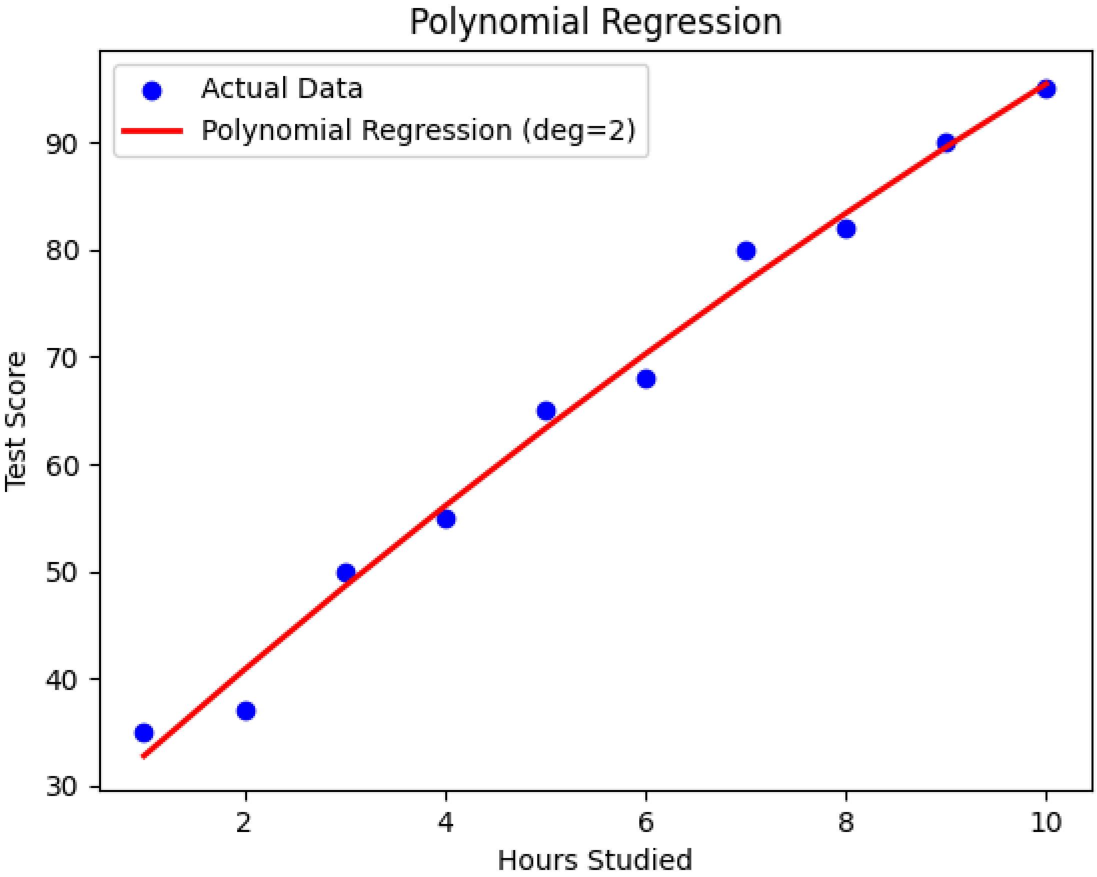
Polynomial Regression

In [1]:

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

# Sample dataset: Hours Studied vs Test Score (non-linear pattern)
X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).reshape(-1, 1)
y = np.array([35, 37, 50, 55, 65, 68, 80, 82, 90, 95])

# --- Step 1: Transform features to polynomial features ---
degree = 2  # quadratic
poly = PolynomialFeatures(degree=degree)
X_poly = poly.fit_transform(X)  # adds X^2, X^1, and bias term

# --- Step 2: Fit Linear Regression on polynomial features ---
model = LinearRegression()
model.fit(X_poly, y)

# --- Step 3: Predict ---
y_pred = model.predict(X_poly)

# --- Step 4: Evaluation ---
mse = mean_squared_error(y, y_pred)
r2 = r2_score(y, y_pred)
print(f"Mean Squared Error: {mse:.3f}")
print(f"R^2 Score: {r2:.3f}")
print(f"Intercept: {model.intercept_:.3f}")
print(f"Coefficients: {model.coef_}")

# --- Step 5: Plot ---
plt.scatter(X, y, color='blue', label='Actual Data')
plt.plot(X, y_pred, color='red', linewidth=2, label=f'Polynomial Regression (c
plt.xlabel("Hours Studied")
plt.ylabel("Test Score")
plt.title("Polynomial Regression")
plt.legend()
plt.show()
```

```
Mean Squared Error: 4.221
R^2 Score: 0.990
Intercept: 24.483
Coefficients: [ 0.          8.4219697  -0.13257576]
```


Polynomial Regression
● Actual Data
Polynomial Regression (deg=2)
Test Score
90
80
70
60
50
40
30
2
4
6
8
10
Hours Studied