```
# Que 1. What is a parameter?
# Ans 1. A parameter is a value that a machine learning model learns from the training data to make accurate predictions

#Que 2. what is correlation ?
#Ans 2. Correlation is a statistical measure that shows how strongly two variables are related to each other.
#.      It tells us whether, and how, two things move together.
#.      If both go up together → Positive correlation.
#       If one goes up and the other goes down → Negative correlation.
#       If they don't affect each other → No correlation.

#Que 3 what does negative correlation mean ?
#Ans 3 Negative correlation means that when one variable increases, the other decreases
#      Negative correlation = Two variables move in opposite directions.
#      Correlation value:
#
#.      Negative correlation values are between −1 and 0
#      −1 means perfect negative relationship

#Que 4 Define Machine Learning. What are the main components in Machine Learning?
#Ans 4 Machine Learning is a type of technology where computers learn from data and improve their performance without being d
#      Main Components of Machine Learning:
#1. Data
#      Collection of information (numbers, text, images, etc.)
#      Example: Student marks, weather reports, product reviews
#2. Model
#      A system or function that makes predictions
#      It learns patterns from the data
#3. Algorithm
#      A step-by-step process used to train the model
#      Example: Linear Regression, Decision Tree, KNN
#4. Training
#      The process of teaching the model using data
#      Model finds patterns in the training data
#5. Testing
#      After training, the model is tested on new (unseen) data
#      To check how well it has learned
#6. Features
#      The input variables used to make predictions
#       Example: Age, Salary, Hours studied
#7. Labels (Target)
#      The correct output we want the model to predict
#       Example: Pass/Fail, Price of house, Disease/No disease

#Qeu 5 How does loss value help in determining whether the model is good or not?
#Ans 5 Loss value tells us how far the model’s prediction is from the actual correct value

#      How it helps in determining if the model is good:
#      Low Loss = Good Model
#      If the loss value is small, it means the model's predictions are close to the actual answers.
#      So, the model is performing well.

#      High Loss = Bad Model
#      If the loss value is large, it means the model is making many mistakes.
#      So, the model needs improvement.

#      Think of it like:
#      Loss is like exam marks (but in reverse):
#      0 loss = full marks (perfect model)
#      High loss = lots of wrong answers

# Example:
#      If you’re training a model to predict house prices:
#      Actual price = ₹50 lakh
#      Model predicts = ₹52 lakh
#      Loss = ₹2 lakh
#      Smaller this number, better the model.

#      During Training:
#      The model keeps adjusting its internal parameters to reduce the loss.

#Que 6 What are continuous and categorical variables?
#Ans 6 Continuous Variables:
#      Continuous variables are numbers that can take any value within a range.
#      They can be measured.
#      Can have decimal values.
#      Infinite possibilities between two numbers.

#      Examples:
#      Height (like 5.4 ft),Weight (70.5 kg),Temperature (36.6°C),Price (₹999.99)

#      Categorical Variables:
```

```
#      Categorical variables are values that represent groups or categories.
#      They can be counted, not measured.
#      Usually labels or names.
#      No in-between values.

#      Examples:
#      Gender (Male, Female), Color (Red, Green, Blue), City (Delhi, Mumbai), Yes/No answ

#Que 7 What do you mean by training and testing a dataset?
#Ans 7 Training a Dataset:
#      Training a dataset means using the data to teach the machine learning model to learn patterns and relationships.
#      The model looks at the inputs and correct outputs.
#      It learns from examples so it can make predictions in the future.
# Example:
#      Giving a child many math problems with answers so they can learn how to solve similar ones.

#      Testing a Dataset:
#      Testing a dataset is used to check how well the trained model performs on new, unseen data.
#      It’s not shown during training.
#      Helps to measure accuracy and performance of the model.
# Example:
#      After learning, the child is given a test with new questions to see how well they have learned.

#Que 8 What is sklearn.preprocessing?
#Ans 8 sklearn.preprocessing is a module in Scikit-learn (sklearn) that provides functions to prepare and
#      transform your data before training a machine learning model.

#.     Machine learning models do not work well if the input data:
#      is not in the right scale
#      contains categories as text
#      has missing or messy values
#      So, sklearn.preprocessing helps to:
#      Scale numbers (e.g., from 0 to 1)
#      Convert text (categorical data) into numbers
#      Normalize data
#      Handle missing values, etc.

#Que 9 What is a Test set?
