In [7]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_s
```

In [15]:
```
try:
    df = pd.read_csv("Student_Performance.csv")
    print("Dataset successfully loaded")
    print(f"Initial shape: {df.shape}")
except FileNotFoundError:
    print("Failed to load dataset")
```

```
Dataset successfully loaded
Initial shape: (10000, 6)
```

In [16]:
```
df.head()
```

Out[16]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	Yes	9	1	91.0
1	4	82	No	4	2	65.0
2	8	51	Yes	7	2	45.0
3	5	52	Yes	5	2	36.0
4	7	75	No	8	5	66.0

In [17]:
```
df['Extracurricular Activities'] = df['Extracurricular Activities'].map({
    'Yes': 1,
    'No': 0
})

df.head()
```

Out[17]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	1	9	1	91.0
1	4	82	0	4	2	65.0
2	8	51	1	7	2	45.0
3	5	52	1	5	2	36.0
4	7	75	0	8	5	66.0

In [21]:
```
X = df.drop('Performance Index', axis=1)
y = df['Performance Index']
```

In [28]:
```
y_scaled = y / 100.0
```

In [29]:
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y_scaled, test_size=0.2, random_state=42
)
```

In [30]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[30]:
```
▾ LinearRegression
▶ Parameters
```

In [31]:
```
y_pred_scaled = model.predict(X_test)
```

In [32]:
```
y_pred = np.clip(y_pred_scaled * 100, 0, 100)
y_test_actual = y_test * 100  # rescale test targets for evaluation
```

In [33]:
```
mae = mean_absolute_error(y_test_actual, y_pred)
mse = mean_squared_error(y_test_actual, y_pred)
r2 = r2_score(y_test_actual, y_pred)

print("✅ Model Evaluation Metrics:")
print(f"MAE: {mae:.2f}")
print(f"MSE: {mse:.2f}")
print(f"R² Score: {r2:.4f}")
```

```
✅ Model Evaluation Metrics:
MAE: 1.61
MSE: 4.08
R² Score: 0.9890
```

In [36]:
```
with open("student_performance_model.pkl", "wb") as f:
    pickle.dump(model, f)

print("✅ Model saved as student_performance_model.pkl")
```

```
✅ Model saved as student_performance_model.pkl
```

In [ ]: