Multiple Linear Regression

Multiple Linear Regression is a statistical technique used to model the linear relationship between one continuous dependent variable (also called the outcome, target, or response variable) and two or more independent variables (also called predictor, explanatory, or input variables).

- In simpler terms: It tries to find the best-fitting straight line (or plane, or hyperplane in higher dimensions) through the data points to predict the value of a single outcome variable based on the values of several predictor variables.

Key Components and Goal

Component	Notation/Symbol	Description	Type/Constraint
Dependent Variable	Y	The single variable you are trying to predict or explain.	Continuous
Independent Variables	$X_1, X_2, \ldots, X_p$	Two or more variables believed to influence the dependent variable.	Continuous or Categorical[1]
Intercept	β_0 (or θ_0)	The predicted value of Y when all independent variables are zero. Also known as the bias term.	Numerical Coefficient
Coefficients	$\beta_1, \beta_2, \ldots, \beta_p$	The change in Y for a one-unit increase in the corresponding X_i, holding all other independent variables constant. Also known as weights or parameters.	Numerical Coefficients
Error Term	ε (epsilon)	Represents the difference between the actual observed Y and the value predicted by the model ($\hat{y}$). Captures variability not explained by the included independent variables.	Residual
Model Equation		$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \ldots + \beta_pX_p + \varepsilon$	
Prediction Eq.	$\hat{y}$	$\hat{y} = \beta_0 + \beta_1X_1 + \beta_2X_2 + \ldots + \beta_pX_p$ (This is the fitted model used for predictions)	

[1] *Categorical variables often require transformation (e.g., dummy variable coding) before being used in the model.*

[2] *Note: Coefficients are often denoted by β in statistics and ϑ (theta) in machine learning contexts.*

Purpose

- Prediction: To predict the value of the dependent variable given specific values of the independent variables.
- Explanation: To understand and quantify the relationship between each independent variable and the dependent variable, controlling for the effects of the other independent variables.
- Quantifying Strength: To determine the overall strength of the relationship between the set of independent variables and the dependent variable (e.g., using R-squared).

Distinction from Other Regression Types

- Simple Linear Regression: Uses only one independent variable.
- Multivariate Linear Regression: Predicts multiple dependent variables simultaneously.

Worked Example: Predicting y from x_1 and x_2

Problem: Predict y using two input features x_1 and x_2. Find a model of the form: $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$.

Dataset (m=3 training examples)

Example	x_1	x_2	y (Actual)
1	1	2	6
2	2	1	8
3	3	3	14

1. Hypothesis Function

The model's prediction $h\theta(x)$ (or $\hat{y}$) for an input x is:
$h\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

- Vector Notation: $h\theta(x) = \theta^T x$

Vector	Components	Description	Example (for i=1)
θ	$[\theta_0, \theta_1, \theta_2]^T$	Parameter vector (coefficients/weights)	To be determined
x	$[x_0, x_1, x_2]^T$	Feature vector (input), with $x_0 = 1$ for the bias	$[1, 1, 2]^T$

2. Cost Function (Mean Squared Error - MSE)

Measures how well the model fits the training data.
$J(\theta_0, \theta_1, \theta_2) = (1 / 2m) * \Sigma$ [from i=1 to m] $(h\theta(x^{(i)}) - y^{(i)})^2$

- m = 3 (number of training examples)
- $h\theta(x^{(i)}) = \theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$

- $y^{(i)}$ = Actual value for the i-th example
- The 1/2 simplifies gradient calculation.

Substituting our data:
$J(\theta_0, \theta_1, \theta_2) = (1/6) * [(\theta_0 + \theta_1*1 + \theta_2*2 - 6)^2 + (\theta_0 + \theta_1*2 + \theta_2*1 - 8)^2 + (\theta_0 + \theta_1*3 + \theta_2*3 - 14)^2]$

3. Gradient (Partial Derivatives of the Cost Function)

The gradient tells us how to change the parameters to minimize the cost.

