CAP 4630 – Artificial Intelligence

Instructor: Aakash Kumar

University of Central Florida

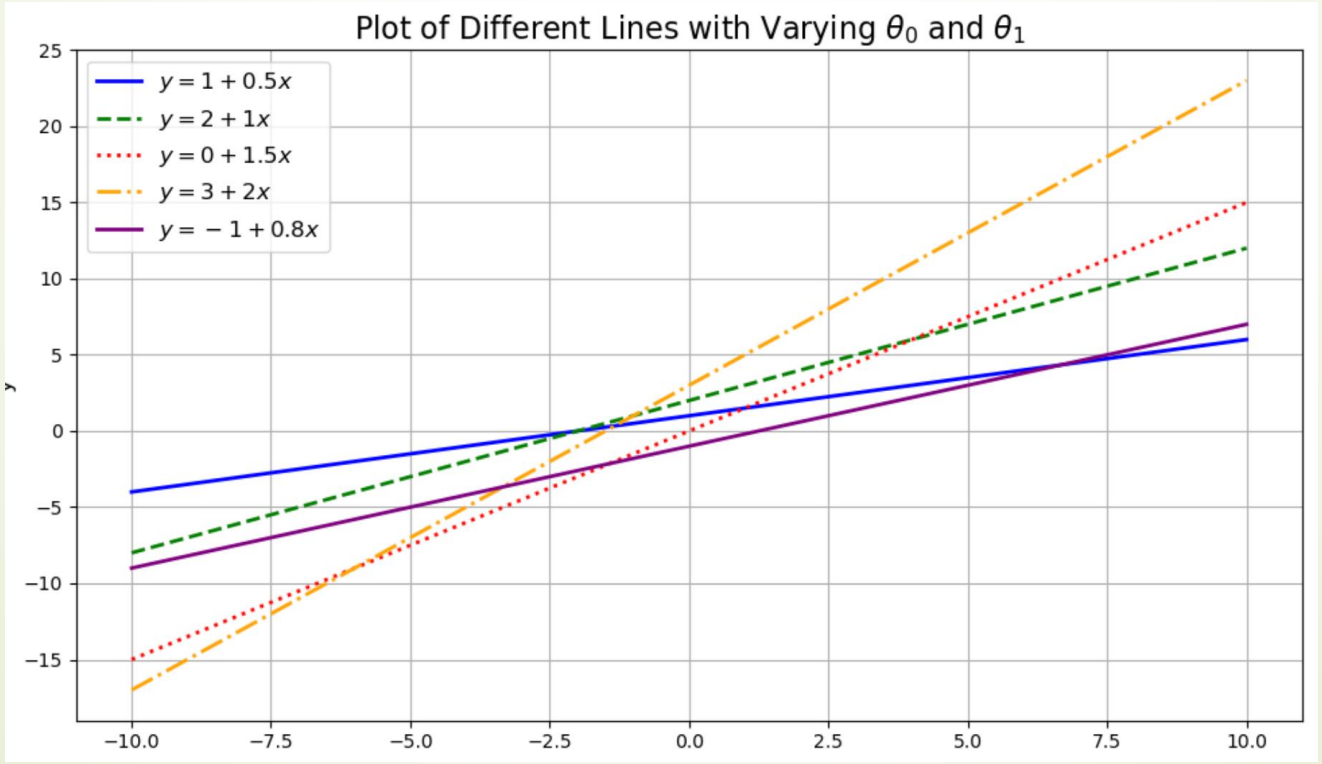
Introduction to Linear Regression

- **What is Linear Regression?**
  - **Linear Regression** is a **supervised learning algorithm** used to model the relationship between a **dependent variable (y)** and one or more **independent variables (x)**.
  - The goal is to find the best-fitting **straight line** (also called a regression line) through the data points.
- **Why Use Linear Regression?**
  - **Simple** yet **effective** for predicting outcomes.
  - Provides insights into the relationship between variables.
  - Useful in many fields like economics, biology, engineering, etc.
- **Key Equation:**

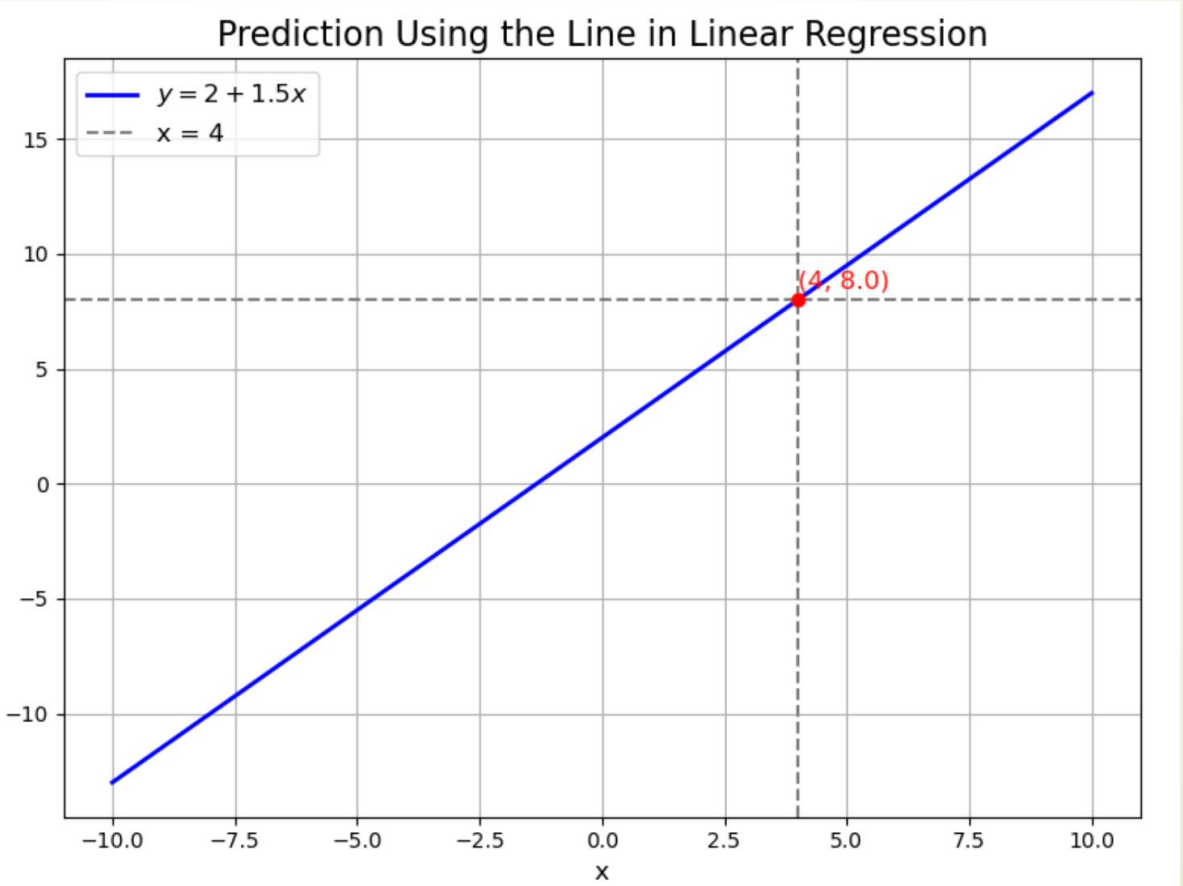
$$y = \theta_0 + \theta_1 \cdot x$$

Understanding the Line in Linear Regression

- What is a Line?
  - A line in 2D space is defined by the equation: $y = \theta_0 + \theta_1 \cdot x$



How Predictions are Made Using the Line in Linear Regression



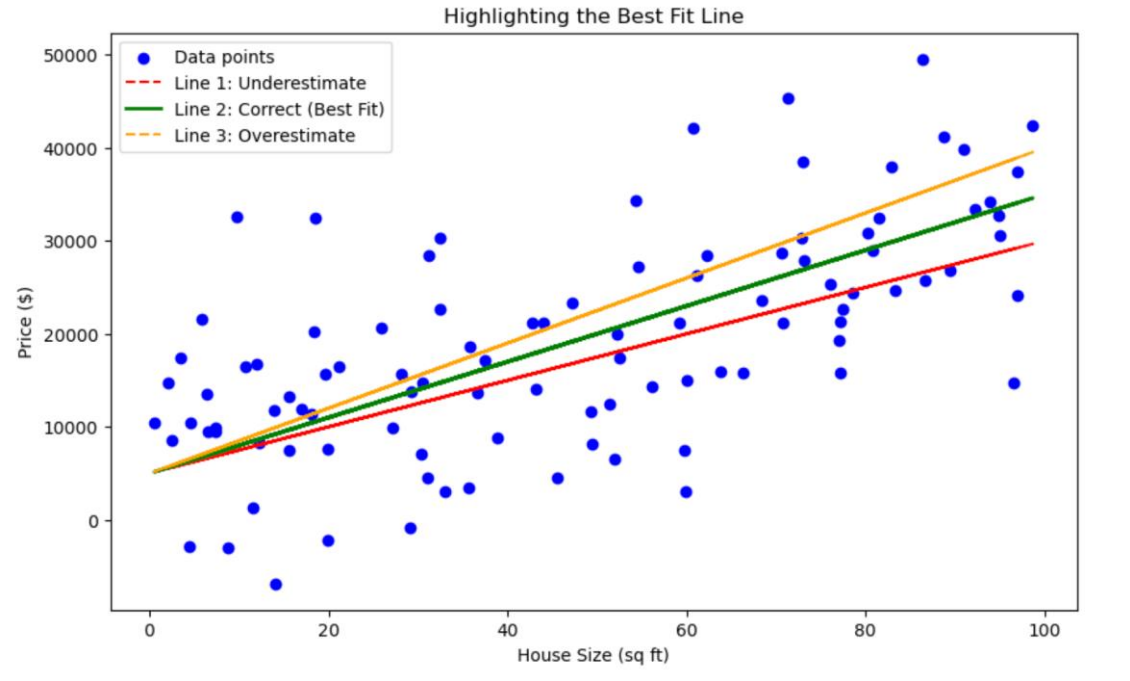
House Price Prediction - Data

	House Size (sq ft)	Price ($)
0	37.454012	17106.674248
1	95.071431	30531.355688
2	73.199394	27877.426020
3	59.865848	3084.065380
4	15.601864	7483.840335
...	...	...
95	49.379560	11655.765041
96	52.273283	19910.967787
97	42.754102	21237.750299
98	2.541913	8529.481796
99	10.789143	16508.575300

100 rows × 2 columns

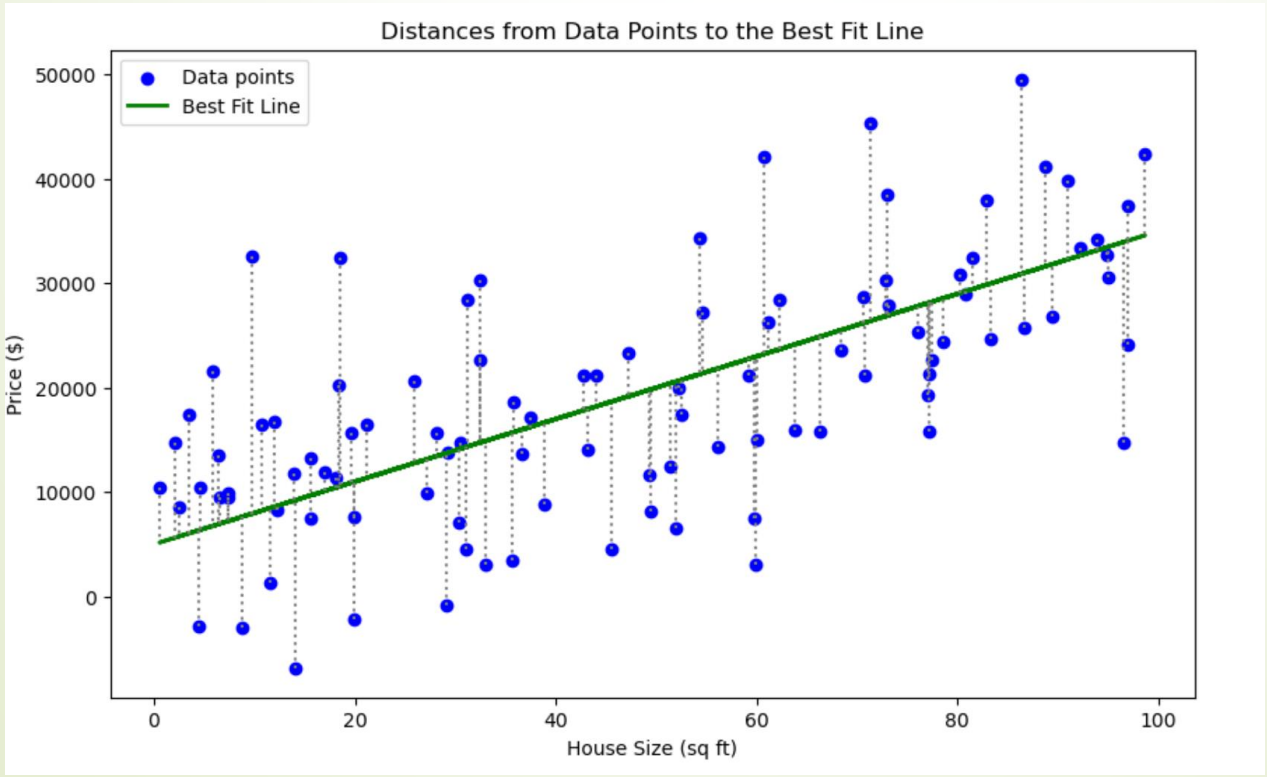
Comparing Possible Regression Lines

- **Line 1 (Red):** An underestimation of the slope, leading to a line that doesn't capture the full trend.
- **Line 2 (Green):** The correct slope (best fit), representing the ideal regression line that captures the relationship between house size and price.
- **Line 3 (Orange):** An overestimation of the slope, leading to a steeper line.



Comparing Possible Regression Lines

- **Best Fit Line (Green):** This line represents the model that best captures the relationship between house size and price.
- **Gray Vertical Lines**: These lines show the residuals, or the difference between actual prices (data points) and the predicted prices (best fit line). The goal of linear regression is to minimize the length of these residuals.



$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

What is the Hypothesis Function?

- In linear regression, the **hypothesis function** is used to predict the dependent variable Y (output) based on the independent variable X (input).
- It represents the relationship between X and Y mathematically.
- The hypothesis function is defined as: $h_\theta(x) = \theta_0 + \theta_1 \cdot x$
- **$h_\theta(x)$**: Predicted value of y for a given x.
  θ_0: The intercept (where the line crosses the y-axis).
  θ_1: The slope (rate at which y changes with respect to x).

Using the Hypothesis Function to Make Predictions

- Once we have the hypothesis function, we can make predictions for new data points.
- Given the hypothesis function:

$$h_\theta(x) = 5000 + 300 \cdot x$$

- If $x = 50$ **sq ft**, the predicted price is:

$$h_\theta(50) = 5000 + 300 \cdot 50 = 5000 + 15000 = 20000$$

- The model predicts a house of 50 sq ft will cost **$20,000**.

The Cost Function in Linear Regression

What is the Cost Function?

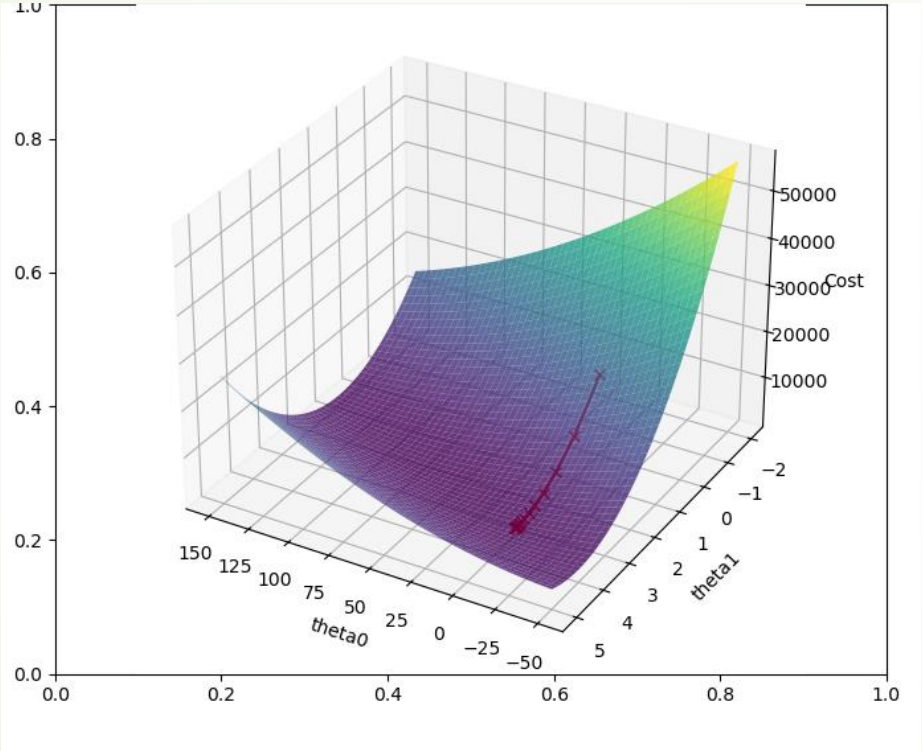
- To measure how well our hypothesis predicts the actual data, we use a **cost function**.
- The cost function quantifies the **error** between the predicted values and the actual values for all the data points.
- In linear regression, the most commonly used cost function is the **Mean Squared Error (MSE)**:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

- $J(\theta_0, \theta_1)$: The cost function that we aim to minimize.
- $h_\theta(x^{(i)})$: The predicted value of y for the i-th data point.
- $y^{(i)}$: The actual value of y for the i-th data point.
- m: The number of training examples.

- **Goal:** Minimize the cost function to find the best-fitting line. This will lead to the optimal values of θ_0 θ_1

Cost Function



Deriving the Gradient Descent Algorithm

How Do We Minimize the Cost Function?

- To find the values of that minimize the cost function, we use **Gradient Descent**.
- Gradient Descent:
  - It is an iterative optimization algorithm that adjusts to gradually reduce the cost.
  - In each iteration, the parameters are updated in the direction that reduces the cost the most.
  - The update rules for Gradient Descent are:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Where:

- θ_j: The parameter to be updated (θ_0 or θ_1).
- α: The **learning rate**, controlling the size of the update step.
- $\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$: The derivative of the cost function with respect to θ_j, representing the slope of the cost function at the current point.

Deriving the Gradient Descent Update

- Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

- Partial Derivatives:

For θ_0:

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

For θ_1:

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(\left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)} \right)$$

Step-by-Step Derivation for

Step-by-Step Derivation for θ_0:

1. **Substitute the hypothesis function** $h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$ into the cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2$$

2. **Compute the partial derivative** of $J(\theta_0, \theta_1)$ with respect to θ_0:

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_0} \left(\frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2 \right)$$

3. Apply the **chain rule**: The derivative of $\left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2$ with respect to θ_0 gives:

$$\frac{\partial}{\partial \theta_0} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2 = 2 \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)$$

4. Now, sum over all the data points:

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)$$

Step-by-Step Derivation for

Step-by-Step Derivation for θ_1:

1. **Substitute the hypothesis function** $h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$ into the cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2$$

2. **Compute the partial derivative** of $J(\theta_0, \theta_1)$ with respect to θ_1:

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_1} \left(\frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2 \right)$$

3. Apply the **chain rule**: The derivative of $\left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2$ with respect to θ_1 gives:

$$2 \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}$$

4. Sum over all the data points:

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(\left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right) x^{(i)} \right)$$

Partial Derivatives

1. **Derivative with respect to θ_0:**

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)$$

2. **Derivative with respect to θ_1:**

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(\left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right) x^{(i)} \right)$$

Gradient Descent Update Rules:

- Gradient Descent is an optimization algorithm that iteratively updates the parameters theta0 and theta1 using these derivatives to minimize the cost function.
- Update rule for theta 0

$$\theta_0 := \theta_0 - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

- Update rule for theta 1

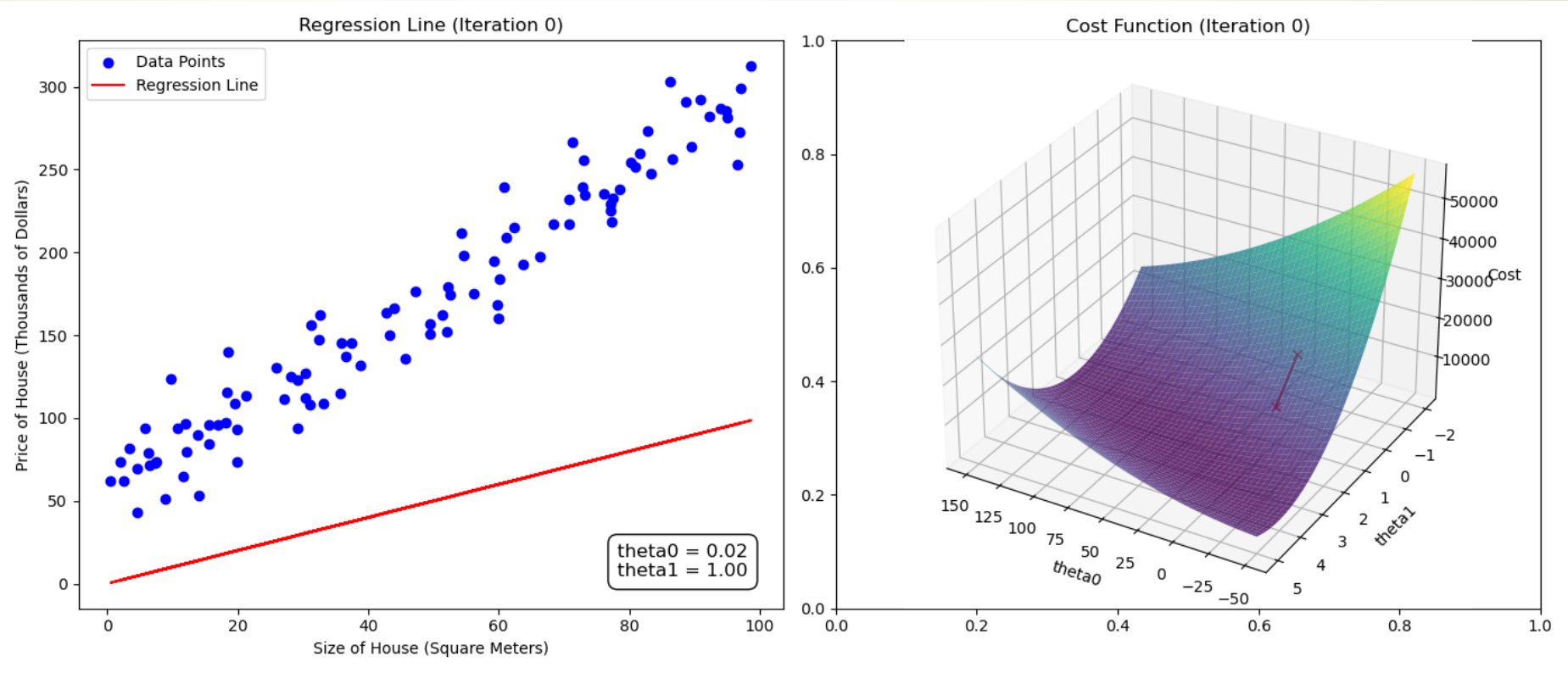
$$\theta_1 := \theta_1 - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \left(\left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)} \right)$$

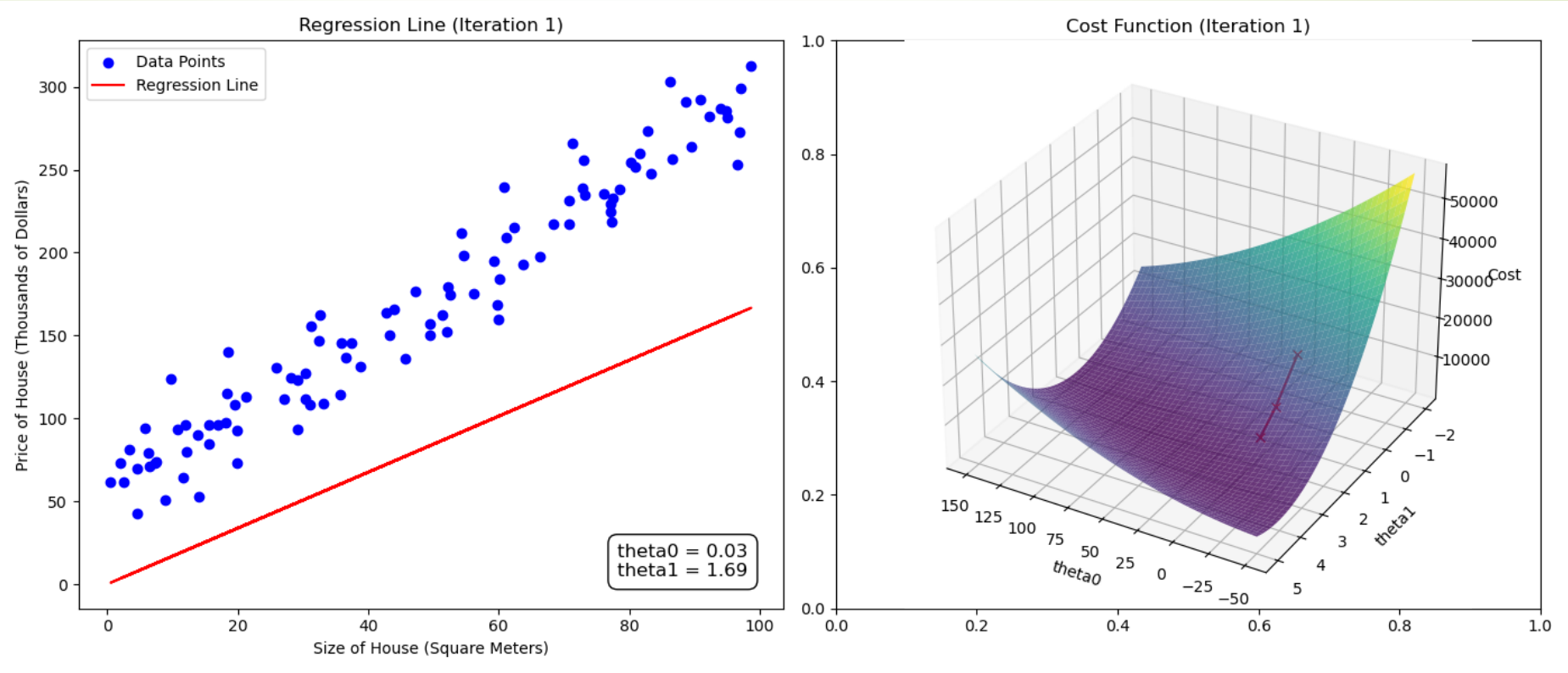
- a is the learning rate, which controls the size of the update step.

