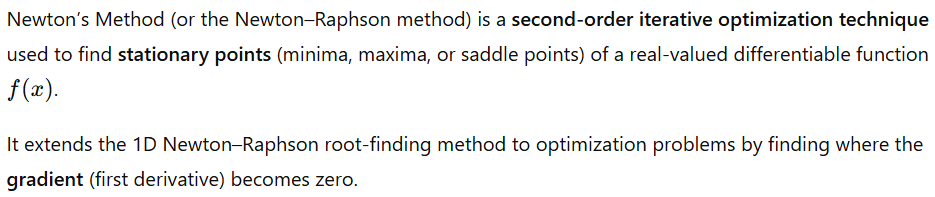
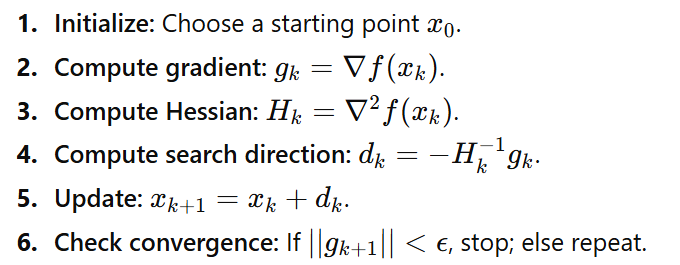
**Newton’s Method in Optimization**