#Ans 9 A test set is a separate portion of the dataset that is used after training to evaluate how well
#      a machine learning model performs on completely unseen data. It helps measure the model’s ability
#      to generalize to new, real-world inputs.

#Que 10 How do we split data for model fitting (training and testing) in Python?
#       How do you approach a Machine Learning problem?

#Ans 10 We use the function train_test_split() from sklearn.model_selection.
#.      from sklearn.model_selection import train_test_split

# Example data
#       X = [[1], [2], [3], [4], [5], [6]]     # Features
#       y = [10, 20, 30, 40, 50, 60]           # Labels (Target)

#       Split data (80% training, 20% testing)
#       X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

#       print("Training data:", X_train)
#       print("Testing data:", X_test)

#.      Parameters of train_test_split():
#      | Parameter      | Description                                         |
#      | -------------- | --------------------------------------------------- |
#      | `test_size`    | Percentage of data for testing (e.g., 0.2 = 20%)    |
#      | `random_state` | Ensures same split every time (for reproducibility) |

#How do you approach a Machine Learning problem?
#    Step-by-Step Approach:
#    Understand the Problem
#    What is the goal? (Prediction, Classification, etc.)

#     What kind of output is expected?
#     Collect Data
#     Get clean and relevant data
#     Use CSV, databases, APIs, etc.
#     Explore the Data (EDA)
#     Use statistics and graphs to understand patterns

#     Check for missing values, outliers
#     Preprocess the Data
#     Handle missing values

#     Convert categories to numbers
#     Scale/normalize data
#     Split the Data
#     Use train_test_split() to divide into training and testing sets

#     Choose a Model
#     Depending on the problem: Linear Regression, Decision Tree, SVM, etc.

#     Train the Model
#     Fit the model using training data (model.fit())

#     Evaluate the Model
#     Use testing data to check accuracy, precision, recall, etc.

#     Improve the Model
#     Tune hyperparameters, try other models, or get more data

#     Deploy the Model
#     Use it in an app, website, or tool for real-world predictions

#Que 11 Why do we have to perform EDA before fitting a model to the data?
#Ans 11 Exploratory Data Analysis (EDA) is performed before fitting a model because it helps us:
#.    1. Understand the Data
#     Identify the types of variables (categorical, continuous)
#     Understand data distributions and relationships

#.   2.  Detect Missing Values
#    Missing data can affect model accuracy
#    EDA helps locate and handle them properly

#    3. Identify Outliers
#    Outliers can mislead the model during training
#    EDA highlights such unusual data points

#    4. Check Data Quality
#    Find inconsistencies, duplicates, wrong formats
#    Ensure the data is clean and usable

#    5. Select Useful Features
#    Identify which features are important or irrelevant
#    Helps in feature selection and dimensionality reduction

#    6. Decide on Preprocessing Steps
#    Helps choose scaling, encoding, or transformation techniques based on data nature


#Que 12 What is correlation?
#Ans 12 Correlation is a statistical measure that shows the strength and direction
#       of the relationship between two variables.

#       It tells how much one variable changes when the other changes.
#       Measured by a value between -1 and +1.

# Correlation Values:
#      Correlation Value  Meaning
#     +1  Perfect positive correlation
#      0  No correlation
#     -1  Perfect negative correlation

# Types of Correlation:
#      Positive Correlation (+):
#      As one variable increases, the other also increases.
#      Example: Height and weight

#      Negative Correlation (-):
#      As one variable increases, the other decreases.
#      Example: Exercise time and body fat

#      No Correlation (0):
#      Variables do not affect each other.
#      Example: Shoe size and intelligence

#Que 13 What does negative correlation mean?
#Ans 13 Negative correlation means that when one variable increases, the other variable decreases.

#     The two variables move in opposite directions.
#     Correlation value lies between -1 and 0.
#     A correlation of -1 means a perfect negative relationship.

# Examples:
#     Variable A  Variable B  Relationship
#     Exercise time ↑ Body fat ↓  More exercise, less body fat
#     Product price ↑ Sales ↓ Higher price, fewer people buy
#     Absenteeism ↑ Exam scores ↓ More absences, lower marks.
```

```
#Que 14 How can you find correlation between variables in Python?
#Ans 14 You can find correlation between variables in Python using the pandas library.

