## 1. Artificial Intelligence Interview Questions

<https://www.geeksforgeeks.org/artificial-intelligenceai-interview-questions-and-answers/>

# Top Artificial Intelligence(AI) Interview Questions and Answers

(Last Updated : 17 Jan, 2025)

## *Basic Artificial Intelligence Interview Questions*

### 1. How does artificial intelligence differ from traditional programming?

[**Artificial Intelligence (AI)**](https://www.geeksforgeeks.org/artificial-intelligence/)is when machines, especially computers, are designed to think and act like humans. AI helps machines learn from information, solve problems, and improve themselves. It allows them to do tasks that usually need human intelligence, like understanding what they see (like recognizing images), understanding and responding to speech, making decisions, and translating languages.

**Differences from Traditional Programming**:

* **Rule-based vs. Learning-based**: Traditional programming involves explicitly coding rules and logic. AI, particularly machine learning, allows systems to learn patterns and rules from data.
* **Adaptability**: AI systems can adapt and improve over time with more data. Traditional programs remain static unless manually updated.
* **Complex Problem Solving**: AI can handle more complex and unstructured problems, while traditional programming is more suited for structured, well-defined tasks.

### 2. What are the main branches of AI?

The main branches of AI are as follows :

* **Machine Learning (ML)**: Algorithms that enable computers to learn from and make predictions based on data.
* **Natural Language Processing (NLP)**: The interaction between computers and humans through natural language.
* **Robotics**: Designing and building robots that can perform tasks autonomously or semi-autonomously.
* **Computer Vision**: Enabling computers to interpret and make decisions based on visual data.
* **Expert Systems**: AI programs that simulate the judgment and behavior of a human or an organization with expert knowledge.
* **Speech Recognition**: Converting spoken language into text.
* **Planning and Scheduling**: Algorithms for planning and optimizing tasks and resources.

### 3. What is the difference between a strong AI and a weak AI?

Difference between a strong AI and a weak AI is as follows :

* [**Strong AI (Artificial General Intelligence)**](https://www.geeksforgeeks.org/what-is-artificial-general-intelligence-agi/): Refers to machines with the ability to apply intelligence to any problem, rather than just specific ones. Strong AI systems can perform any intellectual task that a human can.
* [**Weak AI (Narrow AI)**](https://www.geeksforgeeks.org/what-is-narrow-ai/): Focused on performing a specific task or a narrow range of tasks. These systems are designed to handle only particular problems and do not possess general intelligence.

### 4. What is the difference between symbolic and connectionist AI?

The difference between symbolic and connectionist AI is as follows:

* [**Symbolic AI**](https://www.geeksforgeeks.org/what-is-symbolic-ai/): Uses explicit rules and logic to represent knowledge and solve problems. It relies on symbolic representation of knowledge, such as logic and rules.
* [**Connectionist AI**](https://www.geeksforgeeks.org/difference-between-symbolic-and-connectionist-ai/): Uses neural networks to simulate the human brain's interconnected neuron structure. Learning happens through the adjustment of weights between neurons based on input data.

### 5. What is the difference between parametric and non-parametric models?

The difference between parametric and non-parametric models are as follows:

* **Parametric Models**: Have a fixed number of parameters. Examples include linear regression and logistic regression. These models make assumptions about the data distribution.
* **Non-Parametric Models**: Do not assume a specific form for the data distribution and can have a flexible number of parameters. Examples include k-nearest neighbors (KNN) and decision trees. They can adapt to the complexity of the data.

### **6. What are the steps involved in deploying a machine learning model into production?**

The steps involved in deploying a Machine Learning Model into production typically includes:

* Preprocessing data and training the model.
* Evaluating model performance.
* Containerizing the model using tools like Docker.
* Deploying on platforms like AWS, GCP, or Azure.
* Monitoring and maintaining the model to ensure performance and scalability

### 7. What are the techniques used to avoid overfitting?

The techniques used to avoid overfitting are as follows:

* **Regularization**: Adding a penalty for larger coefficients in the model (e.g., L1 and L2 regularization).
* **Early Stopping**: Halting the training process before the model becomes too complex.
* **Dropout**: Randomly dropping units (along with their connections) from the neural network during training to prevent co-adaptation.
* **Data Augmentation**: Increasing the amount of training data by generating new samples through transformations.

### 8. What is the difference between batch learning and online learning?

The difference between batch learning and online learning are as follows:

**Batch Learning**: In batch learning, the model is trained using the whole dataset all at once. This means you need all the data ready before starting the training. It’s usually used when the model doesn’t need to be updated very often and can be retrained after some time.

**Online Learning**: In online learning, the model is trained little by little as new data comes in. It’s great for situations where data is constantly being generated, like in real-time systems, so the model keeps improving with every new piece of data.

### 9. What is the difference between eigenvalues and eigenvectors?

The difference between eigenvalues and eigenvectors are as follows:

* **Eigenvalues**: Scalars that indicate the magnitude by which the corresponding eigenvector is scaled during a linear transformation.
* **Eigenvectors**: Non-zero vectors that only change by a scalar factor when a linear transformation is applied. They represent the direction in which the transformation acts.

### 10. What are the different platforms for Artificial Intelligence (AI) development?

* **TensorFlow**: An open-source machine learning framework developed by Google.
* **PyTorch**: An open-source machine learning library developed by Facebook, known for its flexibility and ease of use.
* **Keras**: A high-level neural networks API that can run on top of TensorFlow, Theano, or CNTK.
* **Microsoft Azure AI**: A suite of AI services and tools offered by Microsoft.
* **Google Cloud AI**: AI and machine learning services provided by Google Cloud.
* **IBM Watson**: A suite of AI tools and applications developed by IBM.
* **Amazon SageMaker**: A fully managed service by AWS for building, training, and deploying machine learning models.
* **H2O.ai**: An open-source platform for AI and machine learning.
* **RapidMiner**: A data science platform that provides an integrated environment for data preparation, machine learning, deep learning, and predictive analytics.

### 11. Explain Diffusion Model architecture.

[Diffusion models](https://www.geeksforgeeks.org/what-are-diffusion-models/) are generative models that iteratively transform simple noise distributions into complex data distributions.

Key components include:

* **Forward Process**: Gradually adds noise to the data over several steps, leading to a noise distribution.
* **Reverse Process**: Learns to reverse the noise addition process, progressively denoising the data to generate new samples.

The model is trained to predict the noise added at each step, enabling it to generate realistic data by reversing the diffusion process. Diffusion models have shown impressive results in image and audio generation tasks.

### 12. Explain the different agents in Artificial Intelligence.

[Artificial Intelligence (AI) agents](https://www.geeksforgeeks.org/agents-artificial-intelligence/) can be classified into several types based on their capabilities and the complexity of their decision-making processes:

1. **Simple Reflex Agents**: These agents act solely on the current percept, ignoring the rest of the percept history. They use condition-action rules to decide actions.
2. **Model-Based Reflex Agents**: These agents maintain an internal state that depends on the percept history and reflects some of the unobserved aspects of the current state.
3. **Goal-Based Agents**: These agents act to achieve specific goals, considering future consequences of their actions. They use search and planning to decide the best actions.
4. **Utility-Based Agents**: These agents choose actions based on a utility function that evaluates the desirability of different states. They aim to maximize their overall happiness or satisfaction.
5. **Learning Agents**: These agents can learn from their experiences and adapt their behavior. They consist of four main components: the learning element, performance element, critic, and problem generator.

### 13. What is a rational agent, and what is rationality?

* [**Rational Agent**](https://www.geeksforgeeks.org/rational-agent-in-ai/): A rational agent is an agent that acts to achieve the best possible outcome or, when there is uncertainty, the best expected outcome. Rationality is about making the right decisions based on the current information and expected future benefits.
* [**Rationality**](https://www.geeksforgeeks.org/rationality-in-artificial-intelligence-ai/): Rationality refers to the quality of being based on or in accordance with reason or logic. In AI, an agent is considered rational if it consistently performs actions that maximize its performance measure, given its percept sequence and built-in knowledge.

### 14. How do coordination mechanisms impact agents in multiagent environments?

Coordination mechanisms are essential in multiagent environments to manage the interactions between agents. These mechanisms ensure that agents work together harmoniously, avoiding conflicts and enhancing collective performance. Common coordination mechanisms include:

* **Communication Protocols**: Methods for agents to exchange information and negotiate actions.
* **Distributed Planning**: Techniques that enable agents to plan their actions considering others' plans.
* **Market-Based Mechanisms**: Economic models where agents bid for tasks or resources.
* **Coordination Algorithms**: Algorithms designed to optimize the joint performance of all agents.

### 15. How does an agent formulate a problem?

An agent formulates a problem by defining the following components:

* **Initial State**: The starting point or condition of the problem.
* **Actions**: The set of possible actions the agent can take.
* **Transition Model**: The description of what each action does, i.e., the outcome of applying an action to a state.
* **Goal State**: The desired outcome or condition the agent aims to achieve.
* **Path Cost**: A function that assigns a cost to each path or sequence of actions.

### 16. What are the different types of search algorithms used in problem-solving?

Search algorithms are categorized into:

* **Uninformed Search Algorithms**: These algorithms have no additional information about states beyond the problem definition. Examples include:
  + Breadth-First Search (BFS)
  + Depth-First Search (DFS)
  + Uniform Cost Search
  + Iterative Deepening Search
* **Informed Search Algorithms**: These algorithms use heuristics to estimate the cost of reaching the goal from a given state. Examples include:
  + A\* Search
  + Greedy Best-First Search
  + Beam Search

### 17. What is the difference between informed and uninformed search AI algorithms?

The key difference between informed and uninformed search AI algorithms is as follows:

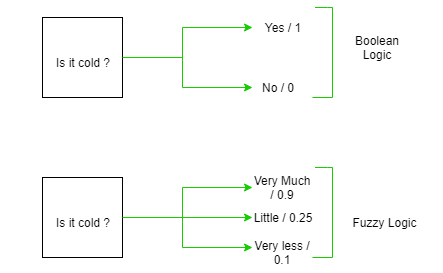
* [**Uninformed Search Algorithms**](https://www.geeksforgeeks.org/uniformed-search-algorithms-in-ai/): These algorithms do not have any domain-specific knowledge beyond the problem definition. They search through the problem space blindly, exploring all possible states.
* [**Informed Search Algorithms**](https://www.geeksforgeeks.org/informed-search-algorithms-in-artificial-intelligence/): These algorithms use heuristics, which provide additional information to guide the search more efficiently towards the goal. They can often find solutions faster and more efficiently than uninformed search algorithms.

### 18. What is the role of heuristics in local search algorithms?

[Heuristics](https://www.geeksforgeeks.org/role-of-heuristics-in-local-search-algorithms/) play a critical role in local search algorithms by providing a way to estimate how close a given state is to the goal state. This guidance helps the algorithm to make more informed decisions about which neighboring state to explore next, improving the efficiency and effectiveness of the search process.

### 19. What is Fuzzy Logic?

[Fuzzy Logic](https://www.geeksforgeeks.org/fuzzy-logic-introduction/) is a type of logic that deals with "in-between" values instead of just "true" or "false." It’s like saying something is "partly true" or "kind of false," making it useful for handling uncertain or vague information, like deciding if it’s "warm" or "cold" when the temperature is neither fully hot nor fully cold.



### 20. What Is Game Theory?

[Game Theory](https://www.geeksforgeeks.org/game-theory-in-ai/) is a branch of mathematics and economics that studies strategic interactions between rational decision-makers. It provides tools to analyze situations where multiple agents make decisions that affect each other's outcomes. Game theory concepts are used to model and predict behaviors in competitive and cooperative scenarios.

### 21. What is Reinforcement Learning, and explain the key components of a Reinforcement Learning problem?

[Reinforcement Learning (RL)](https://www.geeksforgeeks.org/what-is-reinforcement-learning/) is a type of machine learning where an agent learns to make decisions by taking actions in an environment to maximize cumulative rewards. The key components of an RL problem include:

* **Agent**: The learner or decision-maker.
* **Environment**: The external system with which the agent interacts.
* **State**: A representation of the current situation of the agent.
* **Actions**: The set of all possible moves the agent can make.
* **Reward**: A scalar feedback signal indicating the success of an action.
* **Policy**: A strategy that defines the agent's behavior by mapping states to actions.
* **Value Function**: A function that estimates the expected cumulative reward for each state or state-action pair.

### 22. What strategies do you use to optimize AI models for performance in production?

* **Model Quantization:** Reducing the precision of model weights (e.g., from 32-bit to 8-bit) to decrease size and improve inference speed.
* **Pruning:** Removing less important parts of the model (e.g., redundant neurons or weights) to reduce complexity and size.
* **Hardware Acceleration:** Utilizing GPUs, TPUs, or specialized AI chips to speed up computations.
* **Model Caching:** Storing frequently used results to avoid repeated computations.
* **Monitoring and Retraining:** Continuously monitoring model performance and retraining if performance declines due to data drift.

### 23. What are the different components of an expert system?

An [expert system](https://www.geeksforgeeks.org/expert-systems/) consists of several key components:

* **Knowledge Base**: A repository of domain-specific knowledge, including facts and rules.
* **Inference Engine**: The component that applies logical rules to the knowledge base to deduce new information and make decisions.
* **User Interface**: The means through which users interact with the expert system.
* **Explanation Facility**: Provides explanations of the reasoning process and the conclusions reached.
* **Knowledge Acquisition Module**: Tools and techniques used to gather and update the knowledge base.

### 24. What are embeddings in machine learning?

[Embeddings](https://www.geeksforgeeks.org/what-are-embeddings-in-machine-learning-2/) in machine learning are representations of objects, such as words or images, in a continuous vector space. These vectors capture semantic relationships and similarities between the objects. For example, in natural language processing, word embeddings map words to high-dimensional vectors that reflect their meanings and relationships based on their usage in large text corpora.

### 25. How does reward maximization work in Reinforcement Learning?

In Reinforcement Learning, [reward maximization](https://www.geeksforgeeks.org/how-does-reward-maximization-work-in-reinforcement-learning/) involves the agent taking actions that maximize its cumulative reward over time. The agent uses a policy to select actions based on the expected future rewards. The learning process involves updating the value function and policy to improve the expected rewards, typically using algorithms like Q-learning, SARSA, or policy gradient methods.

### 26. What is gradient descent in machine learning?

[Gradient Descent](https://www.geeksforgeeks.org/gradient-descent-algorithm-and-its-variants/) is an optimization algorithm used to minimize the loss function in machine learning models. It iteratively adjusts the model parameters in the opposite direction of the gradient of the loss function with respect to the parameters. The step size is determined by the learning rate.

Gradient descent variants include:

* **Batch Gradient Descent**: Uses the entire dataset to compute the gradient.
* [**Stochastic Gradient Descent (SGD)**](https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/): Uses one sample at a time to compute the gradient.
* [**Mini-Batch Gradient Descent**](https://www.geeksforgeeks.org/ml-mini-batch-gradient-descent-with-python/): Uses a small batch of samples to compute the gradient.

### 27. What is the difference between genetic algorithms and local search optimization algorithms?

The difference between genetic algorithms and local search optimization algorithms are as follows:

* [**Genetic Algorithms (GAs)**](https://www.geeksforgeeks.org/genetic-algorithms/): These are population-based optimization algorithms inspired by the process of natural selection. They use operators like selection, crossover, and mutation to evolve a population of solutions over generations.
* [**Local Search Algorithms**](https://www.geeksforgeeks.org/local-search-algorithm-in-artificial-intelligence/): These algorithms explore the solution space by moving from one solution to a neighboring solution, typically focusing on improving a single solution at a time. Examples include hill climbing and simulated annealing.

### 28. Discuss the concept of local optima and how it influences the effectiveness of local search algorithms.

[Local optima](https://www.geeksforgeeks.org/local-optima-and-influence-of-effectiveness-of-local-search-algorithms/)are solutions that are better than all their neighboring solutions but may not be the best overall solution (global optimum). Local search algorithms can get stuck in local optima, leading to suboptimal solutions. Techniques like simulated annealing and genetic algorithms are used to mitigate this issue by allowing occasional moves to worse solutions, helping the search escape local optima.

### 29. What is the difference between propositional logic and first-order logic, and how are they used in knowledge representation?

* The Key[difference between propositional logic and first-order logic](https://www.geeksforgeeks.org/difference-between-propositional-and-first-order-logic-and-how-are-they-used-in-knowledge-representation/) are as follows:
* **Propositional Logic**: Deals with propositions that can either be true or false. It uses logical connectives like AND, OR, and NOT to build complex statements.
* **First-Order Logic (FOL)**: Extends propositional logic by including quantifiers and predicates that can express relationships between objects. FOL is more expressive and can represent more complex statements about the world.

In knowledge representation, propositional logic is used for simple, straightforward scenarios, while first-order logic is used for more complex representations involving objects and their relationships.

### 30. Discuss the trade-offs between exploration and exploitation in local search algorithms.

In local search algorithms, exploration refers to the process of trying out new, possibly suboptimal solutions to discover better ones. Exploitation involves refining current solutions to improve them. The trade-off involves balancing the two: too much exploitation can lead to getting stuck in local optima, while too much exploration can waste resources and time. Effective algorithms balance both to find optimal solutions efficiently.

### 31. What are the differences between the hill climbing and simulated annealing algorithms?

The key differences between the hill climbing and simulated annealing algorithms are as follows:

* **Hill Climbing**: A local search algorithm that continuously moves towards better neighboring solutions. It can easily get stuck in local optima because it only considers immediate improvements.
* **Simulated Annealing**: A probabilistic algorithm that explores the solution space more broadly. It uses a temperature parameter to occasionally accept worse solutions, helping to escape local optima and potentially find the global optimum.

### 32. Explain the concept of a knowledge base in AI and discuss its role in intelligent systems.

A [knowledge base in AI](https://www.geeksforgeeks.org/role-of-knowledge-bases-in-intelligent-systems/) is a centralized repository of information, including facts, rules, and relationships about a particular domain. It enables intelligent systems to reason, make decisions, and solve problems by applying inference mechanisms to the stored knowledge. The knowledge base is crucial for expert systems, decision support systems, and other AI applications that rely on domain-specific information.

### 33. How do knowledge representation and reasoning techniques support intelligent systems?

[Knowledge representation](https://www.geeksforgeeks.org/knowledge-representation-and-reasoning-techniques-support-intelligent-systems/) and reasoning techniques provide the means to encode information about the world and manipulate it to derive new information, make decisions, and solve problems. They support intelligent systems by enabling:

* **Symbolic Representation**: Capturing complex relationships and entities in a structured form.
* **Logical Reasoning**: Applying rules and logic to infer new knowledge and make decisions.
* **Semantic Understanding**: Interpreting the meaning of information to provide contextually relevant responses.

### 34. State the differences between model-free and model-based Reinforcement Learning.

* [**Model-Free Reinforcement Learning**](https://www.geeksforgeeks.org/model-free-reinforcement-learning-an-overview/): The agent learns to make decisions based solely on the rewards received from the environment. It does not build a model of the environment's dynamics. Examples include Q-learning and SARSA.
* [**Model-Based Reinforcement Learning**](https://www.geeksforgeeks.org/model-based-reinforcement-learning-mbrl-in-ai/): The agent builds a model of the environment's dynamics and uses it to predict future states and rewards. It allows for planning and more efficient learning. Examples include Dyna-Q and Monte Carlo Tree Search.

### 35. What is Generative AI? What are some popular Generative AI architectures?

[Generative AI](https://www.geeksforgeeks.org/what-is-generative-ai/) refers to models that can generate new, original content based on the data they were trained on. These models can create text, images, music, and other media. Popular generative AI architectures include:

* [**Generative Adversarial Networks (GANs)**](https://www.geeksforgeeks.org/generative-adversarial-network-gan/): Consist of a generator and a discriminator that compete to produce realistic data.
* [**Variational Autoencoders (VAEs)**](https://www.geeksforgeeks.org/variational-autoencoders/): Use probabilistic methods to generate new data points similar to the training data.
* [**Transformer Models**](https://www.geeksforgeeks.org/getting-started-with-transformers/): Such as GPT-3 and DALL-E, which are capable of generating coherent text and images based on given prompts.

### 36. What are the key differences between zero-sum and non-zero-sum games?

* **Zero-Sum Games**: In these games, one person’s win means another person’s loss. The total amount of gains and losses always adds up to zero. For example, in chess or poker, if one player wins, the other loses by the same amount.
* **Non-Zero-Sum Games**: In these games, everyone’s outcome is not connected like a seesaw. All players can either win or lose together, depending on how they play. For example, in trade negotiations, both sides can benefit if they cooperate, or both can lose if they don’t agree. Similarly, in the Prisoner’s Dilemma, players can choose to help each other and gain, or betray and lose.

### 37. What is the concept of constraint satisfaction problem (CSP)?

A [**Constraint Satisfaction Problem (CSP)**](https://www.geeksforgeeks.org/constraint-satisfaction-problems-csp-in-artificial-intelligence/) is a mathematical problem defined by a set of variables, a domain of possible values for each variable, and a set of constraints specifying allowable combinations of values. The goal is to find a complete assignment of values to variables that satisfies all constraints. CSPs are used in scheduling, planning, and resource allocation problems.

### 38. What do you mean by inference in AI?

[Inference in AI](https://www.geeksforgeeks.org/inference-in-ai/)refers to the process of deriving new knowledge or conclusions from existing information using logical reasoning. It involves applying rules and algorithms to known data to infer new facts, make predictions, or solve problems. Inference is a key component of expert systems, decision-making processes, and machine learning models.

### 39. What are the advantages and disadvantages of forward chaining and backward chaining inference in rule-based systems?

The advantages and disadvantages of forward chaining and backward chaining inference in rule-based systems are as follows:

**Forward Chaining**:

* Advantages: Efficient for problems where all data is available from the start. Suitable for data-driven scenarios.
* Disadvantages: Can generate a large number of intermediate facts, leading to inefficiency.

**Backward Chaining**:

* Advantages: Goal-directed, focusing on relevant data and rules. Efficient for goal-driven scenarios.
* Disadvantages: Can be inefficient if the search space is large or if there are many possible rules to apply.

### 40. How do Bayesian networks model probabilistic relationships between variables?

[Bayesian networks](https://www.geeksforgeeks.org/basic-understanding-of-bayesian-belief-networks/) model probabilistic relationships using a directed acyclic graph (DAG) where nodes represent random variables, and edges represent conditional dependencies. Each node has a probability distribution that quantifies the effect of its parents. Bayesian networks allow for efficient representation and computation of joint probabilities, enabling reasoning under uncertainty and probabilistic inference.

### 41. What are the key differences between Q-learning and SARSA?

The key differences between Q-learning and SARSA are as follows

**1.**[**Q-learning**](https://www.geeksforgeeks.org/q-learning-in-python/)

* **Type**: Off-policy learning algorithm.
* **Update Rule**: Uses the maximum possible reward of the next state (greedy policy) for updates.
* **Exploration**: Does not depend on the current policy being followed; can explore different actions.
* **Formula**: Q(s,a)←Q(s,a)+α[r+γmax⁡a′Q(s′,a′)−Q(s,a)]*Q*(*s*,*a*)←*Q*(*s*,*a*)+*α*[*r*+*γ*max*a*′​*Q*(*s*′,*a*′)−*Q*(*s*,*a*)]

**2.**[**SARSA (State-Action-Reward-State-Action)**](https://www.geeksforgeeks.org/sarsa-reinforcement-learning/)

* **Type**: On-policy learning algorithm.
* **Update Rule**: Uses the actual reward of the next action taken (current policy) for updates.
* **Exploration**: Follows the current policy, considering exploration during updates.
* **Formula**: Q(s,a)←Q(s,a)+α[r+γQ(s′,a′)−Q(s,a)]*Q*(*s*,*a*)←*Q*(*s*,*a*)+*α*[*r*+*γQ*(*s*′,*a*′)−*Q*(*s*,*a*)]

### 42. Discuss the concept of alpha-beta pruning in adversarial search algorithms.

[Alpha-beta pruning](https://www.geeksforgeeks.org/alpha-beta-pruning-in-adversarial-search-algorithms/) is an optimization technique for the minimax algorithm in adversarial search (e.g., game playing). It eliminates branches in the search tree that cannot affect the final decision, thus reducing the number of nodes evaluated.

* **Alpha (α)**: The best value that the maximizer currently can guarantee at that level or above.
* **Beta (β)**: The best value that the minimizer currently can guarantee at that level or below.

During the search, branches are pruned if:

* **Maximizer**: Finds a move that is better than the current beta value (beta cut-off).
* **Minimizer**: Finds a move that is worse than the current alpha value (alpha cut-off).

### 43. Explain the concept of backtracking search and its role in finding solutions to CSPs.

[Backtracking search](https://www.geeksforgeeks.org/explain-the-concept-of-backtracking-search-and-its-role-in-finding-solutions-to-csps/) is a depth-first search algorithm for solving Constraint Satisfaction Problems (CSPs). It incrementally builds candidates for the solutions and abandons a candidate ("backtracks") as soon as it determines that the candidate cannot possibly be completed to a valid solution.

* **Role in CSPs**: Backtracking is used to systematically explore the possible assignments of values to variables while ensuring that the constraints are satisfied. If a partial assignment violates a constraint, the algorithm backtracks to the previous step to try a different value.

### 44. Explain the role of the minimax algorithm in adversarial search for optimal decision-making.

The [minimax algorithm](https://www.geeksforgeeks.org/minimax-algorithm-in-adversarial-search-for-optimal-decision-making/) is used in decision-making for two-player games, where one player (maximizer) tries to maximize their score while the other player (minimizer) tries to minimize it. The algorithm evaluates the game tree, considering all possible moves:

* **Maximizer's Turn**: Chooses the move with the highest score.
* **Minimizer's Turn**: Chooses the move with the lowest score.

The goal is to find the optimal strategy by assuming that both players play optimally. The minimax algorithm recursively evaluates the game tree until the terminal nodes, assigning values to each move and choosing the best move for the current player.

### 45. Explain the A\* algorithm and its heuristic search strategy.

[A\* algorithm](https://www.geeksforgeeks.org/a-algorithm-and-its-heuristic-search-strategy-in-artificial-intelligence/) is used to find the shortest path in a graph. It uses both the actual cost to reach a node (g(n)) and a heuristic estimate of the cost to reach the goal from that node (h(n)).

* **Formula**: f(n)=g(n)+h(n)
* **Heuristic**: A function that estimates the cost of reaching the goal from the current node. It guides the search process by prioritizing nodes with the lowest estimated total cost (f(n)).

A\* efficiently finds the shortest path by balancing exploration of new nodes (guided by the heuristic) and exploitation of known paths (guided by the actual cost).

### 46. Explain the concept of the Markov Decision Process (MDP) and its relevance to Reinforcement Learning.

[Markov Decision Process (MDP)](https://www.geeksforgeeks.org/what-is-markov-decision-process-mdp-and-its-relevance-to-reinforcement-learning/) is a mathematical framework for modeling decision-making in environments with stochastic outcomes. It consists of:

* **States (S)**: The possible situations in the environment.
* **Actions (A)**: The set of all possible actions the agent can take.
* **Transition Model (P)**: The probability of moving from one state to another, given an action.
* **Rewards (R)**: The immediate reward received after transitioning from one state to another.
* **Policy (π)**: A strategy that specifies the action to take in each state.

In Reinforcement Learning, MDPs provide the foundation for defining the environment, modeling the agent's interactions, and optimizing the policy to maximize cumulative rewards.

### 47. Explain the Hidden Markov Model.

[Hidden Markov Model (HMM)](https://www.geeksforgeeks.org/hidden-markov-model-in-machine-learning/) is a statistical model used to represent systems that have hidden (unobservable) states. It consists of:

* **States**: A set of hidden states the system can be in.
* **Observations**: The observed data, which are probabilistically related to the hidden states.
* **Transition Probabilities**: The probabilities of transitioning between hidden states.
* **Emission Probabilities**: The probabilities of observing a certain output given a hidden state.
* **Initial Probabilities**: The probabilities of starting in each hidden state.

HMMs are used in various applications like speech recognition, natural language processing, and bioinformatics to model sequences with underlying hidden patterns.

### 48. Explain the concept of autoencoders in deep learning.

Autoencoders are a type of neural network used for unsupervised learning, specifically for dimensionality reduction and feature learning. They consist of two main parts:

* **Encoder**: Maps the input data to a lower-dimensional latent space.
* **Decoder**: Reconstructs the input data from the latent representation.

The goal is to train the network so that the output closely matches the input, forcing the model to learn efficient representations of the data. Variants like denoising autoencoders and variational autoencoders add additional constraints or probabilistic elements to improve robustness and generative capabilities.

### 49. Explain Generative Adversarial Networks (GANs) architecture.

[Generative Adversarial Networks (GANs)](https://www.geeksforgeeks.org/generative-adversarial-network-gan/) consist of two neural networks, a generator and a discriminator, that compete in a zero-sum game:

* **Generator**: Creates fake data resembling the real data.
* **Discriminator**: Distinguishes between real and fake data.

A **GAN** works like a game between a **fake image creator (generator)** and a **fake image detector (discriminator)**. The creator tries to make fake images look real, while the detector tries to tell real images from fake ones. They improve together, and over time, the creator becomes so good that the detector can't tell the difference between real and fake images.

### 50. Explain Transformer Model architecture.

[Transformer model](https://www.geeksforgeeks.org/getting-started-with-transformers/) was introduced in the paper "Attention is All You Need," revolutionized natural language processing with its attention mechanisms and parallel processing capabilities.

Key components include:

* **Self-Attention Mechanism**: Allows each input token to attend to all other tokens, capturing long-range dependencies.
* **Positional Encoding**: Adds information about the position of tokens in the sequence.
* **Encoder-Decoder Structure**:
  + **Encoder**: Consists of multiple layers, each with self-attention and feed-forward neural networks.
  + **Decoder**: Similar to the encoder but includes an additional attention layer to attend to the encoder's output.

Transformers enable efficient training on large datasets and achieve state-of-the-art performance in tasks like machine translation and text generation.

## 2. Machine Learning Interview Questions

<https://www.geeksforgeeks.org/machine-learning-interview-questions/>

# (Top 50+ Machine Learning Interview Questions and Answers

Last Updated : 27 Dec, 2024)

## *Machine Learning Interview Questions For Freshers*

### **1. What are some real-life applications of clustering algorithms?**

[Clustering algorithms](https://www.geeksforgeeks.org/clustering-in-machine-learning/) are used in various real-life applications such as:

* Customer segmentation for targeted marketing
* Recommendation systems for personalized suggestions
* Anomaly detection in fraud prevention
* Image compression to reduce storage
* Healthcare for grouping patients with similar conditions
* Document categorization in search engines

### **2. How to choose an optimal number of clusters?**

* [**Elbow Method**](https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/): Plot the explained variance or within-cluster sum of squares (WCSS) against the number of clusters. The "elbow" point, where the curve starts to flatten, indicates the optimal number of clusters.
* [**Silhouette Score**](https://www.geeksforgeeks.org/silhouette-algorithm-to-determine-the-optimal-value-of-k/): Measures how similar each point is to its own cluster compared to other clusters. A higher silhouette score indicates better-defined clusters. The optimal number of clusters is the one with the highest average silhouette score.
* **Gap Statistic**: Compares the clustering result with a random clustering of the same data. A larger gap between the real and random clustering suggests a more appropriate number of clusters.

### **3. What is feature engineering? How does it affect the model’s performance?**

[Feature engineering](https://www.geeksforgeeks.org/what-is-feature-engineering/) refers to developing some new features by using existing features. Sometimes there is a very subtle mathematical relation between some features which if explored properly then the new features can be developed using those mathematical operations.

Also, there are times when multiple pieces of information are clubbed and provided as a single data column. At those times developing new features and using them help us to gain deeper insights into the data as well as if the features derived are significant enough helps to improve the model’s performance a lot.

### **4. What is overfitting in machine learning and how can it be avoided?**

[Overfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/) happens when the model learns patterns as well as the noises present in the data this leads to high performance on the training data but very low performance for data that the model has not seen earlier.

To avoid overfitting there are multiple methods that we can use:

* Early stopping of the model’s training in case of validation training stops increasing but the training keeps going on.
* Using regularization methods like L1 or L2 regularization which is used to penalize the model's weights to avoid overfitting.

### **5. Why we cannot use linear regression for a classification task?**

The main reason why we cannot use [linear regression](https://www.geeksforgeeks.org/ml-linear-regression/) for a classification task is that the output of linear regression is continuous and unbounded, while classification requires discrete and bounded output values.

If we use linear regression for the classification task the error function graph will not be convex. A convex graph has only one minimum which is also known as the global minima but in the case of the non-convex graph, there are chances of our model getting stuck at some local minima which may not be the global minima. To avoid this situation of getting stuck at the local minima we do not use the linear regression algorithm for a classification task.

### **6. Why do we perform normalization?**

To achieve stable and fast training of the model we use [normalization](https://www.geeksforgeeks.org/what-is-data-normalization/) techniques to bring all the features to a certain scale or range of values. If we do not perform normalization then there are chances that the gradient will not converge to the global or local minima and end up oscillating back and forth.

### **7. What is the difference between precision and recall?**

Precision is the ratio between the true positives(TP) and all the positive examples (TP+FP) predicted by the model. In other words, precision measures how many of the predicted positive examples are actually true positives. It is a measure of the model's ability to avoid false positives and make accurate positive predictions.

Precision=TPTP+FPPrecision=*TP*+*FPTP*​

In recall, we calculate the ratio of true positives (TP) and the total number of examples (TP+FN) that actually fall in the positive class. Recall measures how many of the actual positive examples are correctly identified by the model. It is a measure of the model's ability to avoid false negatives and identify all positive examples correctly.

Recall=TPTP+FNRecall=*TP*+*FNTP*​

### **8. What is the difference between upsampling and downsampling?**

In upsampling method, we increase the number of samples in the minority class by randomly selecting some points from the minority class and adding them to the dataset repeat this process till the dataset gets balanced for each class. But, here is a disadvantage the training accuracy becomes high as in each epoch model trained more than once in each epoch but the same high accuracy is not observed in the validation accuracy.

In downsampling, we decrease the number of samples in the majority class by selecting some random number of points that are equal to the number of data points in the minority class so that the distribution becomes balanced. In this case, we have to suffer from data loss which may lead to the loss of some critical information as well.

### **9. What is data leakage and how can we identify it?**

If there is a high correlation between the target variable and the input features then this situation is referred to as data leakage. This is because when we train our model with that highly correlated feature then the model gets most of the target variable's information in the training process only and it has to do very little to achieve high accuracy. In this situation, the model gives pretty decent performance both on the training as well as the validation data but as we use that model to make actual predictions then the model’s performance is not up to the mark. This is how we can identify data leakage.

### **10. Explain the classification report and the metrics it includes.**

The [**classification report**](https://www.geeksforgeeks.org/compute-classification-report-and-confusion-matrix-in-python/) provides key metrics to evaluate a model’s performance, including:

* **Precision**: The proportion of true positives to all predicted positives, measuring accuracy of positive predictions.
* **Recall**: The proportion of true positives to all actual positives, indicating how well the model finds positive instances.
* **F1-Score**: The harmonic mean of precision and recall, balancing the two metrics.
* **Support**: The number of true instances for each class in the dataset.
* **Accuracy**: The overall proportion of correct predictions.
* **Macro Average**: The average of precision, recall, and F1-score across all classes, treating them equally.
* **Weighted Average**: The average of metrics, weighted by class support, giving more importance to frequent classes.

### **11. What are some of the hyperparameters of the random forest regressor which help to avoid overfitting?**

The most important hyperparameters of a Random Forest are:

* **max\_depth:** Sometimes the larger depth of the tree can create overfitting. To overcome it, the depth should be limited.
* **n-estimator:**It is the number of decision trees we want in our forest.
* **min\_sample\_split:** It is the minimum number of samples an internal node must hold in order to split into further nodes.
* **max\_leaf\_nodes:** It helps the model to control the splitting of the nodes and in turn, the depth of the model is also restricted.

### **12. What is the bias-variance tradeoff?**

First, let’s understand what is bias and variance:

* **Bias**refers to the difference between the actual values and the predicted values by the model. Low bias means the model has learned the pattern in the data and high bias means the model is unable to learn the patterns present in the data i.e the underfitting.
* **Variance**refers to the change in accuracy of the model's prediction on which the model has not been trained. Low variance is a good case but high variance means that the performance of the training data and the validation data vary a lot.

If the bias is too low but the variance is too high then that case is known as overfitting. So, finding a balance between these two situations is known as the [**bias-variance trade-off**](https://www.geeksforgeeks.org/ml-bias-variance-trade-off/).

### **13. Is it always necessary to use an 80:20 ratio for the train test split?**

No, there is no such necessary condition that the data must be split into 80:20 ratio. The main purpose of the splitting is to have some data which the model has not seen previously so, that we can evaluate the performance of the model.

If the dataset contains let’s say 50,000 rows of data then only 1000 or maybe 2000 rows of data is enough to evaluate the model’s performance.

### **14. What is Principal Component Analysis?**

[**PCA(Principal Component Analysis)**](https://www.geeksforgeeks.org/ml-principal-component-analysispca) is an unsupervised machine learning dimensionality reduction technique in which we trade off some information or patterns of the data at the cost of reducing its size significantly. In this algorithm, we try to preserve the variance of the original dataset up to a great extent let’s say 95%. For very high dimensional data sometimes even at the loss of 1% of the variance, we can reduce the data size significantly.

By using this algorithm we can perform image compression, visualize high-dimensional data as well as make data visualization easy.

### **15. What is one-shot learning?**

[**One-shot learning**](https://www.geeksforgeeks.org/one-shot-learning-in-machine-learning-1/) is a concept in machine learning where the model is trained to recognize the patterns in datasets from a single example instead of training on large datasets. This is useful when we haven't large datasets. It is applied to find the similarity and dissimilarities between the two images.

### **16. What is the difference between Manhattan Distance and Euclidean distance?**

Both [Manhattan Distance](https://www.geeksforgeeks.org/maximum-manhattan-distance-between-a-distinct-pair-from-n-coordinates/) and [Euclidean distance](https://www.geeksforgeeks.org/pairs-with-same-manhattan-and-euclidean-distance) are two distance measurement techniques.

* Manhattan Distance (MD) is calculated as the sum of absolute differences between the coordinates of two points along each dimension.

*MD*=∣*x*1​−*x*2​∣+ ∣*y*1​−*y*2​∣

* Euclidean Distance (ED) is calculated as the square root of the sum of squared differences between the coordinates of two points along each dimension.

ED=(x1−x2)2+(y1−y2)2*ED*=(*x*1​−*x*2​)2+(*y*1​−*y*2​)2​

Generally, these two metrics are used to evaluate the effectiveness of the clusters formed by a clustering algorithm.

### **17. What is the difference between one hot encoding and ordinal encoding?**

[One Hot encoding](https://www.geeksforgeeks.org/ml-one-hot-encoding/) and ordinal encoding both are different methods to convert categorical features to numeric ones the difference is in the way they are implemented. In one hot encoding, we create a separate column for each category and add 0 or 1 as per the value corresponding to that row.

In [ordinal encoding](https://www.geeksforgeeks.org/how-to-perform-ordinal-encoding-using-sklearn/), we replace the categories with numbers from 0 to n-1 based on the order or rank where n is the number of unique categories present in the dataset. The main difference between one-hot encoding and ordinal encoding is that one-hot encoding results in a binary matrix representation of the data in the form of 0 and 1, it is used when there is no order or ranking between the dataset whereas ordinal encoding represents categories as ordinal values.

### **18. How can you conclude about the model's performance using the confusion matrix?**

[Confusion matrix](https://www.geeksforgeeks.org/confusion-matrix-machine-learning/)summarizes the performance of a classification model. In a confusion matrix, we get four types of output (in case of a binary classification problem) which are TP, TN, FP, and FN. As we know that there are two diagonals possible in a square, and one of these two diagonals represents the numbers for which our model's prediction and the true labels are the same. Our target is also to maximize the values along these diagonals. From the confusion matrix, we can calculate various evaluation metrics like accuracy, precision, recall, F1 score, etc.

### **19. Explain the working principle of SVM.**

A data set that is not separable in different classes in one plane may be separable in another plane. This is exactly the idea behind the [**SVM**](https://www.geeksforgeeks.org/support-vector-machine-algorithm)in this a low dimensional data is mapped to high dimensional data so, that it becomes separable in the different classes. A hyperplane is determined after mapping the data into a higher dimension which can separate the data into categories.

SVM model can even learn non-linear boundaries with the objective that there should be as much margin as possible between the categories in which the data has been categorized. To perform this mapping different types of kernels are used like radial basis kernel, gaussian kernel, polynomial kernel, and many others.

### **20. What is the difference between the k-means and k-means++ algorithms?**

The only difference between the two is in the way centroids are initialized. In the [k-means algorithm](https://www.geeksforgeeks.org/k-means-clustering-introduction/), the centroids are initialized randomly from the given points. There is a drawback in this method that sometimes this random initialization leads to non-optimized clusters due to maybe initialization of two clusters close to each other.

To overcome this problem k-means++ algorithm was formed. In k-means++, the first centroid is selected randomly from the data points. The selection of subsequent centroids is based on their separation from the initial centroids. The probability of a point being selected as the next centroid is proportional to the squared distance between the point and the closest centroid that has already been selected. This guarantees that the centroids are evenly spread apart and lowers the possibility of convergence to less-than-ideal clusters. This helps the algorithm reach the global minima instead of getting stuck at some local minima.

Read more about it [here](https://www.geeksforgeeks.org/ml-k-means-algorithm).

### **21. Explain some measures of similarity which are generally used in Machine learning.**

Some of the most commonly used similarity measures are as follows:

* **Cosine Similarity:** By considering the two vectors in n - dimension we evaluate the cosine of the angle between the two. The range of this similarity measure varies from [-1, 1] where the value 1 represents that the two vectors are highly similar and -1 represents that the two vectors are completely different from each other.
* **Euclidean or Manhattan Distance:**These two values represent the distances between the two points in an n-dimensional plane. The only difference between the two is in the way the two are calculated.
* **Jaccard Similarity:** It is also known as IoU or Intersection over union it is widely used in the field of object detection to evaluate the overlap between the predicted bounding box and the ground truth bounding box.

### **22. Whether decision tree or random forest is more robust to the outliers.**

Decision trees and random forests are both relatively robust to outliers. A random forest model is an ensemble of multiple decision trees so, the output of a random forest model is an aggregate of multiple decision trees.

So, when we average the results the chances of overfitting get reduced. Hence we can say that the random forest models are more robust to outliers.

### **23. What is the difference between L1 and L2 regularization? What is their significance?**

[**L1 regularization (Lasso regularization)**](https://www.geeksforgeeks.org/what-is-lasso-regression/)adds the sum of the absolute values of the model's weights to the loss function. This penalty encourages sparsity in the model by pushing the weights of less important features to exactly zero. As a result, L1 regularization automatically performs **feature selection**, removing irrelevant or redundant features from the model, which can improve interpretability and reduce overfitting.

[**L2 regularization (Ridge regularization)**](https://www.geeksforgeeks.org/what-is-ridge-regression/) in which we add the square of the weights to the loss function. In both of these regularization methods, weights are penalized but there is a subtle difference between the objective they help to achieve.

In L2 regularization the weights are not penalized to 0 but they are near zero for irrelevant features. It is often used to prevent overfitting by shrinking the weights towards zero, especially when there are many features and the data is noisy.

### **24. What is a radial basis function?**

[RBF (radial basis function)](https://www.geeksforgeeks.org/radial-basis-function-kernel-machine-learning)is a real-valued function used in machine learning whose value only depends upon the input and fixed point called the center.

The formula for the radial basis function is as follows:

K(x,x′)=exp(−∥x−x′∥22σ2)*K*(*x*,*x*′)=*exp*(−2*σ*2∥∥​*x*−*x*′∥∥​2​)

Machine learning systems frequently use the RBF function for a variety of functions, including:

* RBF networks can be used to approximate complex functions. By training the network's weights to suit a set of input-output pairs,
* RBF networks can be used for unsupervised learning to locate data groups. By treating the RBF centers as cluster centers,
* RBF networks can be used for classification tasks by training the network's weights to divide inputs into groups based on how far from the RBF nodes they are.

It is one of the very famous kernels which is generally used in the SVM algorithm to map low dimensional data to a higher dimensional plane so, we can determine a boundary that can separate the classes in different regions of those planes with as much margin as possible.

### **25. Explain SMOTE method used to handle data imbalance.**

In [SMOTE](https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/), we synthesize new data points using the existing ones from the minority classes by using linear interpolation. The advantage of using this method is that the model does not get trained on the same data. But the disadvantage of using this method is that it adds undesired noise to the dataset and can lead to a negative effect on the model’s performance.

### **26. Does the accuracy score always a good metric to measure the performance of a classification model?**

No, there are times when we train our model on an imbalanced dataset the accuracy score is not a good metric to measure the performance of the model. In such cases, we use precision and recall to measure the performance of a classification model.

Also, f1-score is another metric that can be used to measure performance but in the end, f1-score is also calculated using precision and recall as the f1-score is nothing but the harmonic mean of the precision and recall.

### **27.**What is KNN Imputer and how does it work?

[KNN Imputer](https://www.geeksforgeeks.org/python-imputation-using-the-knnimputer/)imputes missing values in a dataset compared to traditional methods like using mean, median, or mode. It is based on the **K-Nearest Neighbors (KNN)** algorithm, which fills missing values by referencing the values of the nearest neighbors.

Here’s how it works:

* **Neighborhood-based Imputation:** The KNN Imputer identifies the **k nearest neighbors** to the data point with the missing value, based on a distance metric (e.g., Euclidean distance).
* **Imputation Process:** Once the nearest neighbors are found, the missing value is imputed (filled) using a statistical measure, such as the mean or median, of the values from these neighbors.
* **Distance Parameter:** The **k parameter** is used to define how many neighbors to consider when imputing a missing value, and the distance metric controls how similarity is measured between data points.

### **28. Explain the working procedure of the XGBoost model.**

[XGBoost model](https://www.geeksforgeeks.org/ml-xgboost-extreme-gradient-boosting) is an ensemble technique of machine learning in this method weights are optimized in a sequential manner by passing them to the decision trees. After each pass, the weights become better and better as each tree tries to optimize the weights, and finally, we obtain the best weights for the problem at hand. Techniques like regularized gradient and mini-batch gradient descent have been used to implement this algorithm so, that it works in a very fast and optimized manner.

### **29. What is the purpose of splitting a given dataset into training and validation data?**

The main purpose is to keep some data left over on which the model has not been trained so, that we can evaluate the performance of our machine learning model after training. Also, sometimes we use the validation dataset to choose among the multiple state-of-the-art machine learning models. Like we first train some models let’s say LogisticRegression, XGBoost, or any other than test their performance using validation data and choose the model which has less difference between the validation and the training accuracy.

### **30. Explain some methods to handle missing values in that data.**

Some of the [methods to handle missing](https://www.geeksforgeeks.org/working-with-missing-data-in-pandas) values are as follows:

* Removing the rows with null values may lead to the loss of some important information.
* Removing the column having null values if it has very less valuable information. it may lead to the loss of some important information.
* Imputing null values with descriptive statistical measures like mean, mode, and median.
* Using methods like KNN Imputer to impute the null values in a more sophisticated way.

### **31. What is the difference between k-means and the KNN algorithm?**

K-means algorithm is one of the popular unsupervised machine learning algorithms which is used for clustering purposes. But, KNN is a model which is generally used for the classification task and is a supervised machine learning algorithm. The k-means algorithm helps us to label the data by forming clusters within the dataset.

### **32. What is Linear Discriminant Analysis?**

[Linear Discriminant Analysis (LDA)](https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/) is a supervised machine learning dimensionality reduction technique because it uses target variables also for dimensionality reduction. It is commonly used for classification problems. The LDA mainly works on two objectives:

* Maximize the distance between the means of the two classes.
* Minimize the variation within each class.

### **33. How can we visualize high-dimensional data in 2-d?**

One of the most common and effective methods is by using the t-SNE algorithm which is a short form for t-Distributed Stochastic Neighbor Embedding. This algorithm uses some non-linear complex methods to reduce the dimensionality of the given data. We can also use PCA or LDA to convert n-dimensional data to 2 - dimensional so, that we can plot it to get visuals for better analysis. But the difference between the PCA and t-SNE is that the former tries to preserve the variance of the dataset but the t-SNE tries to preserve the local similarities in the dataset.

### **34. What is the reason behind the curse of dimensionality?**

As the dimensionality of the input data increases the amount of data required to generalize or learn the patterns present in the data increases. For the model, it becomes difficult to identify the pattern for every feature from the limited number of datasets or we can say that the weights are not optimized properly due to the high dimensionality of the data and the limited number of examples used to train the model. Due to this after a certain threshold for the dimensionality of the input data, we have to face the curse of dimensionality.

### **35.**Which metric is more robust to outliers: MAE, MSE, or RMSE?

Out of the three metrics—Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)—**MAE** is more robust to outliers.

The reason behind this is the way each metric handles error values:

* **MSE and RMSE** both square the error values. When there are outliers, the error is typically large, and squaring it results in even larger error values. This causes outliers to disproportionately affect the overall error, leading to misleading results and potentially distorting the model’s performance.
* **MAE**, on the other hand, takes the absolute value of the errors. Since it does not square the error terms, the influence of large errors (outliers) is linear rather than exponential, making MAE less sensitive to outliers.

### **36. Why removing highly correlated features are considered a good practice?**

When two features are highly correlated, they may provide similar information to the model, which may cause overfitting. If there are highly correlated features in the dataset then they unnecessarily increase the dimensionality of the feature space and sometimes create the problem of the curse of dimensionality. If the dimensionality of the feature space is high then the model training may take more time than expected, it will increase the complexity of the model and chances of error. This somehow also helps us to achieve data compression as the features have been removed without much loss of data.

### **37. What is the difference between the content-based and collaborative filtering algorithms of recommendation systems?**

In a content-based recommendation system, similarities in the content and services are evaluated, and then by using these similarity measures from past data we recommend products to the user. But on the other hand in collaborative filtering, we recommend content and services based on the preferences of similar users.

**For example**, if one user has taken A and B services in past and a new user has taken service A then service A will be recommended to him based on the other user's preferences.

### 38. How you would assess the goodness-of-fit for a linear regression model? Which metrics would you consider most important and why?

