

**INTERVIEW
PREPARATION
MATERIAL**

**ARTIFICIAL
INTELLIGENCE,
MACHINE LEARNING,
DEEP LEARNING,
NATURAL LANGUAGE
PROCESSING**

5/11/2020

COMPILED BY ABHISHEK PRASAD

INDEX

Contents	Page Number
CHAPTER 1: Interview Questions on Artificial Intelligence	2-17
CHAPTER 2: Interview Questions on Machine Learning	18-56
CHAPTER 3: Interview Questions on Deep Learning	57-84
CHAPTER 4: Interview Questions on Natural Language Processing	85-95

Number of Questions on Artificial Intelligence = 40

Number of Questions on Machine Learning = 85

Number of Questions on Deep Learning = 50

Number of Questions on Natural Language Processing = 35

Total Number of Questions = 210

CHAPTER 1

INTERVIEW QUESTIONS

ON

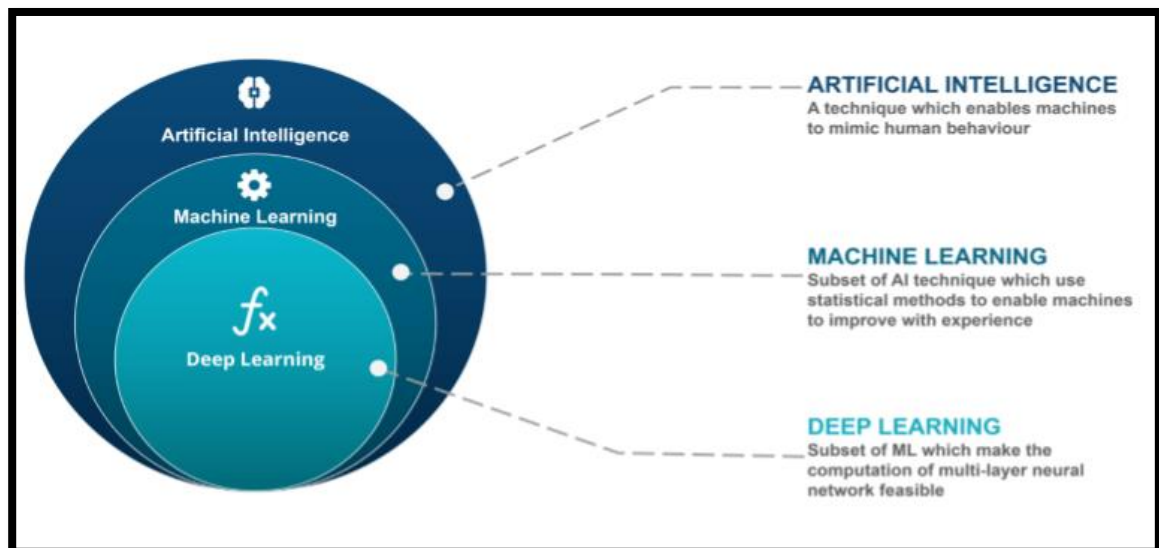
ARTIFICIAL INTELLIGENCE

(TOP 40 QUESTIONS)

Q1. Differentiate Machine Learning, Deep Learning and Artificial Intelligence.

Answer 1:

- **Machine Learning:** Machine learning is nothing but building an algorithmic model that can make sense out of data. In case of any prediction error, tuning is done manually by the developer. Machine learning is a subset of artificial intelligence.
- **Deep Learning:** Deep learning is a subset of machine learning and performs actions similar to machine learning. It makes use of neural networks instead of generic algorithms to make sense out of data.
- **Artificial Intelligence:** The goal here is to build an automated model that can think and react to a situation like a human. Deep learning and machine learning algorithms can be integrated together to create a model that can mimic human behaviour. For example, voice assistants make use of supervised learning(Classification) to categorize user input and respond accordingly.



Q2. Differentiate AI systems based on their functionalities.

Answer 2:

1. **Reactive Memory:** The most basic form of AI. It does not store or make use of previous experience. Reacts to an input based on pre-fed information. Example: Chess engines like Stock fish or fritz.
2. **Limited Memory:** Models that can store past experience for a short period of time. For example, in a self-driving car, the speed and other factors of surrounding cars are recorded and stored in the memory until the ride is over. It is not stored in their built-in library.
3. **Theory of Mind:** This type of AI will focus more on understanding human emotions so that it can have a better understanding of human actions.
4. **Self-Awareness:** The future of AI. These types of AI can understand the surrounding circumstances as well as express themselves. Sophia robot is a great example of a self-aware AI.

Q3. Differentiate statistical AI and classical AI.**Answer 3:**

Statistical AI leans more towards inductive thought i.e. given a set of patterns identify and produce the trend in that pattern. Whereas Classical AI, leans more towards deductive thought i.e. given a set of relations or constraints deduce a conclusion.

Q4. What are the different domains of artificial intelligence?**Answer 4:**

- **Machine Learning:** It's the science of getting computers to act by feeding them data so that they can learn a few tricks on their own, without being explicitly programmed to do so.
- **Neural Networks:** Neural networks are inspired by human brains. They are created with human brains as their reference and try to replicate human thinking.
- **Robotics:** An AI Robot works by manipulating the objects in its surroundings, by perceiving, moving and taking relevant actions. This is achieved using various decision-making algorithms.
- **Expert Systems:** An expert system is a computer system that mimics the decision-making ability of a human. It is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behaviour of a human or an organization that has expert knowledge and experience in a particular field.
- **Fuzzy Logic Systems:** Traditional logic reasoning contains only two possible outcomes either true or false (0 or 1). But fuzzy logic involves all intermediate results too i.e. it contains values in the range 0 to 1. It tries to mimic human decision making.
- **Natural Language Processing:** NLP refers to the Artificial Intelligence method that analyses natural human language to derive useful insights in order to solve problems.

Q5. Why are voice assistants like Siri, Alexa and Echo considered as weak AI?**Answer 5:**

Voice assistants like Siri, Alexa and Echo rely highly on user input and they classify them based on pre-fed information. Even some of the most complex chess programs are considered to be weak AI as they make use of a chess database to make their next move. On the other hand, strong AI makes use of clustering instead of classification. Strong AI is designed to think and react like a human instead of relying on pre-fed information.

Q6. How do you assess whether an AI is capable of thinking like a human or not?

Answer 6:

Turing test is one of the most famous methods that is used to assess an AI machine. This method contains three terminals. The first terminal is an interrogator who is isolated from the other two terminals, i.e., machine and a human. The interrogator will ask questions and predict who is more likely to be a human using the response that he gets.

Q7. Why use semantic analysis in AI?

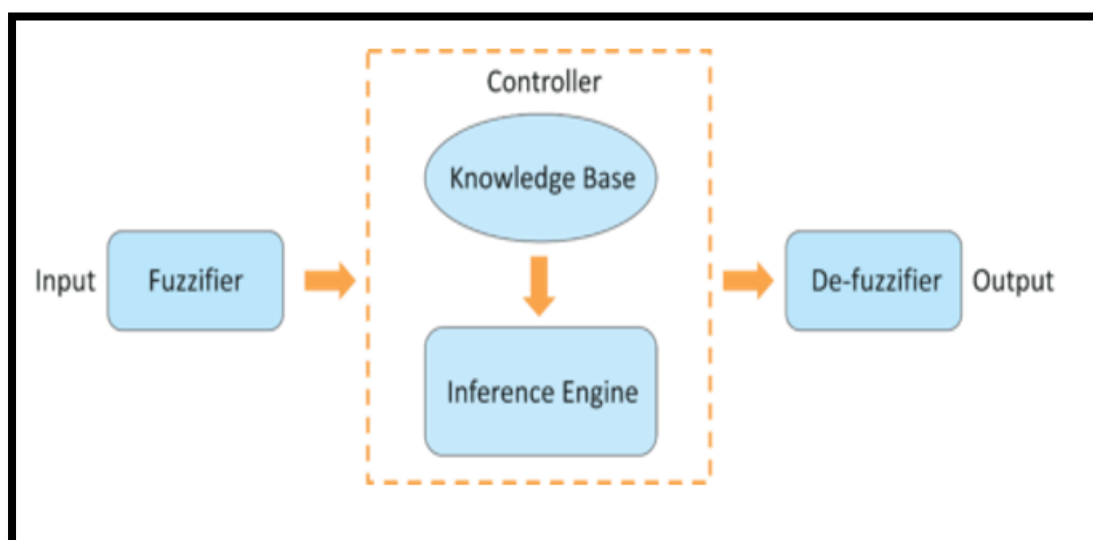
Answer 7:

Semantic analysis can be used to extract meaning from a given data so that it can be used to train a model. This comes handy when we have to develop a chatbot or any other AI application that makes use of text data.

Q8. What is fuzzy logic and explain its architecture?

Answer 8:

Traditional logic reasoning contains only two possible outcomes either true or false (0 or 1) but, Fuzzy logic involves all intermediate results too i.e. it contains values in the range 0 to 1. It tries to mimic human decision making. For example, when you want to grade a group of students instead of having just two grades pass or fail, you can have different types of grades like outstanding, average, pass and fail. Fuzzy logic is used in decision-making tasks where an AI needs to make a decision.



- **Fuzzification Model:** Inputs are fed in here which is then converted from crisp sets to fuzzy sets.

- **Knowledge Base:** Knowledge base is a must for any system that works on AI. Here the rules of the fuzzy logic set theory are stored which is in the form of if-else statements.
- **Inference Engine:** Simulates human reasoning by making inference on inputs based on the if-else rules
- **Defuzzification model:** Converts the fuzzy sets obtained from the inference engine back to the crisp set.

Q9. You are asked to create a model that can classify images. Since you are limited by computer power, you have to choose either supervised or unsupervised learning to implement it. Which technique do you prefer? and why?

Answer 9:

Both of the techniques can be used to implement image classification. But I would prefer supervised learning over unsupervised learning. In supervised learning, the ML expert feeds and interprets the image to create the required feature classes, whereas in unsupervised learning the model creates the feature classes on its own making it difficult to make some changes in it if required.

Q10. How can the Bayesian model be helpful to create an AI model?

Answer 10:

Bayesian networks make use of probabilistic values instead of binary values to make a decision. So, if an AI model needs to make a decision for a probabilistic query, then Bayesian networks can be implemented.

Q11. Explain the different types of hill climbing algorithm.

Answer 11:

There are three types of hill climbing algorithms.

1. **Simple hill climbing:** In this method, the nearby nodes are examined one by one and the first node which optimizes the current cost value is selected as the next node.
2. **Steepest-Ascent hill climbing:** In this method, all the nearby nodes are examined first. Then it selects the node which takes us closer to the solution state as the next node.
3. **Stochastic hill climbing:** In this method, a random neighbouring node is selected first. Then based on the improvement in that node, it decides whether to move to that node or to examine other nodes.

Q12. What is the purpose of search algorithms in AI?**Answer 12:**

In artificial intelligence, search algorithms are widely used to solve and provide the best possible result for a given problem statement. They are generally used in goal-based agents. Goal-based agents choose the actions that take them closer to the end goal. Future actions are taken into consideration here.

Q13. When you are limited by computer memory, which search algorithm would you prefer and why?**Answer 13:**

The depth-first search algorithm is preferred here as it consumes less space in memory. It is because only the nodes in the current path are stored whereas, in breadth-first search, all of the trees that have been generated must be stored.

Q14. Scenario: You are asked to develop an AI that can teach itself to play chess using search algorithms. What kind of search approach do you prefer and why?**Answer 14:**

Traditional search algorithms use exhaustive search approach. This type of approach tends to explore all possible combinations in an environment to provide a solution. This can be good when the total number of possibilities is less (eg: tic-tac-toe). But in our case, the total number of possibilities is very high.

So, it is preferred to use the combinatorial search approach. It makes use of pruning strategies to eliminate some of the possibilities making it less complex to compute. One of the most famous pruning strategies is Alpha-Beta pruning, where it avoids searching the parts of trees that do not contain the solution.

Q15. Why do we use a heuristic function? How can it be useful in a chess engine?**Answer 15:**

The heuristic function calculates an approximate cost for a given problem. For example, it can calculate the cost to move from one point to another. It ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow. Heuristic search makes use of this function to calculate the cost value.

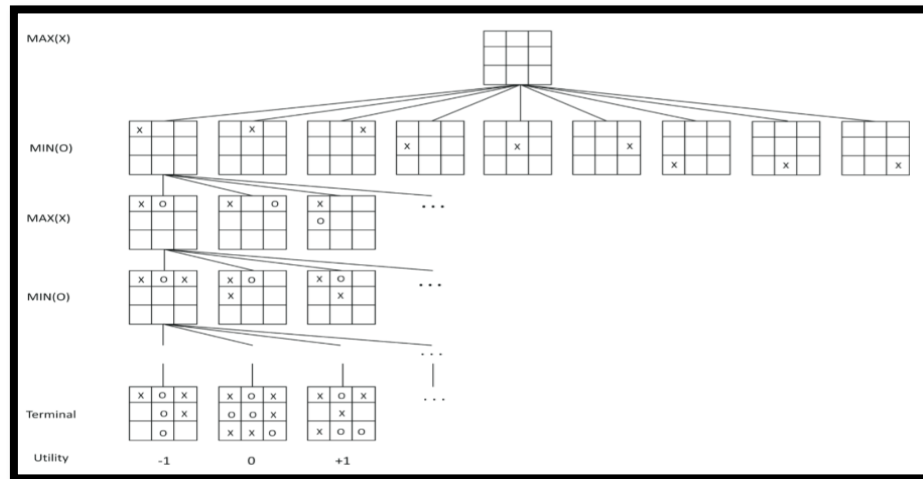
In a chess engine, the heuristic function can be applied to remove all possible moves that will lead to a bad position/loss. This will enable the chess engine to explore more moves in less time since it's not wasting time on bad moves.

Q16. How does the minimax algorithm make a decision? Also, explain its working using the tic-tac-toe game.

Answer 16:

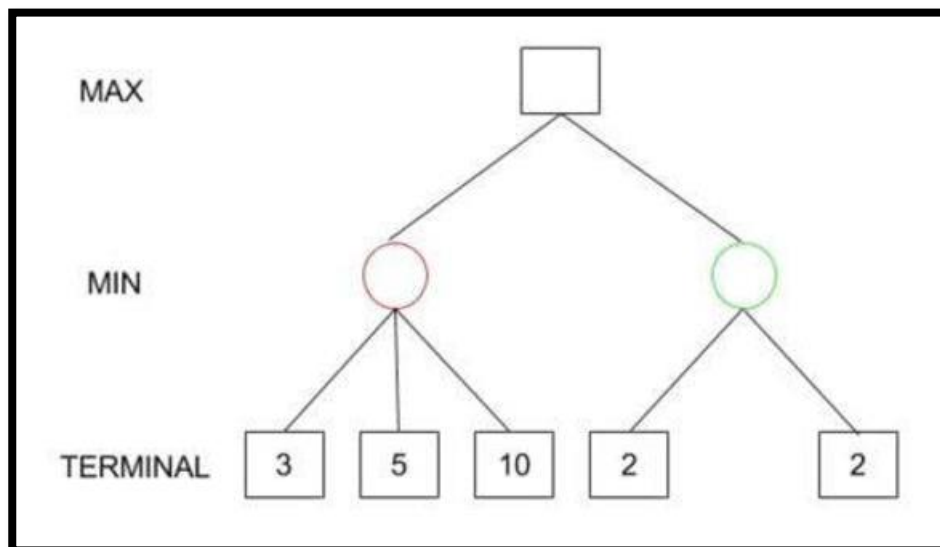
The ideology behind the minimax algorithm is to choose the move that maximizes the worst-case scenario for the opponent instead of choosing a move that maximizes its own win chances. The following approach is taken for a Tic-Tac-Toe game using the Minimax algorithm:

Step 1: First, generate the entire game tree starting with the current position of the game all the way up to the terminal states.



Step 2: Apply the utility function to get the utility values for all the terminal states.

Step 3: Determine the utilities of the higher nodes with the help of the utilities of the terminal nodes. For instance, in the diagram below, we have the utilities for the terminal states written in the squares.

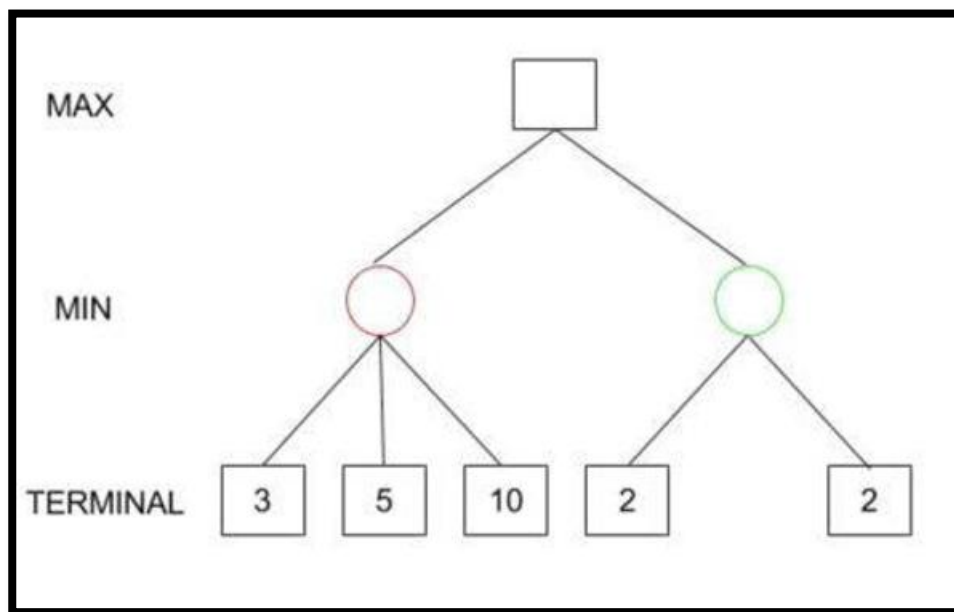


Let us calculate the utility for the left node(red) of the layer above the terminal:

$\text{MIN}\{3, 5, 10\}$, i.e. 3.

Therefore, the utility for the red node is 3. Similarly, for the green node in the same layer:

$\text{MIN}\{2,2\}$, i.e. 2.



Step 4: Calculate the utility values.

Step 5: Eventually, all the backed-up values reach to the root of the tree. At that point, MAX has to choose the highest value: i.e. $\text{MAX}\{3,2\}$ which is 3.

Therefore, the best opening move for MAX is the left node (or the red one).

To summarize, $\text{Minimax Decision} = \text{MAX}\{\text{MIN}\{3,5,10\}, \text{MIN}\{2,2\}\} = \text{MAX}\{3,2\} = 3$

Q17. What is an intelligent agent?

Answer 17:

An intelligent agent makes use of sensors to analyze the environment and make decisions according to the current situation.

Q18. Differentiate single-agent systems and multi-agent systems with examples.**Answer 18:**

- When there is only one agent in the defined environment then it is considered as a single agent system. For example, consider a maze environment where the agent has to navigate and find the shortest path possible to exit the maze.
- Similarly, when there is more than one agent in the defined environment then it is considered as a multi-agent system. For example, consider the environment as a 4*4 chessboard and 4 queens as agents. Q learning is used to place the queens on the chessboard in a manner that no 2 queens should be placed on the same row, the same column or the same diagonal.

Q19. Explain Model-based learning vs model-free learning.**Answer 19:**

Model-free learning: In model-free learning, the agent makes a decision based on some of its previous trial and error experience. That is it removes a possible action based on its previous experience that can lead to a bad result. Model-free learning is more time consuming but usually provides more efficient results.

Model-based learning: In model-based learning, the agent makes use of a pre-trained model to make decisions. That is the agent gains values from a previously trained model and makes decisions based on those values. Learning is less time consuming, but if the model is inaccurate then the results can be completely different from expected.

Q20. Explain exploration vs exploitation trade-off.**Answer 20:**

- Exploration, as the name suggests is about navigating or exploring the environment to collect information about it. It uses the hit and trial method to explore the environment and stores the collected information.
- Exploitation, on the other hand, makes use of already known information to make a decision that can increase the reward value.

For example, if you go to the same clothing store in your favorite mall all the time you can predict the type of collections you can get from there but will miss out on the other options that are available nearby. But if you visit all possible options in a mall you will occasionally come across a few stores that have a bad set of collections.

If you decide to go to your favorite store in the same mall, then it is known as exploitation (making use of known data).

If you decide to explore more to find alternate options then it is known as exploration (gaining new information of the environment).

Q21. Difference between deep Q-learning and deep learning.**Answer 21:**

The major difference between them is that in a deep q learning, the current state of the model changes often, thus resulting in a change of the target. Therefore, the target in deep q learning is considered unstable. A deep learning model learns first from the train set and then implements it in a new dataset of unseen data. The target variable does not change and it is stable.

Q22. When and why do you choose deep Q-learning over Q-learning?**Answer 22:**

As the number of states increases, the size of Q-table increases as well. This will increase the memory used to store and update the values of Q-table as well as the time needed to explore each step. This is where deep Q learning comes handy where all the past experience is stored in the memory and used for future exploration.

For example, consider a self-driving car that needs to find the shortest route from the start point to the endpoint. Deep Q learning will be used here to explore all possible routes, avoid some routes based on previous experiences and then find the best route possible by comparing the Q-values obtained at the end of each action.

Q23. Why do we initialize a negative threshold value for the deep Q-learning model?**Answer 23:**

A negative threshold value is initialized to terminate the action in case of any senseless roaming. For example, let us imagine a simple maze environment where the agent cannot die. Then there comes a possibility where the agent can move to a square that takes him far away from the end-point or he can move to a square which he has already visited. This may make the model to run in an infinite loop or produce results that are not optimal. Initializing a negative threshold value will remove all these possibilities.

Q24. Scenario: Consider an environment where the agent has to navigate his way from the start point to the endpoint. The environment contains two types of cells: free cell and closed cell. The agent can move only one step at a time and is allowed to move only towards the free cells. The agent can move only in four directions (Top, Down, Left, Right). How can deep reinforcement learning be implemented here to navigate the agent from the start point to the endpoint?