****

**Objective:**

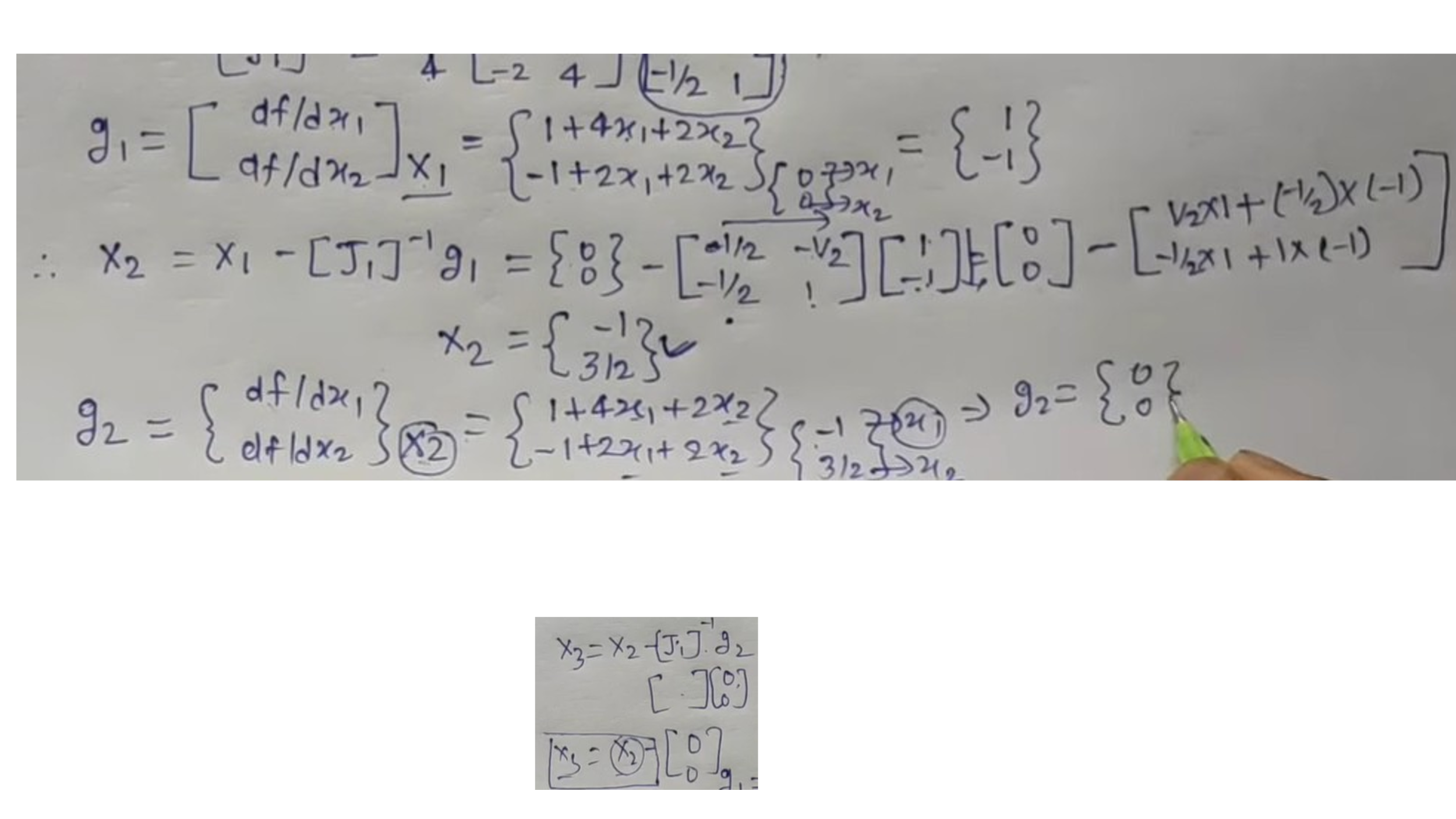
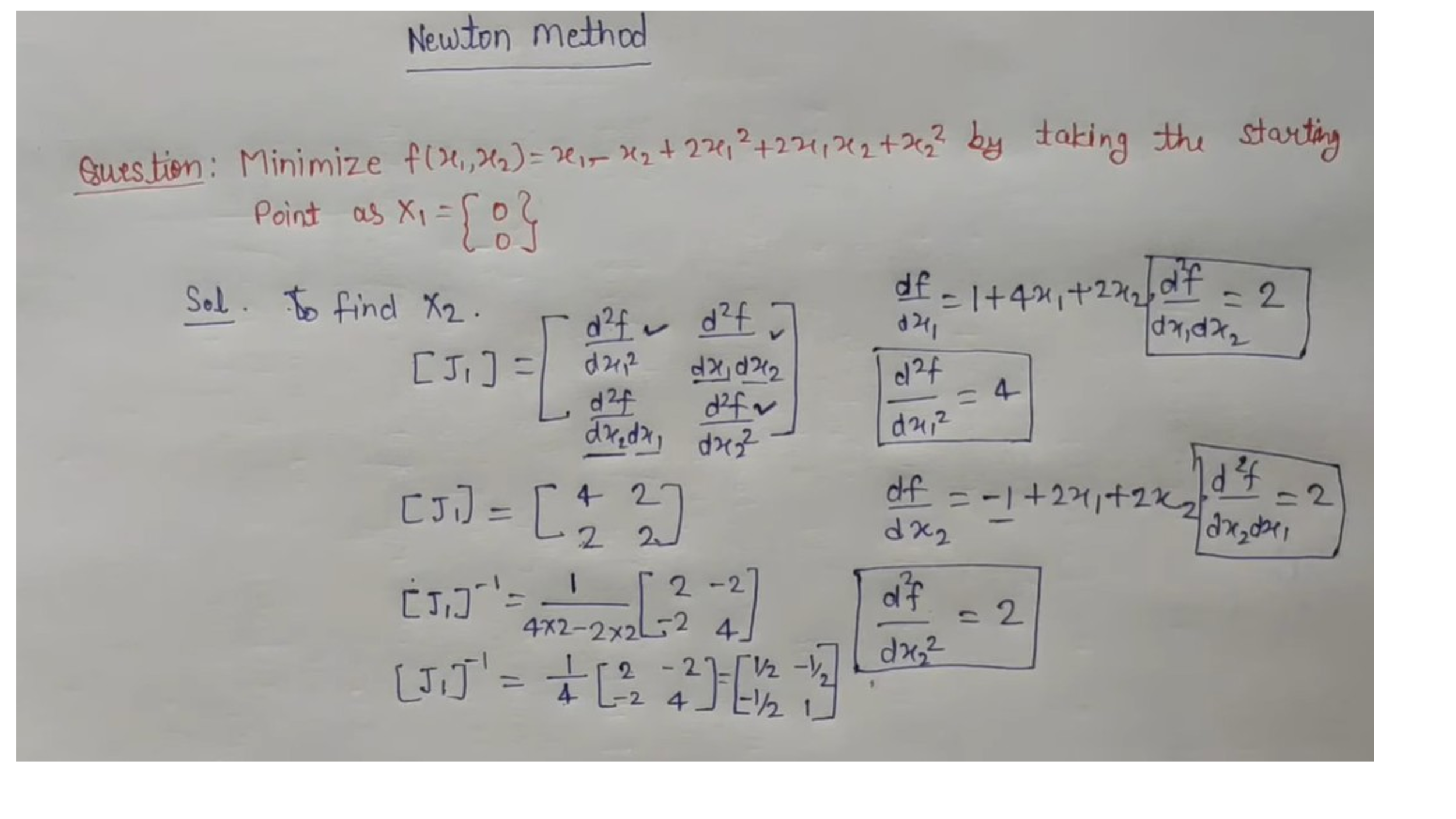
****

