struggle with another type of data set. Another model may perform well with the data set which the first one struggle with. This approach of combining different models with diverse strengths is known as ensemble. Ensemble helps in averaging out biases of the different underlying models and also reducing the variance. Ensemble methods combines weaker learners to create stronger ones. A performance boost can be expected even if models are built as usual and then ensemble.

Scanned by TapScanner

Following are the typical steps in ensemble process:

- Build a number of models based on the training data.
- For diversifying the models generated, the training data subset can be varied using the allocation function. Sampling techniques like bootstrapping may be used to generate unique training data sets.
- Alternatively, the same training data may be used but the models combined are quite varying, e.g. SVM, neural network, kNN, etc.
- The outputs from the different models are combined using a combination function. A very simple strategy of combining, say in case of a prediction task using ensemble, can be majority voting of the different models combined. For example, 3 out of 5 classes predict 'win' and 2 predict 'loss'- then the final outcome of the ensemble using majority vote would be 'win'.

Long Answer Type Questions

1. How to find the best solution? **[MODEL QUESTION]**

Answer:

Different machine learning algorithms search for different trends and patterns. One algorithm isn't the best across all data sets or for all use cases. To find the best solution, you need to conduct many experiments, evaluate machine learning algorithms, and tune their hyperparameters.

First, you choose, justify, and apply a model performance indicator to assess your model and justify the choice of an algorithm. Examples of model performance indicators include the F1 score, true positive rate, and within cluster sum of squared error.

Implement your algorithm in at least one deep-learning and at least one non-deep learning algorithm. Then, compare and document model performance.

Apply at least one more iteration in the process model that involves at least the feature creation task. Record the impact on model performance, such as data normalizing and principal component analysis (PCA). Depending on the algorithm class and data set size, you might choose specific technologies or frameworks to solve your problem.

As you evaluate algorithms, it's useful to know related terminology:

Term	Description
Descriptive analytics	Descriptive analytics is likely the most common type of analytics that is used to create dashboards and reports. It describes and summarizes events that already occurred. One example is a grocery store owner who wants to know how items of each product were sold in all stores within a region in the last five years.
Predictive analytics	Predictive analytics refers to using mathematical and statistical methods to forecast future outcomes. The grocery store owner wants to understand how many products can be sold in the next couple of months so that he can decide on inventory levels.
Prescriptive analytics	Prescriptive analytics is used to optimize business decisions by simulating scenarios that are based on a set of constraints. The grocery

Scanned by TapScanner

Term	Description
	store owner wants to create a staffing schedule for employees, but to do so, he must account for factors such as availability, vacation time, number of hours of work, and potential emergencies (constraints). He also must create a schedule that works for everyone and ensure that his business is able to function on a daily basis.
Supervised learning	Supervised learning teaches a model from labelled training data and helps you to make predictions about unseen or future data. During the training, you give the algorithm a dataset that contains correct answers (label y). Then, you validate the model accuracy with a test data set with correct answers. A data set must be split into training and test sets.
Classification	With classification, you're trying to predict one of a small number of discrete-valued outputs. For example, you might try to predict whether your label is binary (binary classification) or categorical (multiclass classification).
Regression	In regression, the goal of the learning problem is to predict continuous value output.
Unsupervised learning	Given a data set, try to find tendencies in the data by using techniques like clustering.
Feature	Feature is an attribute that is used as input for the model to train. Other names include dimension or column.
Bias	Bias is the expected difference between the parameters of a model that perfectly fits your data and the parameters that your algorithm learned. Low-bias algorithms, such as Decision Trees, K-nearest Neighbours, and Support Vector Machines, tend to find more complex patterns than high-bias algorithms.
Variance	Variance is how much the algorithm is impacted by the training data and how much the parameters change with new training data. Low-variance algorithm, such as Linear Regression, Logistic Regression, and Naive Bayes, tend to find less complex patterns than high-variance algorithms.
Underfitting	The model is too simple to capture the patterns within the data. The model performs poorly on data that it was trained on and on unseen data. High bias, low variance. High training error and high test error.
Overfitting	The model is too complicated or too specific, capturing trends that don't generalize. The model accurately predicts data that it was trained on but doesn't accurately predict unseen data. Low bias, high variance. Low training error and high test error.
Bias-Variance Trade-off	Bias-Variance Trade-off refers to finding a model with the right complexity, minimizing both the train and test error.

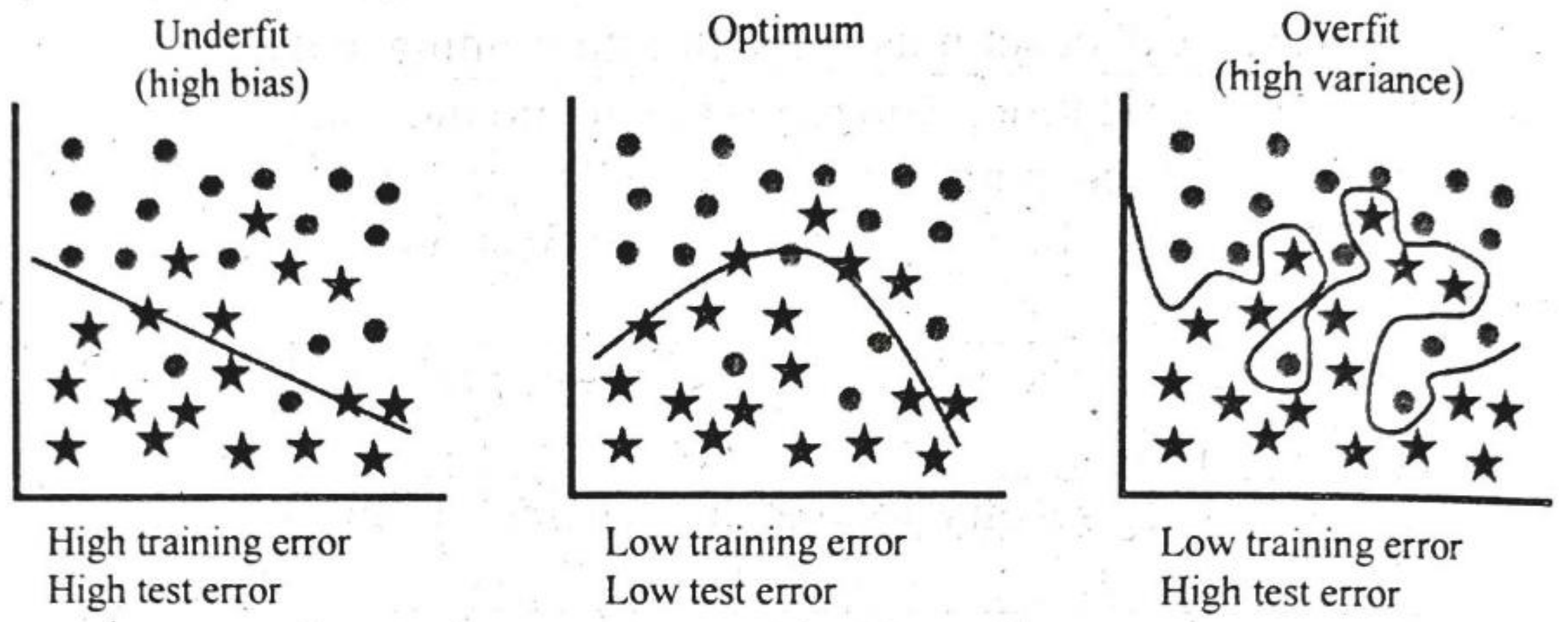
Scanned by TapScanner

2. How to train, validate and test algorithms? **[MODEL QUESTION]**

Answer:

Machine learning algorithms learn from examples. If you have good data, the more examples you provide, the better the model is at finding patterns in the data. However, be cautious of overfitting. As described in the first table, **overfitting** is where a model can accurately make predictions for data that it was trained on but can't generalize to other data.

Overfitting is why you split your data into training data, validation data, and test data. **Validation data** a **test data** are often referred to interchangeably, but they have distinct purposes.



Get familiar with the terminology around training, testing, and validating data:

Term	Description
Training data	This data is used to train the model and to fit the model parameters. It accounts for the largest proportion of data because you want the model to see as many examples as possible.
Validation data	This data is used to fit hyperparameters and for feature selection. Although the model never sees this data during training, by selecting particular features or hyperparameters based on this data, you introduce bias and risk overfitting.
Test data	This data is used to evaluate and compare your tuned models. Because this data wasn't seen during training or tuning, it can provide insight into whether your models generalize well to unseen data.
Cross-validation	The purpose of cross-validation is to evaluate whether a particular algorithm is suited for your data and use case. Cross-validation is also used for hyperparameter tuning and feature selection. The data is split into train and validation sets, though you need test data put to one side, too. You build a model with each of the slices of data. The final evaluation of the algorithm is the average performance of each of the models.
Hold-out method	The simplest version of cross-validation is the hold-out method, where you randomly split your data into two sets: a training set and a validation set. This method is the quickest because it requires building a model only once. However, with only one validation data set, you risk

Scanned by TapScanner

Term	Description
	that it contains easy or difficult observations to predict. Therefore, you might find that you're overfitting to this validation data, and it will perform poorly on a test set.
K-fold cross-validation	This method involves splitting the data into k-subsets. You then train a model k times, each time by using one of the k-subsets as its validation data. The training data is all other observations that aren't in the validation set. Your final evaluation is the average across all k folds.
Leave-one-out cross-validation	This method is the most extreme version of k-fold cross-validation. In it, your k is N (the number of observations in your data set). You train a model N separate times by using all data except for one observation and then validate its accuracy with its prediction for that observation. Although you're thoroughly evaluating how well this algorithm works with your data set, this method is expensive because it requires you to build N models.
Stratified cross-validation	This method enforces that the k-fold sets have similar proportions of observations for each class in either categorical features or the label.

3. How we can explore algorithms? **[MODEL QUESTION]**

Answer:

The algorithms that you explore must be driven by your use case. By first identifying what you're trying to achieve, you can narrow the scope of searching for solutions. Possible methods include, but aren't limited to algorithms for regression, classification, clustering, recommendations, and anomaly detection. You can also use the Algorithm Explorer to guide algorithm selection.

Regression

Regression algorithms are machine learning techniques for predicting continuous numerical values. They're supervised learning tasks, so they require labelled training examples.

Classification

Classification algorithms are machine learning techniques for predicting which category the input data belongs to. They're supervised learning tasks, so they require labeled training examples.

Clustering

Clustering algorithms are machine learning techniques to divide data into a number of groups where points in the groups have similar traits. They're unsupervised learning tasks and don't require labelled training examples.

Recommendation engines

Recommendation engines are created to predict a preference or rating that indicates a user's interest in an item or product. The algorithms that are used to create this system find similarities between the users, the items, or both.

Scanned by TapScanner

Anomaly detection
Anomaly detection is a technique that is used to identify unusual events or patterns that don't conform to expected behaviour. The items that are identified are often referred to as **anomalies** or **outliers**.

4. Explain methods of cross validation technique in machine learning. **[MODEL QUESTION]**

Answer:
In machine learning, we couldn't fit the model on the training data and can't say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. For this purpose, we use the cross-validation technique.
Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

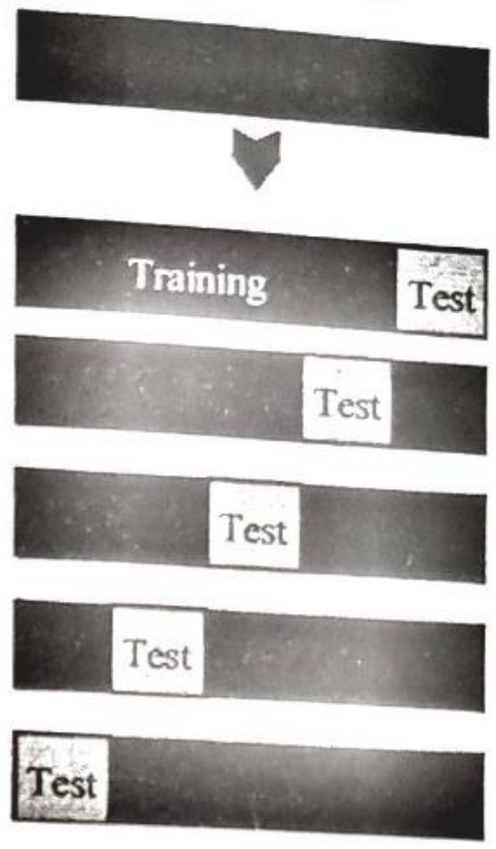
The three steps involved in cross-validation are as follows:

1. Reserve some portion of sample data-set.
2. Using the rest data-set train the model.
3. Test the model using the reserve portion of the data-set.

Methods of Cross Validation
Validation
In this method, we perform training on the 50% of the given data-set and rest 50% is used for the testing purpose. The major drawback of this method is that we perform training on the 50% of the dataset, it may possible that the remaining 50% of the data contains some important information which we are leaving while training our model i.e., higher bias.
LOOCV (Leave One Out Cross Validation)
In this method, we perform training on the whole data-set but leaves only one data-point of the available data-set and then iterates for each data-point. It has some advantages as well as disadvantages also.
An advantage of using this method is that we make use of all data points and hence it is low bias.
The major drawback of this method is that it leads to higher variation in the testing model as we are testing against one data point. If the data point is an outlier it can lead to higher variation. Another drawback is it takes a lot of execution time as it iterates over 'the number of data points' times.
K-Fold Cross Validation
In this method, we split the data-set into k number of subsets(known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.
Example
The diagram below shows an example of the training subsets and evaluation subsets generated in k-fold cross-validation. Here, we have total 25 instances. In first iteration we

Scanned by TapScanner

use the first 20 percent of data for evaluation, and the remaining 80 percent for training ([1-5] testing and [5-25] training) while in the second iteration we use the second subset of 20 percent for evaluation, and the remaining three subsets of the data for training ([5-10] testing and [1-5 and 10-25] training), and so on.



Total instances: 25
Value of k : 5

No. Iteration	Training set observations	Testing set observations
1	[5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]	[0 1 2 3 4]
2	[0 1 2 3 4 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]	[5 6 7 8 9]
3	[0 1 2 3 4 5 6 7 8 9 15 16 17 18 19 20 21 22 23 24]	[10 11 12 13 14]
4	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 20 21 22 23 24]	[15 16 17 18 19]
5	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]	[20 21 22 23 24]

Advantages of train/test split:

1. This runs K times faster than Leave One Out cross-validation because K-fold cross-validation repeats the train/test split K-times.
2. Simpler to examine the detailed results of the testing process.

Advantages of cross-validation:

1. More accurate estimate of out-of-sample accuracy.
2. More "efficient" use of data as every observation is used for both training and testing.

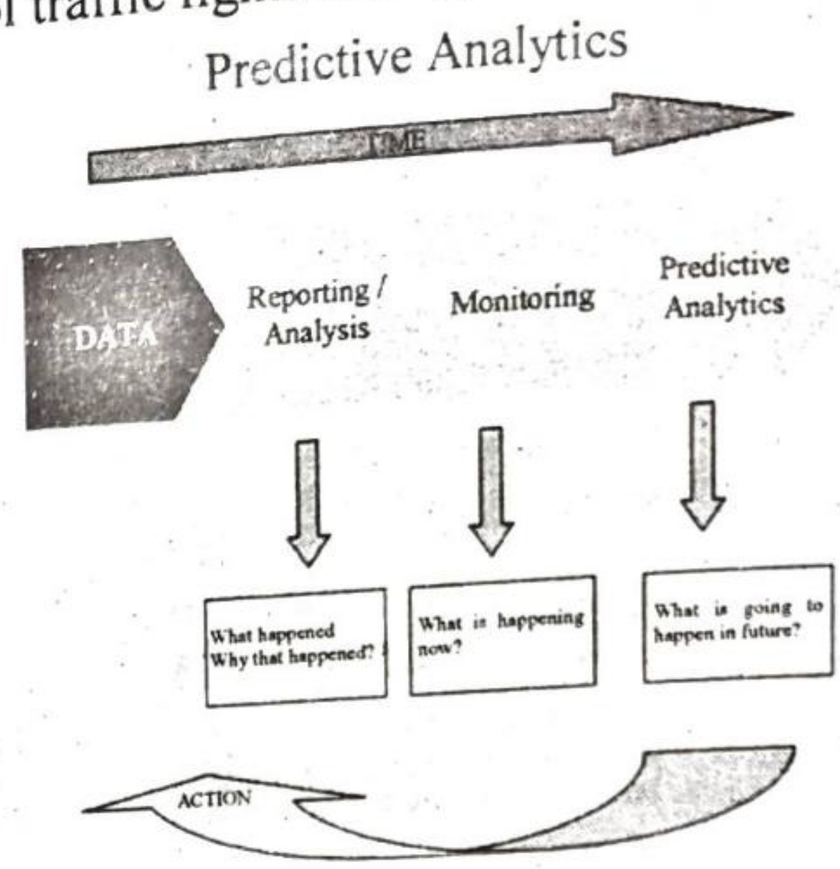
5. Explain predictive model in machine learning. **[MODEL QUESTION]**

Answer:

A **predictive model** is used for tasks that involve the prediction of one value using other values in the dataset. The learning algorithm attempts to discover and model the relationship between the *target feature* (the feature being predicted) and the other

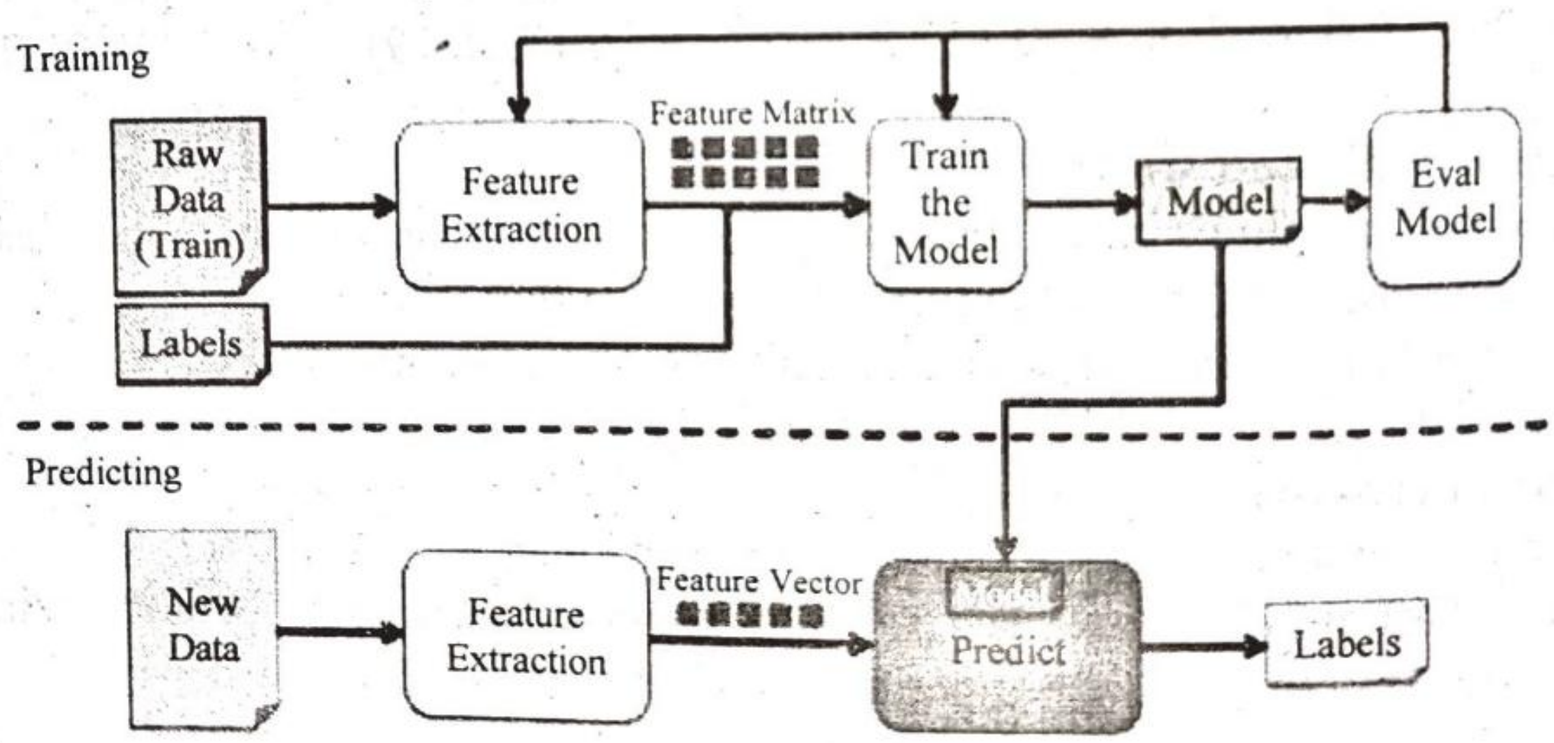
Scanned by TapScanner

features. Despite the common use of the word "prediction" to imply forecasting, predictive models need not necessarily foresee events in the future. For instance, a predictive model could be used to predict past events, such as the date of a baby's conception using the mother's present-day hormone levels. Predictive models can also be used in real time to control traffic lights during rush hours.