Answer 24:

Deep Q learning can be used here to find the shortest path possible through a reward system. Reward agent:

- (i) +10 points if the agent moves to a new cell
- (ii) -8 if the agent tries to move to a closed-cell or a cell outside the environment

- (iii) -5 if the agent tries to move to a cell that it has already visited

This will help the agent to learn to avoid blocked cells or already visited cells and encourage him to move towards a new cell. Let the agent explore all the possibilities and store the Q-action values that can be fed as input for the next model. The agent will make use of its past experience to avoid any previous mistake that it has committed. This will make the agent find the shortest path possible from the start point to the end-point.

Q25. Differentiate Markov models based on their two main characteristics (Control over states and the observability of a state)?

Answer 25:

- **Markov Decision Process (MDP):** The agent has complete control over state transitions and the states are observable.
- **Partially Observable MDP (POMDP):** The agent has complete control over state transitions but the states are only partially observable.
- **Hidden Markov Model (HMM):** The agent does not have control over the state transitions and the states are partially observable.
- **Markov Chains:** The agent does not have control over state transitions but the states are observable.

Q26. Why do we calculate the probability of the system in the Markov decision process?

Answer 26:

We calculate the probability of the system to capture the transition of the system from one state to another. It is influenced by the chosen action and the next state depends on the current probability value.

Q27. What is the necessity of value function and how do you choose an optimal value function for a Markov decision process model?

Answer 27:

Value function tells the agent how good it is to be in a state, how good it is to perform a certain action and gives an expected reward value if the agent performs a certain action. In simple words, value function tells the agent which state is important or good to be in.

Bellman equation is used to calculate the optimal value function for a given state. Bellman equation decomposes the value function into two parts:

1. **Instant reward:** Reward value that will be obtained from the successor state.

2. **Discounted future value:** Reward value that the agent will receive overtime starting from the current state.

These values are used to calculate an optimal policy. Bellman equation can be calculated using the following formula: $V(s) = \max(R(s,a) + \gamma V(s'))$

Where,

- a - action
- s - a particular state
- s' - the next state where the agent moves from s
- V(s) and V(s') - value for the state s and s' respectively
- γ - discount factor
- R(s, a) - reward value received after performing an action (a) from the state (s)

Q28. Differentiate Markov process and Hidden Markov models (HMM)?

Answer 28:

Markov process is a stochastic process wherein random variables transitions from one state to the other in such a way that the future state of a variable only depends on the present state.

Hidden Markov models are similar to a Markov process except that the states of the process are hidden here. They are used to model sequence data behavior or in the modeling of time series data.

Q29. What is the major application of HMM?

Answer 29: HMM is used in almost all speech recognition systems nowadays. The voice input from the user is the observations here and the part of speech is to be predicted, which are the hidden states of the model.

Q30. What are the terms required to create a Bayes model?

Answer 30:

We need three terms to build a Bayesian model, one conditional probability and two unconditional probability.

Q31. What is the use of incremental mean value in the Monte Carlo method?

Answer 31:

Incremental mean value return is used to measure the progress made by the model after each episode. Monte Carlo method learns from previous episodes and this can be used to measure the model performance.

We calculate the mean return value after each episode, convert them into an incremental update value so that the difference between two mean values can be calculated easily.

Q32. Does the Monte Carlo approach require prior MDP transition values to make decisions?

Answer 32:

No, the Monte Carlo approach can directly learn from episodes of previous experiences without any prior knowledge of Markov's Decision process transition.

Monte Carlo approach receives reward at the end of each episode. When they reach the terminal state, they make use of the total cumulative reward received and start over again with newly gained knowledge.

Q33. Monte Carlo tree search (MCTS) algorithms tend to perform better when merged with reinforcement learning. What is the reason behind it?

Answer 33:

Monte Carlo fails to perform well on a large scale. Integrating MCTS with reinforcement learning solves this issue. When integrated, MCTS makes use of strong learning techniques from reinforcement learning to create a model that performs well on a large scale. This is proven by the AlphaGO (an ai developed by Google) engine, which makes use of this concept to defeat the best GO (board game) players in the world.

Q34. What is the need for reward maximization in reinforcement learning?

Answer 34:

The Reinforcement learning agent works on the principle of reward maximization. When we train the RL agent to maximize the reward value, it will help the agent to choose the best possible action. Making use of reward maximization makes the agent more optimal.

Q35. What is the function of the neural networks in artificial intelligence?

Answer 35:

Neural networks are inspired by human brains. They are created with human brains as their reference and try to replicate human thinking. They are composed of artificial nodes and neurons that can solve complex problems by mimicking the human decision-making approach. An AI model can be created with neural networks that can perform tasks that can produce solutions faster than humans.

Q36. How can search engines like google can produce better search results with the help of deep learning?

Answer 36:

Search engines generally make use of machine learning algorithms to find results for a search. They make use of various predictive analysis algorithms to find the best result. With deep learning integration into the search engine, the search results can be more relevant than to the specific user rather than a generalized result. The major problem arises when you need to understand the basis of classification on a search query because the neural network model produces machine-readable information which is really hard to interpret.

Q37. What is the need for hyperparameters in neural networks?

Answer 37:

Hyperparameters can be used to define the learning rate and the number of hidden layers that should be present in a neural network model.

- Learning rate value defines the speed at which the neural network should learn. Having a higher learning rate may cause the model to understand only one single feature from the data and use only that for identification.
- Having a low learning rate will cause the model to take more time to get trained.
- So, we need the right learning rate that is low enough to learn something useful from the data and at the same time high enough to train the model in a possible time frame.
- Increasing the number of hidden layers can improve the accuracy of the model and can solve underfitting.

Q38. How to avoid overfitting in neural networks?

Answer 38:

- **Reducing the complexity** of the neural network model can help to avoid overfitting. Reducing the number of neurons can avoid overfitting but reducing too many can decrease the performance of the model.
- **Early stopping:** Training the data for too long can cause overfitting. So it is preferred to stop the training when the performance of the model starts to degrade. This can be achieved by having a validation dataset which evaluates the model after every iteration. The training process can be stopped when the loss in the model begins to increase.
- **L1 and L2 Regularization:** Regularization can be achieved by adding a penalty term to the loss function. This can reduce the complexity of the model.
- **Dropout:** Dropping random neurons from the neural network during every iteration in training. It is a type of regularization.

- **Data augmentation:** It is nothing but increasing the data by artificial means. For example, where there is overfitting in an image classifier model, new images can be added by making modifications to the existing images.

Q39. Why should we prefer sigmoid neurons?

Answer 39:

In perceptrons due to harsh thresholding, even a small amount of difference between the threshold and weighted sum will change the output value completely. To make the concept clear, let's consider a scenario where you created a neural network model with perceptrons to predict whether a customer will buy a product or not, based on his salary. You have defined threshold value for the salary of 30000 INR. If the input salary is above the threshold value, the customer will purchase the product. So, if a customer who has a salary value of 29999 INR will be categorized with people who will not buy the product or have very less salary. But this will not be the case in the real-world scenario where the user with a salary value of 29999 INR has a chance of buying the product when compared to a user with a salary value of 9000 INR.

To overcome harsh thresholding, **sigmoid neurons** are used. In sigmoid neurons, a small change in input won't affect the output significantly instead causes a small change in the output. This makes the sigmoid output smoother than the step functional output.

Q40. Scenario: An AI model has been trained using deep learning neural networks to identify and classify cars for the given data. How do you convert the existing model to identify trucks that have similar features to a car? (Transfer learning)

Answer 40:

We can fine-tune the existing model so that it can identify trucks instead of cars. We can do the following changes to our model to fine-tune it to identify trucks.

- The first step is to replace the existing output layer which identifies cars with a new output layer. This new output layer will be used to identify trucks.
- The second step is to remove the features that are unique to the car. These unique features will decrease the model performance as they are just irrelevant to the target variable.
- Add features that are unique to a truck so that the training of the model is more efficient. This will increase the model performance as there is a better chance of identifying a truck with such unique features.
- Freeze the layers so that the layer weights of the pre-trained models are not changed. These layers can be reused when we train our new model. Only layer weights of newly added hidden layers should be updated. This is extremely useful when the dataset is large as it reduces the time required to re-train all hidden layers.

CHAPTER 2

INTERVIEW QUESTIONS

ON

MACHINE LEARNING

(TOP 85 QUESTIONS)

Q1: What are the different types of Machine Learning?**Ans1:**

	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definition	The machine learns by using labelled data	The machine is trained on Unlabelled data without any guidance	An agent interacts with its environment by producing actions & discovers errors or rewards
Types of problems	Regression & Classification	Association & Clustering	Reward based
Type of data	Labelled data	Unlabelled data	No pre-defined data
Training	External supervision	No supervision	No supervision
Approach	Map labelled input to known output	Understand patterns and discover output	Follow trail and error method
Popular algorithm	Linear Regression, Logistic Regression, Support Vector Machine	k-means, C-means, etc.	Q-Learning, SARSA, etc.

Q2: Differentiate between inductive learning and deductive learning?**Ans2:**

In **inductive learning**, the model learns by examples from a set of observed instances to draw a generalized conclusion. On the other side, in deductive learning, the model first applies the conclusion, and then the conclusion is observed. Inductive learning is the method of using observations to draw conclusions. Deductive learning is the method of using conclusions to form observations. Let me explain it with an example.

Example: If we have to explain to someone that driving fast is dangerous. There are two ways to do this. We can just show him the pictures of various accidents and pictures of the injured ones. In this case, he will understand with the help of examples and he will not drive fast again. It is the form of Inductive machine learning. The other way to teach him the same thing is to let him drive and wait to see what happens. If he gets injured in the accident, it will teach him not to drive fast again. It is the form of deductive learning.

Q3: Define parametric models? What are its examples?

Ans3:

- **Parametric models:** These models can be defined as the one which has a finite number of parameters means you only need to know the parameters of the model to predict new data. Examples are linear regression, logistic regression, and linear SVM.
- **Non-parametric models:** These models can be defined as the one which is not bound with the number of parameters means you need to know the parameters of the model and the state of the data that has been observed to predict new data. These models allow more flexibility. Examples include decision trees, k-nearest neighbors and topic models using latent Dirichlet analysis.

Q4: When we use One-hot encoding, the dimensionality of a dataset increases. But when we use label encoding it remains the same. Why?

Ans4:

In One Hot Encoding, if we have 'n' unique number of values in the column, then it will create the new 'n' number of columns with binary values in it. Then we can concatenate these columns with the dataframe, as a result, it will also increase the dimensionality of data. In Label Encoding, it will create only one column with an 'n' number of numerical values in it. Then we can replace this column with the original column and as a result, dimensionality will remain the same. For Example, we have a dataframe given below which has 3 unique values (Gas, Fuel, and Electricity).

	A
0	Gas
1	Fuel
2	Electricity
3	Gas
4	Fuel
5	Gas
6	Fuel
7	Electricity
8	Fuel
9	Fuel
10	Gas

In one hot encoding, it will return three columns named Gas, Fuel, Electricity. Each column will contain binary values (0 and 1). But when we use label encoding, it will return only one column which contains numerical values (1,2 and 3).

Q5: Suppose you have created a Linear regression model. After you run your model on different subsets, you realize that the beta values(coefficients) widely vary in each subset. What could be the problem here?

Ans5:

This case arises when the dataset is heterogeneous. So, to overcome this kind of problem, we should cluster the dataset into different subsets and then build the model separately for each cluster. Another way to solve such a problem is to use non-parametric models, such as decision trees, which can quite efficiently handle the heterogeneous data.

Q6: Define data augmentation? What are its examples?

Ans6:

Data augmentation occurs when you create new data by modifying existing data in such a way that the target is not changed, which means you will make reasonable modifications. For example, you have an image of a Lion who is faced to the right. After training the model, if you give it an image of a Lion who is not facing to the right, it will not consider it as a Lion, which isn't the right concept. Our model should perform on any image of a lion whether it is facing any direction. In the field of Computer vision data, augmentation is very useful. There are many types of modifications that you can make but the common ones are:

- Rotate
- Resize
- Horizontal or vertical flip
- Color Modifications
- Noise manipulation
- Deformation

Each problem needs a customized data augmentation pipeline. For example, on optical character recognition (OCR), doing flips will change the text and won't be beneficial; however, resizes and small rotations may help.

Q7: Why is Pearson's correlation different from correlation?

Ans7.

Pearson's correlation is important because it is used to find the linear relationship between independent and dependent variables. While Correlation can be used to find relationships between two variables.

Q8: What is univariate analysis, bivariate analysis, and multivariate analysis?**Ans8:**

- **Univariate analysis:** This is the part of exploratory data analysis in which we analyze each independent variable with target separately. For example, we have 4 predictors and 1 target. Then we can create 4 distribution plots to analyze the effect of every single variable separately.
- **Bivariate analysis:** This is the part of exploratory data analysis in which we analyze two predictors with the target at the same time. In simple words, we can say it is an analysis of bivariate data. For example, if we have 2 categorical predictor variables. We can create a box plot to analyze the effect of 2 predictors at the same time.
- **Multivariate analysis:** This is the part of exploratory data analysis in which we analyze more than 2 variables at the same time. In simple words, we can say it is an analysis of more than two variables. For example, we have 4 categorical variables. We can create a count plot of multiple features to analyze the majority of values in each feature at the same time.

Q9: What are the basic requirements you need to check before applying linear regression?**Ans9:**

The requirements are:

- **Linear relationship**
You have to check if there is a linear relationship between a predictor and the target variable. One way to check this is `scipy.stats.pearsonr(predict_column, Target_column)`. This will return two values, the first one will tell us how strong the linear relationship is and the other one will be “p_value” which is used to check the dependency between them.
- **Multivariate normality**
You have to check whether the data is normal or not. If not, you have to clean it and remove outliers so that you can use the proper sampling.
- **No or little Multicollinearity**
You have to check if there is a relationship between independent/ predictor variables. you can't have predictors (independent variables) that are dependent on each other.
- **Homoscedasticity**
In this case, we are trying to find out is there a situation in which the error term is the same across all values of the independent variables. This error term could be the “noise” or random disturbance in the relationship between the independent variables and the dependent variables.

Q10: How can we reduce multicollinearity from data?**Ans10:**

In simple words, multicollinearity occurs when we have independent variables that are correlated with each other. It occurs when your model has multiple features which aren't correlated just to your target variable, but also with each other. Let me explain this with the help of an example: suppose you went for a concert where two rappers say Eminem and Jay-z are singing at the same stage and at the same time. It will be very hard to decide which one is impacting more on the audience because both of them are singing totally different words. Multicollinearity makes it hard to interpret your coefficients, and it reduces the power of your model to identify independent variables that are statistically significant. These are definitely serious problems. However, the good news is that you don't always have to find a way to fix multicollinearity. The need to reduce multicollinearity depends on its severity and your primary goal for your regression model. Some of the ways to reduce multicollinearity are:

- **Principal Components Analysis (PCA):** This method is used to cut the number of predictors into a smaller set of uncorrelated components.
- **Partial least squares (PLS):** This method is an extension of PCA. This is a widely used technique in chemometrics, especially in the case where the number of independent variables is significantly larger than the number of data points. It constructs new predictors(independent variables), known as components, as linear combinations of the original predictors(independent variables). It creates components to explain the observed variability in the predictor variables, by taking the response variable in the account.
- **Variance inflation factor(VIF):** After calculating VIF for each column, if you have two or more factors with a high VIF, we have to remove one from the model. Because they supply nonuseful information, removing one of the correlated factors usually doesn't drastically reduce the Rsquared. We can use stepwise regression, best subsets regression and the important thing is we should have specialized knowledge of the data set to remove these variables. In the end, we can select the model that has the highest R-squared value.

Q11: What is the Q-Q plot in linear Regression? How can we interpret this plot?**Ans11.**

Q-Q plot stands for a quantile-quantile plot. These plots are ubiquitous (very common) in statistics. As the name suggests, we are plotting quantiles against quantiles. So the Q-Q plot can be defined as the graphical plotting of the two distributions of quantiles with respect to each other. We should keep in mind, whenever we interpret a Q-Q plot, our concentration should be on the 'y = x' line. That is the reason it is also called a 45-degree line in statistics because it entails us that each of our distributions has the same quantiles. In case we witness a deviation from this line, one of the distributions could be skewed when compared to the other.

Q12: Why do we use regularisation?**Ans12:**

Regularisation is mainly used to tackle the problem of the overfitted model. Whenever we implement a very complex model on the training data, the chances for it overfits are very high. In such cases, the simple model might not be able to generalize the data, so that is the reason we use regularisation.

Q13: L1 or L2, which performs better?**Ans13:**

You might already know that L1 is a technique used by Lasso and L2 is a technique used by Ridge. Generally, L2 performs better because it is efficient in terms of computations. But there is a case when L1 performs better. L1 supports build-in feature selection for the sparse matrix. It means L1 can perform feature selection as well as parameter shrinkage while the L2 can perform feature selection but not parameter shrinkage.

Q14: When should you choose Logistic Regression over Linear regression?**Ans14:**

- Logistic Regression can work with any type of data whether it is continuous or categorical. While the Linear Regression can only be used when the values of the target variable are continuous. • Logistic regression doesn't care about the relation of predictors with each other, but in Linear Regression there shouldn't be any correlation between the predictor variables.
- Logistic Regression can work with any type of relationship whether it is linear or nonlinear. On the other hand, it is required to have a linear relationship between the predictors and the target.

Q15: What is a cost function? Which type of cost functions are used in linear and logistic regression?**Ans15:**

In machine learning, cost-functions are used to check how badly models are performing. In simple words, a cost function is used to measure how wrong the model is doing in terms of its ability to find the relationship between X(predictors) and y(target). This can be expressed as a difference(or distance) between the predicted value and the actual value. This function is also known as loss function or error. It can be calculated by iteratively running the model to compare estimated predictions against actual values. Therefore, the objective of a Machine learning model is to find parameters or structure that can minimize the cost function. In linear regression, we can use mean squared error(MSE) as a cost-function and in logistic regression, we can use Log-loss function as cost-function. The perfect model would have a log loss of Zero.

Q16: What is the difference between Type I and Type II error? Also, give an example.

Ans16:

This type of question in an interview is just to make them sure that you know the basics very well. Type I error is when we have false positives and Type II error is when we have a false negative. Let me explain it more briefly. Type I error means we are claiming some event has occurred when it hasn't. Type II error means we are claiming some hasn't occurred when it has occurred.

For example, let us suppose there is a final cricket match going on between two teams, say India and Pakistan. After a very serious game, India won. Now there is someone who is claiming Pakistan has won the game which isn't true because India is the winner. This is an example of a Type I error. Now again there is another guy who is claiming India will not get the trophy which is not true because the winner will get the trophy for sure. This is an example of a Type II error.

Q17: Let us suppose there is a hospital who is treating only two types of diseases. They are using a totally different approach for each disease. If the patient suffering from disease 1 treated with the approach used for disease 2, he/she could lose his/her life. They are hiring an analyst to predict which type of disease a patient could probably have. After building the classification model, you observe:

Type I: you predicted yes, but they don't actually have the disease. Type II: you predicted no, but they actually do have the disease. Which type of error you could ignore and you couldn't ignore?

Ans17:

Type I error will not put any patient's life in danger so we can ignore it. But when it comes to Type II error it can put a patient's life in danger, it will be dangerous to ignore. We have to warn the hospital about this error so that they can make some adjustments and be more cautious with these types of patients.

Q18: Can you explain the kernel trick in the Support vector machine?

Ans18:

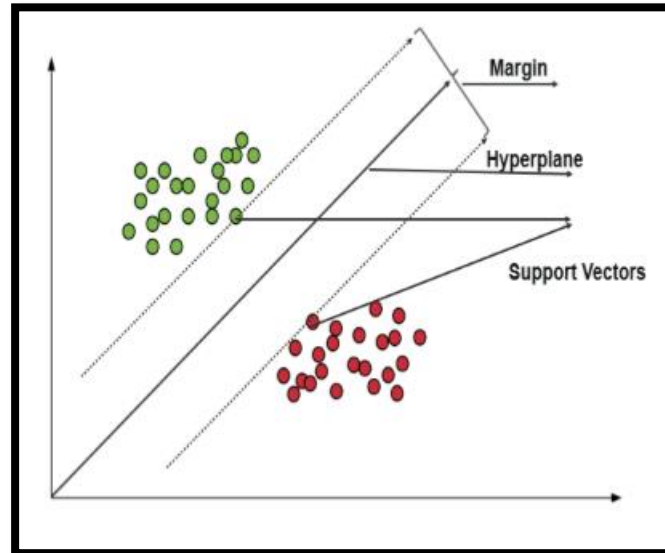
Kernel Trick is a mathematical function when it is applied to the data points. It will find the region of classification between two different classes. We can build a classifier based on the choice of function (it can be linear or radial), which purely depends upon the distribution of data.

Q19: What is Convex Hull? Why is it so important in SVM?

Ans19:

SVM is a supervised model that can solve linear or nonlinear problems. SVM creates a hyperplane (line) which divides the data into classes. From both classes, the data points which are closest to the line are known as support vectors. The distance between a support vector and a line is known as

margin. This algorithm tries to create a decision boundary that has maximum margin and optimal hyperplane. The boundaries of data can be obtained by using the convex hull. The key formation of SVM's is a kernel function, that uses convex hull to choose the extreme points. Its advantages are incremental training, parallel training, and reduced time complexity.

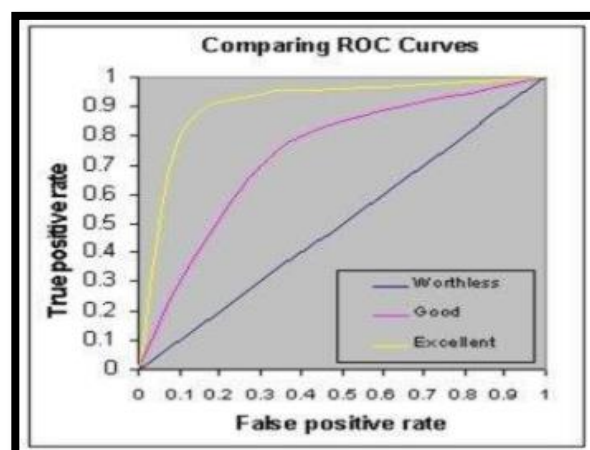


Q20: For what purpose the ROC curve is used?

