CUET PG 2026 Data Science Question Paper with Solutions(Memory Based)

Time Allowed :1 Hour 30 Mins	**Maximum Marks :**300	**Total Questions :**75

General Instructions

Read the following instructions very carefully and strictly follow them:

- The exam lasts 90 minutes (1 hour 30 minutes).
- There are 75 Multiple Choice Questions (MCQs) to be answered.
- +4 marks for every correct answer. -1 mark (negative marking) for every incorrect answer. 0 marks for unanswered or un-attempted questions.
- For any discrepancy in questions, the English version is considered final (except for language-specific papers).
- Click one of the four options to choose an answer.
- You must click "Save Next" to confirm your response. Only saved answers are considered for evaluation.
- Use "Mark for Review Next" to flag a question for later. You can unselect or change your answer using the "Clear Response" button.
- All calculations must be done on the Rough Sheets provided at the centre. These must be returned to the invigilator after the exam.

1. What is the time complexity of building a heap from an unsorted array of size n?
(A) $O(n \log n)$
(B) $O(n)$
(C) $O(\log n)$
(D) $O(n^2)$

Correct Answer: (B) $O(n)$

Solution:
Concept: A binary heap can be constructed from an unsorted array using the bottom-up heap construction method (also known as Floyd's Heap Construction Algorithm).
Key ideas:

- The array representation of a heap allows efficient parent–child relationships.
- Starting from the last non-leaf node, we apply the *heapify* operation.
- Although heapify can take $O(\log n)$ time for a single node, most nodes are near the bottom and require much less work.

The total work done across all nodes results in an overall complexity of $O(n)$.
Step 1: Identify the last non-leaf node.
For an array of size n, all elements from index $\lfloor n/2 \rfloor + 1$ to n are leaves. So heap construction begins from node $\lfloor n/2 \rfloor$ and moves upward to the root.

$$i = \left\lfloor \frac{n}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor - 1, \ldots, 1$$

Step 2: Apply heapify to each node.
Each node is heapified to maintain the heap property (max-heap or min-heap).
Heapify takes at most $O(h)$ time, where h is the height of the node. Nodes closer to the leaves have smaller heights and therefore require less work.
Step 3: Compute the total complexity.
The total cost is the sum of heapify costs at each level of the tree:

$$T(n) = \sum_{h=0}^{\log n} \frac{n}{2^{h+1}} \cdot O(h)$$

Evaluating this summation gives:

$$T(n) = O(n)$$

Thus, building a heap from an unsorted array runs in linear time.

Quick Tip

A common misconception is that building a heap takes $O(n \log n)$ because heapify is $O(\log n)$. However, since most nodes are near the bottom of the tree and require minimal work, the total complexity becomes $O(n)$.

---

2. In Operating Systems, which condition ensures that a process cannot be preempted from its resource until it completes its task?
(A) Mutual Exclusion
(B) Hold and Wait
(C) No Preemption
(D) Circular Wait

Correct Answer: (C) No Preemption

Solution:
Concept: In Operating Systems, deadlock occurs when a set of processes are blocked because each process is holding a resource and waiting for another resource held by another process. For a deadlock to occur, four necessary conditions must hold simultaneously:

- Mutual Exclusion
- Hold and Wait
- No Preemption

- Circular Wait

The condition described in the question corresponds to the No Preemption condition.
Step 1: Understand the No Preemption condition.
The No Preemption condition states that once a process acquires a resource, the operating system cannot forcibly take it away. The process must release the resource voluntarily after completing its task.
Step 2: Relate the condition to the question.
The question specifies that a process cannot be preempted from its resource until it completes its task. This directly matches the definition of the No Preemption condition.
Step 3: Conclusion.
Therefore, the operating system condition ensuring that a resource cannot be forcibly taken away from a process is:

No Preemption

Quick Tip

Remember the four necessary conditions for deadlock using the mnemonic: **M H N C** → Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait.