To evaluate the performance of a linear regression model, important key metrics are: R-squared, Adjusted R-squared, RMSE, and F-Statistics. R-squared is particularly important as it reflects the proportion of variance in the dependent variable that can be explained by the independent variables, providing a measure of how well our model fits the data. However, Adjusted R-squared also plays a crucial role, especially when comparing models with different numbers of predictors. It adjusts for the complexity of the model, helping to prevent overfitting and ensuring the robustness of our findings.

To learn more about regression metrics, check out: [**Regression Metrics**](https://www.geeksforgeeks.org/regression-metrics/#:~:text=Squared%20Error%3A%200.057999999999999996-,R%2Dsquared%20(R%C2%B2)%20Score,model's%20independent%20variables%20contribute%20to.)

### 39. What is the null hypothesis in linear regression problem?

In linear regression, the null hypothesis id that there is no relationship between the independent variable(s) and the dependent variable. This is formally represented as H0:β1=0*H*0​:*β*1​=0, where β1*β*1​​ is the coefficient of the independent variable.

Essentially, the null hypothesis suggests that the predictor variable does not contribute to predicting the outcome. For instance, if the null hypothesis states that the slope of the regression line is zero, then a student's score in an English class would not be a useful predictor of their overall grade-point average.

The alternative hypothesis, denoted as H1:β1≠0*H*1​:*β*1​=0, proposes that changes in the independent variable are indeed associated with changes in the dependent variable, indicating a meaningful relationship.

### 40. Can SVMs be used for both classification and regression tasks?

Yes, Support Vector Machines (SVMs) can be used for both classification and regression. For classification, SVMs work by finding a hyperplane that separates different classes in the data with the largest gap possible.

For regression, which involves predicting a continuous number, SVMs are adapted into a version called Support Vector Regression (SVR). SVR tries to fit as many data points as possible within a certain range of the predicted line, allowing some errors but penalizing those that are too large. This makes it useful for predicting values in situations where the data shows complex patterns.

To learn how to implement Support Vector Regression, you can refer to: [**Support Vector Regression (SVR) using Linear and Non-Linear Kernels in Scikit Learn**](https://www.geeksforgeeks.org/support-vector-regression-svr-using-linear-and-non-linear-kernels-in-scikit-learn/)

### 41. Explain the concept of weighting in KNN? What are the different ways to assign weights, and how do they affect the model's predictions?

Weighting in KNN assigns different levels of importance to the neighbors based on their distance from the query point, influencing how each neighbor affects the model's predictions.

The weights can be assigned using:

* **Uniform Weighting:** All neighbors have equal weight regardless of their distance.
* **Distance Weighting:** Weights are inversely proportional to the distance, giving closer neighbors more influence.
* **User-defined Weights:** Weights are assigned based on domain knowledge or specific data characteristics.

Effect on Model's Prediction:

* **Uniform Weighting:** Simple but may not perform well with noisy data or varied distances.
* **Distance Weighting:** Improves accuracy by emphasizing closer neighbors, useful for irregular class boundaries but sensitive to anomalies.
* **User-defined Weights:** Optimizes performance when specific insights about the dataset are applied, though less generalizable.

### 42. What are the assumptions behind the K-means algorithm? How do these assumptions affect the results?

The assumptions of K-Means algorithm include:

1. **Cluster Shape:** Assumes clusters are spherical and of similar size, affecting how well it handles non-spherical groups.
2. **Scale of Features:** Assumes features are on similar scales; different ranges can distort the distance calculation.
3. **Clusters are Balanced:** Assumes clusters have a roughly equal number of observations, which can bias results against smaller clusters.
4. **Similar Density:** Assumes all clusters have similar density, impacting the algorithm's effectiveness with clusters of varying densities.

If these assumptions are not met, the model will perform poorly making difficult to process and select clustering techniques that align with the data characteristics.

Check out the article: [**K Means Clustering Assumptions**](https://www.geeksforgeeks.org/demonstration-of-k-means-assumptions/)

### 43. **Can you explain the concept of convergence in K-means? What conditions must be met for K-means to converge?**

Convergence in K-means occurs when the cluster centroids stabilize, and the assignment of data points to clusters no longer changes. This happens when the algorithm has minimized the sum of squared distances between points and their corresponding centroids.

**Conditions for K-means to Converge:**

1. **Proper Initialization:** The initial placement of centroids significantly impacts convergence. Techniques like k-means++ help ensure a better start.
2. **Data Characteristics:** The algorithm converges more effectively if the data naturally clusters into well-separated groups. Overlapping or complex cluster shapes can hinder convergence.
3. **Correct Number of Clusters (k):** Choosing the right number of clusters is critical; too many or too few can lead to slow convergence or convergence to suboptimal solutions.
4. **Algorithm Parameters:** Setting a maximum number of iterations and a small tolerance for centroid change helps prevent infinite loops and determines when the algorithm should stop if centroids move minimally between iterations.

### 44. **What is the significance of tree pruning in XGBoost? How does it affect the model?**

Tree pruning in XGBoost is used to reduce model complexity and prevent overfitting.[XGBoost](https://www.geeksforgeeks.org/xgboost/)implements a "pruning-as-you-grow" strategy where it starts by growing a full tree up to a maximum depth, then prunes back the branches that contribute minimal gains in terms of loss reduction. This is guided by the gamma parameter, which sets a minimum loss reduction required to make further partitions on a leaf node.

**Effect on the Model:**

1. **Reduces Overfitting:** By trimming unnecessary branches, pruning helps in creating simpler models that generalize better to unseen data, reducing the likelihood of overfitting.
2. **Improves Performance:** Pruning helps in removing splits that have little impact, which can enhance the model's performance by focusing on more significant attributes.
3. **Optimizes Computational Efficiency:** It decreases the complexity of the final model, which can lead to faster training and prediction times as there are fewer nodes to traverse during decision making.

### 45. How does Random Forest ensure diversity among the trees in the model?

Random Forest ensures diversity among the trees in its ensemble through two main mechanisms:

1. [**Bootstrap Aggregating (Bagging)**](https://www.geeksforgeeks.org/ml-bagging-classifier/)**:** Each tree in a Random Forest is trained on a different bootstrap sample, a random subset of the data. This sampling with replacement means that each tree sees different portions of the data, leading to variations in their learning and decision-making processes.
2. **Feature Randomness:** When splitting a node during the construction of the tree, Random Forest randomly selects a subset of features instead of using all available features. This variation in the feature set ensures that trees do not follow the same paths or use the same splits, thereby increasing the diversity among the trees.

The diversity among trees reduces the variance of the model without significantly increasing the bias.

### 46. What is the concept of information gain in decision trees? How does it guide the creation of the tree structure?

[Information gain](https://www.geeksforgeeks.org/information-gain-and-mutual-information-for-machine-learning/)is a measure used in decision trees to select the best feature that splits the data into the most informative subsets. It is calculated based on the reduction in entropy or impurity after a dataset is split on an attribute. Entropy is a measure of the randomness or uncertainty in the data set, and information gain quantifies how much splitting on a particular attribute reduces that randomness.

### 47. How does the independence assumption affect the accuracy of a Naive Bayes classifier?

[Naive Bayes classifier](https://www.geeksforgeeks.org/naive-bayes-classifiers/) operates under the assumption that all features in the dataset are independent of each other given the class label. This assumption simplifies the computation of the classifier's probability model, as it allows the conditional probability of the class given multiple features to be calculated as the product of the individual probabilities for each feature.

Affect of accuracy on a Naive Bayes classifier:

1. **Strengths in High-Dimensional Data:** In practice, the independence assumption can sometimes lead to good performance, especially in high-dimensional settings like text classification, despite the interdependencies among features. This is because the errors in probability estimates may cancel out across the large number of features.
2. **Limitations Due to Feature Dependency:** The accuracy of Naive Bayes can be adversely affected when features are not independent, particularly if the dependencies between features are strong and critical to predicting the class. The model may underperform in such scenarios because it fails to capture the interactions between features.
3. **Generalization Capability:** The simplistic nature of the independence assumption often allows Naive Bayes to perform well on smaller datasets or in cases where data for training is limited, as it does not require as complex a model as other classifiers.

### 48. Why does PCA maximize the variance in the data?

[PCA](https://www.geeksforgeeks.org/principal-component-analysis-pca/)aims to **maximize the variance** because variance represents how much information is spread out in a given direction. The higher the variance along a direction, the more information that direction holds about the data. By focusing on the directions of highest variance, PCA helps us:

* **Preserve information** while reducing the dimensionality.
* **Simplify the data** by eliminating less important features (those with low variance)

### 49. **How do you evaluate the effectiveness of a machine learning model in an imbalanced dataset scenario? What metrics would you use instead of accuracy?**

We can use Precision, Recall, F1 score and ROC-AUC to evaluate the effectiveness of machine learning model in imbalanced dataset scenario. The best metric is F1 score as it combines both precision and recall into single metric that is important in imbalanced datasets where a high number of true negatives can skew accuracy. By focusing on both false positives and false negatives, the F1-score ensures that both the positive class detection and false positives are accounted for.

* If the cost of false positives (Type I errors) and false negatives (Type II errors) is similar, F1-Score strikes a good balance.
* It is especially useful when you need to prioritize performance in detecting the minority class (positive class).

However, if you are more concerned about false positives or false negatives specifically, you may opt for:

* Precision (if false positives are more costly) or
* Recall (if false negatives are more costly).

### 50. How the One-Class SVM algorithm works for anomaly detection?

[One-Class SVM](https://www.geeksforgeeks.org/understanding-one-class-support-vector-machines/) is an unsupervised anomaly detection algorithm. It is often used when only normal data is available. The model learns a decision boundary around normal data points using a kernel, typically an RBF, to map the data into a higher-dimensional space. The algorithm identifies support vectors—data points closest to the boundary—and any new data point outside this boundary is flagged as an anomaly. Key parameters like 'nu' control the fraction of outliers allowed, while the kernel defines the boundary shape.

### 51. Explain the concept of "concept drift" in anomaly detection.

[**Concept drift**](https://www.geeksforgeeks.org/introduction-to-concept-drift/) refers to the change in the underlying distribution or patterns in the data over time, which can make previously normal data points appear as anomalies. In anomaly detection, this is particularly challenging because a model trained on old data may not recognize new, evolving patterns as part of the normal data distribution. Concept drift can occur suddenly or gradually and needs to be monitored closely. To address this, models can be adapted through periodic retraining with new data or by using adaptive anomaly detection algorithms.

## 3. Deep Learning Interview Questions

# <https://www.geeksforgeeks.org/deep-learning-interview-questions/>

# (Last Updated : 02 Sep, 2024)

**Deep learning**

is a part of machine learning that is based on the artificial neural network with multiple layers to learn from and make predictions on data. An artificial neural network is based on the structure and working of the Biological neuron which is found in the brain.

## *Deep Learning Interview Questions For Freshers*

### 1. What is Deep Learning?

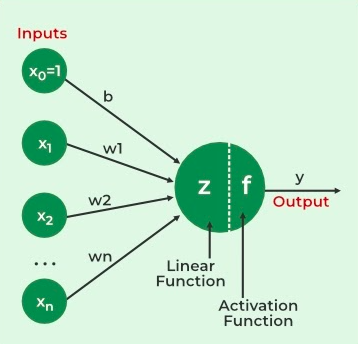
[Deep learning](https://www.geeksforgeeks.org/introduction-deep-learning) is the branch of machine learning which is based on [artificial neural network](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications)architecture which makes it capable of learning complex patterns and relationships within data. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work togeather to process and learn from the input data.

In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer. The output of one neuron becomes the input to other neurons in the next layer of the network, and this process continues until the final layer produces the output of the network. The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the input data.

 Today Deep learning has become one of the most popular and visible areas of machine learning, due to its success in a variety of applications, such as computer vision, natural language processing, and Reinforcement learning.

### 2. What is an artificial neural network?

An[artificial neural network](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications) is inspired by the networks and functionalities of human biological neurons. it is also known as neural networks or neural nets. ANN uses layers of interconnected nodes called artificial neurons that work together to process and learn the input data. The starting layer artificial neural network is known as the input layer, it takes input from external input sources and transfers it to the next layer known as the hidden layer where each neuron received inputs from previous layer neurons and computes the weighted sum, and transfers to the next layer neurons. These connections are weighted means effects of the inputs from the previous layer are optimized more or less by assigning different-different weights to each input and it is adjusted during the training process by optimizing these weights for better performance of the model. The output of one neuron becomes the input to other neurons in the next layer of the network, and this process continues until the final layer produces the output of the network.



artificial neural network

### 3. How does Deep Learning differ from Machine Learning?

[Machine learning](https://www.geeksforgeeks.org/machine-learning) and [deep learning](https://www.geeksforgeeks.org/deep-learning-tutorial) both are subsets of [artificial intelligence](https://www.geeksforgeeks.org/artificial-intelligence-an-introduction) but there are many similarities and differences between them.

| **Machine Learning** | **Deep Learning** |
| --- | --- |
| Apply statistical algorithms to learn the hidden patterns and relationships in the dataset. | Uses artificial neural network architecture to learn the hidden patterns and relationships in the dataset. |
| Can work on the smaller amount of dataset | Requires the larger volume of dataset compared to machine learning |
| Better for the low-label task. | Better for complex tasks like image processing, natural language processing, etc. |
| Takes less time to train the model. | Takes more time ta o train the model. |
| A model is created by relevant features which are manually extracted from images to detect an object in the image. | Relevant features are automatically extracted from images. It is an end-to-end learning process. |
| Less complex and easy to interpret the result. | More complex, it works like the black box interpretations of the result are not easy. |
| It can work on the CPU or requires less computing power as compared to deep learning. | It requires a high-performance computer with GPU. |

### 4. What are the applications of Deep Learning?

Deep learning has many applications, and it can be broadly divided into computer vision, natural language processing (NLP), and reinforcement learning.

* [**Computer vision**](https://www.geeksforgeeks.org/computer-vision-introduction):  Deep learning employs neural networks with several layers, which enables it used for automated learning and recognition of complex patterns in images. and machines can perform image classification, image segmentation, object detection, and image generation task accurately. It has greatly increased the precision and effectiveness of computer vision algorithms, enabling a variety of uses in industries including healthcare, transportation, and entertainment.
* [**Natural language processing (NLP)**](https://www.geeksforgeeks.org/natural-language-processing-nlp-tutorial): Natural language processing (NLP) gained enormously from deep learning, which has enhanced language modeling, sentiment analysis, and machine translation. Deep learning models have the ability to automatically discover complex linguistic features from text data, enabling more precise and effective processing of inputs in natural language.
* [**Reinforcement learning**](https://www.geeksforgeeks.org/what-is-reinforcement-learning): Deep learning is used in reinforcement learning to evaluate the value of various actions in various states, allowing the agent to make better decisions that can maximize the predicted rewards. By learning from these mistakes, an agent eventually raises its performance. Deep learning applications that use reinforcement learning include gaming, robotics, and control systems.

### 5. What are the challenges in Deep Learning?

[Deep learning](https://www.geeksforgeeks.org/introduction-deep-learning) has made significant advancements in various fields, but there are still some challenges that need to be addressed. Here are some of the main challenges in deep learning:

1. Data availability: It requires large amounts of data to learn from. For using deep learning it's a big concern to gather as much data for training.
2. Computational Resources: For training the deep learning model, it is computationally expensive because it requires specialized hardware like GPUs and TPUs.
3. Time-consuming: While working on sequential data depending on the computational resource it can take very large even in days or months.
4. Interpretability: Deep learning models are complex, it works like a black box. it is very difficult to interpret the result.
5. Overfitting: when the model is trained again and again, it becomes too specialized for the training data, leading to overfitting and poor performance on new data.

### 6. How Biological neurons are similar to the Artificial neural network.

The concept of [artificial neural networks](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications) comes from [biological neurons](https://www.geeksforgeeks.org/difference-between-ann-and-bnn) found in animal brains So they share a lot of similarities in structure and function wise.

* **Structure:** The structure of artificial neural networks is inspired by biological neurons. A biological neuron has dendrites to receive the signals, a cell body or soma to process them, and an axon to transmit the signal to other neurons.  In artificial neural networks input signals are received by input nodes, hidden layer nodes compute these input signals, and output layer nodes compute the final output by processing the outputs of the hidden layer using activation functions.
* **Synapses:** In biological neurons, synapses are the connections between neurons that allow for the transmission of signals from dendrites to the cell body and the cell body to the axon like that. In artificial neurons, synapses are termed as the weights which connect the one-layer nodes to the next-layer nodes. The weight value determines the strength between the connections.
* **Learning:** In biological neurons, learning occurs in the cell body or soma which has a nucleus that helps to process the signals. If the signals are strong enough to reach the threshold, an action potential is generated that travels through the axons. This is achieved by synaptic plasticity, which is the ability of synapses to strengthen or weaken over time, in response to increases or decreases in their activity. In artificial neural networks, the learning process is called backpropagations, which adjusts the weight between the nodes based on the difference or cost between the predicted and actual outputs.
* **Activation:** In biological neurons, activation is the firing rate of the neuron which happens when the signals are strong enough to reach the threshold. and in artificial neural networks, activations are done by mathematical functions known as activations functions which map the input to the output.

### 7. How deep learning is used in supervised, unsupervised as well as reinforcement machine learning?

Deep learning can be used for supervised, unsupervised as well as reinforcement machine learning. it uses a variety of ways to process these.

* [**Supervised Machine Learning**](https://www.geeksforgeeks.org/supervised-unsupervised-learning)**:** Supervised machine learning is the machine learning technique in which the neural network learns to make predictions or classify data based on the labeled datasets. Here we input both input features along with the target variables. the neural network learns to make predictions based on the cost or error that comes from the difference between the predicted and the actual target, this process is known as backpropagation.  Deep learning algorithms like Convolutional neural networks, Recurrent neural networks are used for many supervised tasks like image classifications and recognization, sentiment analysis, language translations, etc.
* [**Unsupervised Machine Learning**](https://www.geeksforgeeks.org/supervised-unsupervised-learning)**:** Unsupervised machine learning is the machine learning technique in which the neural network learns to discover the patterns or to cluster the dataset based on unlabeled datasets. Here there are no target variables. while the machine has to self-determined the hidden patterns or relationships within the datasets. Deep learning algorithms like autoencoders and generative models are used for unsupervised tasks like clustering, dimensionality reduction, and anomaly detection.
* [**Reinforcement  Machine Learning**](https://www.geeksforgeeks.org/what-is-reinforcement-learning): Reinforcement  Machine Learning is the machine learning technique in which an agent learns to make decisions in an environment to maximize a reward signal. The agent interacts with the environment by taking action and observing the resulting rewards. Deep learning can be used to learn policies, or a set of actions, that maximizes the cumulative reward over time. Deep reinforcement learning algorithms like Deep Q networks and Deep Deterministic Policy Gradient (DDPG) are used to reinforce tasks like robotics and game playing etc.

### 8. What is a Perceptron?

[Perceptron](https://www.geeksforgeeks.org/single-layer-perceptron-in-tensorflow) is one of the simplest Artificial neural network architectures. It was introduced by Frank Rosenblatt in 1957s. It is the simplest type of feedforward neural network, consisting of a single layer of input nodes that are fully connected to a layer of output nodes. It can learn the linearly separable patterns. it uses slightly different types of artificial neurons known as threshold logic units (TLU). it was first introduced by McCulloch and Walter Pitts in the 1940s. it computes the weighted sum of its inputs and then applies the step function to compare this weighted sum to the threshold. the most common step function used in perceptron is the Heaviside step function.

A [perceptron](https://www.geeksforgeeks.org/single-layer-perceptron-in-tensorflow) has a single layer of threshold logic units with each TLU connected to all inputs. When all the neurons in a layer are connected to every neuron of the previous layer, it is known as a fully connected layer or dense layer. During training, The weights of the perceptron are adjusted to minimize the difference between the actual and predicted value using the perceptron learning rule i.e

w\_i = w\_i + (learning\_rate \* (true\_output - predicted\_output) \* x\_i)

Here, x\_i and w\_i are the ith input feature and the weight of the ith input feature.

### 9.  What is Multilayer Perceptron? and How it is different from a single-layer perceptron?

A [multilayer perceptron (MLP)](https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow) is an advancement of the single-layer perceptron which uses more than one hidden layer to process the data from input to the final prediction. It consists of multiple layers of interconnected neurons, with multiple nodes present in each layer. The MLP architecture is referred to as the feedforward neural network because data flows in one direction, from the input layer through one or more hidden layers to the output layer.

The differences between the single-layer perceptron and multilayer perceptron are as follows:

* **Architecture:** A single-layer perceptron has only one layer of neurons, which takes the input and produces an output. While a multilayer perceptron has one or more hidden layers of neurons between the input and output layers.
* **Complexity:** A single-layer perceptron is a simple linear classifier that can only learn linearly separable patterns. While a multilayer perceptron can learn more complex and nonlinear patterns by using nonlinear activation functions in the hidden layers.
* **Learning:** Single-layer perceptrons use a simple perceptron learning rule to update their weights during training. While multilayer perceptrons use a more complex backpropagation algorithm to train their weights, which involves both forward propagations of input through the network and backpropagation of errors to update the weights.
* **Output:** Single-layer perceptrons produce a binary output, indicating which of two possible classes the input belongs to. Multilayer perceptrons can produce real-valued outputs, allowing them to perform regression tasks in addition to classification.
* **Applications:**Single-layer perceptrons are suitable for simple linear classification tasks whereas Multilayer perceptrons are more suitable for complex classification tasks where the input data is not linearly separable, as well as for regression tasks where the output is continuous variables.

### 10. What are Feedforward Neural Networks?

A [feedforward neural network (FNN)](https://www.geeksforgeeks.org/understanding-multi-layer-feed-forward-networks) is a type of artificial neural network, in which the neurons are arranged in layers, and the information flows only in one direction, from the input layer to the output layer, without any feedback connections. The term "feedforward" means information flows forward through the neural network in a single direction from the input layer through one or more hidden layers to the output layer without any loops or cycles.

In a [feedforward neural network (FNN)](https://www.geeksforgeeks.org/understanding-multi-layer-feed-forward-networks) the weight is updated after the forward pass. During the forward pass, the input is fed and it computes the prediction after the series of nonlinear transformations to the input. then it is compared with the actual output and errors are calculated.

During the backward pass also known as [backpropagation](https://www.geeksforgeeks.org/backpropagation-in-data-mining), Based on the differences, the error is first propagated back to the output layer, where the gradient of the loss function with respect to the output is computed. This gradient is then propagated backward through the network to compute the gradient of the loss function with respect to the weights and biases of each layer. Here chain rules of calculus are applied with respect to weight and bias to find the gradient. These gradients are then used to update the weights and biases of the network so that it can improve its performance on the given task.

### 11. What is GPU?

A [graphics processing unit](https://www.geeksforgeeks.org/what-is-a-graphics-card), sometimes known as a GPU, is a specialized electronic circuit designed to render graphics and images on a computer or other digital device fast and effectively.

Originally developed for use in video games and other graphical applications, GPUs have grown in significance in a number of disciplines, such as artificial intelligence, machine learning, and scientific research, where they are used to speed up computationally demanding tasks like training deep neural networks.

One of the main benefits of GPUs is their capacity for parallel computation, which uses a significant number of processing cores to speed up complicated calculations. Since high-dimensional data manipulations and matrix operations are frequently used in machine learning and other data-driven applications, these activities are particularly well suited for them.

### 12. What are the different layers in ANN? What is the notation for representing a node of a particular layer?

There are commonly three different types of layers in an [artificial neural network (ANN)](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications):

* **Input Layer:** This is the layer that receives the input data and passes it on to the next layer. The input layer is typically not counted as one of the hidden layers of the network.
* **Hidden Layers:**The input layer is the one that receives input data and transfers it to the next layer. Usually, the input layer is not included in the list of the hidden layers of the neural network.
* **Output Layer:**This is the output-producing layer of the network. A binary classification problem might only have one output neuron, but a multi-class classification problem might have numerous output neurons, one for each class. The number of neurons in the output layer depends on the type of problem being solved.

We commonly use a notation like N[i][L]  *N*[*i*][*L*]​   to represent a node of a specific layer in an ANN, where L denotes the layer number and i denotes the node's index inside that layer. For instance, the input layer's first node may be written as N[0][1]  *N*[0][1]​   whereas the third hidden layer's second node might be written as N[2][3]  *N*[2][3]​   With this notation, it is simple to refer to specific network nodes to understand the structure of the network as a whole.

### 13. What is forward and backward propagation?

In [deep learning and neural networks](https://www.geeksforgeeks.org/introduction-deep-learning), In the [forward pass or propagation](https://www.geeksforgeeks.org/deep-neural-net-with-forward-and-back-propagation-from-scratch-python), The input data propagates through the input layer to the hidden layer to the output layer. During this process, each layer of the neural network performs a series of mathematical operations on the input data and transfers it to the next layer until the output is generated.

Once the [forward propagation](https://www.geeksforgeeks.org/deep-neural-net-with-forward-and-back-propagation-from-scratch-python) is complete, the [backward propagation](https://www.geeksforgeeks.org/deep-neural-net-with-forward-and-back-propagation-from-scratch-python), also known as [backpropagation](https://www.geeksforgeeks.org/backpropagation-in-data-mining) or back prop, is started. During the backward pass, the generated output is compared to the actual output and based on the differences between them the error is measured and it is propagated backward through the neural network layer. Where the gradient of the loss function with respect to the output is computed. This gradient is then propagated backward through the network to compute the gradient of the loss function with respect to the weights and biases of each layer. Here chain rules of calculus are applied with respect to weight and bias to find the gradient. These gradients are then used to update the weights and biases of the network so that it can improve its performance on the given task.

 In simple terms, the forward pass involves feeding input data into the neural network to produce an output,  while the backward pass refers to utilizing the output to compute the error and modify the network's weights and biases.

### 14. What is the cost function in deep learning?

The cost function is the mathematical function that is used to measure the quality of prediction during training in deep neural networks. It measures the differences between the generated output of the forward pass of the neural network to the actual outputs, which are known as losses or errors. During the training process, the weights of the network are adjusted to minimize the losses. which is achieved by computing the gradient of the cost function with respect to weights and biases using backpropagation algorithms.

 The cost function is also known as the loss function or objective function. In deep learning, different -different types of cost functions are used depending on the type of problem and neural network used.  Some of the common cost functions are as follows:

* [Binary Cross-Entropy](https://www.geeksforgeeks.org/cross-entropy-cost-functions-used-in-classification) for binary classification measures the difference between the predicted probability of the positive outcome and the actual outcome.
* [Categorical Cross-Entropy](https://www.geeksforgeeks.org/cross-entropy-cost-functions-used-in-classification)for multi-class classification measures the difference between the predicted probability and the actual probability distribution.
* [Sparse Categorical Cross-Entropy](https://www.geeksforgeeks.org/cross-entropy-cost-functions-used-in-classification) for multi-class classification is used when the actual label is an integer rather than in a one-hot encoded vector.
* [Kullback-Leibler Divergence (KL Divergence)](https://www.geeksforgeeks.org/kullback-leibler-divergence) is used in generative learning like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), it measures the differences between two probability distributions.
* Mean Squared Error for regression to measure the average squared difference between actual and predicted outputs.

### 15. What are activation functions in deep learning and where it is used?

[Deep learning](https://www.geeksforgeeks.org/introduction-deep-learning) uses [activation functions](https://www.geeksforgeeks.org/activation-functions), which are mathematical operations that are performed on each neuron's output in a neural network to provide nonlinearity to the network. The goal of activation functions is to inject non-linearity into the network so that it can learn the more complex relationships between the input and output variables.

In other words, the activation function in neural networks takes the output of the preceding linear operation (which is usually the weighted sum of input values i.e w\*x+b) and mapped it to a desired range because the repeated application of weighted sum (i.e w\*x +b) will result in a polynomial function. The activation function transformed the linear output into non-linear output which makes the neural network capable to approximate more complex tasks.

In deep learning, To compute the gradients of the loss function with respect to the network weights during backpropagation, activation functions must be differentiable. As a result, the network may use gradient descent or other optimization techniques to find the optimal weights to minimize the loss function.

Although several activation functions, such as ReLU, and Hardtanh, contain point discontinuities, they are still differentiable almost everywhere. The gradient is not defined at the point of discontinuity, This does not have a substantial impact on the network's overall gradient because the gradient at these points is normally set to zero or a small value.

### 16. What are the different different types of activation functions used in deep learning?

In deep learning, several different-different types of [activation functions](https://www.geeksforgeeks.org/activation-functions) are used. Each of them has its own strength and weakness. Some of the most common activation functions are as follows.

* [**Sigmoid function**](https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function)**:** It maps any value between 0 and 1. It is mainly used in binary classification problems. where it maps the output of the preceding hidden layer into the probability value.
* [**Softmax function**](https://www.geeksforgeeks.org/softmax-regression-using-keras)**:**It is the extension of the sigmoid function used for multi-class classification problems in the output layer of the neural network, where it maps the output of the previous layer into a probability distribution across the classes, giving each class a probability value between 0 and 1 with the sum of the probabilities over all classes is equal to 1. The class which has the highest probability value is considered as the predicted class.
* [**ReLU (Rectified Linear Unit) function**](https://www.geeksforgeeks.org/how-to-apply-rectified-linear-unit-function-element-wise-in-pytorch)**:** It is a non-linear function that returns the input value for positive inputs and 0 for negative inputs. Deep neural networks frequently employ this function since it is both straightforward and effective.
* [**Leaky ReLU function**](https://www.geeksforgeeks.org/python-tensorflow-nn-relu-and-nn-leaky_relu)**:** It is similar to the ReLU function, but it adds a small slope for negative input values to prevent dead neurons.
* [**Tanh (hyperbolic tangent) function**](https://www.geeksforgeeks.org/python-math-tanh-function)**:** It is a non-linear activations function that maps the input's value between -1 to 1. It is similar to the sigmoid function but it provides both positive and negative results. It is mainly used for regression tasks, where the output will be continuous values.

### 17. How do neural networks learn from the data?

In [neural networks](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications), there is a method known as [backpropagation](https://www.geeksforgeeks.org/backpropagation-in-data-mining) is used while training the neural network for adjusting weights and biases of the neural network. It computes the gradient of the cost functions with respect to the parameters of the neural network and then updates the network parameters in the opposite direction of the gradient using optimization algorithms with the aim of minimizing the losses.

During the training, in forward pass the input data passes through the network and generates output. then the cost function compares this generated output to the actual output. then the backpropagation computes the gradient of the cost function with respect to the output of the neural network. This gradient is then propagated backward through the network to compute the gradient of the loss function with respect to the weights and biases of each layer. Here chain rules of differentiations are applied with respect to the parameters of each layer to find the gradient.

Once the gradient is computed, The optimization algorithms are used to update the parameters of the network. Some of the most common optimization algorithms are stochastic gradient descent (SGD), mini-batch, etc.

The goal of the training process is to minimize the cost function by adjusting the weights and biases during the backpropagation.

### 18. How the number of hidden layers and number of neurons per hidden layer are selected?

There is no one-size-fits-all solution to this problem, hence choosing the number of hidden layers and neurons per hidden layer in a neural network is often dependent on practical observations and experimentation. There are, however, a few general principles and heuristics that may be applied as a base.

* The number of hidden layers can be determined by the complexity of the problem being solved. Simple problems can be solved with just one hidden layer whereas more complicated problems may require two or more hidden levels. However adding more layers also increases the risk of overfitting, so the number of layers should be chosen based on the trade-off between model complexity and generalization performance.
* The number of neurons per hidden layer can be determined based on the number of input features and the desired level of model complexity. There is no hard and fast rule, and the number of neurons can be adjusted based on the results of experimentation and validation.

In practice, it is often useful to start with a simple model and gradually increase its complexity until the desired performance is achieved. This process can involve adding more hidden layers or neurons or experimenting with different architectures and hyperparameters. It is also important to regularly monitor the training and validation performance to detect overfitting and adjust the model accordingly.

### 19. What is overfitting and how to avoid it?

Overfitting is a problem in machine learning that occurs when the model learns to fit the training data too close to the point that it starts catching up on noise and unimportant patterns. Because of this, the model performs well on training data but badly on fresh, untested data, resulting in poor generalization performance.

To avoid overfitting in deep learning we can use the following techniques:

1. **Simplify the model:** Overfitting may be less likely in a simpler model with fewer layers and parameters. In practical applications, it is frequently beneficial, to begin with a simple model and progressively increase its complexity until the desired performance is attained.
2. **Regularization:** Regularization is a technique used in machine learning to prevent the overfitting of a model by adding a penalty term, it imposes the constraint on the weight of the model. Some of the most common regularization techniques are as follows:
   1. L1 and L2 regularization: L1 regularization sparse the model by equating many model weights equal to 0 while L2 regularization constrains the weight of the neural network connection.
   2. Dropout: Dropout is a technique that randomly drops out or disables some of the randomly selected neurons. It is applied after the activation functions of the hidden layer. Typically, it is set to a small value like 0.2 or 0.25. For the dropout value of 0.20, Each neuron in the previously hidden layer has a 20% chance of being inactive. It is only operational during the training process.
   3. Max-Norm Regularization: It constrains the magnitude of the weights in a neural network by setting a maximum limit (or norm) on the weights of the neurons, such that their values cannot exceed this limit.
3. **Data augmentation:** By applying various transformations, such as rotating or flipping images, to new training data, it is possible to teach the model to become more robust to changes in the input data.
4. **Increasing the amount of training data:** By increasing the amount of data can provide the model with a diverse set of examples to learn from, which can be helpful to prevent overfitting.
5. **Early stopping:**This involves keeping track of the model's performance on a validation set during training and terminating the training process when the validation loss stops decreasing.

### 20. Define epoch, iterations, and batches.

A complete cycle of deep learning model training utilizing the entire training dataset is called an epoch. Each training sample in the dataset is processed by the model during a single [**epoch**](https://www.geeksforgeeks.org/epoch-in-machine-learning), and its weights and biases are adjusted in response to the estimated loss or error. The number of **epochs** will range from 1 to infinite. User input determines it. It is always an Integral value.

[**Iteration**](https://www.geeksforgeeks.org/epoch-in-machine-learning) refers to the procedure of running a batch of data through the model, figuring out the loss, and changing the model's parameters. Depending on the number of batches in the dataset, one or more iterations can be possible within a single epoch.

A [**batch**](https://www.geeksforgeeks.org/epoch-in-machine-learning) in deep learning is a subset of the training data that is used to modify the weights of a model during training. In batch training, the entire training set is divided into smaller groups, and the model is updated after analyzing each batch.  An epoch can be made up of one or more batches.

* The batch size will be more than one and always less than the number of samples.
* Batch size is a hyperparameter, it is set by the user. where the number of iterations per epoch is calculated by dividing the total number of training samples by the individual batch size.

Deep learning training datasets are often separated into smaller batches, and the model analyses each batch sequentially, one at a time, throughout each epoch.  On the validation dataset, the model performance can be assessed after each epoch. This helps in monitoring the model's progress.

**For example:**Let's use 5000 training samples in the training dataset. Furthermore, we want to divide the dataset into 100 batches. If we choose to use five epochs, the total number of iterations will be as follows:

Total number of training samples = 5000  
Batch size = 100  
Total number of iterations=Total number of training samples/Batch size=5000/100=50  
Total number of iterations = 50  
One epoch = 50 iterations  
Total number of iterations in 5 epochs = 50\*5 = 250 iterations.

### 21. Define the learning rate in Deep Learning.

The learning rate in deep learning is a hyperparameter that controls how frequently the optimizer adjusts the neural network's weights when it is being trained. It determines the step size to which the optimizer frequently updates the model parameters with respect to the loss function. so, that losses can be minimized during training.

With the high learning rate, the model may converge fast, but it may also overshoot or bounce around the ideal solution. On the other hand, a low learning rate might make the model converge slowly, but it could also produce a solution that is more accurate.

Choosing the appropriate learning rate is crucial for the successful training of deep neural networks.

### 22. What is the cross-entropy loss function?

[Cross-entropy](https://www.geeksforgeeks.org/cross-entropy-cost-functions-used-in-classification) is the commonly used loss function in deep learning for classification problems. The cross-entropy loss measures the difference between the real probability distribution and the predicted probability distribution over the classes.

The formula for the Cross-Entropy loss function for the K classes will be:

J(Y,Y^)=−∑kKYklog⁡(Yk^)*J*(*Y*,*Y*^)=−∑*kK*​*Yk*​log(*Yk*​^​)

Here, Y and Y^*Y*^ are actual and predicted values for a single instance. k represents a particular class and is a subset of K.

### 23. What is gradient descent?

Gradient descent is the core of the learning process in machine learning and deep learning. It is the method used to minimize the cost or loss function by iteratively adjusting the model parameters i.e. weight and biases of the neural layer. The objective is to reduce this disparity, which is represented by the cost function as the difference between the model's anticipated output and the actual output.

The gradient is the vector of its partial derivatives with respect to its inputs, which indicates the direction of the steepest ascent (positive gradient) or steepest descent (negative gradient) of the function.

In deep learning, The gradient is the partial derivative of the objective or cost function with respect to its model parameters i.e. weights or biases, and this gradient is used to update the model's parameters in the direction of the negative gradient so that it can reduce the cost function and increase the performance of the model. The magnitude of the update is determined by the learning rate, which controls the step size of the update.

### 24. How do you optimize a Deep Learning model?

A Deep Learning model may be optimized by changing its parameters and hyperparameters to increase its performance on a particular task. Here are a few typical methods for deep learning model optimization:

* Choosing the right architecture
* Adjusting the learning rate
* Regularization
* Data augmentation
* Transfer learning
* Hyperparameter tuning

### 25. Define Batch, Stochastic, and Mini gradient descent.

There are several variants of gradient descent that differ in the way the step size or learning rate is chosen and the way the updates are made. Here are some popular variants:

* **Batch Gradient Descent:** In batch gradient descent, To update the model parameters values like weight and bias, the entire training dataset is used to compute the gradient and update the parameters at each iteration. This can be slow for large datasets but may lead to a more accurate model. It is effective for convex or relatively smooth error manifolds because it moves directly toward an optimal solution by taking a large step in the direction of the negative gradient of the cost function. However, it can be slow for large datasets because it computes the gradient and updates the parameters using the entire training dataset at each iteration. This can result in longer training times and higher computational costs.
* **Stochastic Gradient Descent (SGD):**In SGD, only one training example is used to compute the gradient and update the parameters at each iteration. This can be faster than batch gradient descent but may lead to more noise in the updates.
* **Mini-batch Gradient Descent:**In Mini-batch gradient descent a small batch of training examples is used to compute the gradient and update the parameters at each iteration. This can be a good compromise between batch gradient descent and Stochastic Gradient Descent, as it can be faster than batch gradient descent and less noisy than Stochastic Gradient Descent.

### 26. What are the different types of Neural Networks?

There are different-different [types of neural networks](https://www.geeksforgeeks.org/neural-networks-a-beginners-guide) used in deep learning. Some of the most important neural network architectures are as follows;

1. Feedforward Neural Networks (FFNNs)
2. Convolutional Neural Networks (CNNs)
3. Recurrent Neural Networks (RNNs)
4. Long Short-Term Memory Networks (LSTMs)
5. Gated Recurrent Units (GRU)
6. Autoencoder Neural Networks
7. Attention Mechanism
8. Generative Adversarial Networks (GANs)
9. Transformers
10. Deep Belief Networks (DBNs)

### 27. What is the difference between Shallow Networks and Deep Networks?

Deep networks and shallow networks are two types of artificial neural networks that can learn from data and perform tasks such as classification, regression, clustering, and generation.

* **Shallow networks**: A shallow network has a single hidden layer between the input and output layers, whereas a deep network has several hidden layers. Because they have fewer parameters, they are easier to train and less computationally expensive than deep networks. Shallow networks are appropriate for basic or low-complexity tasks where the input-output relationships are relatively straightforward and do not require extensive feature representation.
* **Deep Networks:** Deep networks, also known as deep neural networks, can be identified by the presence of many hidden layers between the input and output layers. The presence of multiple layers enables deep networks to learn hierarchical data representations, capturing detailed patterns and characteristics at different levels of abstraction. It has a higher capacity for feature extraction and can learn more complex and nuanced relationships in the data. It has given state-of-the-art results in many machine learning and AI tasks.

### 28. What is a Deep Learning framework?

A deep learning framework is a collection of software libraries and tools that provide programmers a better deep learning model development and training possibilities. It offers a high-level interface for creating and training deep neural networks in addition to lower-level abstractions for implementing special functions and topologies. TensorFlow, PyTorch, Keras, Caffe, and MXNet are a few of the well-known frameworks for deep learning.

### 29. What do you mean by vanishing or exploding gradient descent problem?

Deep neural networks experience the vanishing or exploding gradient descent problem when the gradients of the cost function with respect to the parameters of the model either become too small (vanishing) or too big (exploding) during training.

In the case of vanishing gradient descent, The adjustments to the weights and biases made during the backpropagation phase are no longer meaningful because of very small values. As a result, the model could perform poorly because it fails to pick up on key aspects of the data.

In the case of exploding gradient descent, The model surpasses its optimal levels and fails to converge to a reasonable solution because the updates to the weights and biases get too big.

Some of the techniques like Weight initialization, normalization methods, and careful selection of activation functions can be used to deal with these problems.

### 30. What is Gradient Clipping?

Gradient clipping is a technique used to prevent the exploding gradient problem during the training of deep neural networks. It involves rescaling the gradient when its norm exceeds a certain threshold. The idea is to clip the gradient, i.e., set a maximum value for the norm of the gradient, so that it does not become too large during the training process. This technique ensures that the gradients don't become too large and prevent the model from diverging. Gradient clipping is commonly used in recurrent neural networks (RNNs) to prevent the exploding gradient problem.

## *Deep Learning Interview Questions For Experienced*

### 31. What do you mean by momentum optimizations?

[Momentum optimization](https://www.geeksforgeeks.org/ml-momentum-based-gradient-optimizer-introduction) is a method for accelerating the optimization process of a Deep Learning model. It is a modification of the standard gradient descent optimization technique that aids in faster convergence and prevents it from getting stuck in local minima.

In momentum optimization, the update of the model's parameters at each iteration is dependent on both the accumulated gradient from earlier iterations and the current gradient. This accumulated gradient is referred to as the "momentum" because it enables the model to keep travelling in the same direction even when the present gradient is pointing in a different direction.

The amount of the previous gradient that should be integrated into the current update is determined by the momentum term, a hyperparameter. While a low momentum number makes the model more sensitive to changes in gradient direction, a high momentum value indicates that the model will continue to move in the same direction for longer periods of time.

### 32. How weights are initialized in neural networks?

An essential part of training neural networks is [weight initialization](https://www.geeksforgeeks.org/weight-initialization-techniques-for-deep-neural-networks). The objective is to establish the initial weights in such a way that the network may learn efficiently and converge at an appropriate solution. It can be accomplished in several ways:

* **Zero Initialization:** As the name suggests, the initial value of each weight is set to zero during initialization. As a result, all of their derivatives with respect to the loss function are identical, resulting in the same value for each weight in subsequent iterations. The hidden units are also symmetric as a consequence, which may cause training to converge slowly or perhaps prohibit learning altogether.
* **Random Initialization:**The most straightforward approach is to initialize the weights randomly using a uniform or normal distribution. This technique is regularly applied in practice and frequently benefits from shallow networks. However, issues like overfitting, the vanishing gradient problem, and the exploding gradient problem may occur if the weights were assigned values at random.
* **Xavier Initialization:**It sets the initial weights to be drawn from a normal distribution with a mean of zero and a variance of 1/fanavg, where fanavg = (fanin+fanout)/2 is the number of input neurons. This method is commonly used for activation functions like the sigmoid function, softmax function, or tanh function. it is also known as Glorot Initialization.
* **He Initialization:**It is similar to Xavier initialization, but the variance is scaled by a factor of 2/fanavg. This method is used for nonlinear activation functions, such as ReLU and its variants.
* **Orthogonal Initialization:** It initializes the weight matrix to be a random orthogonal matrix. The orthogonal matrix is the square matrix whose columns are orthonormal means dot product or normalized means the column-wise square root of the square of column values is equal to 1. This method has been shown to work well for recurrent neural networks.
* **Pretrained Initialization:**This method initializes the weights based on a pre-trained model on a related task. For example, the weights of a convolutional neural network can be initialized based on a pre-trained model on ImageNet.

### 33. What is fine-tuning in Deep Learning?

Fine-tuning is a technique in deep learning, In which a pre-trained neural network is taken and further customize, to fit a new task by adjusting its weights through further training on a new dataset that is similar to the one that will be used in the final application.

This can be done by replacing the output layer of the pre-trained model with a new layer that is suitable for our problem or freezing some of the layers of the pre-trained model and only training the remaining layers on the new task or dataset. The goal is to modify the pre-trained network's weights by further training in order to adapt it to the new dataset and task.

This procedure enables the network to learn the important characteristics of the new task. The basic objective of fine-tuning is to adapt the pre-trained network to the new job and dataset. This may involve changing the network design or modifying hyperparameters like the learning rate.

### 34. What do you mean by Batch Normalization?

[Batch Normalization](https://www.geeksforgeeks.org/tensorflow-js-tf-layers-batchnormalization-function) is the technique used in deep learning. To prevent the model from vanishing/exploding gradient descent problems It normalizes and scales the inputs before or after the activation functions of each hidden layer. So, the distributions of inputs have zero means and 1 as standard deviation. It computes the mean and standard deviation of each mini-batch input and applies it to normalization so that it is known as batch normalization.

Because the weights of the layer must be changed to adjust for the new distribution, it can be more difficult for the network to learn when the distribution of inputs to a layer changes. This can result in a slower convergence and less precision. By normalizing the inputs to each layer, batch normalization reduces internal covariate shifts. This helps the network to learn more effectively and converge faster by ensuring that the distribution of inputs to each layer stays consistent throughout training.

It prevents vanishing/exploding gradient problems because normalizations of inputs of each layer ensure the gradient is within an appropriate range. It also acts like a regularizer by reducing the need for a regularization technique like a dropout layer.

### 35. What is a dropout in Deep Learning?

[Dropout](https://www.geeksforgeeks.org/dropout-in-neural-networks) is one of the most popular regularization techniques used in deep learning to prevent overfitting. The basic idea behind this is to randomly drop out or set to zero some of the neurons of the previously hidden layer so that its contribution is temporarily removed during the training for both forward and backward passes.

In each iteration, neurons for the dropout are selected randomly and their values are set to zero so that it doesn't affect the downstream neurons of upcoming next-layer neurons during the forward pass, And during the backpropagation, there is no weight update for these randomly selected neurons in current iterations. In this way, a subset of randomly selected neurons is completely ignored during that particular iteration.

This makes the network learn more robust features only and prevents overfitting when the networks are too complex and capture noises during training.

During testing, all the neurons are used and their outputs are scaled or multiplied by the dropout probability to ensure that the overall behaviour of the network is consistent during training.

### 36. What are Convolutional Neural Networks (CNNs)?

[Convolutional Neural Networks (CNNs)](https://www.geeksforgeeks.org/introduction-convolution-neural-network) are the type of neural network commonly used for Computer Vision tasks like image processing, image classification, object detection, and segmentation tasks. It applies filters to the input image to detect patterns, edges, and textures and then uses these features to classify the image.

It is the type of feedforward neural network (FNN) used to extract features from grid-like datasets by applying different types of filters also known as the kernel. For example visual datasets like images or videos where data patterns play an extensive role. It uses the process known as convolution to extract the features from images.

It is composed of multiple layers including the convolution layer, the pooling layer, and the fully connected layer. In the convolutional layers, useful features are extracted from the input data by applying a kernel, The kernel value is adjusted during the training process, and it helps to identify patterns and structures within the input data.

 The pooling layers then reduce the spatial dimensionality of the feature maps, making them more manageable for the subsequent layers. Finally, the fully connected layers use the extracted features to make a prediction or classification.

### 37. What do you mean by convolution?

Convolution is a mathematical operation that is applied in a variety of fields, such as image preprocessing, audio, and signal processing tasks to extract useful features from input data by applying various filters (also known as kernels).

In [CNNs](https://www.geeksforgeeks.org/introduction-convolution-neural-network), It is used to extract the feature from the input dataset. It processes the input images using a set of learnable filters known as kernels. The kernels size are usually smaller like 2×2, 3×3, or 5×5. It computes the dot product between kernel weight and the corresponding input image patch, which comes when sliding over the input image data. The output of this layer is referred ad feature maps.

Convolution is an effective method because it enables CNN to extract local features while keeping the spatial relationships between the features in the input data. This is especially helpful in the processing of images where the location of features within an image is often just as important as the features themselves.

### 38. What is a kernel?

A kernel in [convolutional neural networks (CNNs)](https://www.geeksforgeeks.org/introduction-convolution-neural-network) is a small matrix that is used while performing convolution on the input data. It is also known as a filter or weight. Depending on the size of the input data and the required level of granularity for the extracted features, the kernel shape is chosen. Generally, it is a small matrix like 3x3, 5x5, or 7x7.

In order to extract the most relevant features from the input data, during the training process, the value in the kernel is optimized. When the kernel is applied to the input data, it moves over the data in the form of a sliding window, performing element-wise multiplication at each position and adding the results to create a single output value.

### 39. Define stride.

Stride is the number of pixels or units that a kernel is moved across the input data while performing convolution operations in [Convolutional Neural Networks (CNNs)](https://www.geeksforgeeks.org/introduction-convolution-neural-network). It is one of the hyperparameters of a CNN that can be manipulated to control the output feature map's size.

During the forward pass, we slide each filter through the entire input image matrix step by step, where each step is known as stride (which can have a value of 2, 3, or even 4 for high-dimensional images), and we compute the dot product between the kernel weights and patch from input volume.

### 40. What is the pooling layer?

The pooling layer is a type of layer that usually comes after one or more convolutional layers in [convolutional neural networks (CNNs)](https://www.geeksforgeeks.org/introduction-convolution-neural-network). The primary objective of the pooling layer is to reduce the spatial dimensionality of the feature maps while maintaining the most crucial characteristics produced with the convolution operations. Its main function is to reduce the size of the spatial dimensionality which makes the computation fast reduces memory and also prevents overfitting. It also helps to make the features more invariant to small translations in the input data, which can improve the model's robustness to changes in the input data.

Two common types of pooling layers are max pooling and average pooling.  In max pooling, the maximum value within each subregion is selected and propagated to the output feature map. In average pooling, the average value within each subregion is calculated and used as the output value.