Ans20:

ROC curve(Receiver Operating Characteristic curve) is a fundamental tool for diagnostic test evaluation and is a plot of the sensitivity (true positive rate) against the specificity (false positive rate) for the different possible cut-off points of a diagnostic test.

- It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
- Closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.
- Slope of the tangent line at a cut point gives the likelihood ratio (LR) for that value of the test.
- The area under the curve is a measure of test accuracy.



Q21: How will you find the best value of k in the KNN algorithm?

Ans21:

We can assign multiple values to k (suppose 0-10) and check its accuracy at every value, the one with the highest accuracy will be the best value for k. Now if we have the same accuracy at multiple values, then we should choose the highest value of k so that noise wouldn't have an effect on it. It is advisable to choose an odd value for k in the case of binary classification.

Q22: How would you predict who will renew their subscription next month? What data would you need to solve this? What analysis would you do? Would you build predictive models? If so, which algorithms?

Ans22:

- Let's assume that we're trying to predict the renewal rate for Netflix subscription. So our problem statement is to predict which users will renew their subscription plan for the next month.
- Next, we must understand the data that is needed to solve this problem. In this case, we need to check the number of hours the channel is active for each household, the number of adults in the household, number of kids, which channels are streamed the most, how much time is spent on each channel, how much has the watch rate varied from last month, etc. Such data is needed to predict whether or not a person will continue the subscription for the upcoming month.
- After collecting this data, it is important that you find patterns and correlations. For example, we know that if a household has kids, then they are more likely to subscribe. Similarly, by studying the watch rate of the previous month, you can predict whether a person is still interested in a subscription. Such trends must be studied.
- The next step is analysis. For this kind of problem statement, you must use a classification algorithm that classifies customers into 2 groups:
 1. Customers who are likely to subscribe next month
 - o Customers who are not likely to subscribe next month
- 2. Would you build predictive models? Yes, in order to achieve this you must build a predictive model that classifies the customers into 2 classes like mentioned above.
- Which algorithms to choose? You can choose classification algorithms such as Logistic Regression, Random Forest, Support Vector Machine, etc.
- Once you selected the right algorithm, you must perform a model evaluation to calculate the efficiency of the algorithm. This is followed by deployment.

Q23: What do you mean by odds and odds Ratio?**Ans23:**

- **Odds:**

As we know that odds of an event happening is defined as the ratio of a likelihood that an event will occur and the likelihood that the event will not occur. Therefore, if A is the probability of an event happening and B is the probability of an event isn't happening, then odds = A /B. But if the probability of A is equal to the probability of B, then odds = A (or odds = B). Both cases are explained below:

Case 1: Odds of rolling three on a dice : $P(E) = 1/6$ (getting 3 on rolling dice) $P(E') = 5/6$ (not getting 3 on rolling dice) Here $P(E) \neq P(E')$ Odds = $P(E)/P(E') = \frac{1/6}{5/6} = \frac{1}{5}$ Therefore Odds = 20%

Case 2: Odds of getting head on the coin: $P(E) = \frac{1}{2}$ (getting head) $p(E') = \frac{1}{2}$ (not getting head) Here $P(E) = P(E')$ Odds = $p(E) = \frac{1}{2}$ Odds = 50%

- **Odds ratio:**

Odds ratio(OR) can be defined as the ratio of the odds of event A in the presence of event B and odds of event B in the presence of event A. It is simply defined as a measure of association between exposure and an outcome. OR should be calculated in case-control studies when the incidence of outcome is unknown. Different OR values hold different meanings.

1. If $OR > 1$, it means the increased occurrence of an event.
2. If $OR < 1$, it means decreased the occurrence of an event. In this case, we should check the CI and P-value for the value of significance level.
3. If $OR = RR$ ($RR =$ Relative Risk). This means the incidence of the disease is $< 10\%$.

For example: Suppose there is a disease spreading in the town of 200 people. The company announced they are providing a free cure for the disease. But they could only provide a cure to 100 people. we have to calculate the odd's ratio.

		P(Disease)		Total
		yes	no	
P(cure)	yes	30	70	100
	no	20	80	100
Total		50	150	200

$$P(\text{disease} | \text{cure}) = 30$$

$$P(\text{disease} | \text{not cure}) = 20$$

$$P(\text{no disease} | \text{cure}) = 70$$

$$p(\text{no disease} | \text{no cure}) = 80$$

Odds for disease = 30/70
 Odds for the cure = 20/80

$$\begin{aligned} \text{Odds ratio} &= \frac{\text{odds for disease}}{\text{odds for the cure}} \\ &= \frac{\frac{30}{70}}{\frac{20}{80}} \\ &= \frac{(30 \cdot 80)}{(20 \cdot 70)} \\ &= \frac{2400}{1400} \\ &= 1.71 \end{aligned}$$

Q24: Can you explain how google is training the data for self-driven cars?

Ans24:

To source labeled data, Google is using reCAPTCHA v3 on storefronts and traffic signs. They are also using the data for training, collected by Sebastian Thrun at GoogleX with his grad students.

Q25: How will you handle an imbalanced dataset? How does it occur?

Ans25:

Imbalanced data sets are one of the common problems for classification where the class distribution is not uniform among the classes. So we can solve this problem by using sampling techniques. There are two basic ways of performing sampling.

- **UnderSampling:** In this technique, we reduce the size of the majority class to match minority class thus help by improving performance with respect to storage and run-time execution, but it potentially discards useful information.
- **OverSampling:** In this technique, we upsample the Minority class and thus solve the problem of information loss, however, we can get into the trouble of having Overfitting.

Apart from this, we have other techniques:

- **Cluster-Based Over Sampling:** As we are all well aware that the K-means clustering algorithm can be independently applied to minority and majority class instances. This is to identify clusters in the dataset. One after the other, each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size.
- **Synthetic Minority Over-sampling Technique (SMOTE):** In this case, a subset of data is taken from the minority class as an example and then new synthetic similar instances are

created which are then added to the original dataset. This technique provides good results when there is a numerical data point.

Q26: Why do we call Naive Bayes Classifier 'so naive'?

Ans26:

The reason behind this is that we call naive Bayes classifier 'so Naive' because it makes assumptions that have the same probability of being correct or not. This algorithm works on an assumption that the presence of one feature will not be affected by the presence of other features i.e, there will be complete independence of features. For example, A vehicle may be considered a Bike if it is black in color and it has two wheels, regardless of the other features. This assumption may or may not be correct because the scooter & cycle also matches the description.

Q27: What is pruning? How can we do it?

Ans27.

Pruning is a technique that helps us to reduce the size of a decision tree. It is used to decrease the complexity of a final classifier and therefore improves the accuracy of a model by reducing the chances of overfitting. Pruning can be done by using a top-up or bottom-up approach. There are two popular pruning algorithms:

- **Reduced Error Pruning:** This is one of the simplest forms of pruning. In this kind of pruning, each node is replaced with its popular class by starting from leaves. We keep the change if the prediction accuracy is not affected. There is an advantage of simplicity and speed.
- **Cost complexity Pruning:** This kind of pruning generates a series of trees as T_0 to T_n . Where T_0 = initial tree and T_n is the final Tree or root. At step k a tree $(k-1)$ is created by removing the subtree and replacing it with a leaf node whose value is chosen the same as the tree building algorithm.

Q28: Running a binary classification tree algorithm is quite easy. But do you know how the tree decides on which variable to split at the root node and its succeeding child nodes?

Ans28:

- Measures such as, Gini Index and Entropy can be used to decide which variable is best fitted for splitting the Decision Tree at the root node.
- We can calculate Gini as following: Calculate Gini for sub-nodes, using the formula: Sum of squares of probability for success and failure (p^2+q^2).
- Calculate Gini for split using weighted Gini score of each node of that split
- Entropy is the measure of impurity or randomness in the data, (for binary class):

$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

Here p and q is the probability of success and failure respectively in that node.

- Entropy is zero when a node is homogeneous and is maximum when both the classes are present in a node at 50% – 50%. To sum it up, the entropy must be as low as possible in order to decide whether or not a variable is suitable as the root node.

Q29: You are provided two separate files that have spam and ham(non-spam) emails. Can you create Spam Filtering using the Naive Bayes algorithm? Explain your answer.

Ans29.

Yes. We have to create a single file that contains all emails whether it spam or non-spam.

- Our first step should be to convert the data into a program understandable format that means numbers. So to do this we can save our file into a list by considering each word as an element of a list. Then we have to remove the words that contain any non-alphabets.
- After this step, we have to remove the duplicates but count the occurrences of each word, so we can use a dictionary after using the count function on each word. We can choose the most common words according to our needs.
- Then we have to use the feature vectorization for turning arbitrary features into the indices of a matrix. Finally, we can use the Multinomial Naive Bayes algorithm.

Q30: Can we use categorical values as predictor variables in Naive Bayes?

Ans30.

It depends on data if we are using Gaussian Naive Bayes then we can use only continuous predictor variables. If we are using Multinomial Naive Bayes, then we can use only categorical predictor variables. We also have a Bernoulli Naive Bayes but in this case, our predictor variables can only have boolean values.

Q31: What is the difference between ID3, C4.5, and CART?

Ans31. C4.5 is the extension of ID3 so that it can use both continuous and categorical values

ID3	CART
It stands for Iterative Dichotomiser 3.	It stands for Classification and regression tree.
Used when the target has categorical values.	Used when the target has either categorical or continuous values.
Uses Entropy and information gain for feature selection.	Uses Gini impurity (Gini formula) for feature selection.
Follows a greedy approach to reach the goal.	Suffers from problems like greediness and instability.
It can construct multiple trees in each iteration.	It can only construct a binary tree in each iteration.

Q32: How can you choose the optimal number of k in k-means clustering?

Ans32:

We have to choose the k that has minimum inter-cluster variation or total within sum of square(WSS). WSS is used to find the compactness of clustering. There is no hard and fast rule to calculate the value of k that has minimum WSS value but we have some methods by which we can make the approximation about the best value of k. One of the ubiquitous methods is the Elbow method.

Elbow Method: We will provide it with a different number of k values(say 1-10). This method will try to find the total WSS for given k values. But the number of clusters should be chosen so that adding another cluster doesn't improve total WSS. Wherever the variance of WSS stops dropping significantly, that will be chosen as the optimal value for k.

Q33: What is the difference between Gini Impurity and Entropy in a Decision Tree?

Ans33:

- Gini Impurity and Entropy are the metrics used for deciding how to split a Decision Tree.
- Gini measurement is the probability of a random sample being classified correctly if you randomly pick a label according to the distribution in the branch.
- Entropy is a measurement to calculate the lack of information. You calculate the Information Gain (difference in entropies) by making a split. This measure helps to reduce the uncertainty about the output label.

Q34: What is the difference between Entropy and Information Gain?

Ans34:

- Entropy is an indicator of how messy your data is. It decreases as you reach closer to the leaf node.
- The Information Gain is based on the decrease in entropy after a dataset is split on an attribute. It keeps on increasing as you reach closer to the leaf node.

Q35: Why Unsupervised learning uses Generative models instead of discriminative models?

Ans35:

The Discriminative models(SVM, logistic regression, nearest neighbors, etc) will only learn the difference between two categories of data but, Generative models (eg: naive Bayes, Bayesian networks, etc) will learn multiple categories of data because it learns the joint probability distribution $p(x,y)$. It uses Bayes Theorem to predict conditional probability, while a Discriminative model learns the conditional probability distribution $p(y|x)$. Let me tell you a story to make this thing more clear. One day the king took a walk with his two sons and showed them two animals a horse and the Tiger. After a few days, he again took a walk with his sons and showed them a horse, and asked which animal is it? One son visualizes a picture of what he saw the other day and compared it with the

animal which is in front of him. Based on the closed match he took his decision and said it is a horse. The other son only knows the different physical properties(color, size, etc.) between them, so he examined their properties and said it is a horse. Both of them identified it right but used different approaches. The first person's approach is the same as the Generative model and the other one's approach is the same as the discriminative approach.

Q36: How are Linkages done in hierarchical clustering?

Ans36:

Hierarchical clustering algorithm groups the data on the basis of their similar characteristics. The distance between each cluster can be measured using different methods which are known as linkage or close. The four most popular among them are:

- **Single Linkage:** In this type of linkage, the distance between the two clusters is defined as the shortest distance between two points in each cluster. That is why this type of linkage is also known as Minimum linkage. Sometimes it can produce clusters where the points in different clusters are closer than to points within their own clusters. These clusters can appear spread-out.
- **Complete Linkage:** In this type of linkage, the distance between the two clusters is defined as the longest distance between two points in each cluster. That is why this type of linkage is also known as Maximum linkage. This method usually produces tighter clusters than single-linkage, but these tight clusters can end up very close together.
- **Average Linkage:** In this type of linkage, the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster. That is why this is also known as average linkage. In other words, we can say where the distance between each pair of observations in each cluster are added up and divided by the number of pairs to get an average inter-cluster distance.
- **Centroid Linkage:** In this type of linkage, the distance between two clusters is defined as the distance between the centroids of two clusters when the centroids move with new observations, there are the chances that the smaller clusters are more similar to the new larger cluster than to their individual clusters which causes an inversion in the dendrogram. Since clusters being merged will always be more similar to themselves than to the new larger cluster. That's why this problem doesn't arise in the other linkage methods.

Q37: How is K-means different from KNN?

Ans37:

K-Nearest Neighbor	k-Means Clustering
Supervised technique	Unsupervised Technique
Used for Classification or Regression	Used for Clustering
'K' in KNN represents the number of nearest neighbors used to classify or predict in case of continuous variable/regression	'k' in k-means represents the number of clusters the algorithm is trying to identify or learn from the data.

Q38: Can you explain the target imbalance?

Ans38.

This kind of problem occurs in Clustering Algorithms. If the count of one value in a target column is larger when compared to the count of other values, then we can say it has target imbalance.

For Example: our target column is

[0,1,1,1,1,1,2,1,2,1,1,1,2,0,1,1,2,1,1,1,1,2,1,2,1,1,1,1,1,1,0,1,1,1,0,0,1,1,1,1]. If we count the values, our result will be 0-->5 times, 1-->30 times, 2-->6 times. It is clear that the count of 1 is more when compared to the count of 0 and 2. This is an example of a target imbalance.

Q39: What is DBSCAN?

Ans39:

It stands for the density-based clustering of applications with noise. DBSCAN is a partition-based algorithm that doesn't have the concept of outliers. This technique is used to find the arbitrary shaped cluster or clusters within the clusters because traditional clustering might not be able to provide good results in such cases. For example, we want to cluster the customers into two groups, whether we should or shouldn't provide the loan on the basis of income and also predict whether the customers who will pay back the loan or not.

Q40: What are Association rules? When do you use it?

Ans40:

At a basic level, association rule involves the use of machine learning models to analyze data for patterns, or co-occurrence, in a database. It is used to identify frequent if-then associations, which are called association rules. The item which is found within the data is known as Antecedent and the item

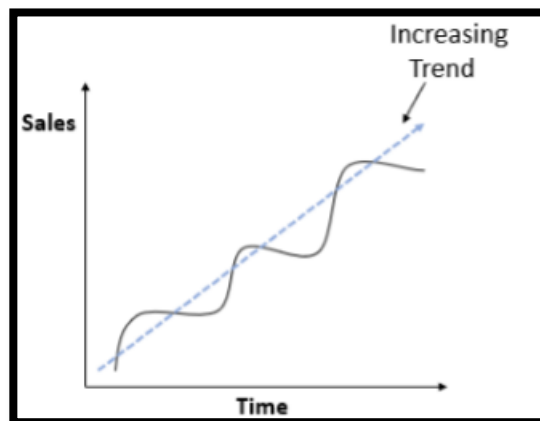
which is found in combination with Antecedent is known as consequent. Example: In a store, all food items are placed in the same path, vegetables on one side and fruits on other, all dairy items are placed together and cosmetics form another set of such groups. The reason for doing this thing is to help the stores for the cross-selling process because it doesn't only save the precious time of a customer but also reminds the customer what relevant items he/she should buy. Association rules help uncover all such kinds of relationships between items from huge databases. These Rules don't draw out an individual's preference, but they can find relationships between a set of elements of every distinct transaction. This is the reason they are different from collaborative filtering. The strength of the association between the two is defined by the various metrics, mentioned below:

- **Support:** These metrics are used to give us information about how frequent an itemset is in all the transactions.
- **Confidence:** These metrics are used to give us information about the likeliness of occurrence of consequent on the cart given that the cart already has the antecedents.
- **Lift:** These metrics are used to compare the confidence with the actual confidence.

Q41: Explain the terms Trend, Seasonality and Cyclicity of a time series?

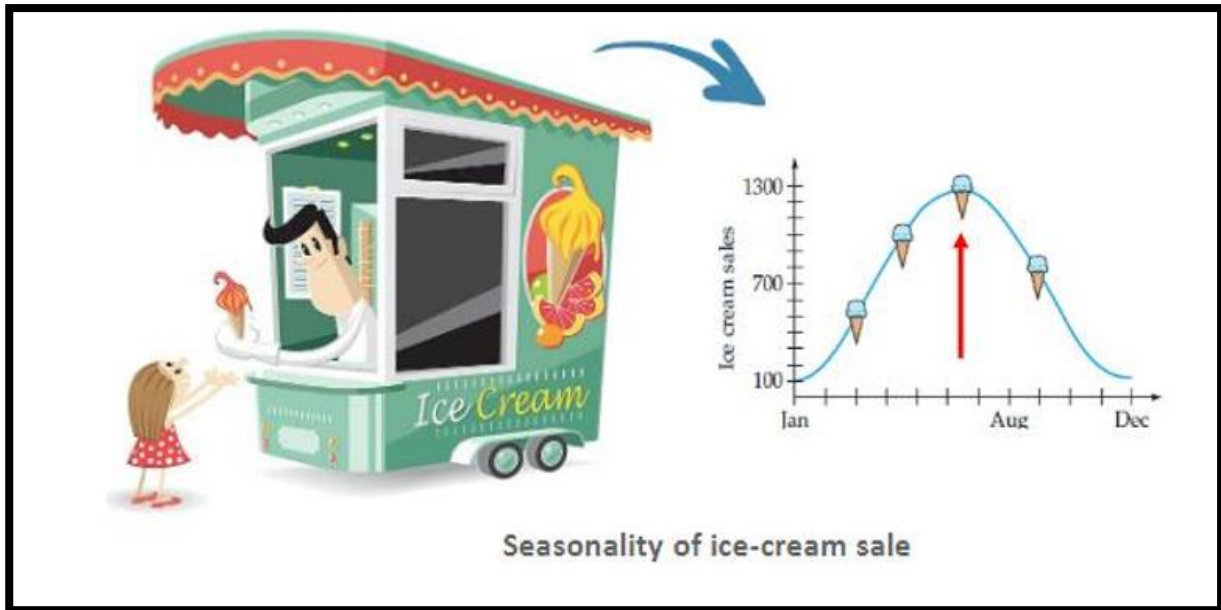
Ans41:

- **Trend:** It is that component of a time series that represents only variations of lower frequency, the medium and high-frequency variations being filtered out. Trends are normally observed in long term or cyclical contexts. Refer the below graph for more understanding.



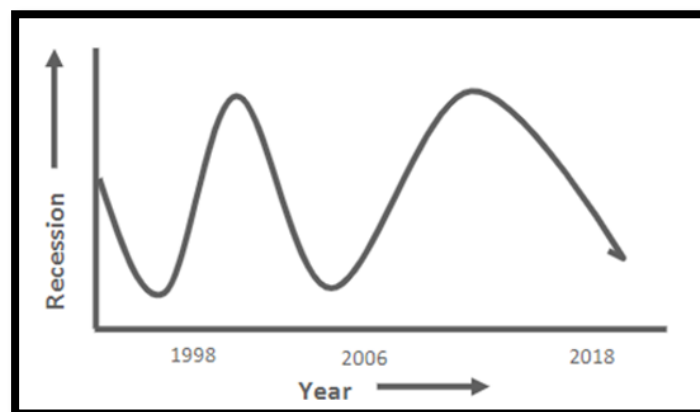
Here, the blue line represents an upward / increasing trend. It can be said that the prices of the stocks seem to show an overall increase/ increasing trend over the years. Similarly, we can observe linear, damped or exponential trends in time series. A time series needs to be detrended, in order to be considered for further analysis.

- **Seasonality:** It is a characteristic of a time series in which the data experiences regular and predictable changes within a fixed and known period. It can be usually observed as a repeating pattern over a specific time frame, typically a year in the time series plot. That is the reason time series is also known as periodic time series. For example, the sales of ice cream show a rise during the summers every year.



Analyzing seasonal patterns can greatly help businesses manage their inventories, staffing and making other key decisions. It can also help investors to minimize risks by understanding the correct time of the year/ season to make their investments and maximize profits.

- **Cyclicality:** A cyclic pattern exists when the data exhibits rises and falls that are not of fixed period (e.g. a country experiences economic boost for 4 years, and then a decline for the next 4 years, and again a boost for next 8 years, followed by a decline in the next 5 years). The period of rise and fall is not fixed and keeps changing with time.



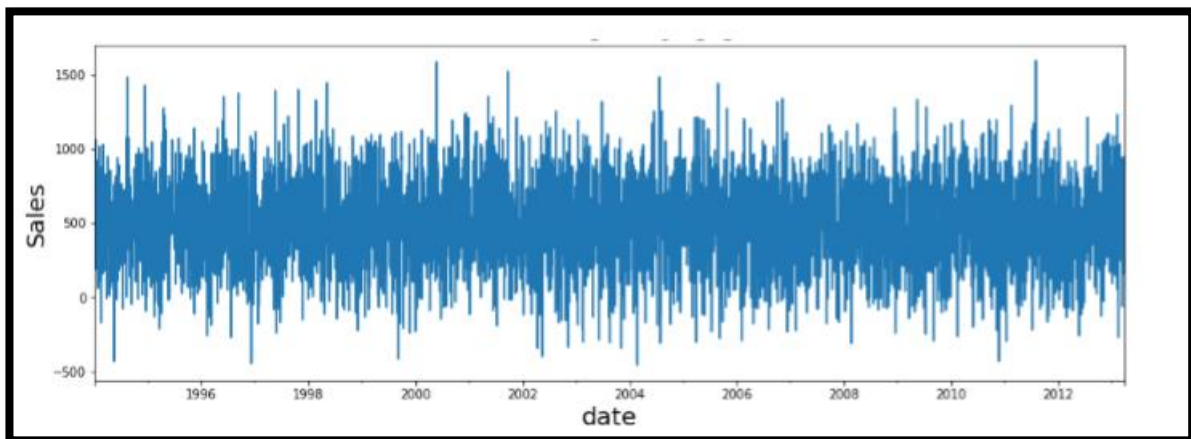
Q41: What is White Noise? What are its characteristics?

Ans41:

White noise is a special type of time series where the data doesn't follow a pattern. It is unpredictable. In order to consider a series as white noise, the following 3 conditions need to be satisfied :

- Constant mean
- Constant Variance
- No autocorrelation in any period, i.e., the future values have no dependency with past values and are totally uncorrelated.

Thus, white noise is a sequence of random data. We can say it behaves “sporadically”, so there is no way to successfully project it into the future. Below is an example of how a white noise time series looks like:



Q42: Why do we need to convert a time series date column to a Date time format? How can you convert it?

Ans42.

To understand this, we consider a dataset having a date column. First of all, we will look at the values and the datatype of those values. We can look at the following code snippet :

```
In [18]: # We will observe the first 5 rows of the 'date' column
dataset.date.head()

Out[18]: 0    07/01/1994
         1    10/01/1994
         2    11/01/1994
         3    12/01/1994
         4    13/01/1994
         Name: date, dtype: object
```

The first date is 7th January 1994. The datatype is ‘object’. This datatype is interpretable by humans, but for the machine to understand it as a DateTime variable, it must first be converted to DateTime format using DateTime library. If not converted, it will treat it as a string object, and cannot be used for further analysis.

The following code snippet shows how to convert to DateTime format :

```

In [19]: # performing datetime conversion
import datetime
dataset.date=pd.to_datetime(dataset.date)

# now checking the type of the date column and the first 5 rows
dataset.date.head()

Out[19]: 0    1994-07-01
         1    1994-10-01
         2    1994-11-01
         3    1994-12-01
         4    1994-01-13
         Name: date, dtype: datetime64[ns]

```

We can use “to_datetime” to convert it to DateTime format. The type of the variable has now become “datetime64”, and will now be correctly interpreted by the machine, and can be used further for time series modeling.

Q43: What is random walk?

Ans43.

A random walk is a special type of time-series, and the difference between consecutive periods are simply white noise. It is often confused with white noise. It is different from white noise as white noise is just a sequence of random numbers, and future values have no dependence on past values. But in a random walk, although the future values cannot be exactly predicted, the best estimators of present values are the values at the time period just preceding it.

If, P_t -> Value at time t

P_{t-1} -> Value at time $t-1$

E_t -> Residuals

Then for a random walk,

$$P_t = P_{t-1} + E_t$$

The residuals are arbitrary and cannot be predicted. This suggests that the best estimators for prices today, are the prices yesterday. It can be visualized as the walking pattern of a “drunkard person”. Below is an example of how a random walk time series looks like. The green line represents a random walk, and the one in red is not a random walk.



If a time series resembles a random walk, the future values cannot be predicted with great accuracy.

Q44: What is autoregression? Explain the AR model?

Ans44.

A time series is said to be autoregressive if the current value (Y_t), can be expressed in terms of its previous 'p' lags (Y_{t-1} , Y_{t-2} ,..... Y_{t-p}), i.e, present values are a weighted average of its past values. AR model is a time series model that uses linear regression to express future values based on past observations. AR model is dependent on 'p' (lagged values | past values), and it is denoted by "AR(p)". 'p' is the signature of the AR model.

$$Y_t = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + \varepsilon_t$$

Where,

- Y_t =function of different past values
- $Y_t = f(Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots)$
- p=past values
- C=constant or intercept
- β =co-efficient of each parameter p
- ε_t =errors in time

Q45: How ACF is different from PACF. How are they used?


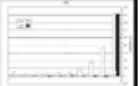
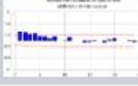
Ans45:

- **ACF** stands for Autocorrelation function. It is a mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals. It is the same as calculating the correlation between two different time series, except that autocorrelation calculates the correlation between a time series with a lagged version of itself.
It can also be called lagged correlation / serial correlation as it measures the relationship between a variable's current value and its past values. It can be used to find the number of MA (Moving Average) terms that are needed (the size of the moving average window).
- **PACF** stands for Partial autocorrelation function. It is a summary of the relationship between an observation in a time series with observations at prior time steps, with the relationships of intervening observations removed. It is used to determine the lag order (value of AR term) for the ARIMA model.

Q46: When to use AR, MA, ARMA model?**Ans46:**

All the 3 models can only be used when the series is stationary. If the null hypothesis for the ADF test is rejected, then the series is stationary. These models are not applicable when the series is non-stationary.

The table below summarizes when each of these models can be used:

Categories	ACF(Auto Correlation Function)	PACF(Partial Auto Correlation Function)	Graph
AR(Auto Regressive)	Geometric	Significant till p lags	 ACF=Geometric PACF=P
MA(Moving Average)	Significant till q lags	Geometric	 ACF=P PACF=Geometric
ARMA(AR+MA)	Geometric	Geometric	 ACF=Geometric PACF=Geometric

Q47: What is differencing of time series? Why is it applied to a time series?**Ans47:**

Differencing of a time series is subtracting the time series with lagged versions of itself. Differencing is performed to remove non-stationary components like trend and seasonality from the time series and make it stationary. The number of times the series needs to be differenced until it becomes stationary is called the order of differentiation (the parameter 'd' of the ARIMA model). Differencing of time series is performed by subtracting current observation with its previous observation: $\text{difference}(t) = \text{observation}(t) - \text{observation}(t-1)$.

Q48: Explain the reason to stationarize a time series before using it for modeling?**Ans48:**

Time series modeling techniques like AR, MA, ARMA are built on the assumption that the time series data is stationary, i.e. its statistical properties like mean and variance remain constant throughout. The presence of trends and seasonal effects make the time series non-stationary and makes it difficult to model. Hence, we need to stationarize a time series before using it for modeling techniques. Differencing can be used to stationarize a time series.

Q49: What is a stationary data series? How can we check for it?**Ans49:**

A stationary series is one in which consecutive samples of data of the same size should have identical covariances, regardless of the starting point. This characteristic of the data is also known as weak form stationarity or covariance stationarity. An example of weak form stationarity is white noise. The assumptions of covariance stationarity are:

- constant mean
- constant variance
- consistent covariances between periods that are at identical distances from one another.

This can be expressed as,

$$\text{COV}(X_n, X_{(n+k)}) = \text{COV}(X_m, X_{(m+k)})$$

The **Dickey-Fuller test** is used to test whether a dataset comes from a stationary process. An augmented version of this test, called ADF(Augmented Dickey-Fuller test) is used for a time-dependent dataset.

H0: Null hypothesis for this test assumes non-stationarity

H1: Dataset is stationary

If, test-statistic < Critical value, H0 is rejected.

The following code is used:

```
import statsmodels.tsa.stattools as sts
sts.adfuller(dataset.column_name)
```

Below is the output of this test, when performed on White Noise data:

```
Out[120]: (-71.29550344322298,
           0.0,
           0,
           5020,
           {'1%': -3.431653316130827,
            '5%': -2.8621159253018247,
            '10%': -2.5670765656497516},
           70910.14554022666)
```

The test statistic (-71.29) is less than the critical value at all 3 levels of significance. Also, the pvalue is close to 0. Hence we reject the null hypothesis. And we can say that White noise is stationary.

Q50: Why are heat maps important in machine learning?**Ans50:**

It is a graphical representation of data where values represented by colors. It makes easier to understand more complex data. This is one of the most important concepts of machine learning because it is used to check the correlation of variables with each other. If we have a dataset with a higher number of columns, it will become very difficult to understand the correlation between variables.

Q51: Why should we divide our data into a training set, Test set & Validation set? What is the difference between them?**Ans51:**

Whenever we are training a model, we should divide the available data into three separate sets. why? I will explain.

- The first one is a training dataset. The machine learning model uses this part as a dataset to learn. But if we train a model on a whole dataset, we will end up overfitting the data.
- The second one is the Test dataset, this is a part which is used to test the accuracy of our model, just to make sure how well it performs when we use the model on real-life data.
- The third one is a Validation dataset, this is a part which we use to test how well our model is performing. Confused Right? We have various classification algorithms like KNN, decision trees, Logistic Regression, Support Vector Machine(SVM), Random forests, etc but how can we calculate the model performance and if a model is performing well, does it overfit? To solve this kind of problem, we choose the validation set. Note: So if we omit the test set and only use a validation set, the validation score won't be a good estimate of the generalization of the model.

Q52: How does Cross-validation work?**Ans52:**

It's a model validation technique for assessing how the results of a model will generalize to an independent dataset. It is mainly used when our goal is prediction, and we want to estimate the accuracy of a predictive model. The cross-validation is used to define a data set to test the model in the training phase in order to overcome the problems like overfitting, underfitting and to find out how the model will generalize to an independent data set. Its working depends on the validation strategy we use. We have different types of validation strategies based on the number of splits being done in a dataset.

- **K fold:** K-fold divides the total number of samples into k numbers of groups (called folds) of equal size. The prediction function is learned using folds, and the fold left out is used for the test. In the end, it returns the average accuracy of a model which can help us with model

evaluation. For example, we have a dataset with 15 rows after choosing $n\text{-splits} = 3$. It will work in 4 steps. It can choose the first 5 rows as a test set and remaining as a train set and calculate its accuracy. Then it will choose the next 5 rows for the test set and remaining for the training set, and calculate its accuracy, the same will go with the next 5 rows. After these three steps, it will calculate the average of all the accuracy. This average accuracy will be the final accuracy of the model.

- **Leave one out (Loo):** Loo is the special case of k-fold when the total number of samples is equal to the number of splits. Each training set is created by taking all the samples except one, the sample which is left out is test set. In this case, it will iterate through every sample in our dataset each time using the k-1 object as train samples and 1 object as a test set.
- **Stratified K fold:** StratifiedKFold is a variation of k-fold which returns stratified folds instead of k folds in which each set contains approximately the same percentage of samples of each target class as the complete set.

Q53: What is A/B Testing?

Ans53:

- A/B is Statistical hypothesis testing for a randomized experiment with two variables A and B. It is used to compare two models that use different predictor variables in order to check which variable fits best for a given sample of data.
- Consider a scenario where you've created two models (using different predictor variables) that can be used to recommend products for an e-commerce platform.
- A/B Testing can be used to compare these two models to check which one best recommends products to a customer.



Q54: What are out of sample accuracy and in sample accuracy?**Ans54:**

Whenever we build a model we train it on a trainset. The accuracy we receive when we test our model on unseen data (test set) is known as out of sample accuracy. And the accuracy we receive when we train our model on the same trainset is known as in sample accuracy. Our goal should be to obtain a model that has the highest out of sample accuracy.

Q55: Why do we use the chi-square test?**Ans55:**

This is one of the important concepts of machine learning because this test can be used to find the relationship between two categorical variables. In simple words, it can be used to figure out the dependency of the target variable on the independent variable. On the statistical level, this test has 3 parts: contingency table(to summarize the relationships), pearson's chi-square test(finds significance level, p-value), and hypothesis testing. If $p\text{-value} \leq \text{significance level}$, then they are dependent otherwise they are independent.

Q56: How is Grid search different from Random search?**Ans56:**

- **Grid Search** : Grid search trains the network for every combination by using the two sets of hyperparameters, learning rate and the number of layers. Then evaluates the model by using Cross-Validation techniques.
- **Random Search**: It randomly samples the search space and evaluates sets from a particular probability distribution. For example: instead of checking all 10,000 samples, randomly selected 100 parameters can be checked.

Q57: What is the curse of dimensionality? How can you deal with it?**Ans57:**

The curse of dimensionality is when we have training data that has many numbers of features(columns), but the dataset does not have enough samples(rows) from which it can learn properly. Let us suppose, we have a training dataset of 100 samples(rows) and with 100 features(columns). It will be very hard to learn from it because the model will find random relations between the columns and the target. However, if we had a dataset of 100k samples with 100 features, then the model has a very high chance to learn the correct relationships between the columns and the target. We have different options available by which we can solve this problem:

- **Dimensionality reduction:** We have many techniques that allow us to reduce the dimensionality of the features. For example Feature selection, Principal component analysis (PCA) and autoencoders.
- **Feature selection:** Instead of using all the features, we can train our model on a smaller subset of features.
- **PCA:** It converts the possible correlated variables into linearly uncorrelated variables.
- **L1 regularization:** Since it produces sparse parameters, L1 regularisation helps to deal with high-dimensionality input.
- **Feature engineering:** We create new features that sum up multiple existing features. For example, if we have one feature 'number of siblings' and the other is the 'number of children'. With these two features, we can create a new feature 'family' which will be the sum of both.

Q58: Explain ANOVA.

Ans58:

ANOVA stands for Analysis of variance. It is a statistical technique that estimates the potential differences in a scale-level dependent variable by a nominal-level variable having 2 or more categories. For example, an ANOVA can examine potential differences in IQ scores by Country (USA vs. UAE vs. India vs. China). This is the expansion of T and the Z test which has a problem of only allowing the nominal level variable to have two categories. This test is also called the Fisher analysis of variance. ANOVAs are further classified in three ways, namely; one-way ANOVA, two-way ANOVA, and Nway ANOVA.

- One-Way ANOVA is just one independent variable. For example, differences in IQ can be assessed by Country, and Country can have 4, 24 or more different categories to compare.
- Two-Way ANOVA uses two independent variables. If we will take the same above example, a 2-way ANOVA can examine differences in IQ scores (the dependent variable) by nationality (independent variable) and Gender (independent variable), it can be used to examine two independent variables interact with each other. This type of ANOVA is also known as factorial ANOVA.
- N-Way ANOVA refers to using more than two independent variables.

Q59: What is p-value?

Ans59:

When you perform a hypothesis test in statistics, a p-value can help you determine the strength of your results. p-value is a number between 0 and 1. Based on the value it will denote the strength of the results. The claim which is on trial is called the Null Hypothesis.

Low p-value (≤ 0.05) indicates strength against the null hypothesis which means we can reject the null Hypothesis. High p-value (≥ 0.05) indicates strength for the null hypothesis which means we can accept the null Hypothesis p-value of 0.05 indicates the Hypothesis could go either way.

To put it in another way, High P values: your data are likely with a true null. Low P values: your data are unlikely with a true null.

Q60: Why do we prefer PCA instead of autoencoders for dimensionality reduction?

Ans60:

It depends on data if the features have a linear relationship with each other then PCA is best but if there is a non-linear relationship then auto-encoders are capable of modeling that function. PCA is faster and computationally cheaper than autoencoders. Since there are a high number of parameters in autoencoder, it is prone to overfitting. So if we use a single layer auto-encoder with the linearly activated function, it will be very similar to PCA.

Q61: Why should we standardize the data before using PCA?

Ans61:

Generally, PCA is used to calculate a new projection of the dataset. We should centralize the data so that we could bring the means to the origin(mean=0). We have to make sure that PCA is in the direction of max variance. To calculate the correct mean squared error, it is required to centralize the data otherwise it can misguide us.

Q62: What is bootstrap Aggregating. How is it done?

Ans62:

Bootstrap Aggregating (also known as bagging) is an ensemble method in which the dataset is first divided into multiple subsets through resampling. Then, each subset is used to train a model, and then the last predictions are made through voting or averaging the component models. Bagging is an ensemble method that follows a parallel approach.

Q63: What is Out of Bag Error and why is it required?

Ans63:

You might know about the bootstrap sample which is a smaller sample,i.e, extracted from a larger sample. Bootstrapping is a type of resampling where a large number of same-sized smaller samples are repeatedly drawn, with replacement, from a single original sample. For each bootstrap sample, there is one-third of data that was not used in the creation of the tree, i.e it was out of the sample. This kind of data is referred to as out of bag data. The out of bag data is passed for each tree and the outputs are aggregated to give out of bag error. Out of bag error is used to get an unbiased measure of

the accuracy of the model over test data. While estimating the error, this percentage is quite effective in the testing set and does not require any more cross-validation.

Q64: How is boosting different from bagging?

Ans64:

Bagging	Boosting
Used to decrease the variance in the prediction.	An iterative technique that adjusts the weight of observation.
Weak learners are produced in parallel.	Weak learners are produced sequentially.
Elements have the same probability to appear in a new dataset.	Some elements will appear more often in a new dataset.
Solve the problem of overfitting.	Chances to increase the problem of overfitting.
Weak learners are equally weighted.	Some weak learners have more weight than others.

Q65: What is the difference between stochastic gradient descent (SGD) and gradient descent (GD)?

Ans65:

Both of them are used to find a set of parameters that can minimize a loss function by evaluating parameters against data and then making some adjustments. In the case of standard gradient descent, the machine will evaluate all training samples for each set of parameters. This is similar to taking big/slow steps toward the solution. In stochastic gradient descent, the machine will evaluate only 1 training sample for the set of parameters before updating them. This is similar to taking small, quick steps toward the solution.

Q66: Why Extreme gradient boosting performs better than gradient boosting?

Ans66:

XGboost is a definitive way for the implementation of the gradient boost method because it uses a more accurate technique to find the best model. The most important techniques it uses are second-order derivative and advanced regularization techniques (L1 & L2). It has an additional advantage because its training speed is very fast.

Scenario-Based Questions

Q67: Let's suppose you have been given a training dataset having 1500 columns and 10 lakh samples. You have to use a classification algorithm to find a solution. Before doing so, you have to reduce the dimension of this data so that the computation time of a model can be reduced. Because of memory issues in your machine, you can't just load the complete dataset into memory. How will you do this task using your machine?

Ans67:

It is a very difficult task to process high dimensional data on a low memory machine. We have various methods to tackle such kind of situation:

- a. We should close all applications in our machine so that we can use most of the memory because they are loaded in RAM.
- b. We can a smaller dataset by using random sampling on the data.
- c. We can separate the numerical and categorical variables and remove the correlated independent variables. For numerical variables, we'll use correlation. For categorical variables, we'll use the chi-square test.
- d. Principal component analysis (PCA) is a dimensionality reduction technique that can help us to identify patterns and find correlations in a dataset. So that when it is transformed into a dataset having lower dimensions, there shouldn't be any loss of important information.
- e. We can build a linear model using Stochastic Gradient Descent.

Q68: Suppose you are given a data set that has missing values spread along 1 standard deviation from the median. What percentage of data would remain unaffected and Why?

Ans68:

Since the data is spread across the median, let's assume it's a normal distribution. As you know, in a normal distribution around 68% of the data lies in 1 standard deviation from the mean (or mode, median), which leaves around 32% of the data unaffected. Therefore, ~32% of the data would remain unaffected by missing values.

Q69: Suppose there is a company that provides the services to their customers (eg telecom) at very low rates for a couple of years. Now they want to increase their rates to make some profit. So, they came up with the idea to provide one of the 3 packs (p1,p2,p3) to their customers on the basis of their descriptive data. To do so, they are hiring you to build a model that can make their task easier to target their customer by providing them the affordable pack which they can subscribe willingly. Which clustering algorithm do you use? Why do you prefer it over other clustering algorithms?

Ans69:

We have three different types of clustering: Exclusive clustering, Overlapping clustering, and Hierarchical clustering. But, which one will be more suitable here? Since each type of customer can have only one type of pack, means each point must be belonging from one cluster only. For example, customer types c1,c2,c3 can have p1,p2,p3 respectively. So, this is a case of Exclusive clustering. Therefore we can use hard clustering algorithms like k-means.

Now if they would have been providing multiple packs to some customers. For example, Customer type c1,c2 can have only p1,p2 respectively. But customer type c3 can have both p1&p2. In that case, we could have used soft clustering techniques like Fuzzy-C-means. Now when should we use hierarchical clustering, For example, there is a pack p1 which is common for all customers but p2 can be provided to customer type c1 only and p3 can be provided to customer type c3 only. In such cases, we can use hierarchical clustering.

Q70: A company has created a new website for movie/video streaming. They are hiring you to create a recommendation system that can suggest movies/videos to the user. What will your approach for this task?

Ans70:

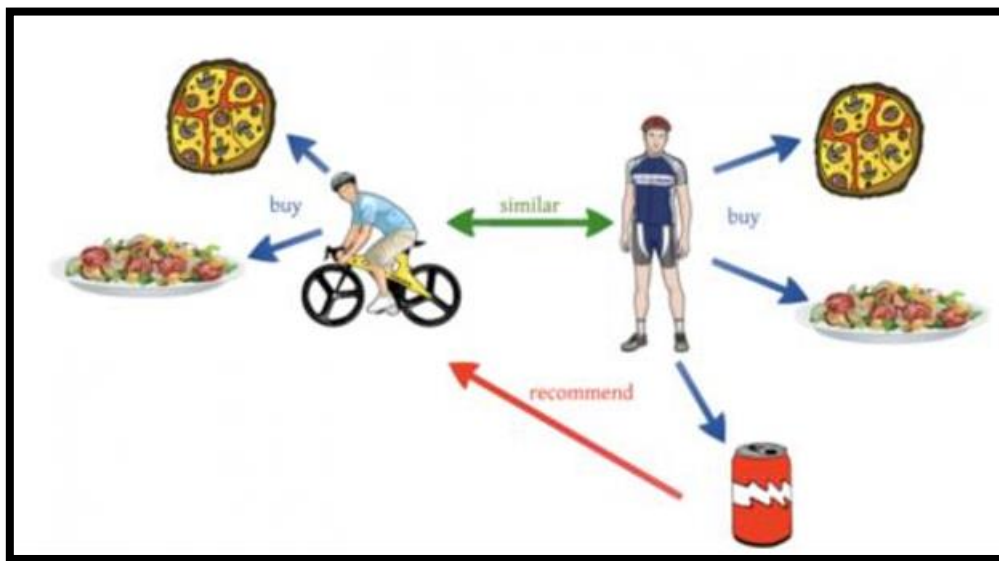
Generally, it depends on what we want to recommend to the user. Since this is a case of movie recommendation, we can go with a collaborative recommendation system. Because in the case of movies users always want to try something new only then he/she can decide whether it comes up with his/her expectation or not. But it shouldn't be like we will end up suggesting something which is totally different from his/her taste. If we discuss movie streaming, its working is simple.