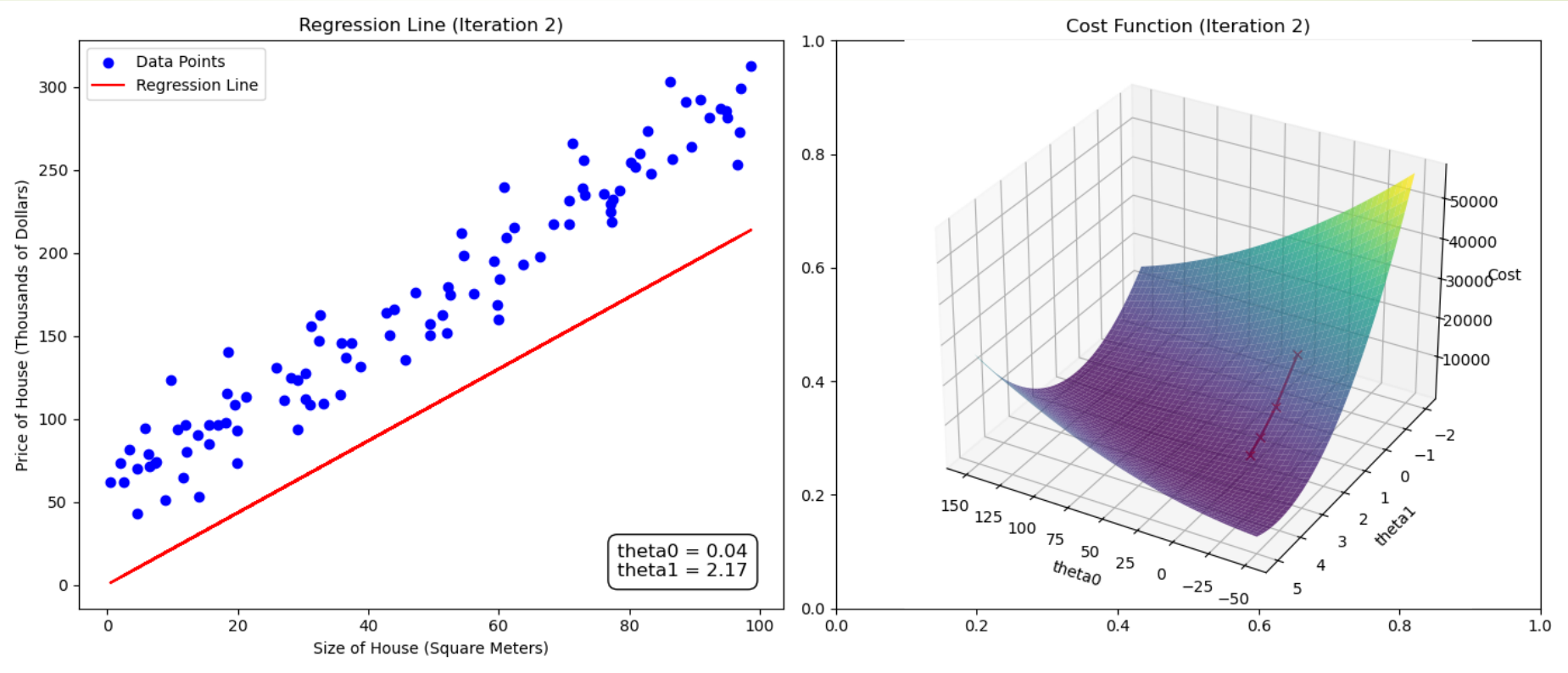
Explanation:

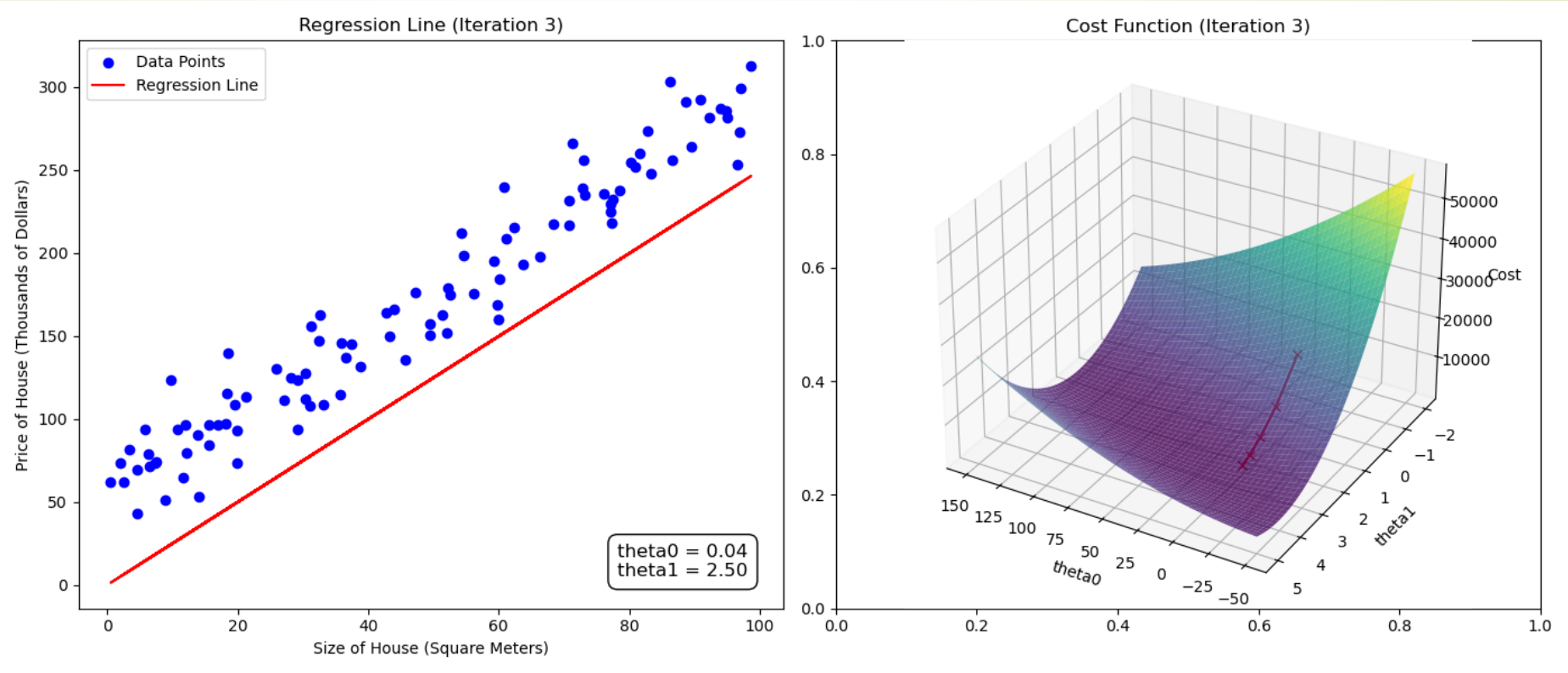
- Learning Rate:
  - The learning rate controls how large a step we take in each iteration.
  - If it is too large, the algorithm may overshoot the minimum; If it is too small, convergence will be slow.

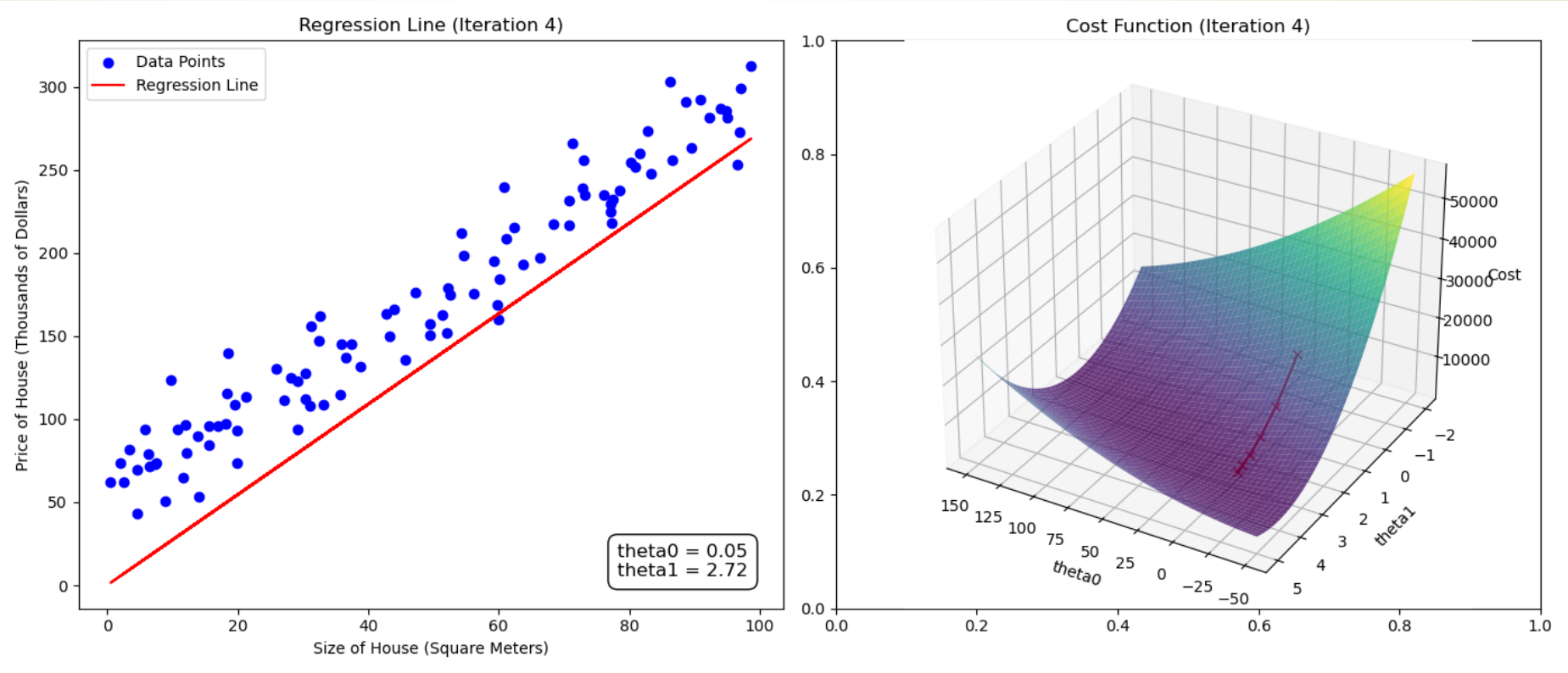
- Iterative Updates
  - In each iteration, we adjust theta 0 and theta 1 by subtracting the product of learning rate and the computed derivative.
  - These updates continue until the parameter converge to values that minimize the cost function.

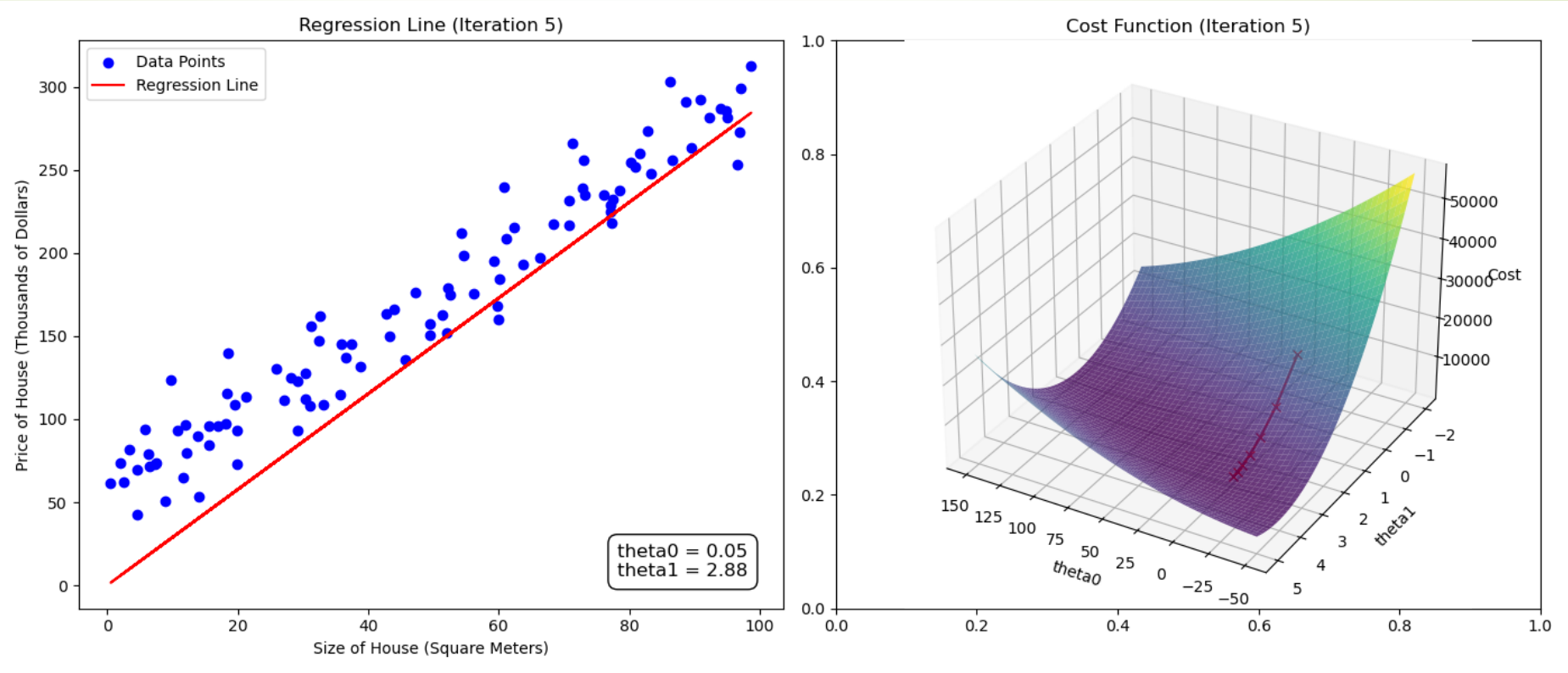

Regression Line (Iteration 0)
Data Points
Regression Line
theta0 = 0.02
theta1 = 1.00
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 0)
Cost
theta0
theta1