### 41. Define the same and valid padding.

Padding is a technique used in [convolutional neural networks](https://www.geeksforgeeks.org/introduction-convolution-neural-network) to preserve the spatial dimensions of the input data and prevent the loss of information at the edges of the image. it is done by adding additional layers of zeros around the edges of the input matrix.

There are two main types of padding: same padding and valid padding.

* **Same Padding:**The term "same padding" describes the process of adding padding to an image or feature map such that the output has the same spatial dimensions as the input. The same padding adds additional rows and columns of pixels around the edges of the input data so that the size of the output feature map will be the same as the size of the input data. This is achieved by adding rows and columns of pixels with a value of zero around the edges of the input data before the convolution operation.
* **Valid Padding:** Convolutional neural networks (CNNs) employ the valid padding approach to analyze the input data without adding any extra rows or columns of pixels around the input data's edges. This means that the size of the output feature map is smaller than the size of the input data. Valid padding is used when it is desired to reduce the size of the output feature map in order to reduce the number of parameters in the model and improve its computational efficiency.

### 42. Write the formula for finding the output shape of the Convolutional Neural Networks model.

The formula for calculating the output shape for the same padding

Outputw=(Iw+2p)−Kwsw+1Outputh=(Ih+2p)−Khsh+1*Outputw*​=*sw*​(*Iw*​+2*p*)−*Kw*​​+1*Outputh*​=*sh*​(*Ih*​+2*p*)−*Kh*​​+1

Where,

                  Iw,Ih       *Iw*​,*Ih*​       = Number of rows and columns of the input image.

                   Kw,Kh       *Kw*​,*Kh*​        = Filter kernel dimesions.

                  sw,sh       *sw*​,*sh*​        = Strides

                  p = Number of layer of zeros.

   For the same output shape: stride = 1 and

(Iw+2p)−Kwsw+1=Iw(Iw+2p)−Kw+1=Iwp=Kw−12*sw*​(*Iw*​+2*p*)−*Kw*​​+1=*Iw*​(*Iw*​+2*p*)−*Kw*​+1=*Iw*​*p*=2*Kw*​−1​

**Example:**

Let's an image of dimension Iw×Ih×c  =28 ×28×3    *Iw*​×*Ih*​×*c*  =28 ×28×3      is having filtered with kernel  Kw×Kh =3×3    *Kw*​×*Kh*​ =3×3      dimension kernel with stride (sw,sh)=(3,3)       (*sw*​,*sh*​)=(3,3)        then Let's calculate the output shape with the formula :

Ow=(Iw+2p)−Kwsw+1 =(28+2×1)−31+1=28Oh=(Ih+2p)−Khsh+1=(28+2×1)−31+1=28*Ow*​​=*sw*​(*Iw*​+2*p*)−*Kw*​​+1 =1(28+2×1)−3​+1=28​*Oh*​​=*sh*​(*Ih*​+2*p*)−*Kh*​​+1=1(28+2×1)−3​+1=28​

### 43. What is the data augmentation technique in CNNs?

[Data augmentation](https://www.geeksforgeeks.org/python-data-augmentation) is a technique used in deep learning during the preprocessing for making little variation in the training dataset, So, that model can improve its generalization ability with a greater variety of data changes. It is also used to increase the training dataset samples by creating a modified version of the original dataset.

In CNNs, data augmentation is often carried out by randomly applying a series of image transformations to the initial training images. that are as follows:

* Rotation
* Scaling
* Flipping
* Cropping
* Sharing
* Translation
* Adding noise
* Changing brightness or contrast

### 44. What do you mean by deconvolution?

Deconvolution is a deep learning method for upscale feature maps in a [convolutional neural network (CNN)](https://www.geeksforgeeks.org/introduction-convolution-neural-network). During the convolution, Kernel slides over the input to extract the important features and shrink the output, while in deconvolution, the kernel slides over the output to generate a larger, more detailed output. Briefly, we can say that deconvolution is the opposite of convolution operations.

Deconvolution may be used for a variety of applications, including object identification, image segmentation, and image super-resolution. For example, in image super-resolution, a CNN is used to extract features from a low-resolution input image, and the feature map is deconvolved to generate a higher-resolution output image.

### 45. What is the difference between object detection and image segmentation?

[Object detection and image segmentation](https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation) are both computer vision tasks used to analyze and understand images, but they differ in their goals and output.

The difference between object detection and image segmentation is as follows:

| **Object Detection** | **Image Segmentation** |
| --- | --- |
| Object detection is used to identify and locate the specific objects within the image or video. | Image Segmentation divides the digital image into multiple image segments or regions (i.e. typically a set of pixels), where each of the image segments belongs to different objects or the parts of image. |
| The main goal of object detection involves detecting the presence of the object within an image, and It typically draws a bounding box around the objects and mentioned the label or name of the objects. | In image segmentations, pixel-wise classification happens, which means the goal in image segmentations is to assign a label for each pixel that which segment it belongs. |
| It is concerned with identifying and localizing specific objects within an image. | It is concerned with pixel-wise segmentation within an image into different regions and assigning labels to each pixel. |
| It uses a combination of feature extraction and classification algorithms, such as Convolutional Neural Networks (CNNs) and object detection frameworks like Faster R-CNN or YOLO. | it uses techniques such as clustering, edge detection, region growing, or CNN-based segmentation methods like U-Net or Mask R-CNN. |
| It is primarily used in surveillance, self-driving car and robotics, etc | It is used in a variety of tasks like object recognition, image editing, scene understanding, and computer graphics. |

### 46. What are Recurrent Neural Networks (RNNs) and How it works?

[Recurrent Neural Networks](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network) are the type of [artificial neural network](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications) that is specifically designed to work with sequential data or time series data. It is specifically used in natural language processing tasks like language translation, speech recognition, sentiment analysis, natural language generation, summary writing, etc. It is different from the feedforward neural networks means in RNN the input data not only flow in a single direction but it also has a loop or cycle within its architecture which has the  "memory" that preserve the information over time. This makes the RNN capable of data where context is important like the natural languages.

The basic concept of RNNs is that they analyze input sequences one element at a time while maintaining track in a hidden state that contains a summary of the sequence's previous elements. The hidden state is updated at each time step based on the current input and the previous hidden state.  This allows RNNs to capture the temporal dependencies between elements of the sequence and use that information to make predictions.

**Working**: The fundamental component of an RNN is the recurrent neuron, which receives as inputs the current input vector and the previous hidden state and generates a new hidden state as output. And this output hidden state is then used as the input for the next recurrent neuron in the sequence. An RNN can be expressed mathematically as a sequence of equations that update the hidden state at each time step:

ht= f(Uht-1+Wxt+b)

Where,

* ht = Current state at time t
* xt = Input vector at time t
* ht-1 = Previous state at time t-1
* U = Weight matrix of recurrent neuron for the previous state
* W = Weight matrix of input neuron
* b = Bias added to the input vector and previous hidden state
* f = Activation functions

And the output of the RNN at each time step will be:

yt = g(Vht+c)

Where,

* y = Output at time t
* V = Weight matrix for the current state in the output layer
* C = Bias for the output transformations.
* g = activation function

Here, W, U, V, b, and c are the learnable parameters and it is optimized during the backpropagation.

### 47. How does the Backpropagation through time work in RNN?

Backpropagation through time (BPTT) is a technique for updating the weights of a recurrent neural network (RNN) over time by applying the backpropagation algorithm to the unfolded network. It enables the network to learn from the data's temporal dependencies and adapt its behaviour accordingly. Forward Pass: The input sequence is fed into the RNN one element at a time, starting from the first element. Each input element is processed through the recurrent connections, and the hidden state of the RNN is updated.

1. Given a sequence of inputs and outputs, the RNN is unrolled into a feed-forward network with one layer per time step.
2. The network of the RNN is initialized with some initial hidden state that contains information about the previous inputs and hidden states in the sequence. It computes the outputs and the hidden states for each time step by applying the recurrent function.
3. The network computes the difference between the predicted and expected outputs for each time step and adds it up across the entire series.
4. The gradients of the error with respect to the weights are calculated by the network by applying the chain rule from the last time step to the first time step, propagating the error backwards through time. The loss is then backpropagated through time, starting from the last time step and moving backwards in time. So, this is known as Backpropagation through time (BPTT).
5. The network's weights are updated using an optimization algorithm, such as gradient descent or its variants, which takes gradients and a learning rate into account.
6. Repeat: The process is repeated for a specified number of epochs or until convergence, during this the training data is iterated through several times.

During the backpropagation process, the gradients at each time step are obtained and used to update the weights of the recurrent networks. The accumulation of gradients over multiple time steps enables the RNN to learn and capture dependencies and patterns in sequential data.

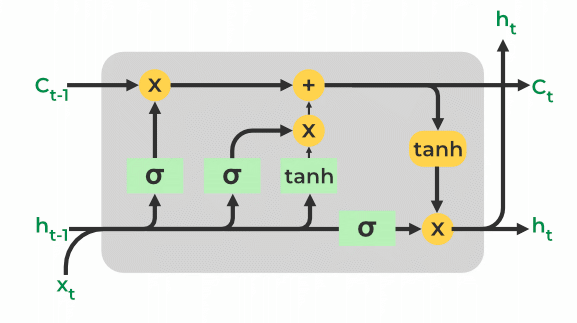
### 48. What is LSTM, and How it works?

LSTM stands for [Long Short-Term Memory](https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory). It is the modified version of RNN (Recurrent Neural Network) that is designed to address the vanishing and exploding gradient problems that can occur during the training of traditional RNNs. LSTM selectively remembers and forgets information over the multiple time step which gives it a great edge in capturing the long-term dependencies of the input sequence.

RNN has a single hidden state that passes through time, which makes it difficult for the network to learn long-term dependencies. To address this issue LSTM uses a memory cell, which is a container that holds information for an extended period of time. This memory cell is controlled by three gates i.e. input gate, forget gate, and the output gate. These gates regulate which information should be added, removed, or output from the memory cell.

LSTMs function by selectively passing or retaining information from one-time step to the next using the combination of memory cells and gating mechanisms. The LSTM cell is made up of a number of parts, such as:

* **Cell state (C):** This is where the data from the previous step is kept in the LSTM's memory component.  It is passed through the LSTM cell via gates that control the flow of information into and out of the cell.
* **Hidden state (h):** This is the output of the LSTM cell, which is a transformed version of the cell state.  It can be used to make predictions or be passed on to another LSTM cell later on in the sequence.
* **Forget gate (f):** The forget gate removes the data that is no longer relevant in the cell state. The gate receives two inputs, xt (input at the current time) and ht-1 (previous hidden state), which are multiplied with weight matrices, and bias is added. The result is passed via an activation function, which gives a binary output i.e. True or False.
* **Input Gate(i):** The input gate uses as input the current input and the previous hidden state and applies a sigmoid activation function to determine which parts of the input should be added to the cell state. The output of the input gate (again a fraction between 0 and 1) is multiplied by the output of the tanh block that produces the new values that are added to the cell state. This gated vector is then added to the previous cell state to generate the current cell state
* **Output Gate(o):** The output gate extracts the important information from the current cell state and delivers it as output.  First, The tanh function is used in the cell to create a vector. Then, the information is regulated using the sigmoid function and filtered by the values to be remembered using inputs ht-1 and xt. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.



LSTM Model architecture

### 49. What is GRU? and How it works?

Ans: GRU stands for Gated Recurrent Unit. GRUs are recurrent neural networks (RNNs) that can process sequential data such as text, audio, or time series.GRU uses gating mechanisms to control the flow of information in and out of the network, allowing it to learn from the temporal dependencies in the data and adjust its behaviour accordingly.

GRU is similar to LSTM in that it uses gating mechanisms, but it has a simpler architecture with fewer gates, making it computationally more efficient and easier to train. It uses two types of Gates: the reset gate (r) and the update gate (z)

1. Rest Gate (r): It determines which parts of the previous hidden state should be forgotten or reset. It takes The update gate decides which parts of the current hidden state should be updated with new information from the current input. Similar to the reset gate, it takes the previous hidden state and the current input as inputs and outputs a value between 0 and 1 for each element of the hidden state.  
   Hidden State: ht=(1−zt)⋅ht−1+zt⋅h^tReset Gate: rt=σ(Wr⋅[ht−1,xt])Hidden State: *ht*​=(1−*zt*​)⋅*ht*−1​+*zt*​⋅*h*^*t*​Reset Gate: *rt*​=*σ*(*Wr*​⋅[*ht*−1​,*xt*​])
2. Update Gate (z): It decides which part of the current hidden state should be updated with the new information from the current input. It takes the previous hidden state and the current input as inputs and the outputs value between 0 and 1 for each element of the hidden state.  
   Update Gate: zt=σ(Wz⋅[ht−1,xt])Current Hidden State: h^t=tanh⁡(Wh⋅[rt⋅ht−1,xt])Update Gate: *zt*​=*σ*(*Wz*​⋅[*ht*−1​,*xt*​])Current Hidden State: *h*^*t*​=tanh(*Wh*​⋅[*rt*​⋅*ht*−1​,*xt*​])

GRU models have been demonstrated to be useful in NLP applications such as language modelling, sentiment analysis, machine translation, and text generation. They are especially beneficial when it is critical to record long-term dependencies and grasp the context. GRU is a popular choice in NLP research and applications due to its simplicity and computational efficiency.

### 50. What is an Encoder-Decoder network in Deep Learning?

An encoder-decoder network is a kind of neural network that can learn to map an input sequence to a different length and structure output sequence. It is made up of two primary parts: an encoder and a decoder.

* Encoder: The encoder takes a variable-length input sequence (such as a sentence, an image, or a video) and processes it step by step steps to build a fixed-length context or encoded vector or representation that captures the important information from the input sequence. The encoded vector condenses the information from the entire input sequence.
* Decoder: Decoder is another neural network that takes the encoded vector as input and generates an output sequence (such as another sentence, an image, or a video) that is related to the input sequence. The decoder generates an output and modifies its internal hidden state based on the encoded vector and previously generated outputs at each step.

The training process of an Encoder-Decoder network involves feeding pairs of input and target sequences to the model and minimizing the difference between the predicted output sequence and the true target sequence using a suitable loss function. Encoder-Decoder networks are used for a variety of tasks, such as machine translation (translating text from one language to another), text summarization, chatbots, and image captioning (turning pictures into meaningful phrases).

### 51. What is an autoencoder?

[Autoencoders](https://www.geeksforgeeks.org/ml-auto-encoders) are a type of neural network architecture used for unsupervised learning tasks like dimensionality reduction, feature learning, etc. Autoencoders work on the principle of learning a low-dimensional representation of high-dimensional input data by compressing it into a latent representation and then reconstructing the input data from the compressed representation. It consists of two main parts an encoder and a decoder.  The encoder maps an input to a lower-dimensional latent representation, while the decoder maps the latent representation back to the original input space. In most cases, neural networks are used to create the encoder and decoder, and they are trained in parallel to reduce the difference between the original input data and the reconstructed data.

### 52. What is a Generative Adversarial Network (GAN)?

[Generative Adversarial Networks (GANs)](https://www.geeksforgeeks.org/generative-adversarial-network-gan) are a type of neural network architecture used for unsupervised learning tasks like image synthesis and generative modeling. It is composed of two neural networks: Generator and Discriminator. The generator takes the random distributions mainly the Gaussian distribution as inputs and generates the synthetic data, while the discriminator takes both real and synthetic data as input and predicts whether the input is real or synthetic. The goal of the generator is to generate synthetic data that is identical to the input data. and the discriminator guesses whether the input data is real or synthetic.

### 53. What is the attention mechanism?

An attention mechanism is a type of neural network that employs a separate attention layer within an Encoder-Decoder neural network to allow the model to focus on certain areas of the input while executing a task. It accomplishes this by dynamically assigning weights to various input components, reflecting their relative value or relevance. This selective attention enables the model to concentrate on key information, capture dependencies, and understand data linkages.

The attention mechanism is especially useful for tasks that need sequential or structured data, such as natural language processing, where long-term dependencies and contextual information are critical for optimal performance. It allows the model to selectively attend the important features or contexts, which increases the model's capacity to manage complicated linkages and dependencies in the data, resulting in greater overall performance in various tasks.

### 54. What is the Transformer model?

Transformer is an important model in neural networks that relies on the attention mechanism, allowing it to capture long-range dependencies in sequences more efficiently than typical RNNs. It has given state-of-the-art results in various NLP tasks like word embedding, machine translation, text summarization, question answering etc.

The key components of the Transformer model are as follows:

* Self-Attention Mechanism: A [self-attention mechanism](https://www.geeksforgeeks.org/self-attention-in-nlp) is a powerful tool that allows the Transformer model to capture long-range dependencies in sequences. It allows each word in the input sequence to attend to all other words in the same sequence, and the model learns to assign weights to each word based on its relevance to the others. This enables the model to capture both short-term and long-term dependencies, which is critical for many NLP applications.
* Encoder-Decoder Network: An encoder-decoder architecture is used in the Transformer model. The encoder analyzes the input sequence and creates a context vector that contains information from the entire sequence. The context vector is then used by the decoder to construct the output sequence step by step.
* Multi-head Attention: The purpose of the multi-head attention mechanism in Transformers is to allow the model to recognize different types of correlations and patterns in the input sequence. In both the encoder and decoder, the Transformer model uses multiple attention heads. This enables the model to recognise different types of correlations and patterns in the input sequence. Each attention head learns to pay attention to different parts of the input, allowing the model to capture a wide range of characteristics and dependencies.
* Positional Encoding: Positional encoding is applied to the input embeddings to offer this positional information like the relative or absolute position of each word in the sequence to the model. These encodings are typically learnt and can take several forms, including sine and cosine functions or learned embeddings. This enables the model to learn the order of the words in the sequence, which is critical for many NLP tasks.
* Feed-Forward Neural Networks: Following the attention layers, the model applies a point-wise feed-forward neural network to each position separately. This enables the model to learn complex non-linear correlations in the data.
* Layer Normalization and Residual Connections: Layer normalization is used to normalize the activations at each layer of the Transformer, promoting faster convergence during training. Furthermore, residual connections are used to carry the original input directly to successive layers, assisting in mitigating the vanishing gradient problem and facilitating gradient flow during training.

### 55. What is Transfer Learning?

[Transfer learning](https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning) is a machine learning approach that involves implementing the knowledge and understanding gained by training a model on one task and applying that knowledge to another related task. The basic idea behind transfer learning is that a model that has been trained on a big, diverse dataset may learn broad characteristics that are helpful for many different tasks and can then be modified or fine-tuned to perform a specific task with a smaller, more specific dataset.

Transfer learning can be applied in the following ways:

1. **Fine-tuning:** Fine-tuning is used to adapt a pre-trained model that has already been trained on a big dataset and refine it with further training on a new smaller dataset that is specific to the present task. With fine-tuning, weights of the pre-trained model can be adjusted according to the new present task while training on the new dataset. This can improve the performance of the model on the new task.
2. **Feature extraction:** In this case, the features of the pre-trained model are extracted, and these extracted features can be used as the input for the new model. This can be useful when the new task involves a different input format than the original task.
3. **Domain adaptation:** In this case, A pre-trained model is adapted from a source domain to a target domain by modifying its architecture or training process to better fit the target domain.
4. **Multi-task learning:**By simultaneously training a single network on several tasks, this method enables the network to pick up common representations that are applicable to all tasks.
5. **One-shot learning:** This involves applying information gained from previous tasks to train a model on just one or a small number of samples of a new problem.

### 56. What are distributed and parallel training in deep learning?

Deep learning techniques like distributed and parallel training are used to accelerate the training process of bigger models. Through the use of multiple computing resources, including CPUs, GPUs, or even multiple machines, these techniques distribute the training process in order to speed up training and improve scalability.

When storing a complete dataset or model on a single machine is not feasible, multiple machines must be used to store the data or model. When the model is split across multiple machines, then it is known as model parallelism. In model parallelism, different parts of the model are assigned to different devices or machines. Each device or machine is responsible for computing the forward and backward passes for the part of the model assigned to it.

When the data is too big that it is distributed across multiple machines, it is known as **data parallelism**. Distributed training is used to simultaneously train the model on multiple devices, each of which processes a separate portion of the data. In order to update the model parameters, the results are combined, which speed-up convergence and improve the performance of the model.

Parallel training, involves training multiple instances of the same model on different devices or machines. Each instance trains on a different subset of the data and the results are combined periodically to update the model parameters. This technique can be particularly useful for training very large models or dealing with very large datasets.

Both parallel and distributed training need specialized hardware and software configurations, and performance may benefit from careful optimization. However, they may significantly cut down on the amount of time needed to train deep neural networks.

1. **Data Analysis Interview Questions**

[**https://www.geeksforgeeks.org/data-analyst-interview-questions-and-answers/**](https://www.geeksforgeeks.org/data-analyst-interview-questions-and-answers/)

# ( Top 80+ Data Analyst Interview Questions and Answers

Last Updated : 21 Apr, 2025 )

## *Data Analyst Interview Questions for Freshers*

### 1. What do you mean by Data Analysis?

[Data analysis](https://www.geeksforgeeks.org/data-analysis-with-python) is a multidisciplinary field of data science, in which data is analyzed using mathematical, statistical, and computer science with domain expertise to discover useful information or patterns from the data. It involves gathering, cleaning, transforming, and organizing data to draw conclusions, forecast, and make informed decisions. The purpose of data analysis is to turn raw data into actionable knowledge that may be used to guide decisions, solve issues, or reveal hidden trends.

### 2. How do data analysts differ from data scientists?

[Data analysts](https://www.geeksforgeeks.org/how-to-become-a-data-analyst-complete-roadmap) and [Data Scientists](https://www.geeksforgeeks.org/how-to-become-data-scientist-a-complete-roadmap) can be recognized by their responsibilities, skill sets, and areas of expertise. Sometimes the roles of data analysts and data scientists may conflict or not be clear.

Data analysts are responsible for collecting, cleaning, and analyzing data to help businesses make better decisions. They typically use statistical analysis and visualization tools to identify trends and patterns in data. Data analysts may also develop reports and dashboards to communicate their findings to stakeholders.

Data scientists are responsible for creating and implementing machine learning and statistical models on data. These models are used to make predictions, automate jobs, and enhance business processes. Data scientists are also well-versed in programming languages and software engineering.

| **Feature** | **Data analyst** | **Data Scientist** |
| --- | --- | --- |
| **Skills** | Excel, SQL, Python, R, Tableau, PowerBI | Machine Learning, Statistical Modeling, Docker, Software Engineering |
| **Tasks** | Data Collection, Web Scrapping, Data Cleaning, Data Visualization, Explanatory Data Analysis, Reports Development and Presentations | Database Management, Predictive Analysis and prescriptive analysis, Machine Learning model building and Deployment, Task automation, Work for Business Improvements Process. |
| **Positions** | Entry Label | Seniors Label |

### 3. How Data analysis is similar to Business Intelligence?

Data analysis and [Business intelligence](https://www.geeksforgeeks.org/what-is-business-intelligence) are both closely related fields, Both use data and make analysis to make better and more effective decisions. However, there are some key differences between the two.

* **Data analysis** involves data gathering, inspecting, cleaning, transforming and finding relevant information, So, that it can be used for the decision-making process.
* **Business Intelligence(BI)** also makes data analysis to find insights as per the business requirements. It generally uses statistical and Data visualization tools popularly known as BI tools to present the data in user-friendly views like reports, dashboards, charts and graphs.

The similarities and differences between the Data Analysis and Business Intelligence are as follows:

| **Similarities** | **Differences** |
| --- | --- |
| Both use data to make better decisions. | Data analysis is more technical, while BI is more strategic. |
| Both involve collecting, cleaning, and transforming data. | Data analysis focuses on finding patterns and insights in data, while BI focuses on providing relevant information |
| Both use visualization tools to communicate findings. | Data analysis is often used to provide specific answers, whereas business intelligence (BI) is used to help broader decision-making. |

### 4. What are the different tools mainly used for data analysis?

There are different tools used for data analysis. each has some strengths and weaknesses. Some of the most commonly used tools for data analysis are as follows:

* [**Spreadsheet Software**](https://www.geeksforgeeks.org/introduction-to-excel-spreadsheet)**:** Spreadsheet Software is used for a variety of data analysis tasks, such as sorting, filtering, and summarizing data. It also has several built-in functions for performing statistical analysis. The top 3 mostly used Spreadsheet Software are as follows:
  + [Microsoft Excel](https://www.geeksforgeeks.org/excel-tutorial)
  + Google Sheets
  + LibreOffice Calc
* [**Database Management Systems (DBMS):**](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1) DBMSs, or database management systems, are crucial resources for data analysis. It offers a secure and efficient way to manage, store, and organize massive amounts of data.
  + MySQL
  + PostgreSQL
  + Microsoft SQL Server
  + Oracle Database
* [**Statistical Software**](https://www.geeksforgeeks.org/sas-vs-spss)**:**There are many statistical software used for Data analysis, Each with its strengths and weaknesses. Some of the most popular software used for data analysis are as follows:
  + **SAS**: Widely used in various industries for statistical analysis and data management.
  + **SPSS**: A software suite used for statistical analysis in social science research.
  + **Stata**: A tool commonly used for managing, analyzing, and graphing data in various fields.SPSS:
* [**Programming Language**](https://www.geeksforgeeks.org/introduction-to-programming-languages)**:**In data analysis, programming languages are used for deep and customized analysis according to mathematical and statistical concepts. For Data analysis, two programming languages are highly popular:
  + [**R**](https://www.geeksforgeeks.org/r-programming-language-introduction)**:** R is a free and open-source programming language widely popular for data analysis. It has good visualizations and environments mainly designed for statistical analysis and data visualization. It has a wide variety of packages for performing different data analysis tasks.
  + [**Python**](https://www.geeksforgeeks.org/python-programming-language): Python is also a free and open-source programming language used for Data analysis. Nowadays, It is becoming widely popular among researchers. Along with data analysis, It is used for Machine Learning, Artificial Intelligence, and web development.

### 5. What is Data Wrangling?

[Data Wrangling](https://www.geeksforgeeks.org/data-wrangling-in-python) is very much related concepts to [Data Preprocessing](https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python). It's also known as Data munging. It involves the process of cleaning, transforming, and organizing the raw, messy or unstructured data into a usable format. The main goal of data wrangling is to improve the quality and structure of the dataset. So, that it can be used for analysis, model building, and other data-driven tasks.

Data wrangling can be a complicated and time-consuming process, but it is critical for businesses that want to make data-driven choices. Businesses can obtain significant insights about their products, services, and bottom line by taking the effort to wrangle their data.

Some of the most common tasks involved in data wrangling are as follows:

* [**Data Cleaning**](https://www.geeksforgeeks.org/data-cleansing-introduction)**:**Identify and remove the errors, inconsistencies, and missing values from the dataset.
* [**Data Transformation**](https://www.geeksforgeeks.org/data-transformation-in-data-mining)**:** Transformed the structure, format, or values of data as per the requirements of the analysis. that may include scaling & normalization, encoding categorical values.
* [**Data Integration**](https://www.geeksforgeeks.org/data-integration-in-data-mining)**:**Combined two or more datasets, if that is scattered from multiple sources, and need of consolidated analysis.
* **Data Restructuring:** Reorganize the data to make it more suitable for analysis. In this case, data are reshaped to different formats or new variables are created by aggregating the features at different levels.
* **Data Enrichment:** Data are enriched by adding additional relevant information, this may be external data or combined aggregation of two or more features.
* **Quality Assurance:** In this case, we ensure that the data meets certain quality standards and is fit for analysis.

### 6. What is the difference between descriptive and predictive analysis?

[Descriptive](https://www.geeksforgeeks.org/descriptive-statistic) and [predictive](https://www.geeksforgeeks.org/step-by-step-predictive-analysis-machine-learning) analysis are the two different ways to analyze the data.

* **Descriptive Analysis:**Descriptive analysis is used to describe questions like "What has happened in the past?" and "What are the key characteristics of the data?". Its main goal is to identify the patterns, trends, and relationships within the data. It uses statistical measures, visualizations, and exploratory data analysis techniques to gain insight into the dataset.  
  The key characteristics of descriptive analysis are as follows:
  + **Historical Perspective**: Descriptive analysis is concerned with understanding past data and events.
  + **Summary Statistics**: It often involves calculating basic statistical measures like mean, median, mode, standard deviation, and percentiles.
  + **Visualizations**: Graphs, charts, histograms, and other visual representations are used to illustrate data patterns.
  + Patterns and Trends: Descriptive analysis helps identify recurring patterns and trends within the data.
  + Exploration: It's used for initial data exploration and hypothesis generation.
* **Predictive Analysis:** Predictive Analysis, on the other hand, uses past data and applies statistical and machine learning models to identify patterns and relationships and make predictions about future events. Its primary goal is to predict or forecast what is likely to happen in future.  
  The key characteristics of predictive analysis are as follows:
  + **Future Projection**: Predictive analysis is used to forecast and predict future events.
  + **Model Building**: It involves developing and training models using historical data to predict outcomes.
  + **Validation and Testing**: Predictive models are validated and tested using unseen data to assess their accuracy.
  + **Feature Selection**: Identifying relevant features (variables) that influence the predicted outcome is crucial.
  + **Decision Making**: Predictive analysis supports decision-making by providing insights into potential outcomes.

### 7. What is univariate, bivariate, and multivariate analysis?

Univariate, Bivariate and multivariate are the three different levels of data analysis that are used to understand the data.

1. [**Univariate analysis**](https://www.geeksforgeeks.org/univariate-bivariate-and-multivariate-data-and-its-analysis)**:** Univariate analysis analyzes one variable at a time. Its main purpose is to understand the distribution, measures of central tendency (mean, median, and mode), measures of dispersion (range, variance, and standard deviation), and graphical methods such as histograms and box plots. It does not deal with the courses or relationships from the other variables of the dataset.   
   Common techniques used in univariate analysis include histograms, bar charts, pie charts, box plots, and summary statistics.
2. [**Bivariate analysis**](https://www.geeksforgeeks.org/what-is-univariate-bivariate-multivariate-analysis-in-data-visualisation)**:** Bivariate analysis involves the analysis of the relationship between the two variables. Its primary goal is to understand how one variable is related to the other variables. It reveals, Are there any correlations between the two variables, if yes then how strong the correlations is? It can also be used to predict the value of one variable from the value of another variable based on the found relationship between the two.  
   Common techniques used in bivariate analysis include scatter plots, correlation analysis, contingency tables, and cross-tabulations.
3. [**Multivariate analysis**](https://www.geeksforgeeks.org/what-is-univariate-bivariate-multivariate-analysis-in-data-visualisation)**:**Multivariate analysis is used to analyze the relationship between three or more variables simultaneously. Its primary goal is to understand the relationship among the multiple variables. It is used to identify the patterns, clusters, and dependencies among the several variables.  
   Common techniques used in multivariate analysis include principal component analysis (PCA), factor analysis, cluster analysis, and regression analysis involving multiple predictor variables.

### 8. Name some of the most popular data analysis and visualization tools used for data analysis.

Some of the most popular data analysis and visualization tools are as follows:

* [**Tableau:**](https://www.geeksforgeeks.org/tableau-tutorial) Tableau is a powerful data visualization application that enables users to generate interactive dashboards and visualizations from a wide range of data sources. It is a popular choice for businesses of all sizes since it is simple to use and can be adjusted to match any organization's demands.
* [**Power BI**](https://www.geeksforgeeks.org/power-bi-tutorial): Microsoft's Power BI is another well-known data visualization tool. Power BI's versatility and connectivity with other Microsoft products make it a popular data analysis and visualization tool in both individual and enterprise contexts.
* [Qlik Sense](https://www.geeksforgeeks.org/power-bi-vs-qlik-sense): Qlik Sense is a data visualization tool that is well-known for its speed and performance. It enables users to generate interactive dashboards and visualizations from several data sources, and it can be used to examine enormous datasets.
* [**SAS**](https://www.geeksforgeeks.org/introduction-to-sas-programming): A software suite used for advanced analytics, multivariate analysis, and business intelligence.
* [IBM SPSS](https://www.geeksforgeeks.org/spss-vs-stata): A statistical software for data analysis and reporting.
* Google Data Studio: Google Data Studio is a free web-based data visualization application that allows users to create customized dashboards and simple reports. It aggregates data from up to 12 different sources, including Google Analytics, into an easy-to-modify, easy-to-share, and easy-to-read report.

### 9. What are the steps you would take to analyze a dataset?

Data analysis involves a series of steps that transform raw data into relevant insights, conclusions, and actionable suggestions. While the specific approach will vary based on the context and aims of the study, here is an approximate outline of the processes commonly followed in data analysis:

* **Problem Definition or Objective:** Make sure that the problem or question you're attempting to answer is stated clearly. Understand the analysis's aims and objectives to direct your strategy.
* **Data Collection:** Collate relevant data from various sources. This might include surveys, tests, databases, web scraping,  and other techniques. Make sure the data is representative and accurate.ALso
* [**Data Preprocessing or Data Cleaning**](https://www.geeksforgeeks.org/difference-between-data-cleaning-and-data-processing)**:** Raw data often has errors, missing values, and inconsistencies. In Data Preprocessing and Cleaning, we redefine the column's names or values, standardize the formats, and deal with the missing values.
* [**Exploratory Data Analysis (EDA)**](https://www.geeksforgeeks.org/what-is-exploratory-data-analysis)**:** EDA is a crucial step in Data analysis. In EDA, we apply various graphical and statistical approaches to systematically analyze and summarize the main characteristics, patterns, and relationships within a dataset. The primary objective behind the EDA is to get a better knowledge of the data's structure, identify probable abnormalities or outliers, and offer initial insights that can guide further analysis.
* [**Data Visualizations**](https://www.geeksforgeeks.org/what-is-data-visualization-and-why-is-it-important)**:** Data visualizations play a very important role in data analysis. It provides visual representation of complicated information and patterns in the data which enhances the understanding of data and helps in identifying the trends or patterns within a data. It enables effective communication of insights to various stakeholders.

### 10. What is data cleaning?

[Data cleaning](https://www.geeksforgeeks.org/data-cleansing-introduction)is the process of identifying the removing misleading or inaccurate records from the datasets. The primary objective of Data cleaning is to improve the quality of the data so that it can be used for analysis and predictive model-building tasks. It is the next process after the data collection and loading.

In Data cleaning, we fix a range of issues that are as follows:

1. **Inconsistencies**: Sometimes data stored are inconsistent due to variations in formats, columns\_name, data types, or values naming conventions. Which creates difficulties while aggregating and comparing. Before going for further analysis, we correct all these inconsistencies and formatting issues.
2. **Duplicate entries:** Duplicate records may biased analysis results, resulting in exaggerated counts or incorrect statistical summaries. So, we also remove it.
3. **Missing Values:** Some data points may be missing. Before going further either we remove the entire rows or columns or we fill the missing values with probable items.
4. **Outlier**: Outliers are data points that drastically differ from the average which may result in machine error when collecting the dataset. if it is not handled properly, it can bias results even though it can offer useful insights. So, we first detect the outlier and then remove it.

### 11. What is the importance of exploratory data analysis (EDA) in data analysis?

[Exploratory data analysis (EDA)](https://www.geeksforgeeks.org/what-is-exploratory-data-analysis) is the process of investigating and understanding the data through graphical and statistical techniques. It is one of the crucial parts of data analysis that helps to identify the patterns and trends in the data as well as help in understanding the relationship between variables.

EDA is a non-parametric approach in data analysis, which means it does take any assumptions about the dataset. EDA is important for a number of reasons that are as follows:

1. With EDA we can get a deep understanding of patterns, distributions, nature of data and relationship with another variable in the dataset.
2. With EDA we can analyze the quality of the dataset by making univariate analyses like the mean, median, mode, quartile range, distribution plot etc and identify the patterns and trends of single rows of the dataset.
3. With EDA we can also get the relationship between the two or more variables by making bivariate or multivariate analyses like regression, correlations, covariance, scatter plot, line plot etc.
4. With EDA we can find out the most influential feature of the dataset using correlations, covariance, and various bivariate or multivariate plotting.
5. With EDA we can also identify the outliers using Box plots and remove them further using a statistical approach.

EDA provides the groundwork for the entire data analysis process. It enables analysts to make more informed judgments about data processing, hypothesis testing, modelling, and interpretation, resulting in more accurate and relevant insights.

### 12. What is Time Series analysis?

[Time Series analysis](https://www.geeksforgeeks.org/time-series-data-visualization-in-python) is a statistical technique used to analyze and interpret data points collected at specific time intervals. Time series data is the data points recorded sequentially over time. The data points can be numerical, categorical, or both. The objective of time series analysis is to understand the underlying patterns, trends and behaviours in the data as well as to make forecasts about future values.

The key components of Time Series analysis are as follows:

* **Trend**: The data's long-term movement or direction over time. Trends can be upward, downward, or flat.
* **Seasonality**: Patterns that repeat at regular intervals, such as daily, monthly, or yearly cycles.
* **Cyclical Patterns**: Longer-term trends that are not as regular as seasonality, and are frequently associated with economic or business cycles.
* **Irregular Fluctuations**: Unpredictable and random data fluctuations that cannot be explained by trends, seasonality, or cycles.
* **Auto-correlations**: The link between a data point and its prior values. It quantifies the degree of dependence between observations at different time points.

Time series analysis approaches include a variety of techniques including Descriptive analysis to identify trends, patterns, and irregularities, smoothing techniques like moving averages or exponential smoothing to reduce noise and highlight underlying trends, Decompositions to separate the time series data into its individual components and forecasting technique like [ARIMA](https://www.geeksforgeeks.org/time-series-analysis-using-arima-model-in-r-programming), SARIMA, and [Regression](https://www.geeksforgeeks.org/types-of-regression-techniques) technique to predict the future values based on the trends.

### 13. What is Feature Engineering?

[Feature engineering](https://www.geeksforgeeks.org/what-is-feature-engineering) is the process of selecting, transforming, and creating features from raw data in order to build more effective and accurate machine learning models. The primary goal of feature engineering is to identify the most relevant features or create the relevant features by combining two or more features using some mathematical operations from the raw data so that it can be effectively utilized for getting predictive analysis by machine learning model.

The following are the key elements of feature engineering:

* **Feature Selection:**In this case we identify the most relevant features from the dataset based on the correlation with the target variables.
* **Create new feature:** In this case, we generate the new features by aggregating or transforming the existing features in such a way that it can be helpful to capture the patterns or trends which is not revealed by the original features.
* **Transformation**: In this case, we modify or scale the features so, that it can helpful in building the machine learning model. Some of the common transformations method are [Min-Max Scaling](https://www.geeksforgeeks.org/data-pre-processing-wit-sklearn-using-standard-and-minmax-scaler), Z-Score Normalization, and log transformations etc.
* **Feature encoding:**Generally ML algorithms only process the numerical data, so, that we need to encode categorical features into the numerical vector. Some of the popular encoding technique are [One-Hot-Encoding](https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python), Ordinal label encoding etc.

### 14. What is data normalization, and why is it important?

[Data normalization](https://www.geeksforgeeks.org/data-normalization-with-pandas) is the process of transforming numerical data into standardised range. The objective of data normalization is scale the different features (variables) of a dataset onto a common scale, which make it easier to compare, analyze, and model the data. This is particularly important when features have different units, scales, or ranges because if we doesn't normalize then each feature has different-different impact which can affect the performance of various machine learning algorithms and statistical analyses.

Common normalization techniques are as follows:

* **Min-Max Scaling:** Scales the data to a range between 0 and 1 using the formula:  
  (x - min) / (max - min)
* **Z-Score Normalization (Standardization):** Scales data to have a mean of 0 and a standard deviation of 1 using the formula:   
  (x - mean) / standard\_deviation
* **Robust Scaling:** Scales data by removing the median and scaling to the interquartile range(IQR) to handle outliers using the formula:   
  (X - Median) / IQR
* **Unit Vector Scaling:**Scales each data point to have a Euclidean norm (length) (||X||) of 1 using the formula:   
  X / ||X||

### 15. What are the main libraries you would use for data analysis in Python?

For data analysis in Python, many great libraries are used due to their versatility, functionality, and ease of use. Some of the most common libraries are as follows:

* [**NumPy**](https://www.geeksforgeeks.org/numpy-tutorial)**:** A core Python library for numerical computations. It supports arrays, matrices, and a variety of mathematical functions, making it a building block for many other data analysis libraries.
* [**Pandas**](https://www.geeksforgeeks.org/pandas-tutorial): A well-known data manipulation and analysis library. It provides data structures (like as DataFrames) that make to easily manipulate, filter, aggregate, and transform data. Pandas is required when working with structured data.
* [**SciPy**](https://www.geeksforgeeks.org/data-analysis-with-scipy): SciPy is a scientific computing library. It offers a wide range of statistical, mathematical, and scientific computing functions.
* [**Matplotlib**](https://www.geeksforgeeks.org/python-introduction-matplotlib): Matplotlib is a library for plotting and visualization. It provides a wide range of plotting functions, making it easy to create beautiful and informative visualizations.
* [**Seaborn**](https://www.geeksforgeeks.org/python-seaborn-tutorial): Seaborn is a library for statistical data visualization. It builds on top of Matplotlib and provides a more user-friendly interface for creating statistical plots.
* [**Scikit-learn**](https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library): A powerful machine learning library. It includes classification, regression, clustering, dimensionality reduction, and model evaluation tools. Scikit-learn is well-known for its consistent API and simplicity of use.
* [**Statsmodels**](https://www.geeksforgeeks.org/how-to-install-statsmodels-in-python): A statistical model estimation and interpretation library. It covers a wide range of statistical models, such as linear models and time series analysis.

### 16. What's the difference between structured and unstructured data?

Structured and unstructured data depend on the format in which the data is stored. Structured data is information that has been structured in a certain format, such as a table or spreadsheet. This facilitates searching, sorting, and analyzing. Unstructured data is information that is not arranged in a certain format. This makes searching, sorting, and analyzing more complex.

The differences between the structured and unstructured data are as follows:

| **Feature** | [Structured Data](https://www.geeksforgeeks.org/what-is-structured-data) | [Unstructured Data](https://www.geeksforgeeks.org/what-is-unstructured-data) |
| --- | --- | --- |
| **Structure of data** | Schema (structure of data) is often rigid and organized into rows and columns | No predefined relationships between data elements. |
| **Searchability** | Excellent for searching, reporting, and querying | Difficult to search |
| **Analysis** | Simple to quantify and process using standard database functions. | No fixed format, making it more challenging to organize and analyze. |
| **Storage** | Relational databases | Data lakes |
| **Examples** | Customer records, product inventories, financial data | Text documents, images, audio, video |

### 17. How can pandas be used for data analysis?

[Pandas](https://www.geeksforgeeks.org/pandas-tutorial) is one of the most widely used Python libraries for data analysis. It has powerful tools and data structure which is very helpful in analyzing and processing data. Some of the most useful functions of pandas which are used for various tasks involved in data analysis are as follows:

1. **Data loading functions:**Pandas provides different functions to read the dataset from the different-different formats like [read\_csv](https://www.geeksforgeeks.org/python-read-csv-using-pandas-read_csv), [read\_excel](https://www.geeksforgeeks.org/working-with-excel-files-using-pandas), and [read\_sql](https://www.geeksforgeeks.org/how-to-convert-pandas-dataframe-into-sql-in-python) functions are used to read the dataset from CSV, Excel, and SQL datasets respectively in a pandas DataFrame.
2. **Data Exploration:** Pandas provides functions like [head](https://www.geeksforgeeks.org/python-pandas-dataframe-series-head-method), [tail](https://www.geeksforgeeks.org/python-pandas-dataframe-series-tail-method), and [sample](https://www.geeksforgeeks.org/python-pandas-series-sample) to rapidly inspect the data after it has been imported. In order to learn more about the different data types, missing values, and summary statistics, use pandas .info and .describe functions.
3. **Data Cleaning:** Pandas offers functions for dealing with missing values ([fillna](https://www.geeksforgeeks.org/python-pandas-dataframe-fillna-to-replace-null-values-in-dataframe)), duplicate rows ([drop\_duplicates](https://www.geeksforgeeks.org/python-pandas-dataframe-drop_duplicates)), and incorrect data types ([astype](https://www.geeksforgeeks.org/python-pandas-dataframe-astype)) before analysis.
4. **Data Transformation:** Pandas may be used to modify and transform data. It is simple to do actions like selecting columns, filtering rows ([loc](https://www.geeksforgeeks.org/python-pandas-extracting-rows-using-loc), [iloc](https://www.geeksforgeeks.org/python-extracting-rows-using-pandas-iloc)), and adding new ones. Custom transformations are feasible using the [apply](https://www.geeksforgeeks.org/python-pandas-apply) and [map](https://www.geeksforgeeks.org/python-map-function) functions.
5. **Data Aggregation:** With the help of pandas, we can group the data using [groupby](https://www.geeksforgeeks.org/python-pandas-dataframe-groupby) function, and also apply aggregation tasks like [sum](https://www.geeksforgeeks.org/sum-function-python), [mean](https://www.geeksforgeeks.org/python-pandas-series-mean), [count](https://www.geeksforgeeks.org/count-values-in-pandas-dataframe), etc., on specify columns.
6. **Time Series Analysis:** Pandas offers robust support for time series data. We can easily conduct date-based computations using functions like [resample](https://www.geeksforgeeks.org/python-pandas-dataframe-resample), [shift](https://www.geeksforgeeks.org/python-pandas-series-shift) etc.
7. **Merging and Joining:** Data from different sources can be combined using Pandas [merge](https://www.geeksforgeeks.org/python-pandas-merging-joining-and-concatenating) and join functions.

### 18. What is the difference between pandas Series and pandas DataFrames?

In pandas, Both Series and Dataframes are the fundamental data structures for handling and analyzing tabular data. However, they have distinct characteristics and use cases.

A [**series**](https://www.geeksforgeeks.org/python-pandas-series) in pandas is a one-dimensional labelled array that can hold data of various types like integer, float, string etc. It is similar to a NumPy array, except it has an index that may be used to access the data. The index can be any type of object, such as a string, a number, or a datetime.

A pandas [**DataFrame**](https://www.geeksforgeeks.org/python-pandas-dataframe) is a two-dimensional labelled data structure resembling a table or a spreadsheet. It consists of rows and columns, where each column can have a different data type. A DataFrame may be thought of as a collection of Series, where each column is a Series with the same index.

The key differences between the pandas Series and Dataframes are as follows:

| **pandas Series** | **pandas DataFrames** |
| --- | --- |
| A one-dimensional labelled array that can hold data of various types like (integer, float, string, etc.) | A two-dimensional labelled data structure that resembles a table or a spreadsheet. |
| Similar to the single vector or column in a spreadsheet | Similar to a spreadsheet, which can have multiple vectors or columns as well as. |
| Best suited for working with single-feature data | The versatility and handling of the multiple features make it suitable for tasks like data analysis. |
| Each element of the Series is associated with its label known as the index | DataFrames can be assumed as a collection of multiple Series, where each column shares the same index. |

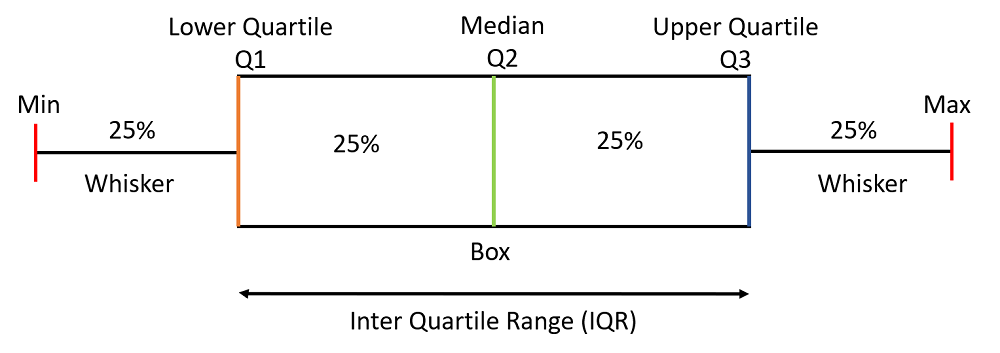
### 19. What is One-Hot-Encoding?

[One-hot encoding](https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python) is a technique used for converting categorical data into a format that machine learning algorithms can understand. Categorical data is data that is categorized into different groups, such as colors, nations, or zip codes. Because machine learning algorithms often require numerical input, categorical data is represented as a sequence of binary values using one-hot encoding.

To one-hot encode a categorical variable, we generate a new binary variable for each potential value of the category variable. For example, if the category variable is "color" and the potential values are "red," "green," and "blue," then three additional binary variables are created: "color\_red," "color\_green," and "color\_blue." Each of these binary variables would have a value of 1 if the matching category value was present and 0 if it was not.

### 20. What is a boxplot and how it's useful in data science?

A [boxplot](https://www.geeksforgeeks.org/box-plot-in-python-using-matplotlib) is a graphic representation of data that shows the distribution of the data. It is a standardized method of the distribution of a data set based on its five-number summary of data points: the minimum, first quartile [Q1], median, third quartile [Q3], and maximum.



Boxplot is used for detection the outliers in the dataset by visualizing the distribution of data.

## *Statistics Interview Questions and Answers for Data Analyst*

### 21. What is the difference between descriptive and inferential statistics?

[Descriptive statistics](https://www.geeksforgeeks.org/descriptive-statistic) and inferential statistics are the two main branches of statistics

* [**Descriptive Statistics**](https://www.geeksforgeeks.org/descriptive-statistic)**:**Descriptive statistics is the branch of statistics, which is used to summarize and describe the main characteristics of a dataset. It provides a clear and concise summary of the data's central tendency, variability, and distribution. Descriptive statistics help to understand the basic properties of data, identifying patterns and structure of the dataset without making any generalizations beyond the observed data. Descriptive statistics compute measures of central tendency and dispersion and also create graphical representations of data, such as histograms, bar charts, and pie charts to gain insight into a dataset.  
  Descriptive statistics is used to answer the following questions:
  + What is the mean salary of a data analyst?
  + What is the range of income of data analysts?
  + What is the distribution of monthly incomes of data analysts?
* [**Inferential Statistics**](https://www.geeksforgeeks.org/difference-between-descriptive-and-inferential-statistics)**:**Inferential statistics is the branch of statistics, that is used to conclude, make predictions, and generalize findings from a sample to a larger population. It makes inferences and hypotheses about the entire population based on the information gained from a representative sample. Inferential statistics use hypothesis testing, confidence intervals, and regression analysis to make inferences about a population.   
  Inferential statistics is used to answer the following questions:
  + Is there any difference in the monthly income of the Data analyst and the Data Scientist?
  + Is there any relationship between income and education level?
  + Can we predict someone's salary based on their experience?

