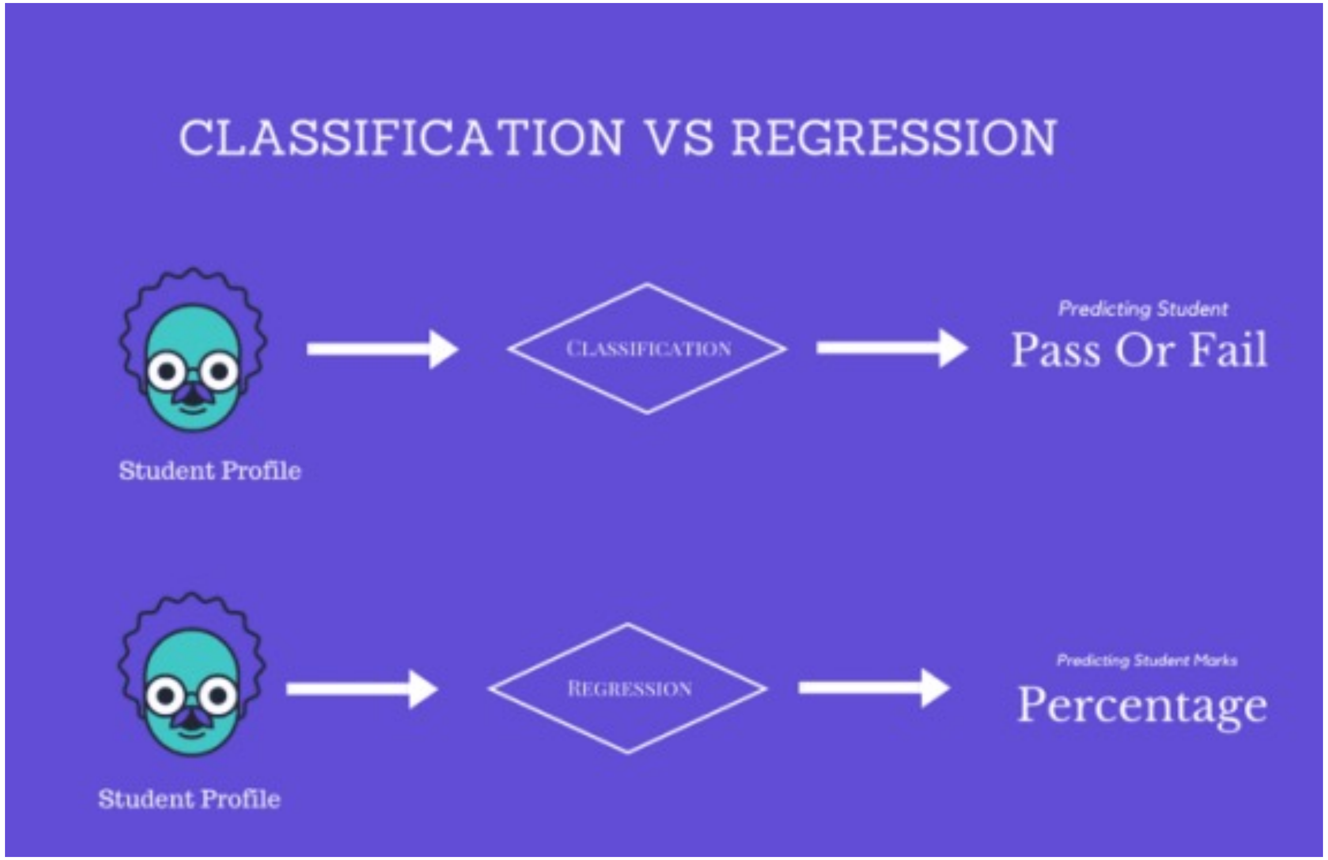
Microsoft Learn
Student Ambassadors

ML Bootcamp Session 3 : Supervised