Parameter	Partial Derivative Formula	Calculation for this Dataset
θ_0	$\partial J/\partial\theta_0 = (1/m) * \Sigma (h\theta(x^{(i)}) - y^{(i)}) * x_0^{(i)}$ (where x_0=1)	$(1/3) * [(\theta_0 + \theta_1 + 2\theta_2 - 6) + (\theta_0 + 2\theta_1 + \theta_2 - 8) + (\theta_0 + 3\theta_1 + 3\theta_2 - 14)]$
θ_1	$\partial J/\partial\theta_1 = (1/m) * \Sigma (h\theta(x^{(i)}) - y^{(i)}) * x_1^{(i)}$	$(1/3) * [(\theta_0 + \theta_1 + 2\theta_2 - 6)*1 + (\theta_0 + 2\theta_1 + \theta_2 - 8)*2 + (\theta_0 + 3\theta_1 + 3\theta_2 - 14)*3]$
θ_2	$\partial J/\partial\theta_2 = (1/m) * \Sigma (h\theta(x^{(i)}) - y^{(i)}) * x_2^{(i)}$	$(1/3) * [(\theta_0 + \theta_1 + 2\theta_2 - 6)*2 + (\theta_0 + 2\theta_1 + \theta_2 - 8)*1 + (\theta_0 + 3\theta_1 + 3\theta_2 - 14)*3]$

4. Optimization Methods

Two common methods to find the optimal θ values that minimize J(θ):

- Gradient Descent: An iterative algorithm.
    1. Initialize $\theta_0, \theta_1, \theta_2$ (e.g., to 0).
    2. Choose a learning rate α.
    3. Repeat until convergence:
        - Calculate partial derivatives $\partial J/\partial\theta_0, \partial J/\partial\theta_1, \partial J/\partial\theta_2$.
        - Update parameters simultaneously:
          $\theta_j := \theta_j - \alpha * (\partial J/\partial\theta_j)$ for j = 0, 1, 2.
- Normal Equation: A direct, analytical solution (often feasible for smaller datasets).
    - Formula: $\theta = (X^T X)^{-1} X^T y$
    - Where X is the design matrix (with x_0=1) and y is the vector of actual values.

5. Solution using the Normal Equation

For this specific dataset, the Normal Equation yields the exact solution:

Parameter	Optimal Value	Approximate Value
θ_0	0	0.00

θ_1	10/3	3.33
θ_2	4/3	1.33

Final Hypothesis Function (Model):
$\hat{y} = 0 + (10/3)x_1 + (4/3)x_2$
or
$\hat{y} = (10x_1 + 4x_2) / 3$

(Note: Gradient Descent, with an appropriate learning rate and enough iterations, would converge to these same values.)

6. Evaluation

Let's evaluate the model using the found parameters.

Predictions and Residuals:

Example	x_1	x_2	y (Actual)	$\hat{y}$ (Predicted) Calculation	$\hat{y}$ (Predicted) Result	Residual $(y - \hat{y})$
1	1	2	6	(10*1 + 4*2) / 3 = (10+8)/3	6	6 - 6 = 0
2	2	1	8	(10*2 + 4*1) / 3 = (20+4)/3	8	8 - 8 = 0
3	3	3	14	(10*3 + 4*3) / 3 = (30+12)/3	14	14 - 14 = 0

Evaluation Metrics:

Metric	Formula	Calculation Details	Result	Interpretation
MSE (Mean Squared Error)	$(1/m) * \Sigma (y_i - \hat{y}_i)^2$	$(1/3) * (0^2 + 0^2 + 0^2)$	0	Average squared difference between actual and predicted values. 0 indicates a perfect fit on the training data.
RMSE (Root MSE)	√MSE	√0	0	Square root of MSE, in the same units as the dependent variable. 0 indicates a perfect fit.
MAE (Mean Absolute Error)	`(1/m) * Σ	$y_i - \hat{y}_i$	`	`(1/3) * (
R-squared (R^2)	1 - (SS_res / SS_tot)	SS_res = $\Sigma(y_i - \hat{y}_i)^2$ = 0 $\bar{y}$ = (6+8+14)/3 = 28/3 SS_tot = $\Sigma(y_i - \bar{y})^2$ = $(6 - 28/3)^2 + (8 - 28/3)^2 +$ (14 -	1.0	Proportion of the variance in the dependent variable that is predictable from the independent variables. 1.0 (or 100%) indicates that the model

