1. What is the difference between Machine Learning and Deep Learning?

Machine Learning forms a subset of Artificial Intelligence, where we use statistics and algorithms to train machines with data, thereby helping them improve with experience.

Deep Learning is a part of Machine Learning, which involves mimicking the human brain in terms of structures called neurons, thereby forming neural networks.

2. What is a perceptron?

A perceptron is similar to the actual neuron in the human brain. It receives inputs from various entities and applies functions to these inputs, which transform them to be the output.

A perceptron is mainly used to perform binary classification where it sees an input, computes functions based on the weights of the input, and outputs the required transformation.

3. How is Deep Learning better than Machine Learning?

Machine Learning is powerful in a way that it is sufficient to solve most of the problems. However, **Deep Learning** gets an upper hand when it comes to working with data that has a large number of dimensions. With data that is large in size, a Deep Learning model can easily work with it as it is built to handle this.

4. What are some of the most used applications of Deep Learning?

Deep Learning is used in a variety of fields today. The most used ones are as follows:

- Sentiment Analysis
- Computer Vision
- Automatic Text Generation
- Object Detection
- Natural Language Processing
- Image Recognition

5. What is the meaning of overfitting?

Overfitting is a very common issue when working with Deep Learning. It is a scenario where the Deep Learning algorithm vigorously hunts through the data to obtain some valid information. This makes the Deep Learning model pick up noise rather than useful data, causing very high variance and low bias. This makes the model less accurate, and this is an undesirable effect that can be prevented.

6. What are activation functions?

Activation functions are entities in Deep Learning that are used to translate inputs into a usable output parameter. It is a function that decides if a neuron needs activation or not by calculating the weighted sum on it with the bias.

Using an activation function makes the model output to be non-linear. There are many types of activation functions:

- ReLU
- Softmax
- Sigmoid
- Linear
- Tanh

can take in real-time array data and process it quickly. This ensures that high efficiency is maintained and also makes the model more open to processing a variety of signals.

8. What are the steps involved in training a perception in Deep Learning?

There are five main steps that determine the learning of a perceptron:

1. Initialize thresholds and weights
2. Provide inputs
3. Calculate outputs
4. Update weights in each step
5. Repeat steps 2 to 4

9. What is the use of the loss function?

The loss function is used as a measure of accuracy to see if a neural network has learned accurately from the training data or not. This is done by comparing the training dataset to the testing dataset. The loss function is a primary measure of the performance of the neural network. In Deep Learning, a good performing network will have a low loss function at all times when training.

10. What are some of the Deep Learning frameworks or tools that you have used?

This question is quite common in a Deep Learning interview. Make sure to answer based on the experience you have with the tools.

However, some of the top Deep Learning frameworks out there today are:

- TensorFlow
- Keras
- PyTorch
- Caffe2
- CNTK
- MXNet
- Theano

11. What is the use of the swish function?

The swish function is a self-gated activation function developed by Google. It is now a popular activation function used by many as Google claims that it outperforms all of the other activation functions in terms of computational efficiency.

12. What are autoencoders?

Autoencoders are artificial neural networks that learn without any supervision. Here, these networks have the ability to automatically learn by mapping the inputs to the corresponding outputs.

Autoencoders, as the name suggests, consist of two entities:

- Encoder: Used to fit the input into an internal computation state
- Decoder: Used to convert the computational state back into the output

13. What are the steps to be followed to use the gradient descent algorithm?

There are five main steps that are used to initialize and use the gradient descent algorithm:

- Initialize biases and weights for the network
- Send input data through the network (the input layer)
- Calculate the difference (the error) between expected and predicted values
- Change values in neurons to minimize the loss function
- Multiple iterations to determine the best weights for efficient working

14. Differentiate between a single-layer perceptron and a multi-layer perceptron.

Single-layer Perceptron	Multi-layer Perceptron
Cannot classify non-linear data points	Can classify non-linear data
Takes in a limited amount of parameters	Withstands a lot of parameters
Less efficient with large data	Highly efficient with large datasets

-

15. What is data normalization in Deep Learning?

Data normalization is a preprocessing step that is used to refit the data into a specific range. This ensures that the network can learn effectively as it has better convergence when performing backpropagation.

16. What is forward propagation?

Forward propagation is the scenario where inputs are passed to the hidden layer with weights. In every single hidden layer, the output of the activation function is calculated until the next layer can be processed. It is called forward propagation as the process begins from the input layer and moves toward the final output layer.

17. What is backpropagation?

Backprobation is used to minimize the cost function by first seeing how the value changes when weights and biases are tweaked in the neural network. This change is easily calculated by understanding the gradient at every hidden layer. It is called backpropagation as the process begins from the output layer, moving backward to the input layers.

18. What are hyperparameters in Deep Learning?

Hyperparameters are variables used to determine the structure of a neural network. They are also used to understand parameters, such as the learning rate and the number of hidden layers and more, present in the neural network.

19. How can hyperparameters be trained in neural networks?

Hyperparameters can be trained using four components as shown below:

- Batch size: This is used to denote the size of the input chunk. Batch sizes can be varied and cut into sub-batches based on the requirement.
- Epochs: An epoch denotes the number of times the training data is visible to the neural network so that it can train. Since the process is iterative, the number of epochs will vary based on the data.
- Momentum: Momentum is used to understand the next consecutive steps that occur with the current data being executed at hand. It is used to avoid oscillations when training.
- Learning rate: Learning rate is used as a parameter to denote the time required for the network to update the parameters and learn.

20. What is the meaning of dropout in Deep Learning?

Dropout is a technique that is used to avoid overfitting a model in Deep Learning. If the dropout value is too low, then it will have minimal effect on learning. If it is too high, then the model can under-learn, thereby causing lower efficiency.

21. What are tensors?

Tensors are multidimensional arrays in Deep Learning that are used to represent data. They represent the data with higher dimensions. Due to the high-level nature of the programming languages, the syntax of tensors are easily understood and broadly used.

22. What is the meaning of model capacity in Deep Learning?

In Deep Learning, model capacity refers to the capacity of the model to take in a variety of mapping functions. Higher model capacity means a large amount of information can be stored in the network.

We will check out neural network interview questions alongside as it is also a vital part of Deep Learning.

23. What is a Boltzmann machine?

A Boltzmann machine is a type of recurrent neural network that uses binary decisions, alongside biases, to function. These neural networks can be hooked up together to create deep belief networks, which are very sophisticated and used to solve the most complex problems out there.

24. What are some of the advantages of using TensorFlow?

TensorFlow has numerous advantages, and some of them are as follows:

- High amount of flexibility and platform independence
- Trains using CPU and GPU
- Supports auto differentiation and its features
- Handles threads and asynchronous computation easily
- Open-source
- Has a large community

25. What is a computational graph in Deep Learning?

A computation graph is a series of operations that are performed to take in inputs and arrange them as nodes in a graph structure. It can be considered as a way of implementing mathematical calculations into a graph. This helps in parallel processing and provides high performance in terms of computational capability.

26. What is a CNN?

CNNs are convolutional neural networks that are used to perform analysis on images and visuals. These classes of neural networks can input a multi-channel image and work on it easily.

These Deep Learning questions must be answered in a concise way. So make sure to understand them and revisit them if necessary.

27. What are the various layers present in a CNN?

There are four main layers that form a convolutional neural network:

- Convolution: These are layers consisting of entities called filters that are used as parameters to train the network.
- ReLu: It is used as the activation function and used always with the convolution layer.
- Pooling: Pooling is the concept of shrinking the complex data entities that form after convolution and is primarily used to maintain the size of an image after shrinkage.
- Connectedness: This is used to ensure that all of the layers in the neural network are fully connected and activation can be computed using the bias easily.

28. What is an RNN in Deep Learning?

RNNs stand for recurrent neural networks, which form to be a popular type of artificial neural network. They are used to process sequences of data, text, genomes, handwriting, and more. RNNs make use of backpropagation for the training requirements.

29. What is a vanishing gradient when using RNNs?

Vanishing gradient is a scenario that occurs when we use RNNs. Since RNNs make use of backpropagation, gradients at every step of the way will tend to get smaller as the network traverses through backward iterations. This equates to the model learning very slowly, thereby causing efficiency problems in the network.

30. What is exploding gradient descent in Deep Learning?

Exploding gradients are an issue causing a scenario that clumps up the gradients. This creates a large number of updates of the weights in the model when training.

The working of gradient descent is based on the condition that the updates are small and controlled. Controlling the updates will directly affect the efficiency of the model.

31. What is the use of LSTM?

LSTM stands for long short-term memory. It is a type of RNN that is used to sequence a string of data. It consists of feedback chains that give it the ability to perform like a general-purpose computational entity.

32. Where are autoencoders used?

Autoencoders have a wide variety of usage in the real world. The following are some of the popular ones:

- Adding color to black–white images
- Removing noise from images
- Dimensionality reduction
- Feature removal and variation

33. What are the types of autoencoders?

There are four main types of autoencoders:

- Deep autoencoders
- Convolutional autoencoders
- Sparse autoencoders
- Contractive autoencoders

34. What is a Restricted Boltzmann Machine?

A Restricted Boltzmann Machine, or RBM for short, is an undirected graphical model that is popularly used in Deep Learning today. It is an algorithm that is used to perform:

- Dimensionality reduction
- Regression
- Classification
- Collaborative filtering
- Topic modeling

35. What are some of the limitations of Deep Learning?

There are a few disadvantages of Deep Learning as mentioned below:

- Networks in Deep Learning require a huge amount of data to train well.
- Deep Learning concepts can be complex to implement sometimes.
- Achieving a high amount of model efficiency is difficult in many cases.

These are some of the vital advanced deep learning interview questions that you have to know about!

36. What are the variants of gradient descent?

There are three variants of gradient descent as shown below:

- Stochastic gradient descent: A single training example is used for the calculation of gradient and for updating parameters.
- Batch gradient descent: Gradient is calculated for the entire dataset, and parameters are updated at every iteration.
- Mini-batch gradient descent: Samples are broken down into smaller-sized batches and then worked on as in the case of stochastic gradient descent.

37. Why is mini-batch gradient descent so popular?

Mini-batch gradient descent is popular as:

- It is more efficient when compared to stochastic gradient descent.
- Generalization is done by finding the flat minima.
- It helps avoid the local minima by allowing the approximation of the gradient for the entire dataset.

38. What are deep autoencoders?

Deep autoencoders are an extension of the regular autoencoders. Here, the first layer is responsible for the first-order function execution of the input. The second layer will take care of the second-order functions, and it goes on.

Usually, a deep autoencoder is a combination of two or more symmetrical deep-belief networks where:

- The first five shallow layers consist of the encoding part
- The other layers take care of the decoding part

39. Why is the Leaky ReLU function used in Deep Learning?

Leaky ReLU, also called LReL, is used to manage a function to allow the passing of small-sized negative values if the input value to the network is less than zero.

40. What are some of the examples of supervised learning algorithms in Deep Learning?

There are three main supervised learning algorithms in Deep Learning:

- Artificial neural networks
- Convolutional neural networks
- Recurrent neural networks

41. What are some of the examples of unsupervised learning algorithms in Deep Learning?

There are three main unsupervised learning algorithms in Deep Learning:

- Autoencoders
- Boltzmann machines
- Self-organizing maps

Next up, let us look at more neural network interview questions that will help you ace the interviews.

42. Can we initialize the weights of a network to start from zero?

Yes, it is possible to begin with zero initialization. However, it is not recommended to use because setting up the weights to zero initially will cause all of the neurons to produce the same output and the same gradients when performing backpropagation. This means that the network will not have the ability to learn at all due to the absence of asymmetry between each of the neurons.

43. What is the meaning of valid padding and same padding in CNN?

- Valid padding: It is used when there is no requirement for padding. The output matrix will have the dimensions (n – f + 1) X (n – f + 1) after convolution.
- Same padding: Here, padding elements are added all around the output matrix. It will have the same dimensions as the input matrix.

44. What are some of the applications of transfer learning in Deep Learning?

Transfer learning is a scenario where a large model is trained on a dataset with a large amount of data and this model is used on simpler datasets, thereby resulting in extremely efficient and accurate neural networks.

The popular examples of transfer learning are in the case of:

- BERT
- ResNet
- GPT-2
- VGG-16

45. How is the transformer architecture better than RNNs in Deep Learning?

With the use of sequential processing, programmers were up against:

- The usage of high processing power
- The difficulty of parallel execution

This caused the rise of the transformer architecture. Here, there is a mechanism called attention mechanism, which is used to map all of the dependencies between sentences, thereby making huge progress in the case of NLP models.

46. What are the steps involved in the working of an LSTM network?

There are three main steps involved in the working of an LSTM network:

- The network picks up the information that it has to remember and identifies what to forget.
- Cell state values are updated based on Step 1.
- The network calculates and analyzes which part of the current state should make it to the output.

47. What are the elements in TensorFlow that are programmable?

In TensorFlow, users can program three elements:

- Constants
- Variables
- Placeholders

48. What is the meaning of bagging and boosting in Deep Learning?

Bagging is the concept of splitting a dataset and randomly placing it into bags for training the model.

Boosting is the scenario where incorrect data points are used to force the model to produce the wrong output. This is used to retrain the model and increase accuracy.

49. What are generative adversarial networks (GANs)?

Generative adversarial networks are used to achieve generative modeling in Deep Learning. It is an unsupervised task that involves the discovery of patterns in the input data to generate the output.

The generator is used to generate new examples, while the discriminator is used to classify the examples generated by the generator.

50. Why are generative adversarial networks (GANs) so popular?

Generative adversarial networks are used for a variety of purposes. In the case of working with images, they have a high amount of traction and efficient working.

- Creation of art: GANs are used to create artistic images, sketches, and paintings.
- Image enhancement: They are used to greatly enhance the resolution of the input images.
- Image translation: They are also used to change certain aspects, such as day to night and summer to winter, in images easily.

1. What is the difference between Strong Artificial Intelligence and Weak Artificial Intelligence?

Weak AI	Strong AI
Narrow application, with very limited scope	Widely applied, with vast scope
Good at specific tasks	Incredible human-level intelligence
Uses supervised and unsupervised learning to process data	Uses clustering and association to process data
E.g., Siri, Alexa, etc.	E.g., Advanced Robotics

2. What is Artificial Intelligence?

Artificial Intelligence is a field of computer science wherein the cognitive functions of the human brain are studied and tried to be replicated on a machine/system. Artificial Intelligence is today widely used for various applications like computer vision, speech recognition, decision-making, perception, reasoning, cognitive capabilities, and so on.

3. List some applications of AI.

- Natural language processing
- Chatbots
- Sentiment analysis
- Sales prediction
- Self-driving cars
- Facial expression recognition
- Image tagging

4. List the programming languages used in AI.

- Python
- R
- Lisp
- Prolog
- Java

Wanna become an Expert in AI? Come & join our Artificial Intelligence Course now.

5. What are the Examples of AI in real life?

- **Robo-readers for Grading:**

Many schools, colleges, and institutions are now using AI applications to grade essay questions and assignments on Massive Open Online Courses(MOOCs). In the era of technology, where education is rapidly shifting towards online learning, MOOCs have become a new norm of education. Thousands of assignments are and essay questions are submitted on these platforms on the daily basis, and grading them by hand is next to impossible.

Robo-readers are used to grade essay questions and assignments based on certain parameters acquired from huge data sets. Thousands of hand-scored essays were fed into the deep Neural Networks of these AI systems to pick up the features of good writing assignments. So, the AI system uses previous results to evaluate the present data.

- **Online Recommendation Systems**

Online recommendations Systems study customer behavior by analyzing their keywords, websites, and the content they watch on the internet. From e-commerce to Social Media websites, everyone is using these recommendation systems to provide a better customer experience.

There are two ways to produce a recommendation list for a customer, collaborative and content-based filtering. In collaborative filtering, the system analyzes the past decisions made by the customer and suggests items that he/she might find interesting. Whereas, content-based filtering finds discrete characteristics of the product or service and suggests similar products and deals that might excite the user. The same process goes for social media apps and other websites.

- **Navigation and Travel**

Google Maps, GPS, and Autopilot on Airplanes are some of the best examples of AI in Navigation and travel. Machine Learning algorithms like Dijkstra's algorithm are used to find the shortest possible route between two points on the map. However, certain factors are also taken into account including traffic and road blockage to find an optimal route.

- **Fraud Detection**

Machine Learning models process large amounts of banking data and check if there is any suspicious activity or anomalies in the customer transactions. AI applications proved to be more effective than humans in recognizing fraud patterns as they were trained with historical data with millions of transactions.

- **Autonomous Vehicles**

Human error is responsible for more than 90% of accidents happening on the road every year. A technical failure in a vehicle, Roads and other factors has little contribution to fatal accidents. Autonomous vehicles can reduce these fatal accidents by 90%. Although Self-driving systems require a person to supervise the action and take control of the vehicle in case of emergency, they prove to be very effective when driving on an open highway or parking the vehicle. Also, advancement in technology will further improve the ability to drive in complex situations using high-end AI models and sensors like LIDAR.

6. What is ANN?

Artificial Neural Network(ANN) is a computational model based on the structure of the Biological Neural Network(BNN). The human brain has billions of neurons that collect, process the information, and drive meaning results out of it. The neurons use electro-chemical signals to communicate and pass the information to other neurons. Similarly, ANN consists of artificial neurons called nodes connected with other nodes forming a complex relationship between the output and the input.

There are three layers in the Artificial Neural Network:

- **Input Layer:** The input layer has neurons that take the input from external sources like files, data sets, images, videos, and sensors. This part of the Neural Network doesn't perform any computation. It only transfers the data from the outside world to the Neural Network

- Hidden Layer: The hidden layer receives the data from the input layer and uses it to derive results and train several Machine Learning models. The layer can be further divided into sub-layers that extract features, make decisions, connect with other sources, and predict future actions based on the events that happened.

- Output layer: After processing, the data is transferred to the output layer for delivering it to the outside environment.

7. Difference between AI, ML, and DL?

Although Machine Learning, Artificial Intelligence, and Deep learning are closely related, there are some key differences between them. Artificial Intelligence as an umbrella covers everything related to making a machine think and act like a human. Machine Learning and Deep Learning are subsets of AI and are used to achieve the goals of AI.

Below is the difference between AI, ML, and DL:

- **Artificial Intelligence:** AI consists of the algorithms and techniques that enable a machine to perform the tasks commonly associated with human intelligence. The AI applications are trained to process large amounts of complex information and right decisions without human intervention. Some of the popular examples of AI applications are chatbots, Autonomous Vehicles, Space rovers, and Simulators for mathematical and scientific purposes.
- **Machine Learning:** Machine Learning is a subset of Artificial Intelligence and is mainly used to improve computer programs through experience and training on different models. There are three main methods of Machine Learning:

  - **Supervised Learning**: In supervised learning, the machine gets the input for twitch the output is already known. After the processing is completed, the algorithm compared the output produced from the original output and measure the degree of errors in it.

  - **Unsupervised Learning:** Here, the instructor has no output or historical labels for the input data. So, the algorithm is expected to figure out the right path and extract the features from the given dataset. The goal is to allow the algorithm to search the data and s some structure in it.

  - **Reinforcement Learning:** In this method of learning there are three components, the agent, environment, and actions. An agent is a decision-maker whose goal is to choose the right actions and maximize the expected reward within a set timeframe. Reinforcement learning is mainly used in robotics where the machine learns about the environment through trial and error.
- **Deep Learning:** In Machine Learning, where the model tends to surrender to environmental changes, Deep Learning adapts to the changes by updating the models based on constant feedback. It's facilitated by the Artificial Neural Networks that mimic the cognitive behaviour of the human brain.

8. How to choose an algorithm for a problem?

To solve a problem, there can be multiple Machine Learning algorithms with different approaches and constraints. However, a generic approach can be applied to most of the problems and find a suitable algorithm. Below are the steps you need to consider while choosing an algorithm:

- **Categorize the Problem**

The first is finding the algorithm, which is to categorize it based on the type of input you have and the output you want from it. If the data is labeled, it's a problem for supervised learning. If the data is not labeled, then it's an unsupervised learning problem. At last, if the problem aims to optimize a model, then it's a reinforcement learning problem.

Similarly, you can categorize a problem based on the outcome you want from the algorithm. If the output is expected to be numerical then it's a regression problem. Is class is the output of a model, it's a classification problem, and grouping of the input values can be categorized as the clustering problems.

- **Understanding the Data**

Data plays an important role in the process of selecting the right algorithm for your problem. This is because, some algorithms can process tons of data, while some works better with smaller samples. Analyzing and transforming your data will also help you to know the constraints and the challenges you t=want to overcome while solving the problem.

- **Find the available Algorithms**

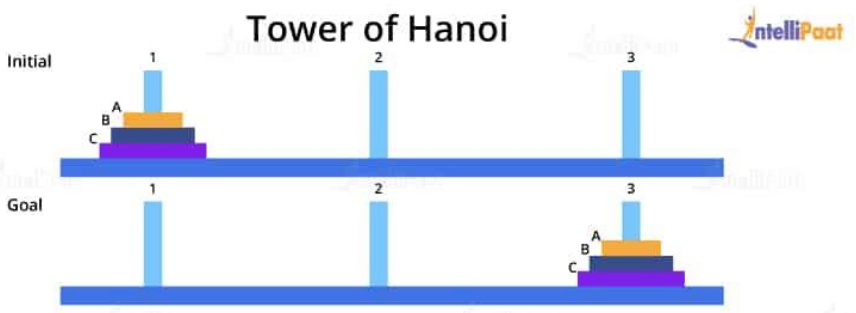
Identify the available algorithms you can apply for solving the problem in a reasonable timeframe. Some of the factors that may affect your choice of the right algorithm include the accuracy of the algorithm, complexity, scalability interpretability, build & training time, space, and the time it takes to solve the problem.

- **Implement the Algorithm**

After selecting the algorithm, you have to make an evaluation criteria by carefully selecting the testing values and subgroups of the datasets. Also, check the time taken by each algorithm to solve the problem. The algorithm that provides accurate results in the given time while acquiring less space, would be the best algorithm for your problem.

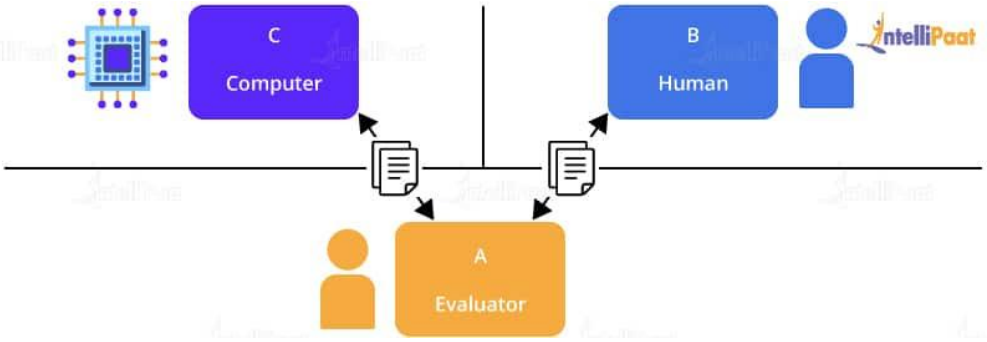
9. What is Tower of Hanoi?

Tower of Hanoi is a mathematical puzzle that shows how recursion might be utilized as a device in building up an algorithm to take care of a specific problem. Using a decision tree and a breadth-first search (BFS) algorithm in AI, we can solve the Tower of Hanoi.



10. What is Turing test?

The Turing test is a method to test a machine's ability to match the human-level intelligence. A machine is used to challenge human intelligence, and when it passes the test it is considered intelligent. Yet a machine could be viewed as intelligent without sufficiently knowing how to mimic a human.



11. What is an expert system? What are the characteristics of an expert system?

An expert system is an Artificial Intelligence program that has expert-level knowledge about a specific area and how to utilize its information to react appropriately. These systems have the expertise to substitute a human expert. Their characteristics include:

- High performance
- Adequate response time
- Reliability
- Understandability
-

12. List the advantages of an expert system.

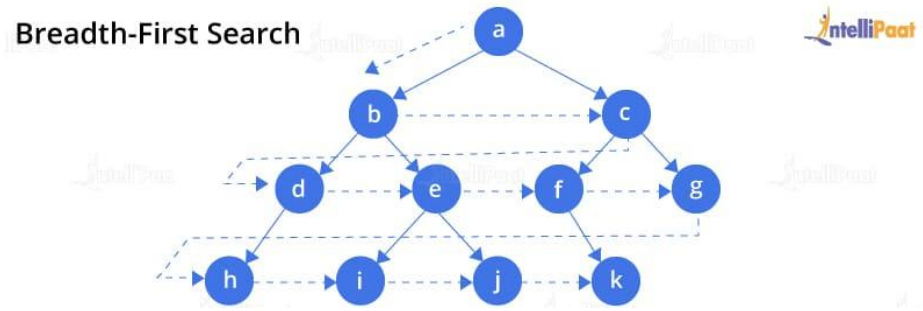
- Consistency
- Memory
- Diligence
- Logic
- Multiple expertise
- Ability to reason
- Fast response
- Unbiased in nature

13. What is an A* algorithm search method?

A* is a computer algorithm that is extensively used for the purpose of finding the path or traversing a graph in order to find the most optimal route between various points called the nodes.

14. What is a breadth-first search algorithm?

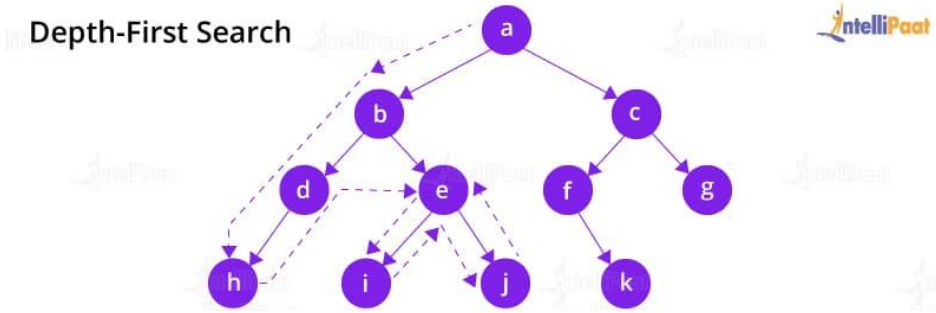
A breadth-first search (BFS) algorithm, used for searching tree or graph data structures, starts from the root node, then proceeds through neighboring nodes, and further moves toward the next level of nodes.



Till the arrangement is found, it produces one tree at any given moment. As this pursuit can be executed utilizing the FIFO (first-in, first-out) data structure, this strategy gives the shortest path to the solution.

15. What is a depth-first search algorithm?

Depth-first search (DFS) is based on LIFO (last-in, first-out). A recursion is implemented with LIFO stack data structure. Thus, the nodes are in a different order than in BFS. The path is stored in each iteration from root to leaf nodes in a linear fashion with space requirement.



Learn more about Artificial Intelligence from this Artificial Intelligence Training in New York to get ahead in your career!

16. What is a bidirectional search algorithm?

In a bidirectional search algorithm, the search begins in forward from the beginning state and in reverse from the objective state. The searches meet to identify a common state. The initial state is linked with the objective state in a reverse way. Each search is done just up to half of the aggregate way.

17. What is an iterative deepening depth-first search algorithm?

The repetitive search processes of level 1 and level 2 happen in this search. The search processes continue until the solution is found. Nodes are generated until a single goal node is created. Stack of nodes is saved.

18. What is a uniform cost search algorithm?

The uniform cost search performs sorting in increasing the cost of the path to a node. It expands the least cost node. It is identical to BFS if each iteration has the same cost. It investigates ways in the expanding order of cost.

19. How are game theory and AI related?

AI system uses game theory for enhancement; it requires more than one participant which narrows the field quite a bit. The two fundamental roles are as follows:

- Participant design: Game theory is used to enhance the decision of a participant to get maximum utility.
- Mechanism design: Inverse game theory designs a game for a group of intelligent participants, e.g., auctions.

20. Explain Alpha–Beta pruning.

Alpha–Beta pruning is a search algorithm that tries to reduce the number of nodes that are searched by the minimax algorithm in the search tree. It can be applied to 'n' depths and can prune the entire subtrees and leaves.

21. What is a fuzzy logic?

Fuzzy logic is a subset of AI; it is a way of encoding human learning for artificial processing. It is a form of many-valued logic. It is represented as IF-THEN rules.

22. List the applications of fuzzy logic.

- Facial pattern recognition
- Air conditioners, washing machines, and vacuum cleaners
- Antiskid braking systems and transmission systems
- Control of subway systems and unmanned helicopters
- Weather forecasting systems
- Project risk assessment
- Medical diagnosis and treatment plans
- Stock trading

23. What is a partial-order planning?

A problem has to be solved in a sequential approach to attain the goal. The partial-order plan specifies all actions that need to be undertaken but specifies an order of the actions only when required.

24. What is FOPL?

First-order predicate logic is a collection of formal systems, where each statement is divided into a subject and a predicate. The predicate refers to only one subject, and it can either modify or define the properties of the subject.

25. What is the difference between inductive, deductive, and abductive Machine Learning?

Inductive Machine Learning	Deductive Machine Learning	Abductive Machine Learning
Learns from a set of instances to draw the conclusion	Derives the conclusion and then improves it based on the previous decisions	It is a Deep Learning technique where conclusions are derived based on various instances
Statistical Machine Learning such as KNN (K-nearest neighbor) or SVM (Support Vector Machine)	Machine Learning algorithm using a decision tree	Deep neural networks

$A \wedge B \vdash A \rightarrow B$ (Induction)	$A \wedge (A \rightarrow B) \vdash B$ (Deduction)	$B \wedge (A \rightarrow B) \vdash A$ (Abduction)

26. List the different algorithm techniques in Machine Learning.

Here are some of the most commonly used Machine Learning Algorithms

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning
- Transduction
- Learning to Learn

27. What is Deep Learning?

Deep Learning is a subset of Machine Learning which is used to create an artificial multi-layer neural network. It has self-learning capabilities based on previous instances, and it provides high accuracy.

28. Differentiate between supervised, unsupervised, and reinforcement learning.

Differentiation Based on	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Features	The training set has both predictors and predictions.	The training set has only predictors.	It can establish state-of-the-art results on any task.
Algorithms	Linear and logistic regression, support vector machine, and Naive Bayes	K-means clustering algorithm and dimensionality reduction algorithms	Q-learning, state-action-reward-state-action (SARSA), and Deep Q Network (DQN)
Uses	Image recognition, speech recognition, forecasting, etc.	Preprocessing data, pre-training supervised learning algorithms, etc.	Warehouses, inventory management, delivery management, power system, financial systems, etc.

29. Differentiate between parametric and non-parametric models.

Differentiation Based on	Parametric Model	Non-parametric Model
Features	A finite number of parameters to predict new data	Unbounded number of parameters
Algorithm	Logistic regression, linear discriminant analysis, perceptron, and Naive Bayes	K-nearest neighbors, decision trees like CART and C4.5, and support vector machines
Benefits	Simple, fast, and less data	Flexibility, power, and performance

Limitations	Constrained, limited complexity, and poor fit	More data, slower, and overfitting

30. Name a few Machine Learning algorithms you know.

- Logistic regression
- Linear regression
- Decision trees
- Support vector machines
- Naive Bayes, and so on

31. What is Naive Bayes?

Naive Bayes Machine Learning algorithm is a powerful algorithm for predictive modeling. It is a set of algorithms with a common principle based on Bayes Theorem. The fundamental Naive Bayes assumption is that each feature makes an independent and equal contribution to the outcome.

32. What is a Backpropagation Algorithm?

Backpropagation is a Neural Network algorithm that is mainly used to process noisy data and detect unrecognized patterns for better clarification. It's a full-state algorithm and has an iterative nature. As an ANN algorithm, Backpropagation has three layers, Input, hidden, and output layer.

The input layers receive the input values and constraints from the user or the outside environment. After that, the data goes to the Hidden layer where the processing is done. At last, the processed data is transformed into some values or patterns that can be shared using the output layer.

Before processing the data, the following values should be there with the algorithm:

- **Dataset:** The dataset which is going to be used for training a model.
- **Target Attributes**: Output values that an algorithm should achieve after processing the data.
- **Weights**: In a neural network, weights are the parameters that transform input data within the hidden layer.
- **Biases**: At each node, some values called bias are added to the sum calculated(except input nodes).

Backpropagation is simple ANN algorithm that follows a standard approach for training ML models. It doesn't require high computational performance and is widely used in speed recognition, image processing, and optical character recognition(OCR).

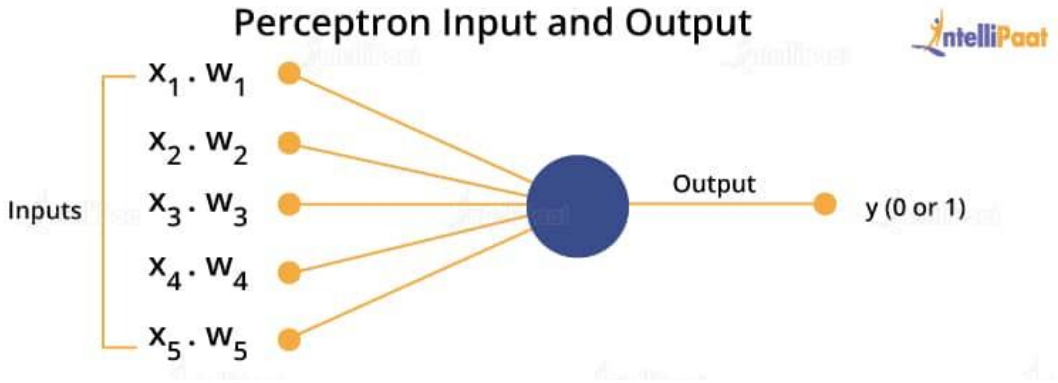
33. How route weights are optimized to reduce the error in the model?

Weights in AI determine how much influence the input is going to have on the output. In neural networks, algorithms use weights to process the information and train the model. The output is expected to be the same as the target attributes.

However, the output may have some errors, which need to be rectified to produce the exact output. For example, in the Backpropagation algorithm when there is an error in the output, the algorithm will backpropagate to the hidden layer and reroute the weights to get an optimized output.

34. What is perceptron in Machine Learning?

Perceptron is an algorithm that is able to simulate the ability of the human brain to understand and discard; it is used for the supervised classification of the input into one of the several possible non-binary outputs.



Wish to achieve a complete knowledge of Artificial Intelligence? Enroll in Artificial Intelligence Training at Hyderabad now!

35. List the extraction techniques used for dimensionality reduction.

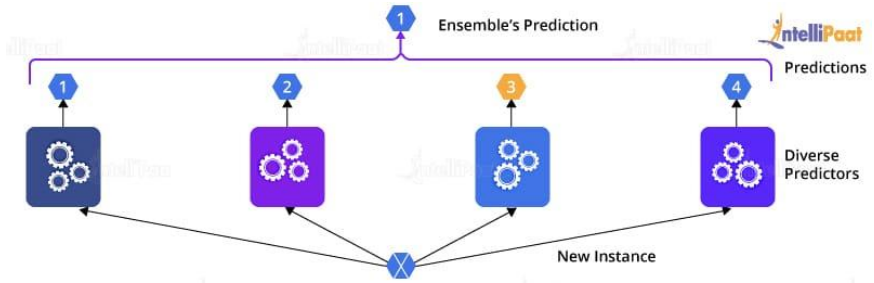
- Independent component analysis
- Principal component analysis
- Kernel-based principal component analysis

36. Is KNN different from K-means Clustering?

KNN	K-means Clustering
Supervised	Unsupervised
Classification algorithms	Clustering algorithms
Minimal training model	Exhaustive training model
Used in the classification and regression of the known data	Used in population demographics, market segmentation, social media trends, anomaly detection, etc.

37. What is ensemble learning?

Ensemble learning is a computational technique in which classifiers or experts are strategically formed and combined. It is used to improve classification, prediction, function approximation, etc. of a model.



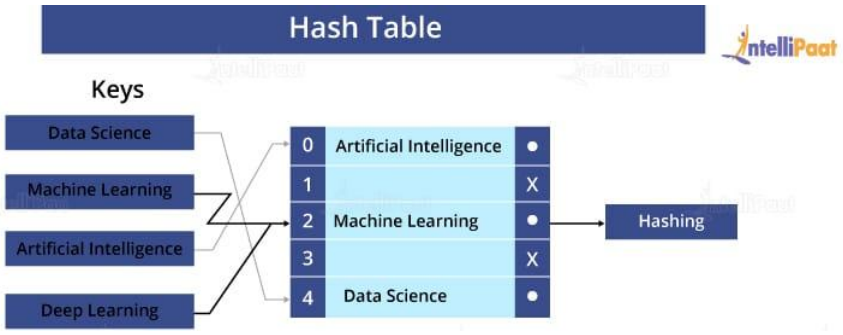
Are you interested in learning AI? Enroll in the [Artificial Intelligence Course in Bangalore] and learn from experts!

38. List the steps involved in Machine Learning.

- Data collection
- Data preparation
- Choosing an appropriate model
- Training the dataset
- Evaluation
- Parameter tuning
- Predictions

39. What is a hash table?

A hash table is a data structure that is used to produce an associative array which is mostly used for database indexing.



40. What is regularization in Machine Learning?

Regularization comes into the picture when a model is either overfit or underfit. It is basically used to minimize the error in a dataset. A new piece of information is fit into the dataset to avoid fitting issues.

41. What are the components of relational evaluation techniques?

- Data acquisition
- Ground truth acquisition

- Cross validation technique
- Query type
- Scoring metric
- Significance test

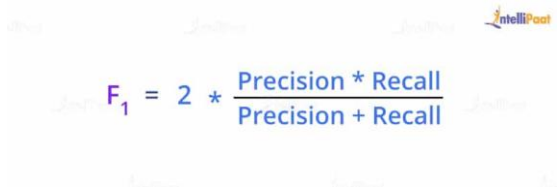
These are described in detail on our AI Community!

42. What is model accuracy and model performance?

Model accuracy, a subset of model performance, is based on the model performance of an algorithm. Whereas, model performance is based on the datasets we feed as inputs to the algorithm.

43. Define F1 score.

F1 score is the weighted average of precision and recall. It considers both false positive and false negative values into account. It is used to measure a model's performance.



44. List the applications of Machine Learning.

- Image, speech, and face detection
- Bioinformatics
- Market segmentation
- Manufacturing and inventory management
- Fraud detection, and so on

45. Can you name three feature selection techniques in Machine Learning?

1. Univariate Selection
2. Feature Importance
3. Correlation Matrix with Heatmap

46. What is a recommendation system?

A recommendation system is an information filtering system that is used to predict user preference based on choice patterns followed by the user while browsing/using the system.

47. What methods are used for reducing dimensionality?

Dimensionality reduction is the process of reducing the number of random variables. We can reduce dimensionality using techniques such as missing values ratio, low variance filter, high correlation filter, random forest, principal component analysis, etc.

48. List different methods for sequential supervised learning.

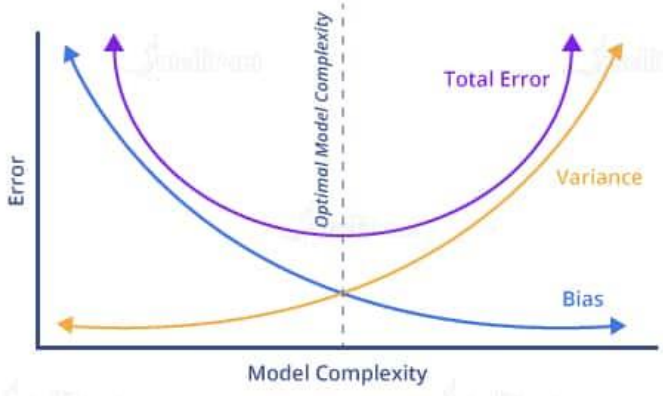
- Sliding window methods
- Recurrent sliding windows methods
- Hidden Markov models
- Maximum entropy Markov models
- Conditional random fields
- Graph transformer networks

49. What are the advantages of neural networks?

- Require less formal statistical training
- Have the ability to detect nonlinear relationships between variables
- Detect all possible interactions between predictor variables
- Availability of multiple training algorithms

50. What is Bias–Variance tradeoff?

Bias error is used to measure how much on an average the predicted values vary from the actual values. In case a high-bias error occurs, we have an under-performing model.



Variance is used to measure how the predictions made on the same observation differ from each other. A high-variance model will overfit the dataset and perform badly on any observation.

51. What is TensorFlow?

TensorFlow is an open-source Machine Learning library. It is a fast, flexible, and low-level toolkit for doing complex algorithms and offers users customizability to build experimental learning architectures and to work on them to produce desired outputs.

52. How to install TensorFlow?

TensorFlow Installation Guide:

CPU : pip install tensorflow-cpu

GPU : pip install tensorflow-gpu

53. What are the TensorFlow objects?

1. Constants
2. Variables
3. Placeholder
4. Graph
5. Session

54. What is a cost function?

A cost function is a scalar function that quantifies the error factor of the neural network. Lower the cost function better the neural network. For example, while classifying the image in the MNIST dataset, the input image is digit 2, but the neural network wrongly predicts it to be 3.

55. List different activation neurons or functions.

1. Linear neuron
2. Binary threshold neuron
3. Stochastic binary neuron
4. Sigmoid neuron
5. Tanh function
6. Rectified linear unit (ReLU)

56. What are the hyper parameters of ANN?

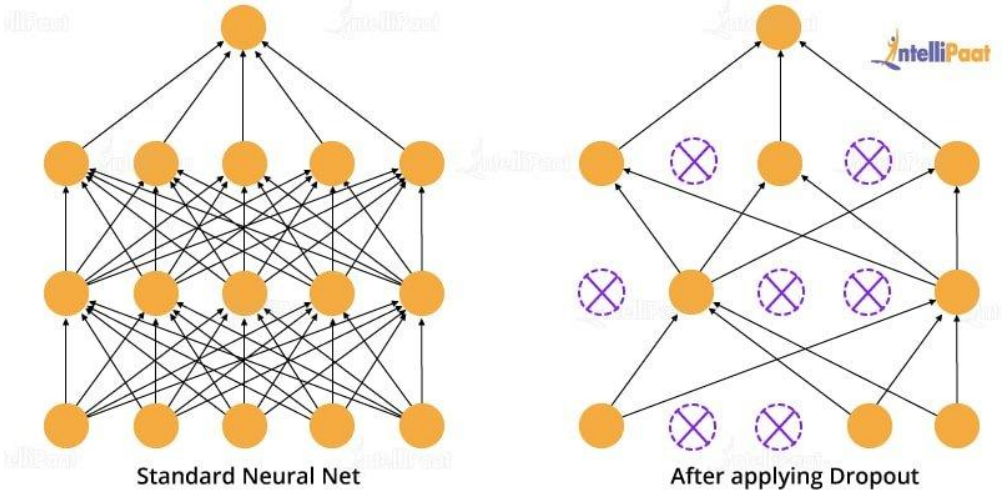
- **Learning rate:** The learning rate is how fast the network learns its parameters.
- **Momentum:** It is a parameter that helps to come out of the local minima and smoothen the jumps while gradient descent.
- **Number of epochs:** The number of times the entire training data is fed to the network while training is referred to as the number of epochs. We increase the number of epochs until the validation accuracy starts decreasing, even if the training accuracy is increasing (overfitting).

57. What is vanishing gradient?

As we add more and more hidden layers, backpropagation becomes less useful in passing information to the lower layers. In effect, as information is passed back, the gradients begin to vanish and become small relative to the weights of the network.

58. What are dropouts?

Dropout is a simple way to prevent a neural network from overfitting. It is the dropping out of some of the units in a neural network. It is similar to the natural reproduction process, where the nature produces offsprings by combining distinct genes (dropping out others) rather than strengthening the co-adapting of them.



59. Define LSTM.

Long short-term memory (LSTM) is explicitly designed to address the long-term dependency problem, by maintaining a state of what to remember and what to forget.

60. List the key components of LSTM.

- Gates (forget, Memory, update, and Read)
- Tanh(x) (values between −1 and 1)
- Sigmoid(x) (values between 0 and 1)

61. List the variants of RNN.

- LSTM: Long Short-term Memory
- GRU: Gated Recurrent Unit
- End-to-end Network
- Memory Network

62. What is an autoencoder? Name a few applications.

An autoencoder is basically used to learn a compressed form of the given data. A few applications of an autoencoder are given below:

1. Data denoising
2. Dimensionality reduction
3. Image reconstruction

4. Image colorization

63. What are the components of the generative adversarial network (GAN)? How do you deploy it?

Components of GAN:

- Generator
- Discriminator

Deployment Steps:

- Train the model
- Validate and finalize the model
- Save the model
- Load the saved model for the next prediction

64. What are the steps involved in the gradient descent algorithm?

Gradient descent is an optimization algorithm that is used to find the coefficients of parameters that are used to reduce the cost function to a minimum.

Step 1: Allocate weights (x,y) with random values and calculate the error (SSE)

Step 2: Calculate the gradient, i.e., the variation in SSE when the weights (x,y) are changed by a very small value. This helps us move the values of x and y in the direction in which SSE is minimized

Step 3: Adjust the weights with the gradients to move toward the optimal values where SSE is minimized

Step 4: Use new weights for prediction and calculating the new SSE

Step 5: Repeat Steps 2 and 3 until further adjustments to the weights do not significantly reduce the error

65. What do you understand by session in TensorFlow?

Syntax: Class Session

It is a class for running TensorFlow operations. The environment is encapsulated in the session object wherein the operation objects are executed and Tensor objects are evaluated.

```
# Build a graph
x = tf.constant(2.0)
y = tf.constant(5.0)
z = x * y
# Launch the graph in a session
sess = tf.Session()
# Evaluate the tensor `z`
print(sess.run(z))
```

66. What do you mean by TensorFlow cluster?

TensorFlow cluster is a set of ‘tasks’ that participate in the distributed execution of a TensorFlow graph. Each task is associated with a TensorFlow server, which contains a ‘master’ that can be used to create sessions and a ‘worker’ that executes operations in the graph. A cluster can also be divided into one or more ‘jobs’, where each job contains one or more tasks.

67. How to run TensorFlow on Hadoop?

To use HDFS with TensorFlow, we need to change the file path for reading and writing data to an HDFS path. For example:

```
filename_queue = tf.train.string_input_producer([
"hdfs://namenode:8020/path/to/file1.csv",
"hdfs://namenode:8020/path/to/file2.csv",
])
```

68. What are intermediate tensors? Do sessions have lifetime?

The intermediate tensors are tensors that are neither inputs nor outputs of the Session.run() call, but are in the path leading from the inputs to the outputs; they will be freed at or before the end of the call.

Sessions can own resources, few classes like tf.Variable, tf.QueueBase, and tf.ReaderBase, and they use a significant amount of memory. These resources (and the associated memory) are released when the session is closed, by calling tf.Session.close.

69. What is the lifetime of a variable?

When we first run the tf.Variable.initializer operation for a variable in a session, it is started. It is destroyed when we run the tf.Session.close operation.

70. Is it possible to solve logical inference in propositional logic?

Yes, logical inference can easily be solved in propositional logic by making use of three concepts:

- Logical equivalence
- Process satisfaction
- Validation checking

71. How does face verification work?

Face verification is used by a lot of popular firms these days. Facebook is famous for the usage of DeepFace for its face verification needs.

There are four main things you must consider when understanding how face verification works:

- **Input**: Scanning an image or a group of images
- **Process**:
-

- Detection of facial features
- Feature comparison and alignment
- Key pattern representation
- Final image classification

- **Output**: Face representation, which is a result of a multilayer neural network
- **Training data**: Involves the usage of thousands of millions of images

The implementation of face verification in Python requires special libraries such as glob, NumPy, OpenCV(cv2), and face_recognisation. Among them, OpenCV is one of the most widely used libraries for computer vision and image processing.