### 22. What are measures of central tendency?

[Measures of central tendency](https://www.geeksforgeeks.org/measures-of-central-tendency) are the statistical measures that represent the centre of the data set. It reveals where the majority of the data points generally cluster. The three most common measures of central tendency are:

* **Mean**: The mean, also known as the average, is calculated by adding up all the values in a dataset and then dividing by the total number of values. It is sensitive to outliers since a single extreme number can have a large impact on the mean.  
  **Mean = (Sum of all values) / (Total number of values)**
* Median: The median is the middle value in a data set when it is arranged in ascending or descending order. If there is an even number of values, the median is the average of the two middle values.
* **Mode**: The mode is the value that appears most frequently in a dataset. A dataset can have no mode (if all values are unique) or multiple modes (if multiple values have the same highest frequency). The mode is useful for categorical data and discrete distributions.

### 23. What are the Measures of dispersion?

[Measures of dispersion](https://www.geeksforgeeks.org/measure-of-dispersion), also known as measures of variability or spread, indicate how much the values in a dataset deviate from the central tendency. They help in quantifying how far the data points vary from the average value.

Some of the common Measures of dispersion are as follows:

* **Range**: The range is the difference between the highest and lowest values in a data set. It gives an idea of how much the data spreads from the minimum to the maximum.
* **Variance**: The variance is the average of the squared deviations of each data point from the mean. It is a measure of how spread out the data is around the mean.  
  Variance(σ2)=∑(X−μ)2NVariance(*σ*2)=*N*∑(*X*−*μ*)2​
* **Standard Deviation**: The standard deviation is the square root of the variance. It is a measure of how spread out the data is around the mean, but it is expressed in the same units as the data itself.
* **Mean Absolute Deviation (MAD)**: MAD is the average of the absolute differences between each data point and the mean. Unlike variance, it doesn't involve squaring the differences, making it less sensitive to extreme values. it is less sensitive to outliers than the variance or standard deviation.
* **Percentiles**: Percentiles are statistical values that measure the relative positions of values within a dataset. Which is computed by arranging the dataset in descending order from least to the largest and then dividing it into 100 equal parts. In other words, a percentile tells you what percentage of data points are below or equal to a specific value. Percentiles are often used to understand the distribution of data and to identify values that are above or below a certain threshold within a dataset.
* **Interquartile Range (IQR)**: The interquartile range (IQR) is the range of values ranging from the 25th percentile (first quartile) to the 75th percentile (third quartile). It measures the spread of the middle 50% of the data and is less affected by outliers.
* **Coefficient of Variation (CV)**: The coefficient of variation (CV) is a measure of relative variability, It is the ratio of the standard deviation to the mean, expressed as a percentage. It's used to compare the relative variability between datasets with different units or scales.

### 24. What is a probability distribution?

A [probability distribution](https://www.geeksforgeeks.org/probability-distribution) is a mathematical function that estimates the probability of different possible outcomes or events occurring in a random experiment or process. It is a mathematical representation of random phenomena in terms of [sample space](https://www.geeksforgeeks.org/what-is-the-probability-sample-space-of-tossing-4-coins) and [event probability](https://www.geeksforgeeks.org/events-in-probability), which helps us understand the relative possibility of each outcome occurring.

There are two main types of probability distributions:

1. **Discrete Probability Distribution**: In a discrete probability distribution, the random variable can only take on distinct, separate values. Each value is associated with a probability. Examples of discrete probability distributions include the binomial distribution, the Poisson distribution, and the hypergeometric distribution.
2. **Continuous Probability Distribution**: In a continuous probability distribution, the random variable can take any value within a certain range. These distributions are described by probability density functions (PDFs). Examples of continuous probability distributions include the normal distribution, the exponential distribution, and the uniform distribution.

### 25. What are normal distributions?

A [normal distribution](https://www.geeksforgeeks.org/normal-distribution-in-r), also known as a Gaussian distribution, is a specific type of probability distribution with a symmetric, bell-shaped curve. The data in a normal distribution clustered around a central value i.e mean, and the majority of the data falls within one standard deviation of the mean. The curve gradually tapers off towards both tails, showing that extreme values are becoming

distribution having a mean equal to 0 and standard deviation equal to 1 is known as standard normal distribution and Z-scores are used to measure how many standard deviations a particular data point is from the mean in standard normal distribution.

Normal distributions are a fundamental concept that supports many statistical approaches and helps researchers understand the behaviour of data and variables in a variety of scenarios.

### 26. What is the central limit theorem?

The [Central Limit Theorem (CLT)](https://www.geeksforgeeks.org/central-limit-theorem) is a fundamental concept in statistics that states that, under certain conditions, the distribution of sample means approaches a normal distribution as sample size rises, regardless of the the original population distribution. In other words, even if the population distribution is not normal, when the sample size is high enough, the distribution of sample means will tend to be normal.

The Central Limit Theorem has three main assumptions:

1. The samples must be independent. This means that the outcome of one sample cannot affect the outcome of another sample.
2. The samples must be random. This means that each sample must be drawn from the population in a way that gives all members of the population an equal chance of being selected.
3. The sample size must be large enough. The CLT typically applies when the sample size is greater than 30.

### 27. What are the null hypothesis and alternative hypotheses?

In statistics, the null and alternate hypotheses are two mutually exclusive statements regarding a population parameter. A hypothesis test analyzes sample data to determine whether to accept or reject the null hypothesis. Both null and alternate hypotheses represent the opposing statements or claims about a population or a phenomenon under investigation.

* [**Null Hypothesis**](https://www.geeksforgeeks.org/difference-between-null-and-alternate-hypothesis)**(**H0 *H*0​ **)**: The null hypothesis is a statement regarding the status quo representing no difference or effect after the phenomena unless there is strong evidence to the contrary.
* [**Alternate Hypothesis**](https://www.geeksforgeeks.org/difference-between-null-and-alternate-hypothesis)**(**Ha or H1 *Ha*​ or *H*1​ ): The alternate hypothesis is a statement that disregards the status quo means supports the difference or effect. The researcher tries to prove the hypothesis.

### 28. What is a p-value, and what does it mean?

A [p-value](https://www.geeksforgeeks.org/p-value), which stands for "probability value," is a statistical metric used in hypothesis testing to measure the strength of evidence against a null hypothesis. When the null hypothesis is considered to be true, it measures the chance of receiving observed outcomes (or more extreme results). In layman's words, the p-value determines whether the findings of a study or experiment are statistically significant or if they might have happened by chance.

The p-value is a number between 0 and 1, which is frequently stated as a decimal or percentage. If the null hypothesis is true, it indicates the probability of observing the data (or more extreme data).

### 29. What is the significance level?

The [significance level](https://www.geeksforgeeks.org/how-to-interpret-significance-codes-in-r), often denoted as α (alpha), is a critical parameter in hypothesis testing and statistical analysis. It defines the threshold for determining whether the results of a statistical test are statistically significant. In other words, it sets the standard for deciding when to reject the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

If the p-value is less than the significance level, we reject the null hypothesis and conclude that there is a statistically significant difference between the groups.

* If p-value ≤ α: Reject the null hypothesis. This indicates that the results are statistically significant, and there is evidence to support the alternative hypothesis.
* If p-value > α: Fail to reject the null hypothesis. This means that the results are not statistically significant, and there is insufficient evidence to support the alternative hypothesis.

The choice of a significance level involves a trade-off between Type I and Type II errors. A lower significance level (e.g., α = 0.01) decreases the risk of Type I errors while increasing the chance of Type II errors (failure to identify a real impact). A higher significance level (e.g., = 0.10), on the other hand, increases the probability of Type I errors while decreasing the chance of Type II errors.

### 30. Describe Type I and Type II errors in hypothesis testing.

In hypothesis testing, When deciding between the null hypothesis (H0) and the alternative hypothesis (Ha), two types of errors may occur. These errors are known as Type I and Type II errors, and they are important considerations in statistical analysis.

* **Type I error (False Positive, α)**: Type I error occurs when the null hypothesis is rejected when it is true. This is also referred as a false positive. The probability of committing a Type I error is denoted by α (alpha) and is also known as the significance level. A lower significance level (e.g., = 0.05) reduces the chance of Type I mistakes while increasing the risk of Type II errors.  
  For example, a Type I error would occur if we estimated that a new medicine was successful when it was not.
  + **Type I Error (False Positive, α): Rejecting a true null hypothesis.**
* [**Type II Error**](https://www.geeksforgeeks.org/type-ii-error-in-two-tailed-test-of-population-mean-with-unknown-variance-in-r)**(False Negative, β)**: Type II error occurs when a researcher fails to reject the null hypothesis when it is actually false. This is also referred as a false negative. The probability of committing a Type II error is denoted by β (beta)  
  For example, a Type II error would occur if we estimated that a new medicine was not effective when it is actually effective.
  + **Type II Error (False Negative, β): Failing to reject a false null hypothesis.**

### 31. What is a confidence interval, and how does it is related to point estimates?

The [confidence interval](https://www.geeksforgeeks.org/confidence-interval) is a statistical concept used to estimates the uncertainty associated with estimating a population parameter (such as a population mean or proportion) from a sample. It is a range of values that is likely to contain the true value of a population parameter along with a level of confidence in that statement.

* **Point estimate:** A point estimate is a single that is used to estimate the population parameter based on a sample. For example, the sample mean (x̄) is a point estimate of the population mean (μ). The point estimate is typically the sample mean or the sample proportion.
* **Confidence interval:** A confidence interval, on the other hand, is a range of values built around a point estimate to account for the uncertainty in the estimate. It is typically expressed as an interval with an associated confidence level (e.g., 95% confidence interval). The degree of confidence or confidence level shows the probability that the interval contains the true population parameter.

The relationship between point estimates and confidence intervals can be summarized as follows:

* A point estimate provides a single value as the best guess for a population parameter based on sample data.
* A confidence interval provides a range of values around the point estimate, indicating the range of likely values for the population parameter.
* The confidence level associated with the interval reflects the level of confidence that the true parameter value falls within the interval.

For example, A 95% confidence interval indicates that you are 95% confident that the real population parameter falls inside the interval. A 95% confidence interval for the population mean (μ) can be expressed as :

(xˉ−Margin of error,xˉ+Margin of error)(*x*ˉ−Margin of error,*x*ˉ+Margin of error)

where x̄ is the point estimate (sample mean), and the margin of error is calculated using the standard deviation of the sample and the confidence level.

### 32. What is ANOVA in Statistics?

[ANOVA](https://www.geeksforgeeks.org/anova-formula), or Analysis of Variance, is a statistical technique used for analyzing and comparing the means of two or more groups or populations to determine whether there are statistically significant differences between them or not. It is a parametric statistical test which means that, it assumes the data is normally distributed and the variances of the groups are identical. It helps researchers in determining the impact of one or more categorical independent variables (factors) on a continuous dependent variable.

ANOVA works by partitioning the total variance in the data into two components:

* **Between-group variance:** It analyzes the difference in means between the different groups or treatment levels being compared.
* **Within-group variance:** It analyzes the variance within each individual group or treatment level.

Depending on the investigation's design and the number of independent variables, ANOVA has numerous varieties:

* **One-Way ANOVA:**Compares the means of three or more independent groups or levels of a single categorical variable. For Example: One-way ANOVA can be used to compare the average age of employees among the three different teams in a company.
* **Two-Way ANOVA:** Compare the means of two or more independent groups while taking into account the impact of a two independent categorical variables (factors) . For example, Two-way ANOVA can be to compare the average age of employees among the three different teams in a company, while also taking into account the gender of the employees.
* **Multivariate Analysis of Variance (MANOVA):** Compare the means of multiple dependent variables. For example, MANOVA can be used to compare the average age, average salary, and average experience of employees among the three different teams in a company.

### 33. What is a correlation?

[Correlation](https://www.geeksforgeeks.org/exploring-correlation-in-python) is a statistical term that analyzes the degree of a linear relationship between two or more variables. It estimates how effectively changes in one variable predict or explain changes in another.Correlation is often used to access the strength and direction of associations between variables in various fields, including statistics, economics.

The correlation between two variables is represented by correlation coefficient, denoted as "r". The value of "r" can range between -1 and +1, reflecting the strength of the relationship:

* **Positive correlation (r > 0):** As one variable increases, the other tends to increase. The greater the positive correlation, the closer "r" is to +1.
* **Negative correlation (r < 0):** As one variable rises, the other tends to fall. The closer "r" is to -1, the greater the negative correlation.
* **No correlation (r = 0):**There is little or no linear relationship between the variables.

### 34. What are the differences between Z-test, T-test and F-test?

The Z-test, t-test, and F-test are statistical hypothesis tests that are employed in a variety of contexts and for a variety of objectives.

* [**Z-test**](https://www.geeksforgeeks.org/z-test)**:** The Z-test is performed when the population standard deviation is known. It is a parametric test, which means that it makes certain assumptions about the data, such as that the data is normally distributed. The Z-test is most accurate when the sample size is large.
* [**T-test**](https://www.geeksforgeeks.org/t-test)**:** The T-test is performed when the population standard deviation is unknown. It is also a parametric test, but unlike the Z-test, it is less sensitive to violations of the normality assumption. The T-test is most accurate when the sample size is large.
* [**F-test**](https://www.geeksforgeeks.org/how-to-perform-an-f-test-in-python)**:**The F-test is performed to compare two or more groups' variances. It assume that populations being compared follow a normal distribution.. When the sample sizes of the groups are equal, the F-test is most accurate.

The key differences between the Z-test, T-test, and F-test are as follows:

|  | **Z-Test** | **T-Test** | **F-Test** |
| --- | --- | --- | --- |
| **Assumptions** | 1. Population follows a normal distribution. 2. Population standard deviation is known | 1. Population follows a normal distribution or the sample size is large enough for the Central Limit Theorem to apply. 2. Also applied when the population standard deviation is unknown. | 1. The variances of the populations from which the samples are drawn should be equal (homoscedastic). 2. Populations being compared have normal distributions and that the samples are independent. |
| **Data** | N>30 | N<30 or population standard deviation is unknown. | Used to test the variances |
| **Formula** | Z-Test=xˉ−μσ/NZ-Test=*σ*/*N*​*x*ˉ−*μ*​ | T-test=xˉ−μS/nT-test=*S*/*n*​*x*ˉ−*μ*​ | F-Test=σ12σ22F-Test=*σ*22​*σ*12​​ |

### 35. What is linear regression, and how do you interpret its coefficients?

[Linear regression](https://www.geeksforgeeks.org/ml-linear-regression) is a statistical approach that fits a linear equation to observed data to represent the connection between a dependent variable (also known as the target or response variable) and one or more independent variables (also known as predictor variables or features). It is one of the most basic and extensively used regression analysis techniques in statistics and machine learning. Linear regression presupposes that the independent variables and the dependent variable have a linear relationship.

A simple linear regression model can be represented as:

Y=β0+β1X+ϵ*Y*=*β*0​+*β*1​*X*+*ϵ*

Where:

* Y: Dependent variable or Target
* X: Independent variables
* β0 *β*0​ is the intercept (i.e value of Y when X =0)
* β1 *β*1​ is the coefficient for the independent variable X, representing the change in Y for a one-unit change in X.
* ϵ *ϵ* is represents the error term (i.e Difference between the actual and predicted value from the linear relationship.

## *SQL Interview Questions for Data Analysts*

### 36. What is DBMS?

[DBMS](https://www.geeksforgeeks.org/dbms) stands for Database Management System. It is software designed to manage, store, retrieve, and organize data in a structured manner. It provides an interface or a tool for performing CRUD operations into a database. It serves as an intermediary between the user and the database, allowing users or applications to interact with the database without the need to understand the underlying complexities of data storage and retrieval.

### 37. What are the basic SQL CRUD operations?

[SQL CRUD](https://www.geeksforgeeks.org/sql-server-crud-operations) stands for CREATE, READ(SELECT), UPDATE, and DELETE statements in SQL Server. CRUD is nothing but Data Manipulation Language (DML) Statements. CREATE operation is used to insert new data or create new records in a database table, READ operation is used to retrieve data from one or more tables in a database, UPDATE operation is used to modify existing records in a database table and DELETE is used to remove records from the database table based on specified conditions. Following are the basic query syntax examples of each operation:

**CREATE**

It is used to create the table and insert the values in the database. The commands used to create the table are as follows:

INSERT INTO employees (first\_name, last\_name, salary)

VALUES ('Pawan', 'Gunjan', 50000);

**READ**

Used to retrive the data from the table

SELECT \* FROM employees;

**UPDATE**

Used to modify the existing records in the database table

UPDATE employees

SET salary = 55000

WHERE last\_name = 'Gunjan';

**DELETE**

Used to remove the records from the database table

DELETE FROM employees

WHERE first\_name = 'Pawan';

### 38. What is the SQL statement used to insert new records into a table?

We use the '[INSERT](https://www.geeksforgeeks.org/sql-insert-statement)' statement to insert new records into a table. The 'INSERT INTO' statement in SQL is used to add new records (rows) to a table.

**Syntax**

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

**Example**

INSERT INTO Customers (CustomerName, City, Country)

VALUES ('Shivang', 'Noida', 'India');

### 39. How do you filter records using the WHERE clause in SQL?

We can filter records using the '[WHERE](https://www.geeksforgeeks.org/sql-where-clause)' clause by including 'WHERE' clause in 'SELECT' statement, specifying the conditions that records must meet to be included.

**Syntax**

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

**Example :**In this example, we are fetching the records of employee where job title is Developer.

SELECT \* FROM employees

WHERE job\_title = 'Developer';

### 40. How can you sort records in ascending or descending order using SQL?

We can sort records in ascending or descending order by using '[ORDER BY](https://www.geeksforgeeks.org/sql-order-by); clause with the 'SELECT' statement. The 'ORDER BY' clause allows us to specify one or more columns by which you want to sort the result set, along with the desired sorting order i.e ascending or descending order.

**Syntax for sorting records in ascending order**

SELECT column1, column2, ...

FROM table\_name

ORDER BY Column\_To\_Sort1 ASC, Column\_To\_Sort2 ASC, ...;

**Example:**This statement selects all customers from the 'Customers' table, sorted ascending by the 'Country'

SELECT \* FROM Customers

ORDER BY Country ASC;

**Syntax for sorting records in descending order**

SELECT column1, column2, ...

FROM table\_name

ORDER BY column\_to\_sort1 DESC, column\_to\_sort2 DESC, ...;

**Example:**This statement selects all customers from the 'Customers' table, sorted descending by the 'Country' column

SELECT \* FROM Customers

ORDER BY Country DESC;

### 41. Explain the purpose of the GROUP BY clause in SQL.

The purpose of [GROUP BY](https://www.geeksforgeeks.org/sql-group-by) clause in SQL is to group rows that have the same values in specified columns. It is used to arrange different rows in a group if a particular column has the same values with the help of some functions.

**Syntax**

SELECT column1, function\_name(column2)

FROM table\_name

GROUP BY column\_name(s);

**Example:**This SQL query groups the 'CUSTOMER' table based on age by using GROUP BY

SELECT AGE, COUNT(Name)

FROM CUSTOMERS

GROUP BY AGE;

### 42. How do you perform aggregate functions like SUM, COUNT, AVG, and MAX/MIN in SQL?

An [aggregate](https://www.geeksforgeeks.org/aggregate-functions-in-sql) function groups together the values of multiple rows as input to form a single value of more significant meaning. It is also used to perform calculations on a set of values and then returns a single result. Some examples of aggregate functions are SUM, COUNT, AVG, and MIN/MAX.

**SUM:**It calculates the sum of values in a column.

**Example:**In this example, we are calculating sum of costs from cost column in PRODUCT table.

SELECT SUM(Cost)

FROM Products;

**COUNT:**It counts the number of rows in a result set or the number of non-null values in a column.

**Example:**Ij this example, we are counting the total number of orders in an "orders" table.

SELECT COUNT(\*)

FROM Orders;

**AVG:**It calculates the average value of a numeric column.

**Example:**In this example, we are finding average salary of employees in an "employees" table.

SELECT AVG(Price)

FROM Products;

**MAX:** It returns the maximum value in a column.

**Example:**In this example, we are finding the maximum temperature in the 'weather' table.

SELECT MAX(Price)

FROM Orders;

**MIN:**It returns the minimum value in a column.

**Example:**In this example, we are finding the minimum price of a product in a "products" table.

SELECT MIN(Price)

FROM Products;

### 43. What is an SQL join operation? Explain different types of joins (INNER, LEFT, RIGHT, FULL).

[SQL Join](https://www.geeksforgeeks.org/joining-three-tables-sql) operation is used to combine data or rows from two or more tables based on a common field between them. The primary purpose of a join is to retrieve data from multiple tables by linking records that have a related value in a specified column. There are different types of join i.e, [INNER, LEFT, RIGHT, FULL.](https://www.geeksforgeeks.org/sql-join-set-1-inner-left-right-and-full-joins) These are as follows:

**INNER JOIN:**The INNER JOIN keyword selects all rows from both tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e the value of the common field will be the same.

**Example:**

SELECT customers.customer\_id, orders.order\_id

FROM customers

INNER JOIN orders

ON customers.customer\_id = orders.customer\_id;

**LEFT JOIN:**A LEFT JOIN returns all rows from the left table and the matching rows from the right table.

**Example:**

SELECT departments.department\_name, employees.first\_name

FROM departments

LEFT JOIN employees

ON departments.department\_id = employees.department\_id;

**RIGHT JOIN:**RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join.

**Example:**

SELECT employees.first\_name, orders.order\_id

FROM employees

RIGHT JOIN orders

ON employees.employee\_id = orders.employee\_id;

**FULL JOIN:**FULL JOIN creates the result set by combining the results of both LEFT JOIN and RIGHT JOIN. The result set will contain all the rows from both tables.

**Example:**

SELECT customers.customer\_id, orders.order\_id

FROM customers

FULL JOIN orders

ON customers.customer\_id = orders.customer\_id;

### 44. How can you write an SQL query to retrieve data from multiple related tables?

To retrieve data from multiple related tables, we generally use 'SELECT' statement along with help of '[JOIN](https://www.geeksforgeeks.org/sql-join-set-1-inner-left-right-and-full-joins)' operation by which we can easily fetch the records from the multiple tables. Basically, JOINS are used when there are common records between two tables. There are different types of joins i.e. INNER, LEFT, RIGHT, FULL JOIN. In the above question, detailed explanation is given regarding JOIN so you can refer that.

### 45. What is a subquery in SQL? How can you use it to retrieve specific data?

A [subquery](https://www.geeksforgeeks.org/sql-subquery) is defined as query with another query. A subquery is a query embedded in WHERE clause of another SQL query. Subquery can be placed in a number of SQL clause: WHERE clause, HAVING clause, FROM clause. Subquery is used with SELECT, INSERT, DELETE, UPDATE statements along with expression operator. It could be comparison or equality operator such as =>,=,<= and like operator.

**Example 1: Subquery in the SELECT Clause**

SELECT customer\_name,

(SELECT COUNT(\*) FROM orders WHERE orders.customer\_id = customers.customer\_id) AS order\_count

FROM customers;

**Example 2: Subquery in the WHERE Clause**

SELECT employee\_name, salary

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

**Example 3: Subquery in the FROM Clause (Derived Tables)**

SELECT category, SUM(sales) AS total\_sales

FROM (SELECT product\_id, category, sales FROM products) AS derived\_table

GROUP BY category;

46. Can you give an example of using a subquery in combination with an IN or EXISTS condition?

We can use subquery in combination with [IN or EXISTS](https://www.geeksforgeeks.org/in-vs-exists-in-sql) condition. Example of using a subquery in combination with IN is given below. In this example, we will try to find out the geek’s data from table geeks\_data, those who are from the computer science department with the help of geeks\_dept table using sub-query.

**Using a Subquery with IN**

SELECT f\_name, l\_name

FROM geeks\_data

WHERE dept IN

(SELECT dep\_name FROM geeks\_dept WHERE dept\_id = 1);

**Using a Subquery with EXISTS:**

SELECT DISTINCT store\_t

FROM store

WHERE EXISTS (SELECT \* FROM city\_store WHERE city\_store.store\_t = store.store\_t);

### 47. What is the purpose of the HAVING clause in SQL? How is it different from the WHERE clause?

In SQL, the HAVING clause is used to filter the results of a GROUP BY query depending on aggregate functions applied to grouped columns. It allows you to filter groups of rows that meet specific conditions after grouping has been performed. The HAVING clause is typically used with aggregate functions like SUM, COUNT, AVG, MAX, or MIN.

The main differences between [HAVING](https://www.geeksforgeeks.org/sql-having-clause-with-examples) and [WHERE](https://www.geeksforgeeks.org/sql-where-clause) clauses are as follows:

| **HAVING** | **WHERE** |
| --- | --- |
| The HAVING clause is used to filter groups of rows after grouping. It operates on the results of aggregate functions applied to grouped columns. | The WHERE clause is used to filter rows before grouping. It operates on individual rows in the table and is applied before grouping and aggregation. |
| The HAVING clause is typically used with GROUP BY queries. It filters groups of rows based on conditions involving aggregated values. | The WHERE clause can be used with any SQL query, whether it involves grouping or not. It filters individual rows based on specified conditions. |
| In the HAVING clause, you generally use aggregate functions (e.g., SUM, COUNT) to reference grouped columns and apply conditions to groups of rows. | In the WHERE clause, you can reference columns directly and apply conditions to individual rows. |
| Command:   SELECT customer\_id, SUM(order\_total) AS total\_order\_amount  FROM orders  GROUP BY customer\_id  HAVING SUM(order\_total) > 1000; | Command:  SELECT customer\_id, SUM(order\_total) AS total\_order\_amount  FROM orders  GROUP BY customer\_id  WHERE total\_order\_amount > 1000; |

### 48. How do you use the UNION and UNION ALL operators in SQL?

In SQL, the [UNION](https://www.geeksforgeeks.org/union-and-union-all-in-ms-sql-server) and [UNION ALL](https://www.geeksforgeeks.org/union-union_all-functions-in-dplyr-package-in-r) operators are used to combine the result sets of multiple SELECT statements into a single result set. These operators allow you to retrieve data from multiple tables or queries and present it as a unified result. However, there are differences between the two operators:

#### 1. UNION Operator:

The UNION operator returns only distinct rows from the combined result sets. It removes duplicate rows and returns a unique set of rows. It is used when you want to combine result sets and eliminate duplicate rows.

**Syntax:**

SELECT column1, column2, ...

FROM table1

UNION

SELECT column1, column2, ...

FROM table2;

**Example:**

select name, roll\_number

from student

UNION

select name, roll\_number

from marks

#### **2. UNION ALL Operator:**

The UNION ALL operator returns all rows from the combined result sets, including duplicates. It does not remove duplicate rows and returns all rows as they are. It is used when you want to combine result sets but want to include duplicate rows.  
**Syntax:**

SELECT column1, column2, ...

FROM table1

UNION ALL

SELECT column1, column2, ...

FROM table2;

**Example:**

select name, roll\_number

from student

UNION ALL

select name, roll\_number

from marks

### 49. Explain the concept of database normalization and its importance.

[Database Normalization](https://www.geeksforgeeks.org/database-normalization-vs-database-optimization) is the process of reducing data redundancy in a table and improving data integrity. It is a way of organizing data in a database. It involves organizing the columns and tables in the database to ensure that their dependencies are correctly implemented using database constraints.

It is important because of the following reasons:

* It eliminates redundant data.
* It reduces the chances of data error.
* The normalization is important because it allows the database to take up less disk space.
* It also helps in increasing the performance.
* It improves the data integrity and consistency.

### 50. Can you list and briefly describe the normal forms (1NF, 2NF, 3NF) in SQL?

Normalization can take numerous forms, the most frequent of which are 1NF (First Normal Form), 2NF (Second Normal Form), and 3NF (Third Normal Form). Here's a quick rundown of each:

* [**First Normal Form (1NF)**](https://www.geeksforgeeks.org/first-normal-form-1nf)**:** In 1NF, each table cell should contain only a single value, and each column should have a unique name. 1NF helps in eliminating duplicate data and simplifies the queries. It is the fundamental requirement for a well-structured relational database. 1NF eliminates all the repeating groups of the data and also ensures that the data is organized at its most basic granularity.
* [**Second Normal Form (2NF)**](https://www.geeksforgeeks.org/second-normal-form-2nf)**:** In 2NF, it eliminates the partial dependencies, ensuring that each of the non-key attributes in the table is directly related to the entire primary key. This further reduces data redundancy and anomalies. The Second Normal form (2NF) eliminates redundant data by requiring that each non-key attribute be dependent on the primary key. In 2NF, each column should be directly related to the primary key, and not to other columns.
* [**Third Normal Form (3NF)**](https://www.geeksforgeeks.org/third-normal-form-3nf)**:** Third Normal Form (3NF) builds on the Second Normal Form (2NF) by requiring that all non-key attributes are independent of each other. This means that each column should be directly related to the primary key, and not to any other columns in the same table.

### 51. Explain window functions in SQL. How do they differ from regular aggregate functions?

In SQL, [window functions](https://www.geeksforgeeks.org/window-functions-in-sql) provide a way to perform complex calculations and analysis without the need for self-joins or subqueries.

SELECT col\_name1,

window\_function(col\_name2)

OVER([PARTITION BY col\_name1] [ORDER BY col\_name3]) AS new\_col

FROM table\_name;provides

**Example**:

SELECT

department,

AVG(salary) OVER(PARTITION BY department ORDER BY employee\_id) AS avg\_salary

FROM

employees;

**Window vs**[**Regular Aggregate Function**](https://www.geeksforgeeks.org/how-to-use-select-with-aggregate-functions-in-sql)

| **Window Functions** | **Aggregate Functions** |
| --- | --- |
| Window functions perform calculations within a specific "window" or subset of rows defined by an OVER() clause. It can be customized based on specific criteria, such as rows with the same values in a certain column or rows that are ordered in a specific way. | Regular aggregate functions operate on the entire result set and return a single value for the entire set of rows. |
| Window functions return a result for each row in the result set based on its specific window. Each row can have a different result. | Aggregate functions return a single result for the entire dataset. Each row receives the same value. |
| Window functions provide both an aggregate result and retain the details of individual rows within the defined window. | Regular aggregates provide a summary of the entire dataset, often losing detail about individual rows. |
| Window functions require the use of the OVER() clause to specify the window's characteristics, such as the partitioning and ordering of rows. | Regular aggregate functions do not use the OVER() clause because they do not have a notion of windows. |

### 52. What are primary keys and foreign keys in SQL? Why are they important?

Primary keys and foreign keys are two fundamental concepts in SQL that are used to build and enforce connections between tables in a relational database management system (RDBMS).

* [**Primary key:**](https://www.geeksforgeeks.org/primary-key-in-dbms)Primary keys are used to ensure that the data in the specific column is always unique. In this, a column cannot have a NULL value. The primary key is either an existing table column or it's specifically generated by the database itself according to a sequence.  
  **Importance of Primary Keys:**
  + Uniqueness
  + Query Optimization
  + Data Integrity
  + Relationships
  + Data Retrieval
* [**Foreign key**](https://www.geeksforgeeks.org/difference-between-primary-key-and-foreign-key)**:** Foreign key is a group of column or a column in a database table that provides a link between data in given two tables. Here, the column references a column of another table.  
  **Importance of Foreign Keys:**
  + Relationships
  + Data Consistency
  + Query Efficiency
  + Referential Integrity
  + Cascade Actions

### 53. Describe the concept of a database transaction. Why is it important to maintain data integrity?

[Database transactions](https://www.geeksforgeeks.org/sql-transactions) are the set of operations that are usually used to perform logical work. Database transactions mean that data in the database has been changed. It is one of the major characteristics provided in DBMS i.e. to protect the user's data from system failure. It is done by ensuring that all the data is restored to a consistent state when the computer is restarted. It is any one execution of the user program. Transaction's one of the most important properties is that it contains a finite number of steps.

They are important to maintain data integrity because they ensure that the database always remains in a valid and consistent state, even in the presence of multiple users or several operations. Database transactions are essential for maintaining data integrity because they enforce ACID properties i.e, atomicity, consistency, isolation, and durability properties. Transactions provide a solid and robust mechanism to ensure that the data remains accurate, consistent, and reliable in complex and concurrent database environments. It would be challenging to guarantee data integrity in relational database systems without database transactions.

### 54. Explain how NULL values are handled in SQL queries, and how

### you can use functions like IS NULL and IS NOT NULL.

In SQL, [NULL](https://www.geeksforgeeks.org/sql-null-functions)is a special value that usually represents that the value is not present or absence of the value in a database column. For accurate and meaningful data retrieval and manipulation, handling NULL becomes crucial. SQL provides IS NULL and IS NOT NULL operators to work with NULL values.

**IS NULL:**IS NULL operator is used to check whether an expression or column contains a NULL value.

**Syntax:**

SELECT column\_name(s) FROM table\_name WHERE column\_name IS NULL;

**Example:** In the below example, the query retrieves all rows from the employee table where the middle name contains NULL values.

SELECT \* FROM employees WHERE mid\_name IS NULL;

**IS NOT NULL:** IS NOT NULL operator is used to check whether an expression or column does not contain a NULL value.

**Syntax:**

SELECT column\_name(s) FROM table\_name WHERE column\_name IS NOT NULL;

**Example:**In the below example, the query retrieves all rows from the employee table where the first name does not contains NULL values.

SELECT \* FROM employees WHERE first\_name IS NOT NULL;

### 55. What is the difference between normalization and denormalization in database design.

[Normalization](https://www.geeksforgeeks.org/difference-between-normalization-and-denormalization) is used in a database to reduce the data redundancy and inconsistency from the table. [Denormalization](https://www.geeksforgeeks.org/denormalization-in-databases) is used to add data redundancy to execute the query as quick as possible.

| **S.NO** | **Normalization** | **Denormalization** |
| --- | --- | --- |
| 1. | Non-redundant and consistent data are stored in set schema. | Data are combined to execute a query as quick as possible |
| 2. | Data inconsistency and redundancy is reduced. | Addition of redundancy takes place for better execution of queries |
| 3. | Data integrity takes place and maintained. | Data integrity is not maintained |
| 4. | Data redundancy is eliminated or reduced. | Redundancy is added instead of elimination or reduction. |
| 5. | Number of tables is increased. | Number of tables is decreased. |
| 6. | Optimized the use of disk space. | Does not optimize the use of disk space. |

## *Data Visualizations or BI tools Interview questions*

### **56. Explain the difference between a dimension and a measure in Tableau.**

In Tableau, [dimensions and measures](https://www.geeksforgeeks.org/introduction-to-tableau) are two fundamental types of fields used for data visualization and analysis. They serve distinct purposes and have different characteristics:

| **Attributes** | **Dimension** | **Measure** |
| --- | --- | --- |
| **Nature** | They are categorical or qualitative data fields. They represent categories, labels or attributes by which you can segment and group your data. | They are numerical or quantitative data fields. They represent quantities, amounts or values that can be aggregated, or calculated. |
| **Usage** | They are used for grouping and segmenting data, creating hierarchies and the structure for visualizations. | They are used for performing calculations, and creating the numerical representation of the data as sum, average, etc. |
| **Example** | Category, Region, Product name, etc. | Sales(sum of sales), Profit(sum of profit), Quantity(sum of quantity), etc. |

### 57. **What are the dashboard, worksheet, Story, and Workbook in Tableau?**

Tableau is a robust data visualization and business intelligence solution that includes a variety of components for producing, organizing, and sharing data-driven insights. Here's a rundown of some of Tableau's primary components:

* [**Dashboard**](https://www.geeksforgeeks.org/tableau-objects-on-dashboard) : A dashboard is a collection of views(worksheets) arranged on a single page, designed to provide an interactive and holistic view of data. They include charts, maps, tables and other web content. Dashboards combine different visualizations into a single interface to allow users to comprehensively display and understand data. They are employed in the production of interactive reports and the provision of quick insights.   
  Dashboards support the actions and interactivity, enabling the users to filter and highlight the data dynamically. Dashboard behaviour can be modified with parameters and quick filters.
* **Worksheet:**A worksheet serves as the fundamental building element for creating data visualizations. To build tables, graphs, and charts, drag and drop fields onto the sheet or canvas. They are used to design individual visualizations and we can create various types of charts, apply filters, and customize formatting within a worksheet.  
  Worksheets offer a wide range of visualization options, including bar charts, line charts, scatter plots, etc. It also allows you to use reference lines, blend data and create calculated fields.
* **Story**: A story is a sequence or narrative created by combining sheets into a logical flow. Each story point represents a step in the narrative. Stories are used to systematically lead viewers through a set of visualizations or insights. They are useful for telling data-driven stories or presenting data-driven narratives.   
  Stories allow you to add text descriptions, annotations, and captions to every story point. Users can navigate through the story interactively.
* **Workbook**: It is the highest-level container in Tableau. It is a file that has the capacity to hold a number of worksheets, dashboards, and stories. The whole tableau project, including data connections and visuals, is stored in workbooks. They are the primary files used for creating, saving and sharing tableau projects. They store all the components required for data analysis and visualization.  
  Multiple worksheets, dashboards and tales can be organized in workbooks. At the workbook level, you can set up data source connections, define parameters and build computed fields.

### **58. Name the different products of Tableau with their significance.**

The different products of Tableau are as follows :

* [**Tableau Desktop**](https://www.geeksforgeeks.org/tableau-public-vs-tableau-desktop): It is the primary authoring and publishing tool. It allows data professionals to connect to various data sources, create interactive and shareable visualizations, and develop dashboards and reports for data analysis. Users can use the drag-and-drop interface to generate insights and explore data.
* **Tableau Server**: This is an enterprise-level platform tableau server that enables safe internal collaboration and sharing of tableau information. It manages access, centralizes data sources, and maintains data security. It is appropriate for bigger businesses with numerous users who require access to tableau content.
* **Tableau Online**: It is an online version of tableau. In a scalable and adaptable cloud environment, it enables users to publish, share, and collaborate on tableau content. For businesses searching for cloud-based analytics solutions without managing their infrastructure.
* [**Tableau Public**](https://www.geeksforgeeks.org/tableau-public-vs-tableau-desktop): It is a free version of tableau that enables users to create, publish and share dashboards and visualizations publicly on the web. The ability to share their data stories with a larger audience is perfect for data enthusiasts and educators.
* **Tableau Prep**: It is a tool for data preparation that makes it easier and faster to clean, shape, and combine data from diverse sources. Data specialists can save time and effort because it makes sure that the data is well-structured and ready for analysis.
* **Tableau Mobile**: A mobile application that extends tableau's capabilities to smartphones and tablets. By allowing users to access and interact with tableau content while on the go, it ensures data accessibility and decision-making flexibility.
* **Tableau Reader**: It is a free desktop application that enables users to view and interact with tableau workbooks and dashboards shared by the tableau desktop users. This tool is useful for those who require access to and exploration of tableau material without a tableau desktop license.
* **Tableau Prep Builder**: It is an advanced data preparation tool designed for data professionals. In order to simplify complicated data preparation operations, it provides more comprehensive data cleaning, transformation, and automation tools.

### **59. What is the difference between joining and blending in Tableau?**

In Tableau,[joining and blending](https://www.geeksforgeeks.org/data-blending-in-tableau) are ways for combining data from various tables or data sources. However, they are employed in various contexts and have several major differences:

| **Basis** | **Joining** | **Blending** |
| --- | --- | --- |
| **Data Source Requirement** | Joining is basically used when you have data from the same data source, such as a relational database, where tables are already related through primary and foreign keys. | Blending is used when we have data from different data sources. such as a combination of Excel spreadsheets, CSV files, and databases. These sources may not have predefined relationships. |
| **Relationships** | Foundation for joins is the use of common data like a customer ID or product code to establish predetermined links between tables. These relations are developed within same data source. | There is no need for pre-established links between tables while blending. Instead, you link different data sources separately and combine them by matching fields with comparable values. |
| **Data Combining** | When tables are joined, a single unified data source with a merged schema is produced. A single table with every relevant fields is created by combining the two tables. | Data blending maintains the separation of the data sources. At query time, tableau gathers and combines data from several sources to produce a momentary, in-memory blend for visualization needs. |
| **Data Transformation** | It is useful for data transformation, aggregations and calculations on the combined data. The information from many connected tables can be used to build computed fields. | It is only useful for data transformation and calculations. It cannot create calculated fields that involves data from different blended data sources. |
| **Performance** | Joins are more effective and quicker than blending because they leverage the database's processing power to perform the merge | It can be slower than joining because it involves querying and combining the data from the different sources at runtime. Large datasets in particular may have an impact on performance. |

### **60. What is the difference between a discrete and a continuous field in Tableau?**

In Tableau, fields can be classified as discrete or continuous, and the categorization determines how the field is utilized and shown in visualizations. The following are the fundamental distinctions between discrete and continuous fields in Tableau:

* **Discrete Fields:**They are designed for handling categorical or qualitative data such as names, categories, or labels. Each value within a discrete field represents a distinct category or group, with nor inherent order or measure associated with these values. Discrete fields are added to a tableau view and are identified by blue pill-shaped headers that are commonly positioned on the rows or column shelves. They successfully divide the data into distinct groups, generating headers for each division.
* **Continuous Fields:**They are designed for handling quantitative or numerical data, encompassing measurements, values, or quantities. Mathematical procedures like summation and averaging are possible because continuous fields have a natural order by nature. In tableau views, these fields are indicated by pill-shaped heads in a green color that are frequently located on the rows or columns shelf. Continuous fields when present in a view, represent a continuous range of value within the chosen measure or dimension.

### 61. **Explain the difference between live connections and extracts.**

In Tableau, There are two ways to attach data to visualizations: live connections and data extracts (also known as extracts). Here's a rundown of the fundamental distinctions between the two:

* **Live Connections:**Whether its a database, spreadsheet, online service or other data repository, live connections offers a real-time access to the data source. The visualizations always represent the most recent information available since they dynamically fetch data. When speed and current data are important, live connections are the best. However, they ca be demanding on the performance of the data source, as every interaction triggers a query to the source system. As a result, the responsiveness of the data source has a significant impact on how well live connections perform.
* **Extracts:**They involve producing and archiving a static snapshot of the original data in Tableau's exclusive .hyper format. Extracts can be manually or automatically renewed to allow for recurring updates. The ability of extracts to greatly improve query performance is what makes them unique. They are particularly useful for huge datasets or circumstances where the source system's performance may be subpar because they are optimized for quick data retrieval. Extracts are particularly helpful when building intricate, high-performing dashboards.

### **62. What Are the Different Joins in Tableau?**

Tableau allows you to make many sorts of joins to mix data from numerous tables or data sources. Tableau's major join types are:

* **Inner Join:** An inner join returns only the rows that have matching values in both tables. Rows that do not have a match in the other table are excluded from the result.
* **Left Join:** A left join returns all the rows from the left table and matching rows present in the right table. If there is no match in the right table, null values are included in the result.
* **Right Join:** A right join returns all the rows from the right table and matching rows present in the left table. If there is no match in the left table, null values are included.
* **Full Outer Join:** A full outer join returns all the rows where there is a match in either the left or right table. It includes all the rows from both tables and fills in null values where there is no match.

### **63. How can we create a calculated field in Tableau?**

You may use calculated fields in Tableau to make calculations or change data based on your individual needs. Calculated fields enable you to generate new values, execute mathematical operations, use conditional logic, and many other things. Here's how to add a calculated field to Tableau:

* Open the Tableau workbook or the data source.
* In the "data" pane on the left, right-click anywhere and choose "Create Calculated Field".
* In the calculated field editor, write your custom calculation using fields, functions, and operators.
* Click "OK" to save the calculated field.

### **64. What are the different data aggregation functions used in Tableau?**

Tableau has many different data aggregation functions used in tableau:

* SUM: calculates the sum of the numeric values within a group or partition.
* AVG: Computes the average of the numeric values.
* MIN: Determines the minimum value.
* MAX: Determines the maximum value.
* COUNT: Count the number of records or non-null values.
* VAR: Computes the variance of the sample population.
* VARP: Computes the variance of the entire population.
* STEDV: Compute the standard deviation of the sample population.
* STEDVP: Calculate the standard deviation of the entire population.

### **65. What is the Difference Between .twbx And .twb?**

**The Difference Between .twbx And .twb are as follows:**

* .twb: It represents a tableau workbook, focusing on the layout and visualization details created in the tableau desktop. It only contains the references to the location of the data source rather than the actual data itself. .twb files are less in size due to their lightweight nature. Recievers of .twb files must have access to the associated data source in order for the workbook to operate properly.
* .twbx: It is known as tableau packaged workbooks, provide a comprehensive solution for sharing tableau workbooks. They include both actual data source and the workbook layout, including any custom calculations and visualizations. This embedded data ensures that recipients can open and view the workbook independently of the original data source. However, .twbx files tend to be larger due to the included data.

### **66. What are the different data types used by Tableau?**

Tableau supports 7 variousvarious different data types:

* String
* Numerical values
* Date and time values
* Boolean values
* Geographic values
* Date values
* Cluster Values

### **67. What is a Parameter in Tableau?**

The parameter is a dynamic control that allows a user to input a single value or choose from a predefined list of values. In Tableau, dashboards and reports, parameters allow for interactivity and flexibility by allowing users to change a variety of visualization-related elements without having to perform substantial editing or change the data source.

### **68. What Are the Filters? Name the Different types of Filters available in Tableau.**

Filters are the crucial tools for data analysis and visualization in Tableau. Filters let you set the requirements that data must meet in order to be included or excluded, giving you control over which data will be shown in your visualizations.   
There are different types of filters in Tableau:

* **Extract Filter**: These are used to filter the extracted data from the main data source.
* **Data Source Filter**: These filters are used to filter data at the data source level, affecting all worksheets and dashboards that use the same data source.
* **Dimension Filter**: These filters are applied to the qualitative field and a non-aggregated filter.
* **Context Filter**: These filters are used to define a context to your data, creating a temporary subset of data based on the filter conditions.
* **Measure Filter**: These filters can be used in performing different aggregation functions. They are applied to quantitative fields.
* **Table Calculation Filter**: These filters are used to view data without filtering any hidden data. They are applied after the view has been created.

### **69. What are Sets and Groups in Tableau?**

The difference between Sets and Groups in Tableau are as follows:

* **Sets:**Sets are used to build custom data subsets based on predefined conditions or standards. They give you the ability to dynamically segment your data, which facilitates the analysis and visualization of particular subsets. Sets can be categorical or numeric and can be built from dimensions or measures. They are flexible tools that let you compare subsets, highlight certain data points, or perform real-time calculations. For instance, you can construct a set of "Hot Leads" based on the potential customers with high engagement score or create a set of high-value customers by choosing customers with total purchases above a pre-determined level. Sets are dynamic and adaptable for a variety of analytical tasks because they can change as the data does.
* **Groups:** Groups are used to combine people (dimension values) into higher level categories. They do this by grouping comparable values into useful categories, which simplifies complex data. Group members are fixed and do not alter as a result of the data since groups are static. Groups, which are typically constructed from dimensions, are crucial for classifying and labeling data points. For instance, you can combine small subcategories of product into larger categories or make your own dimension by combining different dimensions. Data can be presented and organized in a structed form using groups, which makes it easier to analyze and visualize.

### **70. Explain the different types of charts available in Tableau with their significance.**

Tableau offers a wide range of charts and different visualizations to help users explore and present the data effectively. Some of the charts in Tableau are:

* **Bar Chart:** They are useful for comparing categorical data and can be used show the distribution of data across categories or to compare value between categories.
* **Line Chart:**Line chart are excellent for showing trends and changes over time. They are commonly used for time series data to visualize how single measure changes over time.
* **Area Chart:** They are same as line chart but the area under the line is colored in area chart. They are used with different multiple variables in data to demonstrate the differences between the variables.
* **Pie Chart:**It shows parts of a whole. They are useful for illustrating the distribution of data where each category corresponds to a share of the total.
* **Tree Maps:** They show hierarchical data as nested rectangles. They are helpful for illustrating hierarchical structures, such as organizational or file directories.
* **Bubble chart:** Bubble charts are valuable for visualizing and comparing data points with three different attributes. They are useful when you want to show relationships, highlight data clusters, etc.
* **Scatter Plot:** They are used to display the relationship between two continuous variables. They help find correlations, clusters or outliers in the data.
* **Density Map**: Density maps are used to represent the distribution and concentration of data points or values within a 2D space.
* **Heat Map:** Heat maps are used to display data on a grid, where color represents values. They are useful for visualizing large datasets and identifying patterns.
* **Symbol Map:** Symbol maps are used to represent geographic data by placing symbols or markers on a map to convey information about specific locations.
* **Gannt Chart:** Gantt charts are used for project management to visualize tasks, their durations, and dependencies over time.
* **Bullet Graph:**They are used for tracking progress towards a goal. They provide a compact way to display a measure, target and performance ranges.
* **Box Plot(Box and Whisker) :**They are used to display the distribution of data and identify outliers. They show median, quartiles, and potential outliers.

### **71. How can you create a map in Tableau?**

The key steps to create a map in Tableau are:

* Open your tableau workbook and connect to a data source containing geographic information.
* Drag the relevant geographic dimensions onto the "Rows" and "Columns" shelves.
* Use a marks card to adjust marker shapes, colour and sizes. Apply size encoding and color based on the data values.
* Add background images, reference lines, or custom shapes to enhance the map, optionally.
* Save and explore your map by zooming, panning and interacting with map markers. Use it to analyze the spatial data, identify trends and gain insights from the data.

### **72. How can we create a doughnut chart in Tableau?**

The key steps to create a doughnut chart in tableau:

* Open the Tableau desktop and connect to the data source.
* Go to the sheet and in the marks card, select a pie chart with categories and values. Drag the dimensions and measure in the "column" and "row" shelf, respectively.
* Duplicate the sheet, in the new sheet right click on the "axis" on the left side of the chart and select "Dual Axis" chart.  
  On the right axis, right click on the axis and select "edit axis". In edit axis, set the "Fixed" range for both minimum and maximum to be the same and click ok.
* Now, right click on both axes and select "Synchronize Axis" to make sure that both pie charts share the same scale.
* Create a circle on the second chart by dragging dimensions to Rows in second chart and remove all labels and headers to make it a blank circle.
* Select the "Circle" chart in the second chart and set the opacity in the marks card to be 0% to make circle transparent.
* In the marks card. set the "color" to white or transparent and adjust the size of the circle as needed to create the desired doughnut hole.  
  Customize the colors and labels for both pie charts to make them visually attractive and informative.