**Algorithm**

****

**Disadvantages of Newton’s Method**

|  |  |
| --- | --- |
| **Disadvantage** | **Explanation** |
| 1. Requires Hessian computation | The Hessian ((n \times n) matrix) must be computed and inverted — expensive for large (n). |
| 2. May not converge | If the Hessian is not positive definite (saddle point or maximum), the step can move away from minimum. |
| 3. Sensitive to initial guess | Poor starting point can lead to divergence or convergence to the wrong stationary point. |
| 4. High computational cost | Computing and inverting the Hessian costs (O(n^3)). |
| 5. Not suitable for non-smooth functions | Requires continuous second derivatives. |
| 6. Step may overshoot | If the step size is too large, the quadratic approximation fails — often a line search or damping factor is added. |

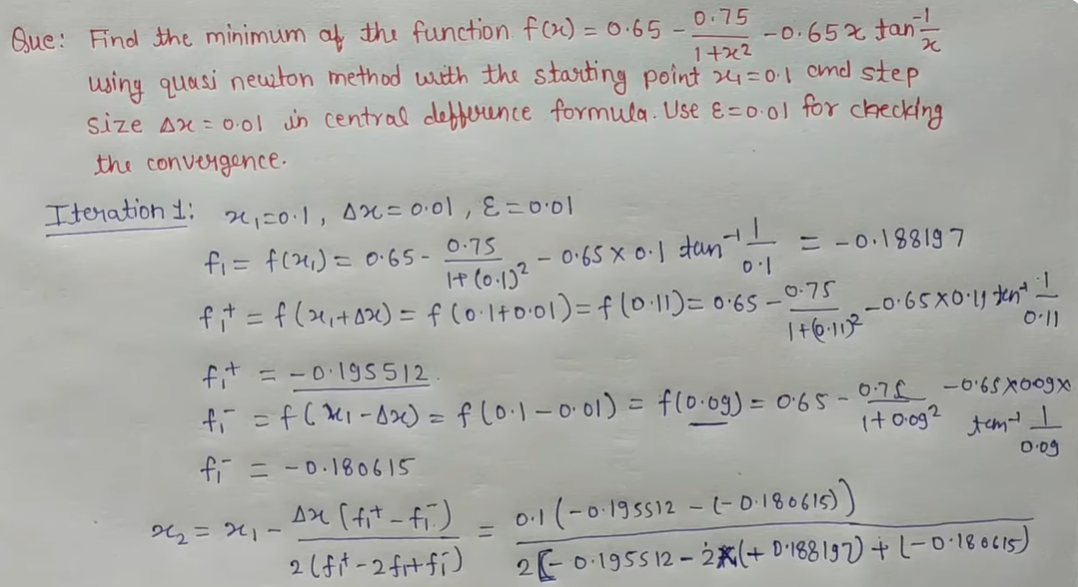


**Newton vs. Gradient Descent**

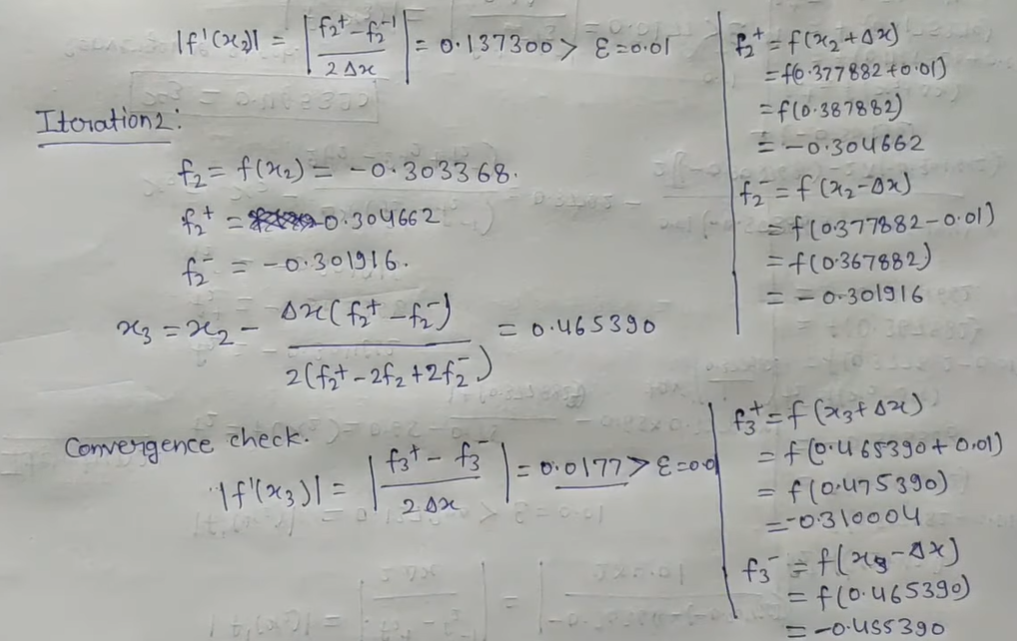
| **Feature** | **Gradient Descent** | **Newton’s Method** |
| --- | --- | --- |
| Uses | Gradient only | Gradient + Hessian |
| Step Direction | Negative gradient | Inverse Hessian × gradient |
| Convergence Rate | Linear | Quadratic (faster near optimum) |
| Cost per Iteration | Low | High (Hessian inversion) |
| Suitable for | Large-scale, simple problems | Smaller, well-behaved quadratic problems |