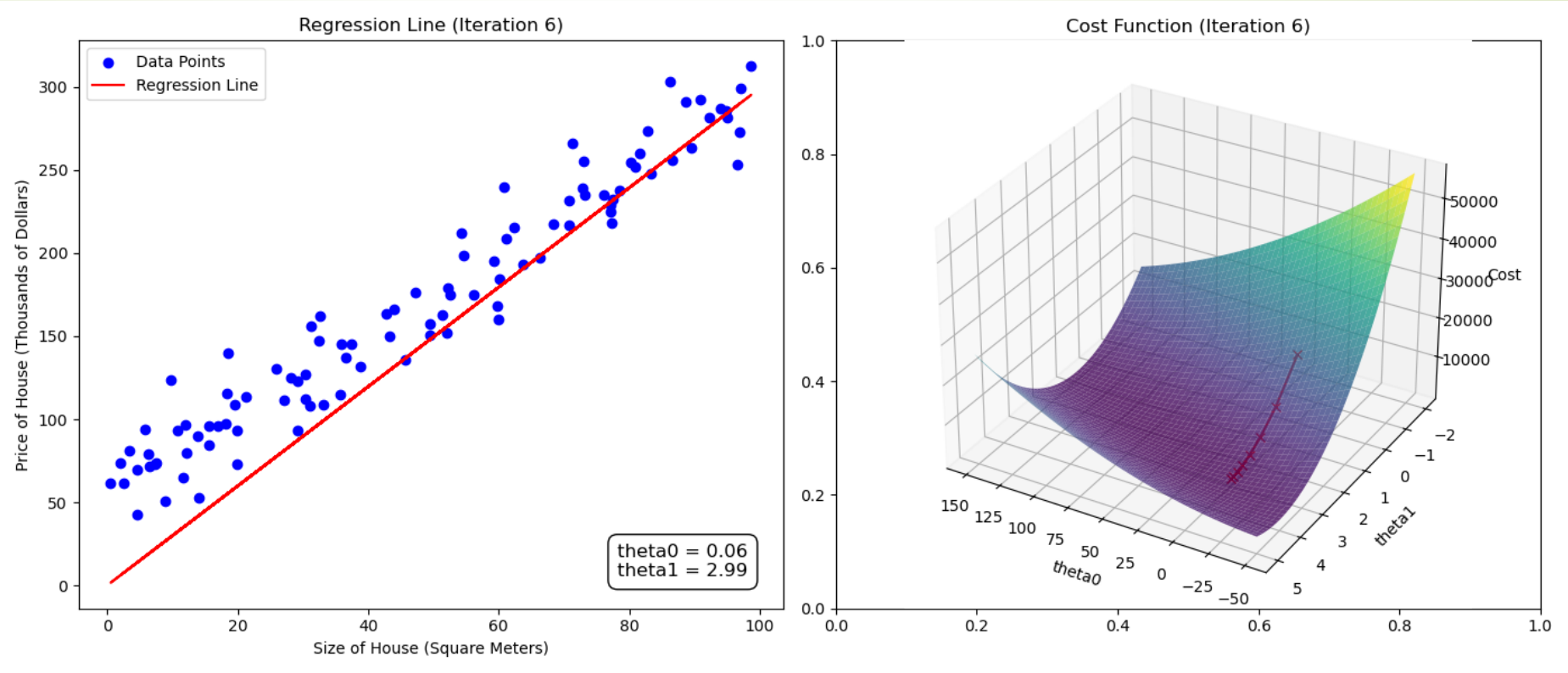

Regression Line (Iteration 1)
Data Points
Regression Line
theta0 = 0.03
theta1 = 1.69
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 1)
Cost
theta0
theta1

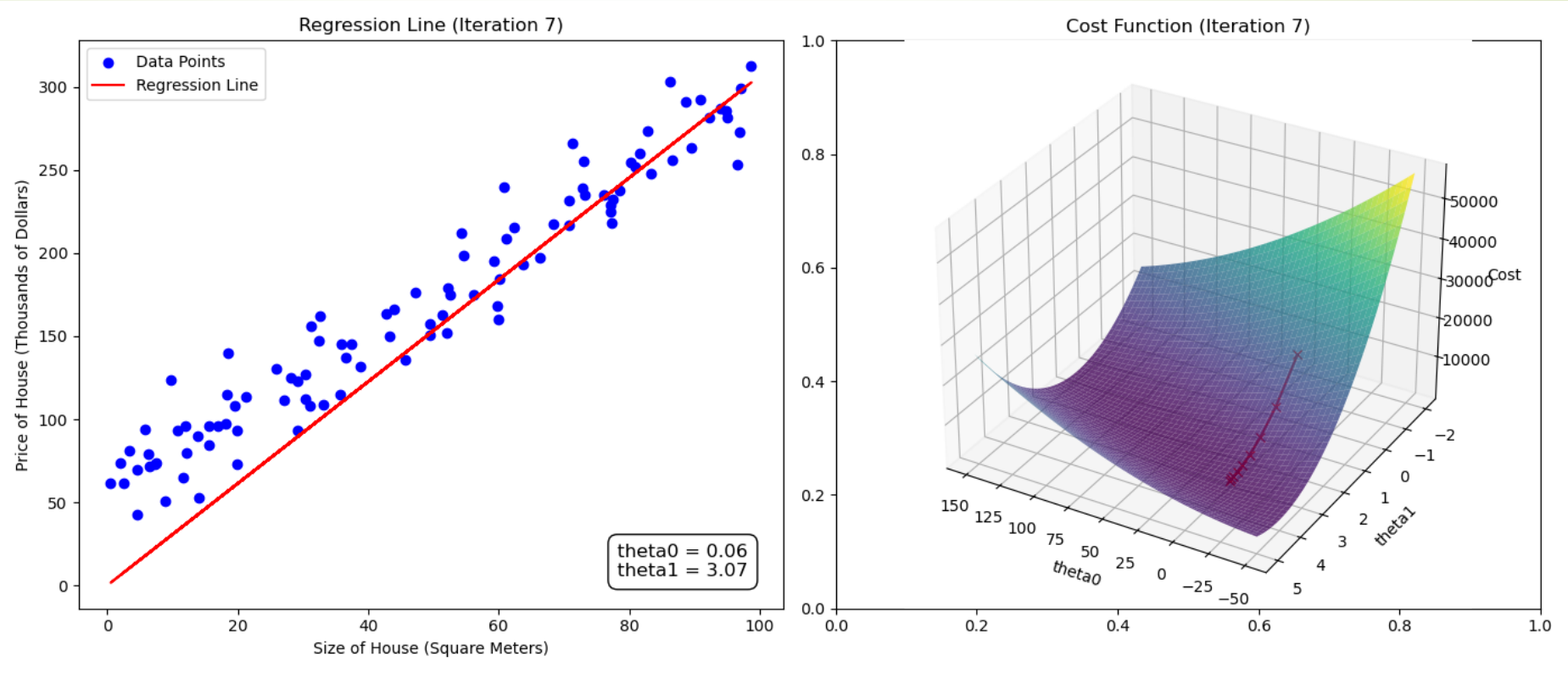

Regression Line (Iteration 2)
Data Points
Regression Line
theta0 = 0.04
theta1 = 2.17
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 2)
Cost
theta0
theta1

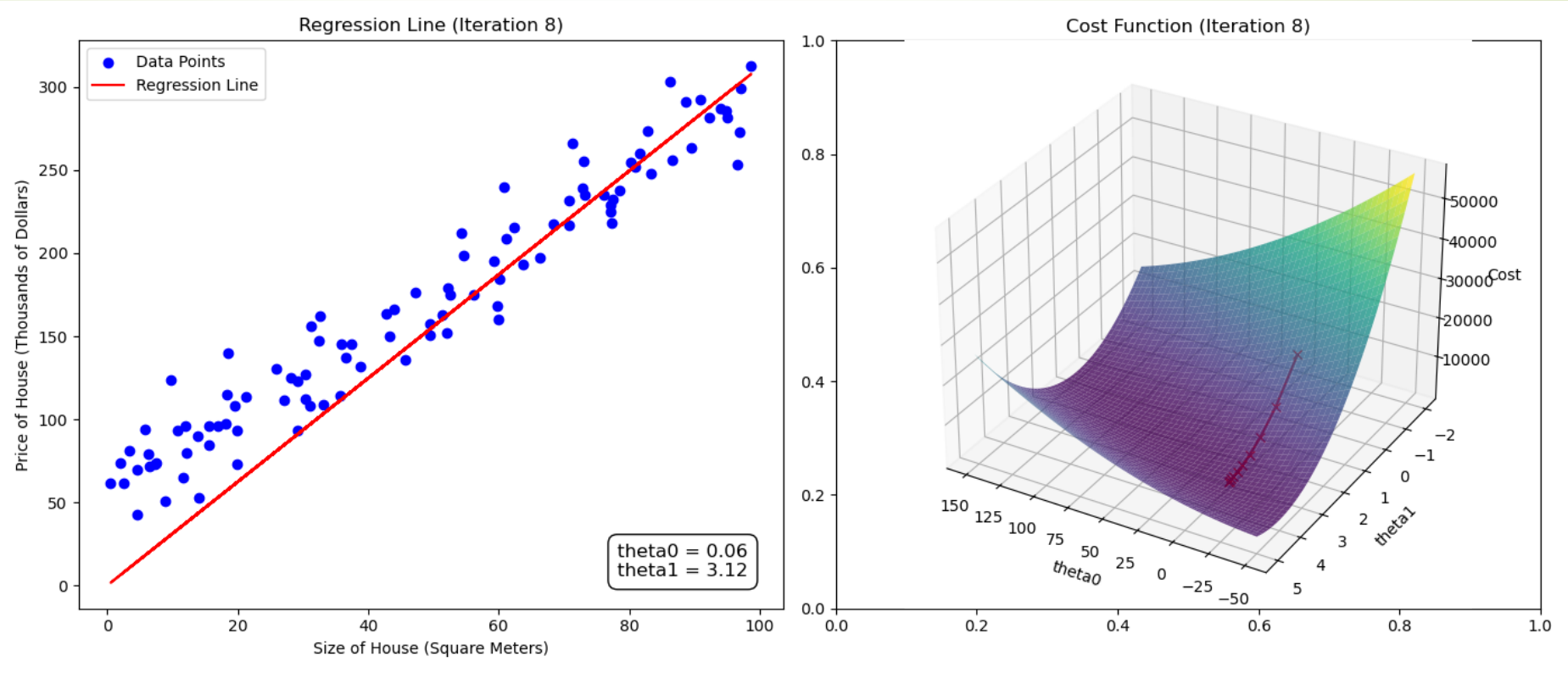

Regression Line (Iteration 3)
Data Points
Regression Line
theta0 = 0.04
theta1 = 2.50
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 3)
Cost
theta0
theta1

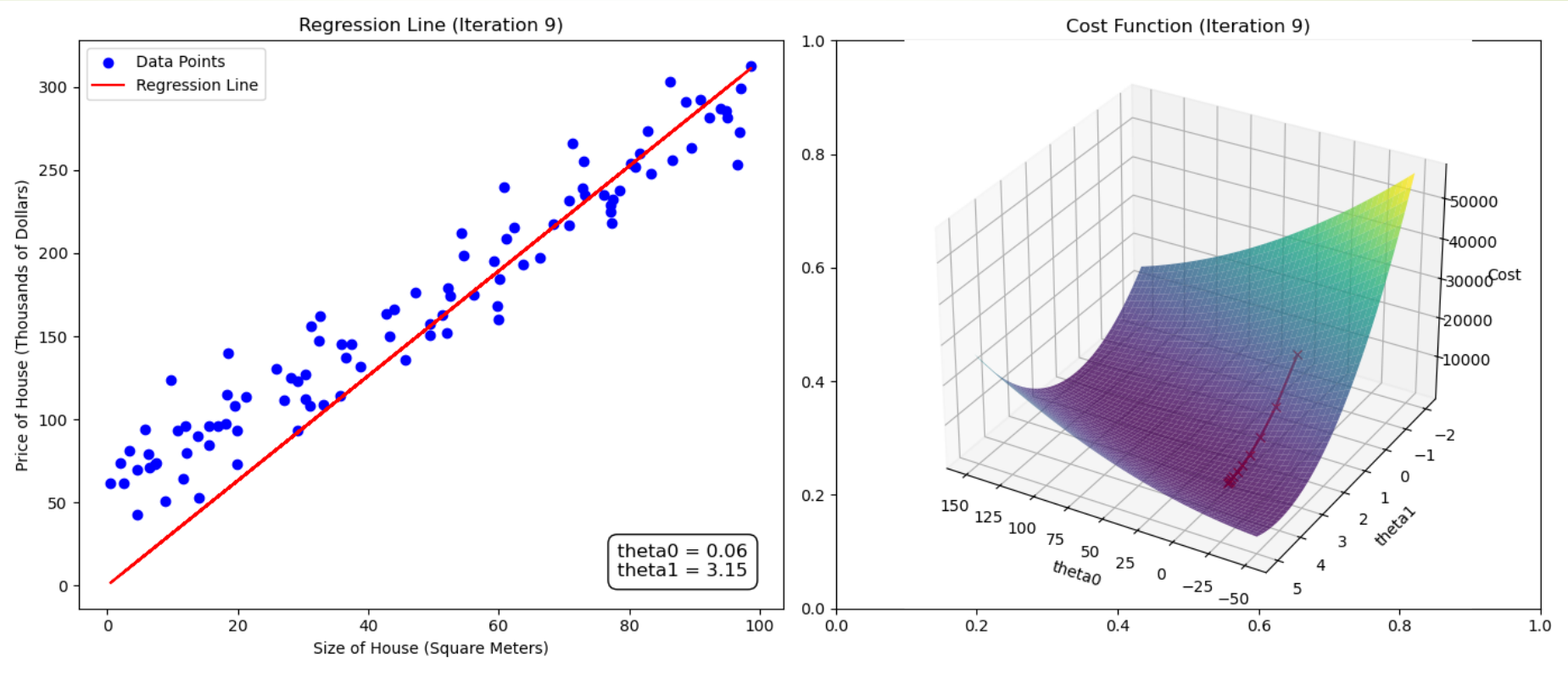

Regression Line (Iteration 4)
Data Points
Regression Line
theta0 = 0.05
theta1 = 2.72
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 4)
Cost
theta0
theta1