### **73. How can we create a Dual-axis chart in Tableau?**

The key steps to create a dual-axis chart in tableau are as follows:

* Connect with the data source. Create a chart by dragging and dropping the dimension and measure into "column" and "rows" shelf, respectively.
* Duplicate the chart by right click on the chart and select "Duplicate". This will create the duplicate of the chart.
* In the duplicated chart, change the measure you want to display by dragging the new measure to the "columns" or "rows" shelf, replacing the existing measure.
* In the second chart, assign the measure to different axis by clicking on the "dual-axis". This will create two separate axes on the chart.
* Right click on one of the axes and select "synchronize axis". Adjust formatting, colors and labels as needed. You now have a dual-axis chart.

### **74. What is a Gantt Chart in Tableau?**

A Gantt Chart has horizontal bars and sets out on two axes. The tasks are represented by Y-axis, and the time estimates are represented by the X-axis. It is an excellent approach to show which tasks may be completed concurrently, which needs to be prioritized, and how they are dependent on one another.  
Gantt Chart is a visual representation of project schedules, timelines or task durations. To illustrate tasks, their start and end dates, and their dependencies, this common form of chat is used in project management. Gantt charts are a useful tool in tableau for tracking and analyzing project progress and deadlines since you can build them using a variety of dimensions and measures.

### **75. What is the Difference Between Treemaps and Heat Maps?**

The Difference Between Treemaps and Heat Maps are as follows:

| **Basis** | **Tree Maps** | **Heat Maps** |
| --- | --- | --- |
| **Representation** | Tree maps present hierarchical data in a nested, rectangular format. The size and color of each rectangle, which each represents a category or subcategory, conveys information. | Heat maps uses color intensity to depict values in a grid. They are usually used to depict the distribution or concentration of data points in a 2D space. |
| **Data Type** | They are used to display hierarchical and categorical data. | They are used to display continuous data such as numeric values. |
| **Color Usage** | Color is frequently used n tree maps to represent a particular attribute or measure. The intensity of the color can convey additional information. | In heat maps, values are typically denoted by color intensity. Lower values are represented by lighter colors and higher values by brighter or darker colors. |
| **Interactivity** | It is possible for tree maps to be interactive, allowing users to click on the rectangle to uncover subcategories and drill down into hierarchical data. | Heat maps can be interactive, allowing users to hover over the cells to see specific details or values. |
| **Use Case** | They are used for visualizing organizational structures, hierarchical data and categorical data. | They are used in various fields like finance, geographic data, data analysis, etc. |

### **76. What is the blended axis in Tableau?**

If two measures have the same scale and share the same axis, they can be combined using the blended axis function. The trends could be misinterpreted if the scales of the two measures are dissimilar.  
  
**77. What is the Level of Detail (LOD) Expression in Tableau?**

A Level of Detail Expression is a powerful feature that allows you to perform calculations at various levels of granularity within your data visualization regardless of the visualization's dimensions and filters. For more control and flexibility when aggregating or disaggregating data based on the particular dimensions or fields, using LOD expressions.  
There are three types of LOD:

* Fixed LOD: The calculation remains fixed at a specified level of detail, regardless of dimensions or filters in the view.
* Include LOD: The calculation considers the specified dimensions and any additional dimensions in the view.
* Exclude LOD: The calculation excludes the specified dimensions from the view's context.

### **78. How to handle Null, incorrect data types and special values in Tableau?**

Handling null values, erroneous data types, and unusual values is an important element of Tableau data preparation. The following are some popular strategies and recommended practices for coping with data issues:

* For Handling Null values:  
  You can filter out the null values in specified field by right clicking on the field and choosing "Filter". Then exclude null values in the filter options.  
  Using the 'ZN()' or 'IFNULL()' functions in the calculated fields to replace null values.
* For incorrect data types:  
  Modify data types in the data pane, use calculated fields or use tableau's data interpreter.
* For special Values:  
  Use data transformations tools like split, replace, etc., using calculated fields or data blending to handle special values.

### **79. How can we create a Dynamic webpage in Tableau?**

To create dynamic webpages with interactive tableau visualizations, you can embed tableau dashboard or report into a web application or web page. It provides embedding options and APIs that allows you to integrate tableau content into a web application.  
Following steps to create a dynamic webpage in tableau:

* Go to the dashboard and click the webpage option in the 'Objects'.
* In the dialog box that displays, don't enter a URL and then click 'OK'.
* choose 'Action' by clicking on the dashboard menu. Click on the 'Add Action' in action and select 'Go to URL' .
* Enter the 'URL' of the webpage and click on the arrow next to it. Click 'OK'.

### **80. What are the KPI or Key Performance Indicators in Tableau?**

Key Performance Indicators or KPI are the visual representations of the significant metrics and performance measurements that assist organizations in monitoring their progress towards particular goals and objectives. KPIs offer a quick and simple approach to evaluate performance, spot patterns, and make fact-based decisions.

### **81. what is a context filter in Tableau?**

Context filter is a feature that allows you to optimize performance and control data behavior by creating a temporary data subset based on a selected filter. When you designate a filter as a context filter, tableau creates a smaller temporary table containing only the data that meets the criteria of that particular filter. This decrease in data capacity considerably accelerates processing and rendering for visualization, which is especially advantageous for huge datasets. When handling several filters in a workbook, context filters are useful because they let you select the order in which filters are applied, ensuring a sensible filtering process.

### **82.How can you create a dynamic title in a Tableau worksheet?**

You can create a dynamic title for a worksheet by using parameters, calculated fields and dashboards. Here are some steps to achieve this:

* Creating a Parameter: Go to data pane, right click on it and select "Create Parameter". Choose the data type for the parameter. For a dynamic title, yo can choose "string" or "integer". Then define the allowable values for the parameter. You can choose all values or some specific values.
* Create a calculated field: Now create a calculated field that will be used to display the dynamic title. You can use the parameters in the calculated field to create a dynamic title. Create a new worksheet. Drag and drop the calculated field you created in the "Title" shelf of the worksheet.
* Create a Dashboard: Go to the "dashboard" and add a parameter control and connect it to the worksheet and then select parameter control in the dashboard. This will allow the parameter control to change parameter value dynamically.   
  Now, whenever you will interact with the parameter control on the dashboard, the title of the worksheet will update based on the parameter's value.

### **83. What is data source filtering, and how does it impact performance?**

Data Source filtering is a method used in reporting and data analysis applications like Tableau to limit the quantity of data obtained from a data source based on predetermined constraints or criteria. It affects performance by lowering the amount of data that must be sent, processed, and displayed, which may result in a quicker query execution time and better visualization performance. It involves applying filters or conditions at the data source level, often within

the SQL query sent to the database or by using mechanisms designed specially for databases.  
Impact on performance:   
Data source filtering improves performance by reducing the amount of data retrieved from the source. It leads to faster query execution. shorter data transfer times, and quick visualization rendering. by applying filters based on criteria minimizes resource consumption and optimizes network traffic, resulting in a more efficient and responsive data analysis process.

### **84. How do I link R and Tableau?**

To link R and Tableau, we can use R integration features provided by Tableau. Here are the steps to do so:

* Install R and R Integration Package:  
  we have to install R on the computer. Then install the "RServe" package by using "Install.packages("Rserve")". Open R and load the RServe library and start running it.
* Connect Tableau to R:   
  Open the tableau desktop and go to "Help" menu. Select "settings and performance" then select "Manage External service connection".   
  In the "External Service" section , select "R integration".   
  Specify the R server details, such as host, port and any necessary authentication credentials. Test the connection to ensure its working properly.

### **85. How do you export Tableau visualizations to other formats, such as PDFs or images?**

Exporting tableau visualizations to other formats such as PDF or images, is a common task for sharing or incorporating your visualizations into reports or presentations. Here are the few steps to do so:

* Open the tableau workbook and select the visualization you want to export.
* Go to the "File" menu, select "Export".
* After selecting "Export" a sub menu will appear with various export options. Choose the format you want to export to. (PDF, image, etc.,)
* Depending on the chosen export format, you may have some configuration options that you can change according to the needs.
* Specify the directory or the folder where you want to save the exported fie and name it.
* Once the settings are configured, click on "save" or "Export".

## 5. Data Visualization Interview Questions

<https://www.geeksforgeeks.org/data-visualization-interview-questions/>

(Last Updated : 18 Dec, 2024)

## Q.1 What is data visualization, and why is it important?

Data visualization is the graphical representation of data to help individuals, organizations, and analysts to better understand patterns, trends, and insights within the data. It involves the use of visual elements like charts, graphs, maps, and infographics to convey complex information in a more accessible and comprehensible format.

## Q.2 What are the key components of good data visualization?

Effectively communicating knowledge and insights while being simple to understand and aesthetically beautiful are all qualities of successful data visualization. A strong data visualization should have the following critical elements:

1. Data Accuracy
2. Clear and Relevant Title
3. Appropriate Visual Representation
4. Data Labels and Legends
5. Consistent Scale and Units.

## Q.3 How can color be utilized in data visualization?

In data visualisation, colour is a potent tool that can improve comprehension, draw attention to patterns, and effectively communicate ideas. When applied carefully, colour may increase the interest and clarity of your data visualisation. Following are some examples of how colour can be used in data visualisation:

1. Differentiating Categories or Groups
2. Highlighting Data Points or Trends
3. Gradient Scales
4. Colour Coding for Meaning
5. Colour Legends and Labels

## Q.4 What are the different types of data visualizations?

Data visualisations come in a variety of forms, each of which is intended to effectively communicate a particular type of knowledge and insight. Here are a few examples of prevalent data visualisations:

* **Bar Charts:**Bar charts use rectangular bars to represent data values, making them suitable for comparing data across categories or groups.
* **Line Charts:** Line charts display data points connected by lines, making them useful for showing trends and changes over time.
* **Scatter Plots:**Scatter plots use individual data points to display the relationship between two continuous variables, making them helpful for identifying correlations or patterns.
* **Pie Charts:** Pie charts represent parts of a whole, with each slice of the pie corresponding to a percentage or proportion of the total.
* **Histograms:** Histograms display the distribution of a single variable's values, showing how data is distributed across different bins or intervals.
* **Box Plots:** Box plots provide a summary of the distribution of data, including measures such as the median, quartiles, and potential outliers.
* **Heatmaps:** Heatmaps use color to represent data values in a grid, making them suitable for visualizing correlations or patterns in large datasets.
* **Treemaps:** Treemaps represent hierarchical data structures, such as the organization of files on a computer, using nested rectangles.
* **Sankey Diagrams:** Sankey diagrams illustrate the flow or distribution of data between categories or entities, often used in energy or resource analysis.
* **Bubble Charts:**Bubble charts extend scatter plots by using bubbles of varying sizes to represent data points, with the size of the bubble indicating an additional variable.
* **Choropleth Maps:** Choropleth maps use color-coding to represent data values in geographic regions, making them useful for visualizing regional data.
* **Parallel Coordinates Plots:** Parallel coordinates plots visualize multivariate data by representing each data point as a line crossing parallel axes.
* **Waterfall Charts:**Waterfall charts display incremental changes in data values, commonly used for financial or budget analysis.
* **Radar Charts (Spider Charts):** Radar charts display data points on a circular grid, making them useful for comparing multiple variables across different categories.
* **Network Diagrams:**Network diagrams illustrate relationships between entities in a network, such as social networks or transportation systems.
* **Word Clouds:** Word clouds visually represent the frequency of words in a text, with more frequently occurring words displayed in larger text.
* **Bullet Graphs:**Bullet graphs provide a compact way to display a single data point in relation to a target or benchmarks, often used in dashboards.
* **Sunburst Charts:** Sunburst charts display hierarchical data in a radial layout, with segments representing parent and child categories.
* **3D Plots:**3D plots add a third dimension to 2D plots, allowing for the visualization of data in three-dimensional space.

These are just some of the data visualization types. The choice of visualization method depends on the nature of the data, the goals of the analysis, and the audience's needs for understanding the information presented.

## Q.5 What is a bar chart, and when it is typically used for data visualization?

A bar chart, also called a bar graph, is a tool for data visualisation. Each bar in a bar chart is proportional to the value it displays in terms of height or length. The bars are normally aligned along an axis either horizontally or vertically.

Here are some of the main key components of a bar chart.

1. Bars: These are the rectangular elements that visually represent the data values. The length or height of each bar corresponds to the magnitude of the data it represents.
2. Axes: A bar chart usually has two axes: a vertical or y-axis (on the left or bottom) and a horizontal or x-axis (on the bottom or left). The y-axis typically represents the data values, while the x-axis represents categories or data points.
3. Labels: The axes are labeled to indicate the scale and the categories being represented. The bars may also have data labels or values at their endpoints.

Bar charts are typically used for the following purposes in data visualization:

1. Comparing Categories
2. Displaying Discrete Data
3. Showing Rankings
4. Tracking Changes Over Time
5. Part-to-Whole Relationships

## Q.6 Define outliers and discuss potential methods for handling them.

Outliers are the data point that significantly different from the rest of the data points. Outliers can occur for various reasons, including data entry errors, measurement errors, natural variation, or the presence of rare events. Identifying and handling outliers is important in data analysis because they can have a significant impact on statistical analyses and machine learning models.

***Here are some methods for handling outliers:***

* ***Data Trimming***
* ***Data Transformation***
* ***Robust Statistical Methods***
* ***Machine Learning Models***
* ***Visualization***
* ***Ensemble Methods***

## Q.7 How do you choose the appropriate visualization type for your data?

It is important to carefully analyse the nature of the data, the objectives of the research, and the audience you're attempting to reach before selecting the right visualisation method for your data. Here is a step-by-step tutorial to assist you in selecting the best option:

* Understand Your Data
* Identify Your Goals
* Consider Your Audience
* Choose the Right Chart Type
* Document and Explain

## ****Q.8 What is the importance of storytelling in data visualization?****

Storytelling is a crucial aspect of data visualization because it transforms raw data into a compelling narrative that can inform, persuade, and engage the audience. Here are several reasons why storytelling is important in data visualization.

* Contextualization
* Clarity and Comprehension
* Engagement
* Emotional Connection
* Memory Retention
* Decision-Making

## Q.9 How can you choose an appropriate color palette for your visualizations?

Choosing an appropriate color palette for our visualizations is crucial for ensuring clarity, readability, and effective communication of data. Here's a step-by-step guide on how to choose a suitable color palette:

1. Understand the Data and Context
2. Consider Color Meaning and Symbolism
3. Ensure Accessibility
4. Start with a Base Color
5. Select Additional Colors

## Q.10 What are some common mistakes to avoid when creating data visualizations?

Creating effective data visualizations requires careful attention to detail and thoughtful design choices. Here are some common mistakes to avoid when creating data visualizations:

* **Misleading Scaling:**Misrepresenting the scale of axes or using inconsistent scales can distort the data and lead to incorrect interpretations. Ensure that scales accurately reflect the data.
* **Incomplete or Missing Labels:** Labels on axes, data points, and legends are essential for context. Missing or incomplete labels can confuse viewers and hinder understanding.
* **Overloading with Data:** Avoid cluttering your visualization with too much information. Overloading with data points, labels, or details can overwhelm the audience and reduce clarity.
* **Non-Zero Baseline for Bar Charts:**When using bar charts, make sure the baseline starts at zero. Truncated axes can exaggerate differences and mislead viewers.
* **Ignoring Data Outliers:** Ignoring or mishandling outliers in your visualization can lead to skewed perceptions of the data. Consider whether to address or mention outliers, depending on their relevance.
* **Inadequate Data Cleaning:** Failure to clean and preprocess data before visualization can result in inaccuracies and visual artifacts. Ensure data quality and consistency.

## Q.11 How can you assess the effectiveness of data visualization?

Assessing the effectiveness of data visualization involves evaluating how well it achieves its intended goals, communicates insights, and engages the audience. Here are several methods and considerations for assessing the effectiveness of your data visualization:

* Clearly Defined Objectives
* Audience Feedback
* [Usability Testing](https://www.geeksforgeeks.org/usability-testing/)
* Objective Metrics
* Comparative Analysis

## Q.13 Describe the concept of data-ink ratio in data visualization.

The concept of the data-ink ratio is a principle introduced by Edward Tufte, a prominent expert in data visualization. It emphasizes the idea that in a data visualization, every piece of ink or pixel used to represent data should contribute directly to the audience's understanding of the information. In other words, unnecessary ink or non-data ink should be minimized to maximize the efficiency and clarity of the visualization.

Here are key components and principles related to the data-ink ratio:

* Data-Ink
* Non-Data Ink
* Maximizing Data-Ink
* Simplicity and Clarity
* Enhancing Readability

## Q.14 What is the purpose of a legend in a chart or graph?

A chart or graph's legend serves as a guide or explanation for the different data series or components displayed in the visualisation. It aids the viewer in comprehending the significance of the many hues, symbols, or lines used to represent various data categories, variables, or groupings in the chart or graph.

## Q.15 What is a pie chart, and when is it suitable for visualizing data?

The circular data visualisation tool known as a pie chart shows data as a segmented circle, with each segment (or "slice") denoting a certain category or percentage of the overall data. Each segment's size is proportionate to the amount or percentage it contributes to the dataset. In situations when the categories are distinct and do not follow a logical order, pie charts are frequently used to depict categorical or nominal data.

**When to Use Pie Charts:**

* Showing Part-to-Whole Relationships
* Comparing Categories
* Highlighting Percentages
* Simple Data Structures
* Visual Appeal

## Q.16 Explain the main elements of a pie chart.

A pie chart consists of several main elements that work together to visually represent data as a circular graph. Understanding these elements is essential for interpreting and creating pie charts effectively. Here are the key components of a pie chart:

1. Circle (or Pie)
2. Slices (Segments)
3. Central Angle
4. Category Labels
5. Data Labels
6. Legend
7. Title
8. Exploded or Offset Slices
9. Colors
10. Lines or Leader Lines

## Q.17 What is a line chart, and when is it commonly employed for data visualization?

A style of data visualisation called a line chart shows data points connected by straight lines. It is especially useful for identifying trends, patterns, and relationships in time-series data since it is frequently used to represent data that changes continuously over a predetermined period or sequence. Line graphs are another name for line charts.

Common Use Cases for Line Charts:

* Time-Series Data
* Trend Analysis
* Comparing Multiple Data Series
* Forecasting
* Performance Metrics
* Scientific Data
* Economic and Financial Data
* Population and Demographic Trends

## Q.18 Describe the components of a line chart.

A line chart consists of several components that work together to visually represent data and convey trends or patterns effectively. Understanding these components is essential for interpreting and creating line charts. Here are the key components of a typical line chart:

1. Title
2. X-Axis (Horizontal Axis)
3. Y-Axis (Vertical Axis)
4. Axis Labels
5. Data Points

## Q.19 What is a scatter plot, and under what circumstances would you use it for data visualization?

Individual data points can be seen on a two-dimensional graph using a technique called a scatter plot. The values of two variables, one depicted on the horizontal (X) axis and the other on the vertical (Y) axis, are represented by each data point on the scatter plot. The relationship, correlation, or dispersion of data points between two variables can be visualised using scatter plots.

**Characteristics of Scatter Plots:**

* Two Variables
* Data Points
* No Connecting Lines
* Variable Scales

## Q.20 Explain the key elements of a scatter plot.

A scatter plot consists of several key elements that work together to visually represent the relationship between two variables. Understanding these elements is essential for interpreting and creating scatter plots effectively. Here are the key components of a typical scatter plot:

* Title
* X-Axis (Horizontal Axis)
* Y-Axis (Vertical Axis)
* Axis Labels
* Data Points

## Q.21 What is a histogram, and when is it employed for data visualization?

A histogram is a graph that shows how a dataset is distributed. It shows the frequency or count of data points along a continuous range that fall into predetermined intervals or "bins". Histograms are frequently used to visualise the frequency and distribution of numerical data, which makes them very helpful for examining trends and traits in datasets.

**Common Use Cases for Histograms:**

* Data Distribution Analysis
* Frequency Count
* Outlier Detection
* Data Transformation
* Quality Control
* [Statistical Analysis](https://www.geeksforgeeks.org/statistics/)

## Q.22 Describe the essential features of a histogram.

A histogram is a graphical representation of the distribution of a dataset, displaying the frequency or count of data points within specified intervals or "bins" along a continuous range. To understand and interpret a histogram effectively, it's important to be familiar with its essential features. Here are the key components and features of a histogram:

* Bins or Intervals
* Frequency or Count
* Continuous Scale

## Q.23 What is a heatmap, and when is it useful for data visualization?

A heatmap is a data visualization technique that uses colors to represent the values of a matrix or a table of data. It is particularly useful for visualizing patterns, relationships, and variations in data, especially when dealing with large datasets or data organized in a two-dimensional format. Heatmaps are versatile and can be applied to various types of data analysis.

* Common Use Cases for Heatmaps
* Genomic Data Analysis
* Website User Behavior
* Financial Data Analysis
* Sports Analytics

## Q.24 Explain the primary components of a heatmap.

A heatmap is a data visualization that uses color to represent the values of a matrix or a table of data. It consists of several primary components that work together to convey information effectively. Understanding these components is crucial for interpreting and creating heatmaps. Here are the primary components of a heatmap:

1. Color Scale
2. Matrix of Data
3. Row Labels and Column Labels
4. X-Axis and Y-Axis
5. Color Legend

## Q.25 What is a box plot and why is it used for data visualization?

A box plot, also known as a box-and-whisker plot, is a graphical representation of a dataset's distribution and central tendency. It is used to visualize the spread, variability, and potential outliers within the data. Box plots are particularly useful for comparing multiple datasets or identifying patterns in a single dataset.

**Reasons for Using Box Plots**

* Summary of Data Distribution
* Comparison of Distributions
* Identification of Skewness
* Detection of Outliers
* Robustness to Extreme Values
* Statistical Insights

## Q.26 Explain the differences between descriptive and inferential statistics.

Descriptive statistics and inferential statistics are two branches of statistics used to analyze and interpret data. They serve different purposes and employ distinct methods. Here are the key differences between descriptive and inferential statistics:

| **Function** | **Descriptive Statistics** | **inferential statistics** |
| --- | --- | --- |
| Purpose | Descriptive statistics are used to summarize, describe, and present data in a meaningful and understandable way. | Inferential statistics are used to make inferences, predictions, or generalizations about a population based on a sample of data. |
| Data Usage | Descriptive statistics focus on the data that are available and provide a summary of these data. | Inferential statistics use sample data to make inferences about a larger population. |
| Methods | Descriptive statistics use various measures and techniques to describe the characteristics of data. | Inferential statistics involve hypothesis testing, confidence intervals, regression analysis, and various statistical tests. |

## Q.27 What is the purpose of a box plot in statistics visualization.

A box plot, commonly referred to as a box-and-whisker plot, is a graphical representation used in statistics to show summary statistics, such as measures of central tendency and spread, and to visualise the distribution of a dataset.

## Q.28 When is a quantile-quantile (Q-Q) plot used in statistics, and how does it help assess the normality of a dataset?

A Quantile-Quantile (Q-Q) plot is a statistical visual aid for evaluating the[normality](https://www.geeksforgeeks.org/normality/) or closeness of a dataset's distribution to a theoretical normal distribution. When determining if your dataset follows a normal (Gaussian) distribution or any other particular distribution, it is especially helpful.

Here's how a Q-Q plot works and how it helps assess the normality of a dataset:

1. Basic Concept
2. Procedure
3. Interpretation
4. Assessing Normality
5. Outliers

## Q.29 What is a heat map, and how is it useful for visualizing correlations and patterns in a matrix of data in statistics?

A heatmap is a type of graphic that uses colour to show a data matrix's values. When dealing with numerical or categorical data structured in a matrix or table, heatmaps are extremely helpful for visualising relationships and patterns within huge datasets. For the following reasons, they are frequently used in statistics, data analysis, and data visualisation:

* Correlation Analysis
* Pattern Recognition
* Data Comparison
* Hierarchical Clustering
* Anomaly Detection
* Decision-Making

## Q.30 Describe the purpose of a violin plot in statistics visualization.

A violin plot is a data visualisation technique used in statistics to show the distribution of a dataset and reveal both its underlying probability density function (PDF) and summary statistics. Its major objective is to combine elements of a kernel density plot and a box plot, providing a more thorough understanding of the data distribution. A violin plan has the following objectives and elements:

## Q.31 What is univariate data visualization, and why is it important in data analysis?

A approach for exploring and displaying the distribution and properties of a single variable or one-dimensional dataset is called univariate data visualisation. Without taking into account its relationships with other variables, univariate data visualisation focuses on helping you comprehend the characteristics and patterns of a single variable. Data analysis requires this kind of visualisation for a number of reasons:

* Data Exploration
* Summary Statistics
* Identifying Patterns and Trends
* Outlier Detection
* Distribution Assessment
* Variable Transformation

## Q.32 Describe the purpose of a density plot in univariate data visualization.

The probability density function (PDF) of a continuous variable can be calculated and displayed using a density plot, sometimes referred to as a kernel density plot. Its main objective is to visualise the distribution of a single variable and reveal information about the underlying data distribution. The function and properties of a density plot in univariate data visualisation are described as follows:

1. Estimation of Probability Density
2. Smoothed Curve
3. Visualizing Distribution Shape
4. Complementing Histograms
5. Probability and Relative Likelihood

## Q.33 What are the different plots used for univaraite analysis

Univariate analysis focuses on exploring and summarizing a single variable at a time. There are several common types of plots and visualizations used in univariate analysis to gain insights into the distribution and characteristics of a single variable. Here are some of the most commonly used univariate plots:

1. Histogram
2. Box Plot (Box-and-Whisker Plot)
3. Density Plot (Kernel Density Plot)
4. Bar Chart
5. Frequency Plot
6. Pie Chart
7. Dot Plot
8. Violin Plot
9. Time Series Plot
10. Probability Plot (Q-Q Plot)

## Q. 34 What is bubble chart?

A bubble chart is a data visualization technique used to display three-dimensional data in a two-dimensional space. It is an extension of a scatter plot, where each data point is represented as a circle (or "bubble") on a two-dimensional coordinate system, with the size of the circle indicating a third variable.

## Q.35 What is a grouped bar chart?

A grouped bar chart, also known as a clustered bar chart, is a type of data visualization used to display and compare data for multiple categories or groups across two or more subcategories or variables. It is an extension of a standard bar chart, where bars are grouped together to show the relationships between multiple sets of data within each category or group.

## Q.36 Explain the importance of data visualization in statistics.

Data visualization is a vital component of statistics that enhances data exploration, communication, and decision-making. It transforms raw data into actionable insights, making statistics more accessible and impactful in various domains. Effective visualization can lead to better-informed decisions and a deeper understanding of data patterns and relationships.

## Q.37 What are some common methods for visualizing correlations between variables?

Visualizing correlations between variables is essential for understanding relationships and dependencies in data. Several common methods for visualizing correlations between variables include:

* Scatter Plots
* Correlation Matrix Heatmap
* Correlation Matrix Dendrogram
* Scatterplot Matrix
* Line Plots with Multiple Variables
* Bubble Charts
* Correlograms
* Parallel Coordinates Plot

## Q.38 How can you determine if a dataset follows a normal distribution using visualizations?

To determine if a dataset follows a normal distribution using visualizations, you can use various graphical tools and techniques to assess the distribution's shape and characteristics. While visual inspection is not a formal statistical test for normality, it can provide valuable insights. Here's how you can use visualizations to assess normality:

1. Histogram
2. Probability Plot (Q-Q Plot)
3. Normal Probability Plot (P-P Plot)
4. Kernel Density Plot
5. Box Plot

## Q.39 What is the key advantage of using a logarithmic scale in a visualization?

The key advantage of using a logarithmic scale in a visualization is its ability to effectively represent and visualize data that spans a wide range of values or exhibits exponential growth or decay.

## Q.40 When would you choose a bar chart over a pie chart for displaying categorical data?

Choosing between a bar chart and a pie chart for displaying categorical data depends on the nature of the data and the specific message you want to convey. Here are some situations in which you would prefer a bar chart over a pie chart:

* Comparing Categories
* Showing Relative Magnitudes
* Handling Many Categories
* Displaying Ranking
* Showing Trends Over Time

## Q.41 What is the primary difference between a line chart and a scatter plot.

A line chart connects data points with lines and is ideal for visualizing trends or changes in data over a continuous scale or time. It is commonly used for time-series data, such as stock prices or temperature variations.

Scatter Plot: A scatter plot represents individual data points as unconnected dots, making it suitable for showing the relationship or correlation between two continuous variables. It helps identify patterns, clusters, or outliers in the data.

Key Difference: The primary distinction is that a line chart emphasizes connected data points to depict trends, while a scatter plot displays unconnected data points to reveal relationships between two variables without assuming a specific sequence.

## Q.42 What does the term "overplotting" mean in the context of scatter plots?

"overplotting" refers to a situation where multiple data points on the plot overlap or occupy the same or nearly the same position on the graph. Overplotting can occur when you have a large number of data points or when the data values are tightly clustered, making it difficult to discern individual points.

## Q.43 Why is it important to consider colorblindness when designing visualizations?

Considering colorblindness in visualization design is essential for inclusivity, effective communication, and avoiding misinterpretation. Approximately 8% of the population has some form of color vision deficiency, so using distinguishable color palettes, adding labels and annotations, and providing alternative representations ensures that visualizations are accessible and informative to a broader audience. Testing for accessibility and promoting awareness of colorblindness are also crucial steps in creating inclusive visualizations.

## Q.44 What is the purpose of jitter in scatter plots?

The purpose of jitter in scatter plots is to add a small amount of random noise or displacement to the data points along one or both axes. This is done to prevent overplotting, which occurs when multiple data points share the same or very close coordinates, making it difficult to discern individual points.

## Q.45 Explain the concept of a "word cloud" in text data visualization.

A "word cloud" is a text data visualization technique used to represent the frequency or importance of words in a given text or document. In a word cloud, words are displayed graphically, and their size or prominence is determined by their frequency or significance within the text. The more frequently a word appears in the text, the larger and more prominent it appears in the word cloud.

## Q.46 What is the significance of word size and color in a word cloud?

Word size and color in a word cloud serve as effective visual cues to highlight word frequency, importance, and categorical information. When used appropriately, they enhance the readability and informativeness of the word cloud, aiding in the quick understanding of key insights within the textual data.

## Q.47 How can you address the issue of word overlap or crowding in a word cloud?

The significance of word size and color in a word cloud lies in their role in visually representing the importance or prominence of words within a given text or dataset. These visual attributes are essential for conveying information and insights in a word cloud.

## Q.48 What are the main limitations of using word clouds for text analysis?

Loss of Context: Word clouds don't capture the context in which words appear, leading to a loss of critical information and nuances in meaning.

Limited Vocabulary: They display only the most frequent words, excluding potentially meaningful terms, resulting in a biased representation.

Equal Treatment of Words: All words are treated equally, regardless of their importance or relevance, which can be misleading and overlook significant terms.

## Q.49 What is the difference between a word cloud and a tag cloud?

Word Cloud: Emphasizes word frequency in a given text, with word size based on frequency, typically used for exploratory analysis.

Tag Cloud: Displays keywords or tags associated with a collection of content, with tag size and style reflecting importance or relevance within a specific context, often used in information retrieval systems.

Visual Cues: Word clouds use minimal visual cues, while tag clouds may incorporate color and interactivity to convey context-specific information and enable user interactions.

## Q.50 What are some alternatives to word clouds for visualizing text data?

1. Bar Charts and Histograms
2. Word Frequency Tables
3. Heatmaps
4. Word Cloud Variations
5. Topic Modeling

## 6. Data Engineering Interview Questions

# <https://www.geeksforgeeks.org/data-engineer-interview-questions/>

# (Top 60+ Data Engineer Interview Questions and Answers

Last Updated : 07 Apr, 2025)

### 1. What is data engineering?

Data engineering is the practice of designing, building, and maintaining systems for collecting, storing, and analyzing large volumes of data. It involves creating data pipelines, optimizing data storage, and ensuring data quality and accessibility for [data scientists](https://www.geeksforgeeks.org/data-scientist-roadmap/) and analysts.

### 2. What are the main responsibilities of a data engineer?

The main responsibilities of a data engineer include:

* Designing and implementing data pipelines
* Creating and maintaining data warehouses
* Ensuring data quality and consistency
* Optimizing data storage and retrieval systems
* Collaborating with data scientists and analysts to support their data needs
* Implementing data security and governance measures

### 3. What is the difference between a data engineer and a data scientist?

While both roles work with data, their focus and responsibilities differ:

* Data engineers primarily deal with the infrastructure and systems for data management, ensuring data is accessible, reliable, and efficient to use.
* Data scientists focus on analyzing data, creating models, and extracting insights to solve business problems.

### 4. What is a data pipeline?

A data pipeline is a series of processes that move data from various sources to a destination system, often involving transformation and processing steps along the way. It ensures that data flows smoothly from its origin to where it's needed for analysis or other purposes.

### 5. What are some common challenges in data engineering?

Common challenges in data engineering include:

* Handling large volumes of data efficiently
* Ensuring data quality and consistency
* Managing real-time data processing
* Scaling systems to accommodate growing data needs
* Integrating diverse data sources and formats
* Maintaining data security and privacy

## *Data Engineer Interview Questions on Database Systems and SQL*

### 6. What is a relational database?

A relational database is a type of database that organizes data into tables with predefined relationships between them. It uses SQL (Structured Query Language) for managing and querying the data.

### 7. What are the main differences between SQL and NoSQL databases?

 A: Key differences include:

* Structure: SQL databases use a structured schema, while NoSQL databases are schema-less or have a flexible schema.
* Scalability: NoSQL databases are generally more scalable horizontally, while SQL databases often scale vertically.
* Data model: SQL databases use tables and rows, while NoSQL databases can use various models like document, key-value, or graph.
* ACID compliance: SQL databases typically provide ACID guarantees, while NoSQL databases may sacrifice some ACID properties for performance and scalability.

### 8. What is normalization in database design?

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves breaking down larger tables into smaller, more focused tables and establishing relationships between them.

### 9. Explain the concept of database indexing.

Database indexing is a technique used to improve the speed of data retrieval operations. It creates a data structure that allows the database to quickly locate specific rows based on the values in one or more columns, without having to scan the entire table.

### 10. What is a stored procedure?

A stored procedure is a precompiled collection of SQL statements that are stored in the database and can be executed with a single call. They can accept parameters, perform complex operations, and return results, improving performance and code reusability.

## *****Data Engineer Interview Questions on Big Data Technologies*****

### 11. What is Hadoop?

Hadoop is an open-source framework designed for distributed storage and processing of large datasets across clusters of computers. It consists of two main components: the Hadoop Distributed File System (HDFS) for storage and MapReduce for processing.

### 12. Explain the concept of MapReduce.

MapReduce is a programming model and processing technique for distributed computing. It consists of two main phases:

* Map: Divides the input data into smaller chunks and processes them in parallel
* Reduce: Aggregates the results from the Map phase to produce the final output

### 13. What is Apache Spark?

Apache Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets.

### 14. How does Spark differ from Hadoop MapReduce?

### A: Key differences include:

* Speed: Spark is generally faster due to in-memory processing
* Ease of use: Spark offers more user-friendly APIs in multiple languages
* Versatility: Spark supports various workloads beyond batch processing, including streaming and machine learning
* Iterative processing: Spark is more efficient for iterative algorithms common in machine learning

### 15. What is Apache Kafka?

Apache Kafka is a distributed streaming platform that allows for publishing and subscribing to streams of records, storing streams of records in a fault-tolerant way, and processing streams of records as they occur.

## *****Data Engineer Interview Questions on Data Warehousing and ETL*****

### 16. What is a data warehouse?

A data warehouse is a centralized repository that stores large amounts of structured data from various sources in an organization. It is designed for query and analysis rather than for transaction processing.

### 17. Explain the ETL process.

ETL stands for Extract, Transform, Load. It is a process used to collect data from various sources, transform it to fit operational needs, and load it into the end target, usually a data warehouse. The steps are:

* Extract: Retrieve data from source systems
* Transform: Clean, validate, and convert the data into a suitable format
* Load: Insert the transformed data into the target system

### 18. What is the difference between a data lake and a data warehouse?

### A: Key differences include:

* Data structure: Data warehouses store structured data, while data lakes can store structured, semi-structured, and unstructured data
* Purpose: Data warehouses are optimized for analysis, while data lakes serve as a repository for raw data
* Schema: Data warehouses use schema-on-write, while data lakes use schema-on-read
* Users: Data warehouses are typically used by business analysts, while data lakes are often used by data scientists

### 19. What is the slowly changing dimension (SCD)?

Slowly changing dimension (SCD) is a concept in data warehousing that describes how to handle changes to dimension data over time. There are different types of SCDs, with the most common being:

* Type 1: Overwrite the old value
* Type 2: Create a new row with the changed data
* Type 3: Add a new column to track changes

### 20. What is data mart?

A data mart is a subset of a data warehouse that focuses on a specific business line or department. It contains summarized and relevant data for a particular group of users or a specific area of the business.

### **Cloud Computing for Data Engineering**

### 21. What are the main advantages of cloud computing for data engineering?

Key advantages include:

* Scalability: Easily scale resources up or down based on demand
* Cost-effectiveness: Pay only for the resources you use
* Flexibility: Access to a wide range of services and tools
* Reliability: Built-in redundancy and disaster recovery options
* Global reach: Deploy resources in multiple geographic regions

### 22. What is Amazon S3?

Amazon S3 (Simple Storage Service) is an object storage service offered by Amazon Web Services (AWS). It provides scalable, durable, and highly available storage for various types of data, making it popular for data lakes and backup solutions.

### 23. Explain the concept of a data lake in the context of cloud computing.

A data lake in the cloud is a centralized repository that allows you to store all your structured and unstructured data at any scale. It's typically built using cloud storage services like Amazon S3 or Azure Data Lake Storage, providing a flexible and cost-effective solution for big data analytics and machine learning projects.

### 24 What is Azure Synapse Analytics?

Azure Synapse Analytics is a limitless analytics service that brings together data integration, enterprise data warehousing, and big data analytics. It allows you to query data on your terms, using either serverless or dedicated resources at scale.

### 26. What are some popular programming languages used in data engineering?

A: Popular programming languages for data engineering include:

* Python
* SQL
* Java
* Scala
* R

### 27. Why is Python popular in data engineering?

Python is popular in data engineering due to:

* Ease of use and readability
* Rich ecosystem of libraries and frameworks for data processing (e.g., Pandas, NumPy)
* Support for big data technologies (e.g., PySpark)
* Integration with various data sources and APIs
* Strong community support and documentation

### 28. What is PySpark?

PySpark is the Python API for Apache Spark. It allows you to write Spark applications using Python, combining the simplicity of Python with the power of Spark for distributed data processing.

### 29. What are some key features of Scala for data engineering?

Key features of Scala for data engineering include:

* Compatibility with Java libraries and frameworks
* Strong static typing, which can catch errors at compile-time
* Concise syntax for functional programming
* Native language for Apache Spark
* Good performance for large-scale data processing

### 30. How does R compare to Python for data engineering tasks?

While R is more popular in statistical computing and data analysis, it can also be used for data engineering tasks. Compared to Python:

* R has stronger statistical and visualization capabilities out-of-the-box
* Python has a more general-purpose nature and is often easier to integrate with other systems
* Both have packages for data manipulation (e.g., dplyr in R, Pandas in Python)
* Python is generally faster for large-scale data processing
* R has a steeper learning curve for those without a statistical background

## *Data Engineer Interview Questions on Data Modeling and Design*

### 31. What is data modeling?

Data modeling is the process of creating a visual representation of data structures and relationships within a system. It helps in understanding, organizing, and standardizing data elements and their relationships.

### 32. What are the three main types of data models?

The three main types of data models are:

1. Conceptual data model: High-level view of data structures and relationships
2. Logical data model: Detailed view of data structures, independent of any specific database management system
3. Physical data model: Representation of the data model as implemented in a specific database system

### 33. What is star schema?

Star schema is a data warehouse schema where a central fact table is surrounded by dimension tables. It's called a star schema because the diagram resembles a star, with the fact table at the center and dimension tables as points.

### 34. What is snowflake schema?

Snowflake schema is a variation of the star schema where dimension tables are normalized into multiple related tables. This creates a structure that looks like a snowflake, with the fact table at the center and increasingly granular dimension tables branching out.

### 35. What are the advantages and disadvantages of denormalization?

**Advantages of denormalization:**

* Improved query performance
* Simplifies queries
* Reduces the need for joins

**Disadvantages of denormalization:**

* Increased data redundancy
* More complex data updates and inserts
* Potential data inconsistencies

## *Data Engineer Interview Questions on Data Processing and Analytics*

### 36. What is batch processing?

Batch processing is a method of running high-volume, repetitive data jobs where a group of transactions is collected over time, then processed all at once. It's efficient for processing large amounts of data when immediate results are not required.

### 37. What is stream processing?

Stream processing is a method of processing data continuously as it is generated or received. It allows for real-time or near real-time analysis and action on incoming data streams.

### 38. What is the Lambda architecture?

The Lambda architecture is a data processing architecture designed to handle massive quantities of data by taking advantage of both batch and stream processing methods. It consists of three layers:

1. Batch layer: Manages the master dataset and pre-computes batch views
2. Speed layer: Handles real-time data processing
3. Serving layer: Responds to queries by combining results from batch and speed layers

### 39. What is Apache Flink?

Apache Flink is an open-source stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications. It provides precise control of time and state, allowing for consistent and accurate results even in the face of out-of-order or late-arriving data.

### 40. Explain the concept of data partitioning.

Data partitioning is the process of dividing a large dataset into smaller, more manageable pieces called partitions. This technique is used to improve query performance, enable parallel processing, and manage large datasets more effectively. Common partitioning strategies include:

* Range partitioning
* Hash partitioning
* List partitioning

## *Data Engineer Interview Questions on Data Security and Governance*

### 41. What is data governance?

Data governance is a set of processes, roles, policies, standards, and metrics that ensure the effective and efficient use of information in enabling an organization to achieve its goals. It establishes the processes and responsibilities for data quality, security, and compliance.

### 42. What is data encryption?

Data encryption is the process of converting data into a code to prevent unauthorized access. It involves using an algorithm to transform the original data (plaintext) into an unreadable format (ciphertext) that can only be decrypted with a specific key.

### 43. What is GDPR and how does it affect data engineering?

GDPR (General Data Protection Regulation) is a regulation in EU law on data protection and privacy. For data engineering, it impacts:

* Data collection and storage practices
* Data processing and usage
* Data subject rights (e.g., right to be forgotten)
* Data breach notification requirements
* Cross-border data transfers

### 44. What is data masking?

Data masking is a technique used to create a structurally similar but inauthentic version of an organization's data. It's used to protect sensitive data while providing a functional substitute for purposes such as software testing and user training.

### 45. What is role-based access control (RBAC)?

Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an organization. In RBAC, permissions are associated with roles, and users are assigned to appropriate roles, simplifying the management of user rights.

## *****Data Engineer Interview Questions on Soft Skills and Problem-Solving*****

### 46. How do you approach learning new technologies in the rapidly evolving field of data engineering?

Possible approaches include:

* Regularly reading tech blogs and articles
* Participating in online courses and certifications
* Attending conferences and workshops
* Experimenting with new tools in personal projects
* Collaborating with colleagues and sharing knowledge
* Following industry experts on social media

### 47. How do you ensure data quality in your projects?

Strategies for ensuring data quality include:

* Implementing data validation checks at ingestion
* Using data profiling tools to understand data characteristics
* Establishing clear data quality metrics and monitoring them
* Implementing data cleansing processes
* Conducting regular data audits
* Establishing a data governance framework

### 48. How do you handle conflicts in a team environment?

Strategies for handling conflicts include:

* Active listening to understand all perspectives
* Focusing on the issue, not personal differences
* Seeking common ground and shared goals
* Proposing and discussing potential solutions
* Escalating to management when necessary, with proposed resolutions

### 49. How do you prioritize tasks in a data engineering project?

Prioritization strategies might include:

* Assessing business impact and urgency of each task
* Considering dependencies between tasks
* Evaluating resource availability and constraints
* Using techniques like the Eisenhower Matrix or MoSCoW method
* Regular communication with stakeholders to align priorities

### 50. How do you stay updated with the latest trends and best practices in data engineering?

Methods to stay updated include:

* Following relevant blogs, podcasts, and YouTube channels
* Participating in online communities (e.g., Stack Overflow, Reddit)
* Attending webinars and virtual conferences
* Subscribing to industry newsletters
* Networking with other professionals in the field
* Experimenting with new tools and technologies in personal projects

### 51. How would you design a system to handle real-time streaming data?

When designing a system for real-time streaming data, consider:

* Using a distributed streaming platform like Apache Kafka or Amazon Kinesis
* Implementing stream processing with tools like Apache Flink or Spark Streaming
* Ensuring low-latency data ingestion and processing
* Designing for fault tolerance and scalability
* Implementing proper error handling and data validation
* Considering data storage for both raw and processed data

### 52. What strategies do you use for optimizing query performance in large datasets?

Strategies for optimizing query performance include:

* Proper indexing of frequently queried columns
* Partitioning large tables
* Using materialized views for complex, frequently-run queries
* Query optimization and rewriting
* Implementing caching mechanisms
* Using columnar storage formats for analytical workloads
* Leveraging distributed computing for large-scale data processing

### 53. How do you approach data pipeline testing?

 Approaches to data pipeline testing include:

* Unit testing individual components
* Integration testing to ensure components work together
* End-to-end testing of the entire pipeline
* Data validation testing to ensure data integrity
* Performance testing under various load conditions
* Fault injection testing to verify error handling
* Regression testing after making changes

### 54. What is your experience with data versioning and how do you implement it?

Data versioning involves tracking changes to datasets over time. Implementation strategies include:

* Using version control systems for code and configuration files
* Implementing slowly changing dimensions in data warehouses
* Using data lake technologies that support versioning (e.g., Delta Lake)
* Maintaining metadata about dataset versions
* Implementing a robust backup and restore strategy

### 55. How do you handle data skew in distributed processing systems?

Strategies for handling data skew include:

* Identifying and analyzing skewed keys
* Implementing salting or hashing techniques to distribute data more evenly
* Using broadcast joins for small datasets
* Adjusting partition sizes or using custom partitioners
* Implementing two-phase aggregation for skewed aggregations
* Considering alternative data models or schema designs

### 56. Explain the concept of data lineage and why it's important.

Data lineage refers to the lifecycle of data, including its origins, movements, transformations, and impacts. It's important because it:

* Helps in understanding data provenance and quality
* Facilitates impact analysis for proposed changes
* Aids in regulatory compliance and auditing
* Supports troubleshooting and debugging of data issues
* Enhances data governance and metadata management

### 57. How do you approach capacity planning for data infrastructure?

Capacity planning involves:

* Analyzing current resource usage and growth trends
* Forecasting future data volumes and processing requirements
* Considering peak load scenarios and seasonality
* Evaluating different scaling options (vertical vs. horizontal)
* Assessing costs and budget constraints
* Planning for redundancy and fault tolerance
* Considering cloud vs. on-premises infrastructure options

### 58. What is your experience with data catalogs and metadata management?

Data catalogs and metadata management involve:

* Implementing tools for documenting datasets, their schemas, and relationships
* Establishing processes for metadata creation and maintenance
* Integrating metadata across different systems and tools
* Implementing data discovery and search capabilities
* Supporting data governance and compliance initiatives
* Facilitating self-service analytics for business users

### 59. How do you handle schema evolution in data pipelines?

Approaches to handling schema evolution include:

* Using schema-on-read formats like Parquet or Avro
* Implementing backward and forward compatibility in schema designs
* Versioning schemas and maintaining compatibility between versions
* Using schema registries for centralized schema management
* Implementing data migration strategies for major schema changes
* Testing schema changes thoroughly before deployment

### 60. What is your approach to monitoring and alerting in data engineering systems?

Effective monitoring and alerting involves:

* Implementing comprehensive logging across all system components
* Setting up real-time monitoring dashboards
* Defining key performance indicators (KPIs) and service level objectives (SLOs)
* Implementing proactive alerting for potential issues
* Using anomaly detection techniques for identifying unusual patterns
* Establishing an incident response process
* Conducting regular system health checks and audits

61. How do you ensure data consistency in distributed systems?

A: Strategies for ensuring data consistency include:

* Implementing strong consistency models where necessary
* Using eventual consistency for improved performance in certain scenarios
* Implementing distributed transactions when needed
* Using techniques like two-phase commit or saga pattern for complex operations
* Implementing idempotent operations to handle duplicate requests
* Designing for conflict resolution in multi-master systems

### 62. What is your experience with data modeling for NoSQL databases?

Data modeling for NoSQL databases involves:

* Understanding the specific [NoSQL](https://www.geeksforgeeks.org/introduction-to-nosql/)database type (document, key-value, column-family, graph)
* Designing for query patterns rather than normalized data structures
* Considering denormalization and data duplication for performance
* Planning for scalability and partitioning
* Implementing appropriate indexing strategies
* Handling schema flexibility and evolution

### 63. How do you approach data quality assurance in ETL processes?

Data quality assurance in ETL involves:

* Implementing data validation rules at the source and target
* Performing data profiling to understand data characteristics
* Implementing data cleansing and standardization processes
* Using data quality scorecards to track improvements over time
* Implementing data reconciliation checks between source and target
* Establishing a process for handling and resolving data quality issues

### 64. What strategies do you use for managing technical debt in data engineering projects?

Strategies for managing technical debt include:

* Regular code reviews and refactoring sessions
* Implementing CI/CD practices for consistent deployments
* Maintaining comprehensive documentation
* Prioritizing critical updates and migrations
* Allocating time for system improvements in project planning
* Conducting periodic architecture reviews
* Implementing automated testing to catch regressions

### 65. How do you handle data privacy and compliance requirements in your projects?

Approaches to handling data privacy and compliance include:

* Implementing data classification and tagging
* Applying appropriate data masking and encryption techniques
* Implementing [role-based access control](https://www.geeksforgeeks.org/role-based-access-control/) (RBAC)
* Maintaining audit logs for data access and modifications
* Implementing data retention and deletion policies
* Conducting regular privacy impact assessments
* Staying updated with relevant regulations (e.g., GDPR, CCPA)

## 7. Data Science Interview Questions

## <https://www.geeksforgeeks.org/data-science-interview-questions-and-answers/>

## (Last Updated : 19 Oct, 2024)

## What is Data Science?

Data science is a field that extracts knowledge and insights from structured and unstructured data by using scientific methods, algorithms, processes, and systems. It combines expertise from various domains, such as statistics, computer science, machine learning, data engineering, and domain-specific knowledge, to analyze and interpret complex data sets.