**Quasi Newton Method**

* Quasi-Newton methods are optimization algorithms that find minima or maxima of functions by approximating the Hessian matrix of second derivatives.
* Unlike Newton's method, they avoid the computational cost of calculating the true Hessian, instead updating an approximation using gradient and position information from previous steps.
* This makes them more efficient for large-scale problems while still achieving superlinear convergence.

****

**So, we have x2 = 0.377882**





**Concept of Regression Tree**

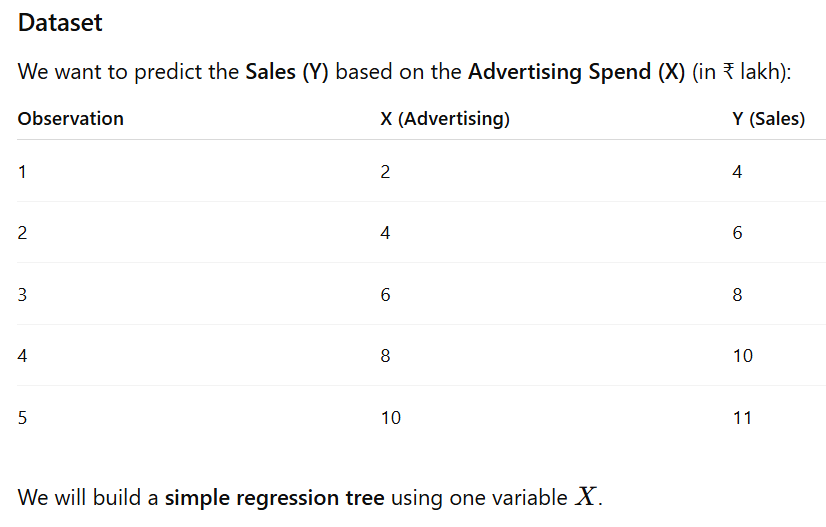
A Regression Tree is a type of Decision Tree used when the target (output) variable is continuous (not categorical).

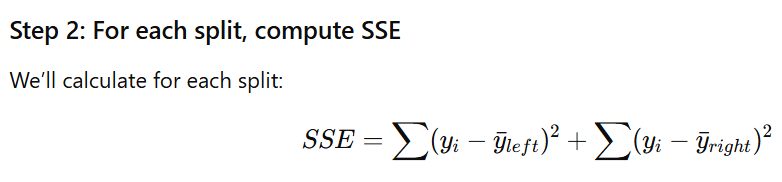
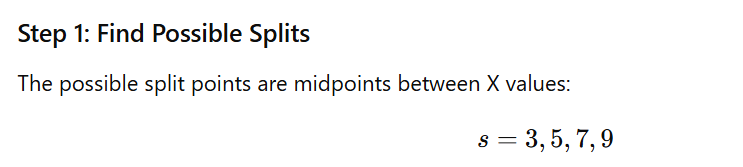
It divides the data into smaller and smaller regions so that the value of the dependent variable (Y) within each region is as homogeneous as possible.

