ASSIGNMENT 3 : LINEAR REGRESSION

Link for dataset: Assignment 2 dataset
Name:____________________

Polynomial Regression via Optimization

In this assignment, you will extend the concepts of linear regression to **Polynomial Regression**. Instead of fitting a straight line, you will fit curves to the dataset A and B from Assignment 2 by creating higher-order features (x^2, x^3) and using scipy.optimize.minimize to find the optimal coefficients.

Objectives

1. **Feature Engineering**: Transform the input x into a feature matrix containing x^2 and
2. x^3.
3. **Loss Function Design**: Write a generalized Mean Squared Error (MSE) function for polynomials.
4. **Optimization**: Use scipy to find the best-fit parameters for degrees 2 and 3.
5. **Visualization**: Compare the fits of the two models against the original data.

Part 1: Feature Engineering

A polynomial of degree n is represented as:

y = a1 + a2 * x + a3 * x^2 + ….. + an x^n

To solve this using linear optimization techniques, we treat x^2 and x^3 as independent "features."

Task: Load your dataset and create two new columns: x_sq (x^2) and x_cube (x^3).

Part 2: Polynomial Regression (Degree 2)

For degree 2, our model is: y = a x^2 + b x + c.

Task: Define a loss function that takes a parameter vector [a, b, c] and returns the MSE. Use minimize to find the values.

Part 3: Polynomial Regression (Degree 3)

For degree 3, our model is: y = a x^3 + b x^2 + c x + d.

Task: Implement the loss function for a 3rd-degree polynomial and optimize it.

Part 4: Visualization

To plot a smooth polynomial curve, it is best to generate a range of "test" x values and calculate the y for each.

Task: Plot the original scatter data, the Degree 2 fit, and the Degree 3 fit on the same graph.

Python

```python
# Generate smooth X values for plotting
x_range = np.linspace(df['x'].min(), df['x'].max(), 100)
# Calculate Y values for the curves
y_fit2 = a_opt*(x_range**2) + b_opt*x_range + c_opt
y_fit3 = params3[0]*(x_range**3) + params3[1]*(x_range**2) + params3[2]*x_range +
params3[3]

plt.figure(figsize=(10,6))
plt.scatter(df['x'], df['y'], color='gray', alpha=0.5, label='Data')
plt.plot(x_range, y_fit2, color='red', label='Degree 2 (Parabolic)')
plt.plot(x_range, y_fit3, color='green', label='Degree 3 (Cubic)', linestyle='--')
plt.legend()
plt.title("Polynomial Regression Comparison")
plt.show()
```

Submission Requirements

1. **Coefficients**: Report the optimized coefficients for both models.
2. **Comparison**: Which degree fits the parabola dataset better? Why? (Hint: Look at the final value of the loss function res.fun).
3. **The Plot**: A single plot showing the data points and both regression curves.
4. Answer the following question : did polynomial regression work as you expected?