



Multilinear Regression

```
In [4]: import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

# Sample dataset
data = {
    "Hours_Studied": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    "Hours_Slept": [7, 6, 8, 5, 6, 7, 8, 5, 6, 7],
    "Test_Score": [35, 38, 50, 52, 60, 65, 70, 75, 85, 90]
}

df = pd.DataFrame(data)

# Features and target
X = df[["Hours_Studied", "Hours_Slept"]]
y = df["Test_Score"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

# Predictions
y_pred = model.predict(X_test)

# --- Coefficients & Intercept ---
b1, b2 = model.coef_
intercept = model.intercept_
print(f"Intercept (b0): {intercept:.3f}")
print(f"Coefficient b1 (Hours_Studied): {b1:.3f}")
print(f"Coefficient b2 (Hours_Slept): {b2:.3f}")

# --- Metrics ---
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("\nRegression Metrics:")
print(f"Mean Absolute Error (MAE): {mae:.3f}")
print(f"Mean Squared Error (MSE): {mse:.3f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.3f}")
print(f"R^2 Score: {r2:.3f}")
```

Intercept (b_0): 25.007
Coefficient b_1 (Hours_Studied): 5.884
Coefficient b_2 (Hours_Slept): 0.718

Regression Metrics:

Mean Absolute Error (MAE): 2.905
Mean Squared Error (MSE): 8.468
Root Mean Squared Error (RMSE): 2.910
 R^2 Score: 0.985