OpenCV is a beginner-friendly, cross-platform python library that is mainly used for real-time image and video processing applications. WithOpenCV, you can create applications used for object detection, facial recognition, and object tracking. It can also be used to extract the facial features and identify unique patterns for face verification.

72. What are some of the algorithms used for hyperparameter optimization?

There are many algorithms that are used for hyperparameter optimization, and following are the three main ones that are widely used:

- Bayesian optimization
- Grid search
- Random search

73. What is overfitting? How is overfitting fixed?

Overfitting is a situation that occurs in statistical modeling or Machine Learning where the algorithm starts to over-analyze data, thereby receiving a lot of noise rather than useful information. This causes low bias but high variance, which is not a favorable outcome.

Overfitting can be prevented by using the below-mentioned methods:

- Early stopping
- Ensemble models
- Cross-validation
- Feature removal
- Regularization

74. How is overfitting avoided in neural networks?

Overfitting is avoided in neural nets by making use of a regularization technique called ‘dropout.’

By making use of the concept of dropouts, random neurons are dropped when the neural network is being trained to use the model doesn’t overfit. If the dropout value is too low, it will have a minimal effect. If it is too high, the model will have difficulty in learning.

1. What do you understand by linear regression?

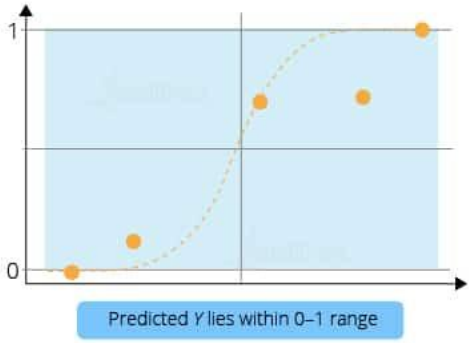
Linear regression helps in understanding the linear relationship between the dependent and the independent variables. Linear regression is a supervised learning algorithm, which helps in finding the linear relationship between two variables. One is the predictor or the independent variable and the other is the response or the dependent variable. In Linear Regression, we try to understand how the dependent variable changes w.r.t the independent variable. If there is only one independent variable, then it is called simple linear regression, and if there is more than one independent variable then it is known as multiple linear regression.

Interested in learning Data Science? Click here to learn more in this Data Science Course!

2. What do you understand by logistic regression?

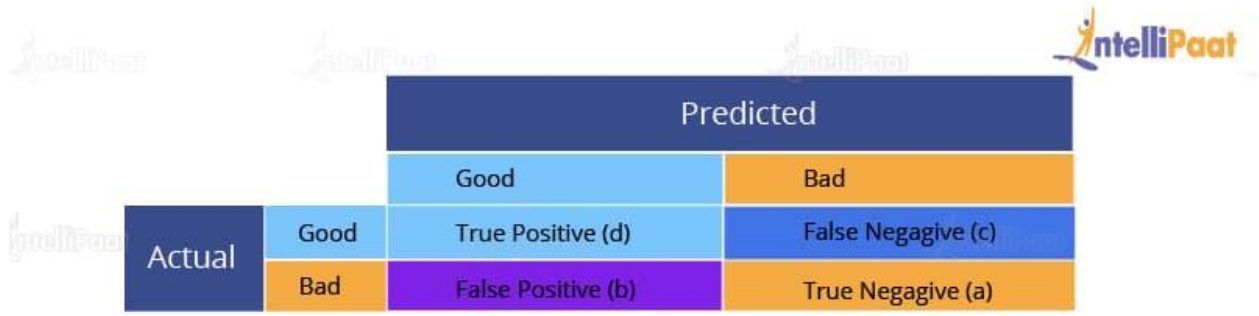
Logistic regression is a classification algorithm that can be used when the dependent variable is binary. Let's take an example. Here, we are trying to determine whether it will rain or not on the basis of temperature and humidity.

Temperature and humidity are the independent variables, and rain would be our dependent variable. So, the logistic regression algorithm actually produces an **S** shape curve. Now, let us look at another scenario: Let's suppose that x-axis represents the runs scored by Virat Kohli and the y-axis represents the probability of the team India winning the match. From this graph, we can say that if Virat Kohli scores more than 50 runs, then there is a greater probability for team India to win the match. Similarly, if he scores less than 50 runs then the probability of team India winning the match is less than 50 percent. So, basically in logistic regression, the *y* value lies within the range of 0 and 1. This is how logistic regression works.



3. What is a confusion matrix?

The confusion matrix is a table that is used to estimate the performance of a model. It tabulates the actual values and the predicted values in a 2×2 matrix. **True Positive (d)**: This denotes all of those records where the actual values are true and the predicted values are also true. So, these denote all of the true positives. **False Negative (c)**: This denotes all of those records where the actual values are true, but the predicted values are false. **False Positive (b):** In this, the actual values are false, but the predicted values are true. **True Negative (a):** Here, the actual values are false and the predicted values are also false. So, if you want to get the correct values, then correct values would basically represent all of the true positives and the true negatives. This is how the confusion matrix works.



4. What do you understand by true positive rate and false positive rate?

True positive rate: In Machine Learning, true positives rates, which are also referred to as sensitivity or recall, are used to measure the percentage of actual positives which are correctly indentified. **Formula**: True Positive Rate = True Positives/Positives **False positive rate**: False positive rate is basically the probability of falsely rejecting the null hypothesis for a particular test. The false positive rate is calculated as the ratio between the number of negative events wrongly categorized as positive (false positive) upon the total number of actual events. **Formula**: False Positive Rate = False Positives/Negatives.

5. What is Data Science?

Data Science is a field of computer science that explicitly deals with turning data into information and extracting meaningful insights out of it. The reason why Data Science is so popular is that the kind of insights it allows us to draw from the available data has led to some major innovations in several products and companies. Using these insights, we are able to determine the taste of a particular customer, the likelihood of a product succeeding in a particular market, etc.

6. How is Data Science different from traditional application programming?

Data Science takes a fundamentally different approach to building systems that provide value than traditional application development.

In traditional programming paradigms, we used to analyze the input, figure out the expected output, and write code, which contains rules and statements needed to transform the provided input into the expected output. As we can imagine, these rules were not easy to write, especially for those data that even computers had a hard time understanding, e.g., images, videos, etc.

Data Science shifts this process a little bit. In it, we need access to large volumes of data that contain the necessary inputs and their mappings to the expected outputs. Then, we use Data Science algorithms, which use mathematical

analysis to generate rules to map the given inputs to outputs. This process of rule generation is called training. After training, we use some data that was set aside before the training phase to test and check the system's accuracy. The generated rules are a kind of a black box, and we cannot understand how the inputs are being transformed into outputs. However. if the accuracy is good enough, then we can use the system (also called a model).

As described above, in traditional programming, we had to write the rules to map the input to the output, but in Data Science, the rules are automatically generated or learned from the given data. This helped solve some really difficult challenges that were being faced by several companies.

7. Explain the differences between supervised and unsupervised learning.

Supervised and unsupervised learning are two types of Machine Learning techniques. They both allow us to build models. However, they are used for solving different kinds of problems.

Supervised Learning	**Unsupervised Learning**
Works on the data that contains both inputs and the expected output, i.e., the labeled data	Works on the data that contains no mappings from input to output, i.e., the unlabeled data
Used to create models that can be employed to predict or classify things	Used to extract meaningful information out of large volumes of data
Commonly used supervised learning algorithms: Linear regression, decision tree, etc.	Commonly used unsupervised learning algorithms: K-means clustering, Apriori algorithm, etc.

8. What is dimensionality reduction?

Dimensionality reduction is the process of converting a dataset with a high number of dimensions (fields) to a dataset with a lower number of dimensions. This is done by dropping some fields or columns from the dataset. However, this is not done haphazardly. In this process, the dimensions or fields are dropped only after making sure that the remaining information will still be enough to succinctly describe similar information.

9. What is bias in Data Science?

Bias is a type of error that occurs in a Data Science model because of using an algorithm that is not strong enough to capture the underlying patterns or trends that exist in the data. In other words, this error occurs when the data is too complicated for the algorithm to understand, so it ends up building a model that makes simple assumptions. This leads to lower accuracy because of underfitting. Algorithms that can lead to high bias are linear regression, logistic regression, etc.

10. Why Python is used for Data Cleaning in DS?

Data Scientists have to clean and transform the huge data sets in a form that they can work with. It's is important to deal with the redundant data for better results by removing nonsensical outliers, malformed records, missing values, inconsistent formatting, etc.

Python libraries such as Matplotlib, Pandas, Numpy, Keras, and SciPy are extensively used for Data cleaning and analysis. These libraries are used to load and clean the data doe effective analysis. For example, a CSV file named "**Student**" has information about the students of an institute like their names, standard, address, phone number, grades, marks, etc.

11. Why R is used in Data Visualization?

R provides the best ecosystem for data analysis and visualization with more than 12,000 packages in Open-source repositories. It has huge community support, which means you can easily find the solution to your problems on various platforms like StackOverflow.

It has better data management and supports distributed computing by splitting the operations between multiple tasks and nodes, which eventually decreases the complexity and execution time of large datasets.

-

12. What are the popular libraries used in Data Science?

Below are the popular libraries used for data extraction, cleaning, visualization, and deploying DS models:

- **TensorFlow:** Supports parallel computing with impeccable library management backed by Google.
- SciPy: Mainly used for solving differential equations, multidimensional programming, data manipulation, and visualization through graphs and charts.
- Pandas: Used to implement the ETL(Extracting, Transforming, and Loading the datasets) capabilities in business applications.
- Matplotlib: Being free and open-source, it can be used as a replacement for MATLAB, which results in better performance and low memory consumption.
- PyTorch: Best for projects which involve Machine Learning algorithms and Deep Neural Networks.

13. What is variance in Data Science?

Variance is a type of error that occurs in a Data Science model when the model ends up being too complex and learns features from data, along with the noise that exists in it. This kind of error can occur if the algorithm used to train the model has high complexity, even though the data and the underlying patterns and trends are quite easy to discover. This makes the model a very sensitive one that performs well on the training dataset but poorly on the testing dataset, and on any kind of data that the model has not yet seen. Variance generally leads to poor accuracy in testing and results in overfitting.

14. What is pruning in a decision tree algorithm?

Pruning a decision tree is the process of removing the sections of the tree that are not necessary or are redundant. Pruning leads to a smaller decision tree, which performs better and gives higher accuracy and speed.

15. What is entropy in a decision tree algorithm?

In a decision tree algorithm, entropy is the measure of impurity or randomness. The entropy of a given dataset tells us how pure or impure the values of the dataset are. In simple terms, it tells us about the variance in the dataset.

For example, suppose we are given a box with 10 blue marbles. Then, the entropy of the box is 0 as it contains marbles of the same color, i.e., there is no impurity. If we need to draw a marble from the box, the probability of it being blue will be 1.0. However, if we replace 4 of the blue marbles with 4 red marbles in the box, then the entropy increases to 0.4 for drawing blue marbles.

16. What is information gain in a decision tree algorithm?

When building a decision tree, at each step, we have to create a node that decides which feature we should use to split data, i.e., which feature would best separate our data so that we can make predictions. This decision is made using information gain, which is a measure of how much entropy is reduced when a particular feature is used to split the data. The feature that gives the highest information gain is the one that is chosen to split the data.

17. What is k-fold cross-validation?

In k-fold cross-validation, we divide the dataset into *k* equal parts. After this, we loop over the entire dataset k times. In each iteration of the loop, one of the k parts is used for testing, and the other k − 1 parts are used for training. Using k-fold cross-validation, each one of the k parts of the dataset ends up being used for training and testing purposes.

18. Explain how a recommender system works.

A recommender system is a system that many consumer-facing, content-driven, online platforms employ to generate recommendations for users from a library of available content. These systems generate recommendations based on what they know about the users' tastes from their activities on the platform.

For example, imagine that we have a movie streaming platform, similar to Netflix or Amazon Prime. If a user has previously watched and liked movies from action and horror genres, then it means that the user likes watching the movies of these genres. In that case, it would be better to recommend such movies to this particular user. These recommendations can also be generated based on what users with a similar taste like watching.

19. What is a normal distribution?

Data distribution is a visualization tool to analyze how data is spread out or distributed. Data can be distributed in various ways. For instance, it could be with a bias to the left or to the right, or it could all be jumbled up. Data may also be distributed around a central value, i.e., mean, median, etc. This kind of distribution has no bias either to the left or to the right and is in the form of a bell-shaped curve. This distribution also has its mean equal to the median. This kind of distribution is called a normal distribution.

20. What is Deep Learning?

Deep Learning is a kind of Machine Learning, in which neural networks are used to imitate the structure of the human brain, and just like how a brain learns from information, machines are also made to learn from the information that is provided to them. Deep Learning is an advanced version of neural networks to make machines learn from data. In Deep Learning, the neural networks comprise many hidden layers (which is why it is called 'deep' learning) that are connected to each other, and the output of the previous layer is the input of the current layer.

-

21. What is an RNN (recurrent neural network)?

A recurrent neural network, or RNN for short, is a kind of Machine Learning algorithm that makes use of the artificial neural network. RNNs are used to find patterns from a sequence of data, such as time series, stock market, temperature, etc. RNNs are a kind of feedforward network, in which information from one layer passes to another layer, and each node in the network performs mathematical operations on the data. These operations are temporal, i.e., RNNs store contextual information about previous computations in the network. It is called recurrent because it

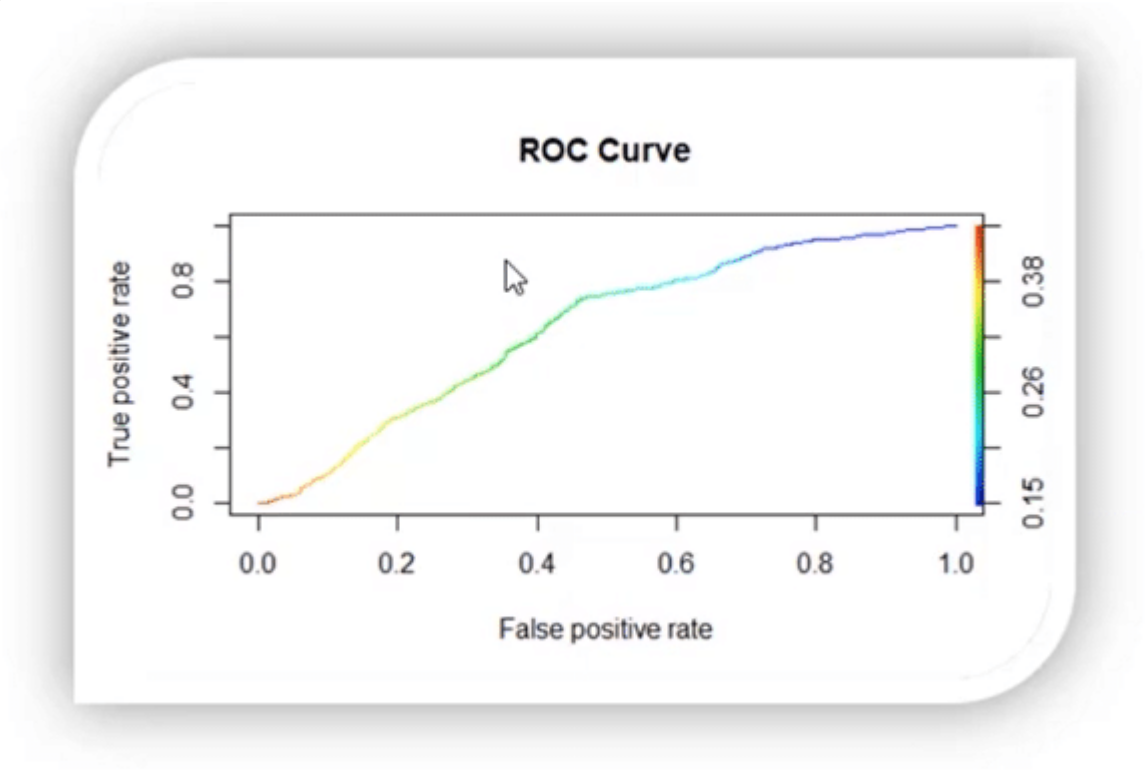
performs the same operations on some data every time it is passed. However, the output may be different based on past computations and their results.

22. Explain selection bias.

Selection bias is the bias that occurs during the sampling of data. This kind of bias occurs when a sample is not representative of the population, which is going to be analyzed in a statistical study.

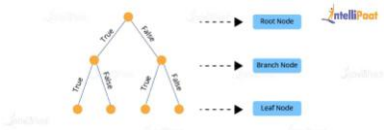
23. What is ROC curve?

It stands for **Receiver Operating Characteristic**. It is basically a plot between a true positive rate and a false positive rate, and it helps us to find out the right tradeoff between the true positive rate and the false positive rate for different probability thresholds of the predicted values. So, the closer the curve to the upper left corner, the better the model is. In other words, whichever curve has greater area under it that would be the better model. You can see this in the below
graph:



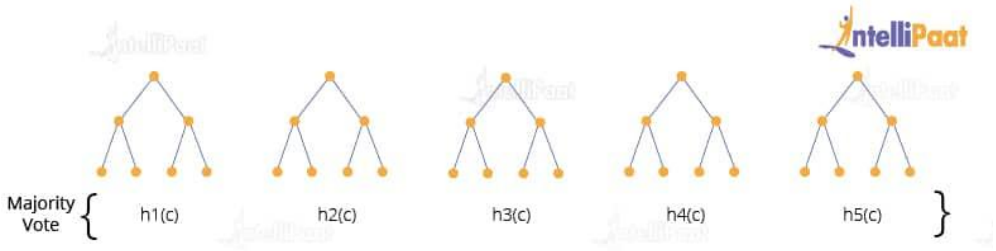
24. What do you understand by a decision tree?

A decision tree is a supervised learning algorithm that is used for both classification and regression. Hence, in this case, the dependent variable can be both a numerical value and a categorical value. Here, each node denotes the test on an attribute, and each edge denotes the outcome of that attribute, and each leaf node holds the class label. So, in this case, we have a series of test conditions which gives the final decision according to the condition.



25. What do you understand by a random forest model?

It combines multiple models together to get the final output or, to be more precise, it combines multiple decision trees together to get the final output. So, decision trees are the building blocks of the random forest model.



26. Two candidates Aman and Mohan appear for a Data Science Job interview. The probability of Aman cracking the interview is 1/8 and that of Mohan is 5/12. What is the probability that at least of them will crack the interview?

The probability of Aman getting selected for the interview is 1/8 P(A) = 1/8
The probability of Mohan getting selected for the interview is 5/12 P(B)=5/12

Now, the probability of at least one of them getting selected can be denoted at the Union of A and B, which means

P(A U B) =P(A)+ P(B) – (P(A ∩ B)) ……(1)

Where P(A ∩ B) stands for the probability of both Aman and Mohan getting selected for the job. To calculate the final answer, we first have to find out the value of P(A ∩ B) So, P(A ∩ B) = P(A) * P(B)

1/8 * 5/12

5/96

Now, put the value of P(A ∩ B) into equation 1

P(A U B) =P(A)+ P(B) – (P(A ∩ B))

1/8 + 5/12 -5/96

So, the answer will be 47/96.

27. How is Data modeling different from Database design?

Data Modeling: It can be considered as the first step towards the design of a database. Data modeling creates a conceptual model based on the relationship between various data models. The process involves moving from the conceptual stage to the logical model to the physical schema. It involves the systematic method of applying data modeling techniques. **Database Design**: This is the process of designing the database. The database design creates an output which is a detailed data model of the database. Strictly speaking, database design includes the detailed logical model of a database but it can also include physical design choices and storage parameters.

28. What are precision?

Precision: When we are implementing algorithms for the classification of data or the retrieval of information, precision helps us get a portion of positive class values that are positively predicted. Basically, it measures the accuracy of correct positive predictions. Below is the formula to calculate precision:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

29. What is recall?

Recall: It is the set of all positive predictions out of the total number of positive instances. Recall helps us identify the misclassified positive predictions. We use the below formula to calculate recall:

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives}$$

30. What is the F1 score and how to calculate it?

F1 score helps us calculate the harmonic mean of precision and recall that gives us the test's accuracy. If F1 = 1, then precision and recall are accurate. If F1 < 1 or equal to 0, then precision or recall is less accurate, or they are completely inaccurate. See below for the formula to calculate the F1 score:

$$\text{F1-score} \triangleq 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

31. What is p-value?

P-value is the measure of the statistical importance of an observation. It is the probability that shows the significance of output to the data. We compute the p-value to know the test statistics of a model. Typically, it helps us choose whether we can accept or reject the null hypothesis.

32. Why do we use p-value?

We use the p-value to understand whether the given data really describe the observed effect or not. We use the below formula to calculate the p-value for the effect 'E' and the null hypothesis 'H0' as true:

$$P\ Value = P(E \mid H0)$$

33. What is the difference between an error and a residual error?

An **error** occurs in values while the prediction gives us the difference between the observed values and the true values of a dataset. Whereas, the **residual error** is the difference between the observed values and the predicted values. The reason we use the residual error to evaluate the performance of an algorithm is that the true values are never known. Hence, we use the observed values to measure the error using residuals. It helps us get an accurate estimate of the error.

34. Why do we use the summary function?

The summary function in R gives us the statistics of the implemented algorithm on a particular dataset. It consists of various objects, variables, data attributes, etc. It provides summary statistics for individual objects when fed into the function. We use a summary function when we want information about the values present in the dataset. It gives us the summary statistics in the following form: Here, it gives the minimum and maximum values from a specific column of the dataset. Also, it provides the median, mean, 1st quartile, and 3rd quartile values that help us understand the values better .

```
 TotalCharges
Min.   :  18.8
1st Qu.: 401.4
Median :1397.5
Mean   :2283.3
3rd Qu.:3794.7
Max.   :8684.8
NA's   :11
```

35. How are Data Science and Machine Learning related to each other?

Data Science and Machine Learning are two terms that are closely related but are often misunderstood. Both of them deal with data. However, there are some fundamental distinctions that show us how they are different from each other.

Data Science is a broad field that deals with large volumes of data and allows us to draw insights out of this voluminous data. The entire process of Data Science takes care of multiple steps that are involved in drawing insights out of the available data. This process includes crucial steps such as data gathering, data analysis, data manipulation, data visualization, etc.

Machine Learning, on the other hand, can be thought of as a sub-field of Data Science. It also deals with data, but here, we are solely focused on learning how to convert the processed data into a functional model, which can be used to map inputs to outputs, e.g., a model that can expect an image as an input and tell us if that image contains a flower as an output.

In short, Data Science deals with gathering data, processing it, and finally, drawing insights from it. The field of Data Science that deals with building models using algorithms is called Machine Learning. Therefore, Machine Learning is an integral part of Data Science.

36. Explain univariate, bivariate, and multivariate analyses.

When we are dealing with data analysis, we often come across terms such as univariate, bivariate, and multivariate. Let's try and understand what these mean.

- **Univariate analysis**: Univariate analysis involves analysing data with only one variable or, in other words, a single column or a vector of the data. This analysis allows us to understand the data and extract patterns and trends out of it. Example: Analyzing the weight of a group of people.
- **Bivariate analysis**: Bivariate analysis involves analyzing the data with exactly two variables or, in other words, the data can be put into a two-column table. This kind of analysis allows us to figure out the relationship between the variables. Example: Analyzing the data that contains temperature and altitude.
- **Multivariate analysis**: Multivariate analysis involves analyzing the data with more than two variables. The number of columns of the data can be anything more than two. This kind of analysis allows us to figure out the effects of all other variables (input variables) on a single variable (the output variable). Example: Analyzing data about house prices, which contains information about the houses, such as locality, crime rate, area, the number of floors, etc.

37. How can we handle missing data?

To be able to handle missing data, we first need to know the percentage of data missing in a particular column so that we can choose an appropriate strategy to handle the situation. For example, if in a column the majority of the data is missing, then dropping the column is the best option, unless we have some means to make educated guesses about the missing values. However, if the amount of missing data is low, then we have several strategies to fill them up.

One way would be to fill them all up with a default value or a value that has the highest frequency in that column, such as 0 or 1, etc. This may be useful if the majority of the data in that column contain these values.