Because predictive models are given clear instruction on what they need to learn and how they are intended to learn it, the process of training a predictive model is known as **supervised learning**. The supervision does not refer to human involvement, but rather to the fact that the target values provide a way for the learner to know how well it has learned the desired task. Stated more formally, given a set of data, a supervised learning algorithm attempts to optimize a function (the model) to find the combination of feature values that result in the target output.

Supervised Learning Workflow



So, supervised learning consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the

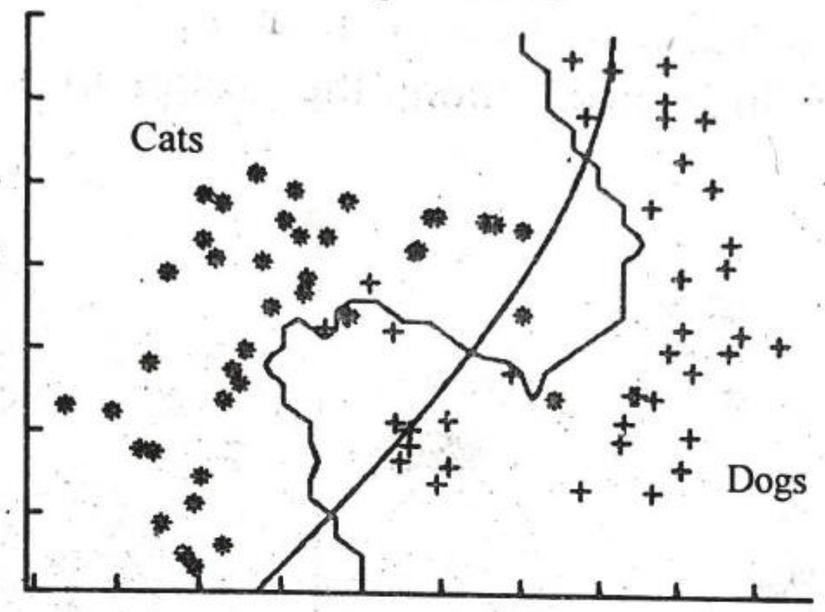
Scanned by TapScanner

training data. Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

The often used supervised machine learning task of predicting which category an example belongs to is known as classification. It is easy to think of potential uses for a classifier. For instance, you could predict whether:

- An e-mail message is spam
- A person has cancer
- A football team will win or lose
- An applicant will default on a loan

In classification, the target feature to be predicted is a categorical feature known as the class, and is divided into categories called levels. A class can have two or more levels, and the levels may or may not be ordinal. Because classification is so widely used in machine learning, there are many types of classification algorithms, with strengths and weaknesses suited for different types of input data.



Supervised learners can also be used to predict numeric data such as income, laboratory values, test scores, or counts of items. To predict such numeric values, a common form of **numeric prediction** fits linear regression models to the input data. Although regression models are not the only type of numeric models, they are, by far, the most widely used. Regression methods are widely used for forecasting, as they quantify in exact terms the association between inputs and the target, including both, the magnitude and uncertainty of the relationship.

6. Explain descriptive model in machine learning. **[MODEL QUESTION]**

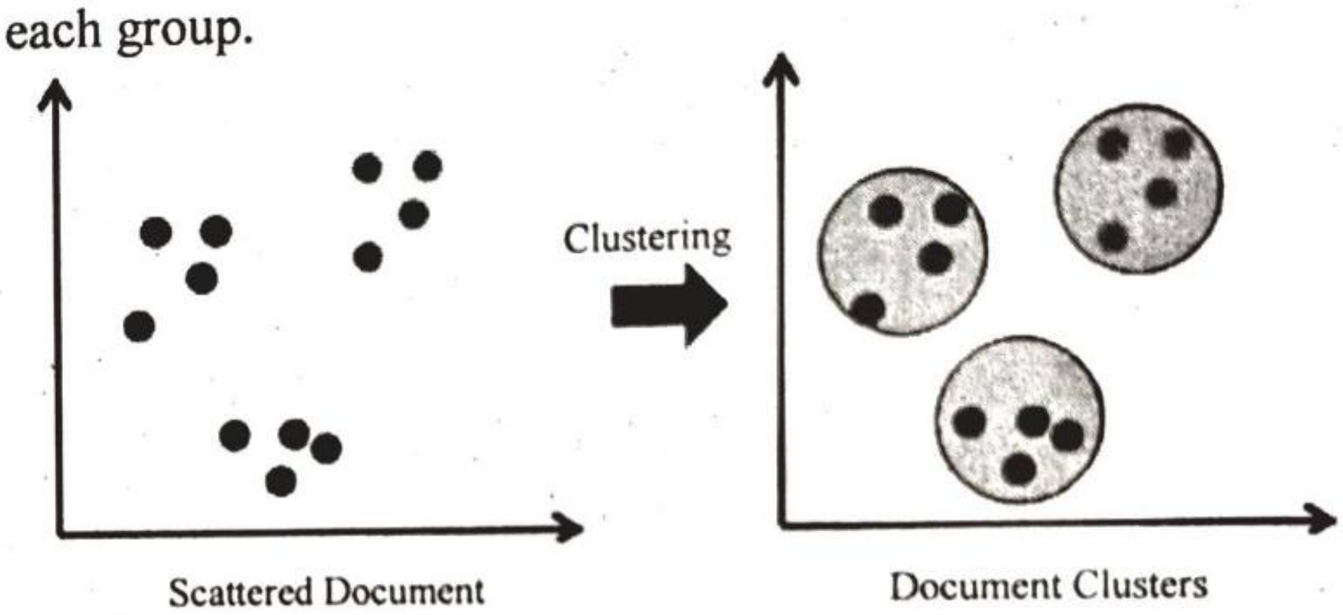
Answer:

A **descriptive model** is used for tasks that would benefit from the insight gained from summarizing data in new and interesting ways. As opposed to predictive models that predict a target of interest, in a descriptive model, no single feature is more important than any other. In fact, because there is no target to learn, the process of training a descriptive model is called **unsupervised learning**. Although it can be more difficult to think of applications for descriptive models, what good is a learner that isn't learning anything in particular - they are used quite regularly for *data mining*.

So, in unsupervised learning algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups,

Scanned by TapScanner

which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means. For example, the descriptive modeling task called **pattern discovery** is used to identify useful associations within data. Pattern discovery is often used for **market basket analysis** on retailers' transactional purchase data. Here, the goal is to identify items that are frequently purchased together, such that the learned information can be used to refine marketing tactics. For instance, if a retailer learns that swimming trunks are commonly purchased at the same time as sunglasses, the retailer might reposition the items more closely in the store or run a promotion to "up-sell" customers on associated items. The descriptive modeling task of dividing a dataset into homogeneous groups is called **clustering**. This is sometimes used for **segmentation analysis** that identifies groups of individuals with similar behaviour or demographic information, so that advertising campaigns could be tailored for particular audiences. Although the machine is capable of identifying the clusters, human intervention is required to interpret them. For example, given five different clusters of shoppers at a grocery store, the marketing team will need to understand the differences among the groups in order to create a promotion that best suits each group.



Lastly, a class of machine learning algorithms known as **meta-learners** is not tied to a specific learning task, but is rather focused on learning how to learn more effectively. A meta-learning algorithm uses the result of some learnings to inform additional learning. This can be beneficial for very challenging problems or when a predictive algorithm's performance needs to be as accurate as possible.

The following table lists only a fraction of the entire set of machine learning algorithms.

Model	Learning task
Supervised Learning Algorithms	
Nearest Neighbour	Classification
Naive Bayes	Classification
Decision Trees	Classification
Classification Rule Learners	**Classification**
Linear Regression	Numeric prediction
Model Trees	Numeric prediction
Regression Trees	Numeric prediction
Neural Networks	Dual use

Scanned by TapScanner

Model	Learning task
Support Vector Machines	Dual use
Unsupervised Learning Algorithms	
Association Rules	Pattern detection
k-means clustering	Clustering
Meta-Learning Algorithms	
Bagging	Dual use
Boosting	Dual use
Random Forests	Dual use

Scanned by TapScanner

SPARSE MODELING AND ESTIMATION

Multiple Choice Type Questions

1. Why is second order differencing in time series needed? [MODEL QUESTION]
   a) To remove stationarity
   b) To find the maxima or minima at the local point
   c) Both a and b
   d) None of the above

Answer: (c)

2. How do you handle missing or corrupted data in a dataset? [MODEL QUESTION]
   a) Drop missing rows or columns
   b) Replace missing values with mean/median/mode
   c) Assign a unique category to missing values
   d) All of the above

Answer: (d)

3. _____ also called representation learning algorithms [MODEL QUESTION]
   a) 1997 b) Feature Learning c) 1982 d) 1985

Answer: (b)

4. Feature learning can be [MODEL QUESTION]
   a) Supervised b) Unsupervised c) Both a and b d) None of these

Answer: (c)

5. Deep learning algorithms are _____ more accurate than machine learning algorithm in image classification. [MODEL QUESTION]
   a) 33% b) 37% c) 40% d) 41%

Answer: (d)

6. What is representation in deep learning? [MODEL QUESTION]
   a) It is a way to look at data to represent or encode
   b) It gets closer to the expected output
   c) RGB and HSV are two different examples of representations
   d) All of the above

Answer: (d)

7. What is learning in deep learning? [MODEL QUESTION]
   a) Learning, in the context of machine learning, describes an automatic search process for better representations
   b) A process that is "learned" from exposure to known examples of inputs and outputs
   c) Learning in school attending 5th grade
   d) Answers a & b

Answer: (d)

Scanned by TapScanner

8. What is deep in deep learning? [MODEL QUESTION]
   a) The deep in deep learning is a reference to any kind of deeper understanding achieved by the approach
   b) It stands for the idea of successive layers of representations in deep learning
   c) Answers a & b
   d) None of the above

Answer: (b)

9. Which statement is true? [MODEL QUESTION]
   a) Deep learning is a mathematical framework for learning representations from data
   b) Deep learning is a biological framework for learning representations from brain data
   c) Deep learning is an analogue framework for learning representations from data
   d) Deep learning is a digital framework for learning representations from data

Answer: (a)

10. What are layers in deep learning? [MODEL QUESTION]
   a) Deep neural networks do this input-to-target mapping via a deep sequence of simple data transformations called layers and that these data transformations are learned by exposure to examples
   b) It is a sort of membrane in brain
   c) None of the above
   d) All of the above

Answer: (a)

11. What is loss function in deep learning? [MODEL QUESTION]
   a) To control the output of a neural network, you need to be able to measure how far this output is from what you expected
   b) To calculate loss in banks
   c) These are true targets of data
   d) These are the predicted values only

Answer: (a)

12. Which statement is true? [MODEL QUESTION]
   a) The fundamental trick in deep learning is to use this score as a feedback signal to adjust the value of the weights a little, in a direction that will lower the loss score
   b) The fundamental trick in deep learning is to use this score as a feedback signal to adjust the value of the weights a little, in a direction that will higher the loss score
   c) Both of the above
   d) None of the above

Answer: (a)

Scanned by TapScanner

13. How deep learning learns from data? **[MODEL QUESTION]**

a) The incremental, layer-by-layer way in which increasingly complex representations are developed, and the fact that these intermediate incremental representations are learned jointly i.e., each layer being updated to follow both the representational needs of the layer above and the needs of the layer below

b) Deep learning learns from input data only

c) All of the above

d) None of the above

Answer: (a)

14. What is training loop in deep learning? **[MODEL QUESTION]**

a) With every step in the network processes, the weights are adjusted a little in the correct direction, and the loss score decreases

b) It is repeated a sufficient number of times (typically tens of iterations over thousands of examples), yields weight values that minimize the loss function

c) All of the above are true

d) None of the above are true

Answer: (a)

15. The two key ideas of deep learning for computer vision: **[MODEL QUESTION]**

a) Convolutional neural networks and back propagation

b) Deep neural networks and kernel functions

c) Support Vector Machines and loss functions

d) None of the above

Answer: (a)

Short Answer Type Questions

1. What are sparse features? **[MODEL QUESTION]**

Answer:

Features with sparse data are features that have mostly zero values. This is different from features with missing data. Examples of sparse features include vectors of one-hot-encoded words or counts of categorical data. On the other hand, features with dense data have predominantly non-zero values.

Sparse features are common in machine learning models, especially in the form of one-hot encoding. These features can result in issues in machine learning models like over fitting, inaccurate feature importance and high variance. It is recommended that sparse features should be pre-processed by methods like feature hashing or removing the feature to reduce the negative impacts on the results.

2. What is the difference between sparse data and missing data? **[MODEL QUESTION]**

Scanned by TapScanner

Answer:
When there is missing data, it means that many data points are unknown. On the other hand, if the data is sparse, all the data points are known, but most of them have zero value.
To illustrate this point, there are two types of features. The feature with sparse data has known values (= 0), but the feature with missing data has unknown values (= null). It is unknown what values should be in the null-valued rows.

Row	Feature with Sparse Data	Feature with Missing Data
1	0	Null
2	1	4
3	0	3
4	0	Null

Table 1: Sample data with two types of features

3. Why is machine learning difficult with sparse features? **[MODEL QUESTION]**

Answer:
Common problems with sparse features include:

1. If the model has many sparse features, it will increase the space and time complexity of models. Linear regression models will fit more coefficients, and tree-based models will have greater depth to account for all features.
2. Model algorithms and diagnostic measures might behave in unknown ways if the features have sparse data. Kuss [2002] shows that goodness-of-fit tests are flawed when the data is sparse.
3. If there are too many features, models fit the noise in the training data. This is called overfitting. When models overfit, they are unable to generalize to newer data when they are put in production. This negatively impacts the predictive power of models.
4. Some models may underestimate the importance of sparse features and given preference to denser features even though the sparse features may have predictive power. Tree-based models are notorious for behaving like this. For example, random forests overpredict the importance of features that have more categories than those features that have fewer categories.

4. What are the methods for dealing with sparse features? **[MODEL QUESTION]**

Answer:

1. Removing features from the model:
Sparse features can introduce noise, which the model picks up and increase the memory needs of the model. To remedy this, they can be dropped from the model. For example, rare words are removed from text mining models, or features with low variance are removed. However, sparse features that have important signals should not be removed in this process.
LASSO regularization can be used to decrease the number of features. Rule-based methods like setting a variance threshold for including features in the model might also be useful.

Scanned by TapScanner

2. Make the features dense:
Principal component analysis (PCA) - PCA methods can be used to project the features into the directions of the principal components and select from the most important components.
Feature hashing - In feature hashing, sparse features can be binned into the desired number of output features using a hash function. Care must be taken to choose a generous number of output features to prevent hash collisions.
3. Using models that are robust to sparse features:
Some versions of machine learning models are robust towards sparse data and may be used instead of changing the dimensionality of the data. For example, the entropy-weighted k-means algorithm is better suited to this problem than the regular k-means algorithm.

5. What is Time Series Analysis? **[MODEL QUESTION]**

Answer:
"Time series analysis is a statistical technique dealing in time series data, or trend analysis."
A time-series contains sequential data points mapped at a certain successive time duration, it incorporates the methods that attempt to surmise a time series in terms of understanding either the underlying concept of the data points in the time series or suggesting or making predictions.

- Forecasting data using time-series analysis comprises the use of some significant model to forecast future conclusions on the basis of known past outcomes.
- An objective of time series analysis is to explore and understand patterns in changes over time where these patterns signify the components of a time series including trends, cycles, and irregular movements.
- When such components reside in a time series, the data model must be considered for these patterns for generating accurate forecasts, such as future sales, GDP, and global temperatures.

Consider an example of a restaurant in which prediction is made on the number of customers as when will more customers appear in the restaurant at a specified time duration based on the previous appearance of customers with time.
We can use Time Series for multiple investigations to predict future as circadian rhythms, seasonal behaviours, trends, changes, etc. to interrogate the questions like predicted values, what is leading and lagging behind, connections and association, control, repetitions, and hidden pattern, etc.
Time series analysis is basically the recording of data at a regular interval of time, which could lead to taking a versed decision, crucial for trade and so have multiple applications such as Stock Market and Trends Analysis, Financial Analysis and forecasting, Inventory analysis, Census Analysis, Yield prediction, Sales forecasting, etc.