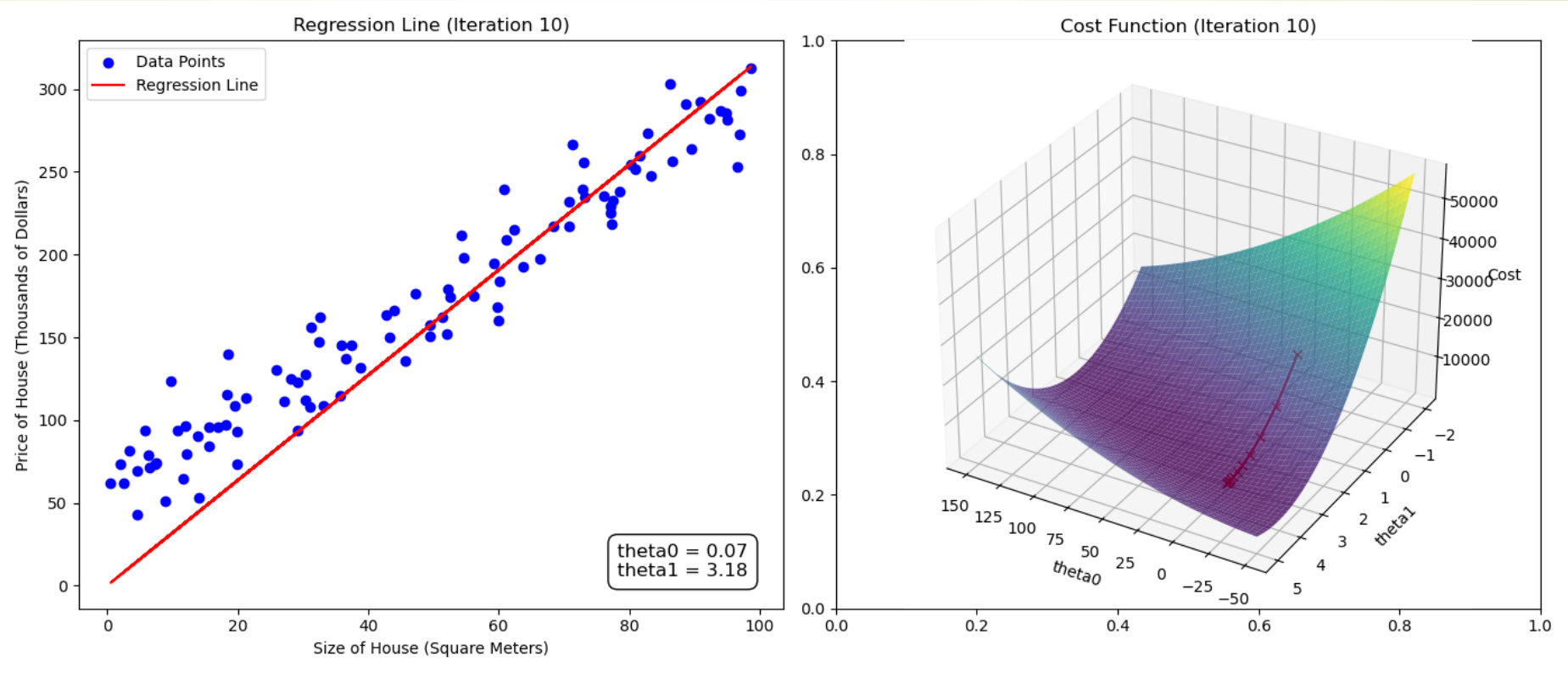

Regression Line (Iteration 5)
Data Points
Regression Line
theta0 = 0.05
theta1 = 2.88
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 5)
Cost
theta0
theta1

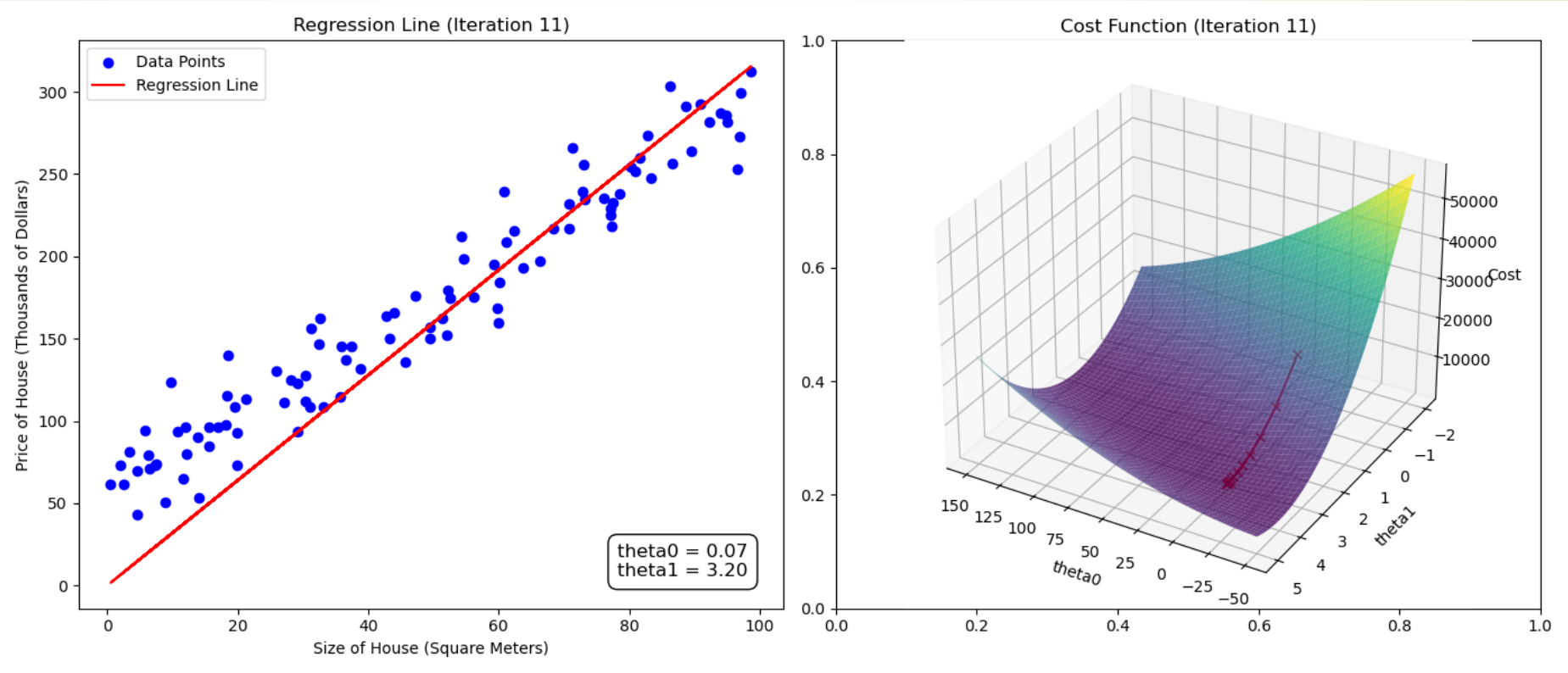

Regression Line (Iteration 6)
Data Points
Regression Line
theta0 = 0.06
theta1 = 2.99
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 6)
Cost
theta0
theta1

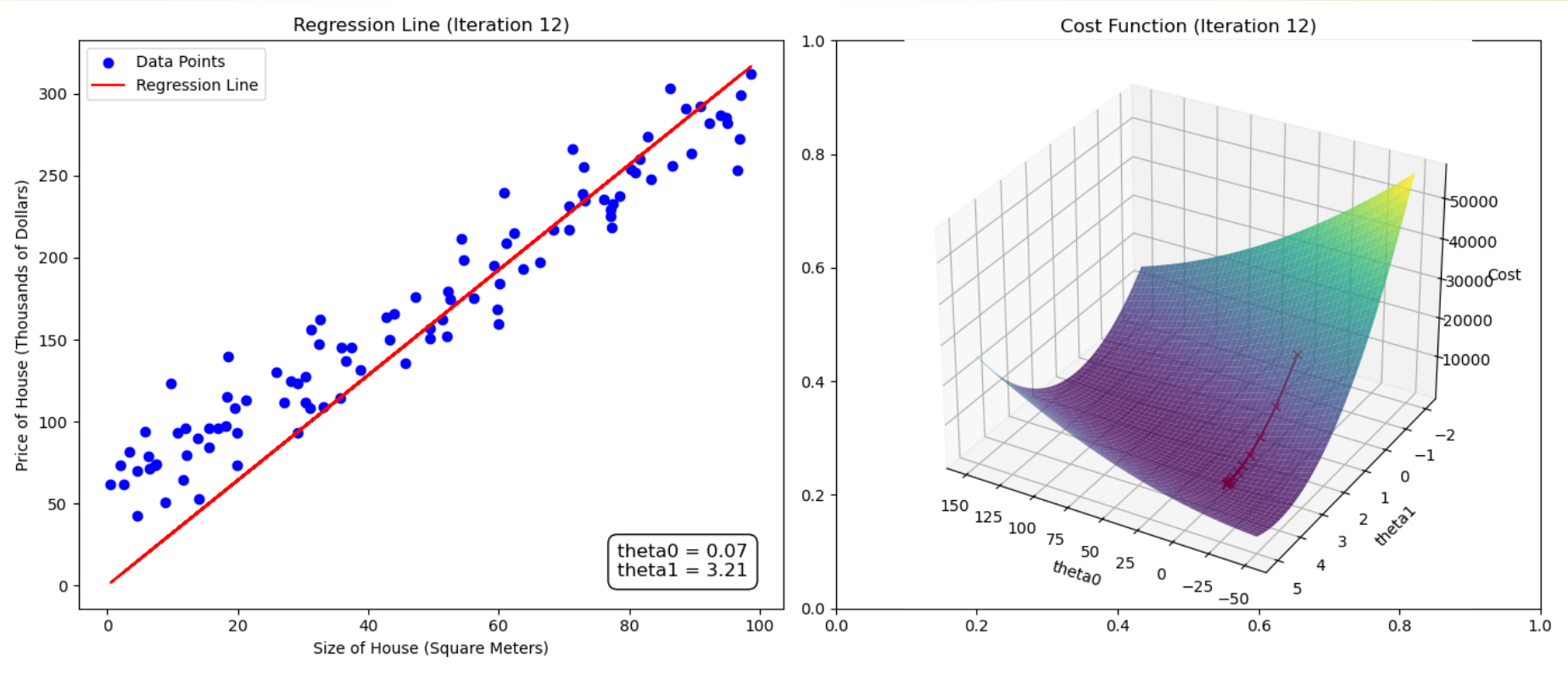

Regression Line (Iteration 7)
Data Points
Regression Line
theta0 = 0.06
theta1 = 3.07
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 7)
Cost
theta0
theta1

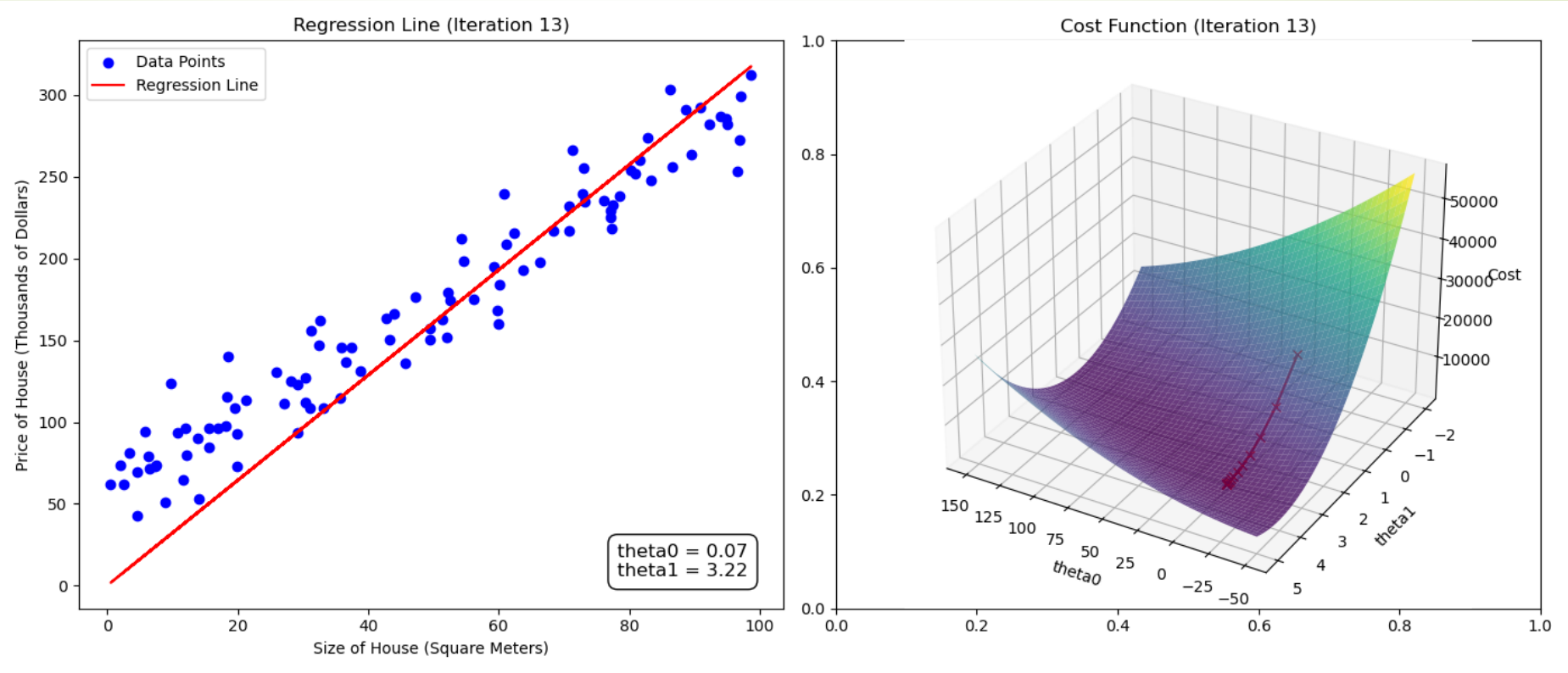

Regression Line (Iteration 8)
Data Points
Regression Line
theta0 = 0.06
theta1 = 3.12
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 8)
Cost
theta0
theta1

