LAB 05

Name: Nguyen Dinh Khanh Ngan – ITCSIU22236

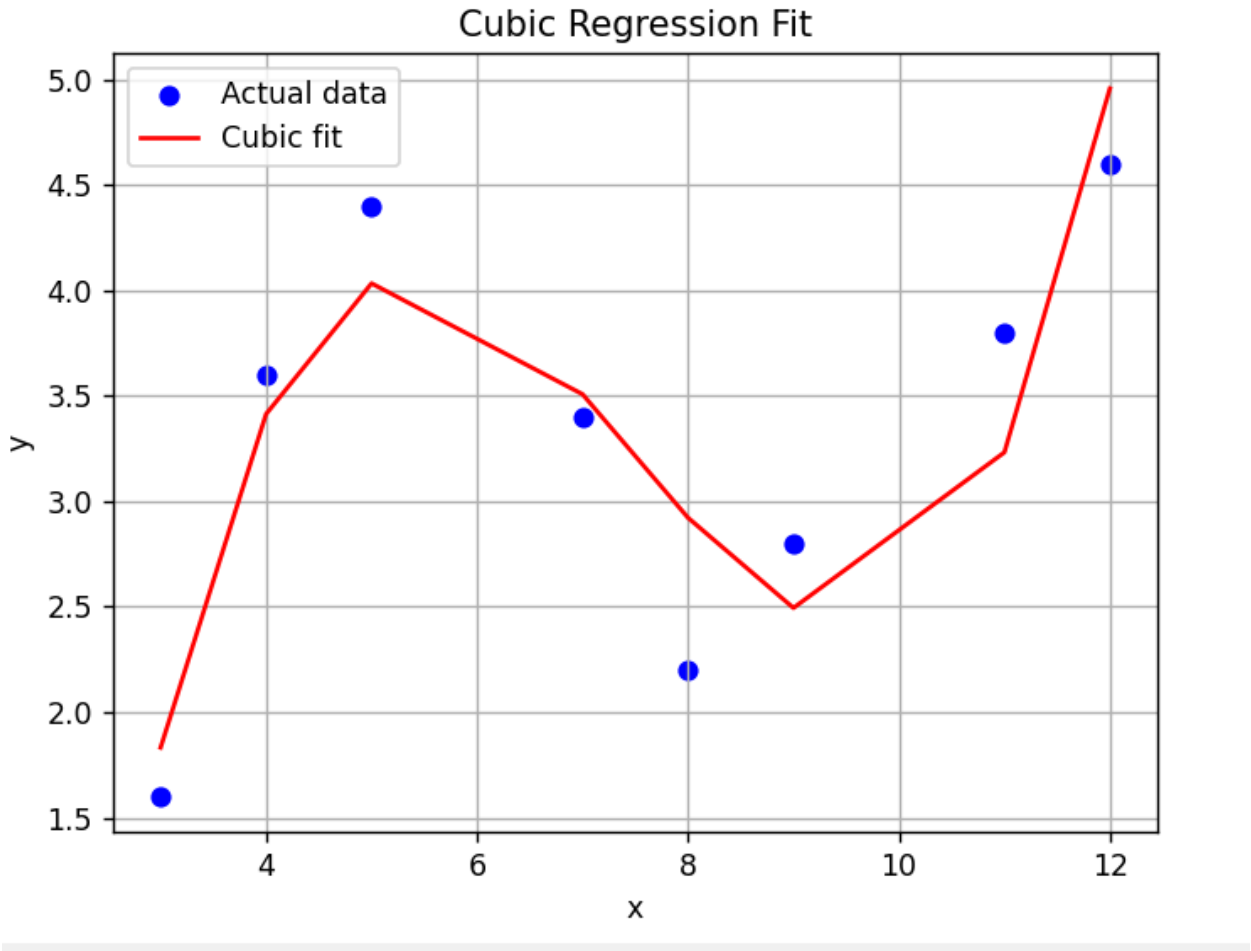
Exercise 1:

```python
import numpy as np
import matplotlib.pyplot as plt

x = np.array([3, 4, 5, 7, 8, 9, 11, 12])
y = np.array([1.6, 3.6, 4.4, 3.4, 2.2, 2.8, 3.8, 4.6])

coefficients = np.polyfit(x, y, 3)

poly = np.poly1d(coefficients)

y_pred = poly(x)

mean_y = np.mean(y)
ss_total = np.sum((y - mean_y)**2)
ss_residual = np.sum((y - y_pred)**2)
r2 = 1 - (ss_residual / ss_total)

n = len(y)
syx = np.sqrt(ss_residual / (n - 4))

print("Cubic equation coefficients (from highest to lowest power):")
print(coefficients)
print("\nCubic equation:")
print(poly)
print(f"\nR-squared (r2): {r2:.4f}")
print(f"Standard error of estimate (sy/x): {syx:.4f}")

plt.scatter(x, y, color='blue', label='Actual data')
plt.plot(x, y_pred, color='red', label='Cubic fit')
plt.title('Cubic Regression Fit')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid(True)
plt.show()
```

Result:

```
Cubic equation coefficients (from highest to lowest power):
[  0.04667602  -1.04120692   7.14381722 -11.48870718]

Cubic equation:
         3         2
0.04668 x - 1.041 x + 7.144 x - 11.49

R-squared (r2): 0.8290
Standard error of estimate (sy/x): 0.5700
```

Plot:



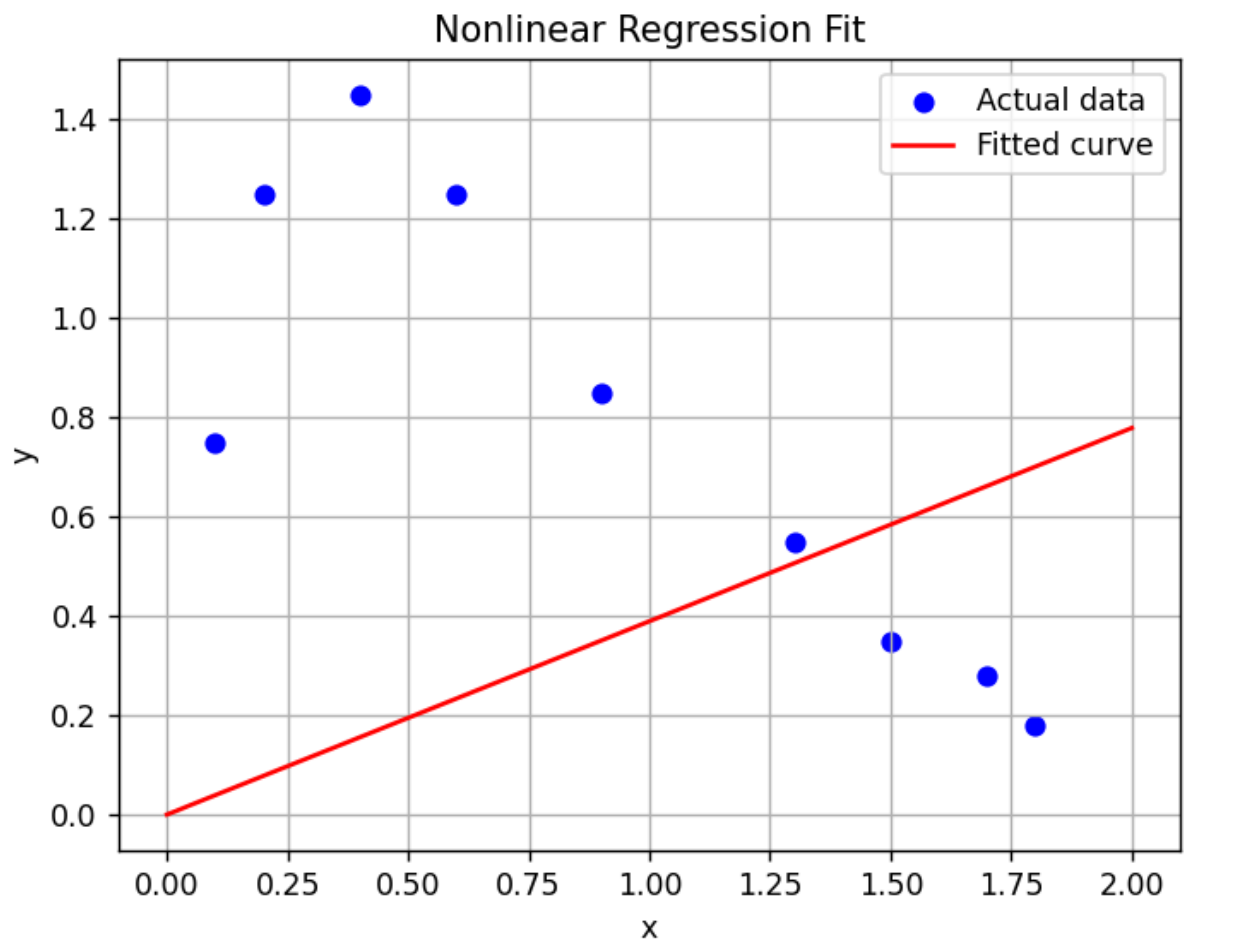
Exercise 2:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

def model_func(x, a4, beta4):
    return a4 * x * (1 - np.exp(-beta4 * x))

x_data = np.array([0.1, 0.2, 0.4, 0.6, 0.9, 1.3, 1.5, 1.7, 1.8])
y_data = np.array([0.75, 1.25, 1.45, 1.25, 0.85, 0.55, 0.35, 0.28, 0.18])

params, covariance = curve_fit(model_func, x_data, y_data, p0=[2, 2])
a4_fit, beta4_fit = params

y_pred = model_func(x_data, a4_fit, beta4_fit)

residuals = y_data - y_pred
ss_res = np.sum(residuals**2)
ss_tot = np.sum((y_data - np.mean(y_data))**2)
r_squared = 1 - (ss_res / ss_tot)

print(f"Fitted parameters:")
print(f"a4 = {a4_fit:.4f}")
print(f"beta4 = {beta4_fit:.4f}")
print(f"R-squared = {r_squared:.4f}")

plt.scatter(x_data, y_data, color='blue', label='Actual data')
x_fit = np.linspace(0, 2, 100)
y_fit = model_func(x_fit, a4_fit, beta4_fit)
plt.plot(x_fit, y_fit, 'r-', label='Fitted curve')
plt.title('Nonlinear Regression Fit')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid(True)
plt.show()
```

Result:

```
Fitted parameters:
a4 = 0.3895
beta4 = 164.2917
R-squared = -2.0471
```

Plot:


Nonlinear Regression Fit
Actual data
Fitted curve
y
x
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00