import pandas as pd

# Create sample DataFrame
data = {
    'Hours_Studied': [1, 2, 3, 4, 5],
    'Exam_Score': [50, 60, 65, 70, 80],
    'Sleep_Hours': [9, 8, 7, 6, 5]
}

df = pd.DataFrame(data)

# Find correlation matrix
correlation = df.corr()

print(correlation)
```

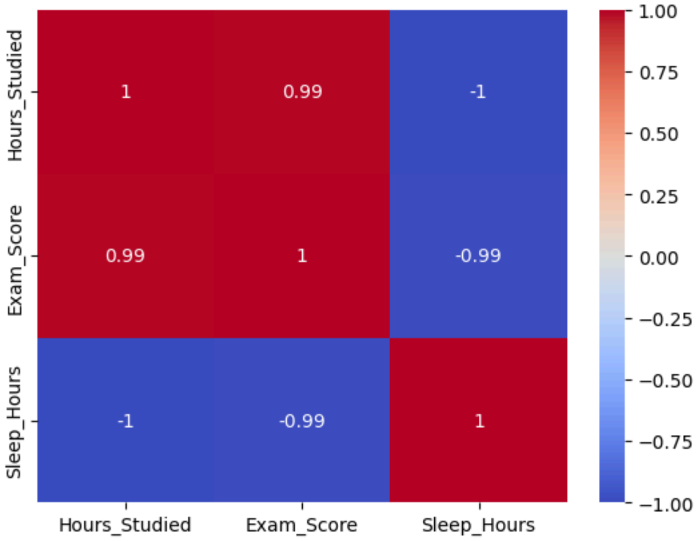
```
               Hours_Studied  Exam_Score  Sleep_Hours
Hours_Studied       1.000000    0.989949    -1.000000
Exam_Score          0.989949    1.000000    -0.989949
Sleep_Hours        -1.000000   -0.989949     1.000000
```

```
#    df.corr() calculates Pearson correlation (by default).
#    Correlation values range from -1 to +1:

#    +1 = Strong positive correlation
#     0 = No correlation
#    -1 = Strong negative correlation

import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.show()
```



```
#Que 15 What is causation? Explain difference between correlation and causation with an example
#Ans 15 Causation means one variable directly affects or causes a change in another variable.
#       In simple terms, if A causes B, then changing A will definitely bring a change in B

#| Feature           | Correlation                                 | Causation                                     |
#| ----------------- | ------------------------------------------- | --------------------------------------------- |
#| Meaning           | Two variables move together (linked)        | One variable **causes** a change in the other |
#| Relationship Type | Can be strong or weak, positive or negative | Always **direct and cause-effect**            |
#| Proof             | Only shows **association**                  | Shows **reason-effect** relationship          |
#| Example Allowed?  | Yes — even by coincidence                   | No — must be **provable**                     |

# Example:
#.    Correlation Only:
#     Ice cream sales .

# Drowning cases
#      Correlated, but one does not cause the other.
#      Hidden cause = Hot weather (people swim more & eat ice cream).

# Causation:
#      More hours studied  Higher exam scores.
#      Studying directly causes better performance.

#Que 16 What is an Optimizer? What are different types of optimizers? Explain each with an example.
#Ans 16 An optimizer is an algorithm used to adjust the weights of a
#       machine learning model to minimize the loss function and improve accuracy.

#.  The goal of training is to reduce error (loss).
#   Optimizers help the model learn faster and better.
#.  They guide the model to find the best weights for predictions.

#   A mountain climber trying to reach the lowest valley (minimum loss).
#.  The optimizer decides how big or small the steps should be

#.  1. Gradient Descent (GD)
#.     Basic optimizer.
#.     Updates weights in the direction of the negative gradient.
#.     Slow for large datasets.

#.   w = w - learning_rate * ∂L/∂w
#    Example:
#    If loss is high and the gradient says “go left,” the model updates the weights in that direction.

#.   2. Stochastic Gradient Descent (SGD)
#    Updates weights after every data point (not whole dataset).
#    Faster, but more noise.

# Example:
#    Used when data is too large to process in one go (like online learning).
#    3. Mini-Batch Gradient Descent
#    Combines benefits of GD and SGD.
#    Updates weights after a small batch of data points.

#    4. Momentum
#    Adds a "velocity" term to updates to avoid oscillations.
#.   Helps the model move faster in right direction.

# Example:
#    Helps reach minimum loss faster by keeping memory of past gradients.

#    5. Adam (Adaptive Moment Estimation)
#.   Most commonly used optimizer today.
#.   Combines Momentum + RMSProp.
#.   Adapts learning rate for each parameter.

# Example:
#.   Used in deep learning models like CNNs, RNNs.