- Let 's suppose there is a user(a) who is watching the movies that have Genre (action, Sci-fi). • There is another user(b) who is watching movies that has a Genre (Action, Romance, Sci-fi).
- So a collaborative recommendation system will suggest movies that have Genre (Romance) to the user(a). While in the content-based recommendation system, the user will only receive the suggestion that has the same Genre as the user has watched, means the user(a) will only receive suggestions that have Genre (Action, Sci-fi) in the content-based recommendation system.

Q71: 'People who bought this also bought...' recommendations seen on Amazon is based on which algorithm?

Ans71:

E-commerce websites like Amazon make use of Machine Learning to recommend products to their customers. The basic idea of this kind of recommendation comes from collaborative filtering. Collaborative filtering is the process of comparing users with similar shopping behaviors in order to recommend products to a new user with similar shopping behavior.



To better understand this, let's look at an example. Let's say a user A who is a sports enthusiast bought, pizza, pasta, and a coke. Now a couple of weeks later, another user B who rides a bicycle buys pizza and pasta. He does not buy the coke, but Amazon recommends a bottle of coke to user B since his shopping behaviors and his lifestyle is quite similar to user A. This is how collaborative filtering works.

Q72: You are asked to build a multiple regression model but your model R^2 isn't as good as you wanted. For improvement, you remove the intercept term now your model R^2 becomes 0.8 from 0.3. Is it possible? How?

Ans72:

Yes, it is possible.

- The intercept term refers to model prediction without any independent variable or in other words, mean prediction $R^2 = 1 - \frac{\sum(Y - Y')^2}{\sum(Y - Y \text{ Mean})^2}$ where Y' is the predicted value.
- In the presence of the intercept term, R^2 value will evaluate your model with respect to the mean model.
- In the absence of the intercept term ($Y \text{ Mean}$), the model can make no such evaluation,
- With a large denominator, the value of $\frac{\sum(Y - Y')^2}{\sum(Y)^2}$ equation becomes smaller than actual, thereby resulting in a higher value of R^2 .

Q73: Let us suppose you have built a random forest model with 30000 trees. You were surprised after getting a Zero training error. But, the validation error is 38. What is the issue here?

Ans73:

There is a simple reason for this, our model is overfitted. Training error 0.00 means the classifier has completely mimicked the training data patterns when we are testing it on unseen data, its unable to find that patterns and returns a prediction with a high error. In a random forest, it happens when we use a larger number of trees than necessary. Hence, to avoid these situations, we should tune the number of trees using cross-validation.

Q74: Let us suppose that your model is suffering from high variance and low bias. Which kind of algorithm should you use? And Why?

Ans74:

First, we will focus on Low bias which occurs when the predicted values of the model are nearly equal to actual values. In other words, we can say that the model becomes flexible enough to mimic/remember all the distribution of the training data. Well, it looks like a great achievement, but there is a catch which we shouldn't forget when it comes to a flexible model that has no generalization capabilities. So this is the reason when we test our model on unseen data, it gives disappointing results. When we have a situation like this.

To solve the Low bias:

- We can use a bagging algorithm like the Random forest to tackle the problem of high variance. These algorithms divide a dataset into subsets made with repeated randomized sampling.
- Then, we use these samples to generate a set of models using a single learning algorithm. Later, the model predictions are combined using classification (voting) or regression (averaging) algorithms.

To solve high variance:

- We can use regularization techniques, where higher model coefficients get penalized, hence lowering model complexity.
- We can use top n features from a variable importance chart. Maybe, with all the variables in the data set, the algorithm is having difficulty in finding a meaningful signal.

Q75: Suppose you have been provided a dataset from a school that contains 40 features. These features hold information about students and the result of those who passed or failed the exams. Now the school has decided not to give admission to students randomly. So they choose you to analyze the dataset carefully and build the model which can help them to make the decision so that the maximum number of students will pass the exam from their school. After building the model, you have to explain to them why you choose this model how can it help them. This model has 30 features which contain continuous variable and rest are categorical values.

Ans75:

Step 1: You have to analyze the dataset and find out what kind of information is stored by each column. Also, remove the null values.

Step 2: Find out the column which holds information about the result of a student, this column should hold two values pass or fail. If not you have to use data binning to change the number of unique values in the column.

Step 3: Now you have to find out the columns on which the resulting column is highly dependent. Since these are forty columns, you can separate the dataset into two parts. One which will hold continuous values and the other will hold the categorical values.

Step 4: You will visualize the continuous values using a distribution plot & box plot, to check the normality of data and check the outliers. To proceed further my data should be uniformly distributed (distplot) and having fewer outliers (boxplot). And to visualize the categorical values, you can use countplot and try to convert categorical values into discrete ones.

Step 5: Now use the ANCOVA test to find the relation between continuous and categorical variables. And to find the relation between categorical values you can use the chi_square test.

Step 6: As we know Logistic Regression is best when it comes to binary classification. But still, you can use model evaluation like cross-validation to avoid overfitting and find the best classification model.

Step 7: The Last step will be the prediction.

Q76: You are given a data set consisting of variables having more than 30% missing values? Let's say, out of 50 variables, 8 variables have missing values higher than 30%. How will you deal with them?

Ans76:

- Assign a unique category to the missing values, who knows the missing values might uncover some trend.
- We can remove them blatantly.
- Or, we can sensibly check their distribution with the target variable, and if found any pattern, we'll keep those missing values and assign them a new category while removing others.

Q77: Let's say that you started an online shopping business and to grow your business, you want to forecast the sales for the upcoming months. How would you do this? Explain.

Ans77.

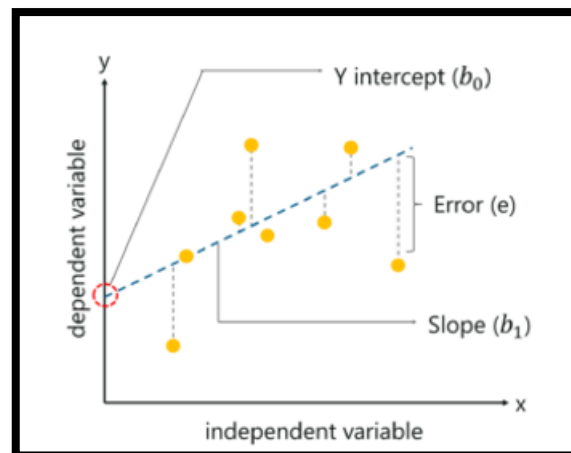
This can be done by studying the past data and building a model that shows how the sales have varied over a period of time. Sales Forecasting is one of the most common applications of AI. Linear Regression is one of the best Machine Learning algorithms used for forecasting sales. When both sales and time have a linear relationship, it is best to use a simple linear regression model. Linear

Regression is a method to predict the dependent variable (Y) based on the values of independent variables (X). It can be used for the cases where we want to predict some continuous quantity.

- **Dependent variable (Y):** The response variable whose value needs to be predicted.
- **Independent variable (X):** The predictor variable used to predict the response variable. In this example, the dependent variable 'Y' represents the sales and the independent variable 'X' represents the time period. Since the sales vary over a period of time, sales is the dependent variable.



The following equation is used to represent a linear regression model: $Y = b_0 + b_1 x + e$



Here,

Y = Dependent variable

b_0 = Y-Intercept

b_1 = Slope of the line

x = Independent variable

e = Error

Therefore, by using the Linear Regression model, wherein Y-axis represents the sales and X-axis denotes the time period, we can easily predict the sales for the upcoming months.

Q78: What is market basket analysis and how can Machine learning be used to perform this?

Ans78:

Market basket analysis explains the combinations of products that frequently co-occur in transactions. For example, if a person buys bread, there is a 40% chance that he might also buy butter. By understanding such correlations between items, companies can grow their businesses by giving relevant offers and discount codes on such items. Market Basket Analysis is a well-known practice that is followed by almost every huge retailer in the market. The logic behind this is Machine Learning algorithms such as Association Rule Mining and Apriori algorithm:

- Association rule mining is a technique that shows how items are associated with each other.
- Apriori algorithm uses frequent itemsets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset.



For example, the above rule suggests that, if a person buys item A then he will also buy item B. In this manner the retailer can give a discount offer which states that on purchasing Item A and B, there will be a 30% off on item C. Such rules are generated using Machine Learning. These are then applied to items in order to increase sales and grow a business.

Q79: We have two options for serving ads within Newsfeed:

1 – out of every 25 stories, one will be an ad

2 – every story has a 4% chance of being an ad

For each option, what is the expected number of ads shown in 100 news stories? If we go with option 2, what is the chance a user will be shown only a single ad in 100 stories? What about no ads at all?

- The expected number of ads shown in 100 new stories for option 1 is equal to 4 ($100/25 = 4$).
- Similarly, for option 2, the expected number of ads shown in 100 new stories is also equal to 4 ($4/100 = 1/25$ which suggests that one out of every 25 stories will be an ad, therefore in 100 new stories there will be 4 ads)
- Therefore for each option, the total number of ads shown in 100 new stories is 4.

- The second part of the question can be solved by using Binomial distribution. Binomial distribution takes three parameters:
 1. The probability of success and failure, which in our case is 4%.
 2. The total number of cases, which is 100 in our case.
 3. The probability of the outcome, which is a chance that a user will be shown only a single ad in 100 stories
- $p(\text{single ad}) = (0.96)^{99} * (0.04)^1$ (Note: here 0.96 denotes the chance of not seeing an ad in 100 stories, 99 denotes the possibility of seeing only 1 ad, 0.04 is the probability of seeing an ad once in 100 stories)
- In total, there are 100 positions for the ad. Therefore, $100 * p(\text{single ad}) = 7.03\%$

Q80: How would you predict who will renew their subscription next month? What data would you need to solve this? What analysis would you do? Would you build predictive models? If so, which algorithms?

Ans80:

- Let's assume that we're trying to predict the renewal rate for Netflix subscription. So our problem statement is to predict which users will renew their subscription plan for the next month.
- Next, we must understand the data that is needed to solve this problem. In this case, we need to check the number of hours the channel is active for each household, the number of adults in the household, number of kids, which channels are streamed the most, how much time is spent on each channel, how much has the watch rate varied from last month, etc. Such data is needed to predict whether or not a person will continue the subscription for the upcoming month.
- After collecting this data, it is important that you find patterns and correlations. For example, we know that if a household has kids, then they are more likely to subscribe. Similarly, by studying the watch rate of the previous month, you can predict whether a person is still interested in a subscription. Such trends must be studied.
- The next step is analysis. For this kind of problem statement, you must use a classification algorithm that classifies customers into 2 groups:
 - o Customers who are likely to subscribe next month
 - o Customers who are not likely to subscribe next month
- Would you build predictive models? Yes, in order to achieve this you must build a predictive model that classifies the customers into 2 classes like mentioned above.
- Which algorithms to choose? You can choose classification algorithms such as Logistic Regression, Random Forest, Support Vector Machine, etc.
- Once you've opted the right algorithm, you must perform a model evaluation to calculate the efficiency of the algorithm. This is followed by deployment.

Q81: After preprocessing and model building, you find out that the accuracy of the model isn't so good. Doing further dimensionality reduction is decreasing the accuracy of the model. What will you do to increase the accuracy?

Ans81:

Since we can't reduce the features anymore, we should try to add more valuable information to the model. One way to solve the problem is by increasing the model's degree of the polynomial. For example, if we have 1-degree polynomial we can transform that data into a 2-degree polynomial, 4-degree polynomial, or any higher degree polynomial that will give the best accuracy of the model. By following the central limit theorem, we can find the threshold at which our model has the best accuracy. With the increase in the degree of the polynomial, the model will be prone to overfit, so it is recommended to use cross-validation before training the data. By using pipelines, we can do the task smoothly. We also have ensemble methods that can help us to achieve the goal like a random-forest that trains multiple weak learners to provide a combined result.

Q82: Suppose you have been given a dataset in which there are some missing values in a column that has a high correlation with the target. The actual values in the column are [1,2,3,4,5,6,7,8,9,10]. But due to some error, the dataframe has values [1,nan,3,4,5,6,nan,8,9,10]. If you fill the null values with the mean, it will become 5.75, which is too far from 2 & 7. Do you have any other methods to tackle this problem?

Ans82:

We can use the divide and conquer method, which can be performed after calculating the mean of a whole series. Then we will separately calculate the mean of the two halves. We will continue this approach until we reach nearer to the index where we have missed value. There are other methods that can be used to handle missing values are:

- Mean imputation
- Substitution
- Hot deck method
- Cold deck method
- Regression imputation
- Stochastic imputation
- Interpolation
- Extrapolation
- Single imputation
- Multiple imputations

Q83: Suppose you've got a dataset with continuous target variables and having more columns than the number of rows. Why regression isn't a good option to work with? Which technique will perform best? Why?

Ans83:

First of all, we have a greater number of features(columns) than the samples(rows). This is a case of 'curse of dimensionality'. In such high dimensional data sets, we can't use classical regression techniques, since their assumptions tend to fail. When $m > n$, we can no longer calculate a unique least-square coefficient estimate, the variances become infinite, so OLS cannot be used at all. To tackle this kind of situation, we can use penalized regression methods such as lasso, LARS, ridge which can reduce variance by shrink the coefficients. Mostly, ridge regression works best in situations where the least square estimates have higher variance.

Q84: In the interview, you have been provided time-series data and asked to build a high accuracy model. You started with the decision tree algorithm since you know it works fairly well on all kinds of data. Later, you tried a time series regression model and got higher accuracy than the decision tree model. Can this happen? Why?

Ans84:

- Time series data is based on linearity while a decision tree algorithm is known to work best to detect nonlinear interactions.
- The decision tree fails to provide robust predictions. Why? The reason is that it couldn't map the linear relationship as good as a regression model did. We also know that a linear regression model can provide a robust prediction only if the data set satisfies its linearity assumptions.

Q85: Suppose you are working with 'edureka' and you have been asked to build a machine learning model that can predict the number of views for a given blog by analyzing certain features like author's name, number of blogs posted by the author, number of questions on the blog, etc. Which machine learning algorithm will you choose? How will you evaluate the model?

Ans85:

Since our target variable (or prediction variable) is the number of views that can have the value of any positive integer that means it is a continuous variable. It is recommended that regression models are the best option, whenever we have to predict a continuous value. We can start with a linear regression model and check its accuracy. For model evaluation, we can go with either R^2 or RMSE to check the accuracy of the model. If the model is overfitted we can try to regularize it. If the model is underfitted we can use the polynomial regression. Then we can again check the accuracy of the model and try to find if it is improved or not.

CHAPTER 3

**INTERVIEW
QUESTIONS**

ON

DEEP

LEARNING

(TOP 50 QUESTIONS)

Q1) What is the difference between Deep Learning and Machine Learning?

Ans1:

Conceptually, Deep Learning is quite similar to Supervised Machine Learning, in which Data Scientists use labeled data to train a model using an algorithm and then use this model to predict labels of new data. Differences between Deep Learning and ML are

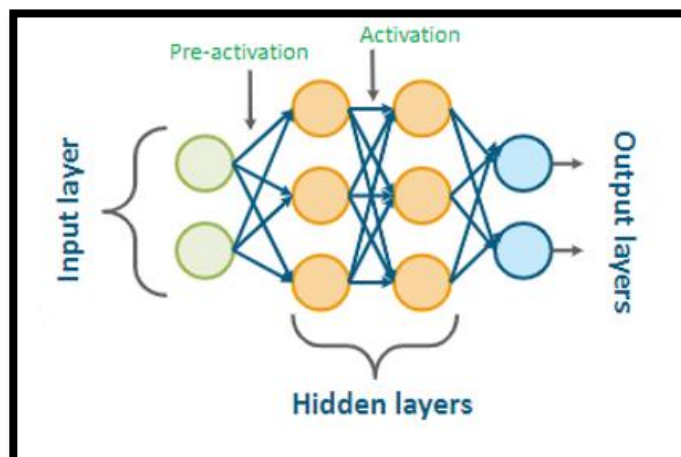
Factors	Deep Learning	Machine Learning
Hardware Requirement	Requires GPU to train properly	Works on CPU as well
Feature Engineering	Facilitates automatic feature extraction	It does not facilitate automatic feature extraction
Accuracy	Provides high accuracy	Gives lesser accuracy
Data Requirement	Requires a large amount of data to understand the patterns completely	Does not require a very large dataset to understand patterns in the data
Training Time	Large number of hyperparameters and complex mathematical vector operations make training slow	ML algorithms require relatively less training time.

Q2) What are Artificial Neural Networks (ANN)?

Ans2:

ANN's were developed in an attempt to help computers simulate the way a human brain works, using a network of neurons to process information. It helps computers learn things and make decisions in a human-like manner. An ANN consists of a few hundred to millions of neurons (also called nodes), which are divided into layers that are interconnected to form a complex network. It consists of 3 different layers:

- **Input Layer:** The layer which receives the inputs from the training datasets.
- **Hidden Layers:** It follows the input layer. There can be one or more hidden layers. It facilitates forward and backward passes and also helps in minimizing the error with each pass.
- **Output Layer:** It outputs a probability, which is used to assign a class to the set of input.



Q3: What are the typical applications where neural networks are being used in the real world?

Ans3:

Some applications in real-world where neural networks are being used are:

- **Security:** Detection of bombs in suitcases
- **Financial Risk Analysis:** Predicting stock prices (helping investors make informed decisions)
- **Weather Prediction:** Forecasting weather patterns
- **Cybersecurity:** Identifying fraudulent credit card transactions
- **Healthcare:** Predicting the risk of heart attacks from ECG output waves

Q4: Explain one real world application where ANN can be used?

Ans4:

Traveling salesman problem: A salesman has to cover all the cities in a given area. He wants to figure out the shortest possible path to travel to all the cities. He uses neural networks to solve this problem. A neural network algorithm like genetic algorithm starts with a random orientation of the network. The algorithm chooses a city in a random manner and finds the nearest city. This process continues several times and after every iteration, the shape of the network changes and the network converges to a ring around all the cities. The algorithm aims to minimize the length of this ring with each iteration.

Q5: What is Multilayer Perceptron (MLP)? How does it overcome the shortcomings of Single Layer perceptron? MLP's are feedforward ANN's that consist of multiple hidden layers and generate a set of outputs from a set of inputs. MLP uses backpropagation as a supervised learning technique. It has 3 layers:

Ans5:

- **Input layer:** The input nodes provide information from the outside world to the network and are together referred to as the "Input Layer". No computation is performed in any of the input nodes, they just pass on the information to the hidden nodes.
- **Hidden layers:** The hidden nodes perform computations and transfer information from the input nodes to the output nodes. A collection of hidden nodes forms a "Hidden Layer".
- **Output layer:** The output nodes are collectively referred to as the "Output Layer" and are responsible for computations and transferring information from the network to the outside world.

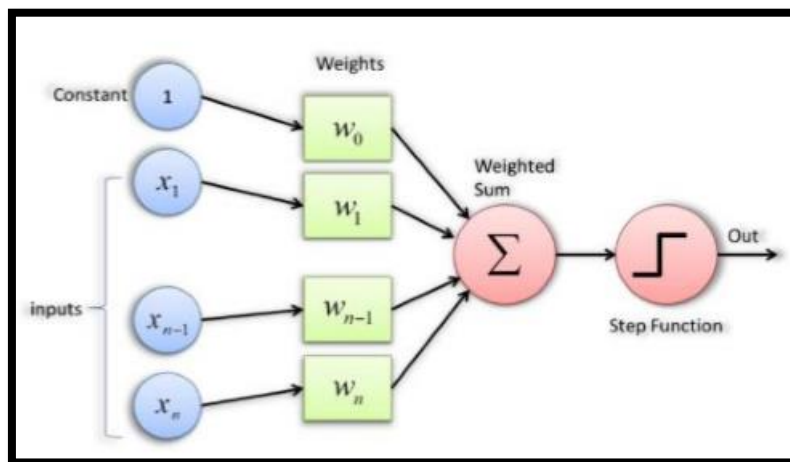
A single-layer perceptron cannot deal with non-linearly separable classes. MLP's overcome this limitation by:

- Using a non-linear activation like logistic sigmoid, ReLU or tanh in its hidden layers and output layer, which help it to understand the non-linearities in the data.
- Using multiple hidden layers and nodes, which help it to understand complex patterns in the data better.

Q6: Which is the simplest type of Artificial neural network? What is its limitation?

Ans6:

Single-layer perceptron is the simplest ANN. It is a single layer, binary linear classifier. It classifies objects and groups by finding a linear separation boundary between different classes. It has only one node. Below is a basic representation of a perceptron:

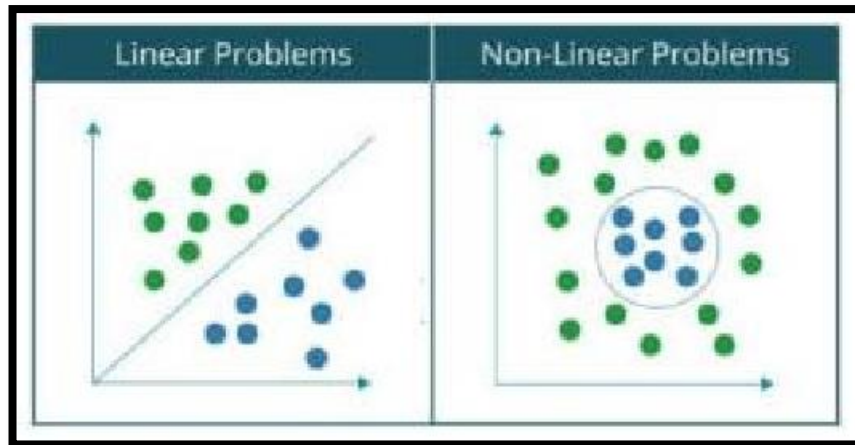


It consists of:

- A set of **input values** with their associated **weights** and **biases**
- A **pre-activation function**, calculated by a sum of products of inputs and their associated weights. A bias is also added to this to provide every node with a trainable constant value in addition to normal inputs
- The **activation function** is a step function. If the weighted sum exceeds the predefined threshold, then it assigns one class, else the other.