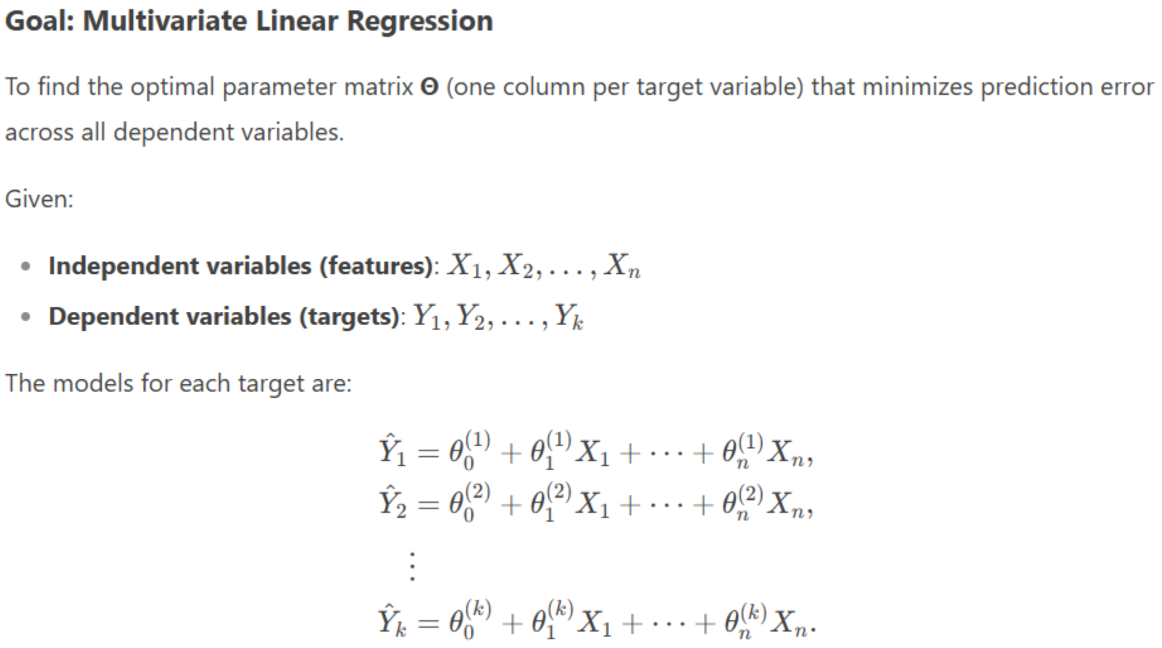
		28/3)² = 312/9 R² = 1 - (0 / (312/9))		explains all the variability of the response data around its mean.

The multiple linear regression model ŷ = (10/3)x_1 + (4/3)x_2 perfectly fits the provided training data. This is demonstrated by the evaluation metrics (MSE=0, RMSE=0, MAE=0, R^2=1.0).

Multivariate Linear Regression: Definition

Multivariate Linear Regression is a statistical technique used to model the linear relationship between **multiple continuous dependent variables** (also called outcomes, targets, or responses) and **one or more independent variables** (also called predictors, explanatory, or input variables).

- **Core Idea:** It extends linear regression to scenarios where we want to predict several related outcomes simultaneously using the same set of input features.
- **Distinction from other types:**
  - **Simple Linear Regression:** One input variable, one output variable.
  - **Multiple Linear Regression:** Multiple input variables, **one** output variable.
  - **Multivariate Linear Regression:** One or more input variables, **multiple** output variables.



Goal: Multivariate Linear Regression

To find the optimal parameter matrix **Θ** (one column per target variable) that minimizes prediction error across all dependent variables.