Furthermore, data scientists use a combination of multiple languages, such as [**Python**](https://www.geeksforgeeks.org/python-programming-language) and [**R**](https://www.geeksforgeeks.org/r-programming-language-introduction). They are also frequent users of data analysis tools like pandas, NumPy, and scikit-learn, as well as machine learning libraries.

After exploring the brief of data science, let's dig into the data science interview questions and answers.

## *Basic Data Science Interview Questions For Fresher*

### Q.1 What is marginal probability?

Marginal probability is the [probability](https://www.geeksforgeeks.org/probability-in-maths) of a single event occurring independently, without considering the outcomes of other events. It focuses solely on the likelihood of one specific event happening, independent of any other events. Marginal probability is obtained by summing or integrating the joint probabilities of the event of interest across all possible outcomes of other events.

### Q.2 What are the probability axioms?

The fundamental rules that control the behaviour and characteristics of probabilities in probability theory and statistics are referred to as the [probability axioms](https://www.geeksforgeeks.org/axiomatic-probability-in-r), sometimes known as the [probability laws](https://www.geeksforgeeks.org/properties-of-probability) or probability principles.

**There are three fundamental axioms of probability:**

1. Non-Negativity Axiom
2. Normalization Axiom
3. Additivity Axiom

### Q.3 What is conditional probability?

Conditional probability refers to the probability of an event occurring given that another event has already occurred. Mathematically, it is defined as the probability of event A occurring, given that event B has occurred, and is denoted by P(A∣B)*P*(*A*∣*B*) .

The formula for conditional probability is:

P(A∣B)=P(A∩B)P(B)*P*(*A*∣*B*)=*P*(*B*)*P*(*A*∩*B*)​

where:

* P(A|B) is the conditional probability of event A given event B.
* P(A∩B)*P*(*A*∩*B*) is the joint probability of both events A and B occurring simultaneously.
* P(B)  is the probability of event B occurring.

### Q.4 What is Bayes’ Theorem and when is it used in data science?

The Bayes theorem predicts the probability that an event connected to any condition would occur. It is also taken into account in the situation of conditional probability. The probability of “causes” formula is another name for the Bayes theorem.

**In data science, Bayes’ Theorem is used primarily in:**

1. Bayesian Inference
2. Machine Learning
3. Text Classification
4. Medical Diagnosis
5. Predictive Modeling

When working with ambiguous or sparse data, Bayes’ Theorem is very helpful since it enables data scientists to continually revise their assumptions and come to more sensible conclusions.

### Q.5 Define variance and conditional variance.

A statistical concept known as variance quantifies the spread or dispersion of a group of data points within a dataset. It sheds light on how widely individual data points depart from the dataset’s mean (average). It assesses the variability or “scatter” of data.

**Conditional Variance**

A measure of the dispersion or variability of a random variable under certain circumstances or in the presence of a particular event, as the name implies. It reflects a random variable’s variance that is dependent on the knowledge of another random variable’s variance.

### Q.6 Explain the concepts of mean, median, mode, and standard deviation.

**Mean:**The mean, often referred to as the average, is calculated by summing up all the values in a dataset and then dividing by the total number of values.

**Median:** When data are sorted in either ascending or descending order, the median is the value in the middle of the dataset. The median is the average of the two middle values when the number of data points is even.  
In comparison to the mean, the median is less impacted by extreme numbers, making it a more reliable indicator of central tendency.

**Mode:**The value that appears most frequently in a dataset is the mode. One mode (unimodal), several modes (multimodal), or no mode (if all values occur with the same frequency) can all exist in a dataset.

**Standard deviation**: The spread or dispersion of data points in a dataset is measured by the standard deviation. It quantifies the variance between different data points.

### Q.7 What is the normal distribution and standard normal distribution?

The normal distribution, also known as the Gaussian distribution or bell curve, is a continuous probability distribution that is characterized by its symmetric bell-shaped curve. The normal distribution is defined by two parameters: the mean (μ) and the standard deviation (σ). The mean determines the center of the distribution, and the standard deviation determines the spread or dispersion of the distribution. The distribution is symmetric around its mean, and the bell curve is centered at the mean. The probabilities for values that are further from the mean taper off equally in both directions. Similar rarity applies to extreme values in the two tails of the distribution. Not all symmetrical distributions are normal, even though the normal distribution is symmetrical.

The standard normal distribution, also known as the Z distribution, is a special case of the normal distribution where the mean (μ) is 0 and the standard deviation (σ) is 1. It is a standardized form of the normal distribution, allowing for easy comparison of scores or observations from different normal distributions.

### Q.8 What is SQL, and what does it stand for?

[**SQL**](https://www.geeksforgeeks.org/what-is-sql) stands for Structured Query Language.It is a specialized programming language used for managing and manipulating relational databases. It is designed for tasks related to database management, data retrieval, data manipulation, and data definition.

### Q.9 Explain the differences between SQL and NoSQL databases.

Both [**SQL**](https://www.geeksforgeeks.org/sql-tutorial) (Structured Query Language) and [NoSQL](https://www.geeksforgeeks.org/introduction-to-nosql) (Not Only SQL) databases, differ in their data structures, schema, query languages, and use cases. The following are the main variations between SQL and NoSQL databases.

| **SQL** | **NoSQL** |
| --- | --- |
| SQL databases are relational databases, they organise and store data using a structured schema with tables, rows, and columns. | NoSQL databases use a number of different types of data models, such as document-based (like JSON and BSON), key-value pairs, column families, and graphs. |
| SQL databases have a set schema, thus before inserting data, we must establish the structure of our data.The schema may need to be changed, which might be a difficult process. | NoSQL databases frequently employ a dynamic or schema-less approach, enabling you to insert data without first creating a predetermined schema. |
| SQL is a strong and standardised query language that is used by SQL databases. Joins, aggregations, and subqueries are only a few of the complicated processes supported by SQL queries. | The query languages or APIs used by NoSQL databases are frequently tailored to the data model. |

### Q.10 What are the primary SQL database management systems (DBMS)?

Relational database systems, both open source and commercial, are the main SQL (Structured Query Language) database management systems (DBMS), which are widely used for managing and processing structured data. Some of the most popular SQL database management systems are listed below:

1. [MySQL](https://www.geeksforgeeks.org/mysql-introdution)
2. [Microsoft SQL Server](https://www.geeksforgeeks.org/introduction-of-ms-sql-server)
3. [SQLite](https://www.geeksforgeeks.org/introduction-to-sqlite)
4. [PostgreSQL](https://www.geeksforgeeks.org/postgresql-tutorial)
5. [Oracle Database](https://www.geeksforgeeks.org/how-to-install-oracle-database-11g-on-windows)
6. [Amazon RDS](https://www.geeksforgeeks.org/amazon-rds-introduction-to-amazon-relational-database-system)

### Q.11 What is the ER model in SQL?

The structure and relationships between the data entities in a database are represented by the Entity-Relationship (ER) model, a conceptual framework used in database architecture. The ER model is frequently used in conjunction with SQL for creating the structure of relational databases even though it is not a component of the SQL language itself.

### Q.12 What is data transformation?

The process of transforming data from one structure, format, or representation into another is referred to as data transformation. In order to make the data more suited for a given goal, such as analysis, visualisation, reporting, or storage, this procedure may involve a variety of actions and changes to the data. Data integration, cleansing, and analysis depend heavily on data transformation, which is a common stage in data preparation and processing pipelines.

### Q.13 What are the main components of a SQL query?

A relational database’s data can be retrieved, modified, or managed via a SQL (Structured Query Language) query. The operation of a SQL query is defined by a number of essential components, each of which serves a different function.

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY
7. LIMIT
8. JOIN

### Q.14 What is a primary key?

A relational database table’s main key, also known as a primary keyword, is a column that is unique for each record. It is a distinctive identifier.The primary key of a relational database must be unique. Every row of data must have a primary key value and none of the rows can be null.

### Q.15 What is the purpose of the GROUP BY clause, and how is it used?

In SQL, the GROUP BY clause is used to create summary rows out of rows that have the same values in a set of specified columns. In order to do computations on groups of rows as opposed to individual rows, it is frequently used in conjunction with aggregate functions like SUM, COUNT, AVG, MAX, or MIN. we may produce summary reports and perform more in-depth data analysis using the GROUP BY clause.

### Q.16 What is the WHERE clause used for, and how is it used to filter data?

In SQL, the WHERE clause is used to filter rows from a table or result set according to predetermined criteria. It enables us to pick only the rows that satisfy particular requirements or follow a pattern. A key element of SQL queries, the WHERE clause is frequently used for data retrieval and manipulation.

### Q.17 How do you retrieve distinct values from a column in SQL?

Using the DISTINCT keyword in combination with the SELECT command, we can extract distinct values from a column in SQL. By filtering out duplicate values and returning only unique values from the specified column, the DISTINCT keyword is used.

### Q.18 What is the HAVING clause?

To filter query results depending on the output of aggregation functions, the HAVING clause, a SQL clause, is used along with the GROUP BY clause. The HAVING clause filters groups of rows after they have been grouped by one or more columns, in contrast to the WHERE clause, which filters rows before they are grouped.

### Q.19 How do you handle missing or NULL values in a database table?

Missing or NULL values can arise due to various reasons, such as incomplete data entry, optional fields, or data extraction processes.

1. Replace NULL with Placeholder Values
2. Handle NULL Values in Queries
3. Use Default Values

### Q.20 What is the difference between supervised and unsupervised machine learning?

The difference between Supervised Learning and Unsupervised Learning are as follow:

| **Category** | **Supervised Learning** | **Unsupervised Learning** |
| --- | --- | --- |
| **Definition** | Supervised learning refers to that part of machine learning where we know what the target variable is and it is labeled. | Unsupervised Learning is used when we do not have labeled data and we are not sure about our target variable |
| **Objective** | The objective of supervised learning is to predict an outcome or classify the data | The objective here is to discover patterns among the features of the dataset and group similar features together |
| **Algorithms** | Some of the algorithm types are:   1. Regression (Linear, Logistic, etc.) 2. Classification (Decision Tree Classifier, Support Vector Classifier, etc.) | Some of the algorithms are :   1. Dimensionality reduction (Principle Component Analysis, etc.) 2. Clustering (KMeans, DBSCAN, etc.) |
| **Evaluation metrics** | Supervised learning uses evaluation metrics like:   1. Mean Squared Error 2. Accuracy | Unsupervised Learning uses evaluation metrics like:   1. Silhouette 2. Inertia |
| **Use cases** | Predictive modeling, Spam detection | Anomaly detection, Customer segmentation |

### Q.21 What is linear regression, and What are the different assumptions of linear regression algorithms?

**Linear Regression -** It is type of Supervised Learning where we compute a linear relationship between the predictor and response variable. It is based on the linear equation concept given by:

y^=β1x+βo*y*^​=*β*1​*x*+*βo*​,

where

* y^*y*^​ = response / dependent variable
* β1*β*1​ = slope of the linear regression
* βo*βo*​ = intercept for linear regression
* x*x* = predictor / independent variable(s)

There are 4 assumptions we make about a Linear regression problem:

* **Linear relationship :** This assumes that there is a linear relationship between predictor and response variable. This means that, which changing values of predictor variable, the response variable changes linearly (either increases or decreases).
* **Normality**: This assumes that the dataset is normally distributed, i.e., the data is symmetric about the mean of the dataset.
* **Independence**: The features are independent of each other, there is no correlation among the features/predictor variables of the dataset.
* **Homoscedasticity**: This assumes that the dataset has equal variance for all the predictor variables. This means that the amount of independent variables have no effect on the variance of data.

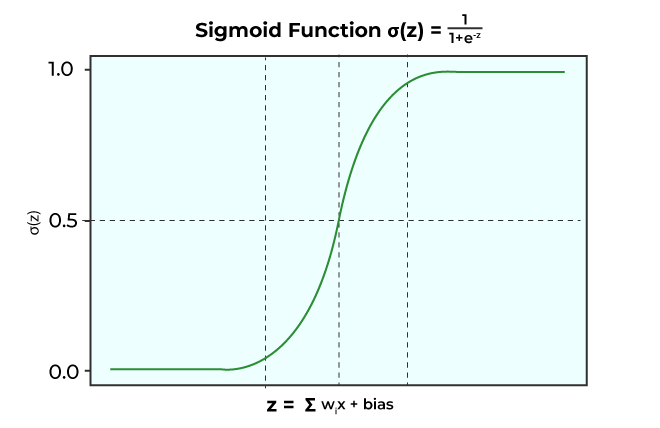
### Q.22 Logistic regression is a classification technique, why its name is regressions, not logistic classifications?

While logistic regression is used for classification, it still maintains a regression structure underneath. The key idea is to model the probability of an event occurring (e.g., class 1 in binary classification) using a linear combination of features, and then apply a logistic (Sigmoid) function to transform this linear combination into a probability between 0 and 1. This transformation is what makes it suitable for classification tasks.

In essence, while logistic regression is indeed used for classification, it retains the mathematical and structural characteristics of a regression model, hence the name.

### Q.23 What is the logistic function (sigmoid function) in logistic regression?

**Sigmoid Function:**It is a mathematical function which is characterized by its S- shape curve. Sigmoid functions have the tendency to squash a data point to lie within 0 and 1. This is why it is also called Squashing function, which is given as:



Some of the properties of Sigmoid function is:

* Range: [0,1]

### Q.24 What is overfitting and how can be overcome this?

Overfitting refers to the result of analysis of a dataset which fits so closely with training data that it fails to generalize with unseen/future data. This happens when the model is trained with noisy data which causes it to learn the noisy features from the training as well.

To avoid Overfitting and overcome this problem in machine learning, one can follow the following rules:

* **Feature selection :** Sometimes the training data has too many features which might not be necessary for our problem statement. In that case, we use only the necessary features that serve our purpose
* **Cross Validation :** This technique is a very powerful method to overcome overfitting. In this, the training dataset is divided into a set of mini training batches, which are used to tune the model.
* **Regularization :** Regularization is the technique to supplement the loss with a penalty term so as to reduce overfitting. This penalty term regulates the overall loss function, thus creating a well trained model.
* **Ensemble models :** These models learn the features and combine the results from different training models into a single prediction.

### Q.25 What is a support vector machine (SVM), and what are its key components?

Support Vector machines are a type of Supervised algorithm which can be used for both Regression and Classification problems. In SVMs, the main goal is to find a hyperplane which will be used to segregate different data points into classes. Any new data point will be classified based on this defined hyperplane.

Support Vector machines are highly effective when dealing with high dimensionality space and can handle non linear data very well. But if the number of features are greater than number of data samples, it is susceptible to overfitting.

The key components of SVM are:

* **Kernels Function**: It is a mapping function used for data points to convert it into high dimensionality feature space.
* **Hyperplane**: It is the decision boundary which is used to differentiate between the classes of data points.
* **Margin**: It is the distance between Support Vector and Hyperplane
* **C:**It is a regularization parameter which is used for margin maximization and misclassification minimization.

### Q.26 Explain the k-nearest neighbors (KNN) algorithm.

The k-Nearest Neighbors (KNN) algorithm is a simple and versatile supervised machine learning algorithm used for both **classification and regression** tasks. KNN makes predictions by memorizing the data points rather than building a model about it. This is why it is also called “**lazy learner**” or “**memory based**” model too.

KNN relies on the principle that similar data points tend to belong to the same class or have similar target values. This means that, In the training phase, KNN stores the entire dataset consisting of feature vectors and their corresponding class labels (for classification) or target values (for regression). It then calculates the distances between that point and all the points in the training dataset. (commonly used distance metrics are Euclidean distance and Manhattan distance).

(Note : Choosing an appropriate value for k is crucial. A small k may result in noisy predictions, while a large k can smooth out the decision boundaries. The choice of distance metric and feature scaling also impact KNN’s performance.)

### Q.27 What is the Naïve Bayes algorithm, what are the different assumptions of Naïve Bayes?

The Naïve Bayes algorithm is a probabilistic classification algorithm based on Bayes’ theorem with a “naïve” assumption of feature independence within each class. It is commonly used for both binary and multi-class classification tasks, particularly in situations where simplicity, speed, and efficiency are essential.

The main assumptions that Naïve Bayes theorem makes are:

1. **Feature independence** – It assumes that the features involved in Naïve Bayes algorithm are conditionally independent, i.e., the presence/ absence of one feature does not affect any other feature
2. **Equality**– This assumes that the features are equal in terms of importance (or weight).
3. **Normality**– It assumes that the feature distribution is Normal in nature, i.e., the data is distributed equally around its mean.

### Q.28 What are decision trees, and how do they work?

Decision trees are a popular machine learning algorithm used for both classification and regression tasks. They work by creating a tree-like structure of decisions based on input features to make predictions or decisions. Lets dive into its core concepts and how they work briefly:

* Decision trees consist of nodes and edges.
* The tree starts with a root node and branches into internal nodes that represent features or attributes.
* These nodes contain decision rules that split the data into subsets.
* Edges connect nodes and indicate the possible decisions or outcomes.
* Leaf nodes represent the final predictions or decisions.

The objective is to increase data homogeneity, which is often measured using standards like mean squared error (for regression) or Gini impurity (for classification). Decision trees can handle a variety of attributes and can effectively capture complex data relationships. They can, however, overfit, especially when deep or complex. To reduce overfitting, strategies like pruning and restricting tree depth are applied.

### Q.29 Explain the concepts of entropy and information gain in decision trees.

**Entropy**: Entropy is the measure of randomness. In terms of Machine learning, Entropy can be defined as the measure of randomness or impurity in our dataset.

**Information gain:**It is defined as the change in the entropy of a feature given that there’s an additional information about that feature. If there are more than one features involved in Decision tree split, then the weighted average of entropies of the additional features is taken.

### Q.30 What is the difference between the bagging and boosting model?

| **Category** | **Bagging Model** | **Boosting model** |
| --- | --- | --- |
| **Definition** | Bagging, or Bootstrap aggregating, is an ensemble modelling method where predictions from different models are combined together to give the aggregated result | Boosting method is where multiple weak learners are used together to get a stronger model with more robust predictions. |
| **Agenda** | This is used when dealing with models that have high variance (overfitting). | This is used when dealing with models with high bias (underfitting) and variance as well. |
| **Robustness to Noise and Sensitivity** | This is more robust due to averaging and this makes it less sensitive | It is more sensitive to presence of outliers and that makes it a bit less robust as compared to bagging models |
| **Model running and dependence** | The models are run in parallel and are typically independent | The models are run in sequential method where the base model is dependent. |
| **Examples** | Random Forest, Bagged Decision Trees | AdaBoost, Gradient Boosting, XGBoost |

### Q.31 Describe random forests and their advantages over single-decision trees.

Random Forests are an ensemble learning technique that combines multiple decision trees to improve predictive accuracy and reduce overfitting. The advantages it has over single decision trees are:

* **Improved Generalization**: Single decision trees are prone to overfitting, especially when they become deep and complex. Random Forests mitigate this issue by averaging predictions from multiple trees, resulting in a more generalized model that performs better on unseen data
* **Better Handling of High-Dimensional Data :**Random Forests are effective at handling datasets with a large number of features. They select a random subset of features for each tree, which can improve the performance when there are many irrelevant or noisy features
* **Robustness to Outliers:**Random Forests are more robust to outliers because they combine predictions from multiple trees, which can better handle extreme cases

### Q.32 What is K-Means, and how will it work?

K-Means is an unsupervised machine learning algorithm used for clustering or grouping similar data points together. It aims to partition a dataset into K clusters, where each cluster represents a group of data points that are close to each other in terms of some similarity measure. The working of K-means is as follow:

* Choose the number of clusters K
* For each data point in the dataset, calculate its distance to each of the K centroids and then assign each data point to the cluster whose centroid is closest to it
* Recalculate the centroids of the K clusters based on the current assignment of data points.
* Repeat the above steps until a group of clusters are formed.

### Q.33 What is a confusion matrix? Explain with an example.

Confusion matrix is a table used to evaluate the performance of a classification model by presenting a comprehensive view of the model’s predictions compared to the actual class labels. It provides valuable information for assessing the model’s accuracy, precision, recall, and other performance metrics in a binary or multi-class classification problem.

A famous example demonstration would be Cancer Confusion matrix:

|  | | **Actual** | |
| --- | --- | --- | --- |
| Cancer | Not Cancer |
| **Predicted** | Cancer | True Positive (TP) | False Positive (FP) |
| Not Cancer | False Negative (FN) | True Negative (TN) |

* **TP (True Positive)** = The number of instances correctly predicted as the positive class
* **TN (True Negative)** = The number of instances correctly predicted as the negative class
* **FP (False Positive)** = The number of instances incorrectly predicted as the positive class
* **FN (False Negative)** = The number of instances incorrectly predicted as the negative class

### Q.34 What is a classification report and explain the parameters used to interpret the result of classification tasks with an example.

A classification report is a summary of the performance of a classification model, providing various metrics that help assess the quality of the model’s predictions on a classification task.

The parameters used in a classification report typically include:

* **Precision**: Precision is the ratio of true positive predictions to the total predicted positives. It measures the accuracy of positive predictions made by the model.

Precision = TP/(TP+FP)

* **Recall (Sensitivity or True Positive Rate)**: Recall is the ratio of true positive predictions to the total actual positives. It measures the model’s ability to identify all positive instances correctly.

Recall = TP / (TP + FN)

* **Accuracy**: Accuracy is the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances. It measures the overall correctness of the model’s predictions.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

* **F1-Score**: The F1-Score is the harmonic mean of precision and recall. It provides a balanced measure of both precision and recall and is particularly useful when dealing with imbalanced datasets.

F1-Score = 2 \* (Precision \* Recall) / (Precision + Recall)

where,

* TP = True Positive
* TN = True Negative
* FP = False Positive
* FN = False Negative

## Intermediate Data Science Interview Questions

***Data Science Interview Questions and Answers***

### Q.35 Explain the uniform distribution.

A fundamental probability distribution in statistics is the uniform distribution, commonly referred to as the rectangle distribution. A constant probability density function (PDF) across a limited range characterises it. In simpler terms, in a uniform distribution, every value within a specified range has an equal chance of occurring.

### Q.36 Describe the Bernoulli distribution.

A discrete probability distribution, the Bernoulli distribution is focused on discrete random variables. The number of heads you obtain while tossing three coins at once or the number of pupils in a class are examples of discrete random variables that have a finite or countable number of potential values.

### Q.37 What is the binomial distribution?

The binomial distribution is a discrete probability distribution that describes the number of successes in a fixed number of independent Bernoulli trials, where each trial has only two possible outcomes: success or failure. The outcomes are often referred to as "success" and "failure," but they can represent any dichotomous outcome, such as heads or tails, yes or no, or defective or non-defective.

The fundamental presumptions of a binomial distribution are that each trial has exactly one possible outcome, each trial has an equal chance of success, and each trial is either independent of the others or mutually exclusive.

### Q.38 Explain the exponential distribution and where it’s commonly used.

The probability distribution of the amount of time between events in the Poisson point process is known as the exponential distribution. The gamma distribution is thought of as a particular instance of the exponential distribution. Additionally, the geometric distribution’s continuous analogue is the exponential distribution.

**Common applications of the exponential distribution include:**

1. Reliability Engineering
2. Queueing Theory
3. Telecommunications
4. Finance
5. Natural Phenomena
6. Survival Analysis

### Q.39 Describe the Poisson distribution and its characteristics.

The Poisson distribution is a probability distribution that describes the number of events that occur within a fixed interval of time or space when the events happen at a constant mean rate and are independent of the time since the last event.

**Key characteristics** of the Poisson distribution include:

1. **Discreteness:** The Poisson distribution is used to model the number of discrete events that occur within a fixed interval.
2. **Constant Mean Rate:** The events occur at a constant mean rate per unit of time or space.
3. **Independence:** The occurrences of events are assumed to be independent of each other. The probability of multiple events occurring in a given interval is calculated based on the assumption of independence.

### Q40. Explain the t-distribution and its relationship with the normal distribution.

The t-distribution, also known as the Student's t-distribution, is used in statistics for inferences about population means when the sample size is small and the population standard deviation is unknown. The shape of the t-distribution is similar to the normal distribution, but it has heavier tails.

**Relationship between T-Distribution and Normal Distribution:** The t-distribution converges to the normal distribution as the degrees of freedom increase. In fact, when the degrees of freedom become very large, the t-distribution approaches the standard normal distribution (normal distribution with mean 0 and standard deviation 1). This is a result of the Central Limit Theorem.

### Q.41 Describe the chi-squared distribution.

The chi-squared distribution is a continuous probability distribution that arises in statistics and probability theory. It is commonly denoted as χ2 (chi-squared) and is associated with degrees of freedom. The chi-squared distribution is particularly used to model the distribution of the sum of squared independent standard normal random variables.It is also used to determine if data series are independent, the goodness of fit of a data distribution, and the level of confidence in the variance and standard deviation of a random variable with a normal distribution.

### Q.42 What is the difference between z-test, F-test, and t-test?

The z-test, t-test, and F-test are all statistical hypothesis tests used in different situations and for different purposes. Here’s a overview of each test and the key differences between them.

| **z-test** | **t-test** | **F-test** |
| --- | --- | --- |
| When we want to compare a sample mean to a known population mean and we know the population standard deviation, we use the z-test. | When we want to compare a sample mean to a known or assumed population mean but don’t know what the population standard deviation is we use the t-test. | The F-test is used to compare the variances of two or more samples. It is commonly used in analysis of variance (ANOVA) and regression analysis. |
| When we dealing with large sample sizes or when we known the population standard deviation it is most frequently used. | The t-test follows a t-distribution, which has different shapes depending on the degrees of freedom. | The two-sample F-test, which analyses the variances of two independent samples, is the most popular of the F-test’s variants. |
| The z-test follows a standard normal distribution when certain assumptions are met. | The sample standard deviation (s) determines the test statistic for the t-test. | One set of degrees of freedom corresponds to each sample’s degrees of freedom in the F-distribution. |

In summary, the choice between a z-test, t-test, or F-test depends on the specific research question and the characteristics of the data.

### Q.43 What is the central limit theorem, and why is it significant in statistics?

The Central Limit Theorem states that, regardless of the shape of the population distribution, the distribution of the sample means approaches a normal distribution as the sample size increases.This is true even if the population distribution is not normal. The larger the sample size, the closer the sampling distribution of the sample mean will be to a normal distribution.

### Q.44 Describe the process of hypothesis testing, including null and alternative hypotheses.

Hypothesis testing is a statistical method used to make inferences about population parameters based on sample data.It is a systematic way of evaluating statements or hypotheses about a population using observed sample data.To identify which statement is best supported by the sample data, it compares two statements about a population that are mutually exclusive.

* **Null hypothesis(H0):**The null hypothesis (H0) in statistics is the default assumption or assertion that there is no association between any two measured cases or any two groups. In other words, it is a fundamental assumption or one that is founded on knowledge of the problem.
* **Alternative hypothesis(H1)**: The alternative hypothesis, or H1, is the null-hypothesis-rejecting hypothesis that is utilised in hypothesis testing.

### Q.45 How do you calculate a confidence interval, and what does it represent?

A confidence interval (CI) is a statistical range or interval estimate for a population parameter, such as the population mean or population proportion, based on sample data. to calculate confidence interval these are the following steps.

1. Collect Sample Data
2. Choose a Confidence Level
3. Select the Appropriate Statistical Method
4. Calculate the Margin of Error (MOE)
5. Calculate the Confidence Interval
6. Interpret the Confidence Interval

Confidence interval represents a range of values within which we believe, with a specified level of confidence (e.g., 95%), that the true population parameter lies.

### Q.46 What is a p-value in Statistics?

The term “p-value,” which stands for “probability value,” is a key one in statistics and hypothesis testing. It measures the evidence contradicting a null hypothesis and aids in determining whether a statistical test’s findings are statistically significant. Here is a definition of a p-value and how it is used in hypothesis testing.

### Q.47 Explain Type I and Type II errors in hypothesis testing.

Rejecting a null hypothesis that is actually true in the population results in a type I error (false-positive); failing to reject a null hypothesis that is actually untrue in the population results in a type II error (false-negative).

type I and type II mistakes cannot be completely avoided, the investigator can lessen their risk by increasing the sample size (the less likely it is that the sample will significantly differ from the population).

### Q.48 What is the significance level (alpha) in hypothesis testing?

A crucial metric in hypothesis testing that establishes the bar for judging whether the outcomes of a statistical test are statistically significant is the significance level, which is sometimes indicated as (alpha). It reflects the greatest possible chance of committing a Type I error, or mistakenly rejecting a valid null hypothesis.

The significance level in hypothesis testing.

1. Setting the Significance Level
2. Interpreting the Significance Level
3. Hypothesis Testing Using Significance Level
4. Choice of Significance Level

### Q.49 How can you calculate the correlation coefficient between two variables?

The degree and direction of the linear link between two variables are quantified by the correlation coefficient. The Pearson correlation coefficient is the most widely used method for determining the correlation coefficient. The Pearson correlation coefficient can be calculated as follows.

1. Collect Data
2. Calculate the Means
3. Calculate the Covariance
4. Calculate the Standard Deviations
5. Calculate the Pearson Correlation Coefficient (r)
6. Interpret the Correlation Coefficient.

### Q.50 What is covariance, and how is it related to correlation?

Both covariance and correlation are statistical metrics that show how two variables are related to one another.However, they serve slightly different purposes and have different interpretations.

* **Covariance** :Covariance measures the degree to which two variables change together. It expresses how much the values of one variable tend to rise or fall in relation to changes in the other variable.
* **Correlation** : A standardised method for measuring the strength and direction of a linear relationship between two variables is correlation. It multiplies the standard deviations of the two variables to scale the covariance.

### Q.51 Explain how to perform a hypothesis test for comparing two population means.

When comparing two population means, a hypothesis test is used to determine whether there is sufficient statistical support to claim that the means of the two distinct populations differ significantly. Tests we can commonly use for include [“paired t-test" or "two -sample t test"](https://www.geeksforgeeks.org/what-is-the-differences-between-the-two-sample-t-test-and-paired-t-test). The general procedures for carrying out such a test are as follows.

1. Formulate Hypotheses
2. Choose the Significance Level
3. Collect Data
4. Define Test Statistic
5. Draw a Conclusion
6. Final Results

### Q.52 Explain the concept of normalization in database design.

By minimising data duplication and enhancing data integrity, normalisation is a method in database architecture that aids in the effective organisation of data. It include dividing a big, complicated table into smaller, associated tables while making sure that connections between data elements are preserved. The basic objective of normalisation is to reduce data anomalies, which can happen when data is stored in an unorganised way and include insertion, update, and deletion anomalies.

### **Q.53 What is database normalization?**

Database denormalization is the process of intentionally introducing redundancy into a relational database by merging tables or incorporating redundant data to enhance query performance. Unlike normalization, which minimizes data redundancy for consistency, denormalization prioritizes query speed. By reducing the number of joins required, denormalization can improve read performance for complex queries. However, it may lead to data inconsistencies and increased maintenance complexity. Denormalization is often employed in scenarios where read-intensive operations outweigh the importance of maintaining a fully normalized database structure. Careful consideration and trade-offs are essential to strike a balance between performance and data integrity.

### Q.54 Define different types of SQL functions.

SQL functions can be categorized into several types based on their functionality.

1. Scalar Functions
2. Aggregate Functions
3. Window Functions
4. Table-Valued Functions
5. System Functions
6. User-Defined Functions
7. Conversion Functions
8. Conditional Functions

### Q.55 Explain the difference between INNER JOIN and LEFT JOIN.

INNER JOIN and LEFT JOIN are two types of SQL JOIN operations used to combine data from multiple tables in a relational database. Here are the some main differences between them.

| **INNER JOIN** | **LEFT JOIN** |
| --- | --- |
| Only rows with a match in the designated columns between the two tables being connected are returned by an INNER JOIN. | LEFT JOIN returns all rows from the left table and the matching rows from the right table. |
| A row is not included in the result set if there is no match for it in either of the tables. | Columns from the right table’s rows are returned with NULL values if there is no match for that row. |
| When we want to retrieve data from both tables depending on a specific criterion, INNER JOIN can be helpful. | It makes sure that every row from the left table appears in the final product, even if there are no matches for that row in the right table. |

### Q.56 What is a subquery, and how can it be used in SQL?

A subquery is a query that is nested within another SQL query, also referred to as an inner query or nested query. On the basis of the outcomes of another query, we can use it to get data from one or more tables. SQL’s subqueries capability is employed for a variety of tasks, including data retrieval, computations, and filtering.

### Q.57 How do you perform mathematical calculations in SQL queries?

In SQL, we can perform mathematical calculations in queries using arithmetic operators and functions. Here are some common methods for performing mathematical calculations.

1. Arithmetic Operators
2. Mathematical Functions
3. Aggregate Functions
4. Custom Expressions

### Q.58 What is the purpose of the CASE statement in SQL?

The SQL CASE statement is a flexible conditional expression that may be used to implement conditional logic inside of a query. we can specify various actions or values based on predetermined criteria.

### Q.59 What is the difference between a database and a data warehouse?

**Database:**Consistency and real-time data processing are prioritised, and they are optimised for storing, retrieving, and managing structured data. Databases are frequently used for administrative functions like order processing, inventory control, and customer interactions.

**Data Warehouse:**Data warehouses are made for processing analytical data. They are designed to facilitate sophisticated querying and reporting by storing and processing massive amounts of historical data from various sources. Business intelligence, data analysis, and decision-making all employ data warehouses.

### Q.60 What is regularization in machine learning, State the differences between L1 and L2 regularization

**Regularization**: Regularization is the technique to restrict the model overfitting during training by inducing a penalty to the loss. The penalty imposed on the loss function is added so that the complexity of the model can be controlled, thus overcoming the issue of overfitting in the model.

The following are the differences between L1 and L2 regularization:

| **category** | **L1 Regularization(Lasso)** | **L2 Regularization (Ridge)** |
| --- | --- | --- |
| **Definition** | L1 regularization is the technique where the induced penalty term changes some of the terms to be exactly zero | L2 regularization is the technique where the induced penalty term changes some of the terms to be as near to zero as possible. |
| **Interpretability** | Selects a subset of most important ones while eliminating less important ones. | Selects all the features but assigns less weights to less important features. |
| **Formula** | where,  L1 = Lasso Loss function   = Model loss   = regularization controlling parameter  w = weights of the model | where,  L2 = Ridge Loss function   = Model loss   = regularization controlling parameter  w = weights of the model |
| **Robustness** | Sensitive to outliers and noisy data as it can eliminate them | More robust to the presence of Outliers and noisy data |
| **Computational efficiency** | Computationally more expensive | Computationally less expensive. |

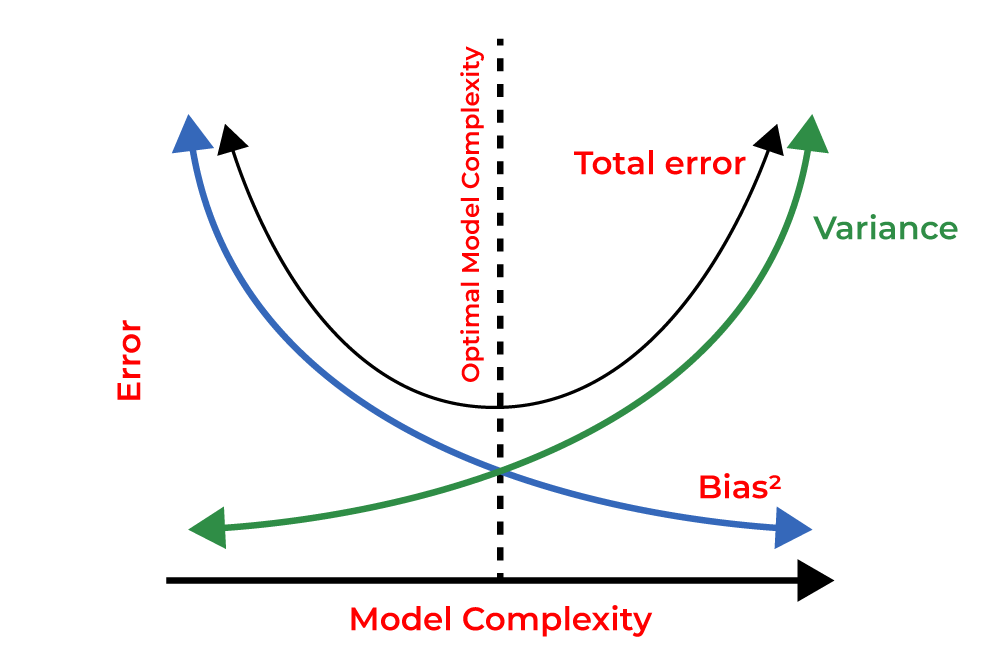
### Q.61 Explain the concepts of bias-variance trade-off in machine learning.

When creating predictive models, the bias-variance trade-off is a key concept in machine learning that deals with finding the right balance between two sources of error, bias and variance. It plays a crucial role in model selection and understanding the generalization performance of a machine learning algorithm. Here’s an explanation of these concepts:

* [**Bias**](https://www.geeksforgeeks.org/bias-vs-variance-in-machine-learning):Bias is simply described as the model’s inability to forecast the real value due of some difference or inaccuracy. These differences between actual or expected values and the predicted values are known as error or bias error or error due to bias.
* [**Variance**](https://www.geeksforgeeks.org/bias-vs-variance-in-machine-learning): Variance is a measure of data dispersion from its mean location. In machine learning, variance is the amount by which a predictive model’s performance differs when trained on different subsets of the training data. More specifically, variance is the model’s variability in terms of how sensitive it is to another subset of the training dataset, i.e. how much it can adapt on the new subset of the training dataset.

|  | **Low Bias** | **High Bias** |
| --- | --- | --- |
| **Low Variance** | Best fit (Ideal Scenario ) | Underfitting |
| **High Variance** | Overfitting | Not capture the underlying patterns (Worst Case) |

As a Data Science Professional, Our focus should be to achieve the the best fit model i.e Low Bias and Low Variance. A model with low bias and low variance suggests that it can capture the underlying patterns in the data (low bias) and is not overly sensitive to changes in the training data (low variance). This is the perfect circumstance for a machine learning model, since it can generalize effectively to new, previously unknown data and deliver consistent and accurate predictions. However, in practice, this is not achievable.



If the algorithm is too simplified (hypothesis with linear equation), it may be subject to high bias and low variance, making it error-prone. If algorithms fit too complicated a hypothesis (hypothesis with a high degree equation), it may have a large variance and a low bias. In the latter case, the new entries will underperform. There is, however, something in between these two situations called as a Trade-off or [Bias Variance Trade-off](https://www.geeksforgeeks.org/bias-vs-variance-in-machine-learning). So, that An algorithm can’t be more complex and less complex at the same time.

### Q.62 How do we choose the appropriate kernel function in SVM?

A kernel function is responsible for converting the original data points into a high dimensionality feature space. Choosing the appropriate kernel function in a Support Vector Machine is a crucial step, as it determines how well the SVM can capture the underlying patterns in your data. Below mentioned are some of the ways to choose the suitable kernel function:

* If the dataset exhibits linear relationship

In this case, we should use Linear Kernel function. It is simple, computationally efficient and less prone to overfitting. For example, text classification, sentiment analysis, etc.

* If the dataset requires probabilistic approach

The sigmoid kernel is suitable when the data resembles a sigmoid function or when you have prior knowledge suggesting this shape. For example, Risk assessment, Financial applications, etc.

* If the dataset is Simple Non Linear in nature

In this case, use a Polynomial Kernel Function. Polynomial functions are useful when we are trying to capture moderate level of non linearity. For example, Image and Speech Recognition, etc.

* If the dataset is Highly Non-Linear in Nature/ we do not know about the underlying relationship

In that case, a Radial basis function is the best choice. RBF kernel can handle highly complex dataset and is useful when you’re unsure about the data’s underlying distribution. For example, Financial forecasting, bioinformatics, etc.

### Q.63 How does Naïve Bayes handle categorical and continuous features?

Naive Bayes is probabilistic approach which assumes that the features are independent of each other. It calculates probabilities associated with each class label based on the observed frequencies of feature values within each class in the training data. This is done by finding the conditional probability of Feature given a class. (i.e., P(feature | class)). To make predictions on categorical data, Naive Bayes calculates the posterior probability of each class given the observed feature values and selects the class with the highest probability as the predicted class label. This is called as “maximum likelihood” estimation.

### Q.64 What is Laplace smoothing (add-one smoothing) and why is it used in Naïve Bayes?

In Naïve Bayes, the conditional probability of an event given a class label is determined as P(event| class). When using this in a classification problem (let’s say a text classification), there could a word which did not appear in the particular class. In those cases, the probability of feature given a class label will be zero. This could create a big problem when getting predictions out of the training data.

To overcome this problem, we use Laplace smoothing. Laplace smoothing addresses the zero probability problem by adding a small constant (usually 1) to the count of each feature in each class and to the total count of features in each class. Without smoothing, if any feature is missing in a class, the probability of that class given the features becomes zero, making the classifier overly confident and potentially leading to incorrect classifications

### Q.65 What are imbalanced datasets and how can we handle them?

Imbalanced datasets are datasets in which the distribution of class labels (or target values) is heavily skewed, meaning that one class has significantly more instances than any other class. Imbalanced datasets pose challenges because models trained on such data can have a bias toward the majority class, leading to poor performance on the minority class, which is often of greater interest. This will lead to the model not generalizing well on the unseen data.

To handle imbalanced datasets, we can approach the following methods:

* **Resampling (Method of either increasing or decreasing the number of samples)**:
  + **Up-sampling**: In this case, we can increase the classes for minority by either sampling without replacement or generating synthetic examples. Some of the popular examples are SMOTE (Synthetic Minority Over-sampling Technique), etc.
  + **Down-sampling**: Another case would be to randomly cut down the majority class such that it is comparable to minority class.
* **Ensemble methods (using models which are capable of handling imbalanced dataset inherently:**
  + **Bagging**: Techniques like Random Forests, which can mitigate the impact of class imbalance by constructing multiple decision trees from bootstrapped samples
  + **Boosting**: Algorithms like AdaBoost and XGBoost can give more importance to misclassified minority class examples in each iteration, improving their representation in the final model

### Q.66 What are outliers in the dataset and how can we detect and remove them?

An Outlier is a data point that is significantly different from other data points. Usually, Outliers are present in the extremes of the distribution and stand out as compared to their out data point counterparts.

For detecting Outliers we can use the following approaches:

* **Visual inspection:** This is the easiest way which involves plotting the data points into scatter plot/box plot, etc.
* **statistics**: By using measure of central tendency, we can determine if a data point falls significantly far from its mean, median, etc. making it a potential outlier.
* **Z-score:** if a data point has very high Z-score, it can be identified as Outlier

For removing the outliers, we can use the following:

* Removal of outliers manually
* Doing transformations like applying logarithmic transformation or square rooting the outlier
* Performing imputations wherein the outliers are replaced with different values like mean, median, mode, etc.

### Q.67 What is the curse of dimensionality And How can we overcome this?

When dealing with a dataset that has high dimensionality (high number of features), we are often encountered with various issues and problems. Some of the issues faced while dealing with dimensionality dataset are listed below:

* **Computational expense**: The biggest problem with handling a dataset with vast number of features is that it takes a long time to process and train the model on it. This can lead to wastage of both time and monetary resources.
* **Data sparsity**: Many times data points are far from each other (high sparsity). This makes it harder to find the underlying patterns between features and can be a hinderance in proper analysis
* **Visualising issues and overfitting**: It is rather easy to visualize 2d and 3d data. But beyond this order, it is difficult to properly visualize our data. Furthermore, more data features can be correlated and provide misleading information to the model training and cause overfitting.

These issues are what are generally termed as “Curse of Dimensionality”.

To overcome this, we can follow different approaches – some of which are mentioned below:

* **Feature Selection**: Many a times, not all the features are necessary. It is the user’s job to select out the features that would be necessary in solving a given problem statement.
* **Feature engineering**: Sometimes, we may need a feature that is the combination of many other features. This method can, in general, reduces the features count in the dataset.
* **Dimensionality Reduction techniques**: These techniques reduce the number of features in a dataset while preserving as much useful information as possible. Some of the famous Dimensionality reduction techniques are: Principle component analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), etc.
* **Regularization:**Some regularization techniques like L1 and L2 regularizations are useful when deciding the impact each feature has on the model training.

### Q.68 How does the random forest algorithm handle feature selection?

Mentioned below is how Random forest handles feature selection

* When creating individual trees in the Random Forest ensemble, a subset of features is assigned to each tree which is called Feature Bagging. Feature Bagging introduces randomness and diversity among the trees.
* After the training, the features are assigned a “importance score” based on how well those features performed by reducing the error of the model. Features that consistently contribute to improving the model’s accuracy across multiple trees are deemed more important
* Then the features are ranked based on their importance scores. Features with higher importance scores are considered more influential in making predictions.

### Q.69 What is feature engineering? Explain the different feature engineering methods.

**Feature Engineering**: It can be defined as a method of preprocessing of data for better analysis purpose which involves different steps like selection, transformation, deletion of features to suit our problem at hand. Feature Engineering is a useful tool which can be used for:

* Improving the model’s performance and Data interpretability
* Reduce computational costs
* Include hidden patterns for elevated Analysis results.

Some of the different methods of doing feature engineering are mentioned below:

* **Principle Component Analysis (PCA)** : It identifies orthogonal axes (principal components) in the data that capture the maximum variance, thereby reducing the data features.
* **Encoding** – It is a technique of converting the data to be represented a numbers with some meaning behind it. It can be done in two ways :
  + **One-Hot Encoding** – When we need to encode Nominal Categorical Data
  + **Label Encoding** – When we need to encode Ordinal Categorical Data
* **Feature Transformation**: Sometimes, we can create new columns essential for better modelling just by combining or modifying one or more columns.

### Q.70 How we will deal with the categorical text values in machine learning?

Often times, we are encountered with data that has Categorical text values. For example, male/female, first-class/second-class/third-class, etc. These Categorical text values can be divided into two types and based on that we deal with them as follows:

* If it is Categorical Nominal Data: If the data does not have any hidden order associated with it (e.g., male/female), we perform One-Hot encoding on the data to convert it into binary sequence of digits
* If it is Categorical Ordinal Data : When there is a pattern associated with the text data, we use Label encoding. In this, the numerical conversion is done based on the order of the text data. (e.g., Elementary/ Middle/ High/ Graduate,etc.)

### Q.71 What is DBSCAN and How we will use it?

Density-Based Spatial Clustering of Applications with Noise (DBSCAN), is a density-based clustering algorithm used for grouping together data points that are close to each other in high-density regions and labeling data points in low-density regions as outliers or noise. Here is how it works:

* For each data point in the dataset, DBSCAN calculates the distance between that point and all other data points
* DBSCAN identifies dense regions by connecting core points that are within each other’s predefined threshold (eps) neighborhood.
* DBSCAN forms clusters by grouping together data points that are density-reachable from one another.

### Q.72 How does the EM (Expectation-Maximization) algorithm work in clustering?

The Expectation-Maximization (EM) algorithm is a probabilistic approach used for clustering data when dealing with mixture models. EM is commonly used when the true cluster assignments are not known and when there is uncertainty about which cluster a data point belongs to. Here is how it works:

* First, the number of clusters K to be formed is specified.
* Then, for each data point, the likelihood of it belonging to each of the K clusters is calculated. This is called the Expectation (E) step
* Based on the previous step, the model parameters are updated. This is called Maximization (M) step.
* Together it is used to check for convergence by comparing the change in log-likelihood or the parameter values between iterations.
* If it converges, then we have achieved our purpose. If not, then the E-step and M-step are repeated until we reach convergence.

### Q.73 Explain the concept of silhouette score in clustering evaluation.

Silhouette score is a metric used to evaluate the quality of clusters produced by a clustering algorithm. Here is how it works:

* the average distance between the data point and all other data points in the same cluster is first calculated. Let us call this as (a)
* Then for the same data point, the average distance (b) between the data point and all data points in the nearest neighboring cluster (i.e., the cluster to which it is not assigned)
* silhouette coefficient for each data point is calculated, which given by: S = (b – a) / max(a, b)
  + if -1<S<0, it signifies that data point is closer to a neighboring cluster than to its own cluster.
  + if S is close to zero, data point is on or very close to the decision boundary between two neighboring clusters.
  + if 0<S<1, data point is well within its own cluster and far from neighboring clusters.

### Q.74 What is the relationship between eigenvalues and eigenvectors in PCA?

In Principal Component Analysis (PCA), eigenvalues and eigenvectors play a crucial role in the transformation of the original data into a new coordinate system. Let us first define the essential terms:

* **Eigen Values**: Eigenvalues are associated with each eigenvector and represent the magnitude of the variance (spread or extent) of the data along the corresponding eigenvector
* **Eigen Vectors**: Eigenvectors are the directions or axes in the original feature space along which the data varies the most or exhibits the most variance

The relationship between them is given as:

AV=λV*AV*=*λV*, where

A = Feature matrix

V = eigen vector

λ*λ* = Eigen value.

A larger eigenvalue implies that the corresponding eigenvector captures more of the variance in the data.The sum of all eigenvalues equals the total variance in the original data. Therefore, the proportion of total variance explained by each principal component can be calculated by dividing its eigenvalue by the sum of all eigenvalues

### Q.75 What is the cross-validation technique in machine learning?

Cross-validation is a resampling technique used in machine learning to assess and validate the performance of a predictive model. It helps in estimating how well a model is likely to perform on unseen data, making it a crucial step in model evaluation and selection. Cross validation is usually helpful when avoiding overfitting the model. Some of the widely known cross validation techniques are:

* **K-Fold Cross-Validation**: In this, the data is divided into K subsets, and K iterations of training and testing are performed.
* **Stratified K-Fold Cross-Validation**: This technique ensures that each fold has approximately the same proportion of classes as the original dataset (helpful in handling data imbalance)
* **Shuffle-Split Cross-Validation**: It randomly shuffles the data and splits it into training and testing sets.