Scanned by TapScanner

6. Give examples of time series analysis. **[MODEL QUESTION]**

Answer:

Consider an example in the financial domain, the main objective is to recognize trends, seasonal behaviour, and correlation through the usage of time series analysis technique and producing filters based on the forecasts, this includes;

1. **To predict expected utilities-** For the perfect and successively trading, it is necessary to have accurate and reliable future predictions such as asset prices, variation in usage, products in demand in statistical form through market research, and time-series dataset.
2. **Simulate series-** After getting statistical output data of financial time series that can be used for creating simulations of future events. It helps us to determine the count of trades, expected trading costs and returns, required financial and technical investment, several risks in trading, etc.
3. **Presume relationship-** Recognition of the relationship between the time series and other quantities gives us trading signs to improve the existing fashion of trading. For example, to know the spreading of foreign exchange pair and its variation with a proposal, estimated trades can be inferred for a certain period for forecasting a widespread to reduce transaction costs.

Moreover, consider the simple example of a train steadiness; it includes the Time Series Data model that can be implemented to represent information such as stopping trains. It can be used;

1. To aggregate over time which train is late at which location, and
2. How many passengers were there on the train?

Furthermore, in order to examine social media hashtags and how frequently a hashtag is utilized in a specified interval of time. More and more, how many responses this particular hashtag used during such an interval.

A table below shows the particular field to analyze using time-series data and what possible instance could be investigated through it,

Sl. No.	Field	Example topics
1	Economics	Gross Domestic Product (GDP), Consumer Price Index (CPI) and unemployment rates
2	Social sciences	Birth rates, population, migration data, political indicators
3	Epidemiology	Disease rates, mortality rates, mosquito populations
4	Medicine	Blood pressure tracking, weight tracking, cholesterol measurements, heart rate monitoring
5	Physical sciences	Global temperatures, monthly sunspot observations, pollution levels.

7. What is deep learning? **[MODEL QUESTION]**

Answer:

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain—albeit far from matching its ability—allowing it to "learn" from large

Scanned by TapScanner

amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.
Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

8. How deep learning works? **[MODEL QUESTION]**

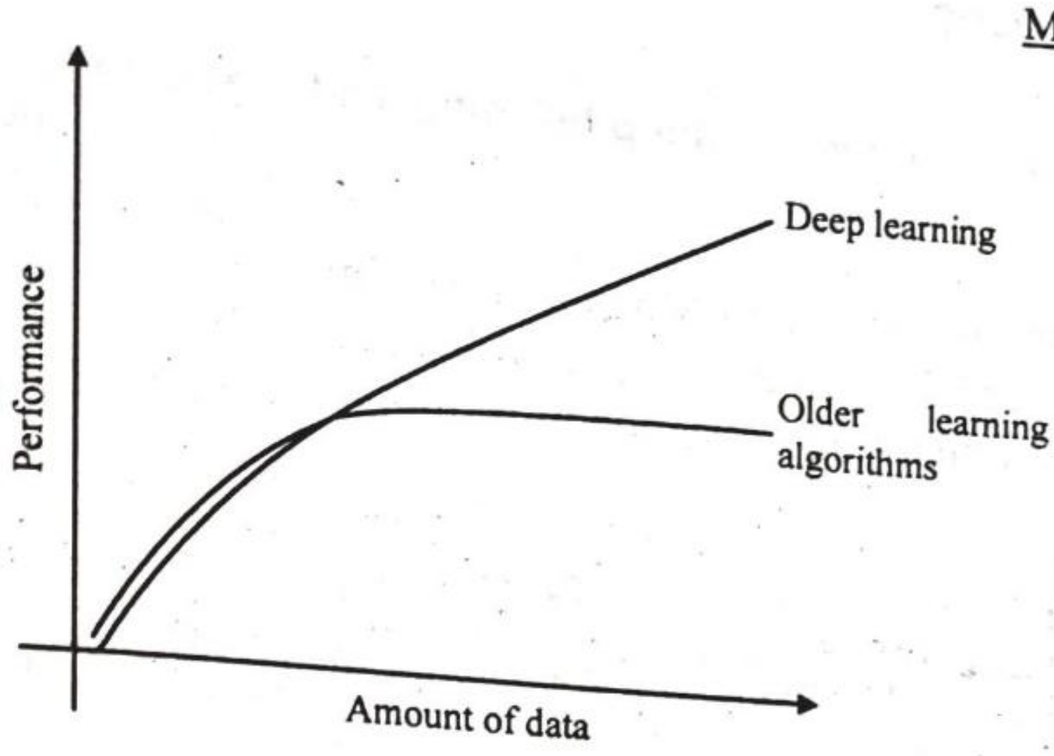
Answer:
Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.
Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.
Another process called back propagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and back propagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.
The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. For example,

- **Convolutional neural networks** (CNNs), used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an object recognition challenge for the first time.
- **Recurrent neural network** (RNNs) are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

9. Why we need deep learning? **[MODEL QUESTION]**

Answer:
When the volume of data increases, Machine learning techniques, no matter how optimized, starts to become inefficient in terms of performance and accuracy, whereas Deep learning performs so much better in such cases.

Scanned by TapScanner



10. What amount of data is called big?

Answer: **[MODEL QUESTION]**

The **Oxford Dictionary** defines Big Data as "Extremely large data sets that may be analyzed computationally to reveal patterns, trends and associations, especially relating to human behaviour and interactions".

"Big Data" refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze. This definition is intentionally subjective and incorporates a moving definition of how big a dataset needs to be in order to be considered Big Data – i.e., we don't define Big Data in terms of being larger that a certain number of terabytes (thousands of gigabytes). We assume that, as technology advances over time, the size of datasets that qualify as Big Data will also increase, also note that the definition can vary by sector, depending on what kinds of software tools are commonly available and what sizes of datasets are common in a particular industry. with those caveats, Big Data in many sectors today will range from a few dozen terabytes to multiple petabytes (thousands of terabytes).

Essentially, Big Data is a term which is used to mean a massive volume of both structured and unstructured data that is so large that it is difficult to process using traditional databases and software techniques. In most enterprise scenarios the volume of data is too big, or it moves too fast, or it exceeds current processing capacity. Interestingly, it is difficult to agree upon a standard definition because different people will use the term Big Data in different contexts and that will determine what they mean when they talk about "Big Data".

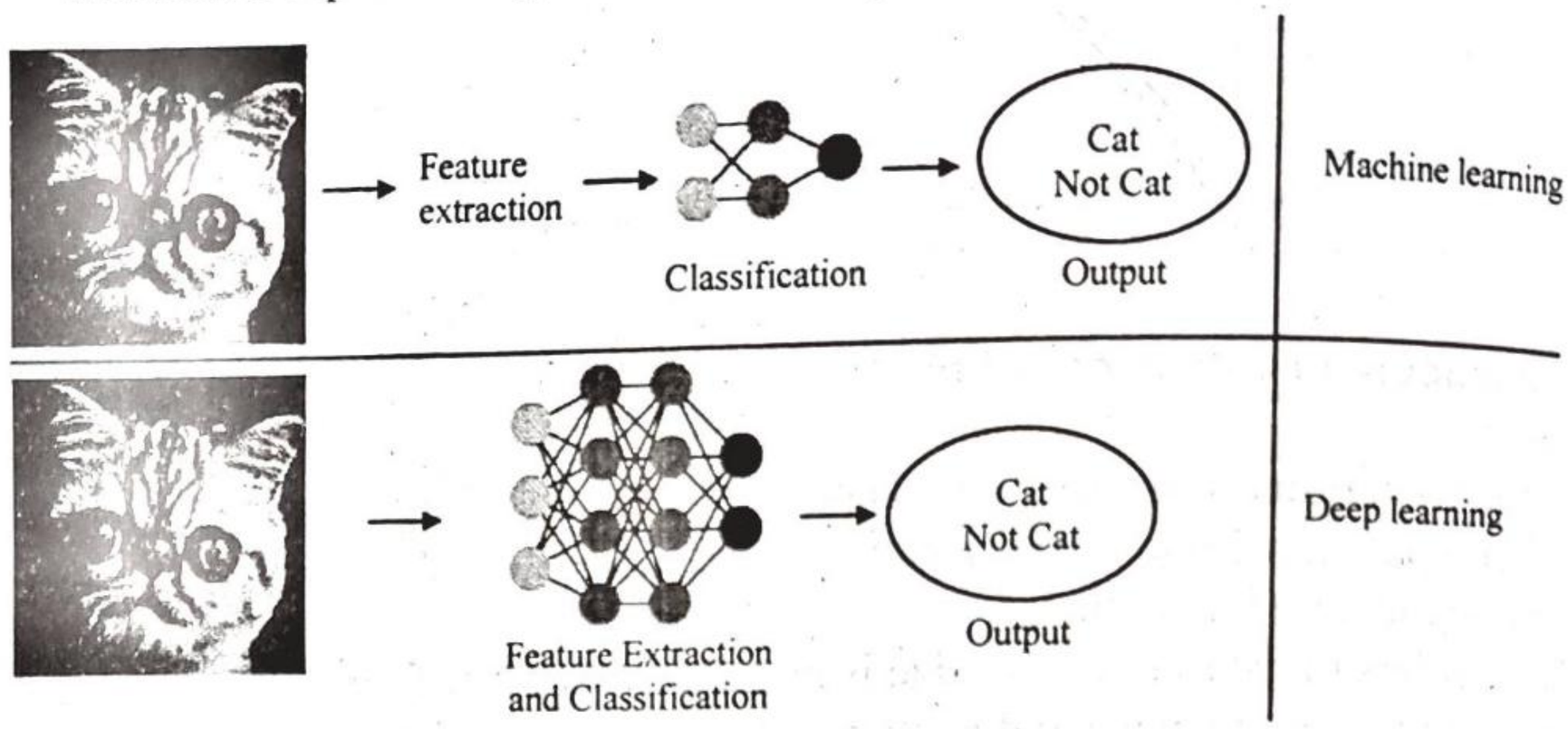
When someone is talking about data-storage capacity, Big Data means the size or volume of the data. When the computing capability is under discussion, Big Data perhaps means the processing capability when discussing computing. The vendors specializing in this technology will refer to the technology and tools which will be used to analyze this data when they talk about Big Data. Organizations will talk more about data generation and accumulation when they mention Big Data. However, there are certain characteristics of Big Data which will be there most of the time and they are: size, unstructured nature and the need for processing to make sense of it.

Scanned by TapScanner

11. What are differences between deep learning and machine learning? **[MODEL QUESTION]**

Answer:

- Deep Learning is a subset of Machine Learning.
- In Machine Learning features are provided manually.
- Whereas Deep Learning learns features directly from the data.



12. What is representation learning? **[MODEL QUESTION]**

Answer:

Representation learning is a class of machine learning approaches that allow a system to discover the representations required for feature detection or classification from raw data. The requirement for manual feature engineering is reduced by allowing a machine to learn the features and apply them to a given activity.

In representation learning, data is sent into the machine, and it learns the representation on its own. It is a way of determining a data representation of the features, the distance function, and the similarity function that determines how the predictive model will perform. Representation learning works by reducing high-dimensional data to low-dimensional data, making it easier to discover patterns and anomalies while also providing a better understanding of the data's overall behaviour.

Basically, Machine learning tasks such as classification frequently demand input that is mathematically and computationally convenient to process, which motivates representation learning. Real-world data, such as photos, video, and sensor data, has resisted attempts to define certain qualities algorithmically. An approach is to examine the data for such traits or representations rather than depending on explicit techniques.

Feature representation/learning is an essential step for many computer vision tasks (like image classification) and is broadly categorized as 1) deep feature representation; 2) shallow feature representation. With the development of deep neural networks, many deep feature representation methods have been proposed and obtained many remarkable results. However, they are limited to real-world applications due to the high demand for storage space and computation ability.

Scanned by TapScanner

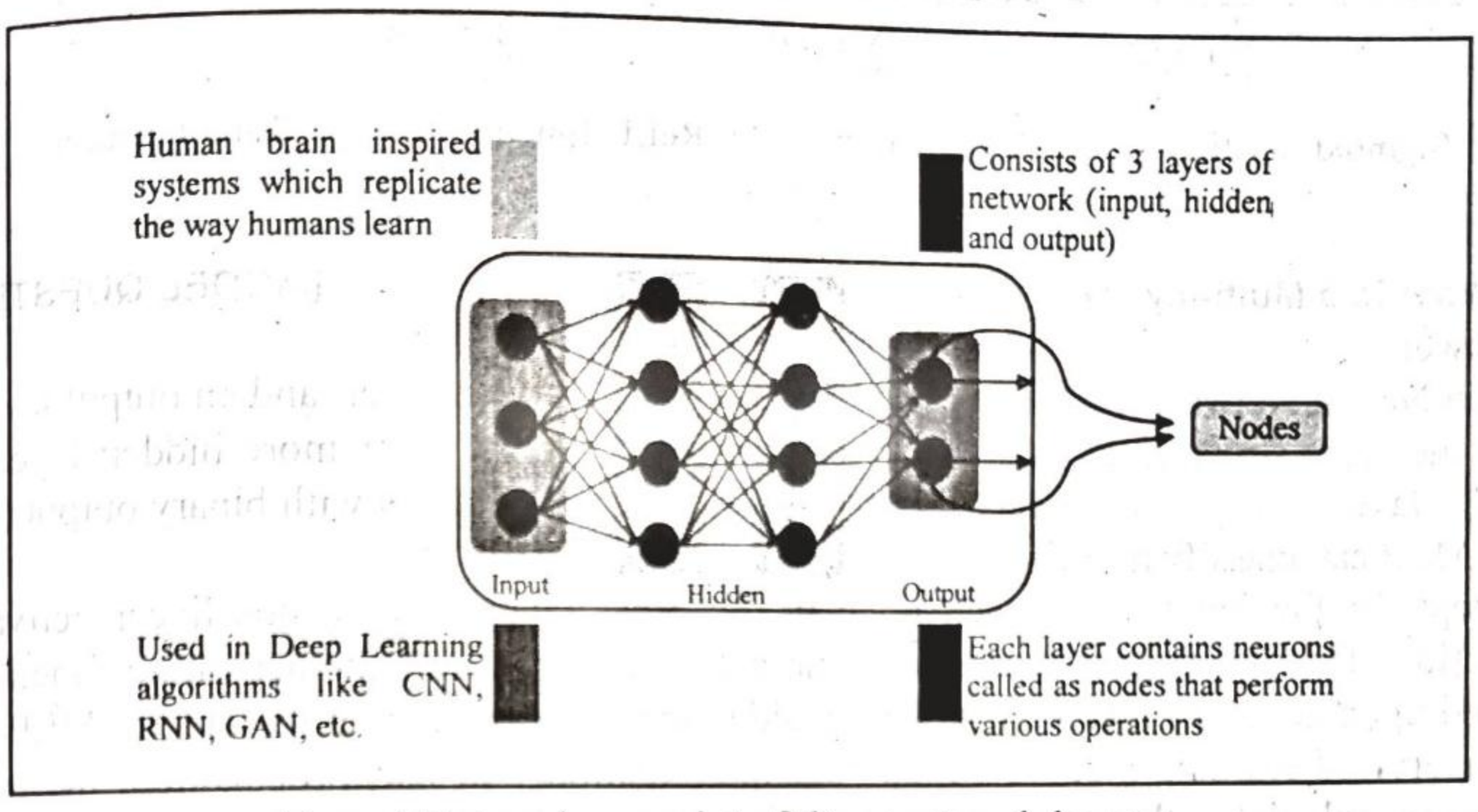
Long Answer Type Questions

1. What is a neural network?

[MODEL QUESTION]

Answer:

Neural Networks replicate the way humans learn, inspired by how the neurons in our brains fire, only much simpler..



The most common Neural Networks consist of three network layers:

1. An input layer
2. A hidden layer (this is the most important layer where feature extraction takes place, and adjustments are made to train faster and function better)
3. An output layer

Each sheet contains neurons called "nodes," performing various operations. Neural Networks are used in deep learning algorithms like CNN, RNN, GAN, etc.

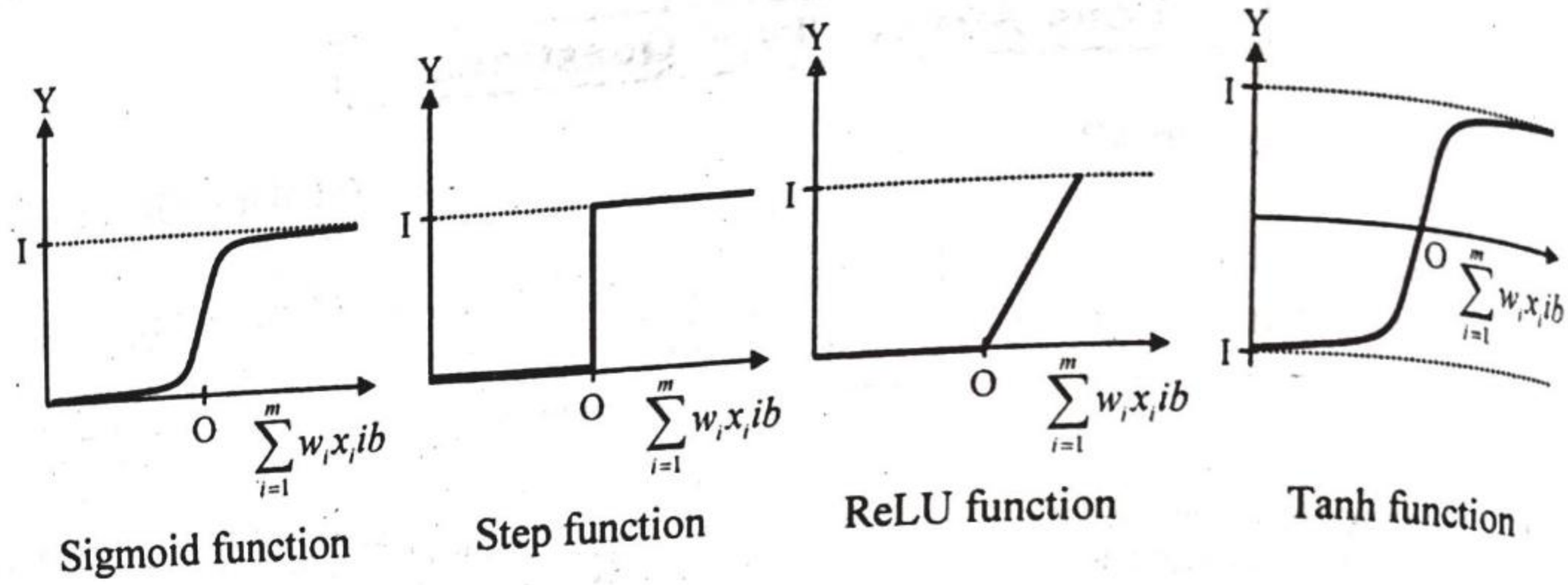
2. What is the role of activation functions in a neural network?

[MODEL QUESTION]

Answer:

At the most basic level, an activation function decides whether a neuron should be fired or not. It accepts the weighted sum of the inputs and bias as input to any activation function. Step function, Sigmoid, ReLU, Tanh, and Softmax are examples of activation functions.