Given:

- **Independent variables (features)**: $X_1, X_2, \ldots, X_n$
- **Dependent variables (targets)**: $Y_1, Y_2, \ldots, Y_k$

The models for each target are:

$$\begin{aligned}
\hat{Y}_1 &= \theta_0^{(1)} + \theta_1^{(1)} X_1 + \cdots + \theta_n^{(1)} X_n, \\
\hat{Y}_2 &= \theta_0^{(2)} + \theta_1^{(2)} X_1 + \cdots + \theta_n^{(2)} X_n, \\
&\vdots \\
\hat{Y}_k &= \theta_0^{(k)} + \theta_1^{(k)} X_1 + \cdots + \theta_n^{(k)} X_n.
\end{aligned}$$

Problem Setup

- **Objective:** Predict two dependent variables (y_1, y_2) from two independent features (x_1, x_2).

- **Dataset:** (m=3 training examples, n=2 features, k=2 outputs)

Example	x_1	x_2	y_1 (Actual)	y_2 (Actual)
1	1	2	6	3
2	2	1	8	5
3	3	3	14	9

- **Model Equations:**
  - $\hat{y}_1 = \theta_0^{(1)} + \theta_1^{(1)}x_1 + \theta_2^{(1)}x_2$
  - $\hat{y}_2 = \theta_0^{(2)} + \theta_1^{(2)}x_1 + \theta_2^{(2)}x_2$
- **Gradient Descent Parameters:**
  - Learning Rate (α): 0.1
  - Number of Iterations: 3
  - Initial Parameter Values: All θ = 0.

1. Calculations Without Vectorization (Loop-based)

This approach calculates gradients and updates parameters for each output variable (y_1 and y_2) independently within each iteration's loop structure.

Initial Parameters (Vectors):

- $\Theta^{(1)} = [\theta_0^{(1)}, \theta_1^{(1)}, \theta_2^{(1)}]^T = [0.000, 0.000, 0.000]^T$
- $\Theta^{(2)} = [\theta_0^{(2)}, \theta_1^{(2)}, \theta_2^{(2)}]^T = [0.000, 0.000, 0.000]^T$

Iteration 1:

Step	Calculation Details per Example (Ex)	Gradients (Avg over m=3)	Parameter Update (θ := θ - 0.1 * ∇J)	Parameter Values at End of Iter 1
Predictions (h)	h_1=0, h_2=0 for all Ex	---	---	---
Errors (h - y)	Ex1: (-6, -3), Ex2: (-8, -5), Ex3: (-14, -9)	---	---	---

Gradients (∇J)	$\partial J/\partial\theta_0^{(1)} = (1/3)(-6-8-14) = -28/3$	$\partial J/\partial\theta_0^{(1)} = -9.333$	$\theta_0^{(1)} = 0 - 0.1(-9.333) = 0.933$	$\theta_0^{(1)} = 0.933$
	$\partial J/\partial\theta_1^{(1)} = (1/3)(-6*1 -8*2 -14*3) = -64/3$	$\partial J/\partial\theta_1^{(1)} = -21.333$	$\theta_1^{(1)} = 0 - 0.1(-21.333) = 2.133$	$\theta_1^{(1)} = 2.133$
	$\partial J/\partial\theta_2^{(1)} = (1/3)(-6*2 -8*1 -14*3) = -62/3$	$\partial J/\partial\theta_2^{(1)} = -20.667$	$\theta_2^{(1)} = 0 - 0.1(-20.667) = 2.067$	$\theta_2^{(1)} = 2.067$
	$\partial J/\partial\theta_0^{(2)} = (1/3)(-3-5-9) = -17/3$	$\partial J/\partial\theta_0^{(2)} = -5.667$	$\theta_0^{(2)} = 0 - 0.1(-5.667) = 0.567$	$\theta_0^{(2)} = 0.567$
	$\partial J/\partial\theta_1^{(2)} = (1/3)(-3*1 -5*2 -9*3) = -40/3$	$\partial J/\partial\theta_1^{(2)} = -13.333$	$\theta_1^{(2)} = 0 - 0.1(-13.333) = 1.333$	$\theta_1^{(2)} = 1.333$
	$\partial J/\partial\theta_2^{(2)} = (1/3)(-3*2 -5*1 -9*3) = -38/3$	$\partial J/\partial\theta_2^{(2)} = -12.667$	$\theta_2^{(2)} = 0 - 0.1(-12.667) = 1.267$	$\theta_2^{(2)} = 1.267$
End of Iteration 1	---	---	---	$\Theta^{(1)}=[.933,2.133,2.067]^T$ $\Theta^{(2)}=[.567,1.333,1.267]^T$