#     6. RMSProp (Root Mean Square Propagation)
#     Adjusts learning rate based on the average of recent gradients.
#.    Works well for non-stationary objectives (e.g., time series).

#Que 17 What is sklearn.linear_model ?
#Ans 18 sklearn.linear_model is a module in Scikit-learn (sklearn) that provides various linear models for regression and cl

#    his module includes classes and functions for:
#    Linear Regression
#    Logistic Regression
#    Ridge Regression
#    Lasso Regression
#    ElasticNet

# Popular Models in sklearn.linear_model

#| Model                  | Purpose                            | Use Case Example              |
#| ---------------------- | ---------------------------------- | ----------------------------- |
#| `LinearRegression()`   | Predict continuous values          | Predict house prices          |
#| `LogisticRegression()` | Classification problems            | Spam or Not Spam              |
#| `Ridge()`              | Regularized linear regression (L2) | Reduce overfitting            |
#| `Lasso()`              | Regularized linear regression (L1) | Feature selection             |
#| `ElasticNet()`         | Combination of Lasso + Ridge       | Works well when many features |


#Que 18 What does model.fit() do? What arguments must be given?
#Ans 19 model.fit() is a method used to train a machine learning model on the given data.
#       It takes the training data (features and labels) and helps the model learn the
#       patterns or relationships between input and output.
#      "fit" means the model is fitting itself to the training data — it learns the rules, trends, and patterns from the dat
#    model.fit(X_train, y_train)
#    X_train: Input features (independent variables)
#    y_train: Output labels or targets (dependent variable)

#    from sklearn.linear_model import LinearRegression
#    model = LinearRegression()
#    model.fit(X_train, y_train)   # X_train = features, y_train = target

#    After this, the model is trained and ready to make predictions using:
#    model.predict(X_test)

#Que 19 What does model.predict() do? What arguments must be given?
#And 19 model.predict() is used after training the model, and it predicts the output for new, unseen input data.
#       It applies what the model has learned during training (via model.fit()) to make predictions on test or real-time dat
#   Syntax:
#.  predictions = model.predict(X_test)

#.  Required Argument:
#.  Argument       Description
#.  X_test.      Input features for which predictions are needed. These should be in the same format as the data used for tr

#example
#from sklearn.linear_model import LinearRegression

#model = LinearRegression()
#model.fit(X_train, y_train)

# Now predict the outputs for test data
#y_pred = model.predict(X_test)

#Que 20 What are continuous and categorical variables?
#Ans 20 In machine learning and statistics, variables (or features) are the data inputs we use to build models.
#       These variables are mainly divided into two types:

#    Continuous Variables
#    A continuous variable is a variable that can take any numeric value within a range, including decimals and fractions.

#  Examples:
#. Height (e.g., 170.5 cm)
#. Weight (e.g., 62.3 kg)
#. Temperature (e.g., 36.7°C)
#. Salary (e.g., ₹45,000.75)

# Characteristics:
#    Measured, not counted
#    Infinite number of values within a range
#    Supports mathematical operations (like mean, standard deviation)

# 2. Categorical Variables
#    A categorical variable is a variable that represents categories or groups.
#    These are not numerical (even if numbers are used as labels).

# Examples:
#    Gender: Male, Female
#    Marital Status: Single, Married, Divorced
#    City: Delhi, Mumbai, Patna
#    Rating: "Low", "Medium", "High"

# Characteristics:
#    Values are labels or categories
#    Can be nominal (no order) or ordinal (ordered)
#    Often need to be encoded for machine learning

# Simple Difference:
#.    | Feature Type    | Example         | Type                  |
#     | --------------- | --------------- | --------------------- |
#     | Height (in cm)  | 170.2           | Continuous            |
#     | Gender          | Male/Female     | Categorical           |
#     | City            | Mumbai, Delhi   | Categorical           |
#     | Salary          | ₹55,000         | Continuous            |
#     | Education Level | High School, PG | Categorical (Ordinal) |

#Que 21 What is feature scaling? How does it help in Machine Learning?