Scanned by TapScanner



3. What is a Multi-layer Perceptron (MLP)? **[MODEL QUESTION]**

Answer:

As in Neural Networks, MLPs have an input layer, a hidden layer, and an output layer. It has the same structure as a single layer perceptron with one or more hidden layers. A single layer perceptron can classify only linear separable classes with binary output (0,1), but MLP can classify non-linear classes.

Except for the input layer, each node in the other layers uses a non-linear activation function. This means the input layers, the data coming in, and the activation function is based upon all nodes and weights being added together, producing the output. MLP uses a supervised learning method called "backpropagation." In backpropagation, the neural network calculates the error with the help of cost function. It propagates this error backward from where it came (adjusts the weights to train the model more accurately).

4. What do you understand by forward propagation? **[MODEL QUESTION]**

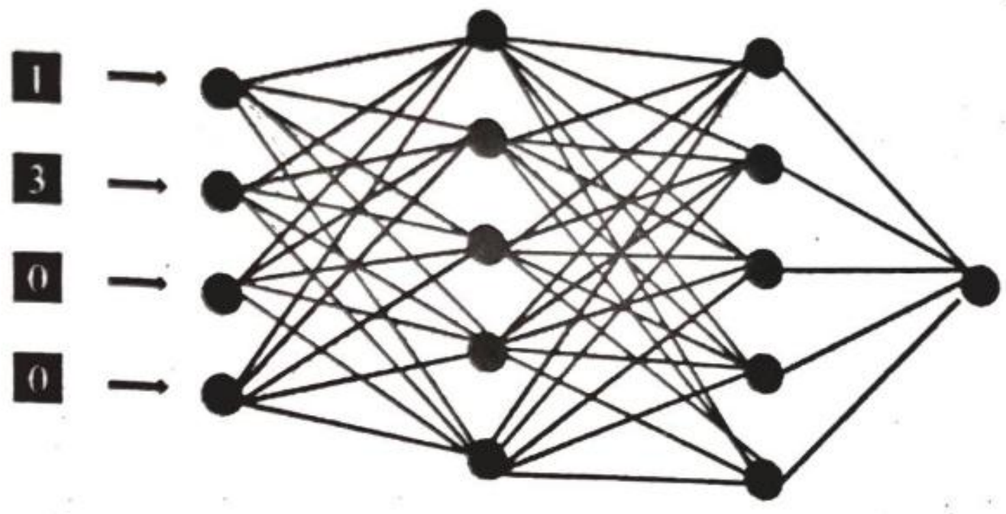
Answer:

Before we start let's have a look at a sample data set which will be used to train the neural network.

Gender	Age	Hypertension	Does smoke	Stroke
1	3	0	0	0
1	58	1	1	1
0	8	0	0	0
0	70	0	1	1
1	14	0	0	0
0	47	0	0	0
0	52	0	1	1
0	75	0	0	0
0	32	0	1	0

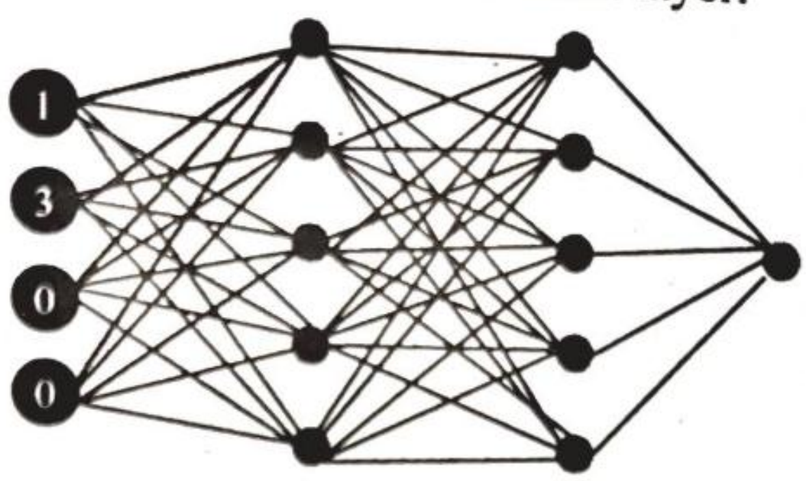
This data set has four independent features represented by first four columns and one dependent feature represented by last column. For the sake of simplicity let's focus on the first row and try to understand forward propagation with this single row input. Each of the independent features in the first row will be passed to the input layer of the neural network to act as input.

Scanned by TapScanner



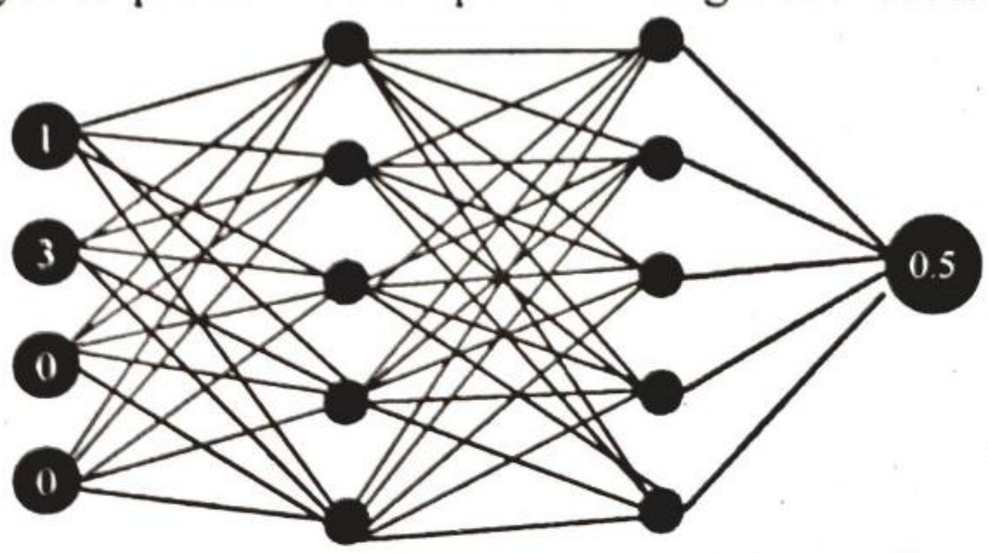
Now we start off the forward propagation by randomly initializing the weights of all neurons. These weights are depicted by the edges connecting two neurons. Hence the weights of a neuron can be more appropriately thought of as weights between two layers since edges connect two layers.

Now let's talk about this first neuron in the first hidden layer.



This neuron receives input from all the neurons in the previous layer and combines it with its own set of weights represented by the thick edges. And after applying some activation function, calculates the output known as Hidden Activation for this neuron. The output of the hidden activation will serve as input for the next layer neurons. Similarly, the hidden activations for all of the neurons in the layer are calculated.

So all the activations for the first hidden layer have been calculated. These activations will serve as inputs to the layer after them. Once the hidden activations for the last hidden layer are calculated, they are combined by a final set of weights between the last hidden layer and the output layer to produce an output for a single row observation.

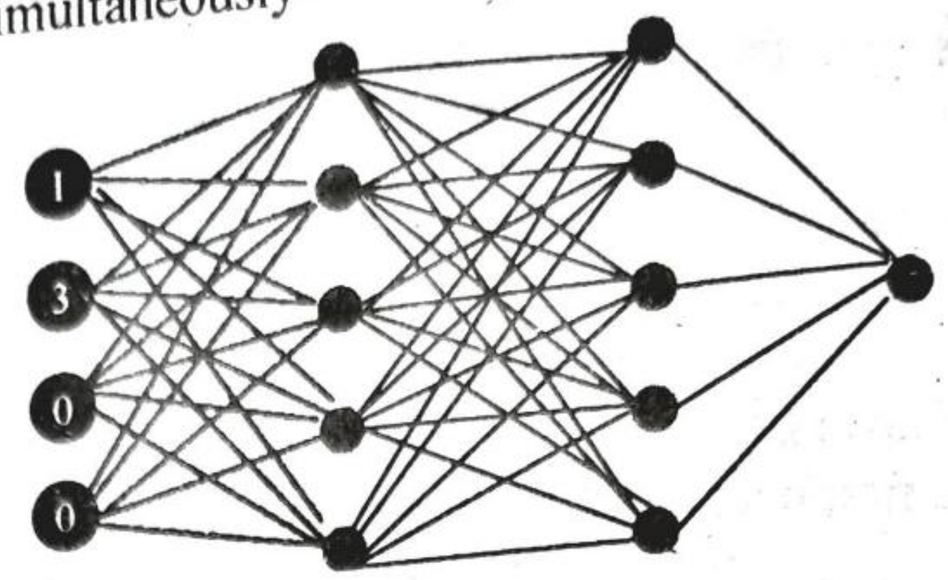


These calculations of the first row features are 0.5 and the predicted value becomes 0.5 against the actual value of 0.

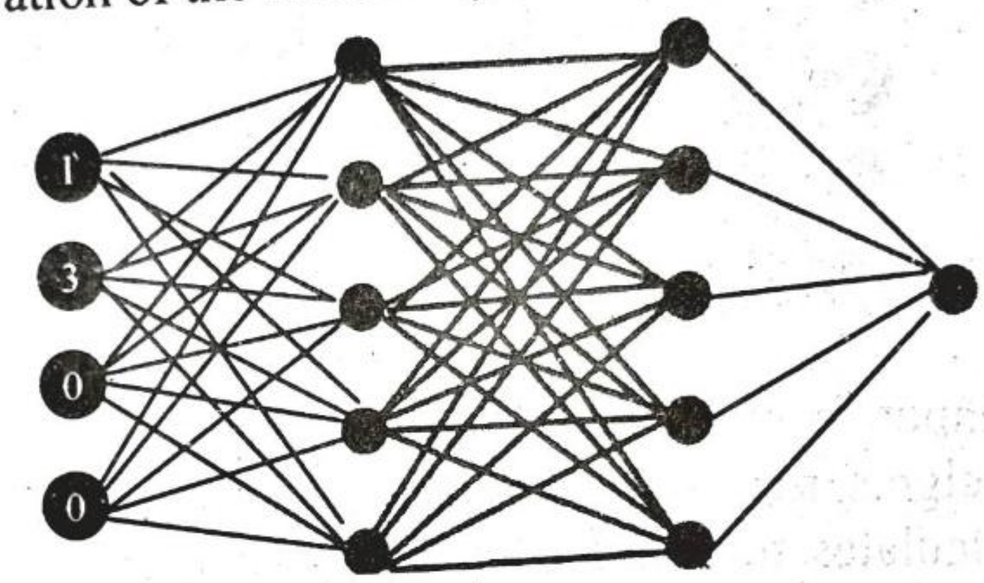
Actual	Predicted
0	0.5

Scanned by TapScanner

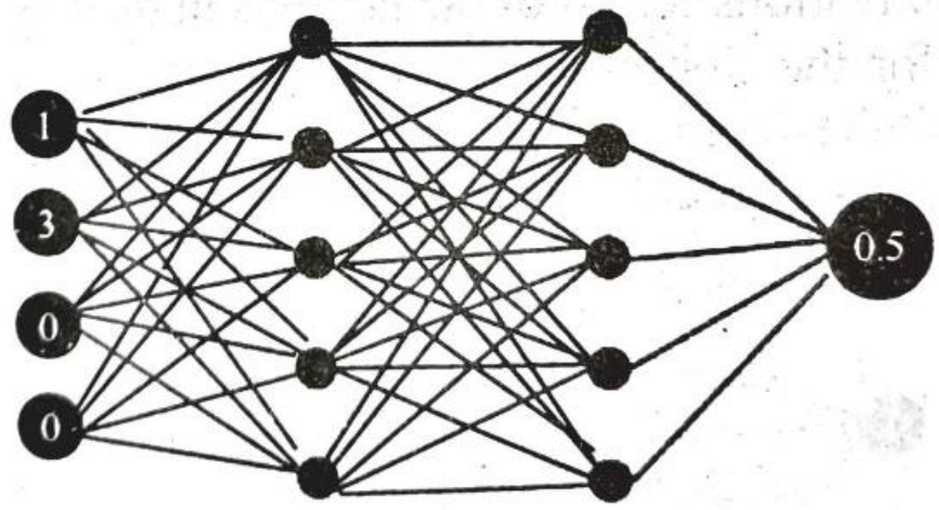
Now if you notice that carefully we calculated the hidden activations for each of the neurons sequentially, that is one after the other. In fact, one optimization that we can make to the calculation is to calculate them in parallel. Since the calculation of neurons is independent of each other they can be computed in parallel easily. We can calculate the first hidden activation simultaneously-



then calculate the activation of the second layer simultaneously,



and then finally calculate the output.



This series of calculations which takes us from the input to output is called Forward Propagation.

We will now understand the error generated during the predictions and what can be the possible reasons behind those errors.

Errors in Neural Network

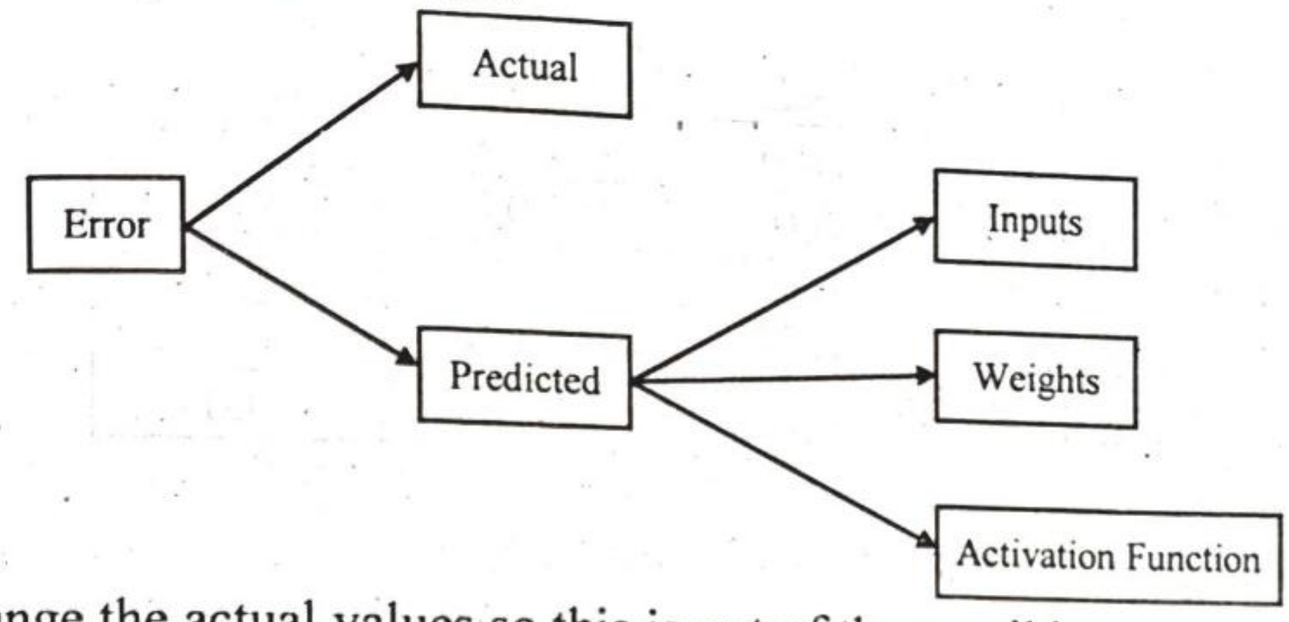
So far we have seen how forward propagation helps us in calculating outputs. Let's say for a particular row the actual target is 0 and the predicted target is 0.5. We can use this predicted value to calculate the error for a particular row. The kind of error that we have chosen here is Squared Error.

Scanned by TapScanner

$$\text{MSE} = \frac{1}{n}\sum_{n}^{1}\left(Y_i - \hat{Y}_i\right)^2$$

There can be other error calculation formulas but for the sake of simplicity, we have chosen this one.

Now that we have an error what can be possible reasons for this and what can be changed to reduce errors? We know that the error is directly dependent upon two quantities- the actual values and the predicted values.



We cannot change the actual values so this is out of the possible suspects. Next, we know that the predicted values which are also the hidden activations are dependent upon 3 things-

- Inputs to the output layer i.e., the hidden activation of the second layer
- Weights between the second hidden layer and the output
- And the activation function present at the output layer

We cannot directly modify the second hidden layer activations since they're calculated at each step. So inputs are also out. We can change both the activation functions and weights but we cannot iteratively change activation functions as it would change distributions during the training time. So activation function is also out of the question.

Now, what remains are the weights between the second hidden layer and output. In fact not only can we change the weights between the second hidden layer and the output but also between the first and the second hidden layer and between the input layer and the first hidden layer. But we cannot change the weights in arbitrary amounts to check out what weights produce the lowest error as this would amount to an infinite amount of time to check what sort of weights produce the lowest error.

Interestingly what we can do is we can use a smart strategy to adjust the weights and reduce the errors. Specifically what we are interested is in two things-

- First is the amount of change in error on changing the weight by a small amount
- And second is the direction of that change.

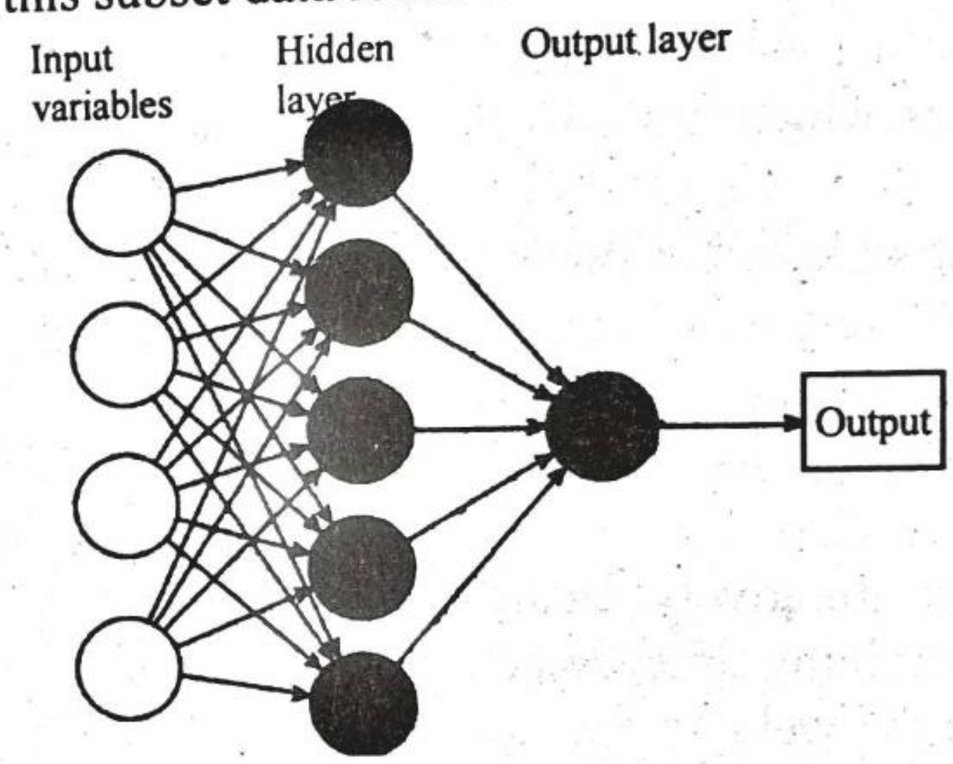
Scanned by TapScanner

5. What do you understand by Back propagation? **[MODEL QUESTION]**

Answer:
Backward Propagation is the preferable method of adjusting or correcting the weights to reach the minimized loss function.
Let's say we want to use the neural network to predict house prices. For our understanding purpose here, we will take a subset dummy dataset having four input variables and six observations here with the input having a dimension of 4*5:

X1	X2	X3	X4
0.00632	18.0	2.31	0.0
0.02731	0.0	7.07	0.0
0.027279	0.0	7.07	0.0
0.03237	0.0	2.18	0.0
0.06905	0.0	2.18	0.0

The neural network for this subset data looks like below:



The architecture of the neural network is [4, 5, 1] with:

- 4 independent variables, Xs in the input layer
- 5 nodes in the hidden layer, and
- Since we have a regression problem at hand, we will have one node in the output layer.