Another way is to fill up the missing values in the column with the mean of all the values in that column. This technique is usually preferred as the missing values have a higher chance of being closer to the mean than to the mode.

Finally, if we have a huge dataset and a few rows have values missing in some columns, then the easiest and fastest way is to drop those columns. Since the dataset is large, dropping a few columns should not be a problem in any way.

38. What is the benefit of dimensionality reduction?

Dimensionality reduction reduces the dimensions and size of the entire dataset. It drops unnecessary features while retaining the overall information in the data intact. Reduction in dimensions leads to faster processing of the data. The reason why data with high dimensions is considered so difficult to deal with is that it leads to high time-consumption while processing the data and training a model on it. Reducing dimensions speeds up this process, removes noise, and also leads to better model accuracy.

39. What is bias–variance trade-off in Data Science?

When building a model using Data Science or Machine Learning, our goal is to build one that has low bias and variance. We know that bias and variance are both errors that occur due to either an overly simplistic model or an

overly complicated model. Therefore, when we are building a model, the goal of getting high accuracy is only going to be accomplished if we are aware of the tradeoff between bias and variance.

Bias is an error that occurs when a model is too simple to capture the patterns in a dataset. To reduce bias, we need to make our model more complex. Although making our model more complex can lead to reducing bias, if we make our model too complex, it may end up becoming too rigid, leading to high variance. So, the tradeoff between bias and variance is that if we increase the complexity, we reduce bias and increase variance, and if we reduce complexity, then we increase bias and reduce variance. Our goal is to find a point at which our model is complex enough to give low bias but not so complex to end up having high variance.

40. What is RMSE?

RMSE stands for the root mean square error. It is a measure of accuracy in regression. RMSE allows us to calculate the magnitude of error produced by a regression model. The way RMSE is calculated is as follows:

First, we calculate the errors in the predictions made by the regression model. For this, we calculate the differences between the actual and the predicted values. Then, we square the errors. After this step, we calculate the mean of the squared errors, and finally, we take the square root of the mean of these squared errors. This number is the RMSE, and a model with a lower value of RMSE is considered to produce lower errors, i.e., the model will be more accurate.

41. What is a kernel function in SVM?

In the SVM algorithm, a kernel function is a special mathematical function. In simple terms, a kernel function takes data as input and converts it into a required form. This transformation of the data is based on something called a kernel trick, which is what gives the kernel function its name. Using the kernel function, we can transform the data that is not linearly separable (cannot be separated using a straight line) into one that is linearly separable.

42. How can we select an appropriate value of k in k-means?

Selecting the correct value of *k* is an important aspect of k-means clustering. We can make use of the elbow method to pick the appropriate k value. To do this, we run the k-means algorithm on a range of values, e.g., 1 to 15. For each value of k, we compute an average score. This score is also called inertia or the inter-cluster variance.

This is calculated as the sum of squares of the distances of all values in a cluster. As k starts from a low value and goes up to a high value, we start seeing a sharp decrease in the inertia value. After a certain value of k, in the range, the drop in the inertia value becomes quite small. This is the value of k that we need to choose for the k-means clustering algorithm.

43. How can we deal with outliers?

Outliers can be dealt with in several ways. One way is to drop them. We can only drop the outliers if they have values that are incorrect or extreme. For example, if a dataset with the weights of babies has a value 98.6-degree Fahrenheit, then it is incorrect. Now, if the value is 187 kg, then it is an extreme value, which is not useful for our model.

In case the outliers are not that extreme, then we can try:

- A different kind of model. For example, if we were using a linear model, then we can choose a non-linear model
- Normalizing the data, which will shift the extreme values closer to other data points

- Using algorithms that are not so affected by outliers, such as random forest, etc.

44. How to calculate the accuracy of a binary classification algorithm using its confusion matrix?

In a binary classification algorithm, we have only two labels, which are True and False. Before we can calculate the accuracy, we need to understand a few key terms:

- True positives: Number of observations correctly classified as True
- True negatives: Number of observations correctly classified as False
- False positives: Number of observations incorrectly classified as True
- False negatives: Number of observations incorrectly classified as False

To calculate the accuracy, we need to divide the sum of the correctly classified observations by the number of total observations. This can be expressed as follows:

45. What is ensemble learning?

When we are building models using Data Science and Machine Learning, our goal is to get a model that can understand the underlying trends in the training data and can make predictions or classifications with a high level of accuracy. However, sometimes some datasets are very complex, and it is difficult for one model to be able to grasp the underlying trends in these datasets. In such situations, we combine several individual models together to improve performance. This is what is called ensemble learning.

46. Explain collaborative filtering in recommender systems.

Collaborative filtering is a technique used to build recommender systems. In this technique, to generate recommendations, we make use of data about the likes and dislikes of users similar to other users. This similarity is estimated based on several varying factors, such as age, gender, locality, etc. If User A, similar to User B, watched and liked a movie, then that movie will be recommended to User B, and similarly, if User B watched and liked a movie, then that would be recommended to User A. In other words, the content of the movie does not matter much. When recommending it to a user what matters is if other users similar to that particular user liked the content of the movie or not.

47. Explain content-based filtering in recommender systems.

Content-based filtering is one of the techniques used to build recommender systems. In this technique, recommendations are generated by making use of the properties of the content that a user is interested in. For example, if a user is watching movies belonging to the action and mystery genre and giving them good ratings, it is a clear indication that the user likes movies of this kind. If shown movies of a similar genre as recommendations, there is a higher probability that the user would like those recommendations as well. In other words, here, the content of the movie is taken into consideration when generating recommendations for users.

48. Explain bagging in Data Science.

Bagging is an ensemble learning method. It stands for bootstrap aggregating. In this technique, we generate some data using the bootstrap method, in which we use an already existing dataset and generate multiple samples of the *N* size. This bootstrapped data is then used to train multiple models in parallel, which makes the bagging model more robust than a simple model. Once all the models are trained, when we have to make a prediction, we make

predictions using all the trained models and then average the result in the case of regression, and for classification, we choose the result, generated by models, that has the highest frequency.

49. Explain boosting in Data Science.

Boosting is one of the ensemble learning methods. Unlike bagging, it is not a technique used to parallelly train our models. In boosting, we create multiple models and sequentially train them by combining weak models iteratively in a way that training a new model depends on the models trained before it. In doing so, we take the patterns learned by a previous model and test them on a dataset when training the new model. In each iteration, we give more importance to observations in the dataset that are incorrectly handled or predicted by previous models. Boosting is useful in reducing bias in models as well.

50. Explain stacking in Data Science.

Just like bagging and boosting, stacking is also an ensemble learning method. In bagging and boosting, we could only combine weak models that used the same learning algorithms, e.g., logistic regression. These models are called homogeneous learners. However, in stacking, we can combine weak models that use different learning algorithms as well. These learners are called heterogeneous learners. Stacking works by training multiple (and different) weak models or learners and then using them together by training another model, called a meta-model, to make predictions based on the multiple outputs of predictions returned by these multiple weak models.

51. Explain how Machine Learning is different from Deep Learning.

A field of computer science, Machine Learning is a subfield of Data Science that deals with using existing data to help systems automatically learn new skills to perform different tasks without having rules to be explicitly programmed.

Deep Learning, on the other hand, is a field in Machine Learning that deals with building Machine Learning models using algorithms that try to imitate the process of how the human brain learns from the information in a system for it to attain new capabilities. In Deep Learning, we make heavy use of deeply connected neural networks with many layers.

52. Why does Naive Bayes have the word ‘naive’ in it?

Naive Bayes is a Data Science algorithm. It has the word ‘Bayes’ in it because it is based on the Bayes theorem, which deals with the probability of an event occurring given that another event has already occurred.

It has ‘naive’ in it because it makes the assumption that each variable in the dataset is independent of the other. This kind of assumption is unrealistic for real-world data. However, even with this assumption, it is very useful for solving a range of complicated problems, e.g., spam email classification, etc.

To learn more about Data Science, check out our Data Science Course in Hyderabad.

Advanced Data Science Interview Questions

53. From the below given ‘diamonds’ dataset, extract only those rows where the ‘price’ value is greater than 1000 and the ‘cut’ is ideal.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
0.24	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48
0.24	Very Good	I	VVS1	62.3	57.0	336	3.95	3.98	2.47
0.26	Very Good	H	SI1	61.9	55.0	337	4.07	4.11	2.53
0.22	Fair	E	VS2	65.1	61.0	337	3.87	3.78	2.49

First, we will load the **ggplot2** package:

```
library(ggplot2)
```

Next, we will use the **dplyr** package:

```
library(dplyr)// It is based on the grammar of data manipulation.
```

To extract those particular records, use the below command:

```
diamonds %>% filter(price>1000 & cut==”Ideal”)-> diamonds_1000_idea
```

54. Make a scatter plot between ‘price’ and ‘carat’ using ggplot. ‘Price’ should be on y-axis, ’carat’ should be on x-axis, and the ‘color’ of the points should be determined by ‘cut.’

We will implement the scatter plot using **ggplot**.

The ggplot is based on the grammar of data visualization, and it helps us stack multiple layers on top of each other.

So, we will start with the data layer, and on top of the data layer we will stack the aesthetic layer. Finally, on top of the aesthetic layer we will stack the geometry layer.

Code:

```
>ggplot(data=diamonds, aes(x=caret, y=price, col=cut))+geom_point()
```

55. Introduce 25 percent missing values in this ‘iris’ datset and impute the ‘Sepal.Length’ column with ‘mean’ and the ‘Petal.Length’ column with ‘median.’

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

To introduce missing values, we will be using the **missForest** package:

```
library(missForest)
```

Using the prodNA function, we will be introducing 25 percent of missing values:

```
Iris.mis<-prodNA(iris,noNA=0.25)
```

For imputing the 'Sepal.Length' column with 'mean' and the 'Petal.Length' column with 'median,' we will be using the Hmisc package and the impute function:

```
library(Hmisc)
iris.mis$Sepal.Length<-with(iris.mis, impute(Sepal.Length,mean))
iris.mis$Petal.Length<-with(iris.mis, impute(Petal.Length,median))
```

56. Implement simple linear regression in R on this 'mtcars' dataset, where the dependent variable is 'mpg' and the independent variable is 'disp.'

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Here, we need to find how 'mpg' varies w.r.t displacement of the column.

We need to divide this data into the training dataset and the testing dataset so that the model does not overfit the data.

So, what happens is when we do not divide the dataset into these two components, it overfits the dataset. Hence, when we add new data, it fails miserably on that new data.

Therefore, to divide this dataset, we would require the **caret** package. This caret package comprises the **createdatapartition()** function. This function will give the true or false labels.

Here, we will use the following code:

```
libraray(caret)

split_tag<-createDataPartition(mtcars$mpg, p=0.65, list=F)

mtcars[split_tag,]->train

mtcars[-split_tag,]->test

lm(mpg-data,data=train)->mod_mtcars

predict(mod_mtcars,newdata=test)->pred_mtcars
```

```
>head(pred_mtcars)
```

Explanation:

Parameters of the createDataPartition function: First is the column which determines the split (it is the mpg column).

Second is the split ratio which is 0.65, i.e., 65 percent of records will have true labels and 35 percent will have false labels. We will store this in split_tag object.

Once we have **split_tag** object ready, from this entire **mtcars dataframe,** we will select all those records where the split tag value is **true** and store those records in the **training** set.

Similarly, from the mtcars dataframe, we will select all those record where the split_tag value is **false** and store those records in the **test** set.

So, the split tag will have true values in it, and when we put '-' symbol in front of it, '-split_tag' will contain all of the false labels. We will select all those records and store them in the test set.

We will go ahead and build a model on top of the training set, and for the simple linear model we will require the **lm function**.

```
lm(mpg-data,data=train)->mod_mtcars
```

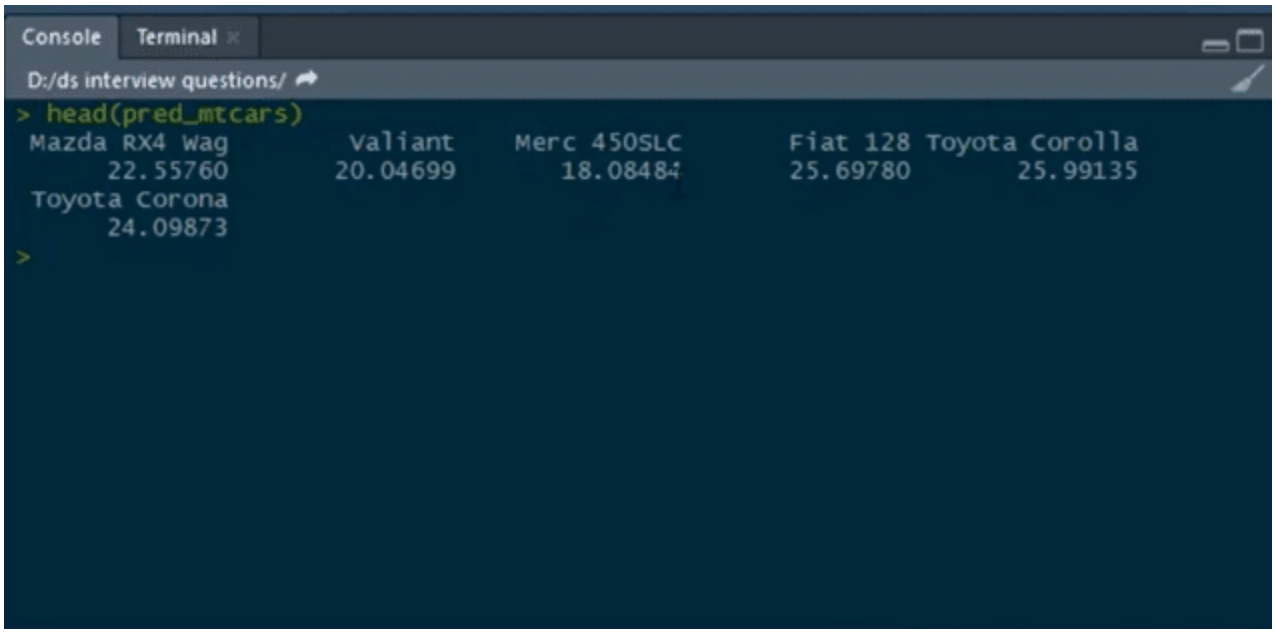
Now, we have built the model on top of the train set. It's time to predict the values on top of the test set. For that, we will use the **predict** function that takes in two parameters: first is the model which we have built and second is the dataframe on which we have to predict values.

Thus, we have to predict values for the test set and then store them in pred_mtcars.

```
predict(mod_mtcars,newdata=test)->pred_mtcars
```

Output:

These are the predicted values of mpg for all of these cars.



So, this is how we can build simple linear model on top of this mtcars dataset.

57. Calculate the RMSE values for the model built.

When we build a regression model, it predicts certain *y* values associated with the given *x* values, but there is always an error associated with this prediction. So, to get an estimate of the average error in prediction, RMSE is used. **Code:**

```
cbind(Actual=test$mpg, predicted=pred_mtcars)->final_data

as.data.frame(final_data)->final_data

error<-(final_data$Actual-final_data$Prediction)

cbind(final_data,error)->final_data

sqrt(mean(final_data$error)^2)
```

Explanation: We have the actual and the predicted values. We will bind both of them into a single dataframe. For that, we will use the **cbind** function:

```
cbind(Actual=test$mpg, predicted=pred_mtcars)->final_data
```

Our actual values are present in the **mpg** column from the test set, and our predicted values are stored in the **pred_mtcars** object which we have created in the previous question. Hence, we will create this new column and name the column **actual.** Similarly, we will create another column and name it **predicted** which will have predicted values and then store the predicted values in the new object which is **final_data**. After that, we will convert a matrix into a dataframe. So, we will use the **as.data.frame** function and convert this object (predicted values) into a dataframe:

```
as.data.frame(final_data)->final_data
```

We will pass this object which is final_data and store the result in final_data again. We will then calculate the error in prediction for each of the records by subtracting the predicted values from the actual values:

```
error<-(final_data$Actual-final_data$Prediction)
```

Then, store this result on a new object and name that object as **error**. After this, we will bind this error calculated to the same final_data dataframe:

```
cbind(final_data,error)->final_data //binding error object to this final_data
```

Here, we bind the error object to this final_data, and store this into final_data again. **Calculating RMSE**:

```
Sqrt(mean(final_data$error)^2)
```

Output:

```
[1] 4.334423
```

Note: Lower the value of RMSE, the better the model. ***R and Python are two of the most important programming languages for Machine Learning Algorithms.***

58. Implement simple linear regression in Python on this 'Boston' dataset where the dependent variable is 'medv' and the independent variable is 'lstat.'

Simple Linear Regression

```
import pandas as pd

data=pd.read_csv('Boston.csv')     //loading the Boston dataset
```

```
data.head()  //having a glance at the head of this data

data.shape
```

Let us take out the dependent and the independent variables from the dataset:

```
data1=data.loc[:,['lstat','medv']]

data1.head()
```

Visualizing Variables

```
import matplotlib.pyplot as plt

data1.plot(x='lstat',y='medv',style='o')

plt.xlabel('lstat')

plt.ylabel('medv')

plt.show()
```

Here, 'medv' is basically the median values of the price of the houses, and we are trying to find out the median values of the price of the houses w.r.t to the lstat column.

We will separate the dependent and the independent variable from this entire dataframe:

```
data1=data.loc[:,['lstat','medv']]
```

The only columns we want from all of this record are 'lstat' and 'medv,' and we need to store these results in data1.

Now, we would also do a visualization w.r.t to these two columns:

```
import matplotlib.pyplot as plt

data1.plot(x='lstat',y='medv',style='o')

plt.xlabel('lstat')

plt.ylabel('medv')

plt.show()
```

Preparing the Data

```
X=pd.Dataframe(data1['lstat'])

Y=pd.Dataframe(data1['medv'])

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=100)

from sklearn.linear_model import LinearRegression

regressor=LinearRegression()
```

```
regressor.fit(X_train,y_train)
print(regressor.intercept_)
```

Output :

```
34.12654201
print(regressor.coef_)//this is the slope
```

Output :

```
[[-0.913293]]
```

By now, we have built the model. Now, we have to predict the values on top of the test set:

```
y_pred=regressor.predict(X_test)//using the instance and the predict function and pass the
X_test object inside the function and store this in y_pred object
```

Now, let's have a glance at the rows and columns of the actual values and the predicted values:

```
Y_pred.shape, y_test.shape
```

Output :

```
((102,1),(102,1))
```

Further, we will go ahead and calculate some metrics so that we can find out the Mean Absolute Error, Mean Squared Error, and RMSE.

```
from sklearn import metrics import NumPy as np

print('Mean Absolute Error: ', metrics.mean_absolute_error(y_test, y_pred))

print('Mean Squared Error: ', metrics.mean_squared_error(y_test, y_pred))

print('Root Mean Squared Error: ', np.sqrt(metrics.mean_absolute_error(y_test, y_pred))
```

Output:

```
Mean Absolute Error: 4.692198

Mean Squared Error: 43.9198

Root Mean Squared Error: 6.6270
```

59. Implement logistic regression on this 'heart' dataset in R where the dependent variable is 'target' and the independent variable is 'age.'

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

For loading the dataset, we will use the **read.csv** function:

```
read.csv("D:/heart.csv")->heart

str(heart)
```

In the structure of this dataframe, most of the values are integers. However, since we are building a logistic regression model on top of this dataset, the final **target column is supposed to be categorical**. It cannot be an integer. So, we will go ahead and convert them into a factor. Thus, we will use the **as.factor** function and convert these integer values into categorical data. We will pass on **heart$target** column over here and store the result in **heart$target** as follows:

```
as.factor(heart$target)->heart$target
```

Now, we will build a logistic regression model and see the different probability values for the person to have heart disease on the basis of different age values.

To build a logistic regression model, we will use the **glm** function:

```
glm(target~age, data=heart, family="binomial")->log_mod1
```

Here, **target~age** indicates that the target is the dependent variable and the age is the independent variable, and we are building this model on top of the dataframe.

family="binomial" means we are basically telling R that this is the logistic regression model, and we will store the result in **log_mod1**.

We will have a glance at the summary of the model that we have just built:

```
summary(log_mod1)
```

```
> summary(log_mod1)

Call:
glm(formula = target ~ age, family = "binomial", data = heart)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7125  -1.1773   0.8296   1.0685   1.5947

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.03623    0.75639   4.014 5.97e-05 ***
age         -0.05235    0.01363  -3.841 0.000122 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.64  on 302  degrees of freedom
Residual deviance: 401.86  on 301  degrees of freedom
AIC: 405.86

Number of Fisher Scoring iterations: 4
```

We can see **Pr** value here, and there are three stars associated with this Pr value. This basically means that we can reject the null hypothesis which states that there is no relationship between the age and the target columns. But since we have three stars over here, this null hypothesis can be rejected. There is a strong relationship between the age column and the target column.

Now, we have other parameters like null deviance and residual deviance. Lower the deviance value, the better the model.

This null deviance basically tells the deviance of the model, i.e., when we don't have any independent variable and we are trying to predict the value of the target column with only the intercept. When that's the case, the null deviance is 417.64.

Residual deviance is wherein we include the independent variables and try to predict the target columns. Hence, when we include the independent variable which is age, we see that the residual deviance drops. Initially, when there are no independent variables, the null deviance was 417. After we include the age column, we see that the null deviance is reduced to 401.

This basically means that there is a strong relationship between the age column and the target column and that is why the deviance is reduced.

As we have built the model, it's time to predict some values:

```
predict(log_mod1, data.frame(age=30), type="response")

predict(log_mod1, data.frame(age=50), type="response")

predict(log_mod1, data.frame(age=29:77), type="response")
```

Now, we will divide this dataset into train and test sets and build a model on top of the train set and predict the values on top of the test set:

```
>library(caret)
```

```
Split_tag<- createDataPartition(heart$target, p=0.70, list=F)

heart[split_tag,]->train

heart[-split_tag,]->test

glm(target~age, data=train,family="binomial")->log_mod2

predict(log_mod2, newdata=test, type="response")->pred_heart

range(pred_heart)
```

60. Build an ROC curve for the model built.

The below code will help us in building the ROC curve:

```
library(ROCR)

prediction(pred_heart, test$target)-> roc_pred_heart

performance(roc_pred_heart, "tpr", "fpr")->roc_curve

plot(roc_curve, colorize=T)
```

61. Build a confusion matrix for the model where the threshold value for the probability of predicted values is 0.6, and also find the accuracy of the model.

Accuracy is calculated as:

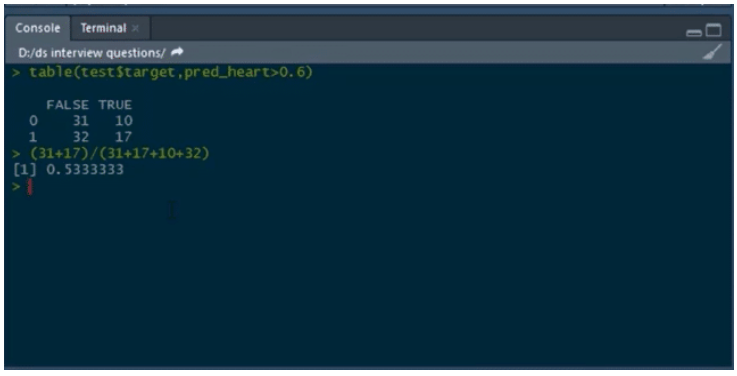
Accuracy = (True positives + true negatives)/(True positives+ true negatives + false positives + false negatives)

To build a confusion matrix in R, we will use the table function:

```
table(test$target,pred_heart>0.6)
```

Here, we are setting the probability threshold as 0.6. So, wherever the probability of pred_heart is greater than 0.6, it will be classified as 0, and wherever it is less than 0.6 it will be classified as 1.

Then, we calculate the accuracy by the formula for calculating **Accuracy**.



62. Build a logistic regression model on the 'customer_churn' dataset in Python. The dependent variable is 'Churn' and the independent variable is 'MonthlyCharges.' Find the log_loss of the model.