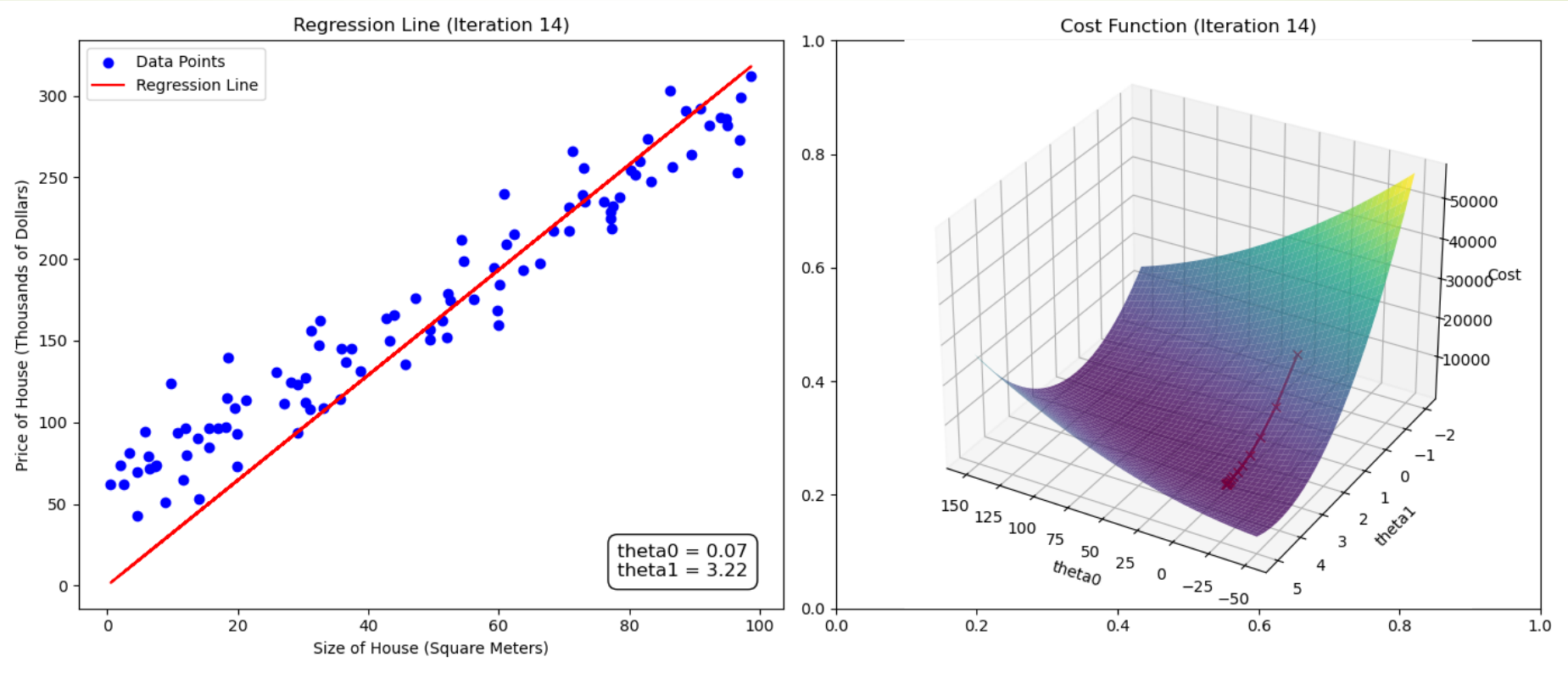

Regression Line (Iteration 9)
Data Points
Regression Line
theta0 = 0.06
theta1 = 3.15
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 9)
Cost
theta0
theta1

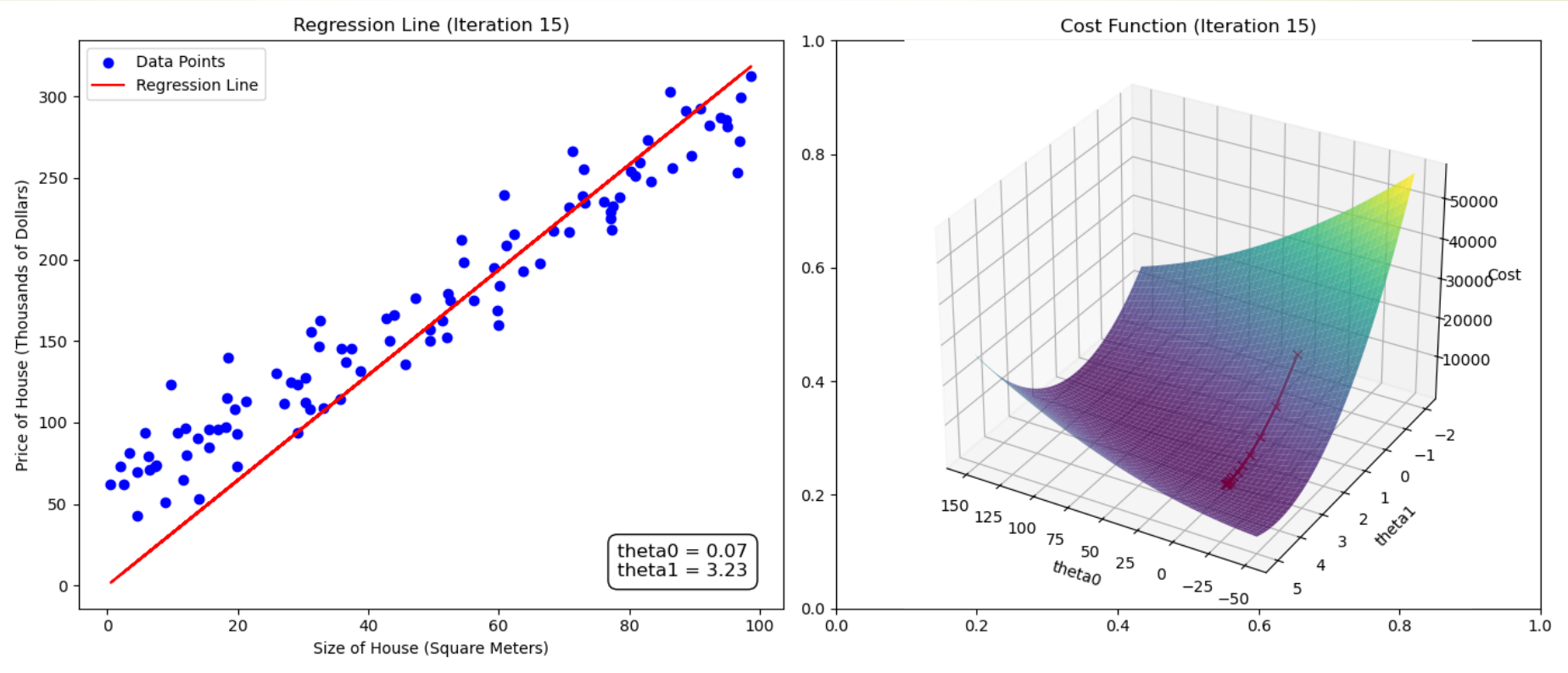

Regression Line (Iteration 10)
Data Points
Regression Line
theta0 = 0.07
theta1 = 3.18
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 10)
Cost
theta0
theta1

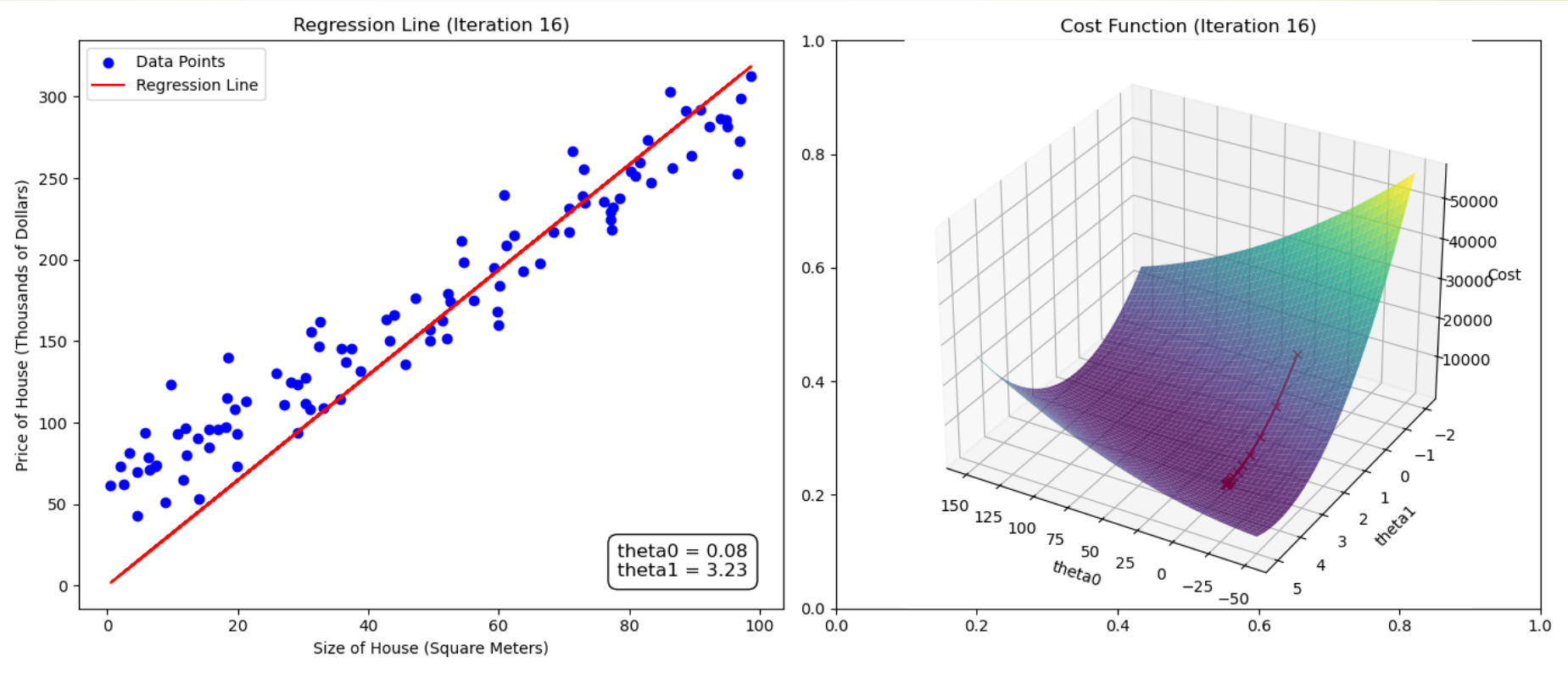

Regression Line (Iteration 11)
Data Points
Regression Line
theta0 = 0.07
theta1 = 3.20
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 11)
Cost
theta0
theta1

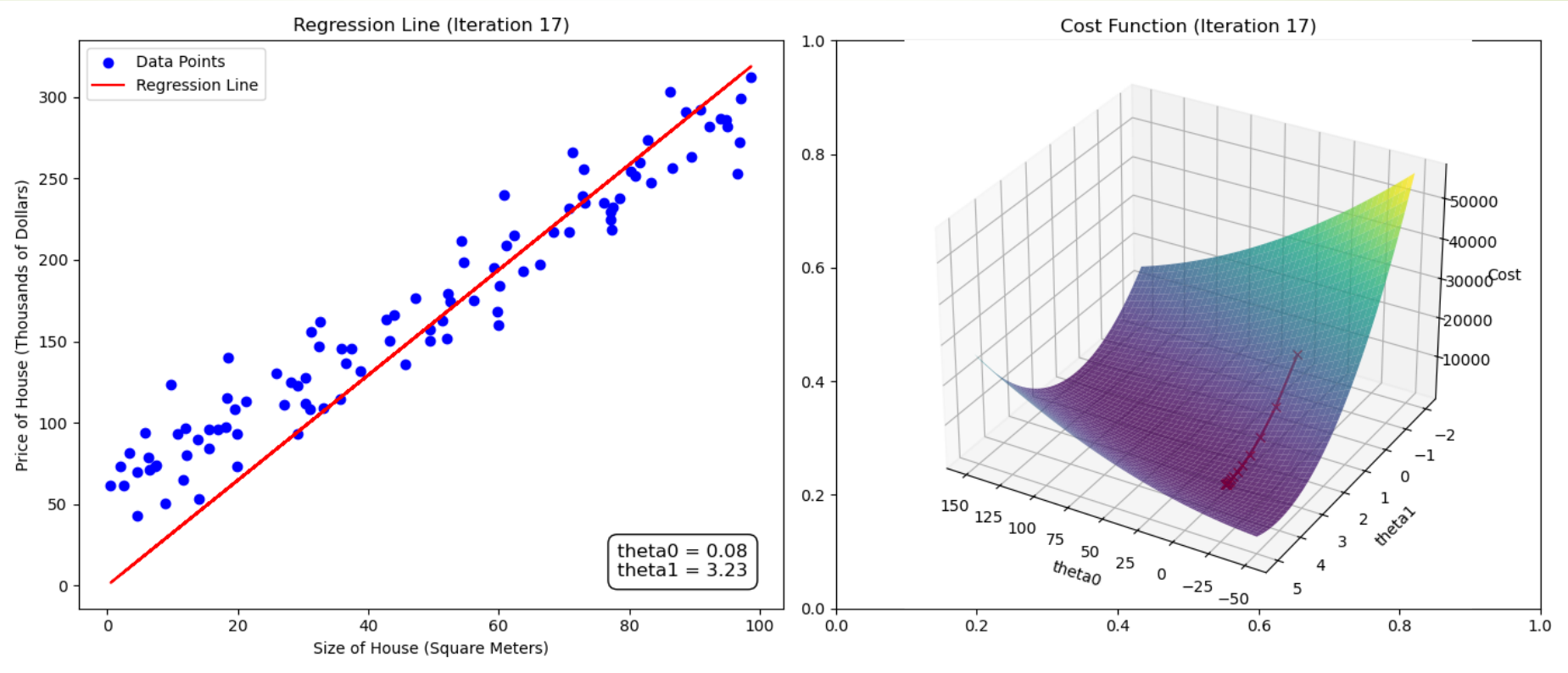

Regression Line (Iteration 12)
Data Points
Regression Line
theta0 = 0.07
theta1 = 3.21
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 12)
Cost
theta0
theta1