A Neural Network operates by:

1. Initializing the weights with some random values, which are mostly between 0 and 1.
2. Compute the output to calculate the loss or the error term.
3. Then, adjust the weights so that to minimize the loss.

We repeat these three steps until have reached the optimum solution of the minimum loss function or exhausted the pre-defined epochs (i.e., the number of iterations).
Now, the computation graph after applying the sigmoid activation function is:

Scanned by TapScanner

$$\boxed{z_1 = w_{ih}^T x + b_{ih}} \rightarrow \boxed{h_1 = \sigma(z_1)} \rightarrow \boxed{z_2 = w_{ho}^T \times h_1 + b_{ho}} \rightarrow \boxed{O = \sigma(z_0)}$$

$$\downarrow$$

$$E = \frac{(Y-O)^2}{2}$$

Building on this, the first step in Backward Propagation to calculate the error. In our regression problem, we shall take the loss function $= \frac{(Y - Y\wedge)^2}{2}$, where Y is actual values and $Y\wedge$ is predicted values. For simplicity, replacing $Y\wedge$ with O, so the error E becomes $= \frac{(Y-O)^2}{2}$.

Our goal is to minimize the error that is clearly dependent on Y, which is the actual observed values, and on the output, which is further is dependent on the:

- input values
- coefficients or betas of the input variables
- biases, the activation function, and
- optimizers

Now, we can neither change the input variables nor the actual Y values however, we can change the other factors. The activation function and the optimizers are the tuning parameters – and we can change these based on our requirement.

The other two factors: the coefficients or betas of the input variables $(W_i s)$ and the biases (b_{ho}, b_{ih}) are updated using the Gradient descent algorithm with the following equation:

$$W_{\text{new}} = W_{\text{old}} - \left(\alpha \times \frac{dE}{dW}\right)$$

where,

W_{new} = the new weight of X_i

W_{old} = the old weight of the X_i

α = learning rate

$\frac{dE}{dW}$ is the partial derivative of the error for each of the Xs. It is the rate of change of the error to the change in weight.

In the backward propagation, we adjust these weights or the betas in the output. The weights and biases between the respective input, hidden and output layers we have here are W_{ih}, b_{ih}, W_{ho} and b_{ho}:

- W_{ih} : weight between the input and the hidden layer
- b_{ih} : bias between the input and the hidden layer
- W_{ho} : weight between the hidden and the output layer
- b_{ho} : bias between the hidden and the output layer

Scanned by TapScanner

In the first iteration, we randomly initialize the weights. In the second iteration, we change the weights of the hidden layer that is closest to the output layer. In this case, we go from the output layer, hidden layer, and then to the input layer.

Now, we have to calculate how much each of these weights $(W_i s)$ and biases $(b_i s)$ contribute to the error term. For this, we need to calculate the rate of change of error to the respective weights and bias parameters.

In other words, we need to compute the terms: $\frac{dE}{dW_{ih}}, \frac{dE}{db_{ih}}, \frac{dE}{dW_{ho}}$ and $\frac{dE}{db_{ho}}$. This is not a direct task. It is a series of steps involving the Chain Rule.

The weight, W_{ho}, between the hidden and the output layer:

From the above graph we can see that the error E is not directly dependent on the W_{ho}:

- The error term is dependent on the Output O
- Output O is further dependent on Z_2 and
- Z_2 is dependent on W_{ho}

Therefore we employ the chain rule to compute the rate of change in error to the change in weight W_{ho} and it becomes:

$$\frac{dE}{dW_{ho}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dW_{ho}}$$

Now, we take the partial derivatives of each of these individual terms:

$$E = \frac{(Y-O)^2}{2}.$$

1. The partial derivative of error with respect to Output is:

$$\frac{dE}{dO} = \frac{2 \times (Y-O) \times (-1)}{2} = (O-Y)$$

2. The partial derivative of Output with respect to Z_2, as output $O =$ Sigmoid of Z_2 and the derivative of sigmoid is:

$$\frac{dO}{dZ_2} = \text{sigmoid}(Z_2) \times (1 - \text{sigmoid}(Z_2)) = O \times (1-O)$$

3. The partial derivative of Z_2 with respect to W_{ho} is:

$$\frac{dZ_2}{dW_{ho}} = \frac{d\left(W_{ho}^T \times h_1 + b_{h0}\right)}{dW_{ho}}$$

$$\frac{dZ_2}{dW_{ho}} = \frac{d\left(W_{ho}^T \times h_1\right)}{dW_{ho}} + d\left(\frac{b_{ho}}{W_{ho}}\right) = h_1 + 0 = h_1$$

Scanned by TapScanner

Therefore, $\frac{dE}{dW_{ho}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dW_{ho}}$ becomes:

$$\frac{dE}{dW_{ho}} = (O - Y) \times O \times (1 - O) \times h_1$$

Similarly, we will calculate the contribution for each of the other parameters in this manner.

For the bias, b_{ho}, between the hidden and the output layer:

$$\frac{dE}{db_{ho}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{db_{ho}}$$

$$\frac{dE}{db_{ho}} = (O - Y) \times O \times (1 - O) \times 1$$

The weight, W_{ih}, between the input and the hidden layer:

From the above graph we can see that the terms are dependent as below:

- Error term is dependent on the Output O
- Output O is dependent on Z_2
- Z_2 this time is dependent on h_1
- h_1 is dependent on Z_1, and
- Z_1 is dependent on W_{ih}

$$\frac{dE}{dW_{ih}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dh_1} \times \frac{dh_1}{dZ_1} \times \frac{dZ_1}{dW_{ih}}$$

So, this time, apart from the initial above $\frac{dE}{dO}, \frac{dO}{dZ_2}$, we have the partial derivatives as follow:

1. The partial derivative of Z_2 with respect to h_1 is:

$$\frac{dZ_2}{dh_1} = \frac{d\left(W_{ho}^T \times h_1 + b_{ho}\right)}{dh_1}$$

$$\frac{dZ_2}{dh_1} = \frac{d\left(W_{ho}^T \times h_1\right)}{dh_1} + d\left(\frac{b_{ho}}{h_1}\right) = W_{ho} + 0 = W_{ho}$$

2. The partial derivative of h_1 with respect to Z_1, as $h_1 =$ Sigmoid of Z_1 and the derivative of sigmoid is:

$$\frac{dh_1}{dZ_1} = \text{sigmoid}(Z_1) \times (1 - \text{sigmoid}(Z_1)) = h_1 \times (1 - h_1)$$

3. The partial derivative of Z_1 with respect to W_{ih} is: X

$$\frac{dZ_1}{dW_{ih}} = \frac{d\left(W_{ih}^T \times X + b_{ih}\right)}{dW_{ih}}$$

Scanned by TapScanner

$$\frac{dZ_1}{dW_{ih}} = \frac{d\left(W_{ih}^T \times X\right)}{dW_{ih}} + d\left(\frac{b_{ih}}{W_{ih}}\right) = X + 0 = X$$

Hence, the equation after plugging the partial derivative value is:

$$\frac{dE}{dW_{ih}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dh_1} \times \frac{dh_1}{dZ_1} \times \frac{dZ_1}{dW_{ih}}$$

$$\frac{dE}{dW_{ih}} = (O - Y) \times O \times (1 - O) \times W_{ho} \times h_1 (1 - h_1) \times X$$

The bias, b_{ih}, between the input and the hidden layer:

$$\frac{dE}{db_{ih}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dh_1} \times \frac{dh_1}{dZ_1} \times \frac{dZ_1}{db_{ih}}$$

$$\frac{dE}{db_{ih}} = (O - Y) \times O \times (1 - O) \times W_{ho} \times h_1 (1 - h_1) \times 1$$

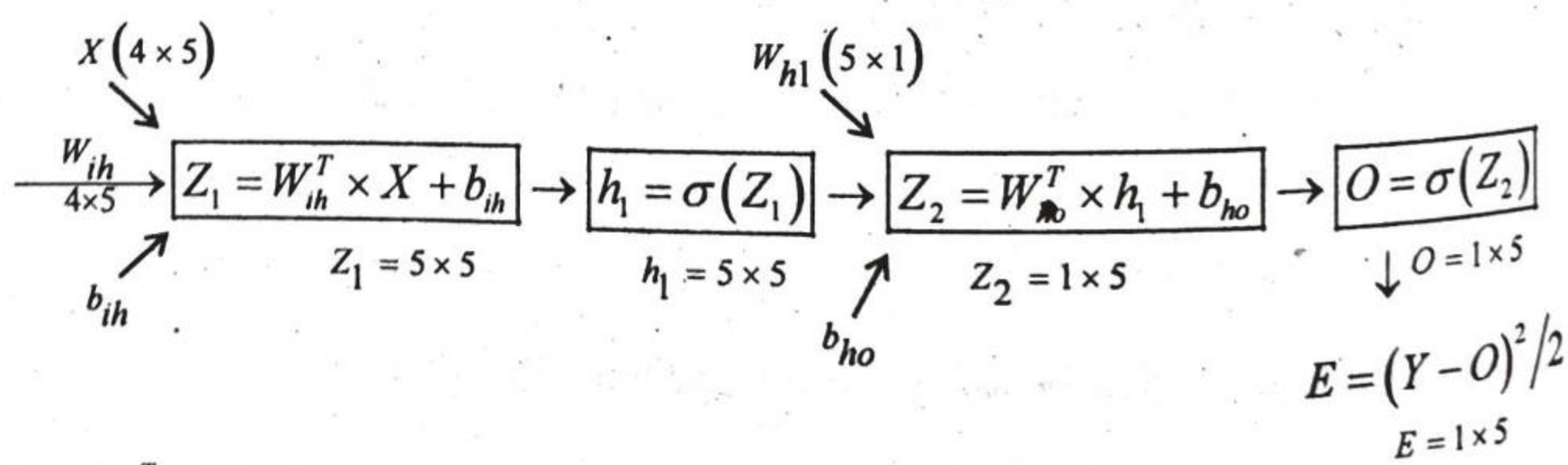
Now, that we have computed these terms we can update the parameters using the following respective update equations:

- $W_{ih} = W_{ih} - \left(\alpha \times \frac{dE}{dW_{ih}}\right)$
- $b_{ih} = b_{ih} - \left(\alpha \times \frac{dE}{db_{ih}}\right)$
- $W_{ho} = W_{ho} - \left(\alpha \times \frac{dE}{dW_{ho}}\right)$
- $b_{ho} = b_{ho} - \left(\alpha \times \frac{dE}{db_{ho}}\right)$

Now, moving to another method to perform backward propagation ...

Matrix form of the backward propagation

The backward propagation can also be solved in the matrix form. The computation graph for the structure along with the matrix dimensions is:



$$Z_1 = W_{ih}^T \times X + b_{ih}$$

Scanned by TapScanner

where,

- W_{ih} is the weight matrix between the input and the hidden layer with the dimension of 4×5
- W_{ih}^T, is the transpose of W_{ih}, having shape 5×4
- X is the input variables having dimension 4×5, and
- b_{ih} is a bias term, has a single value here as considering the same for all the neurons.

$$Z_2 = W_{ho}^T \times h_1 + b_{ho}$$

where,

- W_{ho} is the weight matrix between the hidden and the output layer with shape 5×1
- W_{ho}^T, is the transpose of W_{ho} having a dimension of 1×5
- h_1 is the result after the applying activation function on the outcome from the hidden layer with a shape of 5×5, and
- b_{ho} is the bias term, has a single value here as considering the same for all the neurons.

To summarize, the four equations of the rate of change of error with the different parameters are:

$$\frac{dE}{dW_{ho}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dW_{ho}} = (O-Y) \times O \times (1-O) \times h_1$$

$$\frac{dE}{db_{ho}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{db_{ho}} = (O-Y) \times O \times (1-O) \times 1$$

$$\frac{dE}{dW_{ih}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dh_1} \times \frac{dh_1}{dZ_1} \times \frac{dZ_1}{dW_{ih}} = (O-Y) \times O \times (1-O) \times W_{ho} \times h_1 (1-h_1) \times X$$

$$\frac{dE}{db_{ih}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dh_1} \times \frac{dh_1}{dZ_1} \times \frac{dZ_1}{db_{ih}} = (O-Y) \times O \times (1-O) \times W_{ho} \times h_1 (1-h_1) \times 1$$

Now, lets' see how we can perform matrix multiplication on each of these equations. For the weight matrix between the hidden and the output layer, W_{ho}.

Let us understand how the shape of this W_{ho} must be similar to that of the shape of $\frac{dE}{dW_{ho}}$, which is to used to update the weight in the following equation:

$$W_{ho} = W_{ho} - \left(\alpha \times \frac{dE}{dW_{ho}} \right)$$

Scanned by TapScanner

We saw above that $\frac{dE}{dW_{ho}}$ is computed using the chain rule and is of the result:

$$\frac{dE}{dW_{ho}} = \frac{dE}{dO} \times \frac{dO}{dZ_2} \times \frac{dZ_2}{dW_{ho}}$$

$$\frac{dE}{dW_{ho}} = (O - Y) \times O \times (1 - O) \times h_1$$

Breaking the individual components of this above equation we see each part's dimension:

$\frac{dE}{dO} = (O - Y)$ as both O and Y have the same shape of 1×5. Hence, $\frac{dE}{dO}$ is of dimension 1×5.

$\frac{dO}{dZ_2} = O \times (1 - O)$ having a shape of 1×5, and

$\frac{dZ_2}{dW_{ho}} = h_1$, which is of the shape 5×5

Now, performing matrix multiplication on this equation, as we know, matrix multiplication can be done when the number of columns of the first matrix must be equal to the number of rows of the second matrix. Where this matrix multiplication rule defies, we will take the transpose of one of the matrices to conduct the multiplication.

On applying this our equation takes the form of:

$$\frac{dE}{dW_{ho}} = \frac{dZ_2}{dW_{ho}} \cdot \left[\frac{dE}{dO} \times \frac{dO}{dZ_2} \right]^T$$

$$\frac{dE}{dW_{ho}} = (5 \times 5) \cdot \left[(1 \times 5) \times (1 \times 5) \right]^T$$

$$\frac{dE}{dW_{ho}} = (5 \times 5) \cdot (5 \times 1) = 5 \times 1$$

Therefore, the shape of $\frac{dE}{dW_{ho}} 5 \times 1$ is the same as that of $W_{ho} 5 \times 1$ which will be updated using the Gradient Descent update equation.

In the same manner, we can find perform the backward propagation for the other parameters using matrix multiplication and the respective equations will be:

$$\frac{dE}{dW_{ho}} = \frac{dZ_2}{dW_{ho}} \cdot \left[\frac{dE}{dO} \times \frac{dO}{dZ_2} \right]^T$$

$$\frac{dE}{db_{ho}} = \frac{dZ_2}{db_{ho}} \cdot \left[\frac{dE}{dO} \times \frac{dO}{dZ_2} \right]^T$$

Scanned by TapScanner

$$\frac{dE}{dW_{ih}} = \frac{dZ_1}{dW_{ih}} \cdot \left[\frac{dh_1}{dZ_1} \times \frac{dZ_2}{dh_1} \cdot \left(\frac{dE}{dO} \times \frac{dO}{dZ_2} \right) \right]^T$$

$$\frac{dE}{db_{ih}} = \frac{dZ_1}{db_{ih}} \cdot \left[\frac{dh_1}{dZ_1} \times \frac{dZ_2}{dh_1} \cdot \left(\frac{dE}{dO} \times \frac{dO}{dZ_2} \right) \right]^T$$

Where, (·) dot is the dot product and × is the element wise product.

6. What is the difference between a feedforward neural network and recurrent neural network?

[MODEL QUESTION]

Answer:

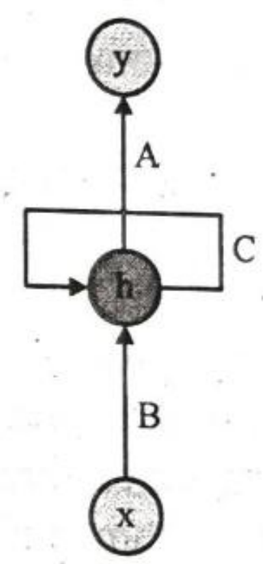
In this deep learning interview question, the interviewee expects you to give a detailed answer.

A Feedforward Neural Network signals travel in one direction from input to output. There are no feedback loops; the network considers only the current input. It cannot memorize previous inputs (e.g., CNN).

A Recurrent Neural Network's signals travel in both directions, creating a looped network. It considers the current input with the previously received inputs for generating the output of a layer and can memorize past data due to its internal memory.

Recurrent Neural Network

- Signals travel in both directions, making it a looped network.
- Considers the current input along with the previously received inputs for generating the output of a layer.
- Has the ability to memorize past data due to its internal memory.



Recurrent Neural Network

7. How we can implement time series analysis in machine learning?

[MODEL QUESTION]

Answer:

It is a well-known fact that Machine Learning is a powerful technique in imagining, speech and natural language processing for a huge explicated dataset available. On the other hand,

1. Problems based on time series do not have usually interpreted datasets, even as data is collected from various sources so exhibit substantial variations in terms of features, properties, attributes, temporal scales, and dimensionality.
2. Time series analysis requires such sorting algorithms that can allow it to learn time-dependent patterns across multiples models different from images and speech.

Scanned by TapScanner

3. Various machine learning tools such as classification, clustering, forecasting, and anomaly detection depend upon real-world business applications.