It updates the values of weights and bias matrices based on the loss (difference between actual and predicted values) and the model learns progressively with each iteration.

Its **limitation** is that it is only able to classify a linearly separable set of inputs. It cannot deal with classification problems where the classes cannot be separated by a linear separation boundary.

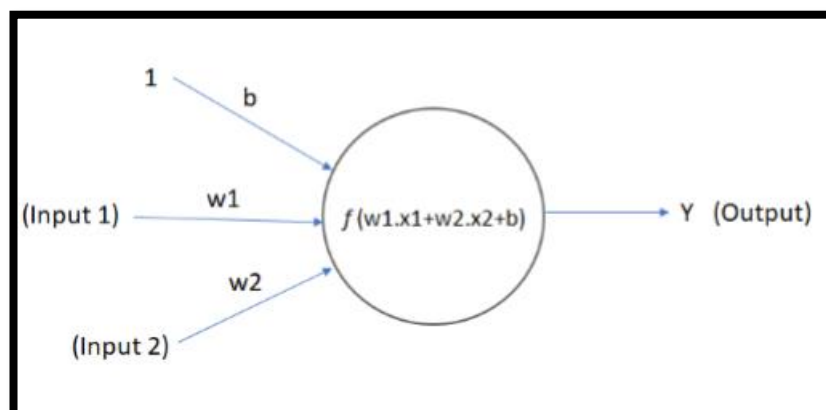


Q7: What is an Activation Function? Explain it with respect to a single node?

Ans7:

The figure shows a single neuron (or node), which receives inputs from nodes of the previous layer, and a bias to add a trainable parameter. Each input to the neuron is associated with a weight, which is assigned based on its relative importance with respect to other inputs. The node computes the weighted sum of the inputs and the bias, and applies a function 'f' to the computed sum. This function 'f' is called activation function (or non-linearity).

The purpose of the activation function is to introduce non-linearity to the output of the neuron (This is important as the most real-world data is non-linear, and we want the neuron to learn these nonlinearities).



Q8: Why is SoftMax Activation function primarily used in the output layer of NN?

Ans8:

When we are using a neural network to solve a “multiclass” classification problem with ‘K’ number of categories/ classes, the SoftMax function is used at the output layer to calculate a probability value

for each class. It outputs values in the range (0,1) for each output class, such that the sum of outputs of all classes is equal to 1.

Advantages of using SoftMax function are:

1) The properties of SoftMax (all output values in the range (0, 1) and sum up to 1.0) make it suitable for a probabilistic interpretation which is very useful in ML.

2) SoftMax normalization is a way of reducing the influence of extreme values or outliers in the data without removing data points from the set.

Q9: What are the different types of activation functions used in neural networks?

Ans9:

Some of the commonly used activation functions are:

- Sigmoid Function: It takes a single real-valued input and squashes it between 0 and 1.

$$\sigma(x) = 1/(1+e^{-x})$$


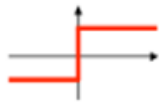



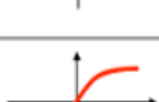
- ReLU (Rectified linear unit): It converts negative values to 0 and retains positive ones.

$$f(x) = \max(0, x)$$

- tanh: It takes a real value and squashes it between [-1,1]

$$\tanh(x) = 2\sigma(2x) - 1$$

Below is a graphical representation of some well-known activation functions:

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer NN	

Q10: How does using the tanh activation function help the neural network to converge faster than logistic sigmoid?

Ans10:

The global minima of the loss function can be achieved much faster i.e, in a fewer number of epochs when using tanh activation function. It outputs values between -1 and 1, which helps the weight vectors to change directions much faster as opposed to logistic sigmoid which outputs values between 0 and 1.

Q11: What are the various techniques of Data Normalization and how does it help in boosting the training of a neural network?

Ans11:

Data Normalization in Deep Learning is often applied as a part of data pre-processing. The goal of normalization is to change the values of numeric columns in the dataset to a common scale. Normalizing the data generally speeds up learning and leads to faster convergence. 3 main methods used for normalizing data are:

- 1) **Rescaling (min-max scaling):** It transforms data to a scale of [0,1].
$$x_{norm} = (x - x_{min}) / (x_{max} - x_{min})$$
- 2) **Standardization:** The data is normalized to a Z-score(standard score).
$$x_{norm} = (x - \mu) / \sigma$$
- 3) **Scaling to unit length:**
$$x_{norm} = x / \|x\|$$
, where $\|x\|$ is the Euclidean length of the feature vector.

Normalization helps in boosting the training of a neural network by:

- Ensuring a feature has both positive and negative values which makes it easier for the weight vectors to change directions. This helps in making the learning flexible and faster by reducing the number of epochs required to reach the minima of the loss function
- Normalization ensures that the magnitude of the values a feature assumes fall within a similar range. The network regards all input features to a similar extent, irrespective of the magnitude of the values they hold.

Q12: What is the cost/ loss function? Explain how it is used to improve the performance of the neural network.

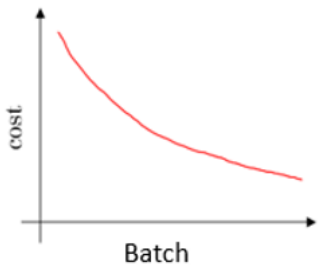
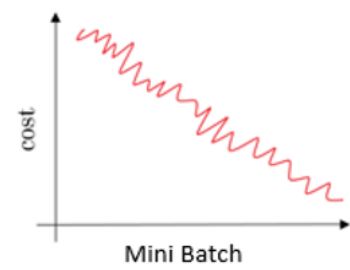
Ans12:

Loss/ cost function is a performance indicator of the neural network. It is a measure of the accuracy of the neural network with respect to a given training sample and an expected output. While training neural networks, the primary goal is to minimize the cost function by making appropriate changes to the trainable parameters of the model (weights and bias values). The steps followed to achieve this goal are:

- For each iteration of the neural network, we calculate the values of the loss function partial derivatives with respect to the trainable hyperparameters of our model
- The values of the weights are then adjusted so that the model moves in a direction opposite to the direction of increasing slope of the cost function (**using Gradient Descent**)
- With each successive epoch, we head closer to the minima of the cost function
- Training stops when the model loss reaches the minima of the loss function

Q13: What is the difference between Batch Gradient Descent and Stochastic Gradient Descent?

Ans13:

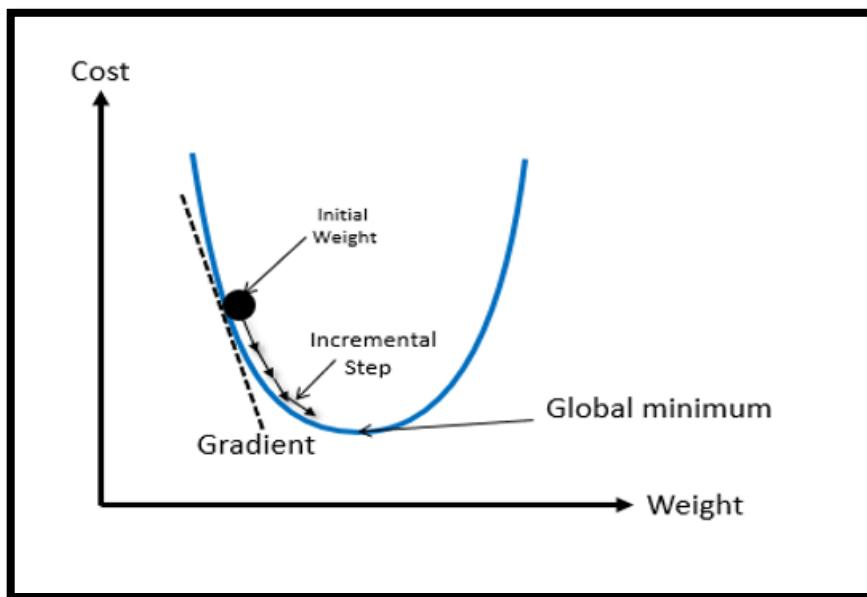
Batch Gradient Descent	Stochastic Gradient Descent
It considers all the training examples at once to take a single step.	We consider just one training example at a time to calculate the gradient and update the parameters.
Not suitable for very large datasets as the process becomes very slow as it considers the entire data for each iteration.	Suitable for very large datasets as it considers only one training example in each iteration.
Since we use all the data for computing the gradients, we move somewhat directly towards the optimum value.	Since we use just one training example at a time, the cost will not necessarily decrease with each iteration but keeps fluctuating up and down with each successive iteration.
Example of how the cost function decreases with each epoch using Batch Gradient descent: 	Example of how the cost function decreases with each epoch using Stochastic Gradient descent: 

Q14: Which optimization algorithm is used in neural networks to automatically update the values of the model parameters in order to minimize the loss function?

Ans14:

Gradient Descent is an optimization algorithm, used to find the optimal set of parameters for a function, for which the function attains its global minimum. For neural networks, this function is the cost function. To achieve this objective, the algorithm follows the following steps iteratively:

- Initialize random weight and bias.
- Pass an input through the network and get values from the output layer.
- Calculate the error between the actual value and the predicted value.
- Go to each neuron which contributes to the error and then change its respective values to reduce the error.
- Iterate until you find the best weights of the network.



Q15: What are the different ways in which Gradient Descent is used to attain a global minimum of cost function?

Ans15:

Let us understand this with an example. Suppose there is a man who wants to trek down a valley. At each step, he takes a step forward so as to get closer to the bottom(global minima in this case). He takes the next step based on his current position and stops when he reaches the bottom, which is his aim. There are different ways in which the man(weights) can reach the bottom. The commonly used ones are:

- **Batch Gradient Descent:** Calculate the gradients for the whole dataset and perform just one update at each iteration.
- **Stochastic Gradient Descent:** Uses only a single training example to calculate the gradient and update parameters.
- **Mini Batch Gradient Descent:** Mini-batch gradient is a variation of stochastic gradient descent where instead of single training example, mini-batch of samples is used. It's one of the most popular optimization algorithms.

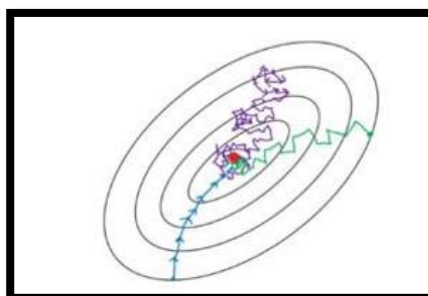
Q16: How does mini-batch Gradient Descent perform better than Batch Gradient Descent and SGD?

Ans16:

In mini-batch gradient descent, we utilize the advantages of both Batch Gradient Descent and SGD. Batch Gradient Descent can be used to obtain smoother curves and converge directly to the minima. SGD can be used for huge datasets as it converges faster, but we cannot use vectorized operations as it uses just a single example at a time. This makes the computations much slower. To tackle this problem, a mixture of Batch Gradient Descent and SGD is used.

Working: We use samples of the training data at a time (batches). For example, if the dataset consists of 10000 examples, and we select a batch size of 1000 examples at a time(called mini-batch). After creating mini-batches of fixed size, we perform the following steps in one epoch:

- Randomly pick a mini-batch from the training data
- Feed it to the NN
- Calculate the mean gradient for that batch
- Use the calculated mean gradient to update the weights
- Repeat the above steps for different samples/batches The figure below makes it more clear. Notice how Batch gradient (blue) descent moves directly towards the center without many fluctuations, SGD (purple) moves towards the center with a lot of fluctuations and mini-batch gradient descent (green) moves towards the center with lesser fluctuations than SGD.



Q17: What is Backpropagation algorithm? What are its drawbacks?**Ans17:**

The process by which an MLP learns is called Backpropagation. It repeatedly adjusts the weights of the connections in the neural network so as to minimize a measure of the difference between the actual output vector and the desired output vector with each successive iteration. BackProp is like “learning from mistakes”. It is a supervised learning algorithm and follows these steps:

- Initially, all weights and biases are randomly assigned
- The input is fed to the net and ANN is activated
- The output is compared with the desired output and the error is calculated
- This error is propagated backward to the previous layers and the weights and biases are adjusted following gradient descent method
- This process is repeated until the error is below a predefined threshold

The drawbacks of using backpropagation are:

- **Local minima problem:** The algorithm always adjusts the weights so as to decrease the error. In this process, it might get stuck at a local minimum where the gradient will be zero, and it will stop the training process.
- **Network paralysis:** Occurs when the weights are adjusted to very large values during training, which can force most of the units to operate at extreme values, in a region where the derivative of the activation function is very small.

Q18: What is the learning rate?**Ans18:**

Learning rate is a hyperparameter that defines how quickly the model moves towards the optimal set of weights and biases to achieve minimal cost.

In Gradient Descent, the updated value of weights is given by:

$$\Delta W = -\alpha \cdot (\partial L / \partial W)$$

$$W_{\text{Updated}} = W_{\text{old}} + \Delta W$$

Here,

α : learning rate

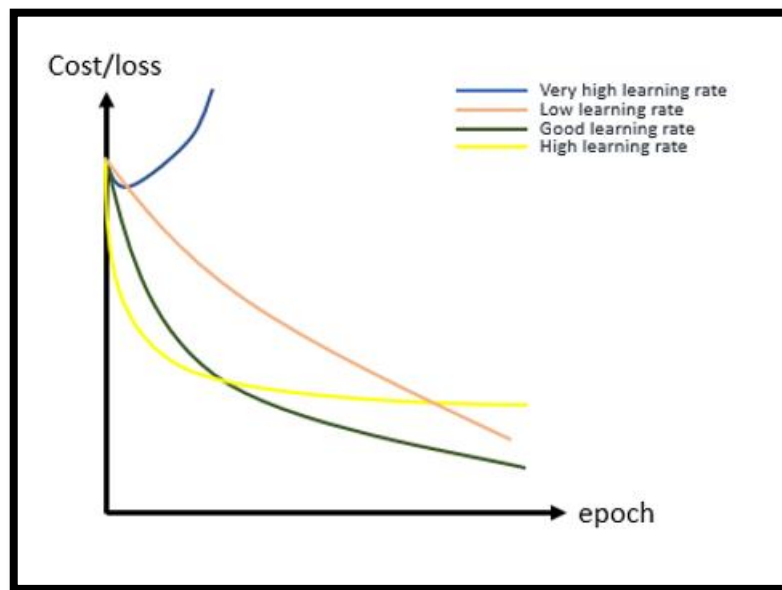
$(\partial L / \partial W)$: the slope of the loss function

Q19: What is an optimal value for learning rate? What are the effects of setting the learning rate to be too high or too low?

Ans19:

For Gradient Descent to perform well, it is important to set the learning rate to an appropriate value. If the learning rate is very large, you will skip the optimal solution and if it is too small you will need too many iterations to converge to the best values. An optimum learning rate is one that's low enough so that the network converges to something useful but high enough so that it can be trained within a reasonable amount of time.

The graph below shows the effect of various learning rates on the cost function convergence:



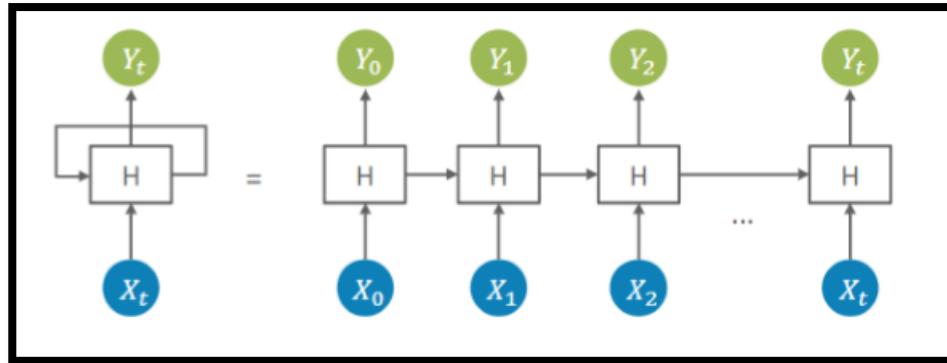
Very high or very low learning rates can lead to waste of time and resources. A lower learning rate implies more training time, which results in increasing GPU costs. A higher learning rate would result in a model that is not able to predict anything accurately.

Q20: Traditional deep learning neural networks are not able to deal with sequential data where the current value has some dependency on the values that come before it i.e, when there is a sense of ordering in the data. Which algorithm is used to deal with ordered data?

Ans20:

Traditional neural networks treat each input example independently i.e, the inputs are not related to each other and there is no sense of ordering in the data. They lose their power in applications like time series forecasting, connected handwriting recognition and speech recognition. RNN is the go-to algorithm in such cases.

An RNN (Recurrent Neural Network) is a generalization of a Feedforward NN with an additional internal "memory". RNN's can use their internal "state" memory to process sequences of inputs. In other neural networks, all the inputs are independent of each other while in RNN, the inputs are related to each other. The following is a diagrammatic representation of how an RNN works:



The formula for the current state can be represented as: $h_t = f(h_{t-1}, x_t)$

The steps followed in RNN are:

- First, it takes the $X(0)$ from the sequence of input and generates $h(0)$ output.
- $h(0)$ combined with $X(1)$ is the input for the next step. So, $h(0)$ and $X(1)$ are the inputs for the next step.
- Similarly, $h(1)$ combined with $X(2)$ is the input for the next step and so on. This way, it keeps remembering the context while training.

Q21: How does the problem of vanishing and exploding gradients affect the performance of an RNN?

Ans21:

While training an RNN, the slope can at times become either too small or too large. This makes the training process difficult. When the slope becomes too small, the problem is known as a “Vanishing Gradient.” and when the slope grows exponentially instead of decaying, it’s referred to as an “Exploding Gradient.”

Gradient problems lead to unacceptably long training time, poor performance, and low accuracy.

Q22: RNN is unable to learn long term dependencies in the data. What is used to combat this problem?

Ans22:

LSTM (Long Term Short Memory) has a default behavior of remembering information for long periods. It is a special kind of RNN capable of learning long-term dependencies. It resolves the problem of vanishing gradients associated with RNN’s. It is well suited to predict time series problems with unknown durations. It trains the model using backpropagation and uses 3 gates, an **input gate**, a **forget gate** and an **output gate**.

Q23: What is the difference between Feedforward neural network and backpropagation?

Ans23:

A Feed-Forward Neural Network is a type of Neural Network architecture where the connections are “fed forward”, i.e. do not form cycles. The term “Feed-Forward” is also used when you input something at the input layer and it travels from input to hidden and from hidden to the output layer.

Backpropagation is a training algorithm consisting of 2 steps:

- Feed-Forward the values.
- Calculate the error and propagate it back to the earlier layers.

To be precise, forward-propagation is part of the backpropagation algorithm but comes before back-propagating.

Q24: What is the difference between Feedforward and Recurrent Neural Networks?

Ans24:

Feedforward Neural Network	Recurrent Neural Network
Signals travel only in one direction, from input towards the output.	Signals travel in both directions making it a looped network.
Considers only current input to generate the output of a layer.	Considers current input as well as previously used inputs for generating the output of a layer.
Cannot memorize previous inputs as it does not have any internal memory (eg: CNN)	Its internal memory enables it to memorize past data.

Q25: What are the techniques by which you can prevent a neural network from overfitting?

Ans25:

Some of the popular methods of avoiding overfitting while training neural networks are:

- **L1 and L2 regularizations:** Regularization involves adding an extra element to the loss function, which punishes our model for being too complex. In simple words, for using too high values in the weight matrix. By this method, we attempt to limit its flexibility and also encourage it to build solutions based on multiple features. Two popular versions of this method are Least Absolute Deviations (LAD or L1) and Least Square Errors (LS or L2).

L1 reduces the weights associated with less important features to zero, thereby completely removing their effect. It is effectively an in-built mechanism for automatic feature selection. In **most**

cases, L1 is preferred over L2, as it does not perform well on datasets with a large number of outliers.

- **Dropout:** In this method, every unit of our neural network (except the output layer) is given a probability ‘p’ of being ignored in the calculations. The hyperparameter ‘p’ is called the dropout rate, and is usually set to 0.2. In each iteration, we randomly select the neurons that we drop based on the value of ‘p’. As a result, each time we work with a smaller neural network and we are able to prevent overfitting by using a fewer number of features at each iteration.
- **Early Stopping:** Many times it is observed that till a certain number of epochs, the error on the training set and the cross-validation set decreases, but after that the error on the cross-validation set starts to increase, while that of the training set still decreases. This is where overfitting starts, and it is advisable to stop the training process after a certain number of epochs. Thus, early stopping means to stop training the model once it starts overfitting.

Q26: Many programmers prefer Deeper Networks over shallow ones. But is it always advisable to use deeper networks over shallow ones?

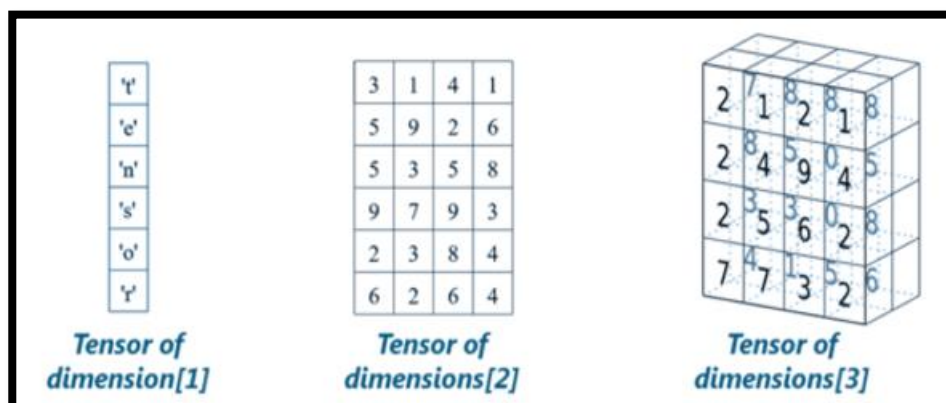
Ans26:

A shallow network is one that has a fewer number of hidden layers. Deeper the network (more the number of nodes and hidden layers), the better the model is able to learn. But, using a very large number of nodes can result in the model following the data too closely, resulting in overfitting. The number of hidden layers to be used, and consequently, the number of nodes depends on the size and complexity of the dataset. For example, if the classes are linearly separable or we can say if the classes do not intersect each other much, a shallow network performs better as it provides a relatively generalized output. So the optimal “depth” of the neural network largely depends on the data and there can be no general rule that deeper networks can outperform shallow ones.

Q27: How is data represented in Deep Learning? List the various types with examples?

Ans27:

A tensor is a standard way of representing data in deep learning. It is an N-dimensional array of data. For example, a tensor holding just a single scalar value is a 0-dimension tensor. A vector is a 1-dimensional tensor whereas a matrix is a 2-dimensional tensor.



The different types of tensors commonly used in Deep Learning are:

Constant: A constant is a tensor whose value cannot be changed and remains the same. Example of constant tensors are:

```
a=tf.constant(5), scalar 0-dimension tensor
```

```
v=tf.constant([4,6,8]) , 2-dimension tensor
```

```
mat=tf.constant([[1,2,3],[4,5,6],[5,7,9]]) , 3.dimensional tensor
```

Variable: A tensor whose value can be changed as and when the program runs. It can be used to add trainable parameters to the model.

```
w=tf.Variable([-0.3], tf.float32)
```

```
b=tf.Variable([0.3], tf.float32)
```

Variables need to be initialized first in order to use them to make computations in a session. It is initialized using the **global_variables_initializer()** function.

Placeholder: It is used to provide external inputs. It is a promise to provide a value later when the session is run.

```
A=tf.placeholder(tf.float32)
```

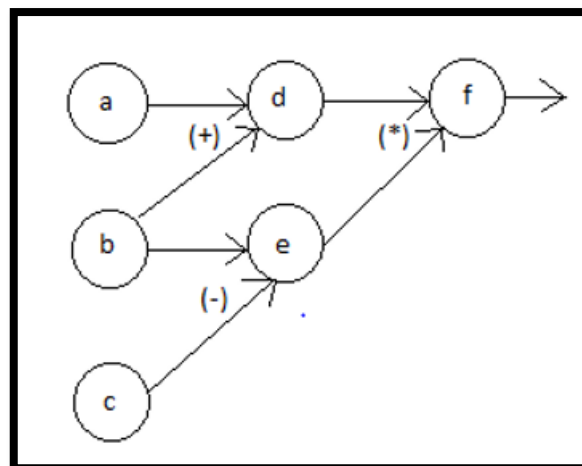
Q28: What is a computational graph? How is it executed?

Ans28:

Tensorflow core programs consist of 2 discrete sections:

- Building a computational graph
- Running a computational graph

Building a Computational Graph: A Computational Graph can be thought of as a network of nodes, with each node known as an operation. Below is an example of a computational graph.



It can be implemented as:

```
a=tf.constant(5.0, tf.float32)
```

```
b=tf.constant(9.0, tf.float32)
```

```
c=tf.constant(7.0, tf.float32)
```

```
d=tf.add(a,b)
```

```
e=tf.subtract(b,c)
```

```
f=tf.multiply(d,e)
```

We have built a computational graph. Now we have to execute it in a session.

Running a Computational Graph: To evaluate a graph, we need to run it within a session. A session encapsulates the control and state of the TensorFlow runtime. It can be implemented as:

```
with tf.Session as sess:  
    print(sess.run(f))
```

This will print the output as: 28

Q29: Which is the go-to algorithm for Image recognition and classification problems? What are the advantages it offers over traditional neural networks?

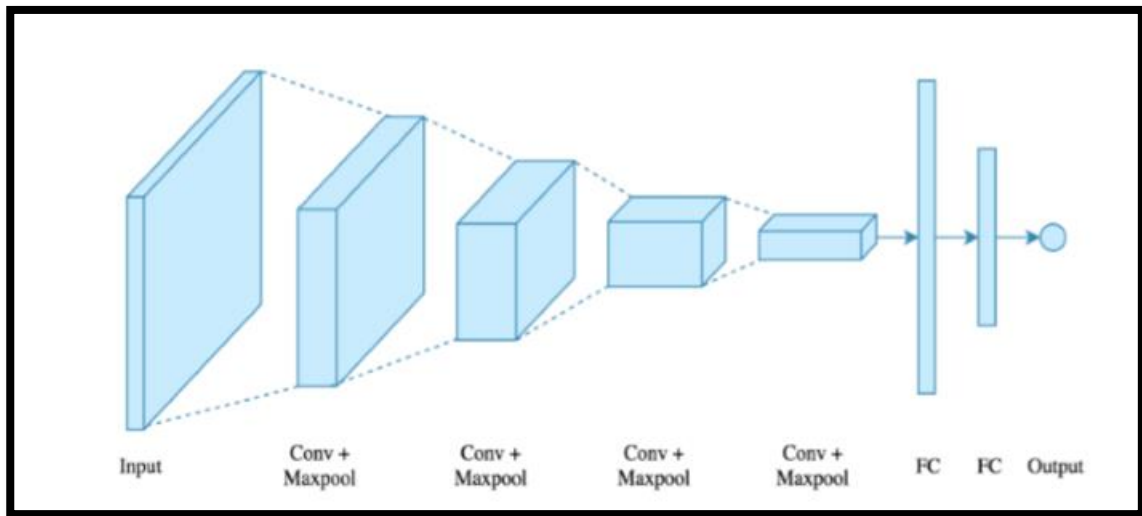
Ans 29:

Convolutional Neural Networks (CNN) is the go-to algorithm for any image-related problems. It can also be applied to recommender systems, NLP and more. The advantages that CNN offers over other Deep Learning methods are:

- CNN is more efficient in terms of memory and complexity
- Convolution and pooling layers of CNN help in reducing the number of parameters to tune.
- CNN is much faster than traditional NN as it extracts important information before feeding the image into fully connected layers, reducing the number of nodes.

Q30: What is the architecture of CNN? Explain the working of each layer in CNN?

Ans30:



We pass an input image through a series of convolution and pooling operations, followed by a number of fully connected layers. The convolution and pooling operations are used to extract the features of interest and reduce the dimensionality of the image before it is passed to the fully connected layers which perform classification.

Input: The input image (an n-D matrix where each element of the matrix contains information about a pixel). The values of the matrix represent the pixel intensities. In most cases, an image is represented as a 3D matrix with dimensions of height, width and depth, where depth corresponds to color channels (RGB).

Convolution layer: Convolution is a mathematical operation that is used to merge two sets of information. The result of applying a convolution operation is called feature map. We can also use multiple filters to extract different types of information. We then stack all these different feature maps obtained to get the result of the convolution layer.

Pooling: Convolution is followed by Pooling, to reduce the dimensionality. Pooling helps to reduce the number of parameters, which reduces the training time and helps to combat overfitting. Pooling helps in downsampling the feature map, while keeping important information. The most common types of pooling are max pooling and average pooling.

Fully Connected Layers: We pass the output of the final pooling layer to a network of fully connected layers. We flatten the output of the final pooling layer to make it a 1-D vector which is the input to the fully connected layer. These fully connected layers work in the same way as traditional ANN's. At the output, we generally use Softmax activation function, which is the most preferred for multiclass classification problems.

Q31: What are the advantages of using convolutional layers over fully connected layers?

Ans31:

The convolution layers are the powerhouse and the most important step in CNN. Convolution + Pooling layers perform feature extraction. For example, given an input image of a cat, it detects features such as 2 eyes, whiskers, small ears, short legs, short tail etc. The fully connected layers then act as a classifier on top of these features and assign a probability of the image being a cat.

Thus, convolution layers enable automatic feature detection and extraction. The convolution layers learn these meaningful features by building on top of each other. The first few layers detect edges, the next layers combine them to detect shapes, and the following layers merge this information to infer that this is the eye.

Another advantage is that it drastically reduces the dimension of the input vector to the fully connected layer, also reducing the number of nodes, thereby reducing the number of matrix multiplications resulting in a reduction in the training time.

Q32: Many times, we do not have a sufficient number of images for CNN to be applied. What can you do when you have limited training data?

Ans32:

Data Augmentation is the process of generating additional training data from the current set. If we have too few training instances, the model has poor regularization, leading to overfitting. Data Augmentation enriches or “augments” the training data by generating new images via random transformations of existing ones. This way we artificially boost the size of the training set, providing more information and thereby reducing overfitting. In this way, it can also be considered as a regularization technique. Data augmentation can boost the size of the training set by even 50 times the original. It’s a very powerful technique that is used in every single image-based deep learning model.

Q33: Given an image of size (n×n×d), filter of size (f×f×d), (nf) is the number of filters used, padding ‘p’, stride of ‘s’, what will be the dimension of the output image?

Ans33:

Input dimension: (n×n×d)

Filter size: (f×f×d)

Number of filters: nf

Padding: p

Stride length: s

The size of the output image will be: $\left[\frac{(n+2p-f)}{s} + 1\right] \times \left[\frac{(n+2p-f)}{s} + 1\right] \times nf$

Q34: We at times tend to lose information at the borders of an image after a convolution operation. How would you deal with this? When we perform convolution operation, the pixels at the borders of the image are used fewer times, as compared to central pixels. We do not focus much on the corners. Also, when we apply convolution operation, the size of the image shrinks. This results in a loss of information.

Ans34:

To combat this, we can pad the image with an additional border, i.e., we add one pixel all around the edges. There are 2 common choices of padding:

- Valid: It means no padding. The output image size will shrink.
- Same: Here we apply padding, so that the output image size is same as the input image size, i.e., $(n+2p-f+1)=n$
 $\Rightarrow p=(f-1)/2$

In this way, we do not lose too much information and the image size does not shrink either.

Q35: How does pooling help in filtering only important features and reducing training time?

Ans35:

Suppose we have an image, where the object of interest has pixels of higher intensity compared to the background. In this case, the pixels of lower intensities are not of interest, and filtering them out will largely reduce the number of nodes and consequently the number of computations and the training time.

For such cases, we can use max pooling which considers only the pixel with maximum value in the window defined by it. Suppose the size of the image is 28x28, then the size of input vector to the fully connected layers would be an array of size 784. But if we apply max pooling with window of size 2x2 and a stride of 2, the image dimension will be reduced to 14x14, and the input vector will now be an array of size 196 (which is 1/4th of the size without pooling). This boosts the training time and also reduces the number of parameters to tune.

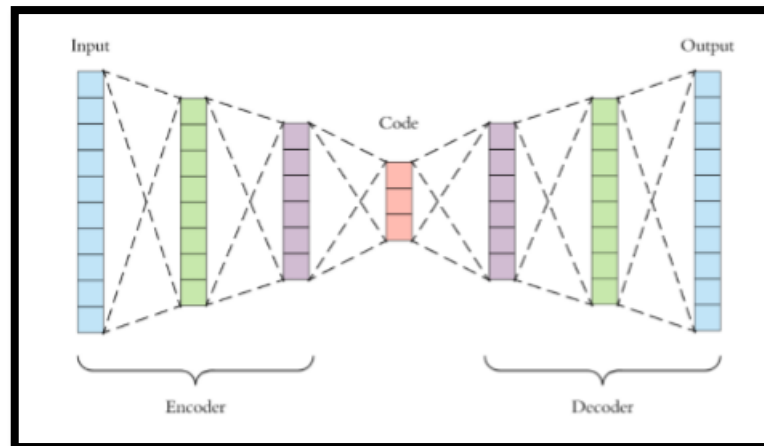
Q36: What are Autoencoders?

Ans36:

Autoencoders are feedforward neural networks where the input is the same as the output. They compress the input into a lower-dimensional code (also called latent space representation) and then reconstruct the output from this latent space representation. The code is a “compression” of the input.

The three components of autoencoder are:

- Encoder: Compresses the input and produces the code
- Code: a compact summary of the input
- Decoder: Reconstructs the input using the code



The input passes through the encoder, which is a fully connected ANN to produce the code. The decoder also has a similar ANN structure, with an architecture that is the mirror image of the encoder (this is not a requirement but is generally the case). The only requirement is that the dimensionality of the output and input should be the same.

The goal is to get an output that is identical to the input. Since the code layer has fewer number of nodes than the input, it is said to be undercomplete. In this way, it won't be able to simply copy the input to the output but will instead learn the important features.

Q37: List the various types of scenarios where autoencoders can be applied?

Ans37:

Image Coloring: Autoencoders are used for converting any grayscale image into a colored image. Depending on what is in the picture, it makes it possible to tell what the color should be.

Feature variation: It extracts only the important features of an image and generates the output by filtering out any noise or unnecessary interruption.

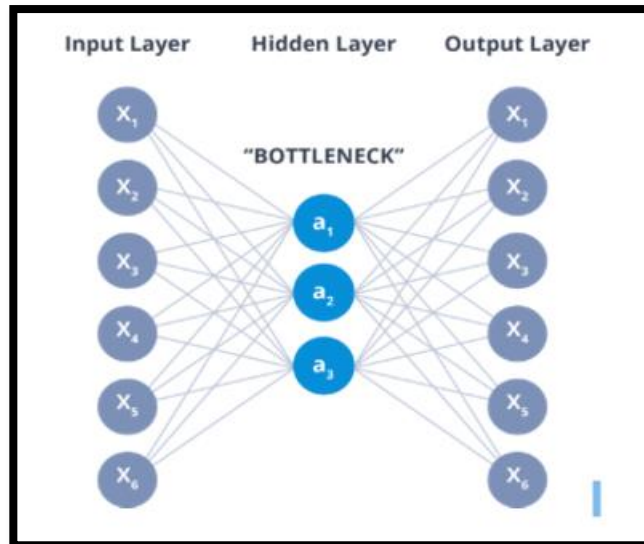
Dimensionality Reduction: It helps in providing a similar image with a reduced pixel value, wherein the reconstructed image is the same as our input but with reduced dimensions.

Denosing Image: The input seen by the autoencoder is a stochastically corrupted version of the raw input. A denoising autoencoder is trained to reconstruct the original input from the noisy version.

Q38: What is Bottleneck in Autoencoder and why is it used?

Ans38:

The layer between the encoder and decoder i.e., the code is also known as Bottleneck. This is a well-designed approach to decide which aspects of observed data are relevant information and what aspects can be discarded.



It does this by balancing two criteria:

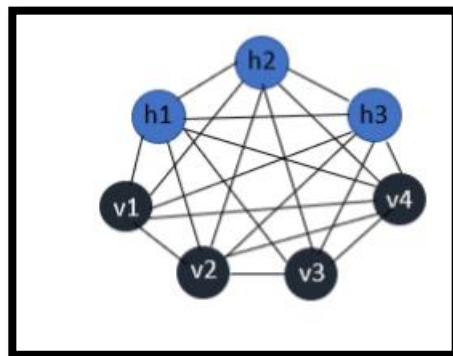
- Compactness of representation, measured as the compressibility.
- It retains some behaviorally relevant variables from the input.

Q39: What is a Boltzmann Machine?

Ans39:

These are Stochastic (or non-deterministic) Generative deep learning models. It has only 2 types of nodes: hidden and visible. They have no output nodes (which gives them this non-deterministic feature). Unlike other traditional deep learning methods, Boltzmann machines have connections between the input nodes as well. All nodes are connected to all other nodes irrespective of whether they are input or hidden nodes. This allows them to share information and self-generate subsequent data. When the input is provided, they are able to capture all the patterns and correlations among the data.

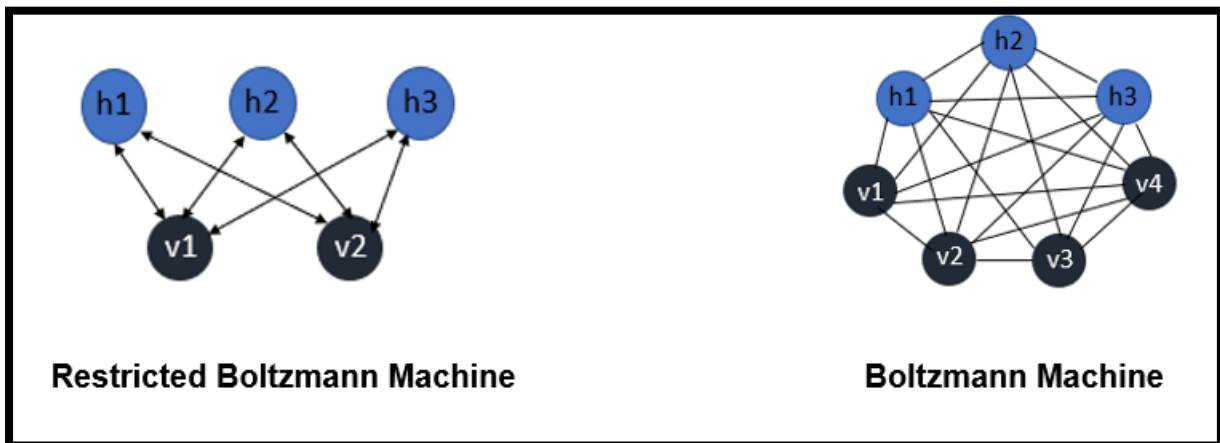
Therefore, they are called Deep Generative models and fall under the class of Unsupervised Deep Learning.



Q40: What are the differences between Boltzmann Machine and Restricted Boltzmann Machines?

Ans40:

Restricted Boltzmann Machines are a special type of Boltzmann Machine, with the restriction that every node in the visible layer is connected to every node in the hidden layer but no two nodes of the same layer are interconnected. This makes them less complicated and easier to implement as compared to Boltzmann Machines. They have the ability to learn a probability distribution over its inputs. They have generative capabilities and can be used for dimensionality reduction, classification, regression, collaborative filtering, feature learning, and topic modeling.



Q41: What is the limitation of CNN in object detection and which algorithm is used to overcome those limitations?

Ans41:

Suppose you are in a hurry and you can't find your room keys as you have misplaced them in your room. CNN can only tell whether the keys are there in the room or not, but it cannot assist in finding the keys. The limitations of CNN are:

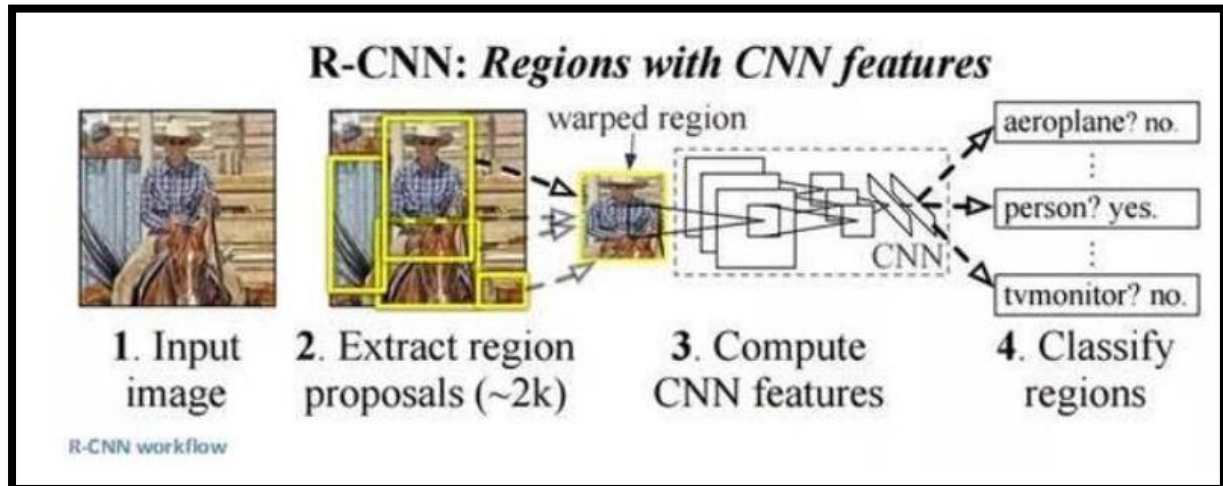
- CNN is used only to identify the class of an image, and not the spatial location of the object in the image
- It cannot be used to identify multiple objects in the image.

For multiple object detection using CNN, you will first have to divide the image into a large number of small images(regions) and look for the objects in those images. Even this fails if the object has different sizes in different images i.e, in some cases, an object might cover the entire image (dividing the image in this case divides the object, and its presence cannot be detected), and in some cases, the object covers only a small portion in the image. CNN loses its power in such cases where objects in the image can have different aspect ratios and spatial locations.

This is where **RCNN** (region-based CNN) comes to the rescue. It outputs an image with each identified object surrounded by a distinct bounding box and a certain level of precision.

Q42) What is the architecture of RCNN?

Ans42:



The steps followed by RCNN algorithm are:

- It first takes an image as input and extracts the ROI (regions of interest) using some proposal method eg: selective search
- All these regions are then reshaped as per the size of the input of the pre-trained CNN, and each region is passed through the ConvNet
- CNN then extracts the features for each region and then SVM is used to classify objects and backgrounds. For each class, we train one binary SVM separately
- In the final step, a Bounding Box regression (Bbox regression) is used to generate a bounding box for each identified image

Q43) How does RCNN identify the regions of interest in an image?

Ans43:

RCNN is used to identify a number of objects in the image along with their spatial locations. It outputs an image with each identified object surrounded by a distinct bounding box and a certain level of precision. RCNN uses selective search to extract these boxes from an image (these boxes are out regions of interest i.e., regions in the image that can contain an object).

There are basically four distinct regions that form an object: varying colors, texture, scale and enclosure. Selective search identifies those patterns in an image and based on that it proposes various regions. The steps followed are:

- It first takes an image as input and generates sub-segmentations so that we have multiple regions from this image
- It then combines similar regions to form larger regions. It combines regions based on color, texture, size and shape compatibility)
- Finally, these regions then produce the ROI (regions of interest i.e., the final object locations

Q44: What is the problem of using RCNN with very large datasets? What are the modifications of RCNN which make the computations faster?

Ans44:

Training an RCNN is computationally slow and expensive because of the following reasons:

- Extracting nearly 2000 regions for each image using selective search itself is a time-consuming process
- Training a CNN for each of these 2000 regions. Suppose we have N images, then the number of features for CNN will become $2000 \times N$, which will be very large
- RCNN required 3 different models, a pre-trained CNN, a Linear SVM classifier and a regression model to tighten the bounding boxes for each identified region.