First, we will load the pandas dataframe and the customer_churn.csv file:

```
customer_churn=pd.read_csv("customer_churn.csv")
```

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL
9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic
9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic
1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic
6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL
7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic

After loading this dataset, we can have a glance at the head of the dataset by using the following command:

```
customer_churn.head()
```

Now, we will separate the dependent and the independent variables into two separate objects:

```
x=pd.Dataframe(customer_churn['MonthlyCharges'])

y=customer_churn[' Churn']

#Splitting the data into training and testing sets

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.3, random_state=0)
```

Now, we will see how to build the model and calculate **log_loss**.

```
from sklearn.linear_model, we have to import LogisticRegression

l=LogisticRegression()

l.fit(x_train,y_train)

y_pred=l.predict_proba(x_test)
```

As we are supposed to calculate the log_loss, we will import it from **sklearn.metrics**:

```
from sklearn.metrics import log_loss

print(log_loss(y_test,y_pred)//actual values are in y_test and predicted are in y_pred
```

Output:

```
0.5555020595194167
```

Become a master of Data Science by going through this online Data Science Course in Toronto!

63. Build a decision tree model on 'Iris' dataset where the dependent variable is 'Species,' and all other columns are independent variables. Find the accuracy of the model built.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

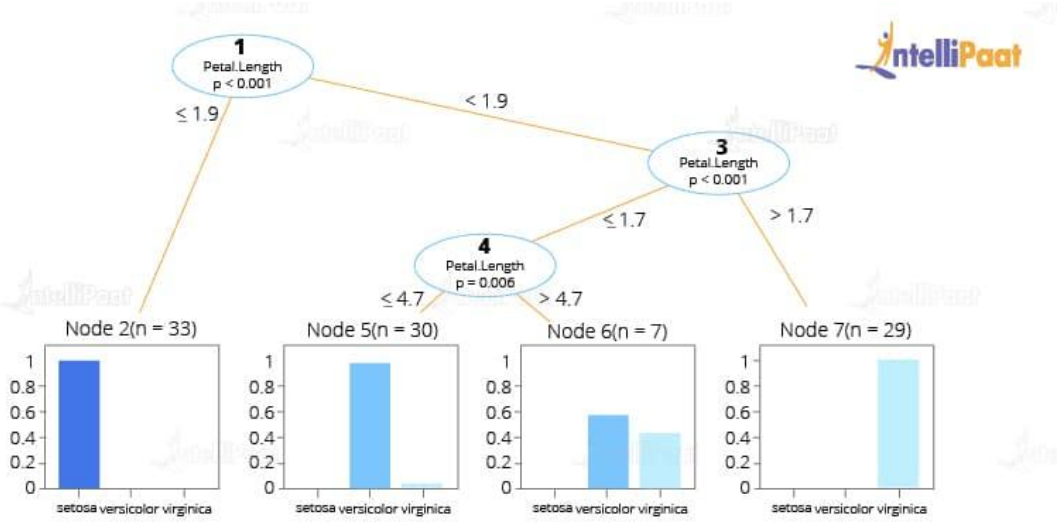
To build a decision tree model, we will be loading the **party** package:

```
#party package

library(party)

#splitting the data

library(caret)

split_tag<-createDataPartition(iris$Species, p=0.65, list=F)

iris[split_tag,]->train

iris[~split_tag,]->test

#building model

mytree<-ctree(Species~.,train)
```

Now we will plot the model

```
plot(mytree)
```

Model:




```
#predicting the values

predict(mytree,test,type='response')->mypred
```

After this, we will predict the confusion matrix and then calculate the accuracy using the table function:

```
table(test$Species, mypred)
```

```
Console   Terminal
D:/ds interview questions/
> table(test$Species,mypred)
            mypred
             setosa versicolor virginica
  setosa         17          0         0
  versicolor      0         16         1
  virginica       0          1        16
> (17+16+16)/(17+16+16+1+1)
[1] 0.9607843
>
```

64. Build a random forest model on top of this 'CTG' dataset, where 'NSP' is the dependent variable and all other columns are independent variables.

Nmax	Nzeros	Mode	Mean	Median	Variance	Tendency	NSP
2	0	120	137	121	73	1	2
6	1	141	136	140	12	0	1
5	1	141	135	138	13	0	1
11	0	137	134	137	13	1	1
9	0	137	136	138	11	1	1
5	3	76	107	107	170	0	3
6	3	71	107	106	215	0	3
0	0	122	122	123	3	1	3
0	0	122	122	123	3	1	3
1	0	122	122	123	1	1	3
2	0	150	148	151	9	1	2

We will load the CTG dataset by using **read.csv**:

```
data<-read.csv("C:/Users/intellipaat/Downloads/CTG.csv",header=True)

str(data)
```

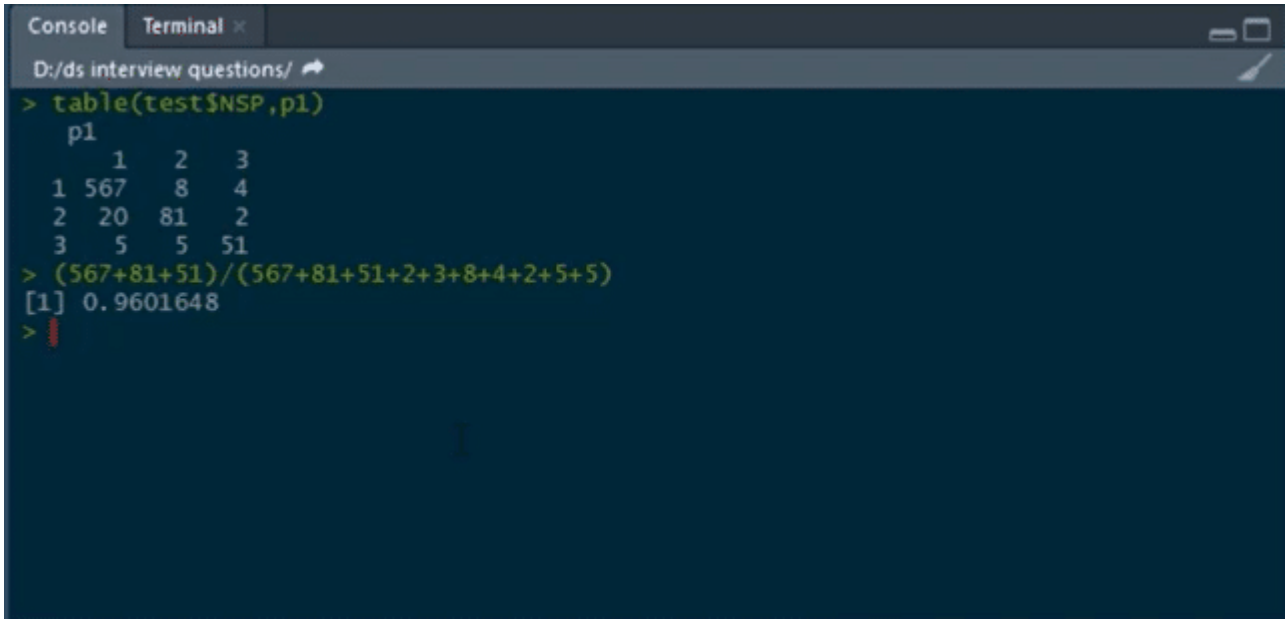
Converting the integer type to a factor

```
data$NSP<-as.factor(data$NSP)

table(data$NSP)

#data partition

set.seed(123)

split_tag<-createDataPartition(data$NSP, p=0.65, list=F)

data[split_tag,]->train

data[~split_tag,]->test

#random forest -1

library(randomForest)

set.seed(222)

rf<-randomForest(NSP~.,data=train)

rf

#prediction
```

```
predict(rf,test)->p1
```

Building confusion matrix and calculating accuracy:

```
table(test$NSP,p1)
```



If you have any doubts or queries related to Data Science, get them clarified from Data Science experts on our [Data Science Community](#)!

65. Write a function to calculate the Euclidean distance between two points.

The formula for calculating the Euclidean distance between two points (x1, y1) and (x2, y2) is as follows:

```
√(((x1 - x2) ^ 2) + ((y1 - y2) ^ 2))
```

Code for calculating the Euclidean distance is as given below:

```
def euclidean_distance(P1, P2):
return (((P1[0] - P2[0]) ** 2) + ((P1[1] - P2[1]) ** 2)) ** .5
```

66. Write code to calculate the root mean square error (RMSE) given the lists of values as actual and predicted.

To calculate the root mean square error (RMSE), we have to:

1. Calculate the errors, i.e., the differences between the actual and the predicted values
2. Square each of these errors
3. Calculate the mean of these squared errors
4. Return the square root of the mean

The code in Python for calculating RMSE is given below:

```
def rmse(actual, predicted):
  errors = [abs(actual[i] - predicted[i]) for i in range(0, len(actual))]
  squared_errors = [x ** 2 for x in errors]
  mean = sum(squared_errors) / len(squared_errors)
```

```
  return mean ** .5
```

Check out this Machine Learning Course to get an in-depth understanding of Machine Learning.

67. Mention the different kernel functions that can be used in SVM.

In SVM, there are four types of kernel functions:

- Linear kernel
- Polynomial kernel
- Radial basis kernel
- Sigmoid kernel

68. How to detect if the time series data is stationary?

Time series data is considered stationary when variance or mean is constant with time. If the variance or mean does not change over a period of time in the dataset, then we can draw the conclusion that, for that period, the data is stationary.

69. Write code to calculate the accuracy of a binary classification algorithm using its confusion matrix.

We can use the code given below to calculate the accuracy of a binary classification algorithm:

```
def accuracy_score(matrix):
  true_positives = matrix[0][0]
  true_negatives = matrix[1][1]
  total_observations = sum(matrix[0]) + sum(matrix[1])
  return (true_positives + true_negatives) / total_observations
```

70. What does root cause analysis mean?

Root cause analysis is the process of figuring out the root causes that lead to certain faults or failures. A factor is considered to be a root cause if, after eliminating it, a sequence of operations, leading to a fault, error, or undesirable result, ends up working correctly. Root cause analysis is a technique that was initially developed and used in the analysis of industrial accidents, but now, it is used in a wide variety of areas.

71. What is A/B testing?

A/B testing is a kind of statistical hypothesis testing for randomized experiments with two variables. These variables are represented as A and B. A/B testing is used when we wish to test a new feature in a product. In the A/B test, we give users two variants of the product, and we label these variants as A and B. The A variant can be the product with the new feature added, and the B variant can be the product without the new feature. After users use these two products, we capture their ratings for the product. If the rating of the product variant A is statistically and significantly higher, then the new feature is considered an improvement and useful and is accepted. Otherwise, the new feature is removed from the product.

Check out this Python Course to get deeper into Python programming.

72. Out of collaborative filtering and content-based filtering, which one is considered better, and why?

Content-based filtering is considered to be better than collaborative filtering for generating recommendations. It does not mean that collaborative filtering generates bad recommendations. However, as collaborative filtering is based on the likes and dislikes of other users we cannot rely on it much. Also, users' likes and dislikes may change in the future. For example, there may be a movie that a user likes right now but did not like 10 years ago. Moreover, users who are similar in some features may not have the same taste in the kind of content that the platform provides.

In the case of content-based filtering, we make use of users' own likes and dislikes that are much more reliable and yield more positive results. This is why platforms such as Netflix, Amazon Prime, Spotify, etc. make use of content-based filtering for generating recommendations for their users.

73. In the following confusion matrix, calculate precision and recall.

Total = 510	**Actual**		
Predicted		P	N
	P	156	11
	N	16	327

The formulae for precision and recall are given below.

```
Precision:
(True Positive) / (True Positive + False Positive)
Recall:
(True Positive) / (True Positive + False Negative)
Based on the given data, precision and recall are:
Precision: 156 / (156 + 11) = 93.4
Recall: 156 / (156 + 16) = 90.7
```

74. Write a function that when called with a confusion matrix for a binary classification model returns a dictionary with its precision and recall.

We can use the below for this purpose:

```
def calculate_precsion_and_recall(matrix):
  true_positive  = matrix[0][0]
  false_positive  = matrix[0][1]
  false_negative = matrix[1][0]
  return {
    'precision': (true_positive) / (true_positive + false_positive),
    'recall': (true_positive) / (true_positive + false_negative)
  }
```

75. What is reinforcement learning?

Reinforcement learning is a kind of Machine Learning, which is concerned with building software agents that perform actions to attain the most number of cumulative rewards. A reward here is used for letting the model know (during training) if a particular action leads to the attainment of or brings it closer to the goal. For example, if we are creating an ML model that plays a video game, the reward is going to be either the points collected during the play or the level reached in it. Reinforcement learning is used to build these kinds of agents that can make real-world decisions that should move the model toward the attainment of a clearly defined goal.

76. Explain TF/IDF vectorization.

The expression ‘TF/IDF’ stands for Term Frequency–Inverse Document Frequency. It is a numerical measure that allows us to determine how important a word is to a document in a collection of documents called a corpus. TF/IDF is used often in text mining and information retrieval.

77. What are the assumptions required for linear regression?

There are several assumptions required for linear regression. They are as follows:

- The data, which is a sample drawn from a population, used to train the model should be representative of the population.
- The relationship between independent variables and the mean of dependent variables is linear.
- The variance of the residual is going to be the same for any value of an independent variable. It is also represented as X.
- Each observation is independent of all other observations.
- For any value of an independent variable, the independent variable is normally distributed.

78. What happens when some of the assumptions required for linear regression are violated?

These assumptions may be violated lightly (i.e., some minor violations) or strongly (i.e., the majority of the data has violations). Both of these violations will have different effects on a linear regression model.

Strong violations of these assumptions make the results entirely redundant. Light violations of these assumptions make the results have greater bias or variance.

1. What is TensorFlow?

TensorFlow is the world’s most used library for Machine Learning. Developed in 2015 by the Google Brain Team, it ensures to provide an easy-to-use low-level toolkit that can handle complex mathematical operations and learning architectures.

2. What are tensors?

Tensors are similar to arrays in programming languages, but here, they are of higher dimensions. It can be considered as a generalization of matrices that form an n-dimensional array. TensorFlow provides methods that can be used to create tensor functions and compute their derivatives easily. This is what sets tensors apart from the NumPy arrays.

3. What is the meaning of TensorBoard?

TensorBoard is a Graphical User Interface (GUI) that is provided by TensorFlow to help users visualize graphs, plots, and other metrics easily without having to write a lot of code. TensorBoard provides an ample number of advantages in terms of readability, ease of use, and performance metrics.

4. What are some of the advantages of using TensorFlow?

TensorFlow has numerous advantages, and this is why it is the most used framework for Machine Learning in the world. Some of its advantages are given below:

- Platform independency
- Usage of GPU for distributed computing
- Auto-differentiation capability
- Open-source and large community
- Highly customizable based on requirements
- Support for asynchronous computations

5. Are there any limitations to using TensorFlow?

Even though TensorFlow provides numerous advantages, it has one or two caveats in the current versions:

- No support for OpenCL (Open Computing Language) yet
- GPU memory conflicts when used with Theano
- Can be overwhelming for beginners to get started

Next up on these TensorFlow coding interview questions, let us check out about the types of tensors.

6. What are the types of tensors available in TensorFlow?

There are three main types of tensors:

- Constant tensors
- Variable tensors
- Placeholder tensors

7. How can data be loaded into TensorFlow?

There are two ways that you can use to load data into TensorFlow before training **Machine Learning algorithms**:

- **Data load into memory**: Here, the data is loaded into the memory as a single array unit. It is the easiest way to load the data.
- **TensorFlow data pipeline**: It is making use of the built-in APIs to load the data and feed it across to the algorithm.

8. What is the simple working of an algorithm in TensorFlow?

There are five main steps that govern the working of the majority of algorithms in **TensorFlow**. They are as follows:

1. Data import or data generation, alongside setting up a data pipeline
2. Data input through computational graphs
3. Generation of the loss function to evaluate the output
4. Backpropagation to modify the data
5. Iterating until output criteria are met

9. What are the methods that can be used to handle overfitting in TensorFlow?

There are three methods, which can be used to easily handle the condition of overfitting when using TensorFlow:

- Batch normalization
- Regularization technique
- Dropouts

10. What are the languages that are supported in TensorFlow?

TensorFlow supports a wide variety of languages for programmers to write the code in. The preferred language presently is Python.

However, experimental support is being implemented for other languages, such as Go, Java, and C++. Also, language bindings for Ruby, Scala, and Julia are being developed by the open-source community.

11. What are placeholder tensors?

Placeholder tensors are entities that provide an advantage over a regular variable. It is used to assign data at a later point in time.

Placeholders can be used to build graphs without any prior data being present. This means that they do not require any sort of initialization for usage.

12. What are managers in TensorFlow?

TensorFlow managers are entities that are responsible for handling the following activities for servable objects:

- Loading
- Unloading
- Lookup
- Lifetime management

13. Where is TensorFlow mostly used?

TensorFlow is used in all of the domains that cover Machine Learning and Deep Learning. Being the most essential tool, the following are some of the main use cases of TensorFlow:

- Time series analysis
- Image recognition
- Voice recognition
- Video upscaling
- Test-based applications

14. What are TensorFlow servables?

Servables in TensorFlow are simply the objects that client machines use to perform computations. The size of these objects is flexible. Servables can include a variety of information like any entity from a lookup table to a tuple needed for inference models.

15. How does the Python API work with TensorFlow?

Python is the primary language when it comes to working with TensorFlow. TensorFlow provides an ample number of functionalities when used with the API, such as:

- Automatic checkpoints
- Automatic logging
- Simple training distribution
- Queue-runner design methods

16. What are some of the APIs outside of the TensorFlow project?

Following are some of the APIs developed by Machine Learning enthusiasts across the globe:

- TFLearn: A popular Python package
- TensorLayer: For layering architecture support
- Pretty Tensor: Google's project providing a chaining interface
- Sonnet: Provides a modular approach to programming

17. What are TensorFlow loaders?

Loaders are used in TensorFlow to load, unload, and work with servable objects. The loaders are primarily used to add algorithms and data into TensorFlow for working.

The load() function is used to pre-load a model from a saved entity easily.

18. What makes TensorFlow advantageous over other libraries?

Following are some of the benefits of TensorFlow over other libraries:

- Pipelines: **data** is used to build efficient pipelines for text and image processing.
- Debugging: **tfdbg** is used to track the state and structure of objects for easy debugging.
- Visualization: TensorBoard provides an elegant user interface for users to visualize graphs.
- Scalability: It can scale Deep Learning applications and their associated infrastructure easily.

19. What are TensorFlow abstractions?

TensorFlow contains certain libraries used for abstraction such as Keras and TF-Slim. They are used to provide high-level access to data and model life cycle for programmers using TensorFlow. This can help them easily maintain clean code and also reduce the length of the code exponentially.

Next up on this top TensorFlow interview questions and answers post, we will take a look at the intermediate set of questions.

Intermediate Interview Questions

20. What are the differences between tf.variable and tf.placeholder in TensorFlow?

tf.variable	tf.placeholder
Defines values for variables that change with time	Defines inputs that do not change with time
Requires initialization when defined	Does not require initialization during defining

21. What is a graph explorer in TensorFlow?

A graph explorer is used to visualize a graph on TensorBoard. It is also used for the inspection operations of a model in TensorFlow. To easily understand the flow in a graph, it is recommended to use a graph visualizer in TensorBoard.

Next up on these TensorFlow coding interview questions, let us check out about variables and their lifetimes.

22. How is variable lifetime tracked in TensorFlow?

The lifetime of a variable is automatically tracked after its initialization, using the **tf.Variable.initializer** operation.

Later, after the usage, the session can be closed and the variable can be destroyed, using the **tf.Session.close** operation.

23. What are the types of dashboards supported by TensorFlow?

TensorFlow supports a variety of dashboards that are used to perform numerous tasks on TensorBoard easily:

- Scalar dashboard
- Image dashboard
- Graph dashboard
- Text dashboard
- Distributer dashboard
- Histogram dashboard

24. Can TensorFlow be deployed onto a container architecture?

Yes, TensorFlow can be used with containerization tools like Docker easily. The containerization tools alongside TensorFlow are mostly used to deploy various models that require text classification using convolutional neural networks.

If you are looking forward to becoming an expert in TensorFlow and AI, make sure to check out Intellipaat's AI Engineer Course.

25. Differentiate between TensorFlow and PyTorch.

TensorFlow	PyTorch
Developed by Google	Developed by Facebook
No support for runtime graph operations	Provides computational graph operations at runtime
Offers TensorBoard for visualization	No visualization tools offered in the bundle
Based on the Theano library	Based on the Torch library

26. Is word embedding supported in TensorFlow?

Yes, word embedding is supported in TensorFlow. It is widely used in the field of Natural Language Processing. When using TensorFlow, it is called Word2vec.

Two models are used for word embedding in TensorFlow:

- The Continuous Bag of Words model
- The Skip-Gram model

Next, it is vital that you understand the use of estimators and that is exactly what we will look at on these TensorFlow coding interview questions.

27. What is the use of estimators in TensorFlow?

Estimators in TensorFlow are high-level APIs used to provide a high amount of code reusability when training a model. They can also override the default behavior of any aspect of the model.

There are two ways of model building using estimators:

- Premade estimator: Used to create a specific model like DNNClassifier
- Base class estimator: Used to control a model using a model_fn function

28. What statistical distribution functions are provided by TensorFlow?

Numerous statistical distribution functions are offered by TensorFlow. They are all located inside the tf.contrib.distributions package.

The distributions supported are:

- Beta
- Bernoulli
- Chi2
- Dirichlet
- Gamma
- Uniform

29. Can you use TensorBoard without installing TensorFlow?

If TensorFlow is not installed, users can still make use of TensorBoard (versions above 1.14) in a standalone mode with redacted features.

Following plugins are supported:

- Scalars
- Image
- Audio
- Graph
- Projector
- Histograms
- Mesh

30. What is the meaning of the embedding projector in TensorFlow?

Embedding projector is an entity in TensorFlow that is used to easily visualize high-dimensional data.

It is used to read the data from the model checkpoint file prior to visualization. It is used to view the input data after it has been embedded into a high-dimensional space by the model.

31. What are the differences between CNN and RNN?

Convolutional Neural Network (CNN)	Recurrent Neural Network (RNN)
Used to handle image data	Best suited to handle sequential data
Fixed input and output data types	Flexible input and output data lengths
Ideal for image and video processing	Ideal for speech and text analysis
Efficient and powerful when compared to RNN	Provides less number of feature sets

32. What is the difference between Type 1 and Type 2 errors?

In simple terms, Type 1 errors refer to the occurrence of a false positive outcome, and Type 2 errors denote the occurrence of a false negative value when performing complex computations.

33. When using TensorFlow, is performance always preferred over accuracy?

No, performance is not always preferred over accuracy when you use TensorFlow. This completely depends on the type of requirement and what the model is trying to achieve. The general rule of thumb is to provide equal weightage to model accuracy and performance.

The next set of TensorFlow interview questions will show the importance of using an example along with concepts to explain.

34. Can you give an example to create a tensor using the constant() function in TensorFlow?

Tensors are most commonly created using the constant() function. The values to be input into the tensor are given as arguments as shown below:

import tensorflow as tf

```
t1 = tf.constant([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
t2 = tf.constant(["String One", "String Two", "String Three"])
sess = tf.Session()
print(t1)
print(sess.run(t1))
print("\n")
print(t2)
print(sess.run(t2))
```

Next up on this top TensorFlow interview questions and answers post, we will take a look at the advanced set of questions.

Advanced Interview Questions

35. What are some of the products that are built using TensorFlow?

There are many products that are built completely using TensorFlow. Some of them are as follows:

- Teachable Machine
- Handwriting Recognition
- Giorgio Cam
- NSynth

36. What is the meaning of Deep Speech?

Deep Speech is a speech-to-text engine that is open-source and uses TensorFlow. It is trained based on Machine Learning techniques and uses a simple syntax to process speech from an input to produce textual output on the other end.