Among various defined applications, **discussing here Time series forecasting**, it is an important area of machine learning because there are multiple problems involving time components for making predictions. There are **multiple models and methods** used as approaches for time series forecasting, let's understand them more clearly;

ML methods for Time-Series Forecasting:

1. In the Univariate Time-series Forecasting method, forecasting problems contain only two variables in which one is time and the other is the field we are looking to forecast.
   - For example, if you want to predict the mean temperature of a city for the coming week, now one parameter is time(week) and the other is a city.
   - Another example could be when measuring a person's heart rate per minute through using past observations of heart rate only. Now one parameter is time(minute) and the other is a heart rate.
2. On the other hand, in the Multivariate Time-series Forecasting method, forecasting problems contain multiple variables keeping one variable as time fixed and others will be multiple in parameters.

Consider the same example, predicting the temperature of a city for the coming week, the only difference would come here now temperature will consider impacting factors such as

- Rainfall and time duration of raining,
- Humidity,
- Wind speed,
- Precipitation,
- Atmospheric pressure, etc,

And then the temperature of the city will be predicted accordingly. All these factors are related to temperature and impact it vigorously.

ML models for Time-Series Forecasting

- **ARIMA Model:** As mentioned in the above section, it is a combination of three different models itself, AR, MA and I, where
  1. "AR" reflects the evolving variable of interest is regressed on its own prior values,
  2. "MA" infers that the regression error is the linear combination of error terms values happened at various stages of time priorly, and
  3. "I" shows the data values are replaced by the difference between their values and the previous values.

  Combined "ARIMA" tries to fit the data into the model, and also ARIMA depends on the accuracy over a broad width of time series.
- **ARCH/GARCH Model:** Being the extended model of its common version GARCH, Autoregressive Conditional Heteroscedasticity (ARCH) is the most volatile model for

Scanned by TapScanner

time series forecasting, and are well trained for catching dynamic variations of volatility from time series.

- **Vector Autoregressive Model or VAR model:** It gives the independencies between various time-series data which as a generalization of the Univariate Autoregression Model.
- **LSTM:** Long-short term memory (LSTM) is a deep learning model, it is a kind of Recurrent Neural Network (RNN) to read the sequence dependencies. It enables us to handle long structures during training the dataset and creates predictions according to previous data.

8. What are methods of representation learning? **[MODEL QUESTION]**

Answer:

We must employ representation learning to ensure that the model provides invariant and untangled outcomes in order to increase its accuracy and performance. In this section, we'll look at how representation learning can improve the model's performance in three different learning frameworks: supervised learning, unsupervised learning.

Supervised Methods

Supervised Dictionary Learning

Dictionary learning creates a set of representative elements (dictionary) from the input data, allowing each data point to be represented as a weighted sum of the representative elements. By minimizing the average representation error (across the input data) and applying L1 regularization to the weights, the dictionary items and weights may be obtained i.e., the representation of each data point has only a few nonzero weights.

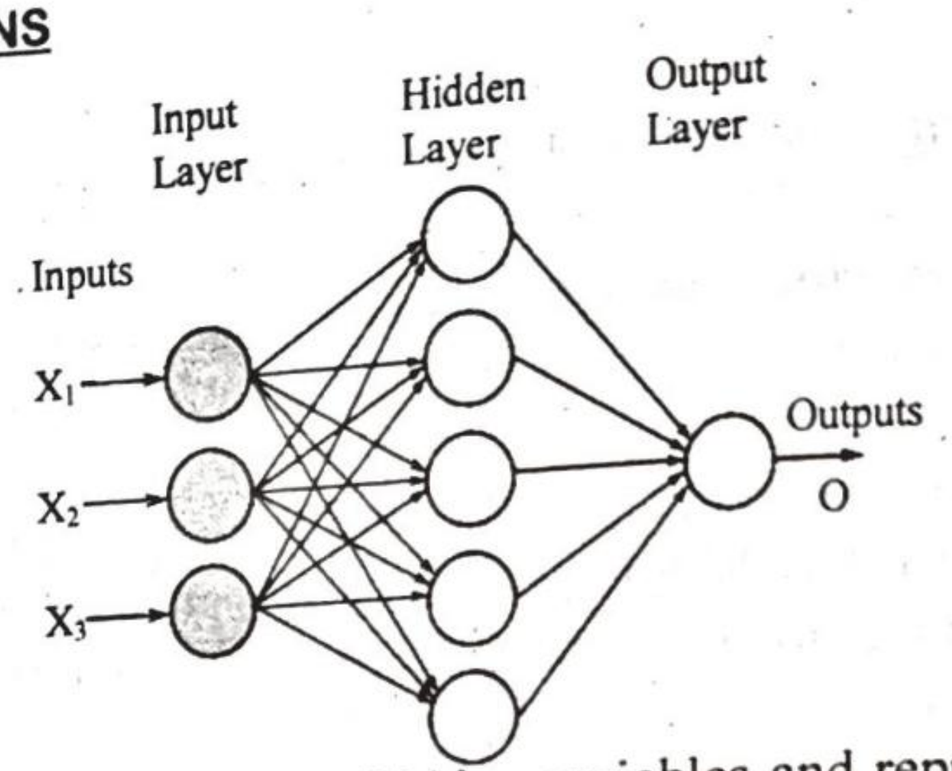
For optimizing dictionary elements, supervised dictionary learning takes advantage of both the structure underlying the input data and the labels. The supervised dictionary learning technique uses dictionary learning to solve classification issues by optimizing dictionary elements, data point weights, and classifier parameters based on the input data.

A minimization problem is formulated, with the objective function consisting of the classification error, the representation error, an L1 regularization on the representing weights for each data point (to enable sparse data representation), and an L2 regularization on the parameters of the classification algorithm.

Multi-Layer Perceptron

The perceptron is the most basic neural unit, consisting of a succession of inputs and weights that are compared to the ground truth. A multi-layer perceptron, or MLP, is a feed-forward neural network made up of layers of perceptron units. MLP is made up of three-node layers: an input, a hidden layer, and an output layer. MLP is commonly referred to as the vanilla neural network because it is a very basic artificial neural network.

Scanned by TapScanner



This notion serves as a foundation for hidden variables and representation learning. Our goal in this theorem is to determine the variables or required weights that can represent the underlying distribution of the entire data so that when we plug those variables or required weights into unknown data, we receive results that are almost identical to the original data. In a word, artificial neural networks (ANN) assist us in extracting meaningful patterns from a dataset.

Neural Networks

Neural networks are a class of learning algorithms that employ a "network" of interconnected nodes in various layers. It's based on the animal nervous system, with nodes resembling neurons and edges resembling synapses. The network establishes computational rules for passing input data from the network's input layer to the network's output layer, and each edge has an associated weight.

The relationship between the input and output layers, which is parameterized by the weights, is described by a network function associated with a neural network. Various learning tasks can be achieved by minimizing a cost function over the network function (w) with correctly defined network functions.

Unsupervised Methods

Learning Representation from unlabeled data is referred to as unsupervised feature learning. Unsupervised Representation learning frequently seeks to uncover low-dimensional features that encapsulate some structure beneath the high-dimensional input data.

K-Means Clustering

K-means clustering is a vector quantization approach. An n-vector set is divided into k clusters (i.e., subsets) via K-means clustering, with each vector belonging to the cluster with the closest mean. Despite the use of inferior greedy techniques, the problem is computationally NP-hard.

K-means clustering divides an unlabeled collection of inputs into k groups before obtaining centroids-based features. These characteristics can be honed in a variety of ways. The simplest method is to add k binary features to each sample, with each feature j having a value of one of the K-means learned jth centroid is closest to the sample under consideration. Cluster distances can be used as features after being processed with a radial basis function.

Scanned by TapScanner

Local Linear Embedding

LLE is a non-linear learning strategy for constructing low-dimensional neighbour-preserving representations from high-dimensional (unlabeled) input. LLE's main goal is to reconstruct high-dimensional data using lower-dimensional points while keeping some geometric elements of the original data set's neighbours.

There are two major steps in LLE. The first step is "neighbour-preserving," in which each input data point Xi is reconstructed as a weighted sum of K nearest neighbour data points, with the optimal weights determined by minimizing the average squared reconstruction error (i.e., the difference between an input point and its reconstruction) while keeping the weights associated with each point equal to one.

The second stage involves "dimension reduction," which entails searching for vectors in a lower-dimensional space that reduce the representation error while still using the optimal weights from the previous step.

The weights are optimized given fixed data in the first stage, which can be solved as a least-squares problem. Lower-dimensional points are optimized with fixed weights in the second phase, which can be solved using sparse eigenvalue decomposition.

Unsupervised Dictionary Mining

For optimizing dictionary elements, unsupervised dictionary learning does not use data labels and instead relies on the structure underlying the data. Sparse coding, which seeks to learn basic functions (dictionary elements) for data representation from unlabeled input data, is an example of unsupervised dictionary learning.

When the number of vocabulary items exceeds the dimension of the input data, sparse coding can be used to learn overcomplete dictionaries. K-SVD is an algorithm for learning a dictionary of elements that allows for sparse representation.

Deep Architectures Methods

Deep learning architectures for feature learning are inspired by the hierarchical architecture of the biological brain system, which stacks numerous layers of learning nodes. The premise of distributed representation is typically used to construct these architectures: observable data is generated by the interactions of many diverse components at several levels.

Restricted Boltzmann Machine (RBMs)

In multilayer learning frameworks, RBMs (restricted Boltzmann machines) are widely used as building blocks. An RBM is a bipartite undirected network having a set of binary hidden variables, visible variables, and edges connecting the hidden and visible nodes. It's a variant of the more general Boltzmann machines, with the added constraint of no intra-node connections. In an RBM, each edge has a weight assigned to it. The connections and weights define an energy function that can be used to generate a combined distribution of visible and hidden nodes.

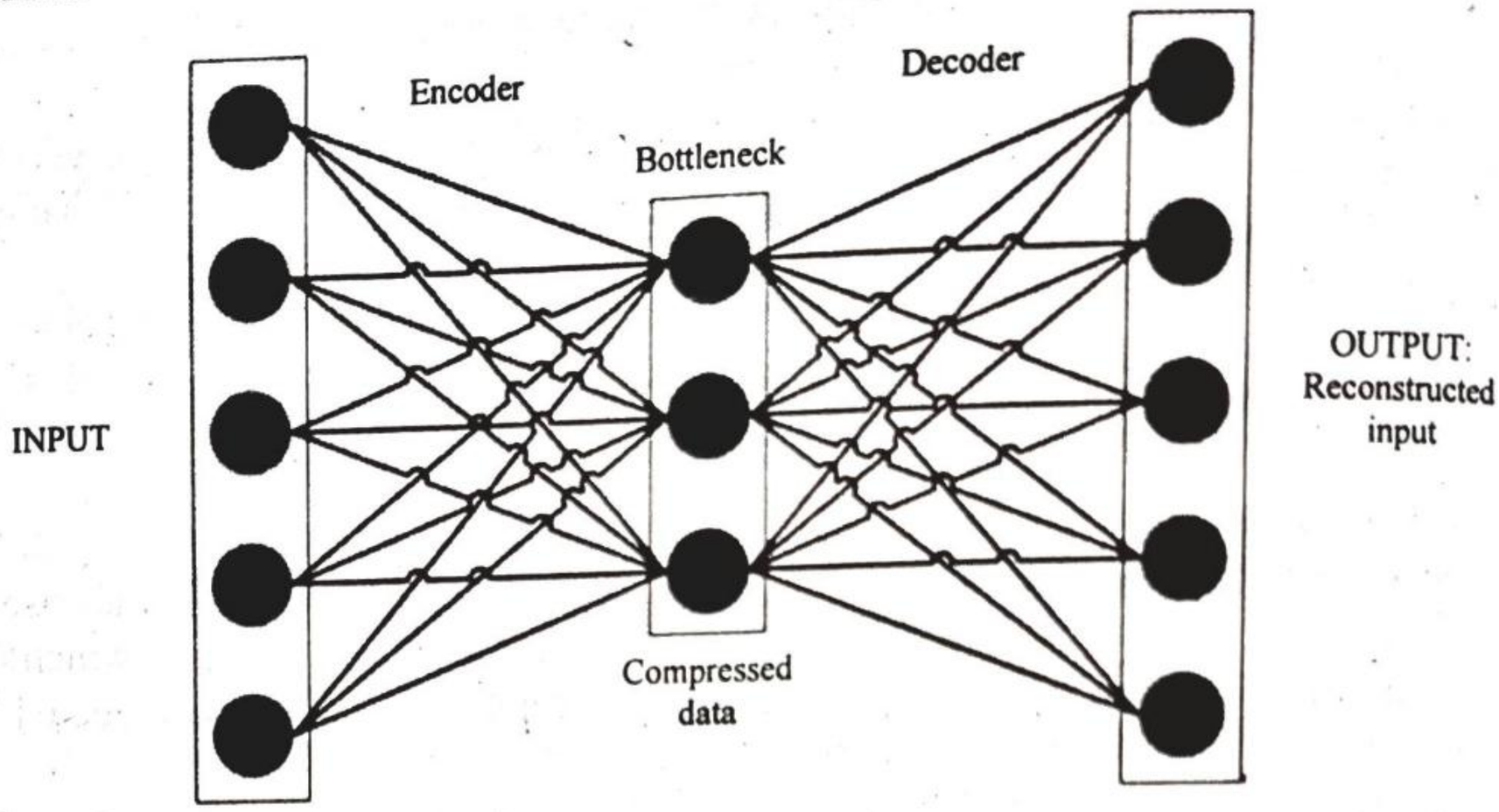
For unsupervised representation learning, an RBM can be thought of as a single-layer design. The visible variables, in particular, relate to the input data, whereas the hidden variables correspond to the feature detectors. Hinton's contrastive divergence (CD)

Scanned by TapScanner

approach can be used to train the weights by maximizing the probability of visible variables.

Autoencoders

Deep network representations have been found to be insensitive to complex noise or data conflicts. This can be linked to the architecture to some extent. The employment of convolutional layers and max-pooling, for example, can be proven to produce transformation insensitivity.



Autoencoders are therefore neural networks that may be taught to do representation learning. Autoencoders seek to duplicate their input to their output using an encoder and a decoder. Autoencoders are typically trained via recirculation, a learning process that compares the activation of the input network to the activation of the reconstructed input.

Scanned by TapScanner

SCALABLE MACHINE LEARNING

Multiple Choice Type Questions

1. The model learns and updates itself through reward/punishment in case of [MODEL QUESTION]
   a) supervised learning algorithm
   b) unsupervised learning algorithm
   c) semi-supervised learning algorithm
   d) reignforcement learning algorithm

Answer: (d)

2. Reinforcement field is studied in many other disciplines except [MODEL QUESTION]
   a) multi-agent systems
   b) control theory Learning
   c) game theory
   d) None of these

Answer: (d)

3. Learning method in which teacher returns reward and punishment to learner is [MODEL QUESTION]
   a) semi supervised learning
   b) supervised learning
   c) reinforcement learning
   d) unsupervised learning

Answer: (c)

4. Which of the following cannot be achieved by using machine learning? [MODEL QUESTION]
   a) proving casual relationships between variables
   b) forecast the outcome variable into the future
   c) accurately predict the outcome using supervised learning algorithms
   d) classify respondents into groups based on their response pattern

Answer: (a)

5. What is semi-supervised learning? [MODEL QUESTION]
   a) All data is unlabelled and the algorithms learn to inherent structure from the input data
   b) All data is labelled and the algorithms learn to predict the output from the input data
   c) It is a framework for learning where an agent interacts with an environment and receives a reward for each interaction
   d) Some data is labelled but most of it is unlabelled and a mixture of supervised and unsupervised techniques can be used

Answer: (a)

Scanned by TapScanner

Short Answer Type Questions

1. What is Scalable Machine Learning? **[MODEL QUESTION]**

Answer:

Talking about the real world where the amount of data is so large, the scalability of machine learning models becomes a primary concern for us. Most of the learners start learning data science with a very small amount of data which is small enough to fit the process and model on a single machine. But in the real world, many applications require scaling ML models to multiple machines. There can be various situations where models need to deal with large data sets like dealing with stock market data where models need to adopt new data and produce predictions quickly. And in a millisecond the predictions from the model become useless. In such a situation Scalable Machine Learning comes in the application which aims to combine statistics, systems, machine learning, and data mining in such a procedure that is flexible to any environment.

More formally we can say the word scalable means here to make a model which can deal with any amount of data without increasing the consumption of resources like memory, time, etc. When it comes to making the computation of machine learning models across the big data faster some problems can accrue due to lack of scalability like we can find the problem in the fitting of any model on large size of data and if we have a model one thing which can slow us the computation speed of the model. This can lead to consequences where we are required to sacrifice accuracy because optimization of models for large-scale data becomes infeasible and time-consuming.

In scalable machine learning, we try to build a system where the components of the system have their own work or task which helps the whole system to lead towards the solution of the problem rapidly, without wasting so much memory and increasing the performance as well. This makes us introduce shared memory models and Distributed Memory Model which are basically two types of hardware structural designs for modelling purposes.

2. What is semi-supervised learning? **[MODEL QUESTION]**

Answer:

Semi-supervised learning is an important category that lies between the Supervised and Unsupervised machine learning. Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for the corporate purpose, it may have few labels.

The basic disadvantage of supervised learning is that it requires hand-labelling by ML specialists or data scientists, and it also requires a high cost to process. Further unsupervised learning also has a limited spectrum for its applications. To overcome these drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced. In this algorithm, training data is a combination of both labelled and unlabeled data. However, labelled data exists with a very small amount while it consists of a huge amount of unlabeled data. Initially, similar data is

Scanned by TapScanner

clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labelled data. It is why label data is a comparatively, more expensive acquisition than unlabeled data.

3. Explain working of semi-supervised learning. **[MODEL QUESTION]**

Answer:

Semi-supervised learning uses pseudo labelling to train the model with less labelled training data than supervised learning. The process can combine various neural network models and training ways. The whole working of semi-supervised learning is explained in the below points:

- Firstly, it trains the model with less amount of training data similar to the supervised learning models. The training continues until the model gives accurate results.
- The algorithms use the unlabeled dataset with pseudo labels in the next step, and now the result may not be accurate.
- Now, the labels from labelled training data and pseudo labels data are linked together.
- The input data in labelled training data and unlabeled training data are also linked.
- In the end, again train the model with the new combined input as did in the first step. It will reduce errors and improve the accuracy of the model.

4. What is active learning technique? **[MODEL QUESTION]**

Answer:

Active learning is a type of semi-supervised learning in which a learning algorithm can interactively query the user to obtain the desired outputs at new data points.
Let P be the population set of all data under consideration. For example, all male students studying in a college are known to have a particularly exciting study pattern.
During each iteration, I, P (total population) is broken up into three subsets.

a) P(K, i): Data points where the label is known (K).
b) P(U, i): Data points where the label is unknown (U).
c) P(C, i): A subset of P(U,i)that is chosen(C) to be labelled.

Current research in this type of learning concentrates on identifying the best method P(C, i) to chose (C) the data points.

