Task: Linear Regression Using Gradient Descent

The following class is provided to you:

```
class LinearRegressionGD:
    ...
```

Task Requirements

Load & Understand the Data

Given the dataset:

```
X = [50, 60, 70, 80, 90]
y = [150, 180, 210, 240, 270]  # house price in thousands
```

1. Explain what **X** and **y** represent.
2. Convert the data to **numpy arrays**.

◆ Create and Train the Model

1. Create an instance of `LinearRegressionGD` with:
    - `learning_rate = 0.001`
    - `n_iters = 100`
2. Train the model using the `fit()` method.
3. Print the learned values of:
    - `theta_0`
    - `theta_1`

Question:
What do `theta_0` and `theta_1` represent in the regression equation?

Prediction

1. Use the trained model to predict the price of a house with size:
    - 70 m^2
2. Print the predicted value.

Question:
Is the prediction reasonable based on the dataset? Why?

Visualization

1. Use the `plot_training()` method to:
    - Visualize **SSE over iterations**
    - Plot the **regression line with data points**
2. Explain:
    - Why SSE decreases over time
    - What convergence means in Gradient Descent

Experimentation

1. Train the model with:
    - a **very large learning rate**
    - a **very small learning rate**
2. Compare:
    - Convergence speed
    - Final SSE value

Question:
What happens if the learning rate is too large?

Bonus Tasks (Optional)

- Add lasso and Ridge method
- Add a method to calculate **MSE**
- Normalize X before training and compare results
- Modify the class to support **multiple features**

Expected Learning Outcomes

By completing this task, students will:

- Understand **Gradient Descent**
- Apply **OOP in Machine Learning**
- Visualize **model convergence**
- Build ML models **without sklearn**