---

3. If a relation is both symmetric and antisymmetric, what must be true about the elements in that relation?
(A) All elements must be comparable
(B) The relation must be transitive
(C) Only pairs of the form (a, a) can exist
(D) The relation must contain all ordered pairs

Correct Answer: (C) Only pairs of the form (a, a) can exist

Solution:
Concept: In discrete mathematics, relations can have different properties. Two important properties are:

- **Symmetric Relation:** If $(a, b) \in R$, then $(b, a) \in R$.
- **Antisymmetric Relation:** If $(a, b) \in R$ and $(b, a) \in R$, then $a = b$.

When a relation is both symmetric and antisymmetric, these definitions must hold simultaneously.
Step 1: Apply the symmetric property.
If a pair (a, b) belongs to relation R, then the pair (b, a) must also belong to R.

$$(a, b) \in R \Rightarrow (b, a) \in R$$

Step 2: Apply the antisymmetric property.
According to antisymmetry, if both (a, b) and (b, a) are present in the relation, then:

$$a = b$$

Step 3: Combine both conditions.
If both properties hold, the only possible pairs are those where:

$$a = b$$

Thus, the relation can contain only pairs of the form:

$$(a, a)$$

These are called **reflexive pairs**.
Step 4: Conclusion.
Therefore, if a relation is both symmetric and antisymmetric, the relation can contain only ordered pairs where both elements are the same.

$$(a, a)$$

Quick Tip

If a relation is both **symmetric** and **antisymmetric**, the only possible pairs are (a, a). Any pair like (a, b) with $a \neq b$ would violate antisymmetry.

---

4. Which activation function is zero-centered and ranges between -1 and 1?
(A) Sigmoid
(B) ReLU
(C) Tanh
(D) Softmax

Correct Answer: (C) Tanh

Solution:
Concept: Activation functions are used in neural networks to introduce non-linearity into the model. Different activation functions have different output ranges and properties.
Some common activation functions include:

- **Sigmoid:** Output range $(0, 1)$
- **ReLU:** Output range $[0, \infty)$
- **Tanh:** Output range $(-1, 1)$
- **Softmax:** Output range $(0, 1)$ and used for probability distributions

The Tanh (Hyperbolic Tangent) function is zero-centered and produces outputs between -1 and 1.
Step 1: Understand the Tanh activation function.
The mathematical form of the Tanh function is:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

This function maps real-valued inputs into the range:

$$-1 < \tanh(x) < 1$$

Step 2: Check the zero-centered property.
When $x = 0$:

$$\tanh(0) = 0$$

Thus, the function is centered around zero, which helps neural networks learn more efficiently during training.
Step 3: Conclusion.
Since the Tanh activation function produces outputs in the range $(-1, 1)$ and is centered at zero, it satisfies the condition given in the question.

Answer: Tanh

Quick Tip

Sigmoid vs Tanh: Sigmoid outputs values between 0 and 1, while **Tanh outputs values between** -1 **and** 1 and is zero-centered, which often leads to faster convergence in neural networks.

---

5. What is the determinant of an identity matrix of any order n?
(A) 0
(B) 1
(C) n
(D) -1

Correct Answer: (B) 1

Solution:
Concept: An identity matrix I_n of order n is a square matrix in which:

- All diagonal elements are 1.
- All non-diagonal elements are 0.

The general form of an identity matrix is:

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

A key property of determinants is that the determinant of a triangular matrix (including diagonal matrices) equals the product of its diagonal elements.
Step 1: Identify the diagonal elements.
For an identity matrix I_n, all diagonal elements are equal to 1.

$$1, 1, 1, \ldots, 1$$

Step 2: Apply the determinant rule.
The determinant of a diagonal matrix is the product of its diagonal elements.

$$\det(I_n) = 1 \times 1 \times 1 \times \cdots \times 1$$

Step 3: Compute the result.
Since the product of n ones is:

$$\det(I_n) = 1$$

Step 4: Conclusion.
Therefore, the determinant of an identity matrix of any order n is:

$$1$$

Quick Tip

The determinant of any **identity matrix** is always 1, regardless of its size. Also, multiplying any matrix by an identity matrix leaves the matrix unchanged.

