from google.colab import drive

drive.mount('/content/drive')

import pandas as pd

# Load CSV file from Google Drive

file\_path = '/content/drive/My Drive/colab1.csv'  # Change path if needed

df = pd.read\_csv(file\_path)

# Display the first few rows

print(df)

# Step 1: Mount Google Drive

from google.colab import drive

drive.mount('/content/drive')

# Step 2: Import required libraries

import pandas as pd

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

# Step 3: Load the dataset from Google Drive

file\_path = '/content/drive/My Drive/colab1.csv'  # Update path if needed

df = pd.read\_csv(file\_path)

# Step 4: Display all rows of the dataset

pd.set\_option('display.max\_rows', None)

print("Dataset:")

print(df)

# Step 5: View and confirm column names

print("\nColumn names:", df.columns.tolist())

# Step 6: Prepare data for linear regression using correct column names

X = df[['Hours\_Studied']]           # Independent variable

y = df['Performance\_Score']         # Dependent variable

# Step 7: Create and train the linear regression model

model = LinearRegression()

model.fit(X, y)

# Step 8: Print model parameters

print("\nLinear Regression Model:")

print("Intercept:", model.intercept\_)

print("Slope:", model.coef\_[0])

# Step 9: Plot the data and the regression line

plt.figure(figsize=(8, 5))

plt.scatter(X, y, color='blue', label='Data Points')

plt.plot(X, model.predict(X), color='red', label='Regression Line')

plt.xlabel('Hours Studied')

plt.ylabel('Performance Score')

plt.title('Linear Regression: Hours Studied vs. Performance Score')

plt.legend()

plt.grid(True)

plt.show()

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

Dataset:

Hours\_Studied Performance\_Score

0 0 85

1 1 70

2 2 60

3 3 55

4 4 50

5 5 55

6 6 60

7 7 70

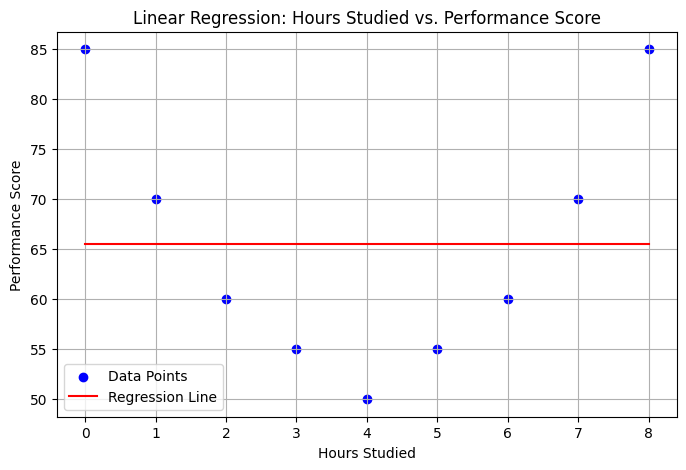
8 8 85

Column names: ['Hours\_Studied', 'Performance\_Score']

Linear Regression Model:

Intercept: 65.55555555555556

Slope: 4.586533637279209e-16



# Step 1: Mount Google Drive

from google.colab import drive

drive.mount('/content/drive')

# Step 2: Import required libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

# Step 3: Load the dataset

file\_path = '/content/drive/My Drive/colab1.csv'

df = pd.read\_csv(file\_path)

# Step 4: Display all rows of the dataset

pd.set\_option('display.max\_rows', None)

print("Dataset:")

print(df)

# Step 5: Prepare data

X = df[['Hours\_Studied']]              # Feature (must be 2D)

y = df['Performance\_Score']            # Target

# Step 6: Transform features to polynomial features (degree 2)

poly = PolynomialFeatures(degree=2)

X\_poly = poly.fit\_transform(X)

# Step 7: Train the polynomial regression model

model = LinearRegression()

model.fit(X\_poly, y)

# Step 8: Predict using the model

y\_pred = model.predict(X\_poly)

# Step 9: Print the model coefficients

print("\nPolynomial Regression Model (Degree 2):")

print("Intercept:", model.intercept\_)

print("Coefficients:", model.coef\_)

# Step 10: Plotting the results

plt.figure(figsize=(8, 5))

plt.scatter(X, y, color='blue', label='Data Points')

# Sort for a smooth curve

X\_sorted = np.sort(X.values, axis=0)

X\_sorted\_poly = poly.transform(X\_sorted)

y\_sorted\_pred = model.predict(X\_sorted\_poly)

plt.plot(X\_sorted, y\_sorted\_pred, color='red', label='Polynomial Curve')

plt.xlabel('Hours Studied')

plt.ylabel('Performance Score')

plt.title('Polynomial Regression (Degree 2)')

plt.legend()

plt.grid(True)

plt.show()

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

Dataset:

Hours\_Studied Performance\_Score

0 0 85

1 1 70

2 2 60

3 3 55

4 4 50

5 5 55

6 6 60

7 7 70

8 8 85

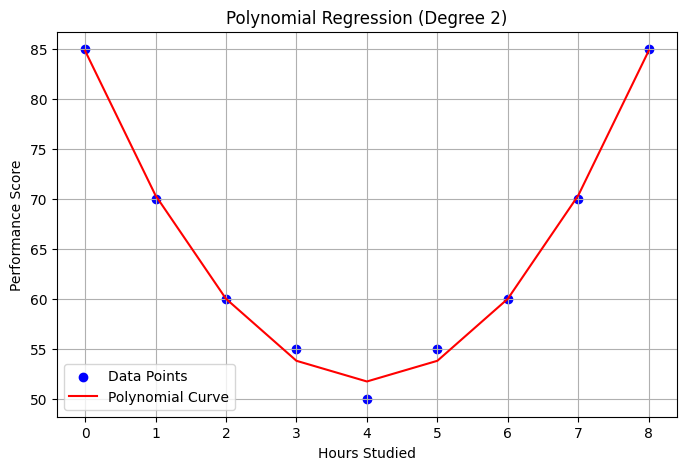
Polynomial Regression Model (Degree 2):

Intercept: 84.84848484848476

Coefficients: [ 0. -16.53679654 2.06709957]

/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but PolynomialFeatures was fitted with feature names

warnings.warn(



from sklearn.datasets import load\_iris

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

import pandas as pd

iris = load\_iris()

X = iris.data[:, :2]

y = (iris.target != 0).astype(int)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LogisticRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print(f"Accuracy: ",accuracy\_score(y\_test, y\_pred))

print(f"",classification\_report(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(iris)

Accuracy: 1.0

precision recall f1-score support

0 1.00 1.00 1.00 10

1 1.00 1.00 1.00 20

accuracy 1.00 30

macro avg 1.00 1.00 1.00 30

weighted avg 1.00 1.00 1.00 30

[[10 0]

[ 0 20]]