### Q.76 What are the ROC and AUC, explain its significance in binary classification.

Receiver Operating Characteristic (ROC) is a graphical representation of a binary classifier’s performance. It plots the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds.

True positive rate (TPR) : It is the ratio of true positive predictions to the total actual positives.

Recall = TP / (TP + FN)

False positive rate (FPR) : It is the ratio of False positive predictions to the total actual positives.

FPR= FP / (TP + FN)

Area Under the Curve (AUC) as the name suggests is the area under the ROC curve. The AUC is a scalar value that quantifies the overall performance of a binary classification model and ranges from 0 to 1, where a model with an AUC of 0.5 indicates random guessing, and an AUC of 1 represents a perfect classifier.

### Q.77 Describe gradient descent and its role in optimizing machine learning models.

Gradient descent is a fundamental optimization algorithm used to minimize a cost or loss function in machine learning and deep learning. Its primary role is to iteratively adjust the parameters of a machine learning model to find the values that minimize the cost function, thereby improving the model’s predictive performance. Here’s how Gradient descent help in optimizing Machine learning models:

1. **Minimizing Cost functions**: The primary goal of gradient descent is to find parameter values that result in the lowest possible loss on the training data.
2. **Convergence**: The algorithm continues to iterate and update the parameters until it meets a predefined convergence criterion, which can be a maximum number of iterations or achieving a desired level of accuracy.
3. **Generalization**: Gradient descent ensure that the optimized model generalizes well to new, unseen data.

### Q.78 Describe batch gradient descent, stochastic gradient descent, and mini-batch gradient descent.

**Batch Gradient Descent:**In Batch Gradient Descent, the entire training dataset is used to compute the gradient of the cost function with respect to the model parameters (weights and biases) in each iteration. This means that all training examples are processed before a single parameter update is made. It converges to a more accurate minimum of the cost function but can be slow, especially in a high dimensionality space.

**Stochastic Gradient Descent:**In Stochastic Gradient Descent, only one randomly selected training example is used to compute the gradient and update the parameters in each iteration. The selection of examples is done independently for each iteration. This is capable of faster updates and can handle large datasets because it processes one example at a time but high variance can cause it to converge slower.

**Mini-Batch Gradient Descent:**Mini-Batch Gradient Descent strikes a balance between BGD and SGD. It divides the training dataset into small, equally-sized subsets called mini-batches. In each iteration, a mini-batch is randomly sampled, and the gradient is computed based on this mini-batch. It utilizes parallelism well and takes advantage of modern hardware like GPUs but can still exhibits some level of variance in updates compared to Batch Gradient Descent.

### Q.79 Explain the Apriori — Association Rule Mining

Association Rule mining is an algorithm to find relation between two or more different objects. Apriori association is one of the most frequently used and most simple association technique. Apriori Association uses prior knowledge of frequent objects properties. It is based on Apriori property which states that:

*“All non-empty subsets of a frequent itemset must also be frequent”*

## *Data Science Interview Questions for Experienced*

### Q.80 Explain multivariate distribution in data science.

A vector with several normally distributed variables is said to have a multivariate normal distribution if any linear combination of the variables likewise has a normal distribution. The multivariate normal distribution is used to approximatively represent the features of specific characteristics in machine learning, but it is also important in extending the central limit theorem to several variables.

### Q.81 Describe the concept of conditional probability density function (PDF).

In probability theory and statistics, the conditional probability density function (PDF) is a notion that represents the probability distribution of a random variable within a certain condition or constraint. It measures the probability of a random variable having a given set of values given a set of circumstances or events.

### Q.82 What is the cumulative distribution function (CDF), and how is it related to PDF?

The probability that a continuous random variable will take on particular values within a range is described by the Probability Density Function (PDF), whereas the Cumulative Distribution Function (CDF) provides the cumulative probability that the random variable will fall below a given value. Both of these concepts are used in probability theory and statistics to describe and analyse probability distributions. The PDF is the CDF’s derivative, and they are related by integration and differentiation.

### Q.83 What is ANOVA? What are the different ways to perform ANOVA tests?

The statistical method known as ANOVA, or Analysis of Variance, is used to examine the variation in a dataset and determine whether there are statistically significant variations between group averages. When comparing the means of several groups or treatments to find out if there are any notable differences, this method is frequently used.

There are several different ways to perform ANOVA tests, each suited for different types of experimental designs and data structures:

1. One-Way ANOVA
2. Two-Way ANOVA
3. Three-Way ANOVA

When conducting ANOVA tests we typically calculate an F-statistic and compare it to a critical value or use it to calculate a p-value.

### **Q.84 How can you prevent gradient descent from getting stuck in local minima?**

**Ans:**The local minima problem occurs when the optimization algorithm converges a solution that is minimum within a small neighbourhood of the current point but may not be the global minimum for the objective function.

To mitigate local minimal problems, we can use the following technique:

1. Use initialization techniques like Xavier/Glorot and He to model trainable parameters. This will help to set appropriate initial weights for the optimization process.
2. Set Adam or RMSProp as optimizer, these adaptive learning rate algorithms can adapt the learning rates for individual parameters based on historical gradients.
3. Introduce stochasticity in the optimization process using mini-batches, which can help the optimizer to escape local minima by adding noise to the gradient estimates.
4. Adding more layers or neurons can create a more complex loss landscape with fewer local minima.
5. Hyperparameter tuning using random search cv and grid search cv helps to explore the parameter space more thoroughly suggesting right hyperparameters for training and reducing the risk of getting stuck in local minima.

### **Q.85 Explain the Gradient Boosting algorithms in machine learning.**

Gradient boosting techniques like XGBoost, and CatBoost are used for regression and classification problems. It is a boosting algorithm that combines the predictions of weak learners to create a strong model. The key steps involved in gradient boosting are:

1. Initialize the model with weak learners, such as a decision tree.
2. Calculate the difference between the target value and predicted value made by the current model.
3. Add a new weak learner to calculate residuals and capture the errors made by the current ensemble.
4. Update the model by adding fraction of the new weak learner’s predictions. This updating process can be controlled by learning rate.
5. Repeat the process from step 2 to 4, with each iteration focusing on correcting the errors made by the previous model.

### Q.86 **Explain convolutions operations of CNN architecture?**

In a CNN architecture, convolution operations involve applying small filters (also called kernels) to input data to extract features. These filters slide over the input image covering one small part of the input at a time, computing dot products at each position creating a feature map. This operation captures the similarity between the filter’s pattern and the local features in the input. Strides determine how much the filter moves between positions. The resulting feature maps capture patterns, such as edges, textures, or shapes, and are essential for image recognition tasks. Convolution operations help reduce the spatial dimensions of the data and make the network translation-invariant, allowing it to recognize features in different parts of an image. Pooling layers are often used after convolutions to further reduce dimensions and retain important information.

### Q.87 **What is feed forward network and how it is different from recurrent neural network?**

Deep learning designs that are basic are feedforward neural networks and recurrent neural networks. They are both employed for different tasks, but their structure and how they handle sequential data differ.

**Feed Forward Neural Network**

* In FFNN, the information flows in one direction, from input to output, with no loops
* It consists of multiple layers of neurons, typically organized into an input layer, one or more hidden layers, and an output layer.
* Each neuron in a layer is connected to every neuron in the subsequent layer through weighted connections.
* FNNs are primarily used for tasks such as classification and regression, where they take a fixed-size input and produce a corresponding output

**Recurrent Neural Network**

* A recurrent neural network is designed to handle sequential data, where the order of input elements matters. Unlike FNNs, RNNs have connections that loop back on themselves, allowing them to maintain a hidden state that carries information from previous time steps.
* This hidden state enables RNNs to capture temporal dependencies and context in sequential data, making them well-suited for tasks like natural language processing, time series analysis, and sequence generation.
* However, standard RNNs have limitations in capturing long-range dependencies due to the vanishing gradient problem.

### **Q.88 Explain the difference between generative and discriminative models?**

Generative models focus on generating new data samples, while discriminative models concentrate on classification and prediction tasks based on input data.

**Generative Models:**

* Objective: Model the joint probability distribution P(X, Y) of input X and target Y.
* Use: Generate new data, often for tasks like image and text generation.
* Examples: Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs).

**Discriminative Models:**

* Objective: Model the conditional probability distribution P(Y | X) of target Y given input X.
* Use: Classify or make predictions based on input data.
* Examples: Logistic Regression, Support Vector Machines, Convolutional Neural Networks (CNNs) for image classification.

### **Q.89 What is the forward and backward propogations in deep learning?**

Forward and backward propagations are key processes that occur during neural network training in deep learning. They are essential for optimizing network parameters and learning meaningful representations from input.

The process by which input data is passed through the neural network to generate predictions or outputs is known as forward propagation. The procedure begins at the input layer, where data is fed into the network. Each neuron in a layer calculates the weighted total of its inputs, applies an activation function, and sends the result to the next layer. This process continues through the hidden layers until the final output layer produces predictions or scores for the given input data.

The technique of computing gradients of the loss function with regard to the network’s parameters is known as backward propagation. It is utilized to adjust the neural network parameters during training using optimization methods such as gradient descent.

The process starts with the computation of the loss, which measures the difference between the network’s predictions and the actual target values. Gradients are then computed by using the chain rule of calculus to propagate this loss backward through the network. This entails figuring out how much each parameter contributed to the error. The computed gradients are used to adjust the network’s weights and biases, reducing the error in subsequent forward passes.

### **Q.90 Describe the use of Markov models in sequential data analysis?**

Markov models are effective methods for capturing and modeling dependencies between successive data points or states in a sequence. They are especially useful when the current condition is dependent on earlier states. The Markov property, which asserts that the future state or observation depends on the current state and is independent of all prior states. There are two types of Markov models used in sequential data analysis:

* Markov chains are the simplest form of Markov models, consisting of a set of states and transition probabilities between these states. Each state represents a possible condition or observation, and the transition probabilities describe the likelihood of moving from one state to another.
* Hidden Markov Models extend the concept of Markov chains by introducing a hidden layer of states and observable emissions associated with each hidden state. The true state of the system (hidden state) is not directly observable, but the emissions are observable.

**Applications:**

* HMMs are used to model phonemes and words in speech recognition systems, allowing for accurate transcription of spoken language
* HMMs are applied in genomics for gene prediction and sequence alignment tasks. They can identify genes within DNA sequences and align sequences for evolutionary analysis.
* Markov models are used in modeling financial time series data, such as stock prices, to capture the dependencies between consecutive observations and make predictions.

### **Q.91 What is generative AI?**

Generative AI is an abbreviation for Generative Artificial Intelligence, which refers to a class of artificial intelligence systems and algorithms that are designed to generate new, unique data or material that is comparable to, or indistinguishable from, human-created data. It is a subset of artificial intelligence that focuses on the creative component of AI, allowing machines to develop innovative outputs such as writing, graphics, audio, and more. There are several generative AI models and methodologies, each adapted to different sorts of data and applications such as:

1. Generative AI models such as GPT (Generative Pretrained Transformer) can generate human-like text.” Natural language synthesis, automated content production, and chatbot responses are all common uses for these models.
2. Images are generated using generative adversarial networks (GANs).” GANs are made up of a generator network that generates images and a discriminator network that determines the authenticity of the generated images. Because of the struggle between the generator and discriminator, high-quality, realistic images are produced.
3. Generative AI can also create audio content, such as speech synthesis and music composition.” Audio content is generated using models such as WaveGAN and Magenta.

### **Q.92 What are different neural network architecture used to generate artificial data in deep learning?**

Various neural networks are used to generate artificial data. Here are some of the neural network architectures used for generating artificial data:

1. GANs consist of two components – generator and discriminator, which are trained simultaneously through adversarial training. They are used to generating high-quality images, such as photorealistic faces, artwork, and even entire scenes.
2. VAEs are generative models that learn a probabilistic mapping from the data space to a latent space. They also consist of encoder and decoder. They are used for generating images, reconstructing missing parts of images, and generating new data samples. They are also applied in generating text and audio.
3. RNNs are a class of neural networks with recurrent connections that can generate sequences of data. They are often used for sequence-to-sequence tasks. They are used in text generation, speech synthesis, music composition.
4. Transformers are a type of neural network architecture that has gained popularity for sequence-to-sequence tasks. They use self-attention mechanisms to capture dependencies between different positions in the input data. They are used in natural language processing tasks like machine translation, text summarization, and language generation.
5. Autoencoders are neural networks that are trained to reconstruct their input data. Variants like denoising autoencoders and contractive autoencoders can be used for data generation. They are used for image denoising, data inpainting, and generating new data samples.

### **Q.93 What is deep reinforcement learning technique?**

Deep Reinforcement Learning (DRL) is a cutting-edge machine learning technique that combines the principles of reinforcement learning with the capability of deep neural networks. Its ability to enable machines to learn difficult tasks independently by interacting with their environments, similar to how people learn via trial and error, has garnered significant attention.

**DRL is made up of three fundamental components:**

1. The agent interacts with the environment and takes decision.
2. The environment is the outside world with which the agent interacts and receives feedback.
3. The reward signal is a scalar value provided by the environment after each action, guiding the agent toward maximizing cumulative rewards over time.

**Applications:**

1. In robotics, DRL is used to control robots, manipulation and navigation.
2. DRL plays a role in self-driving cars and vehicle control
3. Can also be used for customized recommendations

### **Q.94 What is transfer learning, and how is it applied in deep learning?**

Transfer learning is a strong machine learning and deep learning technique that allows models to apply knowledge obtained from one task or domain to a new, but related. It is motivated by the notion that what we learn in one setting can be applied to a new, but comparable, challenge.

**Benefits of Transfer Learning:**

* We may utilize knowledge from a large dataset by starting with a pretrained model, making it easier to adapt to a new task with data.
* Training a deep neural network from scratch can be time-consuming and costly in terms of compute. Transfer learning enables us to bypass the earliest phases of training, saving both time and resources.
* Pretrained models frequently learn rich data representations. Models that use these representations can generalize better, even when the target task has a smaller dataset.

**Transfer Learning Process:**

* Feature Extraction
  + It’s a foundation step in transfer learning. The pretrained data is already trained on large and diverse dataset for a related task.
  + To leverage the knowlege, output layers of the pretrained model are removed leaving the layers responsible for feature extraction. The target data is passed through these layers to extract feature information.
  + using these extracted features, the model captures patterns and representations from the data.
* Fine Tuning
  + After the feature extraction process, the model is fine-tuned for the specific target task.
  + Output layers are added to the model and these layer are designed to produce the desired output for the target task.
  + Backpropagation is used to iteratively update the model’s weights during fine-tuning. This method allows the model to tailor its representations and decision boundaries to the specifics of the target task.
  + Even as the model focuses in the target task, the knowledge and features learned from the pre-trained layers continue to contribute to its understanding. This dual learning process improves the model’s performance and enables it to thrive in tasks that require little data or resources.

### **Q.95 What is difference between object detections and image segmentations.**

Object detection and Image segmentation are both computer vision tasks that entail evaluating and comprehending image content, but they serve different functions and give different sorts of information.

**Object Detection:**

* goal of object detection is to identify and locate objects and represent the object in bounding boxes with their respective labels.
* used in applications like autonomous driving for detecting pedestrians and vehicle

**Image Segmentation:**

* focuses on partitioning an image into multiple regions, where each segment corresponding to a coherent part of the image.
* provide pixel level labeling of the entire image
* used in applications that require pixel level understanding such as medical image analysis for organ and tumor delineation.

### **Q.96 Explain the concept of word embeddings in natural language processing (NLP).**

In NLP, the concept of word embedding is use to capture semantic and contextual information. Word embeddings are dense representations of words or phrases in continuous-valued vectors in a high-dimensional space. Each word is mapped to a vector with the real numbers, these vectors are learned from large corpora of text data.

Word embeddings are based on the Distributional Hypothesis, which suggests that words that appear in similar context have similar meanings. This idea is used by word embedding models to generate vector representations that reflect the semantic links between words depending on how frequently they co-occur with other words in the text.

The most common word embeddings techniques are-

* Bag of Words (BOW)
* Word2Vec
* Glove: Global Vector for word representation
* Term frequency-inverse document frequency (TF-IDF)
* BERT

### **Q.97 What is seq2seq model?**

A neural network architecture called a Sequence-to-Sequence (Seq2Seq) model is made to cope with data sequences, making it particularly helpful for jobs involving variable-length input and output sequences. Machine translation, text summarization, question answering, and other tasks all benefit from its extensive use in natural language processing.

The Seq2Seq consists of two main components: encoder and decoder. The encoder takes input sequence and converts into fixed length vector . The vector captures features and context of the sequence. The decoder takes the vector as input and generated output sequence. This autoregressive technique frequently entails influencing the subsequent prediction using the preceding one.

### Q.98 **What is artificial neural networks.**

Artificial neural networks take inspiration from structure and functioning of human brain. The computational units in ANN are called neurons and these neurons are responsible to process and pass the information to the next layer.

ANN has three main components:

* **Input Layer**: where the network receives input features.
* **Hidden Layer:**one or more layers of interconnected neurons responsible for learning patterns in the data
* **Output Layer**: provides final output on processed information.

### Q.99 What is marginal probability?

A key idea in statistics and probability theory is marginal probability, which is also known as marginal distribution. With reference to a certain variable of interest, it is the likelihood that an event will occur, without taking into account the results of other variables. Basically, it treats the other variables as if they were "marginal" or irrelevant and concentrates on one.

Marginal probabilities are essential in many statistical analyses, including estimating anticipated values, computing conditional probabilities, and drawing conclusions about certain variables of interest while taking other variables' influences into account.

### Q.100 What are the probability axioms?

The fundamental rules that control the behaviour and characteristics of probabilities in probability theory and statistics are referred to as the probability axioms, sometimes known as the probability laws or probability principles.

**There are three fundamental axioms of probability:**

1. Non-Negativity Axiom
2. Normalization Axiom
3. Additivity Axiom

## 8. Computer Vision Interview Questions

# [https://www.geeksforgeeks.org/computer-vision-interview-questions/#](https://www.geeksforgeeks.org/computer-vision-interview-questions/)

# (Top 50 Computer Vision Interview Questions

Last Updated : 18 Dec, 2024)

Computer vision

is a field of[artificial intelligence](https://www.geeksforgeeks.org/artificial-intelligence-an-introduction/) that enables machines to interpret and understand visual information from the world. It encompasses a wide range of tasks such as image classification, object detection, image segmentation, and image generation. As the demand for advanced computer vision applications grows, so does the need for skilled professionals who can develop and implement these technologies effectively.

### 1. What do you mean by Digital Image?

A digital image is a representation of a two-dimensional image using binary values. It is composed of pixels, which are the smallest units of the image, arranged in a grid. Each pixel has a numerical value that represents its color or intensity, allowing the image to be displayed and processed by digital devices.

### 2. How do neural networks distinguish between useful and non-useful features in computer vision?

Neural networks distinguish between useful and non-useful features through a process called feature learning. During training, the network adjusts its weights based on the input data and the desired output. Useful features are those that help the network minimize the loss function and improve accuracy. These features are emphasized through higher weights, while non-useful features have their weights reduced.

### 3. [How do convolutional neural networks (CNNs) work?](https://www.geeksforgeeks.org/how-do-convolutional-neural-networks-cnns-work/)

CNNs work by applying convolution operations to input data, typically images, to extract features. A CNN consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to detect features such as edges, textures, and shapes. Pooling layers reduce the dimensionality of the data, making the network computationally efficient. Fully connected layers combine the features to make predictions.

### 4. [Explain the concept of transfer learning and its application in computer vision.](https://www.geeksforgeeks.org/explain-the-concept-of-transfer-learning-and-its-application-in-computer-vision/)

[Transfer learning](https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning/) is a technique where a pre-trained model on a large dataset is fine-tuned for a specific task on a smaller dataset. In computer vision, this allows leveraging the knowledge gained from a model trained on a large dataset like ImageNet to perform tasks such as image classification, object detection, or segmentation on a smaller dataset, improving performance and reducing training time.

### 5.[What are some common computer vision libraries and frameworks?](https://www.geeksforgeeks.org/what-are-some-common-computer-vision-libraries-and-frameworks/)

Common computer vision libraries and frameworks include:

* [OpenCV](https://www.geeksforgeeks.org/opencv-python-tutorial/)
* [TensorFlow](https://www.geeksforgeeks.org/opencv-python-tutorial/)
* [Keras](https://www.geeksforgeeks.org/how-to-create-models-in-keras/)
* [PyTorch](https://www.geeksforgeeks.org/getting-started-with-pytorch/)
* scikit-image
* MATLAB Image Processing Toolbox

### 6. [How do you handle overfitting in computer vision models?](https://www.geeksforgeeks.org/how-to-handle-overfitting-in-computer-vision-models/)

To handle overfitting in computer vision models, several techniques can be used:

* Data augmentation: Increasing the diversity of training data by applying transformations.
* Regularization: Techniques like L2 regularization to penalize large weights.
* Dropout: Randomly dropping neurons during training to prevent co-adaptation.
* Cross-validation: Using multiple folds for validation to ensure model generalization.
* Early stopping: Halting training when performance on validation data stops improving.

### 7. [What are the different Image denoising techniques in computer vision?](https://www.geeksforgeeks.org/what-are-the-different-image-denoising-techniques-in-computer-vision/)

Image denoising techniques include:

* Gaussian Filtering: Reduces noise using a Gaussian kernel.
* Median Filtering: Replaces each pixel with the median of the surrounding pixels.
* Bilateral Filtering: Smoothens images while preserving edges.
* Non-Local Means: Averages pixels with similar patches.
* Wavelet Transform: Decomposes the image and removes noise in the frequency domain.

### 8. [Briefly explain the different image preprocessing algorithms used in computer vision](https://www.geeksforgeeks.org/image-processing-algorithms-in-computer-vision/).

Image preprocessing algorithms include:

* Normalization: Adjusting pixel values to a common scale.
* Resizing: Changing the dimensions of an image.
* Cropping: Extracting a region of interest from an image.
* Histogram Equalization: Enhancing contrast.
* Smoothing/Blurring: Reducing noise using filters like Gaussian blur.
* Edge Detection: Highlighting edges using algorithms like Sobel or Canny.

### 9. [What are the different Image Thresholding techniques in computer vision?](https://www.geeksforgeeks.org/image-thresholding-techniques-in-computer-vision/)

Image thresholding techniques include:

* Global Thresholding: Applying a single threshold value to all pixels.
* Adaptive Thresholding: Applying different thresholds to different regions based on local statistics.
* Otsu's Method: Automatically determining the optimal global threshold by minimizing intra-class variance.

### 10.[What do you mean by morphological operations? Briefly explain the different morphological operations.](https://www.geeksforgeeks.org/different-morphological-operations-in-image-processing/)

Morphological operations are image processing techniques that process images based on shapes. They include:

* Erosion: Removes pixels on object boundaries.
* Dilation: Adds pixels to object boundaries.
* Opening: Erosion followed by dilation, removes small objects.
* Closing: Dilation followed by erosion, closes small holes.
* Morphological Gradient: Difference between dilation and erosion, highlights edges.

### 11. Which machine learning algorithms are available for use within OpenCV?

OpenCV includes machine learning algorithms such as:

* k-Nearest Neighbors (k-NN)
* [Support Vector Machines](https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/) (SVM)
* [Decision Trees](https://www.geeksforgeeks.org/decision-tree/)
* Random Forests
* [k-Means Clustering](https://www.geeksforgeeks.org/k-means-clustering-introduction/)
* Boosting algorithms (e.g., AdaBoost)

### [12. What do you mean by data augmentation? How does it work for images?](https://www.geeksforgeeks.org/what-is-data-augmentation-how-does-data-augmentation-work-for-images/)

[Data augmentation](https://www.geeksforgeeks.org/what-is-data-augmentation-how-does-data-augmentation-work-for-images/) is a technique to artificially increase the size and diversity of a training dataset by applying transformations such as rotation, scaling, translation, flipping, and adding noise. For images, this helps in improving the robustness and generalization of computer vision models by providing varied examples during training.

### 13. [What are the main steps in a typical Computer Vision Pipeline?](https://www.geeksforgeeks.org/what-are-the-main-steps-in-a-typical-computer-vision-pipeline/)

A typical Computer Vision Pipeline includes:

1. Image Acquisition: Capturing or collecting image data.
2. Image Preprocessing: Normalizing, resizing, and enhancing images.
3. Feature Extraction: Identifying relevant features or patterns.
4. Model Training: Training a machine learning model using extracted features.
5. Model Evaluation: Assessing the model's performance using metrics.
6. Prediction: Applying the trained model to new data for inference.

### 14. What is the difference between Feature extraction and Feature detections?

Feature extraction involves identifying and extracting relevant features from an image for further processing. Feature detection, on the other hand, focuses on locating specific features or key points within an image, such as edges, corners, or blobs.

### 15. [How do you decide whether to utilize grayscale or color images as input for computer vision tasks?](https://www.geeksforgeeks.org/how-do-you-decide-whether-to-utilize-grayscale-or-colour-images-as-input-for-computer-vision-tasks/)

The decision to use grayscale or color images depends on the task. Grayscale images are often sufficient for tasks like edge detection or shape analysis, where color information is not crucial. Color images are preferred for tasks where color plays a significant role, such as object recognition, scene understanding, and segmentation.

### 16. [What are some techniques for image registration?](https://www.geeksforgeeks.org/what-are-some-techniques-for-image-registration/)

Image registration techniques include:

* Feature-based methods: Using key points, descriptors, and matching algorithms (e.g., SIFT, SURF).
* Intensity-based methods: Using image intensity values and[optimization techniques.](https://www.geeksforgeeks.org/optimization-techniques-set-1-modulus/)
* Transform models: Affine, projective, and spline transformations.
* Optimization methods:[Gradient descent,](https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/) least squares, and mutual information.

### 17. What is the difference between deconvolution and transposed convolutions?

Deconvolution, or inverse convolution, aims to reverse the effects of convolution, often used for image restoration. Transposed convolutions, also known as upsampling or deconvolution layers in neural networks, are used to increase the spatial resolution of feature maps, commonly in image generation and segmentation tasks.

### 18. Discuss different models used for object detection and image segmentation.

Models for object detection include:

* R-CNN (Region-based Convolutional Neural Networks)
* YOLO (You Only Look Once)
* SSD (Single Shot MultiBox Detector)
* Faster R-CNN

Models for image segmentation include:

* U-Net
* SegNet
* Mask R-CNN
* DeepLab

### 19. What are the different neural network-based architectures widely used for image generation?

Neural network-based architectures for image generation include:

* GANs (Generative Adversarial Networks)
* Conditional GANs (cGANs)
* Variational Autoencoders (VAEs)
* PixelCNN
* Diffusion Models

### 20. Explain the concept of feature matching in computer vision.

Feature matching involves identifying corresponding features or key points between two or more images. It typically involves detecting features using algorithms like SIFT or ORB, describing them using descriptors, and then matching them using distance metrics like Euclidean distance or Hamming distance.

### 21. [What is the difference between SIFT and SURF?](https://www.geeksforgeeks.org/what-is-the-difference-between-sift-and-surf/)

SIFT (Scale-Invariant Feature Transform) and SURF (Speeded-Up Robust Features) are both feature detection and description algorithms. SIFT is known for its robustness and accuracy, while SURF is faster but less precise. SIFT is scale and rotation-invariant, and SURF is also scale and rotation-invariant but optimized for speed.

### 22. [What is the difference between Object Localization, Object Recognition, and Object Detection?](https://www.geeksforgeeks.org/what-is-the-difference-between-object-localization-object-recognition-and-object-detection/)

* Object Localization: Identifying the location of an object within an image.
* Object Recognition: Identifying the type or[class](https://www.geeksforgeeks.org/object-oriented-programming-in-python-set-1-class-and-its-members/) of an object in an image.
* Object Detection: Combining localization and recognition to identify and locate multiple objects within an image.

### 23. [How do you evaluate the performance of a computer vision model?](https://www.geeksforgeeks.org/evaluation-of-computer-vision-model/)

Performance evaluation metrics for computer vision models include:

* Accuracy: Correct predictions out of total predictions.
* Precision: True positives out of predicted positives.
* Recall: True positives out of actual positives.
* F1 Score: Harmonic mean of precision and recall.
* Intersection over[Union](https://www.geeksforgeeks.org/union-c/) (IoU): Overlap between predicted and ground truth bounding boxes.
* Mean Average Precision (mAP): Average precision across different recall levels.

### 24.[What are some common loss functions used in training computer vision models?](https://www.geeksforgeeks.org/what-are-some-common-loss-functions-used-in-training-computer-vision-models/)

Common loss functions include:

* Cross-Entropy Loss: For classification tasks.
* Mean Squared Error (MSE): For regression tasks.
* Dice Loss: For segmentation tasks.
* Focal Loss: For handling class imbalance.
* Hinge Loss: For SVMs.

### 25.[How do facial recognition systems work?](https://www.geeksforgeeks.org/how-do-facial-recognition-systems-work/)

Facial recognition systems work by:

1. Detecting faces in an image.
2. Extracting facial features or landmarks.
3. Comparing extracted features with a database of known faces.
4. Matching and identifying the face based on similarity measures.

### 26. [Discuss the difference between traditional computer vision techniques and deep learning-based approaches.](https://www.geeksforgeeks.org/difference-between-traditional-computer-vision-techniques-and-deep-learning-based-approaches/)

Traditional computer vision techniques rely on handcrafted features and algorithms to process and analyze images. Deep learning-based approaches use neural networks to automatically learn features from raw data, often resulting in better performance and adaptability to complex tasks.

### 27. Describe the Hough transform and its applications in computer vision.

The Hough transform is a feature extraction technique used to detect shapes such as lines, circles, and ellipses in images. It works by transforming points in the image space to a parameter space, where patterns are easier to detect.

### 28. What is the difference between semantic segmentations and instance segmentations in computer vision?

* Semantic Segmentation: Assigns a class label to each pixel in an image, grouping all pixels of the same class together.
* Instance Segmentation: Identifies and delineates each object instance separately, even if they belong to the same class.

### 29. [What are different evaluations metrics used to evaluate image segmentation model?](https://www.geeksforgeeks.org/what-are-different-evaluation-metrics-used-to-evaluate-image-segmentation-models/)

Evaluation metrics for image segmentation models include:

* Intersection over Union (IoU)
* Pixel Accuracy
* Mean Accuracy
* Dice Coefficient (F1 Score)
* Boundary F1 Score

### 30.[What are some popular algorithms used for image segmentation, and how do they differ in their approach?](https://www.geeksforgeeks.org/what-are-some-popular-algorithms-used-for-image-segmentation-and-how-do-they-differ-in-their-approach/)

Popular image segmentation algorithms include:

* U-Net: Uses a contracting path to capture context and a symmetric expanding path to enable precise localization.
* Mask R-CNN: Extends Faster R-CNN by adding a branch for predicting segmentation masks.
* DeepLab: Employs atrous convolution to capture multi-scale context and conditional random fields for accurate boundary localization.
* FCN (Fully Convolutional Networks): Replaces fully connected layers with convolutional layers to produce spatial maps.

### 31. What is the difference between a region-based CNN (R-CNN) and a fully convolutional network (FCN)?

* R-CNN: Combines region proposals with CNNs to classify and refine object boundaries. Suitable for object detection.
* FCN: Replaces fully connected layers with convolutional layers to generate pixel-wise predictions for segmentation tasks.

### 32. How do you handle image artefacts in computer vision?

Handling image artifacts involves:

* Preprocessing: Applying filters to reduce noise and artifacts.
* Data Augmentation: Adding synthetic artifacts during training to improve robustness.
* Postprocessing: Using techniques like inpainting or morphological operations to correct artifacts.

### 33. What is the Mach band effect, and how can this phenomenon be addressed?

The Mach band effect is an optical illusion that exaggerates the contrast between adjacent regions with different luminance, creating false edges. It can be addressed by using image processing techniques that smooth transitions between regions, such as blurring or edge-preserving filters.

### 34. What is Homography? How to estimate homography between two images?

Homography is a transformation that maps points from one plane to another, used in tasks like image stitching. To estimate homography, key points and corresponding matches between two images are identified using feature detection and matching algorithms. The homography matrix is then computed using methods like RANSAC to minimize errors.

### 35. [Describe the concept of scale-invariant feature transform (SIFT)](https://www.geeksforgeeks.org/describe-the-concept-of-scale-invariant-feature-transform-sift/).

SIFT is an algorithm used to detect and describe local features in images. It identifies key points and computes descriptors that are invariant to scale, rotation, and illumination changes, making it useful for tasks like image matching and object recognition.

### 36. What is the difference between sampling and quantization in computer visions?

* Sampling: The process of converting a continuous signal into a discrete signal by taking samples at regular intervals.
* Quantization: The process of mapping continuous amplitude values of a signal to discrete values.

### 37. [What is the difference between Linear and non-linear filters?](https://www.geeksforgeeks.org/what-is-the-difference-between-linear-and-non-linear-filters/)

* Linear Filters: Apply a linear operation to the image pixels, such as averaging or Gaussian filters, which smooth the image.
* Non-linear Filters: Apply non-linear operations, such as median filtering, which can preserve edges better while reducing noise.

### 38. [How do the image edge detections work?](https://www.geeksforgeeks.org/how-do-the-image-edge-detections-work/)

Edge detection algorithms identify the boundaries of objects within an image by detecting discontinuities in intensity. Common methods include:

* Sobel Operator: Uses gradient magnitude to detect edges.
* Canny Edge Detector: Uses a multi-stage process involving Gaussian filtering, gradient calculation, non-maximum suppression, and hysteresis thresholding.

### 39. What is the difference between Sobel and Canny edge detections?

* Sobel Edge Detection: Uses gradient operators to find edges. It is simpler and faster but less accurate.
* Canny Edge Detection: Uses a multi-step process including Gaussian smoothing, gradient calculation, non-maximum suppression, and hysteresis thresholding. It provides more accurate and reliable edge detection.

### 40. Describe the concept of histogram equalization and its applications.

Histogram equalization is a technique for enhancing the contrast of an image by redistributing pixel intensity values. It spreads out the most frequent intensity values, improving the visibility of details in both dark and bright regions. Applications include medical imaging, remote sensing, and image enhancement.

### 41. What are the different methods used to evaluate the Object detection model?

Evaluation methods for object detection models include:

* Precision and Recall: Measures of accuracy and completeness.
* F1 Score: Harmonic mean of precision and recall.
* Intersection over Union (IoU): Measures overlap between predicted and ground truth bounding boxes.
* Mean Average Precision (mAP): Average precision across different recall levels.

### 42. [What is the difference between sliding window and anchor boxes approach in Object detection?](https://www.geeksforgeeks.org/what-is-the-difference-between-sliding-window-and-anchor-boxes-approach-in-object-detection/)

* Sliding Window: Scans the image with fixed-size windows at different scales and positions. It is computationally intensive.
* Anchor Boxes: Uses predefined boxes of various sizes and aspect ratios at each position in a feature map. It is more efficient and commonly used in modern object detectors like Faster R-CNN and YOLO.

### 43. Could you describe a situation in which anchor boxes would be preferred over sliding window approach?

Anchor boxes are preferred in scenarios where computational efficiency and detection speed are crucial, such as real-time object detection in video streams. They are also more effective for detecting objects of varying sizes and aspect ratios.

### 44. What is non-maximum suppression, and why is it used in object detection?

Non-maximum suppression (NMS) is a technique used to eliminate redundant and overlapping bounding boxes in object detection. It selects the bounding box with the highest confidence score and suppresses others with significant overlap, ensuring that each detected object is represented by a single bounding box.

### 45. [How does R-CNN work for object detection?](https://www.geeksforgeeks.org/how-does-r-cnn-work-for-object-detection/)

R-CNN (Region-based Convolutional Neural Networks) works by:

1. Generating region proposals using selective search.
2. Extracting features from each proposal using a CNN.
3. Classifying each proposal and refining bounding boxes using a classifier and a regressor.

### 46.[How does YOLO work for object detection?](https://www.geeksforgeeks.org/how-does-yolo-work-for-object-detection/)

YOLO (You Only Look Once) divides the input image into a grid and predicts bounding boxes and class probabilities directly from the whole image in a single pass. It uses a single neural network for end-to-end object detection, achieving real-time performance with high accuracy.

### 47. How are generative adversarial networks (GANs) used for artificial image generation?

GANs consist of two neural networks, a generator and a discriminator, trained adversarially. The generator creates synthetic images, while the discriminator evaluates their authenticity. The generator aims to produce images indistinguishable from real ones, leading to high-quality artificial image generation.

### 48. Discuss the concept of conditional GANs and their applications in generating specific types of images.

Conditional GANs (cGANs) extend GANs by conditioning the generation process on additional information, such as class labels or input images. This allows control over the generated content, enabling applications like image-to-image translation, super-resolution, and style transfer.

### 49. [How do diffusion models use iterative processes to generate images?](https://www.geeksforgeeks.org/how-do-diffusion-models-use-iterative-processes-to-generate-images/)

Diffusion models use a gradual process of adding and removing noise to and from an image. Starting with random noise, they iteratively refine the image by denoising steps guided by learned probability distributions, eventually generating a high-quality image that resembles the target distribution.

### 50. [What is the role of noise contrastive estimation (NCE) in training diffusion models for image generation?](https://www.geeksforgeeks.org/what-is-the-role-of-noise-contrastive-estimation-nce-in-training-diffusion-models-for-image-generation/)

Noise Contrastive Estimation (NCE) is used to train diffusion models by differentiating between real and generated (noisy) images. NCE simplifies the learning process by converting it into a binary classification task, improving the efficiency and stability of training diffusion models for image generation

## 9. Natural Language Processing Interview Questions

<https://www.geeksforgeeks.org/nlp-interview-questions/>

# Top 50 NLP Interview Questions and Answers 2024 Updated

(Last Updated : 18 Oct, 2024)

**Natural Language Processing (NLP)**

**is a key area in artificial intelligence that enables computers to understand, interpret, and respond to human language. It powers technologies like chatbots, voice assistants, translation services, and sentiment analysis, transforming how we interact with machines. For those aspiring to become NLP professionals, mastering the core concepts is essential for landing a job in this field.**

## *Basic NLP Interview Questions for Fresher*

### 1. What is NLP?

NLP stands for **Natural Language Processing**. The subfield of [Artificial intelligence](https://www.geeksforgeeks.org/ai-algorithms) and computational linguistics deals with the interaction between computers and human languages. It involves developing algorithms, models, and techniques to enable machines to understand, interpret, and generate natural languages in the same way as a human does.

NLP encompasses a wide range of tasks, including language translation, sentiment analysis, text categorization, information extraction, speech recognition, and natural language understanding. NLP allows computers to extract meaning, develop insights, and communicate with humans in a more natural and intelligent manner by processing and analyzing textual input.

### 2. What are the main challenges in NLP?

The complexity and variety of human language create numerous difficult problems for the study of Natural Language Processing (NLP). The primary challenges in NLP are as follows:

* **Semantics and Meaning:** It is a difficult undertaking to accurately capture the meaning of words, phrases, and sentences. The semantics of the language, including word sense disambiguation, metaphorical language, idioms, and other linguistic phenomena, must be accurately represented and understood by NLP models.
* **Ambiguity**: Language is ambiguous by nature, with words and phrases sometimes having several meanings depending on context. Accurately resolving this ambiguity is a major difficulty for NLP systems.
* **Contextual Understanding:**Context is frequently used to interpret language. For NLP models to accurately interpret and produce meaningful replies, the context must be understood and used. Contextual difficulties include, for instance, comprehending referential statements and resolving pronouns to their antecedents.
* **Language Diversity:** NLP must deal with the world's wide variety of languages and dialects, each with its own distinctive linguistic traits, lexicon, and grammar. The lack of resources and knowledge of low-resource languages complicates matters.
* **Data Limitations and Bias:**The availability of high-quality labelled data for training NLP models can be limited, especially for specific areas or languages. Furthermore, biases in training data might impair model performance and fairness, necessitating careful consideration and mitigation.
* **Real-world Understanding:**NLP models often fail to understand real-world knowledge and common sense, which humans are born with. Capturing and implementing this knowledge into NLP systems is a continuous problem.

### 3. What are the different tasks in NLP?

Natural Language Processing (NLP) includes a wide range of tasks involving understanding, processing, and creation of human language. Some of the most important tasks in NLP are as follows:

* [Text Classification](https://www.geeksforgeeks.org/classification-of-text-documents-using-the-approach-of-naive-bayes)
* [Named Entity Recognition (NER)](https://www.geeksforgeeks.org/python-named-entity-recognition-ner-using-spacy)
* [Part-of-Speech Tagging (POS)](https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging)
* [Sentiment Analysis](https://www.geeksforgeeks.org/what-is-sentiment-analysis)
* [Language Modeling](https://www.geeksforgeeks.org/videos/what-is-language-modelling-in-nlp)
* [Machine Translation](https://www.geeksforgeeks.org/machine-translation-of-languages-in-artificial-intelligence)
* [Chatbots](https://www.geeksforgeeks.org/battle-of-ai-chatbots-which-chatbot-will-rule-the-present-and-future)
* [Text Summarization](https://www.geeksforgeeks.org/python-extractive-text-summarization-using-gensim)
* [Information Extraction](https://www.geeksforgeeks.org/difference-between-information-retrieval-and-information-extraction)
* [Text Generation](https://www.geeksforgeeks.org/text-generation-using-recurrent-long-short-term-memory-network)
* [Speech Recognition](https://www.geeksforgeeks.org/text-generation-using-recurrent-long-short-term-memory-network)

### 4. What do you mean by Corpus in NLP?

In NLP, a [corpus](https://www.geeksforgeeks.org/nlp-wordlist-corpus) is a huge collection of texts or documents. It is a structured dataset that acts as a sample of a specific language, domain, or issue. A corpus can include a variety of texts, including books, essays, web pages, and social media posts. Corpora are frequently developed and curated for specific research or NLP objectives. They serve as a foundation for developing language models, undertaking linguistic analysis, and gaining insights into language usage and patterns.

### 5. What do you mean by text augmentation in NLP and what are the different text augmentation techniques in NLP?

[Text augmentation](https://www.geeksforgeeks.org/text-augmentation-techniques-in-nlp) in NLP refers to the process that generates new or modified textual data from existing data in order to increase the diversity and quantity of training samples. Text augmentation techniques apply numerous alterations to the original text while keeping the underlying meaning.

Different text augmentation techniques in NLP include:

1. **Synonym Replacement:** Replacing words in the text with their synonyms to introduce variation while maintaining semantic similarity.
2. **Random Insertion/Deletion:**Randomly inserting or deleting words in the text to simulate noisy or incomplete data and enhance model robustness.
3. **Word Swapping:**Exchanging the positions of words within a sentence to generate alternative sentence structures.
4. **Back translation:** Translating the text into another language and then translating it back to the original language to introduce diverse phrasing and sentence constructions.
5. **Random Masking:** Masking or replacing random words in the text with a special token, akin to the approach used in masked language models like BERT.
6. **Character-level Augmentation:** Modifying individual characters in the text, such as adding noise, misspellings, or character substitutions, to simulate real-world variations.
7. **Text Paraphrasing:** Rewriting sentences or phrases using different words and sentence structures while preserving the original meaning.
8. **Rule-based Generation:** Applying linguistic rules to generate new data instances, such as using grammatical templates or syntactic transformations.

### 6. What are some common pre-processing techniques used in NLP?

[Natural Language Processing (NLP)](https://www.geeksforgeeks.org/natural-language-processing-nlp-pipeline)preprocessing refers to the set of processes and techniques used to prepare raw text input for analysis, modelling, or any other NLP tasks. The purpose of preprocessing is to clean and change text data so that it may be processed or analyzed later.

Preprocessing in NLP typically involves a series of steps, which may include:

* [Tokenization](https://www.geeksforgeeks.org/tokenize-text-using-nltk-python)
* [Stop Word Removal](https://www.geeksforgeeks.org/removing-stop-words-nltk-python)
* [Text Normalization](https://www.geeksforgeeks.org/normalizing-textual-data-with-python)
  + Lowercasing
  + Lemmatization
  + Stemming
  + Date and Time Normalization
* [Removal of Special Characters and Punctuation](https://www.geeksforgeeks.org/removing-punctuations-given-string)
* [Removing HTML Tags or Markup](https://www.geeksforgeeks.org/program-to-remove-html-tags-from-a-given-string)
* [Spell Correction](https://www.geeksforgeeks.org/correcting-words-using-nltk-in-python)
* [Sentence Segmentation](https://www.geeksforgeeks.org/python-perform-sentence-segmentation-using-spacy)

### 7. What is text normalization in NLP?

Text normalization, also known as text standardization, is the process of transforming text data into a standardized or normalized form It involves applying a variety of techniques to ensure consistency,  reduce variations, and simplify the representation of textual information.

The goal of text normalization is to make text more uniform and easier to process in Natural Language Processing (NLP) tasks. Some common techniques used in text normalization include:

* **Lowercasing**: Converting all text to lowercase to treat words with the same characters as identical and avoid duplication.
* **Lemmatization**: Converting words to their base or dictionary form, known as lemmas. For example, converting "running" to "run" or "better" to "good."
* **Stemming**: Reducing words to their root form by removing suffixes or prefixes. For example, converting "playing" to "play" or "cats" to "cat."
* **Abbreviation Expansion**: Expanding abbreviations or acronyms to their full forms. For example, converting "NLP" to "Natural Language Processing."
* **Numerical Normalization**: Converting numerical digits to their written form or normalizing numerical representations. For example, converting "100" to "one hundred" or normalizing dates.
* **Date and Time Normalization**: Standardizing date and time formats to a consistent representation.

### 8. What is tokenization in NLP?

[Tokenization](https://www.geeksforgeeks.org/tokenization-using-spacy-library) is the process of breaking down text or string into smaller units called tokens. These tokens can be words, characters, or subwords depending on the specific applications. It is the fundamental step in many natural language processing tasks such as sentiment analysis, machine translation, and text generation. etc.

Some of the most common ways of tokenization are as follows:

* **Sentence tokenization:** In Sentence tokenizations, the text is broken down into individual sentences. This is one of the fundamental steps of tokenization.
* **Word tokenization:** In word tokenization, the text is simply broken down into words. This is one of the most common types of tokenization. It is typically done by splitting the text into spaces or punctuation marks.
* **Subword tokenization:** In subword tokenization, the text is broken down into subwords, which are the smaller part of words. Sometimes words are formed with more than one word, for example, Subword i.e Sub+ word, Here sub, and words have different meanings. When these two words are joined together, they form the new word "subword", which means "a smaller unit of a word". This is often done for tasks that require an understanding of the morphology of the text, such as stemming or lemmatization.
* **Char-label tokenization:** In Char-label tokenization, the text is broken down into individual characters. This is often used for tasks that require a more granular understanding of the text such as text generation, machine translations, etc.

### 9. What is NLTK and How it's helpful in NLP?

[NLTK](https://www.geeksforgeeks.org/python-nltk-tokenize-regexp) stands for Natural Language Processing Toolkit. It is a suite of libraries and programs written in Python Language for symbolic and statistical natural language processing. It offers tokenization, stemming, lemmatization, POS tagging, Named Entity Recognization, parsing, semantic reasoning, and classification.

NLTK is a popular NLP library for Python. It is easy to use and has a wide range of features. It is also open-source, which means that it is free to use and modify.

### 10. What is stemming in NLP, and how is it different from lemmatization?

Stemming and lemmatization are two commonly used word normalization techniques in NLP, which aim to reduce the words to their base or root word. Both have similar goals but have different approaches.

In [stemming](https://www.geeksforgeeks.org/python-stemming-words-with-nltk), the word suffixes are removed using the heuristic or pattern-based rules regardless of the context of the parts of speech. The resulting stems may not always be actual dictionary words. Stemming algorithms are generally simpler and faster compared to lemmatization, making them suitable for certain applications with time or resource constraints.

In [lemmatization](https://www.geeksforgeeks.org/python-lemmatization-with-nltk), The root form of the word known as lemma, is determined by considering the word's context and parts of speech. It uses linguistic knowledge and databases (e.g., wordnet) to transform words into their root form. In this case, the output lemma is a valid word as per the dictionary. For example, lemmatizing "running" and "runner" would result in "run." Lemmatization provides better interpretability and can be more accurate for tasks that require meaningful word representations.

### 11. How does part-of-speech tagging work in NLP?

[Part-of-speech tagging](https://www.geeksforgeeks.org/part-speech-tagging-stop-words-using-nltk-python) is the process of assigning a part-of-speech tag to each word in a sentence. The POS tags represent the syntactic information about the words and their roles within the sentence.

There are three main approaches for POS tagging:

* **Rule-based POS tagging:** It uses a set of handcrafted rules to determine the part of speech based on morphological, syntactic, and contextual patterns for each word in a sentence. For example, words ending with '-ing' are likely to be a verb.
* **Statistical POS tagging:**The statistical model like Hidden Markov Model (HMMs) or Conditional Random Fields (CRFs) are trained on a large corpus of already tagged text. The model learns the probability of word sequences with their corresponding POS tags, and it can be further used for assigning each word to a most likely POS tag based on the context in which the word appears.
* **Neural network POS tagging:**The neural network-based model like RNN, LSTM, Bi-directional RNN, and transformer have given promising results in POS tagging by learning the patterns and representations of words and their context.

### 12. What is named entity recognition in NLP?

[Named Entity Recognization (NER)](https://www.geeksforgeeks.org/named-entity-recognition) is a task in natural language processing that is used to identify and classify the named entity in text. Named entity refers to real-world objects or concepts, such as persons, organizations, locations, dates, etc. NER is one of the challenging tasks in NLP because there are many different types of named entities, and they can be referred to in many different ways. The goal of NER is to extract and classify these named entities in order to offer structured data about the entities referenced in a given text.

The approach followed for Named Entity Recognization (NER) is the same as the POS tagging. The data used while training in NER is tagged with persons, organizations, locations, and dates.

### 13. What is parsing in NLP?

In NLP, [parsing](https://www.geeksforgeeks.org/difference-between-top-down-parsing-and-bottom-up-parsing) is defined as the process of determining the underlying structure of a sentence by breaking it down into constituent parts and determining the syntactic relationships between them according to formal grammar rules. The purpose of parsing is to understand the syntactic structure of a sentence, which allows for deeper learning of its meaning and encourages different downstream NLP tasks such as semantic analysis, information extraction, question answering, and machine translation. it is also known as syntax analysis or syntactic parsing.