#Ans 21 Feature scaling is the process of normalizing or standardizing the values of independent variables (features) so tha

#      Many ML algorithms (like KNN, SVM, Gradient Descent, etc.) use distance or magnitude to make predictions.
#      If one feature has large values (e.g., Salary in ₹lakhs) and another has small values (e.g., Age in years),
#      the model may give more importance to higher-value features, which can mislead the learning process.

#| Age (years) | Salary (₹) |
#| ----------- | ---------- |
#| 22          | 400000     |
#| 30          | 800000     |

#      Here, Salary has much larger values than Age.
#      Without scaling, algorithms may think Salary is more important than Age, even if that’s not true.

#    Types of Feature Scaling:
#    Scales features to range [0, 1]
#    Min–Max Scaling (Normalization)
#    Scales features to range [0, 1]

#.   Standardization (Z–score scaling)
#    Scales features to have mean = 0 and standard deviation = 1

#Que 22 How do we perform scaling in Python?
#Ans 22 Feature Scaling is the process of transforming the values of numerical features into a common scale,
#       so that no single feature dominates or influences the model more than others.
#       Why is Feature Scaling Important?
#.     Many machine learning algorithms (like KNN, SVM, linear regression, gradient descent) are sensitive to the scale of ir
#     If features are on different scales (e.g., age in 10s and salary in 10000s), the model may give more importance to lar
#     which can lead to poor performance.
#     Scaling ensures fair treatment of all features by bringing them to the same level.

#| Feature | Before Scaling | After Scaling (0–1) |
#| ------- | -------------- | ------------------- |
#| Age     | 20             | 0.0                 |
#| Age     | 60             | 1.0                 |
#| Salary  | 20000          | 0.0                 |
#| Salary  | 100000         | 1.0                 |

#Que 23 What is sklearn.preprocessing?
#Ans 23 sklearn.preprocessing is a module in Scikit–learn (a popular Python machine learning
#       library) that provides functions and classes to prepare or transform raw data before feeding it to a machine learnir

#  Why is it used?
#     Scaling numerical features (e.g., using StandardScaler, MinMaxScaler)
#     Encoding categorical features (e.g., LabelEncoder, OneHotEncoder)
#     Normalizing data
#     Handling missing values (using imputation)
#     Transforming data distributions

#  Common Tools in sklearn.preprocessing:
#| Tool             | Use                                              |
#| ---------------- | ------------------------------------------------ |
#| `StandardScaler` | Scales data to have mean = 0 and std = 1         |
#| `MinMaxScaler`   | Scales data between a given range (default 0 to 1) |
#| `LabelEncoder`   | Converts categorical labels into numeric form      |
#| `OneHotEncoder`  | Converts categorical values into binary vectors    |
#| `Normalizer`     | Scales each data point (row) to have unit norm     |
#| `Binarizer`      | Converts values above a threshold to 1, else 0     |

#Que 24 How do we split data for model fitting (training and testing) in Python?
#Ans 24 we use train_test_split() from the sklearn.model_selection module to divide our dataset into training and testing se
#    Why split the data?
#    Training set is used to teach the model.
#    Testing set is used to evaluate the model’s performance on unseen data.
#    This helps us check if the model is overfitting (memorizing the data) or generalizing well.

from sklearn.model_selection import train_test_split

# Sample data
X = [[1], [2], [3], [4], [5], [6]]
y = [10, 20, 30, 40, 50, 60]

# Splitting into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training Data:", X_train)
print("Testing Data:", X_test)
```

```
Training Data: [[6], [3], [5], [4]]
Testing Data: [[1], [2]]
```

```
#  Parameters of train_test_split()
#   | Parameter      | Meaning                                                         |
#| -------------- | --------------------------------------------------------------- |
#| `X`            | Features (independent variables)                                |
#| `y`            | Target (dependent variable)                                     |
#| `test_size`    | Percentage of data to use for testing (e.g., 0.2 = 20%)         |
#| `random_state` | Ensures reproducibility by fixing the split pattern             |
#| `shuffle`      | Whether to shuffle the data before splitting (default is `True`) |
#Que 25 Explain data encoding?
#Ans 25 Data Encoding is the process of converting categorical data (like labels or text) into
#       numerical form, so that machine learning models can understand and use it for training and predictions.
#       Most ML models cannot process strings or categories directly — they need numbers.

#.   Types of Encoding:
#   Label Encoding
#['Red', 'Blue', 'Green'] → [2, 0, 1]

#.  One-Hot Encoding
#. ['Red', 'Blue', 'Green'] →

# Red   Blue  Green
# 1     0     0
# 0     1     0
# 0     0     1

# | Type             | Use Case                 | Output Form               |
# | ---------------- | ------------------------ | ------------------------- |
# | Label Encoding   | Ordinal data             | Single column of integers |
# | One-Hot Encoding | Nominal data (no order)  | Multiple binary columns   |
```

```
Start coding or generate with AI.
```