---

6. In Linear Regression, what is the primary goal of the Ordinary Least Squares (OLS) method?
(A) Maximize the variance of predictions
(B) Minimize the sum of squared residuals
(C) Maximize the correlation between variables
(D) Minimize the number of features

Correct Answer: (B) Minimize the sum of squared residuals

Solution:
Concept: In Linear Regression, the objective is to find the best-fitting line that explains the relationship between the independent variable(s) and the dependent variable.
The Ordinary Least Squares (OLS) method is used to estimate the regression coefficients by minimizing the error between the predicted values and the actual observed values.
The error between actual and predicted values is called the residual.
Step 1: Define the linear regression model.
The simple linear regression model can be written as:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where:

- y_i = actual value
- x_i = input feature
- β_0, β_1 = regression coefficients
- ϵ_i = error term

Step 2: Define residuals.
The residual is the difference between the observed value and the predicted value.

$$e_i = y_i - \hat{y}_i$$

Step 3: Apply the OLS objective function.
The OLS method minimizes the sum of squared residuals (SSR):

$$\text{SSR} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Squaring the residuals ensures:

- All errors become positive
- Larger errors are penalized more heavily

Step 4: Conclusion.
Therefore, the main objective of the Ordinary Least Squares method is to:

Minimize the sum of squared residuals

Quick Tip

OLS finds the regression line that minimizes the squared distance between the **actual values** and the **predicted values**. This produces the **best-fitting line** for the data.

---

7. Which data structure is used by the system to implement recursion?
(A) Queue
(B) Stack
(C) Linked List
(D) Heap

Correct Answer: (B) Stack

Solution:
Concept: Recursion occurs when a function calls itself during execution. To manage multiple function calls, the system uses a call stack.
A stack is a linear data structure that follows the Last In First Out (LIFO) principle. Each function call creates a new activation record (stack frame) that is pushed onto the stack.
Step 1: Understand how recursion works.
When a recursive function is called, the system:

- Stores the current function state (parameters, return address, local variables).
- Pushes this information onto the stack as a stack frame.

Step 2: Function calls and stack behavior.
Each recursive call is placed on top of the stack. When the base case is reached, the stack begins to unwind and function calls are removed in reverse order.
Example:

$$\text{factorial}(3) \rightarrow \text{factorial}(2) \rightarrow \text{factorial}(1)$$

Stack behavior:

$$\text{Push } f(3) \rightarrow f(2) \rightarrow f(1)$$

Then they are popped in reverse order.
Step 3: Conclusion.
Since recursion requires storing and retrieving function calls in Last In First Out order, the system uses a:

Stack

Quick Tip

Recursion internally uses the **call stack**. Each recursive call adds a new stack frame, and when the base case is reached, frames are removed in reverse order.

---

8. A high variance in a machine learning model is a primary indicator of which problem?
(A) Underfitting
(B) Overfitting
(C) High bias
(D) Data normalization

Correct Answer: (B) Overfitting

Solution:
Concept: In machine learning, bias and variance are two sources of error that affect model performance.

- **Bias:** Error due to overly simple assumptions in the learning algorithm.
- **Variance:** Error caused by the model being too sensitive to the training data.

A model with high variance learns the training data very well but fails to generalize to new, unseen data.
Step 1: Understand high variance.
High variance occurs when the model captures not only the underlying patterns but also the noise in the training data.
This causes the model to perform:

- Very well on training data
- Poorly on testing or new data

Step 2: Identify the associated problem.
This situation is known as overfitting, where the model becomes too complex and fits the training dataset too closely.
Step 3: Conclusion.
Therefore, a machine learning model with high variance typically indicates:

Overfitting

Quick Tip

High Bias → Underfitting (model too simple)
High Variance → Overfitting (model too complex)

---

9. In a Binary Search Tree, which traversal method produces the elements in non-decreasing order?
(A) Preorder Traversal
(B) Inorder Traversal
(C) Postorder Traversal
(D) Level Order Traversal

Correct Answer: (B) Inorder Traversal

Solution:
Concept: A Binary Search Tree (BST) is a special type of binary tree that satisfies the following property:

- All elements in the left subtree are smaller than the root.
- All elements in the right subtree are greater than the root.
- Both left and right subtrees are also Binary Search Trees.