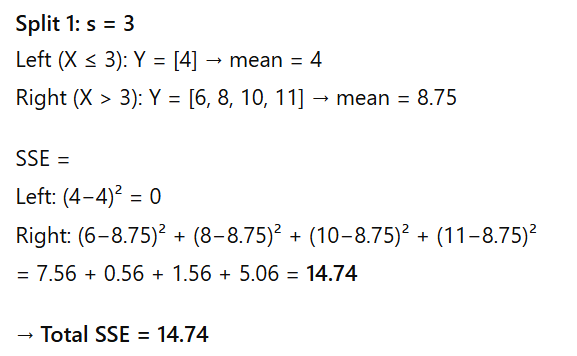
**How It Works**

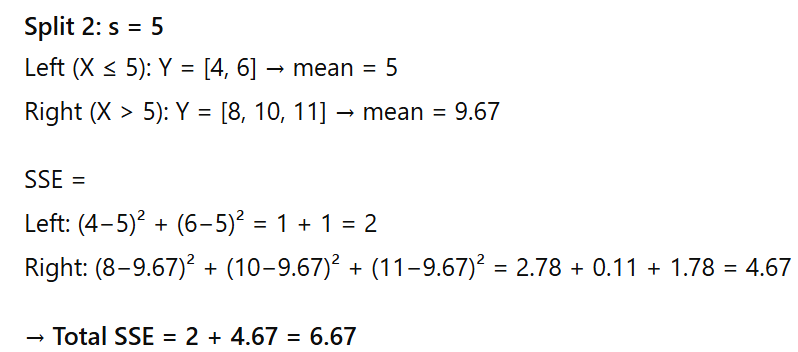
1. Start with all the training data.
2. At each node, select the variable and the split point that minimizes the sum of squared errors (SSE) or variance within the resulting groups.
3. Repeat splitting recursively until a stopping condition is met (e.g., minimum number of samples, or no significant improvement).
4. The predicted value for each terminal node (leaf) is the mean of the target variable in that region.

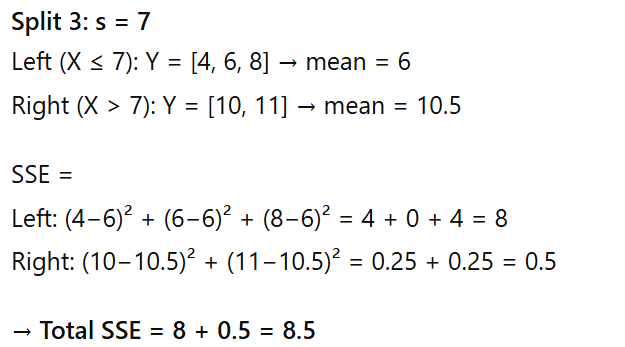
****

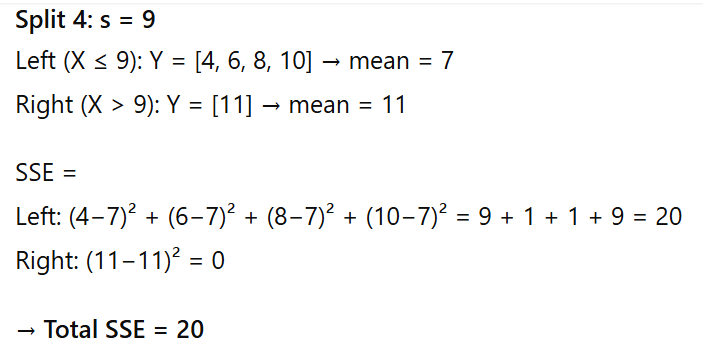
****

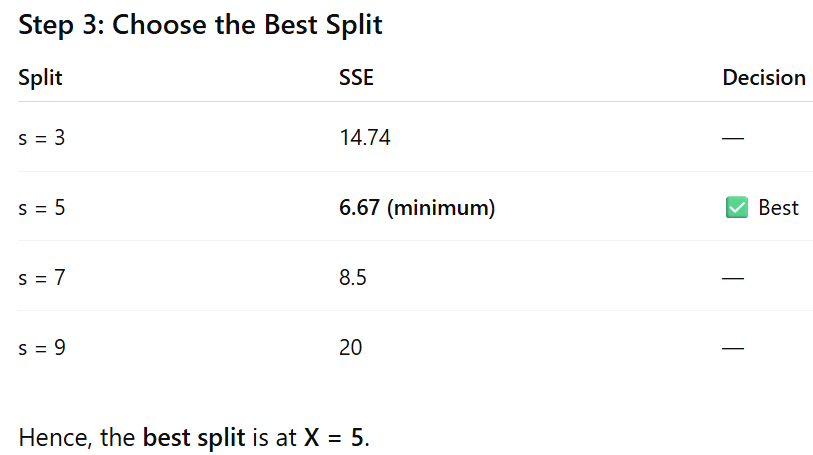
****

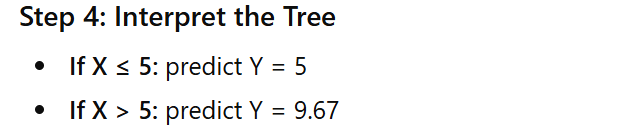
****

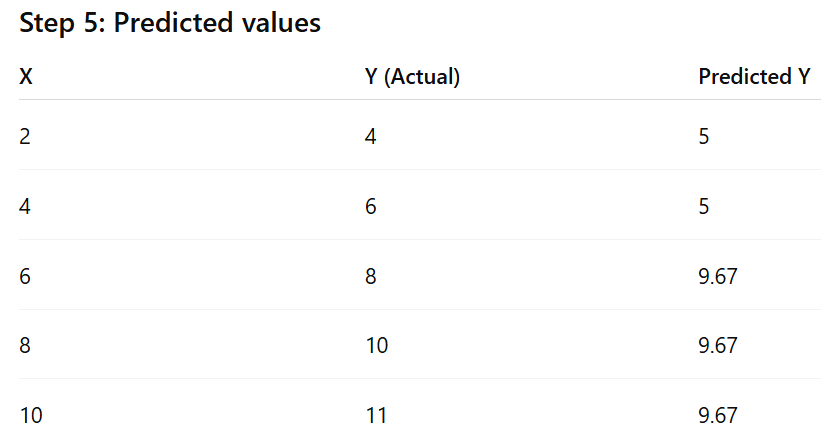
****

****

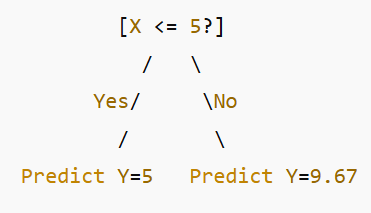
****

****

****

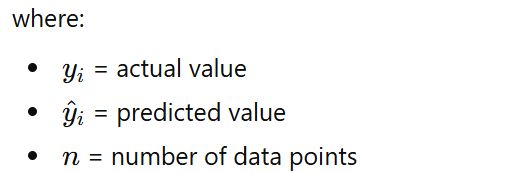
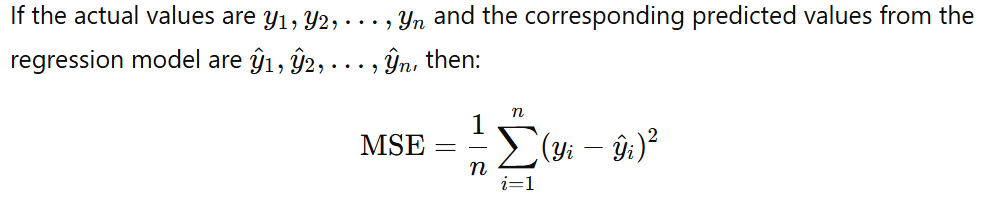
****

**Final Regression Tree**

****

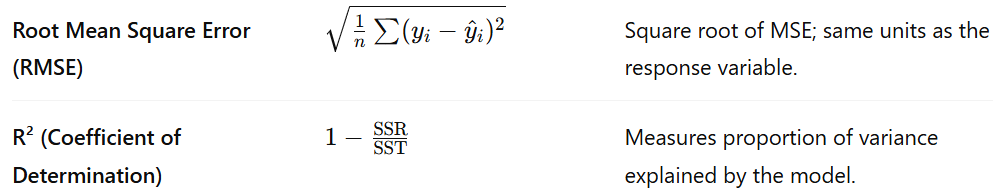
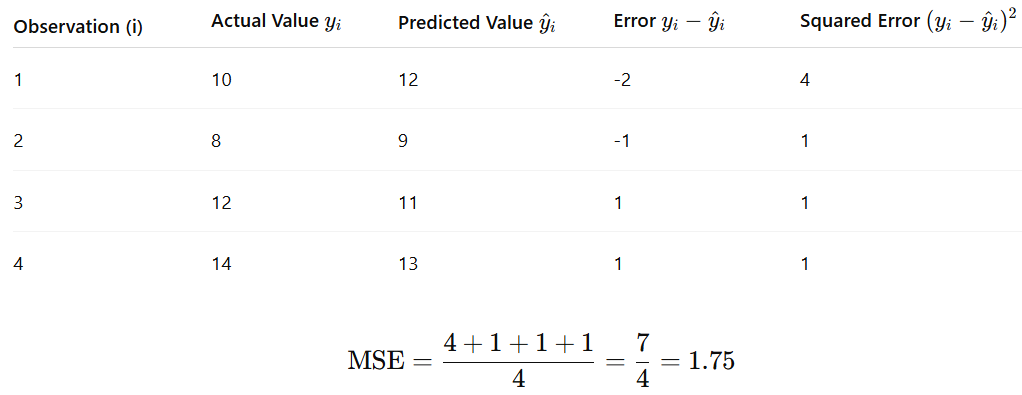
**Mean Squared Error (MSE) in Regression Analysis**

In regression analysis, the **Mean Squared Error (MSE)** is a commonly used metric to measure how well a regression model fits the data. It represents the **average of the squares of the errors** — that is, the average squared difference between the **actual (observed)** and the **predicted** values. It is used as a **loss function** in regression algorithms (e.g., linear regression, neural networks). During model training, the goal is often to **minimize the MSE**.

****

**Interpretation**

* **MSE = 0** means a perfect fit (predictions are exactly equal to actual values).
* A **smaller MSE** value indicates a better fit of the regression model to the data.
* A **larger MSE** value indicates poor predictive accuracy — the model’s predictions deviate more from actual data.



**Advantages of MSE**

* Easy to compute and widely used.
* Useful in optimization algorithms (like gradient descent in machine learning).
* Penalizes large errors heavily due to squaring, which helps discourage outliers.

**Limitations of MSE**

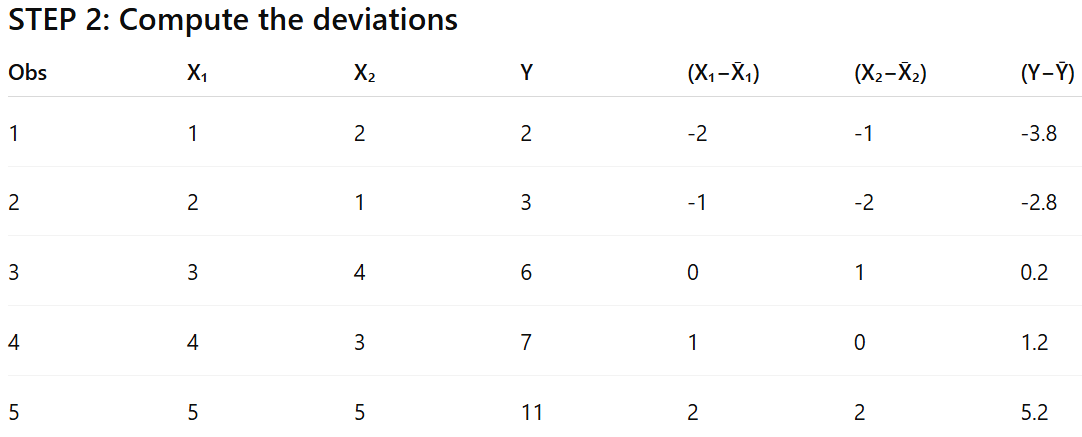
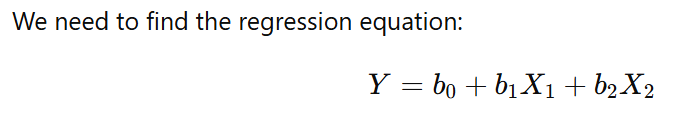
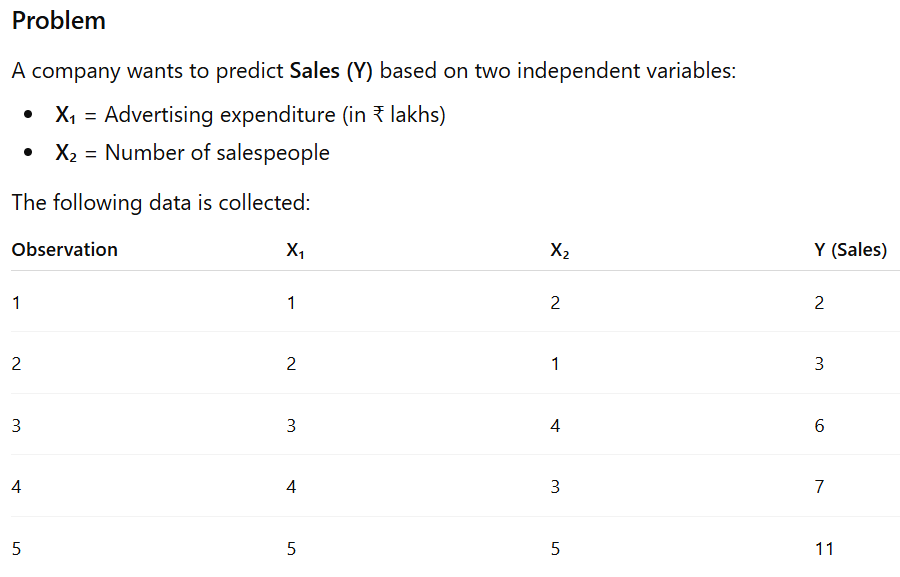
* Sensitive to outliers because errors are squared.
* The value is not in the same units as the original data (since errors are squared).
* May not be intuitive for interpretation compared to MAE.