Iteration 2:

Step	Calculation Details per Example (Ex) using Θ from Iter 1	Gradients (Avg over m=3)	Parameter Update ($\theta := \theta - 0.1 * \nabla J$)	Parameter Values at End of Iter 2
Predictions (h)	Ex1: $h_1=7.200$, $h_2=4.434$ Ex2: $h_1=7.266$, $h_2=4.499$ Ex3: $h_1=13.533$, $h_2=8.367$	---	---	---
Errors (h - y)	Ex1: (1.200, 1.434) Ex2: (-0.734, -0.501) Ex3: (-0.467, -0.633)	---	---	---
Gradients (∇J)	$\partial J/\partial\theta_0^{(1)} = (1/3)(1.2 - 0.734 -0.467) = -0.001/3$	$\partial J/\partial\theta_0^{(1)} = -0.0003$	$\theta_0^{(1)} = 0.933 - 0.1(-0.0003) = 0.933$	$\theta_0^{(1)} = 0.933$

	$\partial J/\partial\theta_1^{(1)} = (1/3)(1.2*1 - 0.734*2 - 0.467*3) = -1.669/3$	$\partial J/\partial\theta_1^{(1)} = -0.556$	$\theta_1^{(1)} = 2.133 - 0.1(-0.556) = 2.189$	$\theta_1^{(1)} = 2.189$
	$\partial J/\partial\theta_2^{(1)} = (1/3)(1.2*2 - 0.734*1 - 0.467*3) = 0.265/3$	$\partial J/\partial\theta_2^{(1)} = 0.088$	$\theta_2^{(1)} = 2.067 - 0.1(0.088) = 2.058$	$\theta_2^{(1)} = 2.058$
	$\partial J/\partial\theta_o^{(2)} = (1/3)(1.434 - 0.501 - 0.633) = 0.300/3$	$\partial J/\partial\theta_o^{(2)} = 0.100$	$\theta_o^{(2)} = 0.567 - 0.1(0.100) = 0.557$	$\theta_o^{(2)} = 0.557$
	$\partial J/\partial\theta_1^{(2)} = (1/3)(1.434*1 - 0.501*2 - 0.633*3) = -1.467/3$	$\partial J/\partial\theta_1^{(2)} = -0.489$	$\theta_1^{(2)} = 1.333 - 0.1(-0.489) = 1.382$	$\theta_1^{(2)} = 1.382$
	$\partial J/\partial\theta_2^{(2)} = (1/3)(1.434*2 - 0.501*1 - 0.633*3) = 0.468/3$	$\partial J/\partial\theta_2^{(2)} = 0.156$	$\theta_2^{(2)} = 1.267 - 0.1(0.156) = 1.251$	$\theta_2^{(2)} = 1.251$
End of Iteration 2	---	---	---	$\Theta^{(1)}=[.933,2.189,2.058]^T$ $\Theta^{(2)}=[.557,1.382,1.251]^T$

Iteration 3:

Step	Calculation Details per Example (Ex) using Θ from Iter 2	Gradients (Avg over m=3)	Parameter Update ($\theta := \theta - 0.1 * \nabla J$)	Parameter Values at End of Iter 3
Predictions (h)	Ex1: h_1=7.238, h_2=4.441 Ex2: h_1=7.369, h_2=4.572 Ex3: h_1=13.674, h_2=8.456	---	---	---
Errors (h - y)	Ex1: (1.238, 1.441) Ex2: (-0.631, -0.428) Ex3: (-0.326, -0.544)	---	---	---
Gradients (∇J)	$\partial J/\partial\theta_o^{(1)} = (1/3)(1.238 - 0.631 - 0.326) = 0.281/3$	$\partial J/\partial\theta_o^{(1)} = 0.094$	$\theta_o^{(1)} = 0.933 - 0.1(0.094) = 0.924$	$\theta_o^{(1)} = 0.924$
	$\partial J/\partial\theta_1^{(1)} = (1/3)(1.238*1 - 0.631*2 - 0.326*3) = -1.002/3$	$\partial J/\partial\theta_1^{(1)} = -0.334$	$\theta_1^{(1)} = 2.189 - 0.1(-0.334) = 2.222$	$\theta_1^{(1)} = 2.222$

	$\partial J/\partial \theta_2^{(1)} = (1/3)(1.238*2 -0.631*1 -0.326*3) = 0.867/3$	$\partial J/\partial \theta_2^{(1)} = 0.289$	$\theta_2^{(1)} = 2.058 - 0.1(0.289) = 2.029$	$\theta_2^{(1)} = 2.029$
	$\partial J/\partial \theta_o^{(2)} = (1/3)(1.441 - 0.428 -0.544) = 0.469/3$	$\partial J/\partial \theta_o^{(2)} = 0.156$	$\theta_o^{(2)} = 0.557 - 0.1(0.156) = 0.541$	$\theta_o^{(2)} = 0.541$
	$\partial J/\partial \theta_1^{(2)} = (1/3)(1.441*1 -0.428*2 -0.544*3) = -1.047/3$	$\partial J/\partial \theta_1^{(2)} = -0.349$	$\theta_1^{(2)} = 1.382 - 0.1(-0.349) = 1.417$	$\theta_1^{(2)} = 1.417$
	$\partial J/\partial \theta_2^{(2)} = (1/3)(1.441*2 -0.428*1 -0.544*3) = 0.822/3$	$\partial J/\partial \theta_2^{(2)} = 0.274$	$\theta_2^{(2)} = 1.251 - 0.1(0.274) = 1.224$	$\theta_2^{(2)} = 1.224$
End of Iteration 3	---	---	---	$\Theta^{(1)}=[.924,2.222,2.029]^T$ $\Theta^{(2)}=[.541,1.417,1.224]^T$

2. Calculations With Vectorization (Matrix-based)

This approach uses matrix operations for efficiency and conciseness.

Matrix Definitions:

- **Design Matrix X (m x (n+1) = 3x3):** (Features + Bias Term)
- [[1.0, 1.0, 2.0],
- [1.0, 2.0, 1.0],
- [1.0, 3.0, 3.0]]

- **Target Matrix Y (m x k = 3x2):** (Actual Outcomes)
- [[6.0, 3.0],
- [8.0, 5.0],
- [14.0, 9.0]]

- **Parameter Matrix Θ ((n+1) x k = 3x2):** (Weights/Coefficients)
- [[$\theta_o^{(1)}$, $\theta_o^{(2)}$],

- [$\theta_1^{(1)}$, $\theta_1^{(2)}$],
- [$\theta_2^{(1)}$, $\theta_2^{(2)}$]]

Vectorized Formulas:

- Hypothesis Matrix: $H = X @ \Theta$ (Matrix multiplication)
- Error Matrix: $E = H - Y$
- Gradient Matrix: $\nabla\Theta = (1/m) * X^T @ E$ (Transpose X, then matrix multiply)
- Parameter Update: $\Theta := \Theta - \alpha * \nabla\Theta$

Initial Parameter Matrix $\Theta^{(0)}$ (3x2):

[[0.000, 0.000],

[0.000, 0.000],

[0.000, 0.000]]

Iteration 1:

Step	Matrix Calculation	Resulting Matrix
Prediction $H^{(0)}$	$X @ \Theta^{(0)}$	[[0.000, 0.000], [0.000, 0.000], [0.000, 0.000]]
Error $E^{(0)}$	$H^{(0)} - Y$	[[-6.000, -3.000], [-8.000, -5.000], [-14.000, -9.000]]
Gradient $\nabla\Theta^{(0)}$	$(1/3) * X^T @ E^{(0)}$	[[-9.333, -5.667], [-21.333, -13.333], [-20.667, -12.667]]
Update to $\Theta^{(1)}$	$\Theta^{(0)} - 0.1 * \nabla\Theta^{(0)}$	[[0.933, 0.567], [2.133, 1.333], [2.067, 1.267]]
End Iter 1 Θ	---	**$\Theta^{(1)} \approx$ [[0.933, 0.567], [2.133, 1.333], [2.067, 1.267]]**

Iteration 2:

Step	Matrix Calculation	Resulting Matrix
Prediction $H^{(1)}$	$X @ \Theta^{(1)}$	[[7.200, 4.434], [7.266, 4.499], [13.533, 8.367]]
Error $E^{(1)}$	$H^{(1)} - Y$	[[1.200, 1.434], [-0.734, -0.501], [-0.467, -0.633]]
Gradient $\nabla\Theta^{(1)}$	$(1/3) * X^T @ E^{(1)}$	[[-0.0003, 0.100], [-0.556, -0.489], [0.088, 0.156]]
Update to $\Theta^{(2)}$	$\Theta^{(1)} - 0.1 * \nabla\Theta^{(1)}$	[[0.933, 0.557], [2.189, 1.382], [2.058, 1.251]]
End Iter 2 Θ	---	**$\Theta^{(2)} \approx$ [[0.933, 0.557], [2.189, 1.382], [2.058, 1.251]]**

Iteration 3:

Step	Matrix Calculation	Resulting Matrix
Prediction $H^{(2)}$	X @ $\Theta^{(2)}$	[[7.238, 4.441], [7.369, 4.572], [13.674, 8.456]]
Error $E^{(2)}$	$H^{(2)}$ - Y	[[1.238, 1.441], [-0.631, -0.428], [-0.326, -0.544]]
Gradient $\nabla\Theta^{(2)}$	(1/3) * X^T @ $E^{(2)}$	[[0.094, 0.156], [-0.334, -0.349], [0.289, 0.274]]
Update to $\Theta^{(3)}$	$\Theta^{(2)}$ - 0.1 * $\nabla\Theta^{(2)}$	[[0.924, 0.541], [2.222, 1.417], [2.029, 1.224]]
End Iter 3 Θ	---	**$\Theta^{(3)}$ ≈ [[0.924, 0.541], [2.222, 1.417], [2.029, 1.224]]**

Summary: Parameter Estimates After 3 Iterations

The estimated parameter matrix Θ after 3 iterations, derived consistently from both methods, is approximately:

Parameter Matrix $\Theta^{(3)}$

[[0.924 , 0.541],   <-- [$\theta_0^{(1)}$, $\theta_0^{(2)}$] (Intercepts)

[2.222 , 1.417],   <-- [$\theta_1^{(1)}$, $\theta_1^{(2)}$] (Coefficients for x_1)

[2.029 , 1.224]]   <-- [$\theta_2^{(1)}$, $\theta_2^{(2)}$] (Coefficients for x_2)

Final Model Equations (after 3 iterations):

- $\hat{y}_1 \approx 0.924 + 2.222 * x_1 + 2.029 * x_2$
- $\hat{y}_2 \approx 0.541 + 1.417 * x_1 + 1.224 * x_2$

1. Evaluation Metrics Definitions

Mean Squared Error (MSE)

- **Definition**: Average of the squared differences between actual and predicted values.
- **Formula**:

$$\mathrm{MSE}_j = \frac{1}{m} \sum_{i=1}^{m} \left(y_j^{(i)} - \hat{y}_j^{(i)} \right)^2$$

- **Interpretation**: Lower values indicate better performance. Units are squared.

Root Mean Squared Error (RMSE)

- **Definition**: Square root of MSE, bringing error back to original units.
- **Formula**:

$$\mathrm{RMSE}_j = \sqrt{\mathrm{MSE}_j}$$

- **Interpretation**: Represents typical prediction error magnitude.