Because of this property, performing an Inorder Traversal on a BST visits nodes in sorted (non-decreasing) order.
Step 1: Understand Inorder Traversal.
Inorder traversal follows the sequence:

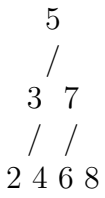
Left Subtree → Root → Right Subtree

Step 2: Apply BST property.
Since:

- Left subtree contains smaller elements
- Root is the next element

- Right subtree contains larger elements

Visiting nodes in this order naturally produces a sorted sequence.
Step 3: Example.
Consider the following BST:



Applying Inorder Traversal:

$$2 \to 3 \to 4 \to 5 \to 6 \to 7 \to 8$$

This sequence is in non-decreasing (sorted) order.
Step 4: Conclusion.
Therefore, the traversal that produces elements in sorted order in a Binary Search Tree is:

Inorder Traversal

Quick Tip

For any **Binary Search Tree**, performing an **Inorder Traversal** always returns the elements in **sorted order**.

---

10. What is the probability of getting a sum of 9 when two fair dice are rolled simultaneously?
(A) $\frac{1}{12}$
(B) $\frac{1}{9}$
(C) $\frac{1}{8}$
(D) $\frac{1}{6}$

Correct Answer: (B) $\frac{1}{9}$

Solution:
Concept: When two fair dice are rolled, each die has 6 possible outcomes. Therefore, the total number of possible outcomes is:

$$6 \times 6 = 36$$

Each outcome is equally likely.
Step 1: List all outcomes whose sum is 9.
The pairs of numbers whose sum equals 9 are:

$$(3, 6), (4, 5), (5, 4), (6, 3)$$

Thus, the number of favorable outcomes is:

4

Step 2: Apply the probability formula.
The probability of an event is given by:

$$P(E) = \frac{\text{Number of favorable outcomes}}{\text{Total number of possible outcomes}}$$

Substituting the values:

$$P(\text{sum } = 9) = \frac{4}{36}$$

Step 3: Simplify the fraction.

$$\frac{4}{36} = \frac{1}{9}$$

Step 4: Conclusion.
Therefore, the probability of obtaining a sum of 9 when two fair dice are rolled is:

$$\frac{1}{9}$$

Quick Tip

When two dice are rolled, the total possible outcomes are always 36. To find probability, count the number of favorable pairs that produce the required sum.

---

11. Which layer of the OSI model is responsible for IP addressing and routing?
(A) Data Link Layer
(B) Network Layer
(C) Transport Layer
(D) Session Layer

Correct Answer: (B) Network Layer

Solution:
Concept: The OSI (Open Systems Interconnection) model is a conceptual framework used to understand how data travels across a network. It consists of seven layers, each responsible for specific networking tasks.
The seven layers are:

- Physical Layer
- Data Link Layer
- Network Layer
- Transport Layer

- Session Layer
- Presentation Layer
- Application Layer

The Network Layer is responsible for logical addressing and routing of data packets across networks.
Step 1: Understand IP addressing.
The Network Layer assigns and handles logical addresses, commonly known as IP addresses.
Example:

192.168.1.1

These addresses uniquely identify devices across networks.
Step 2: Understand routing.
Routing refers to determining the best path for data packets to travel from the source device to the destination device across multiple interconnected networks.
Routers operate primarily at the Network Layer.
Step 3: Protocols used at the Network Layer.
Some important protocols operating at this layer include:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Routing protocols such as OSPF and RIP

Step 4: Conclusion.
Since IP addressing and routing are handled by the Network Layer, the correct answer is:

Network Layer

Quick Tip

Remember: **Routers operate at the Network Layer**, where logical addressing (IP addresses) and packet routing are performed.

---

12. How many edges are there in a complete graph with n vertices?
(A) $n(n-1)$
(B) $\frac{n(n-1)}{2}$
(C) n^2
(D) $\frac{n(n+1)}{2}$

Correct Answer: (B) $\frac{n(n-1)}{2}$

Solution:
Concept: A complete graph, denoted by K_n, is a simple undirected graph in which every pair of distinct vertices is connected by exactly one edge.