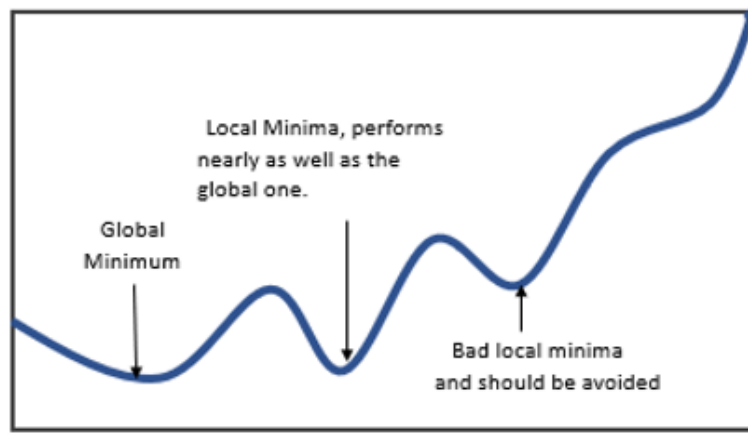
All these steps combine to make RCNN very slow and it takes around 30-40 seconds to make predictions for each new image. This makes the model cumbersome and practically impossible to build when presented with a big dataset.

The modifications of RCNN are:

- **Fast RCNN:** It uses a single model which extracts features from the regions, divides them into different regions, and returns the boundary boxes for the identified classes simultaneously
- **Faster RCNN:** It used RPN (Region Proposal Network) instead of selective search for identifying the regions of interest in the image)

Q45: Many times, while training a neural network, optimization becomes very difficult if the network gets stuck at a local minima. How would you try to overcome this situation? (asked in Amazon)

Ans45:



If the network is stuck at a bad local minimum (like saddle points, which is surrounded by flat regions), then it needs to be optimized. In such cases, we can tune our parameters to make our model perform better:

- **Increasing the learning rate:** If the learning rate set is too small, then it has a higher probability of being stuck at a local minima
- **Increasing the number of hidden layers:** Increasing the number of nodes will help to approximate the function better
- **Trying different combinations of activation functions**
- **Trying different optimization algorithms** like ADAM's optimizer and RMSProp instead of the traditional gradient descent

Q46: What will happen if the activation function will be removed from a neural network? (Google interview question)

Ans46:

Without the activation function (which introduces non-linearity), the weights and bias would simply do a linear transformation. A linear equation is simple to solve but it cannot be used with complex problems. A neural network without an activation function is essentially just a linear regression. The activation function does the non-linear transformation to the input, making the network capable of learning the nonlinearities in the data. Linear transformations would never be able to perform complicated tasks like language translations and image classifications.

Q47: What do we need to use GPUs to run Deep Learning Models?**Ans47:**

GPU's help to reduce the training time of the neural networks. Deep Neural networks can contain hundreds of hidden layers and nodes and can have millions of training parameters and consequently a very large number of matrix multiplication operations. Using GPU's allows us to perform all these operations parallelly at the same time instead of using CPU's, which can perform only one operation at a time.

Q48: What is the difference between generative and discriminative algorithms?**Ans48:**

Discriminative algorithms try to classify input data i.e, given the features of an instance of data, they predict a label or category to which that data belongs. For example, given an input email, a discriminative algorithm would predict whether the email is spam or not spam.

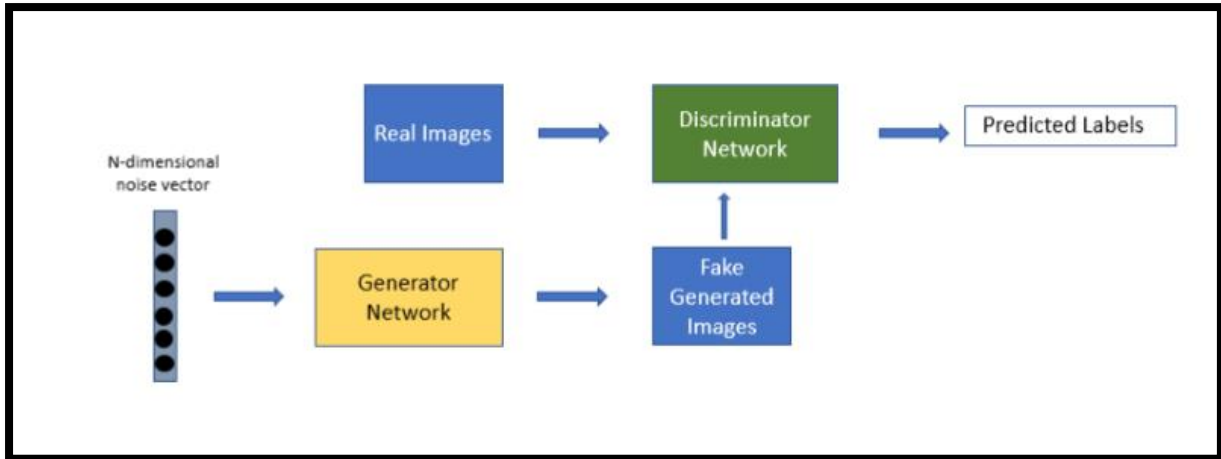
They can assign a label to a set of input features. If labels are represented as 'Y' and features as 'X', then they find $p(Y | X)$ (probability of Y given X), which in this case would translate to "the probability that an email is spam given the words it contains". Discriminative algorithms map features to labels.

On the other hand, the question a generative algorithm tries to answer is: Assuming that an email is spam, how likely are these features? While discriminative models model the relation between y and x, generative models care about "how you get x." They capture $p(X | Y)$ (the probability of x given y), or the probability of features given a label or category. While discriminative models learn the boundary between classes, generative models model the distribution of individual classes.

Q49: How do Generative Adversarial Networks work?**Ans49:**

One neural network, the generator, generates new data instances, while another neural network, the discriminator, evaluates them for authenticity, i.e. the discriminator decides whether each instance of data that it reviews belongs to the actual training dataset or not. Let's say we want to generate handwritten numerals, and we already have a dataset of handwritten digits from the real world. The goal of the discriminator, when shown an instance from the true dataset, is to recognize those that are authentic.

The generator creates new, synthetic images that it passes to the discriminator. It does so in the hopes that they will be deemed authentic, even though they are fake. The goal of the generator is to generate passable hand-written digits (to lie without being caught). The goal of the discriminator is to identify images coming from the generator as fake.



Q50: What are GAN's used for?

Ans50:

Given a training set, GAN technique learns to generate new data with the same statistics as the training set. For example, a GAN trained on photographs can generate new photographs that can look authentic to human observers, having many realistic characteristics. Some of the applications of GAN's are:

- GANs can be used to create photos of imaginary fashion models, with no need to hire a model, photographer, makeup artist.
- GAN's can be used as a method of up-scaling low-resolution 2D textures in old video games by recreating them in 4k or higher resolutions via image training.
- GANs can be used to show how an individual's appearance might change with age.

CHAPTER 4

INTERVIEW QUESTIONS ON NATURAL LANGUAGE PROCESSING (NLP)

(TOP 35 QUESTIONS)

Q1: Difference between NLP, NLU, and NLG?**Ans1:**

NLP	NLU	NLG
Stands for natural language processing.	Stands for natural language understanding.	Stands natural language generation.
Reads and converts the textual data into structured data.	Understands the textual and statistical data.	Converts the structures data into text and write the information in human language.
Superset of NLU and NLG.	Subset of NLP.	Subset of NLP.

Q2: What is perplexity in NLP?**Ans2:**

The meaning of the word ‘perplexed’ is ‘confused’. Thus, perplexity can be defined as not being able to solve the unidentified complex problem. In NLP, it can be defined as a way for language model evaluation. It is a way to find out the value of uncertainty in prediction. If the value is higher, it means the model has a bad performance because it isn’t able to deal with the highly complicated problems.

Q3: How do machines understand the languages in NLP?**Ans3:**

NLP follows a procedure to convert the language into a machine-understandable form. It follows the below steps:

- Tokenizes the data in the first step so that it can split each word.
- In the second step, it uses stemming or lemmatization to bring each word in its root form.
- In the third step, it removes punctuation and stopwords and assigns a POS tag to each word.
- In the next step, it converts words to their vector forms so that the machine can understand it.

Q4: Why do we need text mining?**Ans4:**

With the advancement of technologies, around 85% of data is in unstructured form(images, videos, text-like messages, articles, etc.) We need better techniques and algorithms to extract information from a large amount of textual data (unstructured data). That is the reason text mining and information

extraction came into existence so that it can help us for extracting useful and meaningful information using NLP.

Q5: Why should we use normalization in NLP?

Ans5:

Normalization is a preprocessing technique which can be used to convert the text (string) into words by removing punctuation and save the words in its base form so that it will become easy to extract the most useful information.

Q6: What is tokenization in NLP? Why is it used?

Ans6:

It is a process of converting the string into a list with one or more words as its element. It is used to split the string into words and punctuation. There are four basic techniques for tokenization.

- **Unigram:** In this type of tokenization, each word from a string becomes a separate element of the list.
- **Bigram:** Two consecutive words from a string acts as a tuple that becomes separate elements of a list.
- **Trigram:** Three consecutive words from a string acts a tuple that becomes a separate element of a list.
- **Ngrams:** One can choose how many consecutive words from a string acts a tuple to become the element of a list.

Q7: What are the applications of ngrams?

Ans7:

- n-grams are used in language modeling where the next word can be predicted on the basis of the occurrence of consecutive words. For example, the suggestion words in keypads.
- It can be used in various machine learning algorithms to design a kernel that can learn from string data.
- It can be used in compression algorithms to increase its compression rate, where a small amount of data requires n-grams with the highest value for parameter n.
- Medical record matching, information filtering, and music representation are some other application areas of ngrams.

Q8: Difference between stemming and lemmatization?**Ans8:**

Stemming	Lemmatization
Find the root word, even if the stemmed word isn't a natural language word.	Find the root word, only if the lemmatized word is a natural language word.
It uses a heuristic process that chops off the ends of words.	Uses vocabulary and morphological analysis to eliminate inflectional endings only.
Less computational time means it is faster.	More computational time means it is slower.

Q9: What is the purpose of using Stemming?**Ans9:**

Stemming is used to normalize the words into its base form or root form. For example, the word fish, fishes, and fishing all stem into fish. It can be done by using one of the three algorithms: Porter, Lancaster, and Snowball algorithm. The use of each stemmer depends on a type of task. Generally, Lancaster stemming is considered more aggressive than Porter stemming but in these types of stemming, there is no concept of multiple languages. That is the reason Snowball stemming is more preferred because there is an option for choosing a language.

Q10: How is lemmatization better than stemming?**Ans10:**

Lemmatization and stemming are techniques that are used to normalize the words into their root forms. But in lemmatization, the returned root word is always proper word unlike stemming where there is no assurance that the returned word is a proper word. Lemmatization uses both the WordNet corpus and a corpus for stop words to produce lemma which makes it slower than stemming because stemming normally aims to remove inflectional endings only.

Q11: Why is stopwords removal important?**Ans11:**

It is used to remove unnecessary words from a text. After Tokenization, lemmatization and punctuation removal, the text contains a word that doesn't serve any purpose for the analysis because these words are ignored by most of the search engines. Stopwords removal is the method to remove

common words from the text which will only increase the size of the index without improving the precision or recall.

Q12: What is dependency parsing in NLP?

Ans12:

Once the sentence is recognized, it has been assigned with a syntactic structure. This task is known as dependency parsing or syntactic parsing. We can simply generate a parse tree to solve such problems.

Q13: Difference between Shallow and Deep parsing?

Ans13:

Shallow Parsing	Deep Parsing
It is a rule-based approach.	It is a probabilistic approach.
Learns from rules/ grammar.	Uses probabilistic models to learn the rules/ grammar.
Uses a limited part of syntactic information from a text.	Uses the grammar concepts (CFG and PCFG) and search strategies.
Build a set of partial trees for one sentence.	Build a complete tree for a sentence.

Q14: What is a syntax tree?

Ans14:

The word 'syntax' simply means the arrangement of words. Therefore, syntax tree is a rooted tree that represents the syntactic structure of the sentence based on the grammar. This tree is also been called by the name of a parse tree or derivation tree.

Q15: What are the different types of grammar in the Chomsky hierarchy?

Ans15:

Noam Chomsky first formalized it in 1956 with his hierarchy of grammars. It is a containment hierarchy of classes for formal grammars in computer science and linguistics. According to him, it has four types:

- Unrestricted Grammar (UG)
- Context-Sensitive Grammar (CSG)
- Context-Free Grammar (CFG)
- Regular Grammar (RG)

Q16: What is CFG?**Ans16:**

It stands for context-free grammar. This grammar is accepted by pushdown automata. Push down automata is a machine that can recognize the patterns but unlike finite automata, it has extra memory called stack which can be used to recognize the context-free language.

Q17: How is regular grammar different from regular expression?**Ans17:**

A regular expression is just an expression that is used to check the pattern of the string by taking a sequence of characters as an input from a user that acts as a pattern. On the other hand, regular grammar can be defined as the rules for forming well-structured sentences, and the words that make up those sentences. In simple terms, we use regular expressions to check the regular grammar of a text.

Q18: Why do we use wild token patterns?**Ans18:**

As we already know the token attributes have many options to write highly specific patterns. To represent any token, we can also use an empty dictionary, as a wildcard. This is useful if we have the information about the structure of the text we want to match, but there is a specific token (or its characters) about which we don't have enough information. For example, let's say you're trying to extract people's usernames from the data. All we know is that they are listed as "User name: {username}". The name itself may contain any character, but no whitespace so it will be handled as one token. [{"find": "User"}, {"find": "name"}, {"find": ":"}, {}]

Q19: How chunking is different from chinking?**Ans19:**

Chunking is a strategy to create chunks by collecting data from short-term memories so that they can be used efficiently. After the chunking, we might find some data in the chunk that we still don't need.

The process of removing such chunk data from a chunk is known as chinking. In short, chunking creates chunks while chinking breaks up those chunks.

Q20: What is Text tagging?

Ans20:

It is a process of assigning tags or categories to text according to its content. This is also known as text categorization or text classification. It can be used to organize, structure, and categorize text. It can be used in sentiment analysis, topic labeling, Chatbots, spam detection, and intent detection. For example: organize new articles by topics, organize chat conversations by language, etc.

Q21: Why do we use POS tagging?

Ans21:

POS stands for parts of speech. POS tagging is a method to attach a tag with every word of a sentence that represents the type of POS (noun, verb, adverb, adjective, etc.) it belongs to. It is used to find out the structure of the sentence.

Q22: How are POS tags different from NER tags?

Ans22:

POS stands for parts of speech. POS tags are used to find the exact meaning of the words. In simple words, it finds out whether the word acts as a noun, verb or something else. On the other hand, NER stands for named entity recognition. NER can be used to find and sort the different categories like people, date, time, location, percent, cardinal numbers, ordinal numbers and organizations in text across many languages. In simple words, it can be used to find the naming words (nouns).

Q23: What is text Summarization?

Ans23:

Text summarization is a process of creating a short and coherent version of a longer document. In short, it is the process of creating a summarized content. There are two main approaches for doing this task:

- **Extracted Method:** It involves pulling keywords from the source document and combines them to make a summary. For example: Source Text: Joseph and Mary rode on a donkey to attend the event in Jerusalem, in the city Mary gave birth to a child named Jesus. Extracted summary: Joseph and Mary attend the event in Jerusalem. Mary births Jesus.

- **Abstracted Method:** It creates new phrases and sentences that relay the most useful information from the original text like humans. For example: Source Text: Joseph and Mary rode on a donkey to attend the event in Jerusalem, in the city Mary gave birth to a child named Jesus. Extracted Summary: Joseph and Mary came to Jerusalem where Jesus was born.

Q24: What is the purpose of the Bag of Words approach?

Ans24:

A list of words against their count is known as Bag of words. This approach can be used on any kind of text to decide from which category it belongs to. For example, there is a rating option on the movie site from 1 to 3 stars. We can never find out whether the user gave a positive rating or negative rating unless we read his/her comment (feedback). This approach can be used to decide whether the user's review is positive or negative.

Q25: What is the count vectorization matrix?

Ans25:

It is a matrix that contains the count for the occurrence of each word in a document and saves them in the form of vectors. The process for counting the frequencies of each word in a document is known as count vectorization.

Q26: What is Zipf's Law?

Ans26:

Zipf's law states that in a given corpus of text, the frequency of any word is inversely proportional to its rank in the frequency table. That means the most frequent words will occur around twice as compared to the second frequent words, three times to the third frequent words, four times to the fourth frequent words and so on.

Q27: Why is TF-IDF required in NLP?

Ans27:

It stands for term frequency-inverse term frequency. It is required because It is used to measure the importance of each keyword/ phrase by comparing it to the frequency of the term in a large set of documents. Mathematically TF-IDF score for keyword x in document y = $TF(x,y) * IDF(x)$

$TF(x,y) = X \text{ in document } y / \text{Total number of words in document}$

$IDF(x) = \log(\text{Total number of documents} / \text{Number of documents with keyword } x)$

Q28: Why is word embeddings important in NLP?**Ans28:**

Word embeddings are the representation of human words in the form that a machine can understand. In simple words, these are the vectors that represent the words. It is obvious that machines can't understand the words, so word embeddings are used to convert the words into the vector form. There are different methods for creating word embeddings that are useful in NLP applications. Three most used techniques are:

- **Word2Vec:** It is a group of models that takes a large corpus of text as an input and produces a vector space such that each unique word in the corpus being assigned a corresponding vector in the space. The words that share the common text in the corpus are placed close to each other. It uses either a continuous bag of words (CBOW) approach or a skip-gram model approach. CBOW is used to predict the gaps in the sentence while the skip-gram model works inversely because it is used to predict the surroundings for a given word.
- **GloVe:** This algorithm maps the words into a meaningful space where the distance between words is related to semantic similarity. It combines the features of two models SVG and word embedding concepts. It uses the concept of direct optimization so that the dot product of vectors of two words equals the log of the number of times the two words will occur near each other.
- **ELMO:** It stands for embedded form language models. It is a deep contextualized word representation that can be used to model both complex characteristics of word use (eg: syntax and semantics) and how these uses vary across linguistic contexts (i.e., to model polysemy). These word vectors are learned functions of the internal states of a deep bidirectional language model (BiLM), which is pre-trained on a large text corpus.

Q29: How is Glove different from word2vec?**Ans29:**

Word2Vec	GloVe
Found by Mikolov in 2013.	Founded by Pennington, Socher, and Manning in 2014.
Uses either CBOW or skip-gram model approach for vectoring the words.	Uses the combination of different approaches like SVD and word embedding concepts.
Used to predict the missing word from a sentence or the neighbors of the word.	Used for direct optimization so that the dot product of vectors of two words equals the log of the number of times the two words will occur near each other.
For example, 'He was ____ by a snake'. CBOW can predict a word like 'bitten' has a high probability of filling the gap. The skip-gram model can predict by neighboring words when the middle word (like 'bitten') is given.	For example, 'Romeo' and 'Juliet' occur nearby 15 times in a document, then $\text{Vec}(\text{Romeo}) \cdot \text{Vec}(\text{Juliet}) = \log(15)$.

Q30: Why do we need sentiment analysis?

Ans30:

Sentiment analysis is the analysis of emotions and opinions from the text. With each passing day, more content gets generated especially from the internet and social media. The contents like product reviews, social media posts, forums, and blogs are the goldmine of consumer sentiments. Such content provides exceptional insight into consumer opinions about their products. It can help the companies accomplish the task by enabling them to make faster and more appropriate decisions. This is the reason it is also known as opinion mining.

Q31: Suppose you have been given a file that contains large data. Your requirement is to split the text into sentences. How will you do it?

Ans31:

We can simply solve this kind of task using a textblob library. Textblob library has a lot of features that can split the data into words or sentences and saves a lot of time. For example: "Joseph and Mary rode on a donkey to attend the event in Jerusalem, in the city. Mary gave birth to a child named Jesus." We just have to pass the whole text to the textblob object (say obj = textblob("Text")). Then we can use obj.sentences that will split data into sentences.

Output:

[Sentence("Joseph and Mary rode on a donkey to attend the event in Jerusalem, in the city."), Sentence("Mary gave birth to a child named Jesus.")].

Q32: How will you find the incorrect spellings from the sentence and suggest the correct words for the user?

Ans32:

There are many approaches to solve this kind of problem. One of the easiest ways is by using a textblob library. Let me explain it step by step:

Step 1: import nltk and textblob libraries. *import nltk from textblob import TextBlob from textblob import Word*

Step 2: Put the sentence in the textblob and create its object, so that we can tokenize it by words. *Sent = TextBlob("check the senten ") words = Sent.words*

Step 3: Use a loop to check the spelling of each word and return the suggestions for incorrect words.

```
for i in words:
```

```
w = i.spellcheck()
```

```
if w[0][0] != 1:
```

```
    print(i, end = " ")
```

```
print(w)
```

Output:

```
check    [('check', 1.0)]  
the      [('the', 1.0)]  
sentenc  [('sentence', 1.0)]
```

Q33: What are chatbots? What is its purpose?**Ans33:**

Chatbots are chat robots because they mimic human interaction patterns. In simple words, it is a computer program that simulates human conversation. This interaction can be done by using a written text or voice. If someone asks a question that it can't answer, then it will either deflect the conversation or pass it to the human operator. The purpose of the chatbots is to help the companies or any person to increase their relationship with others. It can be programmed in any chat application to serve many purposes, such as:

- It can be accessible at any time. They will be active 24 x 7 and never get tired, unlike humans.
- It can answer the multiple questions of different customers at the same time without affecting his performance.
- Unlike humans, chatbots can provide quick answers to the queries because it doesn't need time to think since the answers are already saved.
- People can use chatbots as a doctor by just answering its questions and it will recommend them the prescription.

Q34: What are intents and entities in NLP?**Ans34:**

NLP is the processing of a natural language. We can say the intent is the same as verbs and entities are the same as nouns. It is possible to have multiple entities for a single intent or multiple intents for a single entity.

Q35: What are Actions and Story in NLP?

Ans35:

Actions are the responses given by the bot to answer the query/question of a user. The story defines all interaction between the user and bot on the basis of intent and the actions.