The formal grammar rules used in parsing are typically based on Chomsky's hierarchy. The simplest grammar in the Chomsky hierarchy is regular grammar, which can be used to describe the syntax of simple sentences. More complex grammar, such as context-free grammar and context-sensitive grammar, can be used to describe the syntax of more complex sentences.

### 14. What are the different types of parsing in NLP?

In natural language processing (NLP), there are several types of parsing algorithms used to analyze the grammatical structure of sentences. Here are some of the main types of parsing algorithms:

* [**Constituency Parsing**](https://www.geeksforgeeks.org/constituency-parsing-and-dependency-parsing): Constituency parsing in NLP tries to figure out a sentence's hierarchical structure by breaking it into constituents based on a particular grammar. It generates valid constituent structures using context-free grammar. The parse tree that results represents the structure of the sentence, with the root node representing the complete sentence and internal nodes representing phrases. Constituency parsing techniques like as CKY, Earley, and chart parsing are often used for parsing. This approach is appropriate for tasks that need a thorough comprehension of sentence structure, such as semantic analysis and machine translation. When a complete understanding of sentence structure is required, constituency parsing, a classic parsing approach, is applied.
* [**Dependency Parsing**](https://www.geeksforgeeks.org/constituency-parsing-and-dependency-parsing)**:** In NLP, dependency parsing identifies grammatical relationships between words in a sentence. It represents the sentence as a directed graph, with dependencies shown as labelled arcs. The graph emphasises subject-verb, noun-modifier, and object-preposition relationships. The head of a dependence governs the syntactic properties of another word. Dependency parsing, as opposed to constituency parsing, is helpful for languages with flexible word order. It allows for the explicit illustration of word-to-word relationships, resulting in a clear representation of grammatical structure.
* [**Top-down parsing:**](https://www.geeksforgeeks.org/difference-between-top-down-parsing-and-bottom-up-parsing) Top-down parsing starts at the root of the parse tree and iteratively breaks down the sentence into smaller and smaller parts until it reaches the leaves. This is a more natural technique for parsing sentences. However, because it requires a more complicated language, it may be more difficult to implement.
* [**Bottom-up parsing:**](https://www.geeksforgeeks.org/difference-between-top-down-parsing-and-bottom-up-parsing) Bottom-up parsing starts with the leaves of the parse tree and recursively builds up the tree from smaller and smaller constituents until it reaches the root. Although this method of parsing requires simpler grammar, it is frequently simpler to implement, even when it is less understandable.

### 15. What do you mean by vector space in NLP?

In natural language processing (NLP), A [vector space](https://www.geeksforgeeks.org/web-information-retrieval-vector-space-model) is a mathematical vector where words or documents are represented by numerical vectors form. The word or document's specific features or attributes are represented by one of the dimensions of the vector. Vector space models are used to convert text into numerical representations that machine learning algorithms can understand.

Vector spaces are generated using techniques such as word embeddings, bag-of-words, and term frequency-inverse document frequency (TF-IDF). These methods allow for the conversion of textual data into dense or sparse vectors in a high-dimensional space. Each dimension of the vector may indicate a different feature, such as the presence or absence of a word, word frequency, semantic meaning, or contextual information.

### 16. What is the bag-of-words model?

[Bag of Words](https://www.geeksforgeeks.org/bag-of-words-bow-model-in-nlp) is a classical text representation technique in NLP that describes the occurrence of words within a document or not. It just keeps track of word counts and ignores the grammatical details and the word order.

Each document is transformed as a numerical vector, where each dimension corresponds to a unique word in the vocabulary. The value in each dimension of the vector represents the frequency, occurrence, or other measure of importance of that word in the document.

Let's consider two simple text documents:

Document 1: "I love apples."

Document 2: "I love mangoes too."

Step 1: Tokenization

Document 1 tokens: ["I", "love", "apples"]

Document 2 tokens: ["I", "love", "mangoes", "too"]

Step 2: Vocabulary Creation by collecting all unique words across the documents

Vocabulary: ["I", "love", "apples", "mangoes", "too"]

The vocabulary has five unique words, so each document vector will have five dimensions.

Step 3: Vectorization

Create numerical vectors for each document based on the vocabulary.

For Document 1:

- The dimension corresponding to "I" has a value of 1.

- The dimension corresponding to "love" has a value of 1.

- The dimension corresponding to "apples" has a value of 1.

- The dimensions corresponding to "mangoes" and "too" have values of 0 since they do not appear in Document 1.

Document 1 vector: [1, 1, 1, 0, 0]

For Document 2:

- The dimension corresponding to "I" has a value of 1.

- The dimension corresponding to "love" has a value of 1.

- The dimension corresponding to "mangoes" has a value of 1.

- The dimension corresponding to "apples" has a value of 0 since it does not appear in Document 2.

- The dimension corresponding to "too" has a value of 1.

Document 2 vector: [1, 1, 0, 1, 1]

The value in each dimension represents the occurrence or frequency of the corresponding word in the document. The BoW representation allows us to compare and analyze the documents based on their word frequencies.

### 17. Define the Bag of N-grams model in NLP.

The [Bag of n-grams](https://www.geeksforgeeks.org/n-gram-language-modelling-with-nltk) model is a modification of the standard bag-of-words (BoW) model in NLP. Instead of taking individual words to be the fundamental units of representation, the Bag of n-grams model considers contiguous sequences of n words, known as n-grams, to be the fundamental units of representation.

The Bag of n-grams model divides the text into n-grams, which can represent consecutive words or characters depending on the value of n. These n-grams are subsequently considered as features or tokens, similar to individual words in the BoW model.

The steps for creating a bag-of-n-grams model are as follows:

* The text is split or tokenized into individual words or characters.
* The tokenized text is used to construct N-grams of size n (sequences of n consecutive words or characters). If n is set to 1 known as uni-gram i.e. same as a bag of words, 2 i.e. bi-grams, and 3 i.e. tri-gram.
* A vocabulary is built by collecting all unique n-grams across the entire corpus.
* Similarly to the BoW approach, each document is represented as a numerical vector. The vector's dimensions correspond to the vocabulary's unique n-grams, and the value in each dimension denotes the frequency or occurrence of that n-gram in the document.

### 18. What is the term frequency-inverse document frequency (TF-IDF)?

[Term frequency-inverse document frequency (TF-IDF)](https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency)is a classical text representation technique in NLP that uses a statistical measure to evaluate the importance of a word in a document relative to a corpus of documents. It is a combination of two terms: term frequency (TF) and inverse document frequency (IDF).

* **Term Frequency (TF):** Term frequency measures how frequently a word appears in a document. it is the ratio of the number of occurrences of a term or word (t ) in a given document (d) to the total number of terms in a given document (d). A higher term frequency indicates that a word is more important within a specific document.
* **Inverse Document Frequency (IDF):** Inverse document frequency measures the rarity or uniqueness of a term across the entire corpus. It is calculated by taking the logarithm of the ratio of the total number of documents in the corpus to the number of documents containing the term. it down the weight of the terms, which frequently occur in the corpus, and up the weight of rare terms.

The TF-IDF score is calculated by multiplying the term frequency (TF) and inverse document frequency (IDF) values for each term in a document. The resulting score indicates the term's importance in the document and corpus. Terms that appear frequently in a document but are uncommon in the corpus will have high TF-IDF scores, suggesting their importance in that specific document.

### 19. Explain the concept of cosine similarity and its importance in NLP.

The similarity between two vectors in a multi-dimensional space is measured using the cosine similarity metric. To determine how similar or unlike the vectors are to one another, it calculates the cosine of the angle between them.

In natural language processing (NLP), [Cosine similarity](https://www.geeksforgeeks.org/cosine-similarity) is used to compare two vectors that represent text. The degree of similarity is calculated using the cosine of the angle between the document vectors. To compute the cosine similarity between two text document vectors, we often used the following procedures:

* Text Representation: Convert text documents into numerical vectors using approaches like bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings like Word2Vec or GloVe.
* Vector Normalization: Normalize the document vectors to unit length. This normalization step ensures that the length or magnitude of the vectors does not affect the cosine similarity calculation.
* Cosine Similarity Calculation: Take the dot product of the normalised vectors and divide it by the product of the magnitudes of the vectors to obtain the cosine similarity.

Mathematically, the cosine similarity between two document vectors, a⃗   *a*   and b⃗   *b*   , can be expressed as:

Cosine Similarity(a⃗,b⃗)=a⃗⋅b⃗∣a⃗∣∣b⃗∣Cosine Similarity(*a*,*b*)=∣*a*∣∣*b*∣*a*⋅*b*​

Here,

* a⃗⋅b⃗   *a*⋅*b*   is the dot product of vectors a and b
* |a| and |b| represent the Euclidean norms (magnitudes) of vectors a and b, respectively.

The resulting cosine similarity score ranges from -1 to 1, where 1 represents the highest similarity, 0 represents no similarity, and -1 represents the maximum dissimilarity between the documents.

### 20. What are the differences between rule-based, statistical-based and neural-based approaches in NLP?

[Natural language processing (NLP)](https://www.geeksforgeeks.org/natural-language-processing-nlp-pipeline) uses three distinct approaches to tackle language understanding and processing tasks: rule-based, statistical-based, and neural-based.

1. **Rule-based Approach:**Rule-based systems rely on predefined sets of linguistic rules and patterns to analyze and process language.
   * Linguistic Rules are manually crafted rules by human experts to define patterns or grammar structures.
   * The knowledge in rule-based systems is explicitly encoded in the rules, which may cover syntactic, semantic, or domain-specific information.
   * Rule-based systems offer high interpretability as the rules are explicitly defined and understandable by human experts.
   * These systems often require manual intervention and rule modifications to handle new language variations or domains.
2. **Statistical-based Approach:**Statistical-based systems utilize statistical algorithms and models to learn patterns and structures from large datasets.
   * By examining the data's statistical patterns and relationships, these systems learn from training data.
   * Statistical models are more versatile than rule-based systems because they can train on relevant data from various topics and languages.
3. **Neural-based Approach:** Neural-based systems employ deep learning models, such as neural networks, to learn representations and patterns directly from raw text data.
   * Neural networks learn hierarchical representations of the input text, which enable them to capture complex language features and semantics.
   * Without explicit rule-making or feature engineering, these systems learn directly from data.
   * By training on huge and diverse datasets, neural networks are very versatile and can perform a wide range of NLP tasks.
   * In many NLP tasks, neural-based models have attained state-of-the-art performance, outperforming classic rule-based or statistical-based techniques.

### 21. What do you mean by Sequence in the Context of NLP?

A Sequence primarily refers to the sequence of elements that are analyzed or processed together. In [NLP](https://www.geeksforgeeks.org/natural-language-processing-nlp-pipeline), a sequence may be a sequence of characters, a sequence of words or a sequence of sentences.

In general, sentences are often treated as sequences of words or tokens. Each word in the sentence is considered an element in the sequence. This sequential representation allows for the analysis and processing of sentences in a structured manner, where the order of words matters.

By considering sentences as sequences, NLP models can capture the contextual information and dependencies between words, enabling tasks such as part-of-speech tagging, named entity recognition, sentiment analysis, machine translation, and more.

### 22. What are the various types of machine learning algorithms used in NLP?

There are various types of machine learning algorithms that are often employed in natural language processing (NLP) tasks. Some of them are as follows:

* [Naive Bayes:](https://www.geeksforgeeks.org/naive-bayes-classifiers) Naive Bayes is a probabilistic technique that is extensively used in NLP for text classification tasks. It computes the likelihood of a document belonging to a specific class based on the presence of words or features in the document.
* [Support Vector Machines (SVM)](https://www.geeksforgeeks.org/support-vector-machine-algorithm): SVM is a supervised learning method that can be used for text classification, sentiment analysis, and named entity recognition. Based on the given set of features, SVM finds a hyperplane that splits data points into various classes.
* [Decision Trees:](https://www.geeksforgeeks.org/decision-tree) Decision trees are commonly used for tasks such as sentiment analysis, and information extraction. These algorithms build a tree-like model based on an order of decisions and feature conditions, which helps in making predictions or classifications.
* [Random Forests:](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn) Random forests are a type of ensemble learning that combines multiple decision trees to improve accuracy and reduce overfitting.  They can be applied to the tasks like text classification, named entity recognition, and sentiment analysis.
* [Recurrent Neural Networks (RNN):](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network) RNNs are a type of neural network architecture that are often used in sequence-based NLP tasks like language modelling, machine translation, and sentiment analysis. RNNs can capture temporal dependencies and context within a word sequence.
* [Long Short-Term Memory (LSTM)](https://www.geeksforgeeks.org/long-short-term-memory-lstm-rnn-in-tensorflow): LSTMs are a type of recurrent neural network that was developed to deal with the vanishing gradient problem of RNN. LSTMs are useful for capturing long-term dependencies in sequences, and they have been used in applications such as machine translation, named entity identification, and sentiment analysis.
* [Transformer](https://www.geeksforgeeks.org/transformer-neural-network-in-deep-learning-overview): Transformers are a relatively recent architecture that has gained significant attention in NLP. By exploiting self-attention processes to capture contextual relationships in text, transformers such as the BERT (Bidirectional Encoder Representations from Transformers) model have achieved state-of-the-art performance in a wide range of NLP tasks.

### 23. What is Sequence Labelling in NLP?

Sequence labelling is one of the fundamental NLP tasks in which, categorical labels are assigned to each individual element in a sequence. The sequence can represent various linguistic units such as words, characters, sentences, or paragraphs.

Sequence labelling in NLP includes the following tasks.

* Part-of-Speech Tagging (POS Tagging): In which part-of-speech tags (e.g., noun, verb, adjective) are assigned to each word in a sentence.
* Named Entity Recognition (NER): In which named entities like person names, locations, organizations, or dates are recognized and tagged in the sentences.
* Chunking: Words are organized into syntactic units or "chunks" based on their grammatical roles (for example, noun phrase, verb phrase).
* Semantic Role Labeling (SRL): In which, words or phrases in a sentence are labelled based on their semantic roles like Teacher, Doctor, Engineer, Lawyer etc
* Speech Tagging: In speech processing tasks such as speech recognition or phoneme classification, labels are assigned to phonetic units or acoustic segments.

Machine learning models like Conditional Random Fields (CRFs), Hidden Markov Models (HMMs), recurrent neural networks (RNNs), or transformers are used for sequence labelling tasks. These models learn from the labelled training data to make predictions on unseen data.

### 24.What is topic modelling in NLP?

Topic modelling is Natural Language Processing task used to discover hidden topics from large text documents. It is an unsupervised technique, which takes unlabeled text data as inputs and applies the probabilistic models that represent the probability of each document being a mixture of topics. For example, A document could have a 60% chance of being about neural networks, a 20% chance of being about Natural Language processing, and a 20% chance of being about anything else.

Where each topic will be distributed over words means each topic is a list of words, and each word has a probability associated with it. and the words that have the highest probabilities in a topic are the words that are most likely to be used to describe that topic. For example, the words like "neural", "RNN", and "architecture" are the keywords for neural networks and the words like 'language", and "sentiment" are the keywords for Natural Language processing.

There are a number of topic modelling algorithms but two of the most popular topic modelling algorithms are as follows:

* [**Latent Dirichlet Allocation (LDA)**](https://www.geeksforgeeks.org/latent-dirichlet-allocation)**:**LDA is based on the idea that each text in the corpus is a mash-up of various topics and that each word in the document is derived from one of those topics. It is assumed that there is an unobservable (latent) set of topics and each document is generated by Topic Selection or Word Generation.
* [**Non-Negative Matrix Factorization (NMF)**](https://www.geeksforgeeks.org/non-negative-matrix-factorization)**:** NMF is a matrix factorization technique that approximates the term-document matrix (where rows represent documents and columns represent words) into two non-negative matrices: one representing the topic-word relationships and the other the document-topic relationships. NMF aims to identify representative topics and weights for each document.

Topic modelling is especially effective for huge text collections when manually inspecting and categorising each document would be impracticable and time-consuming. We can acquire insights into the primary topics and structures of text data by using topic modelling, making it easier to organise, search, and analyse enormous amounts of unstructured text.

### 25. What is the GPT?

[GPT](https://www.geeksforgeeks.org/gpt-4-vs-gpt-3) stands for "Generative Pre-trained Transformer". It refers to a collection of large language models created by OpenAI. It is trained on a massive dataset of text and code, which allows it to generate text, generate code, translate languages, and write many types of creative content, as well as answer questions in an informative manner. The GPT series includes various models, the most well-known and commonly utilised of which are the GPT-2 and GPT-3.

GPT models are built on the Transformer architecture, which allows them to efficiently capture long-term dependencies and contextual information in text. These models are pre-trained on a large corpus of text data from the internet, which enables them to learn the underlying patterns and structures of language.

## *Advanced NLP Interview Questions for Experienced*

### 26. What are word embeddings in NLP?

[Word embeddings](https://www.geeksforgeeks.org/word-embeddings-in-nlp) in NLP are defined as the dense, low-dimensional vector representations of words that capture semantic and contextual information about words in a language. It is trained using big text corpora through unsupervised or supervised methods to represent words in a numerical format that can be processed by machine learning models.

The main goal of Word embeddings is to capture relationships and similarities between words by representing them as dense vectors in a continuous vector space. These vector representations are acquired using the distributional hypothesis, which states that words with similar meanings tend to occur in similar contexts. Some of the popular pre-trained word embeddings are Word2Vec, GloVe (Global Vectors for Word Representation), or FastText. The advantages of word embedding over the traditional text vectorization technique are as follows:

* It can capture the Semantic Similarity between the words
* It is capable of capturing syntactic links between words. Vector operations such as "king" - "man" + "woman" may produce a vector similar to the vector for "queen," capturing the gender analogy.
* Compared to one-shot encoding, it has reduced the dimensionality of word representations. Instead of high-dimensional sparse vectors, word embeddings typically have a fixed length and represent words as dense vectors.
* It can be generalized to represent words that they have not been trained on i.e. out-of-vocabulary words. This is done by using the learned word associations to place new words in the vector space near words that they are semantically or syntactically similar to.

### 27. What are the various algorithms used for training word embeddings?

There are various approaches that are typically used for training word embeddings, which are dense vector representations of words in a continuous vector space. Some of the popular word embedding algorithms are as follows:

* [**Word2Vec**](https://www.geeksforgeeks.org/python-word-embedding-using-word2vec): Word2vec is a common approach for generating vector representations of words that reflect their meaning and relationships. Word2vec learns embeddings using a shallow neural network and follows two approaches: CBOW and Skip-gram
  + CBOW (Continuous Bag-of-Words) predicts a target word based on its context words.
  + Skip-gram predicts context words given a target word.
* **GloVe**: GloVe (Global Vectors for Word Representation) is a word embedding model that is similar to Word2vec. GloVe, on the other hand, uses  objective function that constructs a co-occurrence matrix based on the statistics of word co-occurrences in a large corpus. The co-occurrence matrix is a square matrix where each entry represents the number of times two words co-occur in a window of a certain size. GloVe then performs matrix factorization on the co-occurrence matrix. Matrix factorization is a technique for finding a low-dimensional representation of a high-dimensional matrix. In the case of GloVe, the low-dimensional representation is a vector representation for each word in the corpus. The word embeddings are learned by minimizing a loss function that measures the difference between the predicted co-occurrence probabilities and the actual co-occurrence probabilities. This makes GloVe more robust to noise and less sensitive to the order of words in a sentence.
* [**FastText**](https://www.geeksforgeeks.org/fasttext-working-and-implementation): FastText is a Word2vec extension that includes subword information. It represents words as bags of character n-grams, allowing it to handle out-of-vocabulary terms and capture morphological information. During training, FastText considers subword information as well as word context..
* [**ELMo**](https://www.geeksforgeeks.org/overview-of-word-embedding-using-embeddings-from-language-models-elmo): ELMo is a deeply contextualised word embedding model that generates context-dependent word representations. It generates word embeddings that capture both semantic and syntactic information based on the context of the word using bidirectional language models.
* [**BERT**](https://www.geeksforgeeks.org/explanation-of-bert-model-nlp): A transformer-based model called BERT (Bidirectional Encoder Representations from Transformers) learns contextualised word embeddings. BERT is trained on a large corpus by anticipating masked terms inside a sentence and gaining knowledge about the bidirectional context. The generated embeddings achieve state-of-the-art performance in many NLP tasks and capture extensive contextual information.

### 28. How to handle out-of-vocabulary (OOV) words in NLP?

OOV words are words that are missing in a language model's vocabulary or the training data it was trained on. Here are a few approaches to handling OOV words in NLP:

1. **Character-level models:**Character-level models can be used in place of word-level representations. In this method, words are broken down into individual characters, and the model learns representations based on character sequences. As a result, the model can handle OOV words since it can generalize from known character patterns.
2. **Subword tokenization:** Byte-Pair Encoding (BPE) and WordPiece are two subword tokenization algorithms that divide words into smaller subword units based on their frequency in the training data. This method enables the model to handle OOV words by representing them as a combination of subwords that it comes across during training.
3. **Unknown token:** Use a special token, frequently referred to as an "unknown" token or "UNK," to represent any OOV term that appears during inference. Every time the model comes across an OOV term, it replaces it with the unidentified token and keeps processing. The model is still able to generate relevant output even though this technique doesn't explicitly define the meaning of the OOV word.
4. **External knowledge:**When dealing with OOV terms, using external knowledge resources, like a knowledge graph or an external dictionary, can be helpful. We need to try to look up a word's definition or relevant information in the external knowledge source when we come across an OOV word.
5. **Fine-tuning:**We can fine-tune using the pre-trained language model with domain-specific or task-specific data that includes OOV words. By incorporating OOV words in the fine-tuning process, we expose the model to these words and increase its capacity to handle them.

### 29. What is the difference between a word-level and character-level language model?

The main difference between a word-level and a character-level language model is how text is represented. A character-level language model represents text as a sequence of characters, whereas a word-level language model represents text as a sequence of words.

Word-level language models are often easier to interpret and more efficient to train. They are, however, less accurate than character-level language models because they cannot capture the intricacies of the text that are stored in the character order. Character-level language models are more accurate than word-level language models, but they are more complex to train and interpret. They are also more sensitive to noise in the text, as a slight alteration in a character can have a large impact on the meaning of the text.

The key differences between word-level and character-level language models are:

|  | **Word-level** | **Character-level** |
| --- | --- | --- |
| **Text representation** | Sequence of words | Sequence of characters |
| **Interpretability** | Easier to interpret | More difficult to interpret |
| **Sensitivity to noise** | Less sensitive | More sensitive |
| **Vocabulary** | Fixed vocabulary of words | No predefined vocabulary |
| **Out-of-vocabulary (OOV) handling** | Struggles with OOV words | Naturally handles OOV words |
| **Generalization** | Captures semantic relationships between words | Better at handling morphological details |
| **Training complexity** | Smaller input/output space, less computationally intensive | Larger input/output space, more computationally intensive |
| **Applications** | Well-suited for tasks requiring word-level understanding | Suitable for tasks requiring fine-grained details or morphological variations |

### 30. What is word sense disambiguation?

The task of determining which sense of a word is intended in a given context is known as [word sense disambiguation (WSD)](https://www.geeksforgeeks.org/word-sense-disambiguation-in-natural-language-processing). This is a challenging task because many words have several meanings that can only be determined by considering the context in which the word is used.

For example, the word "bank" can be used to refer to a variety of things, including "a financial institution," "a riverbank," and "a slope." The term "bank" in the sentence "I went to the bank to deposit my money" should be understood to mean "a financial institution." This is so because the sentence's context implies that the speaker is on their way to a location where they can deposit money.

### 31. What is co-reference resolution?

Co-reference resolution is a natural language processing (NLP) task that involves identifying all expressions in a text that refer to the same entity. In other words, it tries to determine whether words or phrases in a text, typically pronouns or noun phrases, correspond to the same real-world thing. For example, the pronoun "he" in the sentence "Pawan Gunjan has compiled this article, He had done lots of research on Various NLP interview questions" refers to Pawan Gunjan himself. Co-reference resolution automatically identifies such linkages and establishes that "He" refers to "Pawan Gunjan" in all instances.

Co-reference resolution is used in information extraction, question answering, summarization, and dialogue systems because it helps to generate more accurate and context-aware representations of text data. It is an important part of systems that require a more in-depth understanding of the relationships between entities in large text corpora.

### 32.What is information extraction?

[Information extraction](https://www.geeksforgeeks.org/difference-between-information-retrieval-and-information-extraction) is a natural language processing task used to extract specific pieces of information like names, dates, locations, and relationships etc from unstructured or semi-structured texts.

Natural language is often ambiguous and can be interpreted in a variety of ways, which makes IE a difficult process. Some of the common techniques used for information extraction include:

* **Named entity recognition (NER):** In NER, named entities like people, organizations, locations, dates, or other specific categories are recognized from the text documents. For NER problems, a variety of machine learning techniques, including conditional random fields (CRF), support vector machines (SVM), and deep learning models, are frequently used.
* **Relationship extraction:**In relationship extraction, the connections between the stated text are identified. I figure out the relations different kinds of relationships between various things like "is working at", "lives in" etc.
* **Coreference resolution:** Coreference resolution is the task of identifying the referents of pronouns and other anaphoric expressions in the text. A coreference resolution system, for example, might be able to figure out that the pronoun "he" in a sentence relates to the person "John" who was named earlier in the text.
* **Deep Learning-based Approaches:**To perform information extraction tasks, deep learning models such as recurrent neural networks (RNNs), transformer-based architectures (e.g., BERT, GPT), and deep neural networks have been used. These models can learn patterns and representations from data automatically, allowing them to manage complicated and diverse textual material.

### 33. What is the Hidden Markov Model, and How it's helpful in NLP tasks?

[Hidden Markov Model](https://www.geeksforgeeks.org/hidden-markov-model-in-machine-learning) is a probabilistic model based on the Markov Chain Rule used for modelling sequential data like characters, words, and sentences by computing the probability distribution of sequences.

Markov chain uses the Markov assumptions which state that the probabilities future state of the system only depends on its present state, not on any past state of the system. This assumption simplifies the modelling process by reducing the amount of information needed to predict future states.

The underlying process in an HMM is represented by a set of hidden states that are not directly observable. Based on the hidden states, the observed data, such as characters, words, or phrases, are generated.

Hidden Markov Models consist of two key components:

1. Transition Probabilities: The transition probabilities in Hidden Markov Models(HMMs) represents the likelihood of moving from one hidden state to another. It captures the dependencies or relationships between adjacent states in the sequence. In part-of-speech tagging, for example, the HMM's hidden states represent distinct part-of-speech tags, and the transition probabilities indicate the likelihood of transitioning from one part-of-speech tag to another.
2. Emission Probabilities: In HMMs, emission probabilities define the likelihood of observing specific symbols (characters, words, etc.) given a particular hidden state. The link between the hidden states and the observable symbols is encoded by these probabilities.
3. Emission probabilities are often used in NLP to represent the relationship between words and linguistic features such as part-of-speech tags or other linguistic variables. The HMM captures the likelihood of generating an observable symbol (e.g., word) from a specific hidden state (e.g., part-of-speech tag) by calculating the emission probabilities.

Hidden Markov Models (HMMs) estimate transition and emission probabilities from labelled data using approaches such as the Baum-Welch algorithm. Inference algorithms like Viterbi and Forward-Backward are used to determine the most likely sequence of hidden states given observed symbols. HMMs are used to represent sequential data and have been implemented in NLP applications such as part-of-speech tagging. However, advanced models, such as CRFs and neural networks, frequently beat HMMs due to their flexibility and ability to capture richer dependencies.

### 34. What is the conditional random field (CRF) model in NLP?

[Conditional Random Fields](https://www.geeksforgeeks.org/conditional-random-fields-crfs-for-pos-tagging-in-nlp) are a probabilistic graphical model that is designed to predict the sequence of labels for a given sequence of observations. It is well-suited for prediction tasks in which contextual information or dependencies among neighbouring elements are crucial.

CRFs are an extension of Hidden Markov Models (HMMs) that allow for the modelling of more complex relationships between labels in a sequence. It is specifically designed to capture dependencies between non-consecutive labels, whereas HMMs presume a Markov property in which the current state is only dependent on the past state. This makes CRFs more adaptable and suitable for capturing long-term dependencies and complicated label interactions.

In a CRF model, the labels and observations are represented as a graph. The nodes in the graph represent the labels, and the edges represent the dependencies between the labels. The model assigns weights to features that capture relevant information about the observations and labels.

During training, the CRF model learns the weights by maximizing the conditional log-likelihood of the labelled training data. This process involves optimization algorithms such as gradient descent or the iterative scaling algorithm.

During inference, given an input sequence, the CRF model calculates the conditional probabilities of different label sequences. Algorithms like the Viterbi algorithm efficiently find the most likely label sequence based on these probabilities.

CRFs have demonstrated high performance in a variety of sequence labelling tasks like named entity identification, part-of-speech tagging, and others.

### 35. What is a recurrent neural network (RNN)?

[Recurrent Neural Networks](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network) are the type of artificial neural network that is specifically built to work with sequential or time series data. It is utilised in natural language processing activities such as language translation, speech recognition, sentiment analysis, natural language production, summary writing, and so on. It differs from feedforward neural networks in that the input data in RNN does not only flow in a single direction but also has a loop or cycle inside its design that has "memory" that preserves information over time. As a result, the RNN can handle data where context is critical, such as natural languages.

RNNs work by analysing input sequences one element at a time while keeping track in a hidden state that provides a summary of the sequence's previous elements. At each time step, the hidden state is updated based on the current input and the prior hidden state. RNNs can thus capture the temporal connections between sequence items and use that knowledge to produce predictions.

### 36. How does the Backpropagation through time work in RNN?

[Backpropagation through time(BPTT)](https://www.geeksforgeeks.org/ml-back-propagation-through-time) propagates gradient information across the RNN's recurrent connections over a sequence of input data. Let's understand step by step process for BPTT.

1. Forward Pass: The input sequence is fed into the RNN one element at a time, starting from the first element. Each input element is processed through the recurrent connections, and the hidden state of the RNN is updated.
2. Hidden State Sequence: The hidden state of the RNN is maintained and carried over from one time step to the next. It contains information about the previous inputs and hidden states in the sequence.
3. Output Calculation: The updated hidden state is used to compute the output at each time step.
4. Loss Calculation: At the end of the sequence, the predicted output is compared to the target output, and a loss value is calculated using a suitable loss function, such as mean squared error or cross-entropy loss.
5. Backpropagation: The loss is then backpropagated through time, starting from the last time step and moving backwards in time. The gradients of the loss with respect to the parameters of the RNN are calculated at each time step.
6. Weight Update: The gradients are accumulated over the entire sequence, and the weights of the RNN are updated using an optimization algorithm such as gradient descent or its variants.
7. Repeat: The process is repeated for a specified number of epochs or until convergence, during this the training data is iterated through several times.

During the backpropagation step, the gradients at each time step are obtained and used to update the weights of the recurrent connections. This accumulation of gradients over numerous time steps allows the RNN to learn and capture dependencies and patterns in sequential data.

### 37. What are the limitations of a standard RNN?

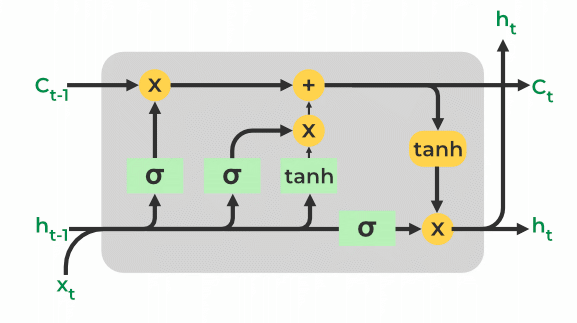
Standard [RNNs (Recurrent Neural Networks)](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network) have several limitations that can make them unsuitable for certain applications:

1. Vanishing Gradient Problem: Standard RNNs are vulnerable to the vanishing gradient problem, in which gradients decrease exponentially as they propagate backwards through time. Because of this issue, it is difficult for the network to capture and transmit long-term dependencies across multiple time steps during training.
2. Exploding Gradient Problem: RNNs, on the other hand, can suffer from the expanding gradient problem, in which gradients get exceedingly big and cause unstable training. This issue can cause the network to converge slowly or fail to converge at all.
3. Short-Term Memory: Standard RNNs have limited memory and fail to remember information from previous time steps. Because of this limitation, they have difficulty capturing long-term dependencies in sequences, limiting their ability to model complicated relationships that span a significant number of time steps.

### 38. What is a long short-term memory (LSTM) network?

A [Long Short-Term Memory (LSTM)](https://www.geeksforgeeks.org/long-short-term-memory-lstm-rnn-in-tensorflow) network is a type of recurrent neural network (RNN) architecture that is designed to solve the vanishing gradient problem and capture long-term dependencies in sequential data. LSTM networks are particularly effective in tasks that involve processing and understanding sequential data, such as natural language processing and speech recognition.

The key idea behind LSTMs is the integration of a memory cell, which acts as a memory unit capable of retaining information for an extended period. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate.



The input gate controls how much new information should be stored in the memory cell. The forget gate determines which information from the memory cell should be destroyed or forgotten. The output gate controls how much information is output from the memory cell to the next time step. These gates are controlled by activation functions, which are commonly sigmoid and tanh functions, and allow the LSTM to selectively update, forget, and output data from the memory cell.

### 39. What is the GRU model in NLP?

The [Gated Recurrent Unit (GRU)](https://www.geeksforgeeks.org/gated-recurrent-unit-networks) model is a type of recurrent neural network (RNN) architecture that has been widely used in natural language processing (NLP) tasks. It is designed to address the vanishing gradient problem and capture long-term dependencies in sequential data.

GRU is similar to LSTM in that it incorporates gating mechanisms, but it has a simplified architecture with fewer gates, making it computationally more efficient and easier to train. The GRU model consists of the following components:

1. **Hidden State:** The hidden state ht−1  *ht*−1​  in GRU represents the learned representation or memory of the input sequence up to the current time step. It retains and passes information from the past to the present.
2. **Update Gate:** The update gate in GRU controls the flow of information from the past hidden state to the current time step. It determines how much of the previous information should be retained and how much new information should be incorporated.
3. **Reset Gate:** The reset gate in GRU determines how much of the past information should be discarded or forgotten. It helps in removing irrelevant information from the previous hidden state.
4. **Candidate Activation:**The candidate activation represents the new information to be added to the hidden state ht′  *ht*′​  . It is computed based on the current input and a transformed version of the previous hidden state using the reset gate.

GRU models have been effective in NLP applications like language modelling, sentiment analysis, machine translation, and text generation. They are particularly useful in situations when it is essential to capture long-term dependencies and understand the context. Due to its simplicity and computational efficiency, GRU makes it a popular choice in NLP research and applications.

### 40. What is the sequence-to-sequence (Seq2Seq) model in NLP?

[Sequence-to-sequence (Seq2Seq)](https://www.geeksforgeeks.org/seq2seq-model-in-machine-learning) is a type of neural network that is used for natural language processing (NLP) tasks. It is a type of recurrent neural network (RNN) that can learn long-term word relationships. This makes it ideal for tasks like machine translation, text summarization, and question answering.

The model is composed of two major parts: an encoder and a decoder. Here's how the Seq2Seq model works:

1. **Encoder**: The encoder transforms the input sequence, such as a sentence in the source language, into a fixed-length vector representation known as the "context vector" or "thought vector". To capture sequential information from the input, the encoder commonly employs recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU).
2. **Context Vector:**The encoder's context vector acts as a summary or representation of the input sequence. It encodes the meaning and important information from the input sequence into a fixed-size vector, regardless of the length of the input.
3. **Decoder**: The decoder uses the encoder's context vector to build the output sequence, which could be a translation or a summarised version. It is another RNN-based network that creates the output sequence one token at a time. At each step, the decoder can be conditioned on the context vector, which serves as an initial hidden state.

During training, the decoder is fed ground truth tokens from the target sequence at each step. Backpropagation through time (BPTT) is a technique commonly used to train Seq2Seq models. The model is optimized to minimize the difference between the predicted output sequence and the actual target sequence.

The Seq2Seq model is used during prediction or generation to construct the output sequence word by word, with each predicted word given back into the model as input for the subsequent step. The process is repeated until either an end-of-sequence token or a predetermined maximum length is achieved.

### 41. How does the attention mechanism helpful in NLP?

An [attention mechanism](https://www.geeksforgeeks.org/ml-attention-mechanism) is a kind of neural network that uses an additional attention layer within an Encoder-Decoder neural network that enables the model to focus on specific parts of the input while performing a task. It achieves this by dynamically assigning weights to different elements in the input, indicating their relative importance or relevance. This selective attention allows the model to focus on relevant information, capture dependencies, and analyze relationships within the data.

The attention mechanism is particularly valuable in tasks involving sequential or structured data, such as natural language processing or computer vision, where long-term dependencies and contextual information are crucial for achieving high performance. By allowing the model to selectively attend to important features or contexts, it improves the model's ability to handle complex relationships and dependencies in the data, leading to better overall performance in various tasks.

### 42. What is the Transformer model?

[Transformer](https://www.geeksforgeeks.org/transformer-neural-network-in-deep-learning-overview) is one of the fundamental models in NLP based on the attention mechanism, which allows it to capture long-range dependencies in sequences more effectively than traditional recurrent neural networks (RNNs). It has given state-of-the-art results in various NLP tasks like word embedding, machine translation, text summarization, question answering etc.

Some of the key advantages of using a Transformer are as follows:

* **Parallelization**: The self-attention mechanism allows the model to process words in parallel, which makes it significantly faster to train compared to sequential models like RNNs.
* **Long-Range Dependencies:** The attention mechanism enables the Transformer to effectively capture long-range dependencies in sequences, which makes it suitable for tasks where long-term context is essential.
* **State-of-the-Art Performance:**Transformer-based models have achieved state-of-the-art performance in various NLP tasks, such as machine translation, language modelling, text generation, and sentiment analysis.

The key components of the Transformer model are as follows:

* Self-Attention Mechanism:
* Encoder-Decoder Network:
* Multi-head Attention:
* Positional Encoding
* Feed-Forward Neural Networks
* Layer Normalization and Residual Connections

### 43. What is the role of the self-attention mechanism in Transformers?

The [self-attention mechanism](https://www.geeksforgeeks.org/self-attention-in-nlp) is a powerful tool that allows the Transformer model to capture long-range dependencies in sequences. It allows each word in the input sequence to attend to all other words in the same sequence, and the model learns to assign weights to each word based on its relevance to the others. This enables the model to capture both short-term and long-term dependencies, which is critical for many NLP applications.

### 44. What is the purpose of the multi-head attention mechanism in Transformers?

The purpose of the [multi-head attention mechanism](https://www.geeksforgeeks.org/ml-attention-mechanism) in Transformers is to allow the model to recognize different types of correlations and patterns in the input sequence. In both the encoder and decoder, the Transformer model uses multiple attention heads. This enables the model to recognise different types of correlations and patterns in the input sequence. Each attention head learns to pay attention to different parts of the input, allowing the model to capture a wide range of characteristics and dependencies.

The multi-head attention mechanism helps the model in learning richer and more contextually relevant representations, resulting in improved performance on a variety of natural language processing (NLP) tasks.

### 45. What are positional encodings in Transformers, and why are they necessary?

The [transformer](https://www.geeksforgeeks.org/transformer-neural-network-in-deep-learning-overview) model processes the input sequence in parallel, so that lacks the inherent understanding of word order like the sequential model recurrent neural networks (RNNs), LSTM possess. So, that. it requires a method to express the positional information explicitly.

Positional encoding is applied to the input embeddings to offer this positional information like the relative or absolute position of each word in the sequence to the model. These encodings are typically learnt and can take several forms, including sine and cosine functions or learned embeddings. This enables the model to learn the order of the words in the sequence, which is critical for many NLP tasks.

### 46. Describe the architecture of the Transformer model.

The architecture of the Transformer model is based on self-attention and feed-forward neural network concepts. It is made up of an encoder and a decoder, both of which are composed of multiple layers, each containing self-attention and feed-forward sub-layers. The model's design encourages parallelization, resulting in more efficient training and improved performance on tasks involving sequential data, such as natural language processing (NLP) tasks.

The architecture can be described in depth below:

1. [**Encoder**](https://www.geeksforgeeks.org/difference-between-encoder-and-decoder):
   * Input Embeddings: The encoder takes an input sequence of tokens (e.g., words) as input and transforms each token into a vector representation known as an embedding. Positional encoding is used in these embeddings to preserve the order of the words in the sequence.
   * Self-Attention Layers: An encoder consists of multiple self-attention layers and each self-attention layer is used to capture relationships and dependencies between words in the sequence.
   * Feed-Forward Layers: After the self-attention step, the output representations of the self-attention layer are fed into a feed-forward neural network. This network applies the non-linear transformations to each word's contextualised representation independently.
   * Layer Normalization and Residual Connections: Residual connections and layer normalisation are used to back up the self-attention and feed-forward layers. The residual connections in deep networks help to mitigate the vanishing gradient problem, and layer normalisation stabilises the training process.
2. [**Decoder**](https://www.geeksforgeeks.org/difference-between-encoder-and-decoder):
   * Input Embeddings: Similar to the encoder, the decoder takes an input sequence and transforms each token into embeddings with positional encoding.
   * Masked Self-Attention: Unlike the encoder, the decoder uses masked self-attention in the self-attention layers. This masking ensures that the decoder can only attend to places before the current word during training, preventing the model from seeing future tokens during generation.
   * Cross-Attention Layers: Cross-attention layers in the decoder allow it to attend to the encoder's output, which enables the model to use information from the input sequence during output sequence generation.
   * Feed-Forward Layers: Similar to the encoder, the decoder's self-attention output passes through feed-forward neural networks.
   * Layer Normalization and Residual Connections: The decoder also includes residual connections and layer normalization to help in training and improve model stability.
3. **Final Output Layer:**
   * Softmax Layer: The final output layer is a softmax layer that transforms the decoder's representations into probability distributions over the vocabulary. This enables the model to predict the most likely token for each position in the output sequence.

Overall, the Transformer's architecture enables it to successfully handle long-range dependencies in sequences and execute parallel computations, making it highly efficient and powerful for a variety of sequence-to-sequence tasks. The model has been successfully used for machine translation, language modelling, text generation, question answering, and a variety of other NLP tasks, with state-of-the-art results.

### 47. What is the difference between a generative and discriminative model in NLP?

Both generative and discriminative models are the types of [machine learning](https://www.geeksforgeeks.org/machine-learning) models used for different purposes in the field of natural language processing (NLP).

[Generative models](https://www.geeksforgeeks.org/the-difference-between-generative-and-discriminative-machine-learning-algorithms) are trained to generate new data that is similar to the data that was used to train them.  For example, a generative model could be trained on a dataset of text and code and then used to generate new text or code that is similar to the text and code in the dataset. Generative models are often used for tasks such as text generation, machine translation, and creative writing.

[Discriminative models](https://www.geeksforgeeks.org/the-difference-between-generative-and-discriminative-machine-learning-algorithms) are trained to recognise different types of data. A discriminative model. For example, a discriminative model could be trained on a dataset of labelled text and then used to classify new text as either spam or ham. Discriminative models are often used for tasks such as text classification, sentiment analysis, and question answering.

The key differences between generative and discriminative models in NLP are as follows:

|  | **Generative Models** | **Discriminative Models** |
| --- | --- | --- |
| **Purpose** | Generate new data that is similar to the training data. | Distinguish between different classes or categories of data. |
| **Training** | Learn the joint probability distribution of input and output data to generate new samples. | Learn the conditional probability distribution of the output labels given the input data. |
| **Examples** | Text generation, machine translation, creative writing, Chatbots, text summarization, and language modelling. | Text classification, sentiment analysis, and named entity recognition. |

### 48. What is machine translation, and how does it is performed?

[Machine translation](https://www.geeksforgeeks.org/machine-translation-of-languages-in-artificial-intelligence) is the process of automatically translating text or speech from one language to another using a computer or machine learning model.

There are three techniques for machine translation:

* Rule-based machine translation (RBMT): RBMT systems use a set of rules to translate text from one language to another.
* Statistical machine translation (SMT): SMT systems use statistical models to calculate the probability of a given translation being correct.
* Neural machine translation (NMT): Neural machine translation (NMT) is a recent technique of machine translation have been proven to be more accurate than RBMT and SMT systems, In recent years, neural machine translation (NMT), powered by deep learning models such as the Transformer, are becoming increasingly popular.

### 49. What is the BLEU score?

[BLEU](https://www.geeksforgeeks.org/nlp-bleu-score-for-evaluating-neural-machine-translation-python) stands for "Bilingual Evaluation Understudy". It is a metric invented by IBM in 2001 for evaluating the quality of a machine translation. It measures the similarity between machine-generated translations with the professional human translation. It was one of the first metrics whose results are very much correlated with human judgement.

The BLEU score is measured by comparing the n-grams (sequences of n words) in the machine-translated text to the n-grams in the reference text. The higher BLEU Score signifies, that the machine-translated text is more similar to the reference text.

The BLEU (Bilingual Evaluation Understudy) score is calculated using n-gram precision and a brevity penalty.

* N-gram Precision: The n-gram precision is the ratio of matching n-grams in the machine-generated translation to the total number of n-grams in the reference translation. The number of unigrams, bigrams, trigrams, and four-grams (i=1,...,4) that coincide with their n-gram counterpart in the reference translations is measured by the n-gram overlap.  
  precisioni=Count of matching n-gramscount of all n-grams in the machine translation  precision*i*​=count of all n-grams in the machine translationCount of matching n-grams​    
  For BLEU score precisioni  precision*i*​  is calculated for the I ranging (1 to N). Usually, the N value will be up to 4.
* Brevity Penalty: Brevity Penalty measures the length difference between machine-generated translations and reference translations. While finding the BLEU score, It penalizes the machine-generated translations if that is found too short compared to the reference translation's length with exponential decay.  
  brevity-penalty=min⁡(1,exp⁡(1−Reference lengthMachine translation length)))brevity-penalty=min(1,exp(1−Machine translation length)Reference length​))
* BLEU Score: The BLEU score is calculated by taking the geometric mean of the individual n-gram precisions and then adjusting it with the brevity penalty.  
  BLEU=brevity-penalty×exp⁡[∑i=1Nlog⁡(precisioni)N]=brevity-penalty×exp⁡[log⁡(∏i=1Nprecisioni)N]=brevity-penalty×(∏i=1Nprecisioni)1N  BLEU​=brevity-penalty×exp[*N*∑*i*=1*N*​log(precision*i*​)​]=brevity-penalty×exp⎣⎡​*N*log(∏*i*=1*N*​precision*i*​)​⎦⎤​=brevity-penalty×(*i*=1∏*N*​precision*i*​)*N*1​​    
  Here, N is the maximum n-gram size, (usually 4).

The BLEU score goes from 0 to 1, with higher values indicating better translation quality and 1 signifying a perfect match to the reference translation

### 50. List out the popular NLP task and their corresponding evaluation metrics.

Natural Language Processing (NLP) involves a wide range of tasks, each with its own set of objectives and evaluation criteria. Below is a list of common NLP tasks along with some typical evaluation metrics used to assess their performance:

| **Natural Language Processing(NLP) Tasks** | **Evaluation Metric** |
| --- | --- |
| Part-of-Speech Tagging (POS Tagging) or Named Entity Recognition (NER) | Accuracy, F1-score, Precision, Recall |
| Dependency Parsing | UAS (Unlabeled Attachment Score), LAS (Labeled Attachment Score) |
| Coreference resolution | B-CUBED, MUC, CEAF |
| Text Classification or Sentiment Analysis | Accuracy, F1-score, Precision, Recall |
| Machine Translation | BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit Ordering) |
| Text Summarization | ROUGE (Recall-Oriented Understudy for Gisting Evaluation), BLEU |
| Question Answering | F1-score, Precision, Recall, MRR(Mean Reciprocal Rank) |
| Text Generation | Human evaluation (subjective assessment), perplexity (for language models) |
| Information Retrieval | Precision, Recall, F1-score, Mean Average Precision (MAP) |
| Natural language inference (NLI) | Accuracy, precision, recall, F1-score, Matthews correlation coefficient (MCC) |
| Topic Modeling | Coherence Score, Perplexity |
| Speech Recognition | Word Error Rate (WER) |
| Speech Synthesis (Text-to-Speech) | Mean Opinion Score (MOS) |

The brief explanations of each of the evaluation metrics are as follows:

* **Accuracy**: Accuracy is the percentage of predictions that are correct.
* **Precision**: Precision is the percentage of correct predictions out of all the predictions that were made.
* **Recall**: Recall is the percentage of correct predictions out of all the positive cases.
* **F1-score**: F1-score is the harmonic mean of precision and recall.
* **MAP(Mean Average Precision)**: MAP computes the average precision for each query and then averages those precisions over all queries.
* **MUC(Mention-based Understudy for Coreference)**: MUC is a metric for coreference resolution that measures the number of mentions that are correctly identified and linked.
* **B-CUBED**: B-cubed is a metric for coreference resolution that measures the number of mentions that are correctly identified, linked, and ordered.
* **CEAF**: CEAF is a metric for coreference resolution that measures the similarity between the predicted coreference chains and the gold standard coreference chains.
* **ROC AUC:** ROC AUC is a metric for binary classification that measures the area under the receiver operating characteristic curve.
* **MRR**: MRR is a metric for question answering that measures the mean reciprocal rank of the top-k-ranked documents.
* **Perplexity**: Perplexity is a language model evaluation metric. It assesses how well a linguistic model predicts a sample or test set of previously unseen data. Lower perplexity values suggest that the language model is more predictive.
* **BLEU**: BLEU is a metric for machine translation that measures the n-gram overlap between the predicted translation and the gold standard translation.
* **METEOR**: METEOR is a metric for machine translation that measures the overlap between the predicted translation and the gold standard translation, taking into account synonyms and stemming.
* **WER(Word Error Rate)**: WER is a metric for machine translation that measures the word error rate of the predicted translation.
* **MCC**: MCC is a metric for natural language inference that measures the Matthews correlation coefficient between the predicted labels and the gold standard labels.
* **ROUGE**: ROUGE is a metric for text summarization that measures the overlap between the predicted summary and the gold standard summary, taking into account n-grams and synonyms.
* **Human Evaluation (Subjective Assessment)**: Human experts or crowd-sourced workers are asked to submit their comments, evaluations, or rankings on many elements of the NLP task's performance in this technique.