Thus, each vertex is connected to every other vertex in the graph.
Step 1: Count connections from each vertex.
In a complete graph with n vertices, each vertex is connected to:

$$n - 1$$

other vertices.
Step 2: Compute total connections.
If we count edges from all vertices, we initially get:

$$n(n - 1)$$

However, this counts each edge twice (once from each endpoint).
Step 3: Correct the double counting.
To avoid counting edges twice, we divide by 2:

$$\text{Number of edges} = \frac{n(n - 1)}{2}$$

Step 4: Conclusion.
Therefore, the number of edges in a complete graph with n vertices is:

$$\frac{n(n - 1)}{2}$$

Quick Tip

In a complete graph K_n, every vertex connects to all other vertices. Hence, the total number of edges is always $\frac{n(n-1)}{2}$.

---

13. In SQL, which clause is used to filter the results of an aggregate function?
(A) WHERE
(B) GROUP BY
(C) HAVING
(D) ORDER BY

Correct Answer: (C) HAVING

Solution:
Concept: In SQL, aggregate functions such as:

- `SUM()`
- `COUNT()`
- `AVG()`
- `MAX()`
- `MIN()`

are used to perform calculations on a set of rows and return a single result.
To filter results based on aggregate values, SQL uses the HAVING clause.
Step 1: Understand the WHERE clause.
The WHERE clause filters rows before grouping takes place.
Example:

```
SELECT * FROM Employees WHERE Salary > 50000;
```

It cannot be used with aggregate functions directly.
Step 2: Understand the HAVING clause.
The HAVING clause is used to filter results after aggregation.
Example:

```
SELECT Department, COUNT(*)
FROM Employees
GROUP BY Department
HAVING COUNT(*) > 5;
```

This query returns only those departments having more than 5 employees.
Step 3: Conclusion.
Since filtering conditions applied to aggregate functions are handled using the HAVING clause, the correct answer is:

HAVING

Quick Tip

WHERE filters rows before grouping, while **HAVING** filters results after aggregate functions are applied.

---

14. What is the value of $(AB)^T$ in matrix algebra?
(A) $A^T B^T$
(B) $B^T A^T$
(C) AB
(D) $A^T B$

Correct Answer: (B) $B^T A^T$

Solution:
Concept: In matrix algebra, the transpose of a matrix is obtained by interchanging its rows and columns.
If A is an $m \times n$ matrix, then its transpose A^T is an $n \times m$ matrix.
One important property of matrix transpose is related to the product of two matrices.
Step 1: Recall the transpose property of matrix multiplication.
For any two matrices A and B whose product AB is defined, the transpose of their product satisfies:

$$(AB)^T = B^T A^T$$

This means the order of multiplication reverses when taking the transpose.
Step 2: Understand the reversal of order.
Matrix multiplication is not commutative, meaning:

$$AB \neq BA$$

Therefore, when taking the transpose of a product, the matrices must appear in reverse order.
Step 3: Conclusion.
Using the transpose rule for matrix multiplication:

$$(AB)^T = B^T A^T$$

Hence, the correct answer is:

$$B^T A^T$$

Quick Tip

When taking the transpose of a matrix product, **reverse the order of multiplication**:

$$(AB)^T = B^T A^T$$

---

15. Which ensemble technique reduces variance by training multiple trees on different subsets of data?
(A) Boosting
(B) Bagging
(C) Stacking
(D) Gradient Descent

Correct Answer: (B) Bagging

Solution:
Concept: In machine learning, ensemble methods combine multiple models to improve predictive performance. One such method is Bagging (Bootstrap Aggregating).
Bagging reduces variance by training multiple models (often decision trees) on different random subsets of the training dataset and then combining their predictions.
Step 1: Understand Bagging.
Bagging works by creating several training datasets using bootstrap sampling (sampling with replacement). Each dataset is used to train a separate model.

- Each model sees a slightly different subset of the data.
- The models are trained independently.

Step 2: Combine predictions.
After training multiple models:

- For classification, predictions are combined using majority voting.
- For regression, predictions are combined using averaging.

This reduces model variance and improves stability.
Step 3: Example.
A popular algorithm based on bagging is the Random Forest, where many decision trees are trained on different subsets of the dataset.
Step 4: Conclusion.
Since the technique described trains multiple trees on different subsets of data to reduce variance, the correct answer is:

Bagging

Quick Tip

Bagging (Bootstrap Aggregating) reduces variance by training multiple models on random subsets of data and combining their predictions.
