Linear Regression is a supervised machine learning algorithm used for **predicting a continuous dependent variable** based on one or more independent variables. It assumes a linear relationship between the input variables (X) and the output variable (y).

1. Simple Linear Regression

Formula:

y=β0+β1x+εy = \beta_0 + \beta_1 x + \varepsilony=β0+β1x+ε

Where:

- yyy: Dependent variable
- xxx: Independent variable
- β0\beta_0β0: Intercept
- β1\beta_1β1: Slope
- ε\varepsilonε: Error term

? 2. Multiple Linear Regression

Formula:

y=β0+β1x1+β2x2+⋯+βnxn+εy = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilony=β0+β1x1+β2x2+⋯+βnxn+ε

Used when there are **multiple input features**.

Steps in Linear Regression

1. **Import Libraries**
2. **Load Dataset**
3. **Preprocess Data** (handle missing values, encode categories)
4. **Split Data** into train and test
5. **Fit Linear Regression Model**
6. **Make Predictions**

Step 1: Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error
```

? **Step 2: Load Dataset**

```
df = pd.read_csv('your_dataset.csv')  # Replace with your actual file
print(df.head())
```

Step 3: Data Preprocessing

- Handle missing values
- Encode categorical variables (if any)
- Normalize/scale if needed

```
df.dropna(inplace=True)  # Simple method to handle missing values
```

Step 4: Define Features (X) and Target (y)

Simple Regression Example:

```
X = df[['Hours_Studied']]  # Independent variable
y = df['Score']            # Dependent variable
```

Multiple Regression Example:

```
X = df[['Hours_Studied', 'Sleep_Hours', 'Practice_Problems']]
y = df['Score']
```

Step 5: Split Dataset into Train & Test

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
```

? **Step 7: Make Predictions**

```
y_pred = model.predict(X_test)
```

? **Step 8:? Predict on New Data # For simple regression**

```
print(model.predict([[6]]))  # Predict for 6 hours studied

# For multiple regression
print(model.predict([[6, 7, 3]]))  # 6 hrs study, 7 hrs sleep, 3 practice problems
```

Step 10: (Optional) Visualize the Model

For Simple Linear Regression:

```
plt.scatter(X, y, color='blue')
plt.plot(X, model.predict(X), color='red')
plt.xlabel("Hours Studied")
plt.ylabel("Score")
plt.title("Linear Regression Line")
plt.show()
```