The following syntax can be used to view all of the CLI options for Deep Speech:

```
./deepspeech.py
```

37. What is the use of a histogram dashboard in TensorFlow?

Histogram dashboards are widely used to display complex statistical distributions of a tensor in a simple way. Every histogram chart will have a slice of data that denotes the data that the tensor has at the point of representation.

38. How is audio stored in the audio dashboard?

The audio dashboard serves to primarily help users embed playable widgets stored in files. Tf.summary.audio is used for the storage of these files, and the tagging system is used to embed the latest audio based on the storage policies.

39. What are some of the components needed to deploy a Lite model file?

In TensorFlow, three main components are used to deploy a Lite model:

1. Java API: Used as a wrapper around the C++ API for Android
2. C++ API: Used to load the TensorFlow Lite model and call the interpreter
3. Interpreter: Used to handle kernel loading and the execution of the model

40. What is TensorFlow JS?

TensorFlow JS is a library that gives users the functionality of using browsers to run Machine Learning models. High-level APIs work with JavaScript to support a variety of entities in the backend, such as WebGL, to use a GPU to render functionality (if available). Models can be imported, re-trained, and executed easily by just using a browser.

41. What are activation functions in TensorFlow?

Activation functions are functions applied to the output side of a neural network that serves to be the input of the next layer. It forms a very important part of neural networks as it provides nonlinearity that sets apart a neural network from logistic regression.

42. What is the code that is used to check the version of TensorFlow using Python?

There are two commands depending on the Python version:

Python 2:

```
python -c 'import tensor flow as tf; print(tf.__version__)'
```

Python 3:

```
python3 -c 'import tensor flow as tf; print(tf.__version__)'
```

43. What is model quantization in TensorFlow?

The process of handling the complexity that follows when optimizing inferences can be greatly minimized using TensorFlow. Model quantization is primarily used to reduce the representation of weights and also for the storage and computation of the activation function.

Using model quantization provides users with two main advantages:

- Support for a variety of CPU platforms
- SIMD instruction handling capabilities

44. What is the simple syntax that can be used to convert a NumPy array into a tensor?

There are two ways a NumPy array can be converted into a tensor when working with Python. The first one is as follows:

- **train.shuffle_batch()**

And the second way is:

- **convert_to_tensor(tensor1d, dtype = tf.float64)**

The high-level code offers a good amount of readability and ease-of-use and denoted by the above piece of code.

45. How is the weighted standard error computed in TensorFlow?

The weighted standard error is a standard metric that is used to compute the coefficient of determination when working with a linear regression model.

It provides an easy way to evaluate the model and can be used as shown below:

```
# Used along with TFLearn estimators
weighted_r2 = WeightedR2()
regression = regression(net, metric=weighted_r2)
```

46. What are some of the commonly used optimizers when training a model in TensorFlow?

You can use many optimizers based on various factors, such as the learning rate, performance metric, dropout, gradient, and more.

Following are some of the popular optimizers:

- AdaDelta
- AdaGrad
- Adam
- Momentum
- RMSprop
- Stochastic Gradient Descent

47. What is the use of ArrayFlow and FeedDictFlow in TensorFlow?

ArrayFlow is used to convert array entities into tensors and store them automatically in a queue data structure.

- **data_flow.ArrayFlow()**

FeedDictFlow is used to generate a stream of batch data from the input dataset. The working is based on two queues, where one is used to generate batches and the other is used to load the data and apply preprocessing methods on it.

- **data_flow.FeedDictFlow()**

48. What are some of the parameters to consider when implementing the Word2vec algorithm in TensorFlow?

The Word2vec algorithm is used to compute the vector representations of words from an input dataset.

There are six parameters that have to be considered:

- **embedding_size**: Denotes the dimension of the embedding vector
- **max_vocabulary_size**: Denotes the total number of unique words in the vocabulary
- **min_occurrence**: Removes all words that do not appear at least 'n' number of times
- **skip_window**: Denotes words to considered or not for processing
- **num_skips**: Denotes the number of times you can reuse an input to generate a label
- **num_sampled**: Denotes the number of negative examples to sample from the input

This next one on the top TensorFlow interview questions and answers blog discusses a tricky question, so pay attention!

49. What are some of the important parameters to consider when implementing a random forest algorithm in TensorFlow?

There are six main parameters you should think about and plan when implementing a random forest algorithm in TensorFlow:

- Number of inputs
- Feature count
- Number of samples per batch
- Total number of training steps
- Number of trees
- Maximum number of nodes

50. What are some of the numerical and categorical loss functions supported when working with TensorFlow?

Following are some of the widely used numerical and categorical loss functions supported when working with TensorFlow:

Numerical loss functions:

- L1 loss
- L2 loss
- Pseudo-Huber loss

Categorical loss functions:

- Hinge loss
- Cross-entropy loss
- Sigmoid-entropy loss
- Weighted cross-entropy loss

Each of the loss functions mentioned above has a specific use based on the input data and the type of modeling involved.

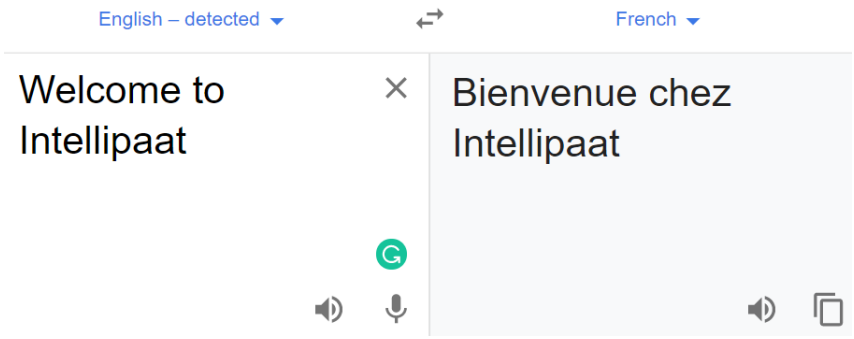
1. What do you understand by Natural Language Processing?

Natural Language Processing is a field of computer science that deals with communication between computer systems and humans. It is a technique used in Artificial Intelligence and Machine Learning. It is used to create automated software that helps understand human spoken languages to extract useful information from the data it gets in the form of audio. Techniques in NLP allow computer systems to process and interpret data in the form of natural languages.

2. List any two real-life applications of Natural Language Processing.

Two real-life applications of Natural Language Processing are as follows:

1. **Google Translate**: Google Translate is one of the famous applications of Natural Language Processing. It helps convert written or spoken sentences into any language. Also, we can find the correct pronunciation and meaning of a word by using Google Translate. It uses advanced techniques of Natural Language Processing to achieve success in translating sentences into various languages.



2. **Chatbots**: To provide a better customer support service, companies have started using chatbots for 24/7 service. Chatbots helps resolve the basic queries of customers. If a chatbot is not able to resolve any query, then it forwards it to the support team, while still engaging the customer. It helps make customers feel that the customer support team is quickly attending them. With the help of chatbots, companies have become capable of building cordial relations with customers. It is only possible with the help of Natural Language Processing.

3. What are stop words?

Stop words are said to be useless data for a search engine. Words such as articles, prepositions, etc. are considered as stop words. There are stop words such as was, were, is, am, the, a, an, how, why, and many more. In Natural Language Processing, we eliminate the stop words to understand and analyze the meaning of a sentence. The removal of stop words is one of the most important tasks for search engines. Engineers design the algorithms of search engines in such a way that they ignore the use of stop words. This helps show the relevant search result for a query.

4. What is NLTK?

NLTK is a Python library, which stands for Natural Language Toolkit. We use NLTK to process data in human spoken languages. NLTK allows us to apply techniques such as parsing, tokenization, lemmatization, stemming, and more to understand natural languages. It helps in categorizing text, parsing linguistic structure, analyzing documents, etc.

A few of the libraries of the NLTK package that we often use in NLP are:

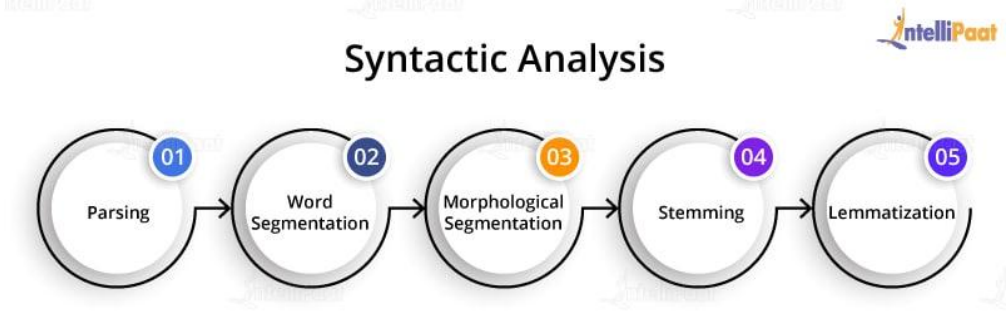
1. SequentialBackoffTagger
2. DefaultTagger
3. UnigramTagger
4. treebank
5. wordnet
6. FreqDist
7. patterns
8. RegexpTagger
9. backoff_tagger
10. UnigramTagger, BigramTagger, and TrigramTagger

---

5. What is Syntactic Analysis?

Syntactic analysis is a technique of analyzing sentences to extract meaning from it. Using syntactic analysis, a machine can analyze and understand the order of words arranged in a sentence. NLP employs grammar rules of a language that helps in the syntactic analysis of the combination and order of words in documents.

The techniques used for syntactic analysis are as follows:

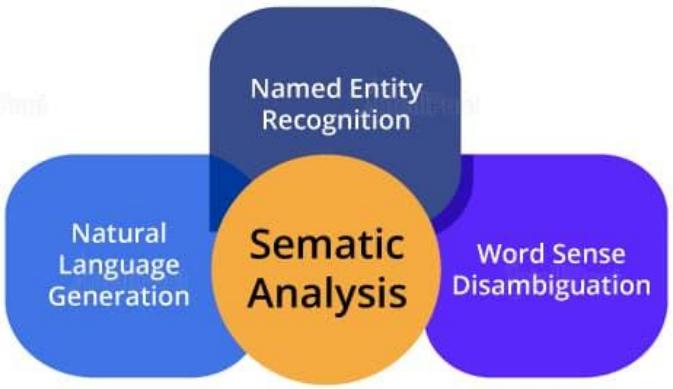


1. **Parsing**: It helps in deciding the structure of a sentence or text in a document. It helps analyze the words in the text based on the grammar of the language.
2. **Word segmentation**: The segmentation of words segregates the text into small significant units.
3. **Morphological segmentation**: The purpose of morphological segmentation is to break words into their base form.
4. **Stemming**: It is the process of removing the suffix from a word to obtain its root word.
5. **Lemmatization**: It helps combine words using suffixes, without altering the meaning of the word.

6. What is Semantic Analysis?

Semantic analysis helps make a machine understand the meaning of a text. It uses various algorithms for the interpretation of words in sentences. It also helps understand the structure of a sentence.

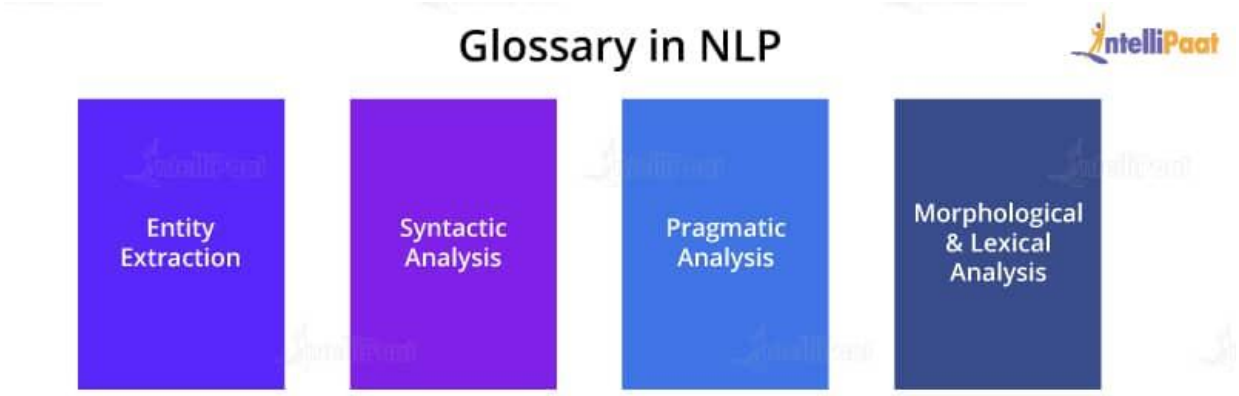
Techniques used for semantic analysis are as given below:



1. **Named entity recognition:** This is the process of information retrieval that helps identify entities such as the name of a person, organization, place, time, emotion, etc.
2. **Word sense disambiguation:** It helps identify the sense of a word used in different sentences.
3. **Natural language generation**: It is a process used by the software to convert the structured data into human spoken languages. By using NLG, organizations can automate content for custom reports.

7. List the components of Natural Language Processing.

The major components of NLP are as follows:



- **Entity extraction:** Entity extraction refers to the retrieval of information such as place, person, organization, etc. by the segmentation of a sentence. It helps in the recognition of an entity in a text.
- **Syntactic analysis:** Syntactic analysis helps draw the specific meaning of a text.
- **Pragmatic analysis:** To find useful information from a text, we implement pragmatic analysis techniques.
- **Morphological and lexical analysis:** It helps in explaining the structure of words by analyzing them through parsing.

8. What is Latent Semantic Indexing (LSI)?

Latent semantic indexing is a mathematical technique used to improve the accuracy of the information retrieval process. The design of LSI algorithms allows machines to detect the hidden (latent) correlation between semantics (words). To enhance information understanding, machines generate various concepts that associate with the words of a sentence.

The technique used for information understanding is called singular value decomposition. It is generally used to handle static and unstructured data. The matrix obtained for singular value decomposition contains rows for words and columns for documents. This method best suits to identify components and group them according to their types.

The main principle behind LSI is that words carry a similar meaning when used in a similar context. Computational LSI models are slow in comparison to other models. However, they are good at contextual awareness that helps improve the analysis and understanding of a text or a document.

9. What are Regular Expressions?

A regular expression is used to match and tag words. It consists of a series of characters for matching strings.

Suppose, if A and B are regular expressions, then the following are true for them:

- If {ε} is a regular language, then ε is a regular expression for it.
- If A and B are regular expressions, then A + B is also a regular expression within the language {A, B}.
- If A and B are regular expressions, then the concatenation of A and B (A.B) is a regular expression.
- If A is a regular expression, then A* (A occurring multiple times) is also a regular expression.

10. What is Regular Grammar?

Regular grammar is used to represent a regular language.

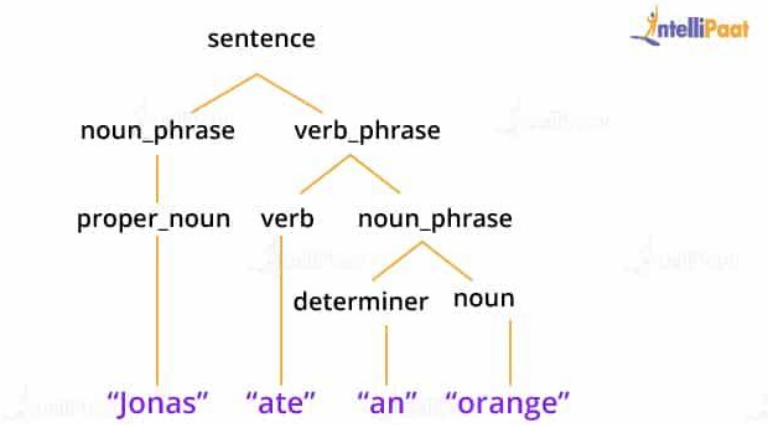
A regular grammar comprises rules in the form of **A -> a**, **A -> aB**, and many more. The rules help detect and analyze strings by automated computation.

Regular grammar consists of four tuples:

1. 'N' is used to represent the non-terminal set.
2. '∑' represents the set of terminals.
3. 'P' stands for the set of productions.
4. 'S € N' denotes the start of non-terminal.

11. What is Parsing in the context of NLP?

Parsing in NLP refers to the understanding of a sentence and its grammatical structure by a machine. Parsing allows the machine to understand the meaning of a word in a sentence and the grouping of words, phrases, nouns, subjects, and objects in a sentence. Parsing helps analyze the text or the document to extract useful insights from it. To understand parsing, refer to the below diagram:



In this, 'Jonas ate an orange' is parsed to understand the structure of the sentence.

Intermediate NLP Interview Questions

12. What is TF-IDF?

TFIDF or Term Frequency-Inverse Document Frequency indicates the importance of a word in a set. It helps in information retrieval with numerical statistics. For a specific document, TF-IDF shows a frequency that helps identify the keywords in a document. The major use of TF-IDF in NLP is the extraction of useful information from crucial documents by statistical data. It is ideally used to classify and summarize the text in documents and filter out stop words.

TF helps calculate the ratio of the frequency of a term in a document and the total number of terms. Whereas, **IDF** denotes the importance of the term in a document.

The formula for calculating TF-IDF:

TF(W) = (Frequency of W in a document)/(The total number of terms in the document)

IDF(W) = log_e(The total number of documents/The number of documents having the term W)

When **TF*IDF** is high, the frequency of the term is less and vice versa.

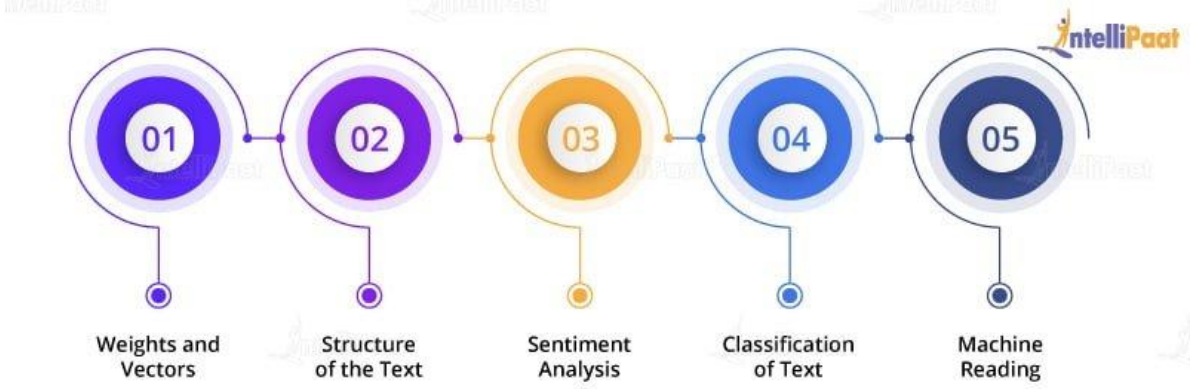
Google uses TF-IDF to decide the index of search results according to the relevancy of pages. The design of the TF-IDF algorithm helps optimize the search results in Google. It helps quality content rank up in search results.

If you want to know more about 'What is Natural Language Processing?' you can go through this Natural Language Processing Using Python course!

13. Define the terminology in NLP.

This is one of the most often asked NLP interview questions.

The interpretation of Natural Language Processing depends on various factors, and they are:



Weights and Vectors

- Use of TF-IDF for information retrieval
- Length (TF-IDF and doc)
- Google Word Vectors
- Word Vectors

Structure of the Text

- POS tagging
- Head of the sentence
- Named Entity Recognition (NER)

Sentiment Analysis

- Knowledge of the characteristics of sentiment
- Knowledge about entities and the common dictionary available for sentiment analysis

Classification of Text

- Supervised learning algorithm

- Training set
- Validation set
- Test set
- Features of the text
- LDA

Machine Reading

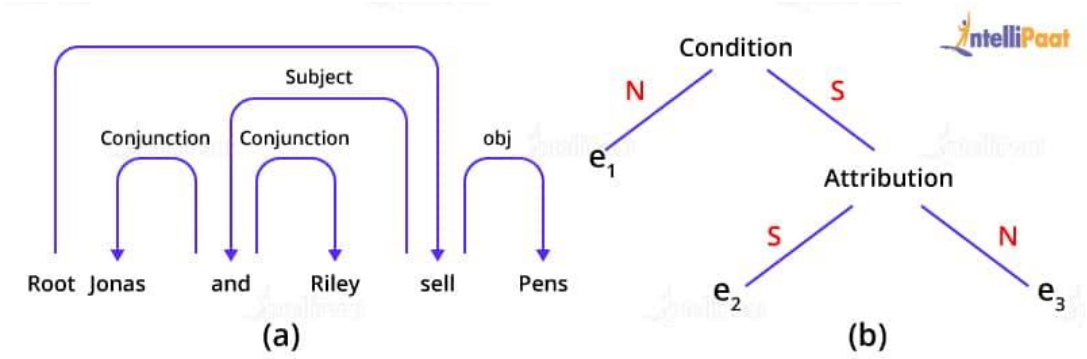
- Removal of possible entities
- Joining with other entities
- DBpedia

14. Explain Dependency Parsing in NLP.

Dependency parsing helps assign a syntactic structure to a sentence. Therefore, it is also called syntactic parsing. Dependency parsing is one of the critical tasks in NLP. It allows the analysis of a sentence using parsing algorithms. Also, by using the parse tree in dependency parsing, we can check the grammar and analyze the semantic structure of a sentence.

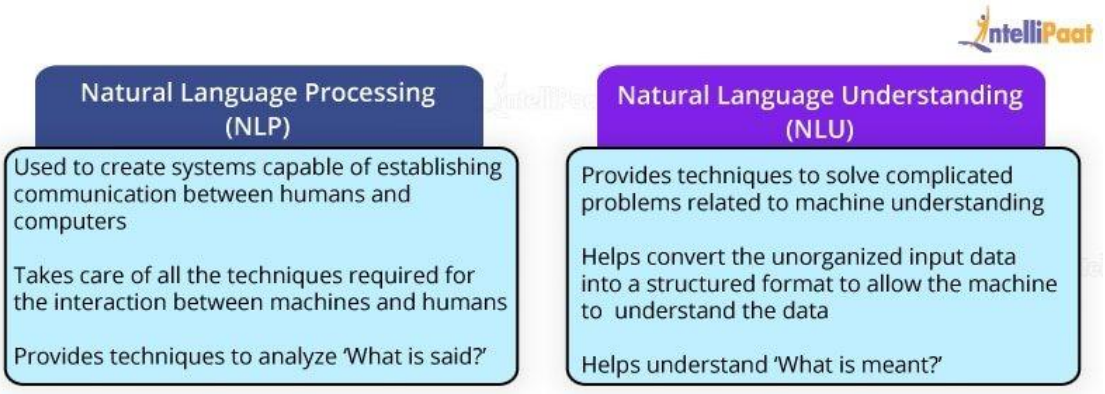
For implementing dependency parsing, we use the spacy package. It implements token properties to operate the dependency parse tree.

The below diagram shows the dependency parse tree:



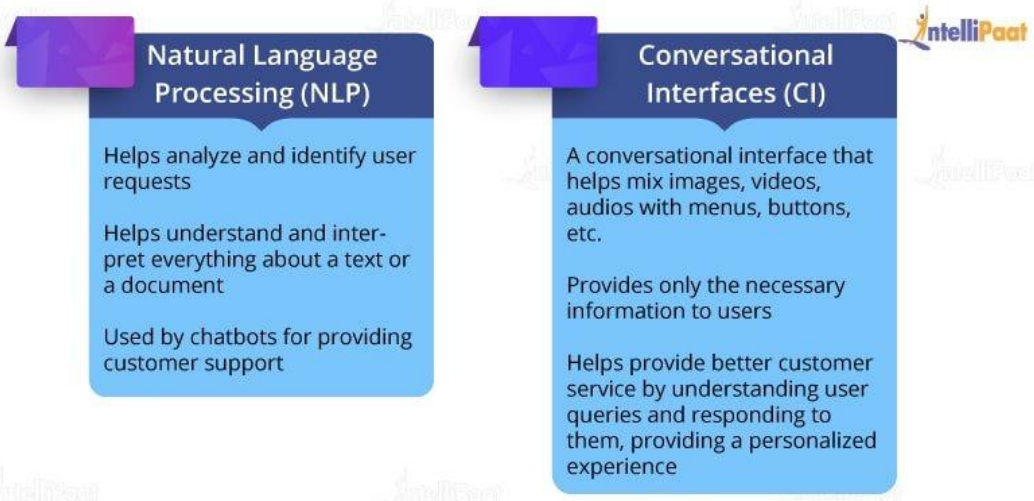
15. What is the difference between NLP and NLU?