Heuristic for active learning

1. Start with a pool of unlabelled data P(U,i)
2. Pick a few points at random and get their labels P(C,i)
3. Repeat

Some active learning algorithms are built upon support vector machines (SVMs) to determine the data points to label. Such methods usually calculate the margin (W) of each

Scanned by TapScanner

unlabelled datum in P(U, i) and assume W as an n-dimensional space distance commencing covering datum and the splitting hyperplane.
Minimum marginal hyperplane (MMH) techniques consent that the data records with the minimum margin (W) are those that the SVM is most uncertain about and therefore should be retained in P(C, i) to be labelled. Other comparable approaches, such as Maximum Marginal Hyperplane, select data with significant W(n-dimensional space). Other approaches to select a combination of the smallest as well as largest Ws.

5. What are active learning query strategies? **[MODEL QUESTION]**

Answer:

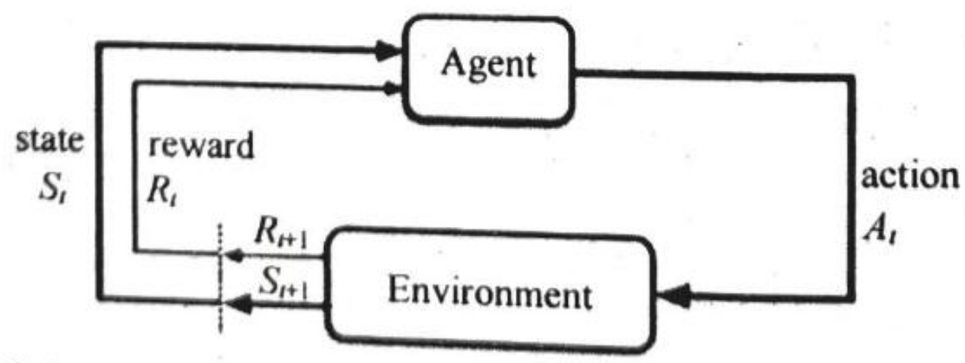
Few logics for determining which data points should be labelled are

1. **Uncertain sampling:** In this method, the active learning algorithm first tries to label the points for which the current model is least specific (as this means a lot of confusion) on the correct output.
2. **Query by committee:** A range of models are trained on the current labelled data and vote on the output for unlabelled data; label those points for which the 'committee' disagrees the most.
3. **Expected model change:** In this method, the active learning algorithm first tries to label the points that would most change the existing model itself.
4. **Expected error reduction:** In this method , the active learning algorithm first tries to label the points that would mostly reduce the model's generalization error.
5. **Variance reduction:** In this method, the active learning algorithm first tries to label the points that would reduce output variance (which is also one of the components of error).

6. What is Reinforcement Learning? **[MODEL QUESTION]**

Answer:

Reinforcement Learning is a part of machine learning. Here, agents are self-trained on reward and punishment mechanisms. It's about taking the best possible action or path to gain maximum rewards and minimum punishment through observations in a specific situation. It acts as a signal to positive and negative behaviours. Essentially an agent (or several) is built that can perceive and interpret the environment in which is placed, furthermore, it can take actions and interact with it.



Through a series of Trial and Error methods, an agent keeps learning continuously in an interactive environment from its own actions and experiences. The only goal of it is to find a suitable action model which would increase the total cumulative reward of the agent. It learns via interaction and feedback.

Scanned by TapScanner

Well, that's the definition of reinforcement learning. Now how we come to this definition, how a machine learns and how it can solve complex problems in the world through reinforcement learning, is something we are going to see further.

7. Define Bayesian learning.

Answer: **[MODEL QUESTION]**

A learning technique that determines model parameters (such as the network weights) by maximizing the posterior probability of the parameters given the training data. The idea is that some parameter values are more consistent with the observed data than others. By Bayes' rule, maximizing the posterior probability amounts to maximizing the so-called model evidence, defined as the conditional probability of the training data given the model parameters. The evidence often can be approximated by some closed formula or by an update rule. Bayesian techniques render it possible to use all data for training instead of reserving patterns for cross-validation of parameters.

Long Answer Type Questions

1. What is Bayes' Theorem?

Answer: **[MODEL QUESTION]**

Simplistically, Bayes' theorem can be expressed through the following mathematical equation

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A is an event and B is evidence. So, $P(A)$ is the prior probability of event A and $P(B)$ is evidence of event B. Hence, $P(B|A)$ is the likelihood. The denominator is a normalizing constant. So, Bayes' Theorem gives us the probability of an event based on our prior knowledge of the conditions that might be related to the event and updates that conditional probability when some new information or evidence comes up.

$$\text{Posterior Probability} = \frac{\text{Prior} * \text{Likelihood Function}}{\text{Evidence}}$$

Now let's focus on the 3 components of the Bayes' theorem

- Prior
- Likelihood
- Posterior

Prior Distribution – This is the key factor in Bayesian inference which allows us to incorporate our personal beliefs or own judgements into the decision-making process through a mathematical representation. Mathematically speaking, to express our beliefs about an unknown parameter θ we choose a distribution function called the prior distribution. This distribution is chosen before we see any data or run any experiment. How do we choose a prior? Theoretically, we define a cumulative distribution function for the unknown parameter θ. In basic context, events with the prior probability of zero

Scanned by TapScanner

will have the posterior probability of zero and events with the prior probability of one, will have the posterior probability of one. Hence, a good Bayesian framework will not assign a point estimate like 0 or 1 to any event that has already occurred or already known not to occur. A very handy widely used technique of choosing priors is using a family of distribution functions that is sufficiently flexible such that a member of the family will represent our beliefs. Now let's understand this concept a little better.

i. **Conjugate Priors** – Conjugacy occurs when the final posterior distribution belongs to the family of similar probability density functions as the prior belief but with new parameter values which have been updated to reflect new evidence/ information. Examples Beta-Binomial, Gamma -Poisson or Normal-Normal.

ii. **Non-conjugate Priors** –Now, it is also quite possible that the personal belief cannot be expressed in terms of a suitable conjugate prior and for those cases simulation tools are applied to approximate the posterior distribution. An example can be Gibbs sampler.

iii. **Un-informative prior** – Another approach is to minimize the amount of information that goes into the prior function to reduce the bias. This is an attempt to have the data have maximum influence on the posterior. These priors are known as uninformative Priors but for these cases, the results might be pretty similar to the frequentist approach.

Likelihood – Suppose θ is the unknown parameter that we are trying to estimate. Let's represent fairness of a coin with θ. Now to check the fairness, we are flipping a coin infinitely and each time it is either appearing as 'head' or 'tail' and we are assigning a 1 or 0 value accordingly. This is known as the Bernoulli Trials. Probability of all the outcomes or 'X's taking some value of x given a value of theta. We're viewing each of these outcomes as independent and hence, we can write this in product notation. This is the probability of observing the actual data that we collected (head or tail), conditioned on a value of the parameter theta (fairness of coin) and can be expressed as follows –

$$p(X \mid \theta) = (\theta)^{x} * (1-\theta)^{1-x}$$

This is the concept of likelihood which is the density function thought of as a function of theta. To maximize the likelihood i.e., to make the event most likely to occur for the data we have, we will choose the theta that will give us the largest value of the likelihood. This is referred to as the maximum likelihood estimate or MLE. Additionally, a quick reminder is that the generalization of the Bernoulli when we have N repeated and independent trials is a binomial. We will see the application later in the article.

Posterior Distribution – This is the result or output of the Bayes' Theorem. A posterior probability is the revised or updated probability of an event occurring after taking into consideration new information. We calculate the posterior probability $p(\theta \mid X)$ i.e., how probable is our hypothesis about θ given the observed evidence.

2. What is mechanism of Bayesian Inference? **[MODEL QUESTION]**

Scanned by TapScanner

Answer:

The Bayesian approach treats probability as a degree of beliefs about certain event given the available evidence. In Bayesian Learning, Theta is assumed to be a random variable. Let's understand the Bayesian inference mechanism a little better with an example.

Inference example using Frequentist vs. Bayesian approach: Suppose my friend challenged me to take part in a bet where I need to predict if a particular coin is fair or not. She told me "Well; this coin turned up 'Head' 70% of the time when I flipped it several times. Now I am giving you a chance to flip the coin 5 times and then you have to place your bet." Now I flipped the coin 5 times and Head came up twice and tail came up thrice. At first, I thought like a frequentist.

So, θ is an unknown parameter which is a representation of fairness of the coin and can be defined as

$$\theta = \{\text{fair, loaded}\}$$

Additionally, I assumed that the outcome variable X (whether head or tail) follows Binomial distribution with the following functional representation

$$(nCx)\, p^x * (1-p)^{n-x}$$

$$nCx = \frac{n!}{x!(n-x)!}$$

Now in our case $n = 5$.

Now my likelihood function will be

$$f(X \mid \theta) = \frac{5!}{X! * (5-X)!} * \left(\frac{1}{2}\right)^5, \text{ if } \theta = \text{fair}$$

$$f(X \mid \theta) = \frac{5!}{X! * (5-X)!} * (0.7)^X * (0.3)^{5-X}, \text{ if } \theta = \text{loaded}$$

Now, I saw that head came up twice, so my $X = 2$.

When $X = 2$, $f(\theta \mid X = 2) = 0.31$ if $\theta = \text{fair}$

$= 0.13$ if $\theta = \text{loaded}$

Therefore, using the frequentist approach I can conclude that maximum likelihood i.e., MLE (theta hat) = fair.

$$\text{MLE}(\hat{\theta}) = \text{fair}$$

Now comes the tricky part. If the question comes how sure am I about my prediction? I will not be able to answer that question perfectly or correctly as in a frequentist world, a coin is a physical object and hence, my probability can be either 0 or 1 i.e., the coin is either fair or not.

Here comes the role of Bayesian inference which will tell us the uncertainty in my prediction i.e., $P(\theta \mid X = 2)$. The Bayesian inference allows us to incorporate our knowledge/information about the unknown parameter θ even before looking at any data. Here, suppose I know my friend pretty well and I can say with 90 % probability that she has given me a loaded coin.

Scanned by TapScanner

Therefore, my prior $P(\text{loaded}) = 0.9$. I can now update my prior belief with data and get the posterior probability using Bayes' Theorem.

$$f(X|\theta) = \frac{f(X|\theta)f(\theta)}{\sum f(X|\theta)f(\theta)}$$

My numerator calculation will be as follows –

$$f(X|\theta) * f(\theta) = \frac{5!}{X! * (5-X)!} * \left(\frac{1}{2}\right)^5 * (0.1), \text{ when } \theta = \text{fair}$$

$$f(X|\theta) * f(\theta) = \frac{5!}{X! * (5-X)!} * (0.7)^X * (0.3)^{5-X} * (0.9), \text{ when } \theta = \text{loaded}$$

The denominator is a constant and can be calculated as the expression below. Please note that we are here basically summing up the expression over all possible values of θ which is only 2 in this case i.e., fair or loaded.

$$\sum f(X|\theta)f(\theta) = \frac{5!}{X! * (5-X)!} * \left(\frac{1}{2}\right)^5 + \frac{5!}{X! * (5-X)!} * (0.7)^X * (0.3)^{5-X}$$

Hence, after replacing X with 2 we can calculate the Bayesian probability of the coin being loaded or fair. Do it yourself and let me know your answer! However, you will realize that this conclusion contains more information to make a bet than the frequentist approach.

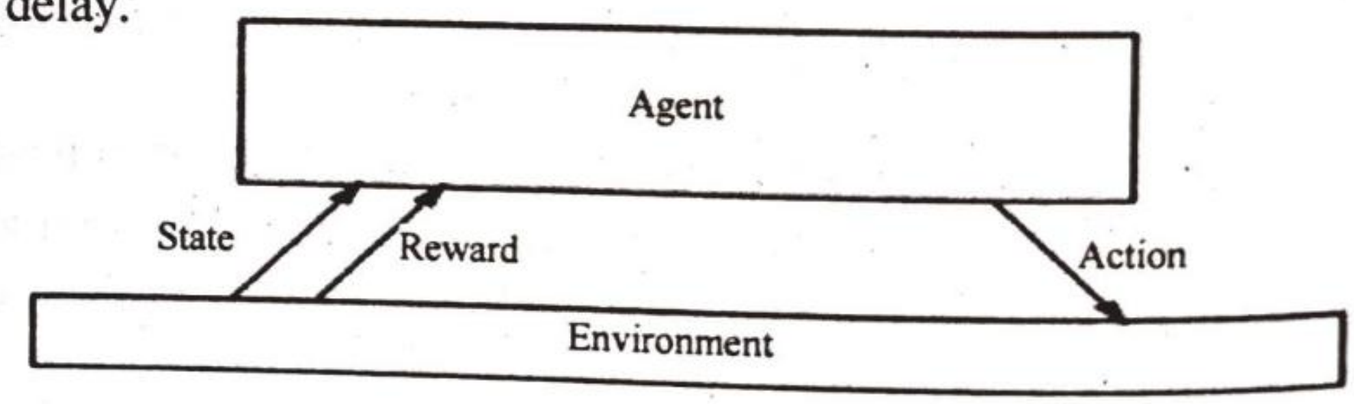
3. What are reinforcement learning problem and its characteristics?

[MODEL QUESTION]

Answer:

Reinforcement Learning Problem

- An agent interacting with its environment. The agent exists in an environment described by some set of possible states S.
- Agent perform any of a set of possible actions A. Each time it performs an action a, in some state st the agent receives a real-valued reward r, that indicates the immediate value of this state-action transition. This produces a sequence of states si, actions ai, and immediate rewards ri as shown in the figure.
- The agent's task is to learn a control policy, $\pi : S \rightarrow A$, that maximizes the expected sum of these rewards, with future rewards discounted exponentially by their delay.



$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \dots$$

Scanned by TapScanner

Goal: Learn to choose actions that maximize

$$r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots, \text{ where } 0 \le \gamma < 1$$

Reinforcement learning problem characteristics

1. Delayed reward: The task of the agent is to learn a target function π that maps from the current state s to the optimal action $a = \pi(s)$. In reinforcement learning, training information is not available in $(s, \pi(s))$. Instead, the trainer provides only a sequence of immediate reward values as the agent executes its sequence of actions. The agent, therefore, faces the problem of temporal credit assignment: determining which of the actions in its sequence are to be credited with producing the eventual rewards.

2. Exploration: In reinforcement learning, the agent influences the distribution of training examples by the action sequence it chooses. This raises the question of which experimentation strategy produces most effective learning. The learner faces a trade-off in choosing whether to favour exploration of unknown states and actions, or exploitation of states and actions that it has already learned will yield high reward.

3. Partially observable states: The agent's sensors can perceive the entire state of the environment at each time step; in many practical situations sensors provide only partial information. In such cases, the agent needs to consider its previous observations together with its current sensor data when choosing actions and the best policy may be one that chooses actions specifically to improve the observability of the environment.

4. Life-long learning: Robot requires to learn several related tasks within the same environment, using the same sensors. For example, a mobile robot may need to learn how to dock on its battery charger, how to navigate through narrow corridors, and how to pick up output from laser printers. This setting raises the possibility of using previously obtained experience or knowledge to reduce sample complexity when learning new tasks.

4. Explain graphical model. **[MODEL QUESTION]**

Answer:

The Graphical model is a subdivision of Machine Learning. It uses a graph to signify a domain problem. A graph states the conditional need structure between random variables. These are being used in many Machine Learning algorithms. For example;

- Naive Bayes' algorithm
- The Hidden Markov Model
- Restricted Boltzmann machine
- Neural Networks

Description: There are lots of causes to learn about graphical or probabilistic modelling. Among them one of that it is a charming scientific field with a beautiful theory. That bonds in astonishing ways two very different branches of mathematics:

- Probability and
- Graph theory

It has similarly interesting links to philosophy, mainly the question of causality. All at once, probabilistic modelling is broadly used all over machine learning. It also used in

Scanned by TapScanner

many real-world applications. These methods may be used to solve problems in fields as varied as drug, linguistic processing, visualization, and many others. This grouping of neat theory and influential applications. It makes graphical models one of the greatest captivating topics in modern artificial intelligence and computer science.

Representation:

- Graphical models are graphs in which nodes signify arbitrary variables.
- The lack of arcs be conditional independence assumptions.
- They deliver a compact representation of mutual probability distributions.
- For instance, an atomic representation of the joint, $P(X_1;::;X_N)$, needs $O(2N)$ parameters if we have N binary random variables.
- While a graphical model can need exponentially fewer.
- That depends on which conditional norms we make.
- This may help together inference and learning.

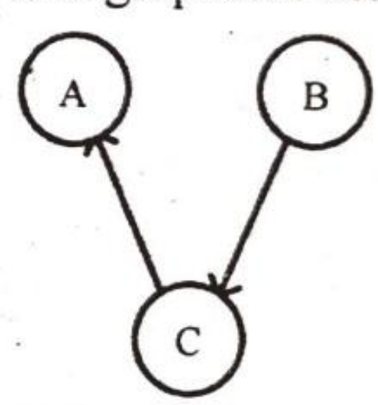
Types of Graphical Model

The two main types of graphical models are;

- Directed graphical models (Bayesian networks BNs)
- Undirected graphical models (Markov networks or Markov random elds MRFs)

Directed graphical models

- We present a graphical language for stipulating a graphical model, called the directed graphical model.
- It gives a dense model and concise way to require probabilistic models.
- It permits the reader to visually parse needs between random variables.
- A graphical model visually seizures the tactic in which the dual distribution over all random variables may be rotten into a product of factors.
- That depends simply on a subset of these variables.
- The joint distribution by itself may be somewhat complicated.
- It does not state us all about structural properties of the probabilistic model.
- For instance, the joint distribution $p(a;b;c)$ does not speak us whatever about independence relations.
- At this point graphical models come into play.
- Nodes are random variables in a graphical model.

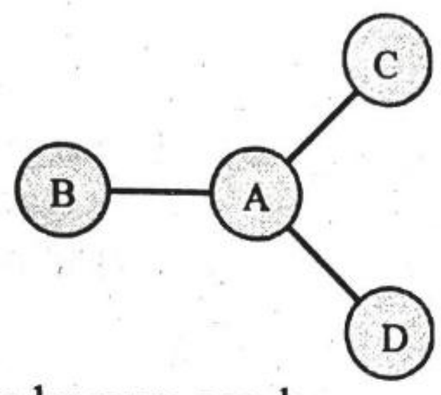


$$P(A,B,C)=P(A|C)\cdot P(C|B)\cdot P(B)$$

- In the above figure, the nodes represent the random variables $a;b;c$.

Scanned by TapScanner

- Edges denote probabilistic relations between variables.
- Not every distribution may be denoted in a specific choice of graphical model.
- They are a modest mode to imagine the structure of a probabilistic model.
- They may be used to design or inspire new kinds of statistical models.
- Check of the graph only provides us insight into properties, for example, conditional independence.
- Intricate computations for inference and learning in statistical models can be said in relations of graphical handlings.
- We may construct the matching directed graphical model from a factorized joint distribution as follows:
  1. Generate a node for all random variables.
  2. We enhance a directed link or arrow to the graph from the nodes matching to the variables on which the distribution is conditioned for each conditional distribution.