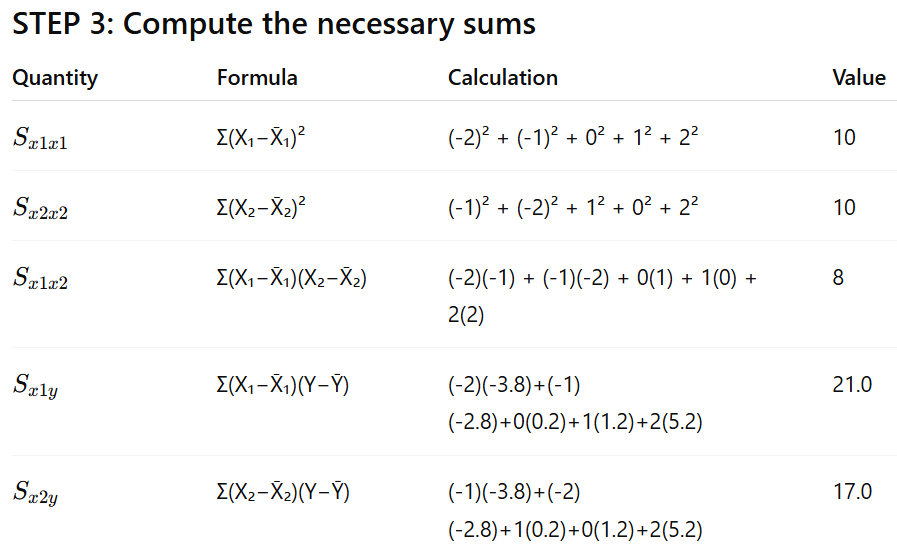
**Usage in Model Evaluation**

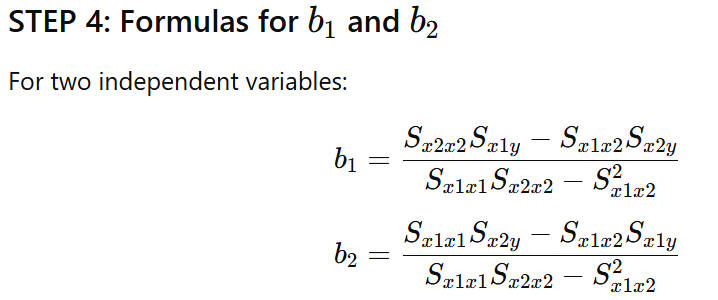
**In regression analysis and machine learning:**

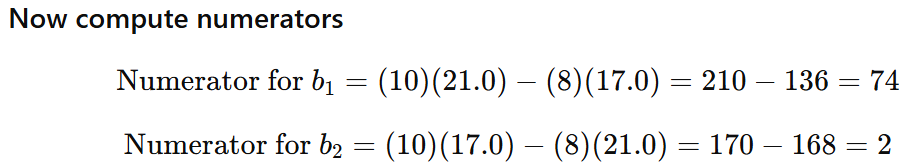
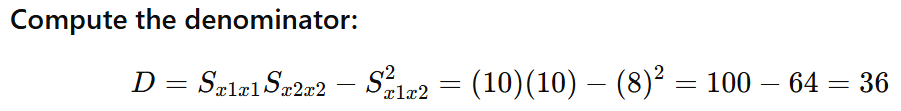
* MSE helps compare different models — the one with the lowest MSE is usually preferred.
* It is commonly used as the loss function in training models like **Linear Regression**, **Ridge Regression**, and **Neural Networks**.

**Numerical on multiple regression analysis**

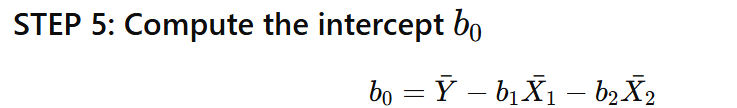








Hence b1 = 74/36 = 2.056, b2 = 2/36 = 0.056



**Value of b0 is = 5.8 – (2.056 \*3) – (0.056 \* 3) = = 5.8 – 6.336 = - 0.536**

**Final Regression Equation is Y = - 0.536 + 2.056 X1 + 0.056 X2**

**Interpretation**

* When both X1 and X2​ = 0, predicted sales = **– 0.536** (the intercept).
* For every unit increase in **advertising expenditure (X₁)**, sales increase by 2.056 **units** (holding X₂ constant).
* For every additional **salesperson (X₂)**, sales increase by **0.667 units** (holding X₁ constant).