Learning – Linear Regression

Types of Supervised Learning



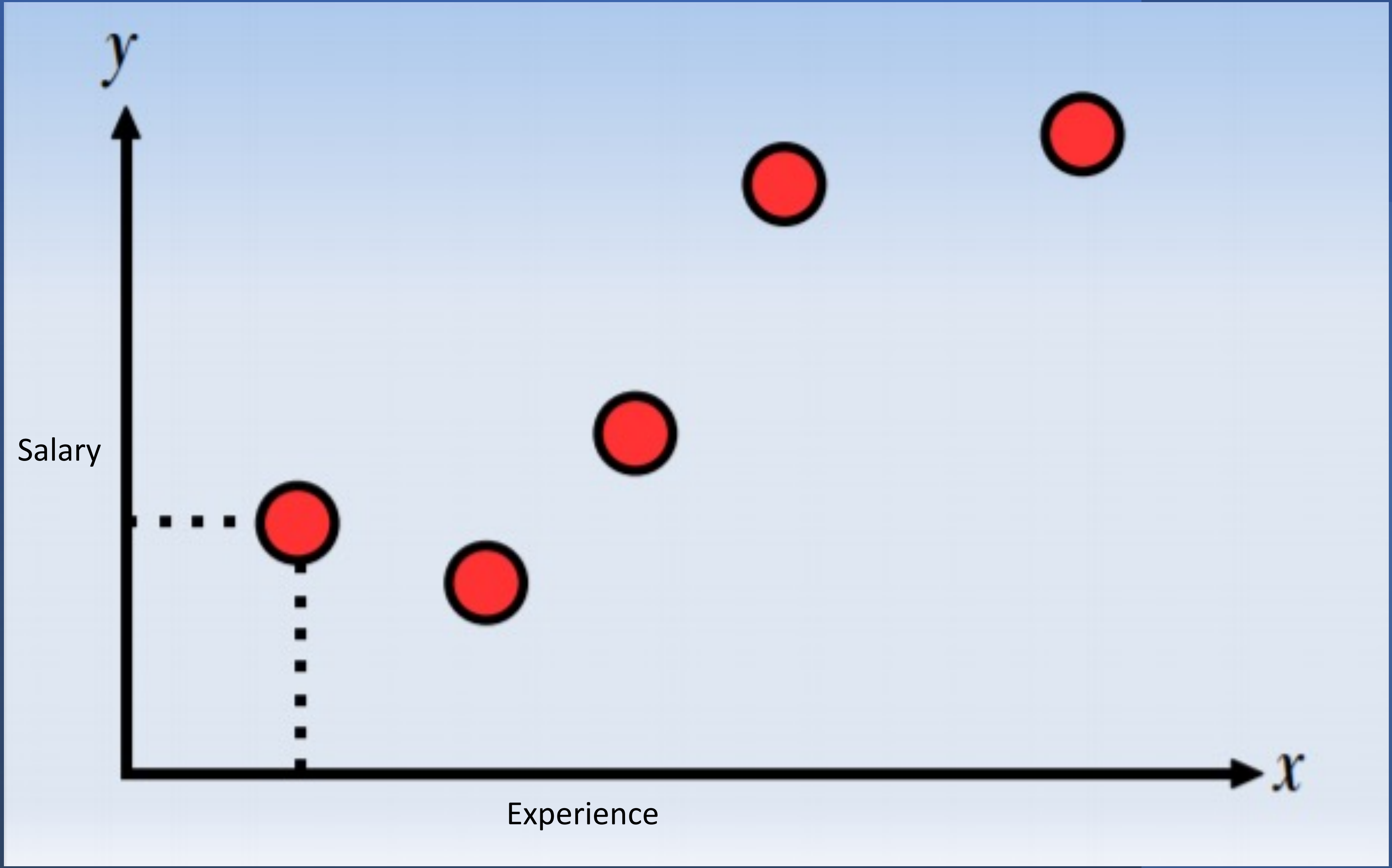
Linear Regression