Mean Absolute Error (MAE)

- **Definition**: Average of absolute differences between actual and predicted values.
- **Formula**:

$$\mathrm{MAE}_j = \frac{1}{m} \sum_{i=1}^{m} \left| y_j^{(i)} - \hat{y}_j^{(i)} \right|$$

- **Interpretation**: Less sensitive to outliers than MSE/RMSE.

R-squared (R^2)

- **Definition**: Proportion of variance in the dependent variable explained by the model.
- **Formula**:

$$R_j^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res},j}}{\mathrm{SS}_{\mathrm{tot},j}}$$

  where:

$$\mathrm{SS}_{\mathrm{res},j} = \sum_{i=1}^{m} \left(y_j^{(i)} - \hat{y}_j^{(i)} \right)^2, \quad \mathrm{SS}_{\mathrm{tot},j} = \sum_{i=1}^{m} \left(y_j^{(i)} - \bar{y}_j \right)^2$$

- **Interpretation**: Ranges from 0 to 1 (higher is better).

2. Example Evaluation (After 3 Iterations)

Data

- **Actual Values (Y)**:

$$\begin{bmatrix} 6 & 3 \\ 8 & 5 \\ 14 & 9 \end{bmatrix}$$

- **Predicted Values ($\hat{Y}$)**:

$$\begin{bmatrix} 7.238 & 4.441 \\ 7.369 & 4.572 \\ 13.674 & 8.456 \end{bmatrix}$$

Residuals ($Y - \hat{Y}$)

$$\begin{bmatrix} -1.238 & -1.441 \\ 0.631 & 0.428 \\ 0.326 & 0.544 \end{bmatrix}$$

3. Calculations for Each Dependent Variable

For y_1 (Exam Score)

- **Sum of Squared Residuals (SS_{res})**:

$$(-1.238)^2 + 0.631^2 + 0.326^2 = 2.037$$

- **Total Sum of Squares (SS_{tot})**:

$$(6 - 9.333)^2 + (8 - 9.333)^2 + (14 - 9.333)^2 = 34.667$$

- **Metrics**:

$$\begin{aligned} \mathrm{MSE}_1 &= \frac{2.037}{3} \approx 0.679, \\ \mathrm{RMSE}_1 &= \sqrt{0.679} \approx 0.824, \\ \mathrm{MAE}_1 &= \frac{1.238 + 0.631 + 0.326}{3} \approx 0.732, \\ R_1^2 &= 1 - \frac{2.037}{34.667} \approx 0.941. \end{aligned}$$

For y_2 (Project Score)

For y_2 (Project Score)

- **Sum of Squared Residuals (SS_{res})**:

$$(-1.441)^2 + 0.428^2 + 0.544^2 = 2.555$$

- **Total Sum of Squares (SS_{tot})**:

$$(3 - 5.667)^2 + (5 - 5.667)^2 + (9 - 5.667)^2 = 18.667$$

- **Metrics**:

$$\begin{aligned} \mathrm{MSE}_2 &= \frac{2.555}{3} \approx 0.852, \\ \mathrm{RMSE}_2 &= \sqrt{0.852} \approx 0.923, \\ \mathrm{MAE}_2 &= \frac{1.441 + 0.428 + 0.544}{3} \approx 0.804, \\ R_2^2 &= 1 - \frac{2.555}{18.667} \approx 0.863. \end{aligned}$$

4. Summary Table of Metrics

Metric	Value for y_1	Value for y_2	Interpretation
MSE	0.679	0.852	Lower is better (avg. squared error).
RMSE	0.824	0.923	Lower is better (error in orig. units).
MAE	0.732	0.804	Lower is better (avg. absolute error).
R^2	0.941	0.863	Closer to 1 is better (variance explained).

5. Key Takeaways

1. **Performance**:
   - y_1 (Exam Score) has better metrics (lower errors, higher R^2) than y_2 (Project Score).
2. **Model Fit**:
   - R^2 values close to 1 indicate the model explains most variance for both targets.
3. **Averaging Metrics**:
   - An overall score (e.g., average MSE) can be computed but may mask individual performance differences.