The below table shows the difference between NLP and NLU:



16. What is the difference between NLP and CI?

The below table shows the difference between NLP and CI:



17. What is Pragmatic Analysis?

Pragmatic analysis is an important task in NLP for interpreting knowledge that is lying outside a given document. The aim of implementing pragmatic analysis is to focus on exploring a different aspect of the document or text in a language. This requires a comprehensive knowledge of the real world. The pragmatic analysis allows software applications for the critical interpretation of the real-world data to know the actual meaning of sentences and words.

Example:

Consider this sentence: 'Do you know what time it is?'

This sentence can either be asked for knowing the time or for yelling at someone to make them note the time. This depends on the context in which we use the sentence.

18. What is Pragmatic Ambiguity?

Pragmatic ambiguity refers to the multiple descriptions of a word or a sentence. An ambiguity arises when the meaning of the sentence is not clear. The words of the sentence may have different meanings. Therefore, in practical situations, it becomes a challenging task for a machine to understand the meaning of a sentence. This leads to pragmatic ambiguity.

Example:

Check out the below sentence.

'Are you feeling hungry?'

The given sentence could be either a question or a formal way of offering food.

19. What are unigrams, bigrams, trigrams, and n-grams in NLP?

When we parse a sentence one word at a time, then it is called a unigram. The sentence parsed two words at a time is a bigram.

When the sentence is parsed three words at a time, then it is a trigram. Similarly, n-gram refers to the parsing of *n* words at a time.

Example: To understand unigrams, bigrams, and trigrams, you can refer to the below diagram:

Therefore, parsing allows machines to understand the individual meaning of a word in a sentence. Also, this type of parsing helps predict the next word and correct spelling errors.

Are you interested in learning Artificial Intelligence from experts? Enroll in our AI Course in Bangalore now!

20. What are the steps involved in solving an NLP problem?

Below are the steps involved in solving an NLP problem:

1. Gather the text from the available dataset or by web scraping
2. Apply stemming and lemmatization for text cleaning
3. Apply feature engineering techniques
4. Embed using **word2vec**
5. Train the built model using neural networks or other Machine Learning techniques
6. Evaluate the model's performance
7. Make appropriate changes in the model
8. Deploy the model

21. What is Feature Extraction in NLP?

Features or characteristics of a word help in text or document analysis. They also help in sentiment analysis of a text. Feature extraction is one of the techniques that are used by recommendation systems. Reviews such as 'excellent,' 'good,' or 'great' for a movie are positive reviews, recognized by a recommender system. The recommender system also tries to identify the features of the text that help in describing the context of a word or a sentence. Then, it makes a group or category of the words that have some common characteristics. Now, whenever a new word arrives, the system categorizes it as per the labels of such groups.

22. What is precision and recall?

The metrics used to test an NLP model are precision, recall, and F1. Also, we use accuracy for evaluating the model's performance. The ratio of prediction and the desired output yields the accuracy of the model.

Precision is the ratio of true positive instances and the total number of positively predicted instances.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

Recall is the ratio of true positive instances and the total actual positive instances.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

23. What is F1 score in NLP?

F1 score evaluates the weighted average of recall and precision. It considers both false negative and false positive instances while evaluating the model. F1 score is more accountable than accuracy for an NLP model when there is an uneven distribution of class. Let us look at the formula for calculating F1 score:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

If you are preparing for an Artificial Intelligence job interview, go through this top Artificial Intelligence Interview Questions and Answers!

Advanced NLP Interview Questions

24. How to tokenize a sentence using the nltk package?

Tokenization is a process used in NLP to split a sentence into tokens. **Sentence tokenization** refers to splitting a text or paragraph into sentences.

For tokenizing, we will import **sent_tokenize** from the **nltk package**:

```
from nltk.tokenize import sent_tokenize<>
```

We will use the below paragraph for sentence tokenization:
Para = “Hi Guys. Welcome to Intellipaat. This is a blog on the NLP interview questions and answers.”

```
sent_tokenize(Para)
```

Output:

```
[ 'Hi Guys.' ,
'Welcome to Intellipaat. ',
'This is a blog on the NLP interview questions and answers. ' ]
```

Tokenizing a word refers to splitting a sentence into words.

Now, to tokenize a word, we will import **word_tokenize** from the nltk package.

```
from nltk.tokenize import word_tokenize
```

Para = “Hi Guys. Welcome to Intellipaat. This is a blog on the NLP interview questions and answers.”

```
word_tokenize(Para)
```

Output:

```
[ 'Hi' , 'Guys' , ' . ' , 'Welcome' , 'to' , 'Intellipaat' , ' . ' , 'This' , 'is' ,  'a',
'blog' , 'on' , 'the' , 'NLP' , 'interview' , 'questions' , 'and' , 'answers' , ' . ' ]
```

25. Explain how we can do parsing.

Parsing is the method to identify and understand the syntactic structure of a text. It is done by analyzing the individual elements of the text. The machine parses the text one word at a time, then two at a time, further three, and so on.

- When the machine parses the text one word at a time, then it is a **unigram**.
- When the text is parsed two words at a time, it is a **bigram**.
- The set of words is a **trigram** when the machine parses three words at a time.

Look at the below diagram to understand unigram, bigram, and trigram.

Now, let's implement parsing with the help of the **nltk** package.

```
import nltk
text = ”Top 30 NLP interview questions and answers”
```

We will now tokenize the text using **word_tokenize**.

```
text_token= word_tokenize(text)
```

Now, we will use the function for extracting unigrams, bigrams, and trigrams.

```
list(nltk.unigrams(text))
```

Output:

```
[ "Top 30 NLP interview questions and answer"]
list(nltk.bigrams(text))
```

Output:

```
["Top 30", "30 NLP", "NLP interview", "interview questions",   "questions and", "and answer"]
list(nltk.trigrams(text))
```

Output:

```
["Top 30 NLP", "NLP interview questions", "questions and answers"]
```

For extracting **n-grams**, we can use the function **nltk.ngrams** and give the argument *n* for the number of parsers.

```
list(nltk.ngrams(text,n))
```

26. Explain Stemming with the help of an example.

In Natural Language Processing, stemming is the method to extract the root word by removing suffixes and prefixes from a word.
For example, we can reduce 'stemming' to 'stem' by removing 'm' and 'ing.'
We use various algorithms for implementing stemming, and one of them is PorterStemmer.
First, we will import **PorterStemmer** from the nltk package.

```
from nltk.stem import PorterStemmer
```

Creating an object for PorterStemmer

```
pst=PorterStemmer()
pst.stem(“running”), pst.stem(“cookies”), pst.stem(“flying”)
```

Output:

```
(‘run’, ‘cooki', ‘fly’ )
```

27. Explain Lemmatization with the help of an example.

We use stemming and lemmatization to extract root words. However, stemming may not give the actual word, whereas lemmatization generates a meaningful word.
In lemmatization, rather than just removing the suffix and the prefix, the process tries to find out the root word with its proper meaning.
Example: 'Bricks' becomes 'brick,' 'corpora' becomes 'corpus,' etc.
Let's implement lemmatization with the help of some nltk packages.
First, we will import the required packages.

```
from nltk.stem import wordnet
from nltk.stem import WordnetLemmatizer
```

Creating an object for WordnetLemmatizer()

```
lemma= WordnetLemmatizer()
list = ["Dogs", "Corpora", "Studies"]
for n in list:
print(n + ":" + lemma.lemmatize(n))
```

Output:

```
Dogs: Dog
Corpora: Corpus
Studies: Study
```

Interested in learning Artificial Intelligence? Go through this Artificial Intelligence Tutorial!

28. What is Parts-of-speech Tagging?

The parts-of-speech (POS) tagging is used to assign tags to words such as nouns, adjectives, verbs, and more. The software uses the POS tagging to first read the text and then differentiate the words by tagging. The software uses algorithms for the parts-of-speech tagging. POS tagging is one of the most essential tools in Natural Language Processing. It helps in making the machine understand the meaning of a sentence.
We will look at the implementation of the POS tagging using stop words.
Let's import the required nltk packages.

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stop_words = set(stopwords.words('english'))
txt = "Sourav, Pratyush, and Abhinav are good friends."
```

Tokenizing using sent_tokenize

```
tokenized_text = sent_tokenize(txt)
```

To find punctuation and words in a string, we will use **word_tokenizer** and then remove the stop words.

```
for n in tokenized_text:
wordsList = nltk.word_tokenize(i)
wordsList = [w for w in wordsList if not w instop_words]
```

Now, we will use the POS tagger.

```
tagged_words = nltk.pos_tag(wordsList)
```

```
print(tagged_words)
```

Output:

```
[('Sourav', 'NNP'), ('Pratyush', 'NNP'), ('Abhinav', 'NNP'), ('good',  'JJ'), ('friends', 'NNS')]
```

29. Explain Named Entity Recognition by implementing it.

Named Entity Recognition (NER) is an information retrieval process. NER helps classify named entities such as monetary figures, location, things, people, time, and more. It allows the software to analyze and understand the meaning of the text. NER is mostly used in NLP, Artificial Intelligence, and Machine Learning. One of the real-life applications of NER is chatbots used for customer support.
Let's implement NER using the **spacy** package.
Importing the spacy package:

```
import spacy
nlp = spacy.load('en_core_web_sm')
Text = "The head office of Google is in California"
document = nlp(text)for ent in document.ents:
print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

Output:

```
Office 9 15 Place
Google 19 25 ORG
California 32 41 GPE
```

30. How to check word similarity using the spacy package?

To find out the similarity among words, we use word similarity. We evaluate the similarity with the help of a number that lies between 0 and 1. We use the spacy library to implement the technique of word similarity.

```
import spacy
nlp = spacy.load('en_core_web_md')
print("Enter the words")
input_words = input()
tokens = nlp(input_words)
for i in tokens:
print(i.text, i.has_vector, i.vector_norm, i.is_oov)
token_1, token_2 = tokens[0], tokens[1]
print("Similarity between words:", token_1.similarity(token_2))
```

Output:

```
hot  True 5.6898586 False
cold True6.5396233 False
Similarity: 0.597265
```

This means that the similarity between the words 'hot' and 'cold' is just 59 percent.

1. What are the types of Machine Learning?

In all the ML Interview Questions that we would be going to discuss, this is one of the most basic questions.

So, basically, there are three types of Machine Learning techniques:

Supervised Learning: In this type of Machine Learning technique, machines learn under the supervision of labeled data. There is a training dataset on which the machine is trained, and it gives the output according to its training.

Unsupervised Learning: Unlike supervised learning, it has unlabeled data. So, there is no supervision under which it works on the data. Basically, unsupervised learning tries to identify patterns in data and make clusters of similar entities. After that, when a new input data is fed into the model, it does not identify the entity; rather, it puts the entity in a cluster of similar objects.

Reinforcement Learning: Reinforcement learning includes models that learn and traverse to find the best possible move. The algorithms for reinforcement learning are constructed in a way that they try to find the best possible suite of action on the basis of the reward and punishment theory.

2. Differentiate between classification and regression in Machine Learning.

In Machine Learning, there are various types of prediction problems based on supervised and unsupervised learning. These are classification, regression, clustering, and association. Here, we will discuss classification and regression.

Classification: In classification, we try to create a Machine Learning model that assists us in differentiating data into separate categories. The data is labeled and categorized based on the input parameters.

For example, imagine that we want to make predictions on the churning out customers for a particular product based on some data recorded. Either the customers will churn out or they will not. So, the labels for this would be 'Yes' and 'No.'

Regression: It is the process of creating a model for distinguishing data into continuous real values, instead of using classes or discrete values. It can also identify the distribution movement depending on the historical data. It is used for predicting the occurrence of an event depending on the degree of association of variables.

For example, the prediction of weather conditions depends on factors such as temperature, air pressure, solar radiation, the elevation of the area, and distance from the sea. The relation between these factors assists us in predicting the weather condition.

3. What is a Linear Regression in Machine Learning?

Linear Regression is a supervised Machine Learning algorithm. It is used to find the linear relationship between the dependent and the independent variables for predictive analysis.

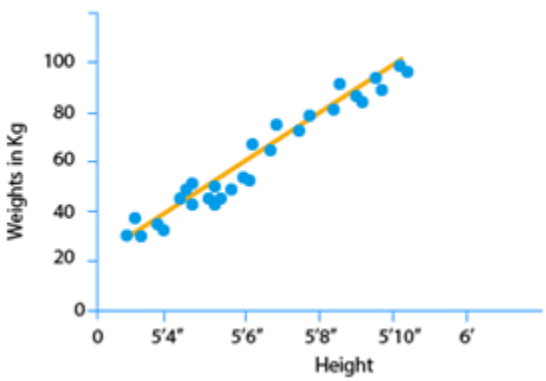
The equation for Linear Regression:

$$Y = A + B.X$$

where:

- ***X*** is the input or the independent variable
- ***Y*** is the output or the dependent variable
- ***a*** is the intercept and ***b*** is the coefficient of *X*

Below is the **best fit line** that shows the data of weight (***Y*** or the dependent variable) and height (***X*** or the independent variable) of 21-years-old candidates scattered over the plot. This straight-line shows the best linear relationship that would help in predicting the weight of candidates according to their height.



To get this **best fit line**, we will try to find the best values of ***a*** and ***b***. By adjusting the values of *a* and *b*, we will try to reduce errors in the prediction of *Y*.

This is how linear regression helps in finding the linear relationship and predicting the output.

4. How will you determine the Machine Learning algorithm that is suitable for your problem?

To identify the Machine Learning algorithm for our problem, we should follow the below steps:

Step 1: Problem Classification: Classification of the problem depends on the classification of input and output:

- **Classifying the input:** Classification of the input depends on whether we have data labeled (supervised learning) or unlabeled (unsupervised learning), or whether we have to create a model that interacts with the environment and improves itself (reinforcement learning).
- **Classifying the output:** If we want the output of our model as a class, then we need to use some classification techniques.

If it is giving the output as a number, then we must use regression techniques and, if the output is a different cluster of inputs, then we should use clustering techniques.

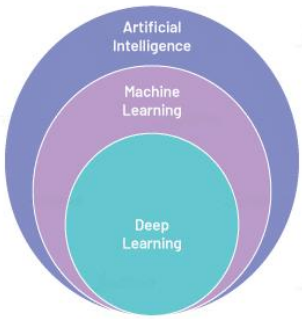
Step 2: Checking the algorithms in hand: After classifying the problem, we have to look for the available algorithms that can be deployed for solving the classified problem.

Step 3: Implementing the algorithms: If there are multiple algorithms available, then we will implement each one of them, one by one. Finally, we would select the algorithm that gives the best performance.

5. Explain Machine Learning, Artificial Intelligence, and Deep Learning?

It is common to get confused between the three in-demand technologies: Machine Learning, Artificial Intelligence, and Deep Learning. These three technologies, though a little different from one another, are interrelated. While Deep Learning is a subset of Machine Learning, Machine Learning is a subset of Artificial Intelligence. Since some terms

and techniques may overlap with each other while dealing with these technologies, it is easy to get confused between them.



Therefore, let's learn about these technologies in detail so that you become capable of differentiating between them:

- **Machine Learning:** Machine Learning involves various statistical and Deep Learning techniques that allow machines to use their past experiences and get better at performing specific tasks without having to be monitored.
- **Artificial Intelligence:** Artificial Intelligence uses numerous Machine Learning and Deep Learning techniques that enable computer systems to perform tasks using human intelligence, with logic and rules.
- **Deep Learning:** Deep Learning comprises several algorithms that enable software to learn from themselves and perform various business tasks, including image and speech recognition. This is possible when the systems expose their multi-layered neural networks to large volumes of data for learning.

6. What is clustering in Machine Learning?

Clustering is a technique used in unsupervised learning that involves grouping data points. If you have a set of data points, you can make use of the clustering algorithm. This technique will allow you to classify all the data points into their particular groups. The data points that are thrown into the same category have similar features and properties, whereas the data points that belong to different groups have distinct features and properties. This method allows you to perform statistical data analysis. Let's take a look at three of the most popular and useful clustering algorithms.

- **K-means clustering:** This algorithm is commonly used when you have data with no specific group or category. It allows you to find the hidden patterns in the data that can be used to classify them into various groups. The variable *k* is used to represent the number of groups they are divided into, and the data points are clustered using the similarity of features. Here, the centroids of the clusters are used for labeling new data.
- **Mean-shift clustering:** The main aim of this algorithm is to update the center point candidates to be the mean and find the center points of all the groups. Unlike k-means clustering, in this, you do not need to select the possible number of clusters as it can automatically be discovered by the mean shift.
- **Density-based spatial clustering of applications with noise (DBSCAN):** This clustering is based on density and has similarities with mean-shift clustering. There is no need to pre-set the number of clusters, but unlike mean-shift, it identifies outliers and treats them like noise. Moreover, it can identify arbitrarily sized and shaped clusters without much effort.

7. What is a hypothesis in Machine Learning?

Machine Learning allows you to use the dataset available to understand a specific function that maps inputs to outputs in the best possible way. This problem is known as function approximation. In this, you need to use an approximation for the unknown target function that maps in the best manner all the plausible observations based on the given problem. Hypothesis in Machine learning is a model that helps in approximating the target function and performing the necessary input-to-output mappings. The choice and configuration of algorithms allows you to define the space of plausible hypotheses that may be represented by the model.

In the hypothesis, lowercase h (h) is used for a specific hypothesis, while uppercase h (H) is used for the hypothesis space that is being searched. Let's briefly understand these notations:

- **Hypothesis (h):** A hypothesis is a specific model that helps in mapping inputs to outputs, which can further be used for evaluation and prediction.
- **Hypothesis set (H):** Hypothesis set consists of a space of hypotheses that can be used to map inputs to outputs, which can be searched. The general constraints include the choice of problem framing, the model, and the model configuration.

8. What are the differences between Deep Learning and Machine Learning?

- **Deep Learning:** Deep Learning allows machines to make various business-related decisions using artificial neural networks, which is one of the reasons why it needs a vast amount of data for training. Since there is a lot of computing power required, it requires high-end systems as well. The systems acquire various properties and features with the help of the given data, and the problem is solved using an end-to-end method.
- **Machine Learning:** Machine Learning technology gives machines the ability to make business decisions without any external help, using the knowledge gained from past data. Machine Learning systems require relatively small amounts of data to train themselves, and most of the features need to be manually coded and understood in advance. Here, the given business problem is dissected into two, and they are solved individually. Once the solutions of both parts have been acquired, they are then combined.

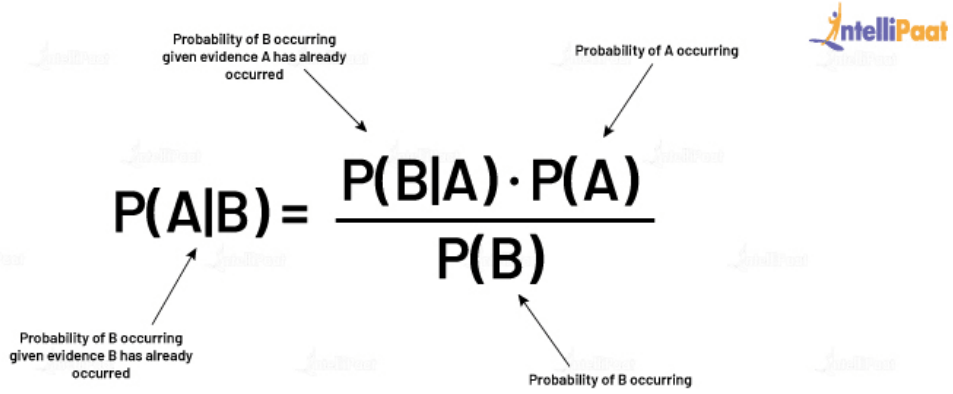
9. What are the differences between Supervised and Unsupervised Machine Learning?

- **Supervised learning:** Algorithms of supervised learning use labeled data to get trained. The models take direct feedback to confirm whether the output that is being predicted is, indeed, correct. Moreover, both the input data and the output data are provided to the model, and the main aim here is to train the model to predict the output when it receives new data. It can largely be divided into two parts, classification and regression. It offers accurate results.
- **Unsupervised learning:** Unsupervised learning algorithms use unlabeled data for training purposes. In this, the models do not take any feedback, and unlike the case of supervised learning, these models identify hidden data trends. The unsupervised learning model is only provided with the input data, and its main aim is to identify hidden patterns to extract information from the unknown sets of data. It can also be classified into two parts, namely, clustering and associations. Unfortunately, unsupervised learning offers results that are comparatively less accurate.

10. What is Bayes' theorem in Machine Learning?

The Bayes' theorem offers the probability of any given event to occur using prior knowledge. In mathematical terms, it can be defined as the true positive rate of the given sample condition divided by the sum of the true positive rate of the said condition and the false positive rate of the entire population.

Two of the most significant applications of the Bayes' theorem in Machine Learning are Bayesian optimization and Bayesian belief networks. This theorem is also the foundation behind the Machine Learning brand that involves the Naive Bayes classifier.



11. What is cross-validation in Machine Learning?

The cross-validation method in Machine Learning allows a system to increase the performance of the given Machine Learning algorithm to which you feed multiple sample data from the dataset. This sampling process is done to break the dataset into smaller parts that have the same number of rows, out of which a random part is selected as a test set, and the rest of the parts are kept as train sets. It consists of the following techniques:

- Holdout method
- K-fold cross-validation
- Stratified k-fold cross-validation
- Leave p-out cross-validation

12. What is entropy in Machine Learning?

Entropy in Machine Learning measures the randomness in the data that needs to be processed. The more entropy in the given data, the more difficult it becomes to draw any useful conclusion from the data. For example, let's take the incident of flipping a coin. The result of this is random as it does not favor heads or tails. Here, the result for any number of tosses cannot be predicted easily as there is no definite relationship between the action of flipping and the possible outcomes.

Courses you may like

13. What is epoch in Machine Learning?

Epoch in Machine Learning is used to indicate the count of passes in a given training dataset where the Machine Learning algorithm has done its job. Generally, when there is a huge chunk of data, it is grouped into several batches.

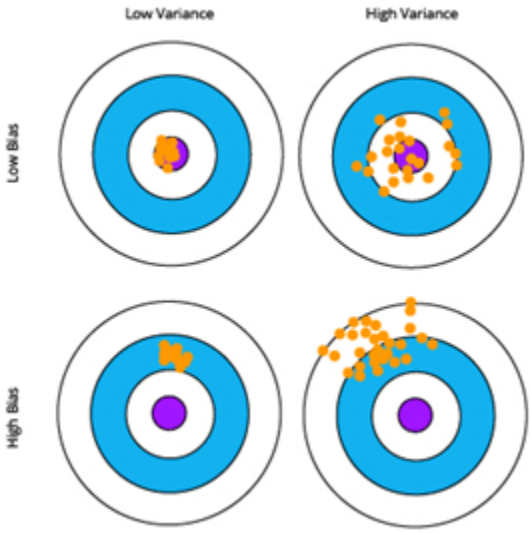
Here, each of these batches goes through the given model, and this process is referred to as iteration. Now, if the batch size comprises the complete training dataset, then the count of iterations is the same as that of epochs.

In case there is more than one batch, d*e=i*b is the formula used, wherein ‘d’ is the dataset, ‘e’ is the number of epochs, ‘i’ is the number of iterations, and ‘b’ is the batch size.

14. What are Bias and Variance in Machine Learning?

- **Bias** is the difference between the average prediction of our model and the correct value. If the bias value is high, then the prediction of the model is not accurate. Hence, the bias value should be as low as possible to make the desired predictions.
- **Variance** is the number that gives the difference of prediction over a training set and the anticipated value of other training sets. High variance may lead to large fluctuation in the output. Therefore, the model’s output should have low variance.