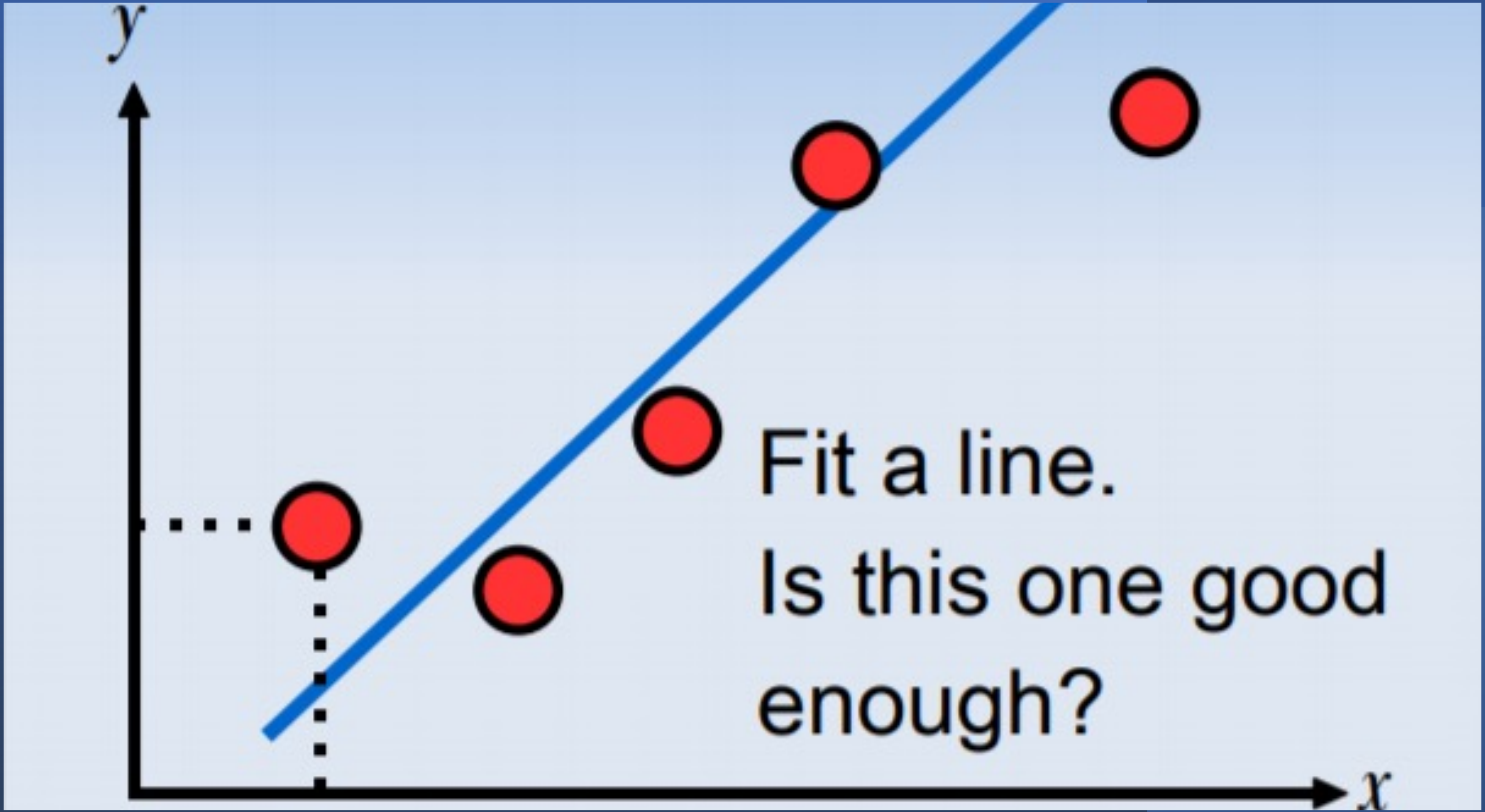
Introduction

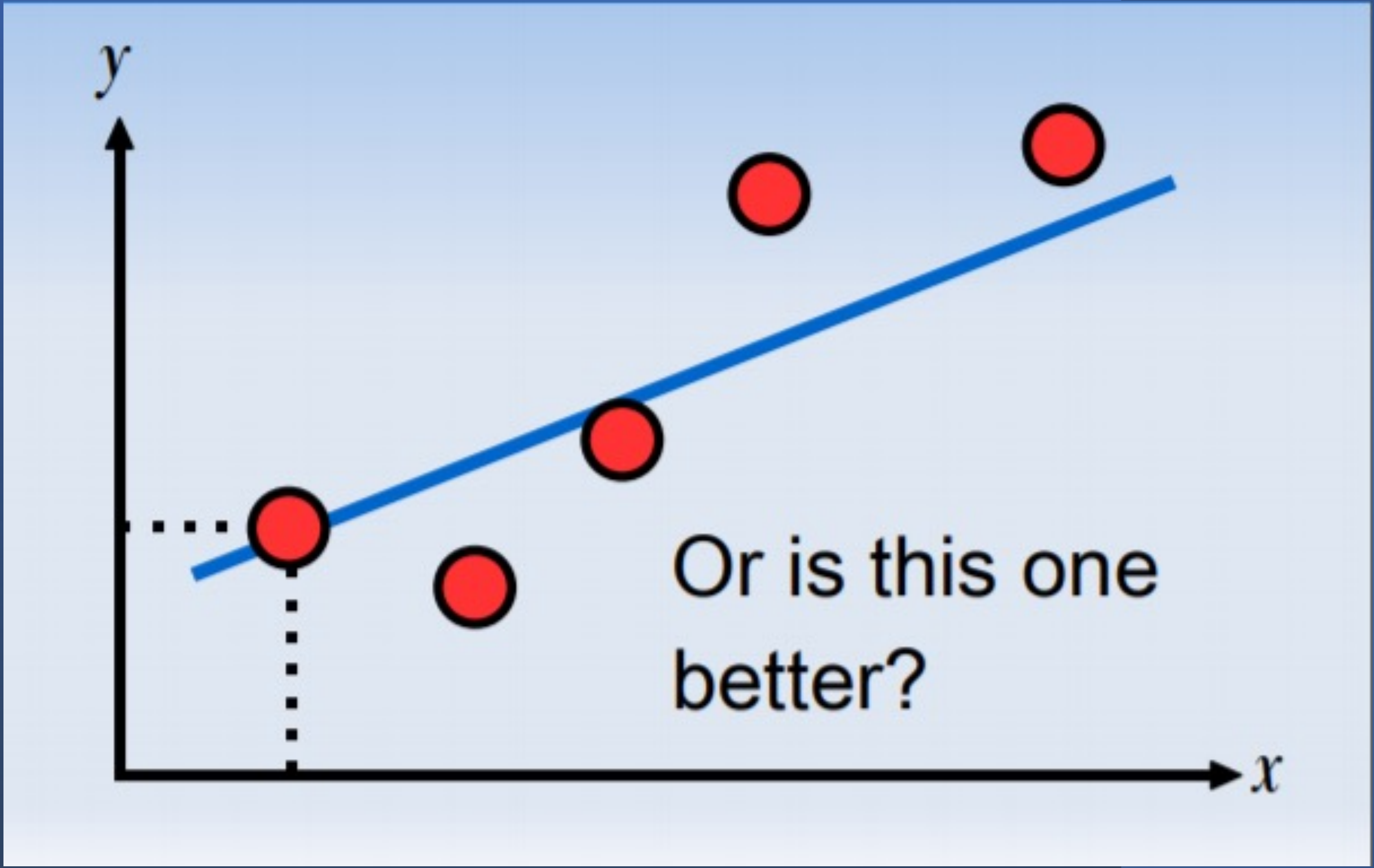
- Linear Regression is a supervised machine learning algorithm.
- Analyze the specific relationships between the two or more variables.