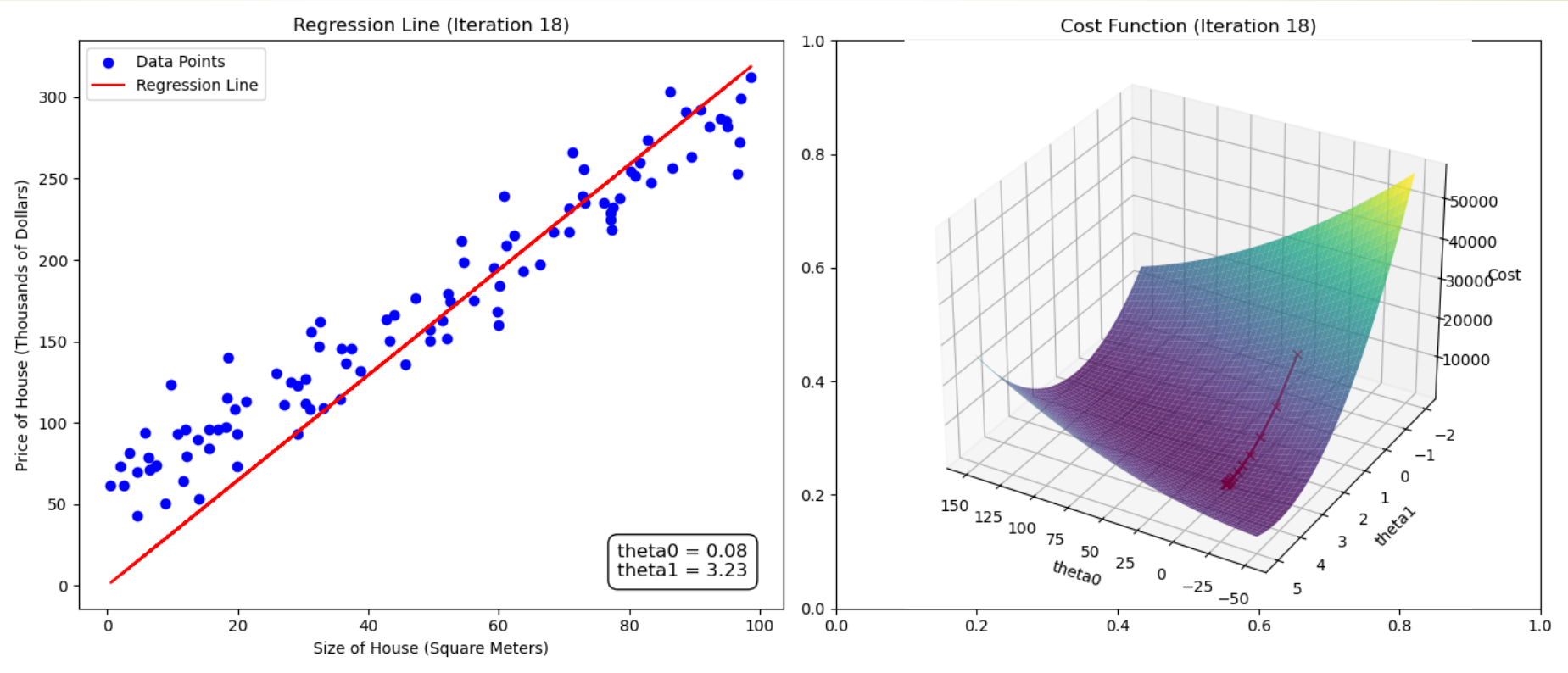

Regression Line (Iteration 13)
Data Points
Regression Line
theta0 = 0.07
theta1 = 3.22
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 13)
Cost
theta0
theta1

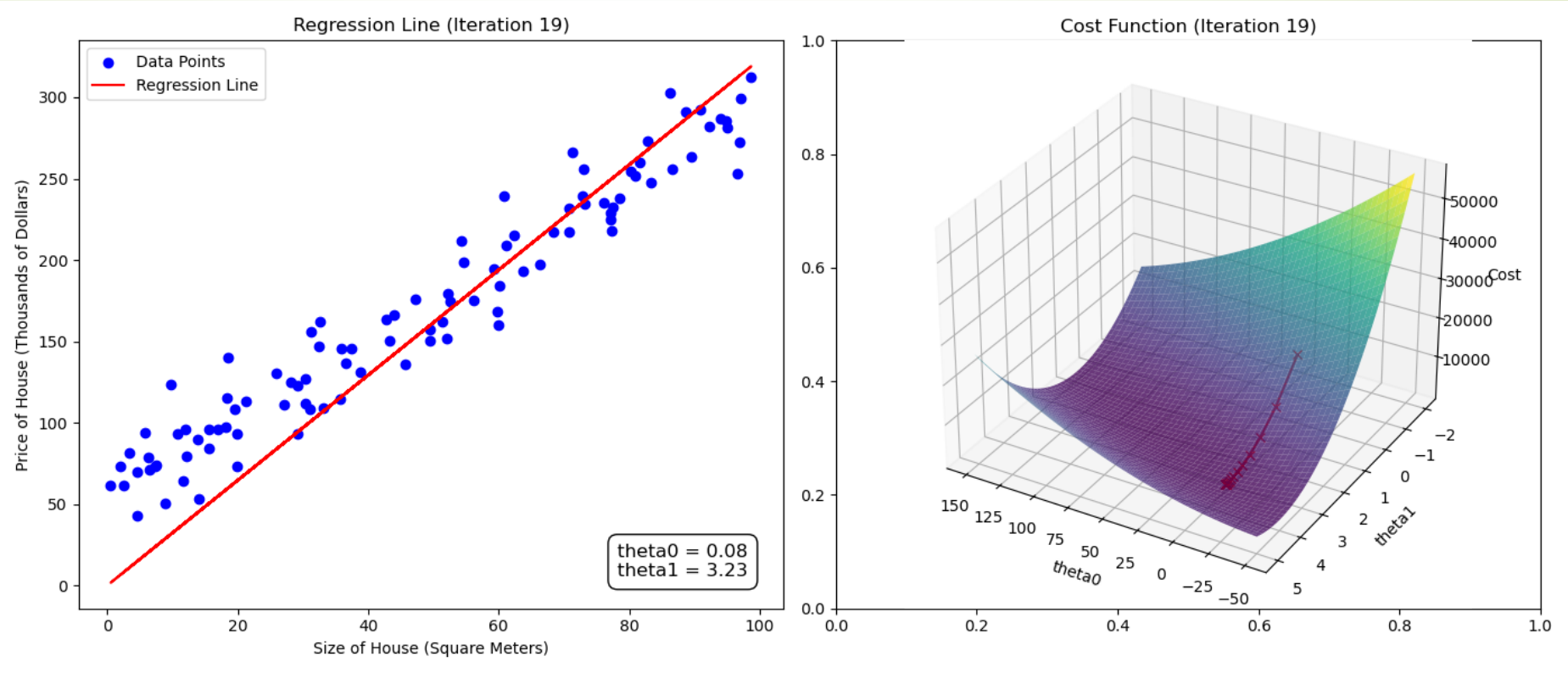

Regression Line (Iteration 14)
Data Points
Regression Line
theta0 = 0.07
theta1 = 3.22
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 14)
Cost
theta0
theta1


Regression Line (Iteration 15)
Data Points
Regression Line
theta0 = 0.07
theta1 = 3.23
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 15)
Cost
theta0
theta1

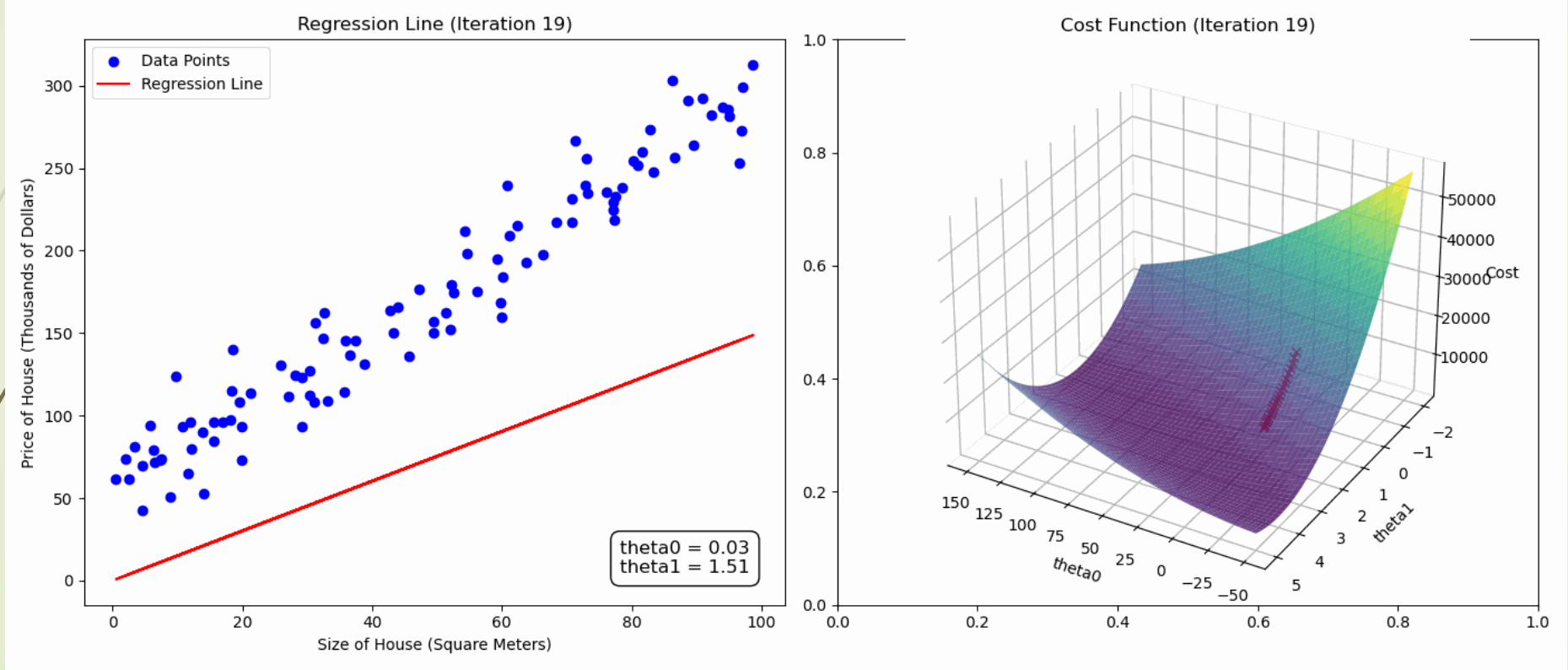

Regression Line (Iteration 16)
Data Points
Regression Line
theta0 = 0.08
theta1 = 3.23
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 16)
Cost
theta0
theta1


Regression Line (Iteration 17)
Data Points
Regression Line
theta0 = 0.08
theta1 = 3.23
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 17)
Cost
theta0
theta1