The below diagram shows the bias–variance trade off:



Here, the desired result is the blue circle at the center. If we get off from the blue section, then the prediction goes wrong.

5. What is Variance Inflation Factor?

Variance Inflation Factor (VIF) is the estimate of the volume of multicollinearity in a collection of many regression variables.

VIF = Variance of the model / Variance of the model with a single independent variable

We have to calculate this ratio for every independent variable. If VIF is high, then it shows the high collinearity of the independent variables.

16. Explain false negative, false positive, true negative, and true positive with a simple example.

True Positive (TP): When the Machine Learning model **correctly** predicts the condition, it is said to have a True Positive value.

True Negative (TN): When the Machine Learning model **correctly** predicts the negative condition or class, then it is said to have a True Negative value.

False Positive (FP): When the Machine Learning model **incorrectly** predicts a negative class or condition, then it is said to have a False Positive value.

False Negative (FN): When the Machine Learning model **incorrectly** predicts a positive class or condition, then it is said to have a False Negative value.

17. What is a Confusion Matrix?

Confusion matrix is used to explain a model's performance and gives the summary of predictions on the classification problems. It assists in identifying the uncertainty between classes.

A confusion matrix gives the count of correct and incorrect values and also the error types.**Accuracy of the model:**

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

For example, consider this confusion matrix. It consists of values as True Positive, True Negative, False Positive, and False Negative for a classification model. Now, the accuracy of the model can be calculated as follows:

	Predicted: No	Predicted: Yes	
Actual: No	TP = 200	FN = 60	260
Actual: Yes	FP = 10	TN = 50	60
	210	110	

Thus, in our example:

```
Accuracy = (200 + 50) / (200 + 50 + 10 + 60) = 0.78
```

This means that the model's accuracy is 0.78, corresponding to its True Positive, True Negative, False Positive, and False Negative values.

18. What do you understand by Type I and Type II errors?

Type I Error: Type I error (False Positive) is an error where the outcome of a test shows the non-acceptance of a true condition.

For example, a cricket match is going on and, when a batsman is not out, the umpire declares that he is out. This is a false positive condition. Here, the test does not accept the true condition that the batsman is not out.

Type II Error: Type II error (False Negative) is an error where the outcome of a test shows the acceptance of a false condition.

For example, the CT scan of a person shows that he is not having a disease but, in reality, he is having it. Here, the test accepts the false condition that the person is not having the disease.

19. When should you use classification over regression?

Both classification and regression are associated with prediction. Classification involves the identification of values or entities that lie in a specific group. The regression method, on the other hand, entails predicting a response value from a consecutive set of outcomes.

The classification method is chosen over regression when the output of the model needs to yield the belongingness of data points in a dataset to a particular category.

For example, we have some names of bikes and cars. We would not be interested in finding how these names are correlated to bikes and cars. Rather, we would check whether each name belongs to the bike category or to the car category.

20. Explain Logistic Regression.

Logistic regression is the proper regression analysis used when the dependent variable is categorical or binary. Like all regression analyses, logistic regression is a technique for predictive analysis. Logistic regression is used to explain data and the relationship between one dependent binary variable and one or more independent variables. Also, it is employed to predict the probability of a categorical dependent variable.

We can use logistic regression in the following scenarios:

- To predict whether a citizen is a Senior Citizen (1) or not (0)
- To check whether a person is having a disease (Yes) or not (No)

There are three types of logistic regression:

- **Binary Logistic Regression**: In this, there are only two outcomes possible.

Example: To predict whether it will rain (1) or not (0)

- **Multinomial Logistic Regression**: In this, the output consists of three or more unordered categories.

Example: Prediction on the regional languages (Kannada, Telugu, Marathi, etc.)

- **Ordinal Logistic Regression**: In ordinal logistic regression, the output consists of three or more ordered categories.

Example: Rating an Android application from 1 to 5 stars.

Interested in learning Machine Learning? Click here to learn more in this Machine Learning Training in Bangalore!

21. Imagine, you are given a dataset consisting of variables having more than 30% missing values. Let's say, out of 50 variables, 8 variables have missing values, which is higher than 30%. How will you deal with them?

To deal with the missing values, we will do the following:

- We will specify a different class for the missing values.
- Now, we will check the distribution of values, and we would hold those missing values that are defining a pattern.
- Then, we will charge these into a yet another class, while eliminating others.

22. How do you handle the missing or corrupted data in a dataset?

In Python Pandas, there are two methods that are very useful. We can use these two methods to locate the lost or corrupted data and discard those values:

- **isNull()**: For detecting the missing values, we can use the isNull() method.
- **dropna()**: For removing the columns/rows with null values, we can use the dropna() method.

Also, we can use **fillna()** to fill the void values with a placeholder value.

23. What is PCA in Machine Learning?

Firstly, this is one of the most important Machine Learning Interview Questions.

In the real world, we deal with multi-dimensional data. Thus, data visualization and computation become more challenging with the increase in dimensions. In such a scenario, we might have to reduce the dimensions to analyze and visualize the data easily. We do this by:

- Removing irrelevant dimensions
- Keeping only the most relevant dimensions

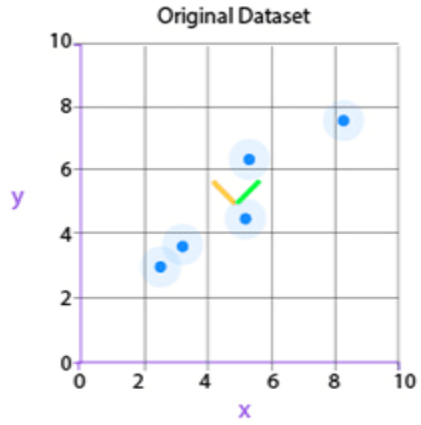
This is where we use Principal Component Analysis (PCA).

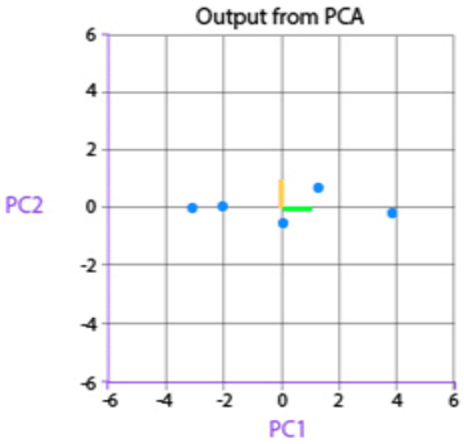
Finding a fresh collection of uncorrelated dimensions (orthogonal) and ranking them on the basis of variance are the goals of Principal Component Analysis.

The Mechanism of PCA:

- Compute the covariance matrix for data objects
- Compute the Eigen vectors and the Eigen values in a descending order
- To get the new dimensions, select the initial *N* Eigen vectors
- Finally, change the initial n-dimensional data objects into N-dimensions

Example: Below are the two graphs showing data points (objects) and two directions: one is 'green' and the other is 'yellow.' We got the Graph 2 by rotating the Graph 1 so that the x-axis and y-axis represent the 'green' and 'yellow' directions, respectively.





After the rotation of the data points, we can infer that the green direction (x-axis) gives us the line that best fits the data points.

Here, we are representing 2-dimensional data. But in real-life, the data would be multi-dimensional and complex. So, after recognizing the importance of each direction, we can reduce the area of dimensional analysis by cutting off the less-significant 'directions.'

Now, we will look into another important Machine Learning Interview Question on PCA.

24. Why rotation is required in PCA? What will happen if you don't rotate the components?

Rotation is a significant step in PCA as it maximizes the separation within the variance obtained by components. Due to this, the interpretation of components becomes easier.

The motive behind doing PCA is to choose fewer components that can explain the greatest variance in a dataset. When rotation is performed, the original coordinates of the points get changed. However, there is no change in the relative position of the components.

If the components are not rotated, then we need more extended components to describe the variance.

25. We know that one hot encoding increases the dimensionality of a dataset, but label encoding doesn't. How?

When we use **one-hot encoding**, there is an increase in the dimensionality of a dataset. The reason for the increase in dimensionality is that, for every class in the categorical variables, it forms a different variable.

Example: Suppose, there is a variable 'Color.' It has three sub-levels as Yellow, Purple, and Orange. So, one hot encoding 'Color' will create three different variables as Color.Yellow, Color.Porple, and Color.Orange.

In **label encoding**, the sub-classes of a certain variable get the value as **0** and **1**. So, we use label encoding only for binary variables.

This is the reason that one hot encoding increases the dimensionality of data and label encoding does not.

Now, if you are interested in doing an end-to-end certification course in Machine Learning, you can check out Intellipaat's Machine Learning Course with Python.

26. What is Overfitting in Machine Learning and how can you avoid?

Overfitting happens when a machine has an inadequate dataset and it tries to learn from it. So, overfitting is inversely proportional to the amount of data.

For small databases, we can bypass overfitting by the cross-validation method. In this approach, we will divide the dataset into two sections. These two sections will comprise testing and training sets. To train the model, we will use the training dataset and, for testing the model for new inputs, we will use the testing dataset.

This is how we can avoid overfitting.

27. Why do we need a validation set and a test set?

We split the data into three different categories while creating a model:

1. **Training set**: We use the training set for building the model and adjusting the model's variables. But, we cannot rely on the correctness of the model build on top of the training set. The model might give incorrect outputs on feeding new inputs.
2. **Validation set**: We use a validation set to look into the model's response on top of the samples that don't exist in the training dataset. Then, we will tune hyperparameters on the basis of the estimated benchmark of the validation data.

When we are evaluating the model's response using the validation set, we are indirectly training the model with the validation set. This may lead to the overfitting of the model to specific data. So, this model won't be strong enough to give the desired response to the real-world data.

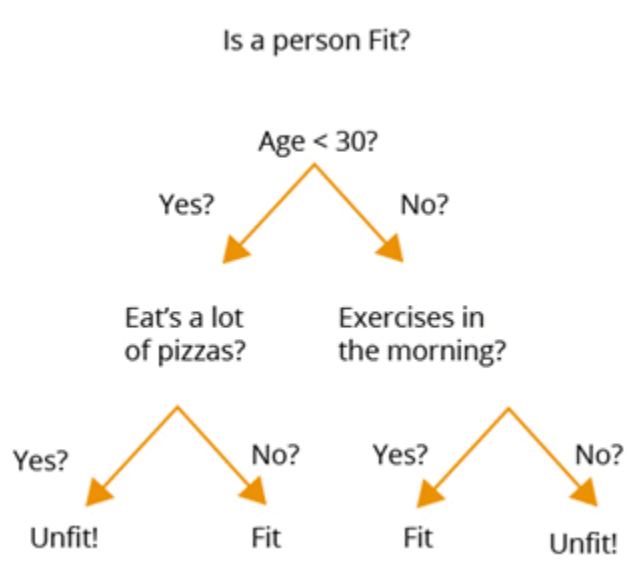
3. **Test set**: The test dataset is the subset of the actual dataset, which is not yet used to train the model. The model is unaware of this dataset. So, by using the test dataset, we can compute the response of the created model on hidden data. We evaluate the model's performance on the basis of the test dataset.

Note: We always expose the model to the test dataset after tuning the hyperparameters on top of the validation set.

As we know, the evaluation of the model on the basis of the validation set would not be enough. Thus, we use a test set for computing the efficiency of the model.

28. What is a Decision Tree in Machine Learning?

A decision tree is used to explain the sequence of actions that must be performed to get the desired output. It is a hierarchical diagram that shows the actions.

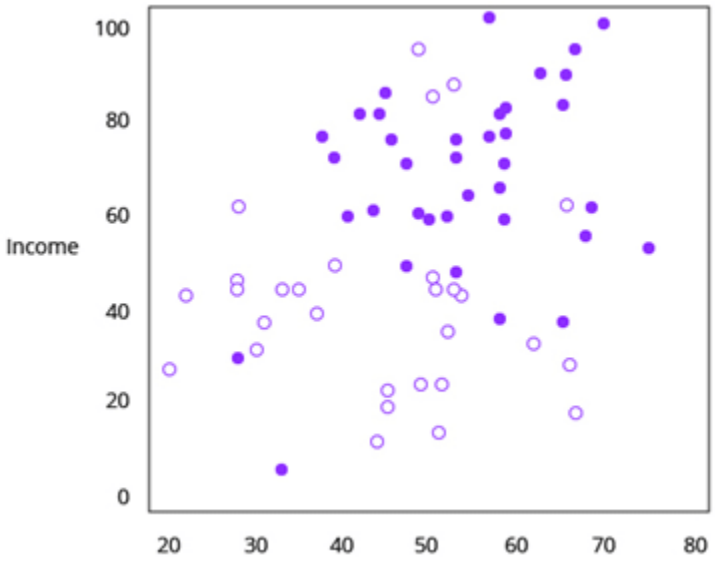


We can create an algorithm for a decision tree on the basis of the hierarchy of actions that we have set.

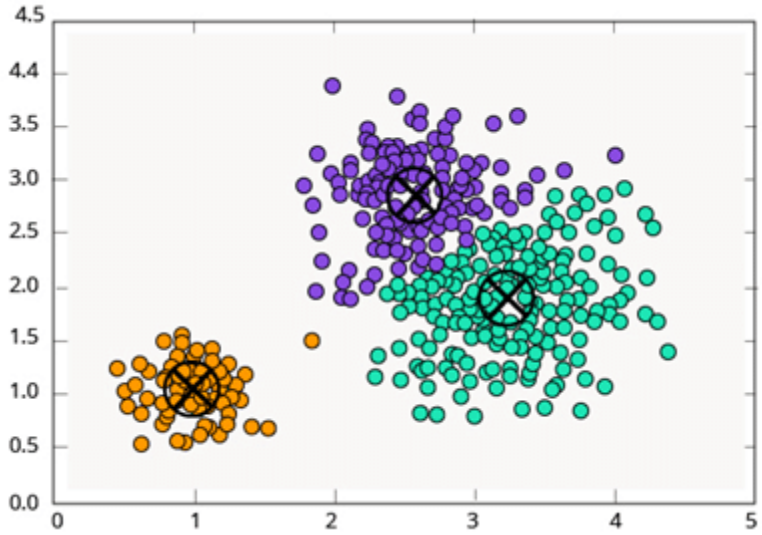
In the above decision tree diagram, we have made a sequence of actions for driving a vehicle with/without a license.

29. Explain the difference between KNN and K-means Clustering.

K-nearest neighbors: It is a supervised Machine Learning algorithm. In KNN, we give the identified (labeled) data to the model. Then, the model matches the points based on the distance from the closest points.



K-means clustering: It is an unsupervised Machine Learning algorithm. In this, we give the unidentified (unlabeled) data to the model. Then, the algorithm creates batches of points based on the average of the distances between distinct points.



30. What is Dimensionality Reduction?

In the real world, we build Machine Learning models on top of features and parameters. These features can be multi-dimensional and large in number. Sometimes, the features may be irrelevant and it becomes a difficult task to visualize them.

Here, we use dimensionality reduction to cut down the irrelevant and redundant features with the help of principal variables. These principal variables are the subgroup of the parent variables that conserve the feature of the parent variables.

31. Both being tree-based algorithms, how is Random Forest different from Gradient Boosting Algorithm (GBM)?

The main difference between a random forest and GBM is the use of techniques. Random forest advances predictions using a technique called 'bagging.' On the other hand, GBM advances predictions with the help of a technique called 'boosting.'

- **Bagging**: In bagging, we apply arbitrary sampling and we divide the dataset into *N* After that, we build a model by employing a single training algorithm. Following, we combine the final predictions by polling. Bagging helps increase the efficiency of the model by decreasing the variance to eschew overfitting.
- **Boosting**: In boosting, the algorithm tries to review and correct the inadmissible predictions at the initial iteration. After that, the algorithm's sequence of iterations for correction continues until we get the desired prediction. Boosting assists in reducing bias and variance, both, for making the weak learners strong.

32. Suppose, you found that your model is suffering from high variance. Which algorithm do you think could handle this situation and why?

Handling High Variance

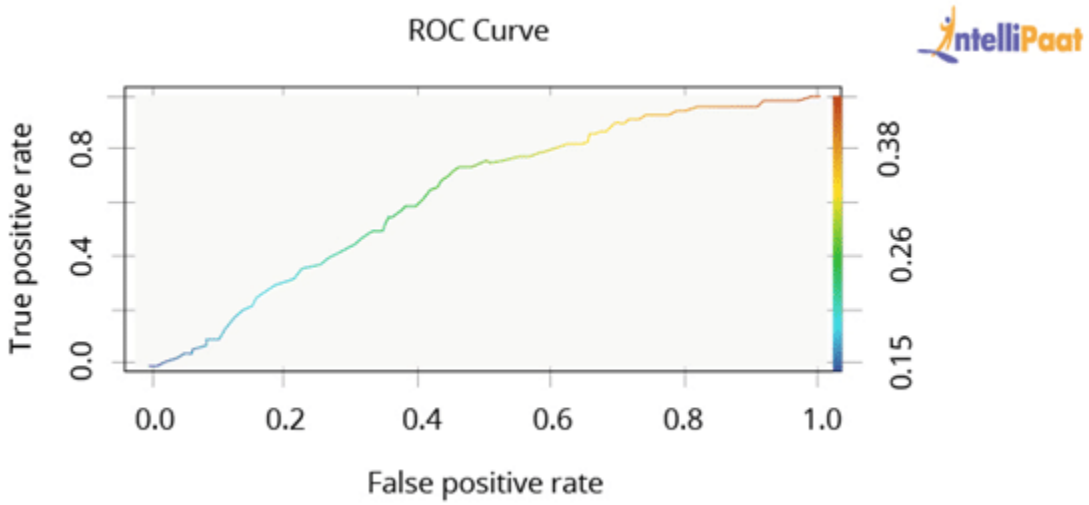
- For handling issues of high variance, we should use the bagging algorithm.
- The bagging algorithm would split data into sub-groups with a replicated sampling of random data.

- Once the algorithm splits the data, we use random data to create rules using a particular training algorithm.
- After that, we use polling for combining the predictions of the model.

33. What is ROC curve and what does it represent?

ROC stands for 'Receiver Operating Characteristic.' We use ROC curve to represent the trade-off between True and False positive rates, graphically.

In ROC, AUC (Area Under the Curve) gives us an idea about the accuracy of the model.



The above graph shows an ROC curve. Greater the Area Under the Curve better the performance of the model.

Next, we would be looking at Machine Learning Interview Questions on Rescaling, Binarizing, and Standardizing.

34. What is Rescaling of data and how is it done?

In real-world scenarios, the attributes present in data will be in a varying pattern. So, rescaling of the characteristics to a common scale gives benefit to algorithms to process the data efficiently.

We can rescale the data using Scikit-learn. The code for rescaling the data using MinMaxScaler is as follows:

```
#Rescaling data
import pandas
import scipy
import numpy
from sklearn.preprocessing import MinMaxScaler
names = ['Abhi', 'Piyush', 'Pranay', 'Sourav', 'Sid', 'Mike', 'pedi', 'Jack', 'Tim']
Dataframe = pandas.read_csv(url, names=names)
Array = dataframe.values
# Splitting the array into input and output
X = array[:,0:8]
Y = array[:,8]
Scaler = MinMaxScaler(feature_range=(0, 1))
rescaledX = scaler.fit_transform(X)
# Summarizing the modified data
numpy.set_printoptions(precision=3)
```

```
print(rescaledX[0:5,:])
```

35. What is Binarizing of data? How to Binarize?

In most of the Machine Learning Interviews, apart from theoretical questions, interviewers focus on the implementation part. So, these ML Interview Questions focused on the implementation of the theoretical concepts.

Converting data into binary values on the basis of threshold values is known as the binarizing of data. The values that are less than the threshold are set to **0** and the values that are greater than the threshold are set to **1**. This process is useful when we have to perform feature engineering, and we can also use it for adding unique features.

We can binarize data using Scikit-learn. The code for binarizing the data using Binarizer is as follows:

```
from sklearn.preprocessing import Binarizer
import pandas
import numpy
names = ['Abhi', 'Piyush', 'Pranay', 'Sourav', 'Sid', 'Mike', 'pedi', 'Jack', 'Tim']
dataframe = pandas.read_csv(url, names=names)
array = dataframe.values
# Splitting the array into input and output
X = array[:,0:8]
Y = array[:,8]
binarizer = Binarizer(threshold=0.0).fit(X)
binaryX = binarizer.transform(X)
# Summarizing the modified data
numpy.set_printoptions(precision=3)
print(binaryX[0:5,:])
```

36. How to Standardize data?

Standardization is the method that is used for rescaling data attributes. The attributes would likely have a value of mean as **0** and the value of standard deviation as **1**. The main objective of standardization is to prompt the mean and standard deviation for the attributes.

We can standardize the data using Scikit-learn. The code for standardizing the data using StandardScaler is as follows:

```
# Python code to Standardize data (0 mean, 1 stdev)
from sklearn.preprocessing import StandardScaler
import pandas
import numpy
names = ['Abhi', 'Piyush', 'Pranay', 'Sourav', 'Sid', 'Mike', 'pedi', 'Jack', 'Tim']
dataframe = pandas.read_csv(url, names=names)
array = dataframe.values
# Separate the array into input and output components
X = array[:,0:8]
Y = array[:,8]
scaler = StandardScaler().fit(X)
rescaledX = scaler.transform(X)
# Summarize the transformed data
numpy.set_printoptions(precision=3)
print(rescaledX[0:5,:])
```

37. Executing a binary classification tree algorithm is a simple task. But, how does a tree splitting take place? How does the tree determine which variable to break at the root node and which at its child nodes?

Gini index and Node Entropy assist the binary classification tree to take decisions. Basically, the tree algorithm determines the feasible feature that is used to distribute data into the most genuine child nodes.

According to Gini index, if we arbitrarily pick a pair of objects from a group, then they should be of identical class and the possibility for this event should be **1**.

To compute the Gini index, we should do the following:

1. Compute Gini for sub-nodes with the formula: The sum of the square of probability for success and failure (p^2 + q^2)
2. Compute Gini for split by weighted Gini rate of every node of the split

Now, Entropy is the degree of indecency that is given by the following:

where ***a*** and ***b*** are the probabilities of success and failure of the node

When **Entropy = 0**, the node is homogenous

When **Entropy is high**, both groups are present at 50–50 percent in the node.

Finally, to determine the suitability of the node as a root node, the entropy should be very low.

38. What is SVM (Support Vector Machines) in Machine Learning?

SVM is a Machine Learning algorithm that is majorly used for classification. It is used on top of the high dimensionality of the characteristic vector.

Below is the code for the SVM classifier:

```
# Introducing required libraries
from sklearn import datasets
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
# Stacking the Iris dataset
iris = datasets.load_iris()
# A -> features and B -> label
A = iris.data
B = iris.target
# Breaking A and B into train and test data
A_train, A_test, B_train, B_test = train_test_split(A, B, random_state = 0)
# Training a linear SVM classifier
from sklearn.svm import SVC
svm_model_linear = SVC(kernel = 'linear', C = 1).fit(A_train, B_train)
svm_predictions = svm_model_linear.predict(A_test)
# Model accuracy for A_test
accuracy = svm_model_linear.score(A_test, B_test)
# Creating a confusion matrix
cm = confusion_matrix(B_test, svm_predictions)
```

39. Implement the KNN classification algorithm.

We will use the Iris dataset for implementing the KNN classification algorithm.

```
# KNN classification algorithm
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
import numpy as np
from sklearn.model_selection import train_test_split
iris_dataset=load_iris()
A_train, A_test, B_train, B_test = train_test_split(iris_dataset["data"],
iris_dataset["target"], random_state=0)
kn = KNeighborsClassifier(n_neighbors=1)
kn.fit(A_train, B_train)
A_new = np.array([[8, 2.5, 1, 1.2]])
prediction = kn.predict(A_new)
print("Predicted target value: {}\n".format(prediction))
print("Predicted feature name: {}\n".format
(iris_dataset["target_names"][prediction]))
print("Test score: {:.2f}".format(kn.score(A_test, B_test)))
Output:
Predicted Target Name: [0]
Predicted Feature Name: [‘ Setosa’]
Test Score: 0.92
```