- The undirected graph made known can have one of numerous interpretations.
- The common story is that the presence of an edge involves some sort of dependence between the matching random variables.
- We might infer from this graph that B, C, D are all equally independent once A is identified.
- Consistently in this case that
- For some non-negative functions f A B, f A C, f A D

5. Explain Bayesian network. **[MODEL QUESTION]**

Answer:

- The model represents a factorization of the joint probability of all random variables if the network structure of the model is a directed acyclic graph.
- More exactly, if the events are $X_1, \ldots, X_n$ then the joint probability fulfils where, pa (X_i) is the set of parents of node X_i.
- As another way, the joint distribution factors into a product of conditional distributions.
- For instance, the directed acyclic graph displayed in the Figure this factorization will be
- Any two nodes are temporarily independent given the values of their parents.
- Any two sets of nodes are in theory independent given a third set if a criterion called d-separation clenches in the graph.
- In Bayesian networks, local and global independences are equal.
- This kind of graphical model is recognized as a directed graphical model.

Scanned by TapScanner

RECENT TRENDS

Multiple Choice Type Questions

1. Who invented machine learning _____ [MODEL QUESTION]
   a) Nelson Mandela
   b) Arthur Samuel
   c) Frank Rosenblatt
   d) Geoffrey Hinton

Answer: (b)

2. Who is know as the father of machine learning _____ [MODEL QUESTION]
   a) Ohn McCarthy
   b) Arthur Samuel
   c) Geoffrey Hinton
   d) Ronald Ross

Answer: (c)

3. Application of machine learning methods to large databases is called [MODEL QUESTION]
   a) Web development
   b) Data mining
   c) App development
   d) Blockchain

Answer: (b)

4. Computers are best at learning [MODEL QUESTION]
   a) facts
   b) concepts
   c) procedures
   d) principles

Answer: (a)

Short Answer Type Questions

1. What is an automated machine learning? [MODEL QUESTION]

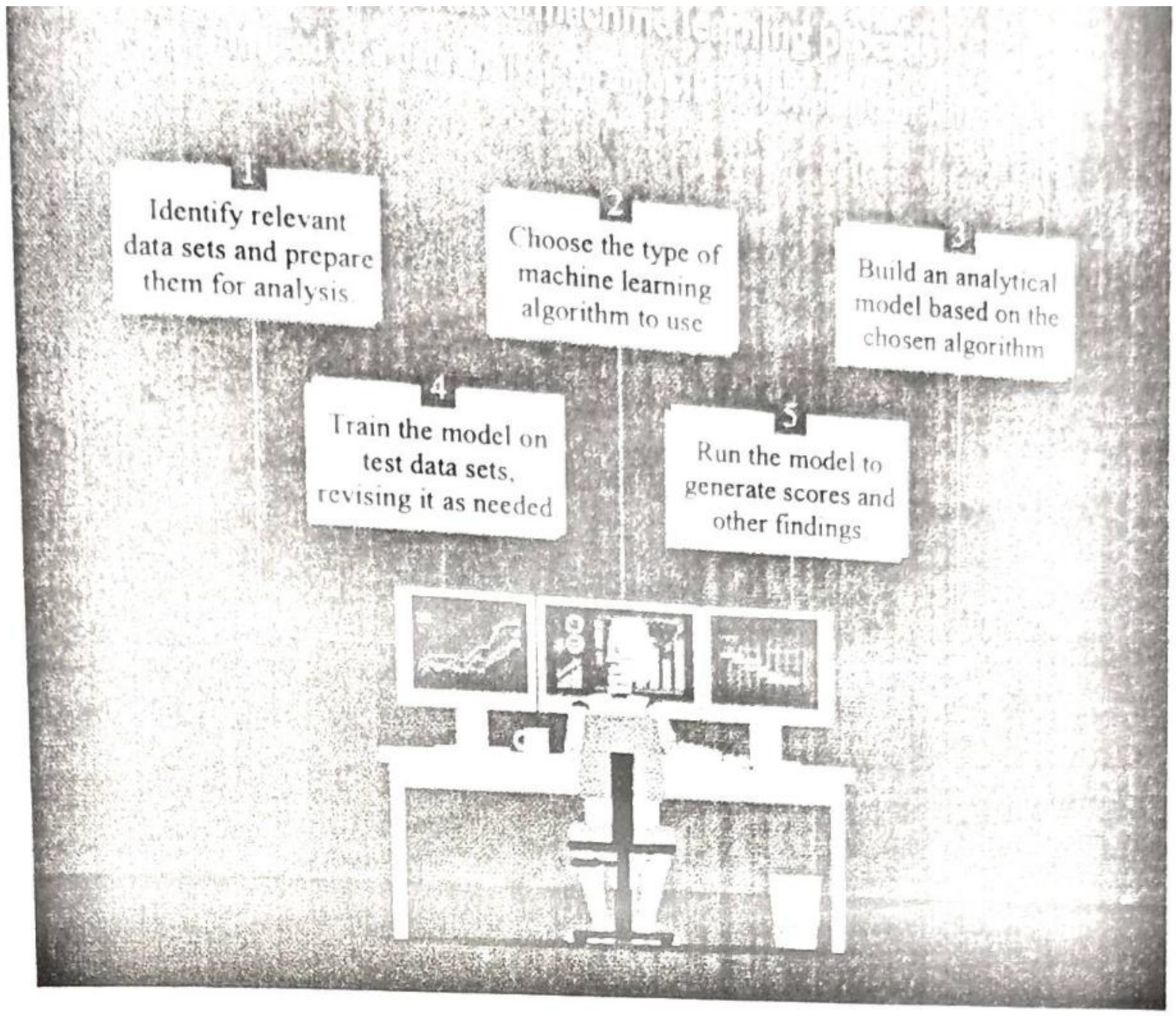
Answer:

Automated machine learning (AutoML) is the process of applying machine learning (ML) models to real-world problems using automation. More specifically, it automates the selection, composition and parameterization of machine learning models. Automating the machine learning process makes it more user-friendly and often provides faster, more accurate outputs than hand-coded algorithms.

AutoML software platforms make machine learning more user-friendly and give organizations without a specialized data scientist or machine learning expert access to machine learning.

AutoML is typically a platform or open source library that simplifies each step in the machine learning process, from handling a raw dataset to deploying a practical machine learning model. In traditional machine learning, models are developed by hand, and each step in the process must be handled separately.

Scanned by TapScanner



2. What is the contribution of machine learning in rise of AI ethics?

[MODEL QUESTION]

Answer:

Now that Artificial Intelligence and Machine Learning is on the rise, it is equally important to discuss the ethics of these technologies. It is very easy to create tech that is intelligent and has independent thinking capabilities but what happens after? What if a self-driving car kills a person? What if a Machine Learning algorithm is biased towards certain people because the data is biased? Should robots have any rights under the law or not? What will happen to people's jobs if Machine Learning becomes more widespread? These are all Ethics questions that are important and need to be discussed more in 2021.

There have already been many controversies relating to AI Ethics. The most popular one was when Amazon found out that their Machine Learning based recruiting algorithm was biased against women in 2018. This is because it was trained on data where most of the candidates were men, so the algorithm also favored men over women. Another much recent scandal was in 2020 only when Google fired Dr. Timnit Gebru, a prominent researcher on racial bias in technology who was studying the bias in Google's Artificial Intelligence systems.

[MODEL QUESTION]

3. What is AI engineering?

Answer:

Everyone has heard about software engineering, but now it's AI Engineering that on the rise as a profession! This is a very important development because the integration of Artificial Intelligence and Machine Learning in the industry has been very ad-hoc and

Scanned by TapScanner

haphazard without any regulations of best practices. That's why Gartner even predicts that only **53%** of AI and ML projects complete the journey from a prototype to full production in a company while the rest **47%** face failure.

That's where AI Engineering comes in! A streamlined AI Engineering strategy for a company provides great performance, reliability, and scalability from a machine learning algorithm which ensures a return on the investment in AI. This includes a heavy focus on DataOps, ModelOps, DevOps, etc. with the artificial intelligence projects being a part of the overall DevOps strategy in a company and not just an ad-hoc practice in some projects.

Long Answer Type Questions

1. What is future of machine learning? **[MODEL QUESTION]**

Answer:

The future of machine learning is exceptionally exciting. At present, almost every common domain is powered by machine learning applications. To name a few such realms, healthcare, search engine, digital marketing, and education are the major beneficiaries. It appears virtually impossible to work on a domain devoid of this new technology to achieve target results efficiently. Machine Learning could be contested merit to an enterprise or an organisation be it a Multi-National Company or an angel company as tasks that are presently being done manually shall be wholly accomplished by the machines in the future.

According to Gartner, the globally leading research, advisory, and consultatory institution, machine learning is remembered for pretty much every latest trends and patterns found in the literary circles, and as it should be. Machine learning is ready to change our lives in manners that were impossible just decades prior. In its rundown of the main 10 key innovation patterns, Gartner states that Computerized reasoning and new ML techniques have arrived at a basic tipping point and will progressively increase and expand for all intents and purposes each innovation empowered assistance, thing, or application. Making advanced intelligent frameworks that learn, adjust, and possibly act self-sufficiently as opposed to just execute predefined guidelines is fundamentally landmark for the innovation merchants and technology vendors.

During the post-industrialization time, individuals have attempted to make a machine that acts and does every activity just as a human. As a result, Machine Learning becomes AI's greatest blessing to the human race for the effective realisation of the targets. On the other hand, self-learned machine techniques have considerably changed the employable guidelines of the big business houses.

As of late, self-driving automatic vehicles, computerized aides, mechanical staff members, robots, and savvy urban areas have demonstrated that smart machines are conceivable and could yield enticing results. Simulated intelligence on the lines of the human mind and brain has changed most industry areas such as retail, production, construction, accounting, medical services, media, and engineering.

Scanned by TapScanner

And it is kept on occupying new regions with increasingly new vigour. The following five areas are thought out as futuristic machine learning advancements.

Accurate Results for the Search on the Web Engine:

When scrolling through Google in search of an article, one probably not be aware of it, but the ranking and the hierarchical order of those outcomes is done with a purpose. The techniques of machine learning have a tremendous impact on search engine outcomes, lately. Over the next few years, search engines will boost both the user experiences and the host experiences rapidly in fast progress. With further neural network growth and development blended with evolving deep learning techniques, the future search engines will be far better in providing responses and perceptions that are significantly germane to the searchers, explorers of the web.

Accurate Tailor-made Customisation:

Corporations could fine-tune their understanding of their target audience using machine learning to inform the enhancement of the existing products, new product development, merchandising, and gross revenue. Developers, programmers, and engineers could customize products far more precisely than ever before with algorithms to break down exactly how their products are used, maximizing value for both the organisation and the clients. With more advancements and discoveries in the dynamic field of machine learning and its algorithms, for the clients on a larger scale, we shall start to see exact targeting and fine-tuned customisation in the near future.

Surge in the Quantum Computing:

No commercially-ready quantum hardware or algorithms applications are readily accessible as of now. Nonetheless, in order to get quantum computing off the ground, several government agencies, academic institutions, and think tanks have spent millions. In the futurity of machine learning, quantum computing is set to have an enormous role. As we witness instant processing, rapidly learning, expanded capacities, and enhanced capabilities the introduction of quantum computing into machine learning would metamorphose the domain completely. This implies that in a tiny split moment, complicated issues that we may not have the capacity to tackle with conventional methods, and existing technologies may well be done so.

Mass Growth of Data Units:

It would not be unusual to be engrossed with coding, systematic activities, engineering by technology, and information units. It can be predicted that further developments in machine learning can further improve these units' everyday operations towards the efficient realization of the targets. In the coming decades, machine learning will be one of the cornerstone methods for creating, sustaining, and developing digital applications. It implies that data curators and technology engineers spend comparatively lesser time period in programming, upgrading ML techniques, so instead make them understand and continuously improve their operations.

Fully Automated Self-Learning System:

In software engineering, machine learning will be just another component. In addition to standardizing the way people implement machine learning algorithms, open-source frameworks such as Keras, PyTorch, and Tensorflow have also eliminated the basic requirements for doing just that. some of this may sound like utopia, but these types of

Scanned by TapScanner

ecosystems are slowly but steadily coming out, with so many technology, databases, and resources accessible online today. This would lead to environments that really are near or close to zero codings, and so an automated system emerges.

2. What is Internet of Things (IOT)? How IOT and Machine Learning are connected to each other? [MODEL QUESTION]

Answer:

1st Part:

The Internet of Things (IoT) has become a common news item and marketing trend. Beyond the hype, IoT has emerged as an important technology with applications in many fields. IoT has roots in several earlier technologies: pervasive information systems, sensor networks, and embedded computing. The term **IoT system** more accurately describes the use of this technology than does **Internet of Things**. Most IoT devices are connected together to form purpose-specific systems; they are less frequently used as general-access devices on a worldwide network.

IoT moves beyond pervasive computing and information systems, which concentrated on data. Smart refrigerators are one example of pervasive computing devices. Several products included built-in PCs and allowed users to enter information about the contents of their refrigerator for menu planning. Conceptual devices would automatically scan the refrigerator contents to take care of data entry. The use cases envisioned for these refrigerators are not so far removed from menu planning applications for stand-alone personal computers.

Sensor network research spanned a range of configurations. Many of these were designed for data collection at very low data rates. The connected data would then be sent to servers for processing. Traditional sensor network research did not emphasize in-network processing.

Embedded computing concentrated on either stand-alone devices or tightly coupled networks such as those used in vehicles. Consumer electronics and cyber-physical systems were two major application domains for embedded computing; both emphasized engineered systems with well-defined goals.

IoT systems are useful in a broad range of applications:

- Industrial systems use sensors to monitor both the industrial processes themselves – the quality of the product – and the state of the equipment. An increasing number of electric motors, for example, include sensors that collect data used to predict impending motor failures.
- Smart buildings use sensors to identify the locations of people as well as the state of the building. That data can be used to control heating/ventilation/air conditioning systems and lighting systems to reduce operating costs. Smart buildings and structures also use sensors to monitor structural health.
- Smart cities use sensors to monitor pedestrian and vehicular traffic and may integrate data from smart buildings.
- Vehicles use networked sensors to monitor the state of the vehicle and provide improved dynamics, reduced fuel consumption, and lower emissions.

Scanned by TapScanner

- Medical systems connect a wide range of patient monitoring sensors that may be located at the home, in emergency vehicles, the doctor's office, or the hospital.

2nd Part:

1. IoT sensors are added to the machinery that looks at discrete variables such as vibration, noise, heat, and temperature. This data is then uploaded to the cloud for analytics.
2. Now ML comes into the picture, the machine learning model sits on the cloud platform feeding on incoming data.
3. The ML model splits the information into data used for training and for verification.
4. The model looks at hundreds of thousands of records for anomalies, correlations and projections, to come up with a hypothesis.
5. Once the hypothesis has been created, it needs to be tested and validated.
6. Once a model has been validated, it's published as an executable endpoint. Next, the live streaming data can be passed through the trained model and make an inference about the status/health of the machinery based upon what it already knows and has been trained to look for.

3. How machine learning is applied to cyber security? [MODEL QUESTION]

Answer:

To better understand previous cyber-attacks, and develop respective defence responses, ML can be leveraged in various domains within Cyber Security to enhance security processes, and make it easier for security analysts to quickly identify, prioritise, deal with and remediate new attacks.

The following headings are just a few methods Machine Learning can be used to aid security.

Automating Tasks

A great benefit of ML in cyber security is its capacity to automate repetitive and time-consuming tasks, such as triaging intelligence, malware analysis, network log analysis and vulnerability assessments. By incorporating ML into the security workflow, organisations can accomplish tasks faster, and act on and remediate threats at a rate that would not be possible with manual human capability alone. Automating repetitive processes means that clients can up or down scale easily, without changing the manpower needed, thus reducing costs in the process.

The method of automating practices via ML is sometimes referred to as AutoML. AutoML signifies when repetitive tasks involved in development are automated to specifically aid the productivity of the analysts, data scientists and developers.

Threat Detection and Classification

Machine learning algorithms are used in applications to detect and respond to attacks. This can be achieved by analysing big data sets of security events and identifying patterns of malicious activities. ML works so that when similar events are detected, they are automatically dealt with by the trained ML model.

For instance, the dataset to feed a machine learning model can be created by using Indicators of Compromise (IOCs). These can help monitor, identify, and respond to

Scanned by TapScanner

threats in real time. ML classification algorithms can be used using IOC data sets to classify the behaviour of malwares.
An example of such a use is evident in a report from Darktrace, an ML based Enterprise Immune Solution, that claims to have prevented attacks during the WannaCry ransomware crisis. According to David Palmer, Director of Technology at Darktrace, "Our algorithms spotted the attack within seconds in one NHS agency's network, and the threat was mitigated without causing any damage to that organization," he said of the ransomware, which infected more than 200,000 victims across 150 countries.

Phishing

Traditional phishing detection techniques alone lacks the speed and accuracy to detect and differentiate between harmless and malicious URLs. Latest ML algorithm predictive URL classification models can identify patterns that reveal malicious emails. To do this, the models are trained on features such as email headers, body-data, punctuation patterns, and more to classify and differentiate the malicious from the harmless.

WebShell

WebShell is a piece of code that is maliciously loaded into a website to provide access to make modifications on the web root directory of the server. This allows attackers to gain access of the database. Which, in turn, enables the bad actor to collect personal information. By using ML, a normal shopping cart behaviour can be detected, and the model can be trained to differentiate between normal and malicious behaviour.
The same goes for User Behaviour Analytics (UBA), which forms a supplementary layer to standard security measures, to provide complete visibility, detect account compromises, and mitigate and detect malicious or anomalous insider activity. By using ML algorithms, patterns of user behaviour are categorised, to understand what constitutes normal behaviour, and to detect abnormal activity. If an unusual action is made on a device on a given network, such as an employee login late at night, inconsistent remote access, or an unusually high number of downloads, the action and user is given a risk score based on their activity, patterns and time.

Network Risk Scoring

Use of quantitative measures to assign risk scores to sections of networks, help organisations to prioritise resources . ML can be used to analyse previous cyber-attack datasets and determine which areas of networks were mostly involved in particular attacks. This score can help quantify the likelihood, and impact of an attack, with respect to a given network area. Thus, helping organisations to reduce the risk of being victimized by further attacks.
When you are doing business profiling, you have to decipher what area, if compromised, is going to destroy your business. It could be a Customer Relationship Management (CRM) system, your accounting system, or your sales system. It's about knowing, within your specific business environment, what area is most vulnerable. Say, for instance, HR goes down, this may have a low risk score within your company. But if your oil trading platform goes down, that could bring down your entire business. Every company has a different way of doing security. And once you understand the specifics of an organisation, you know what to really protect. And if there is a hack, you know what to prioritise.

Scanned by TapScanner