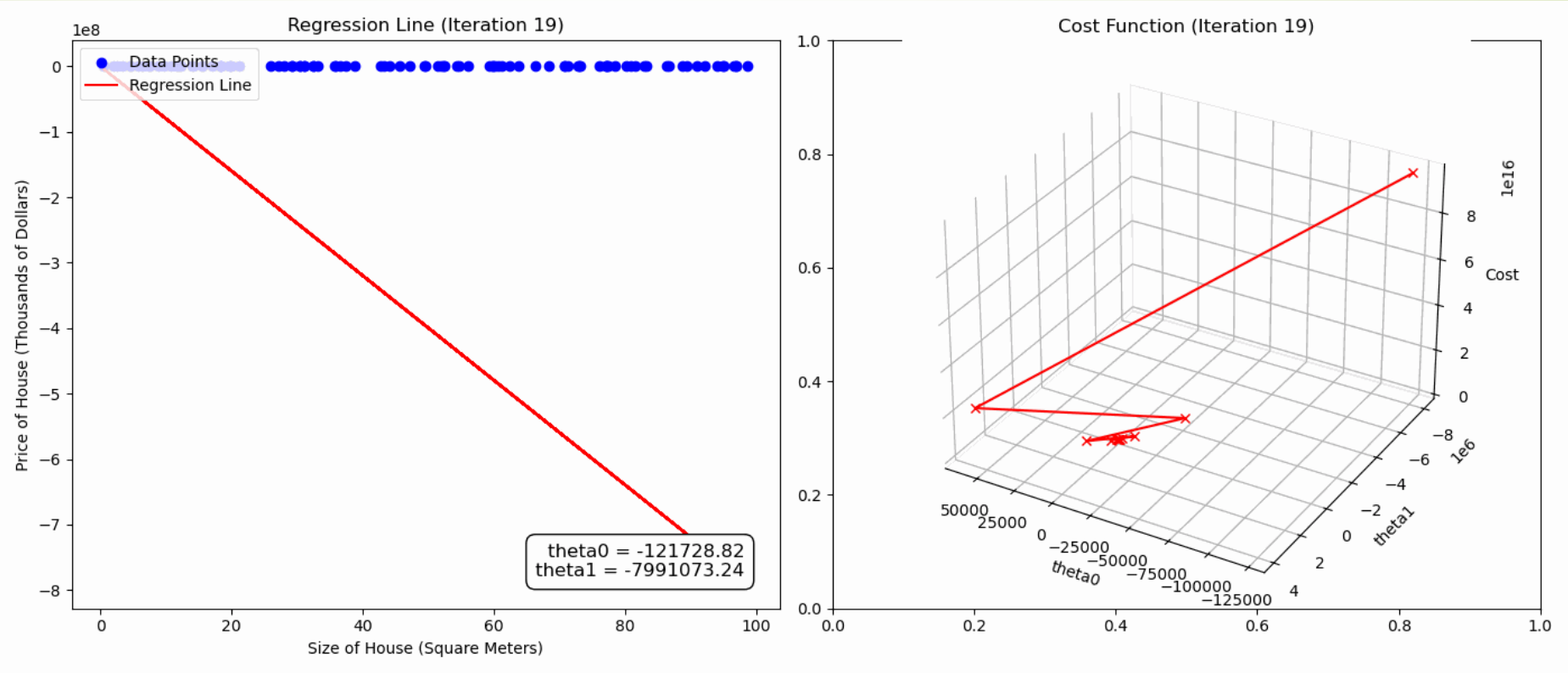

Regression Line (Iteration 18)
Data Points
Regression Line
theta0 = 0.08
theta1 = 3.23
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 18)
Cost
theta0
theta1


Regression Line (Iteration 19)
Data Points
Regression Line
theta0 = 0.08
theta1 = 3.23
Price of House (Thousands of Dollars)
Size of House (Square Meters)
Cost Function (Iteration 19)
Cost
theta0
theta1

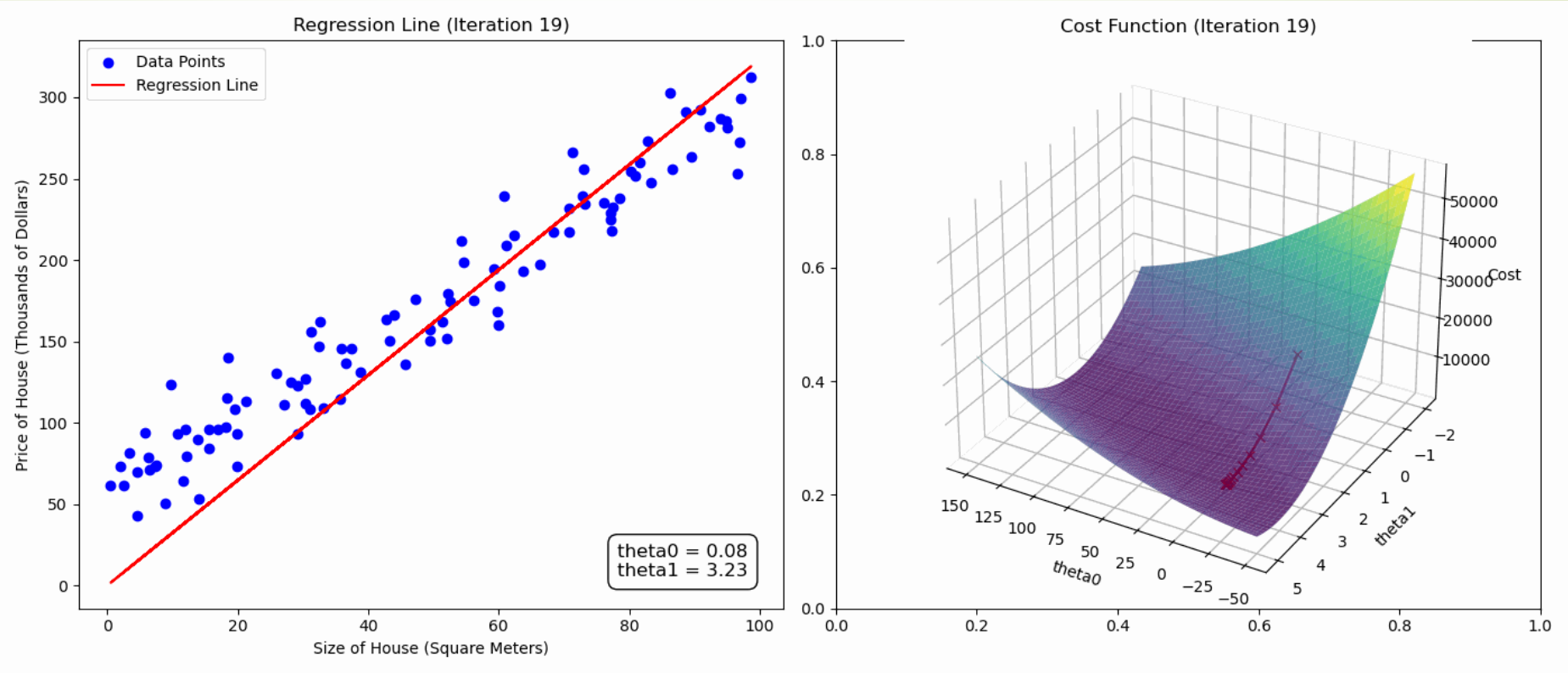
Slower Leaning Rate



Faster Learning Rate – Explode



Proper Learning Rate



Assumptions

Assumptions of Simple Linear Regression

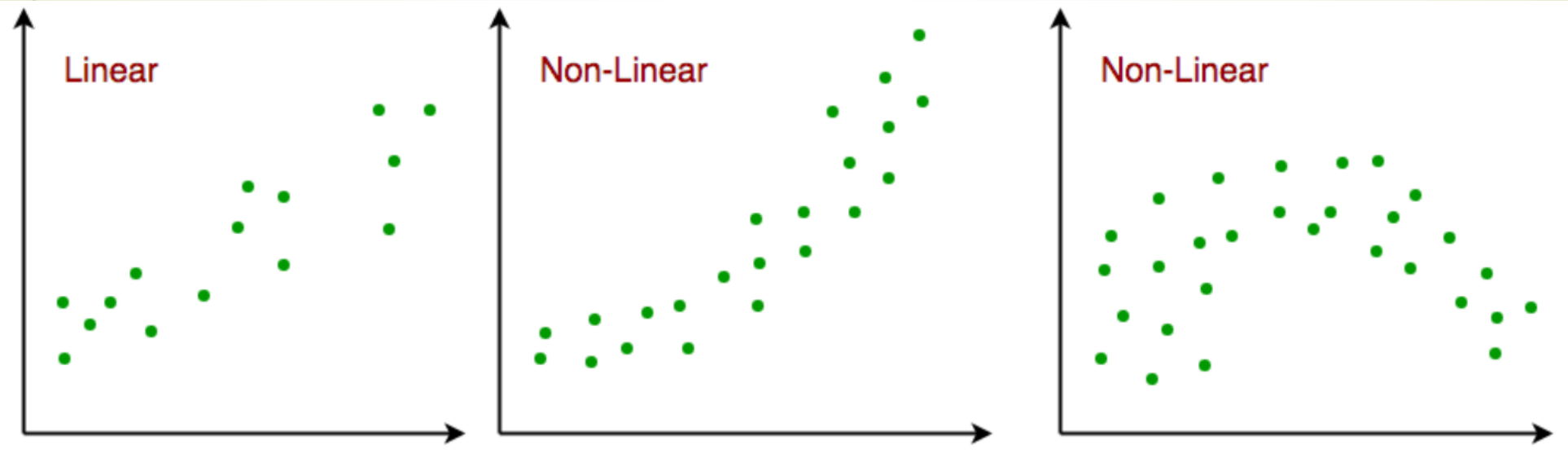
- Linear regression is a widely used tool for predicting and understanding relationships between variables. However, to ensure the **accuracy** and **reliability** of the model, several key **assumptions** must be met.

- There four Key assumption in simple linear regression.

Assumptions of Simple Linear Regression

- **Linearity**: The relationship between the independent variable and the dependent variable must be linear.
  - This means that changes in the dependent variable are proportional to changes in the independent variable.
  - A straight line should reasonably fit the data points.
  - Why it matters: If the relationship is non-linear, linear regression may provide inaccurate predictions.

Assumptions of Simple Linear Regression



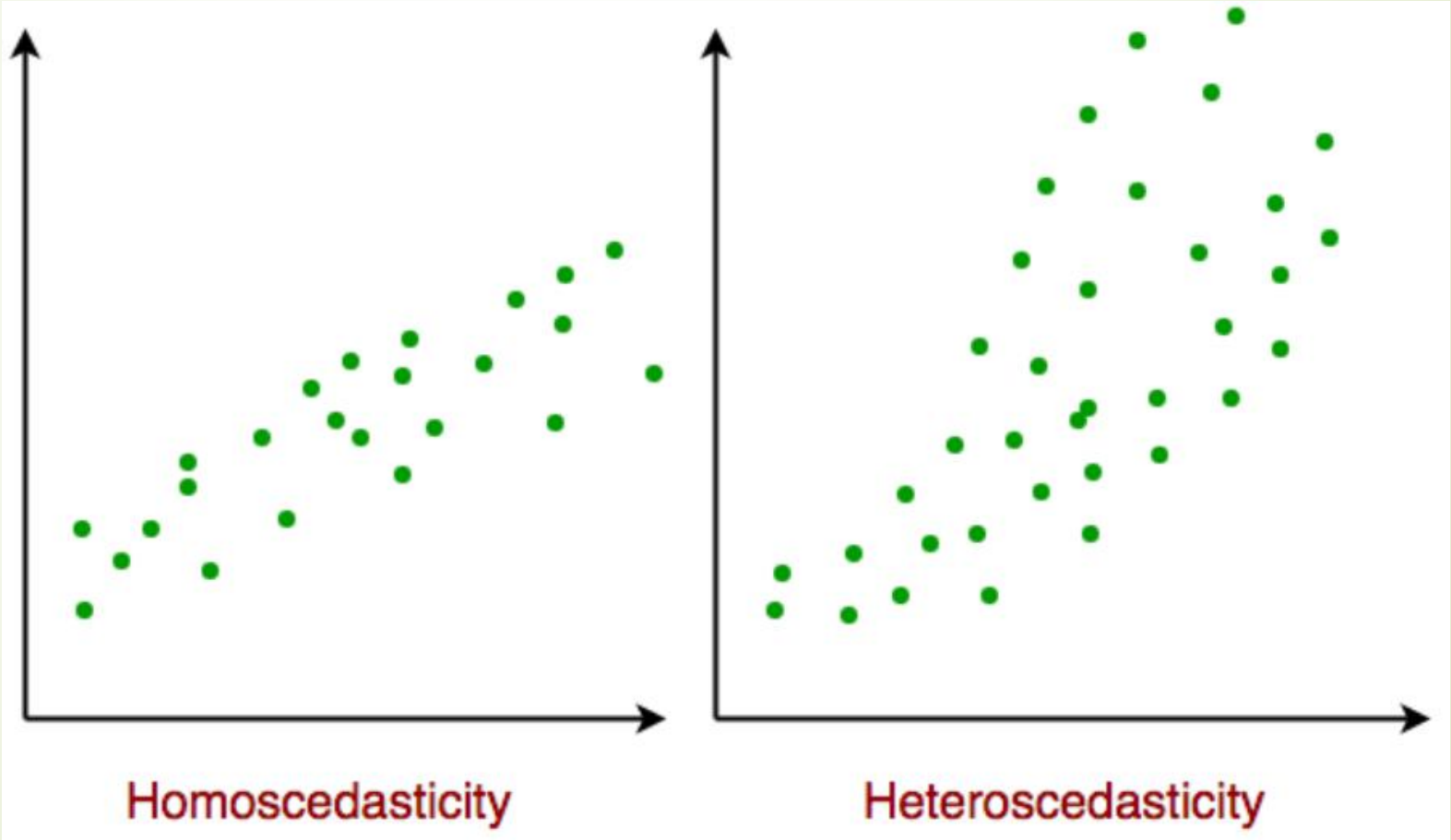
Assumptions of Simple Linear Regression

- **Independence**: The observations should be independent of one another.
  - This means that the value of one data point should not influence the value of another.
  - Why it matters: If the observations are not independent, the model's predictions can be biased or misleading.

Assumptions of Simple Linear Regression

- **Homoscedasticity**: The variance of the residuals (errors) should be constant across all levels of the independent variable.
  - The spread of the errors should be the same for all predicted values.
  - Why it matters: If the variance changes (heteroscedasticity), the model's predictions may be unreliable, especially for certain ranges of the data.

Assumptions of Simple Linear Regression



Assumptions of Simple Linear Regression

- **Normality**: The residuals (errors) should be normally distributed.
  - Residuals should form a bell curve, with most errors centered around zero and fewer large errors.
  - Why it matters: Normality of residuals is important for hypothesis testing and constructing accurate confidence intervals.

References

- https://www.geeksforgeeks.org/ml-linear-regression/
- https://www.analyticsvidhya.com/blog/2021/04/gradient-descent-in-linear-regression/#:~:text=Learning%20rate%20gives%20the%20rate,2.