- This is done to gain the information about one through knowing values of the others

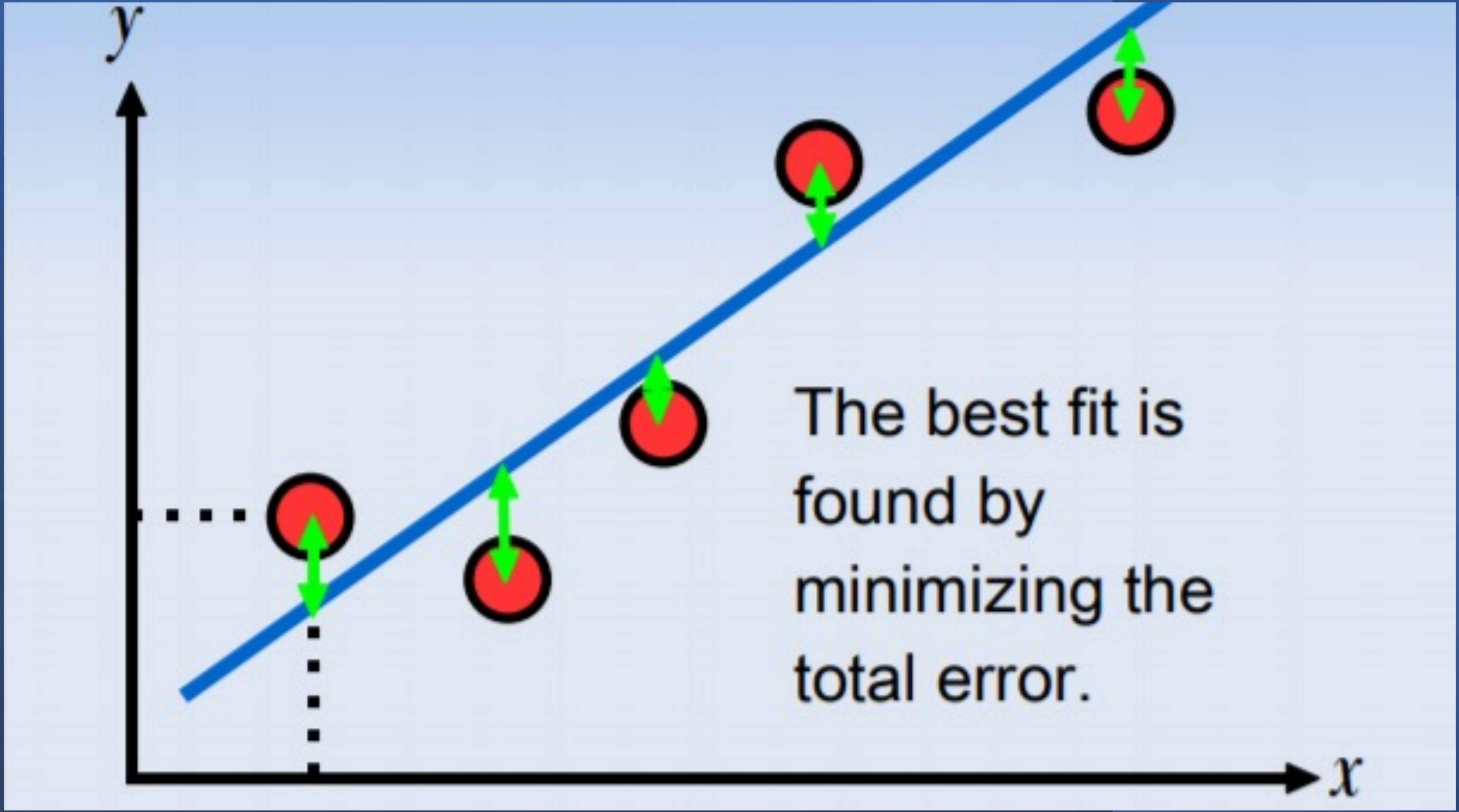
Regression

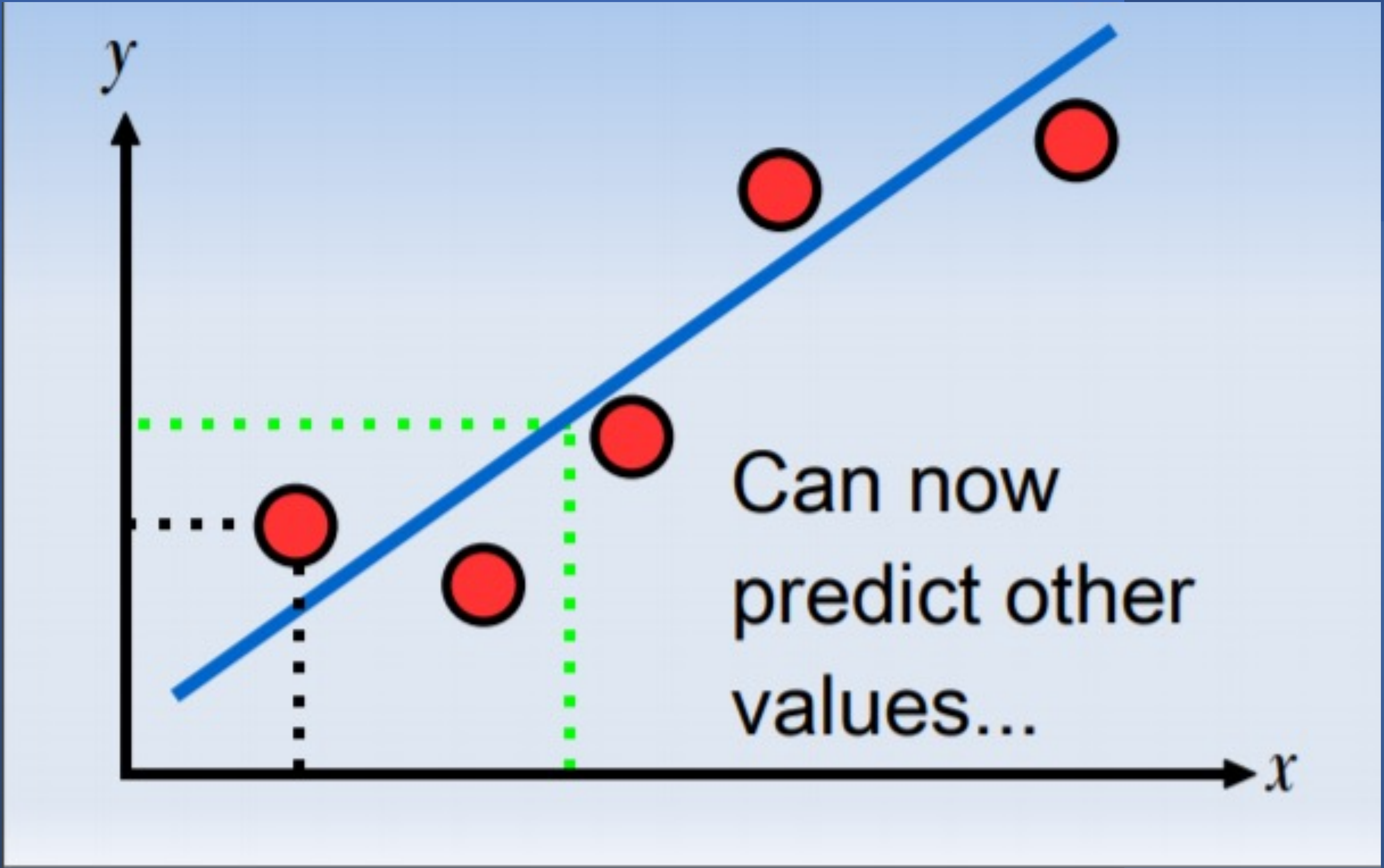
- It attempts to determine the relationship between one dependent variable (usually denoted by Y) and a series of other changing variables (known as independent variables).
- Forecast value of a dependent variable (Y) from the value of independent variables (X_1, X2,....).

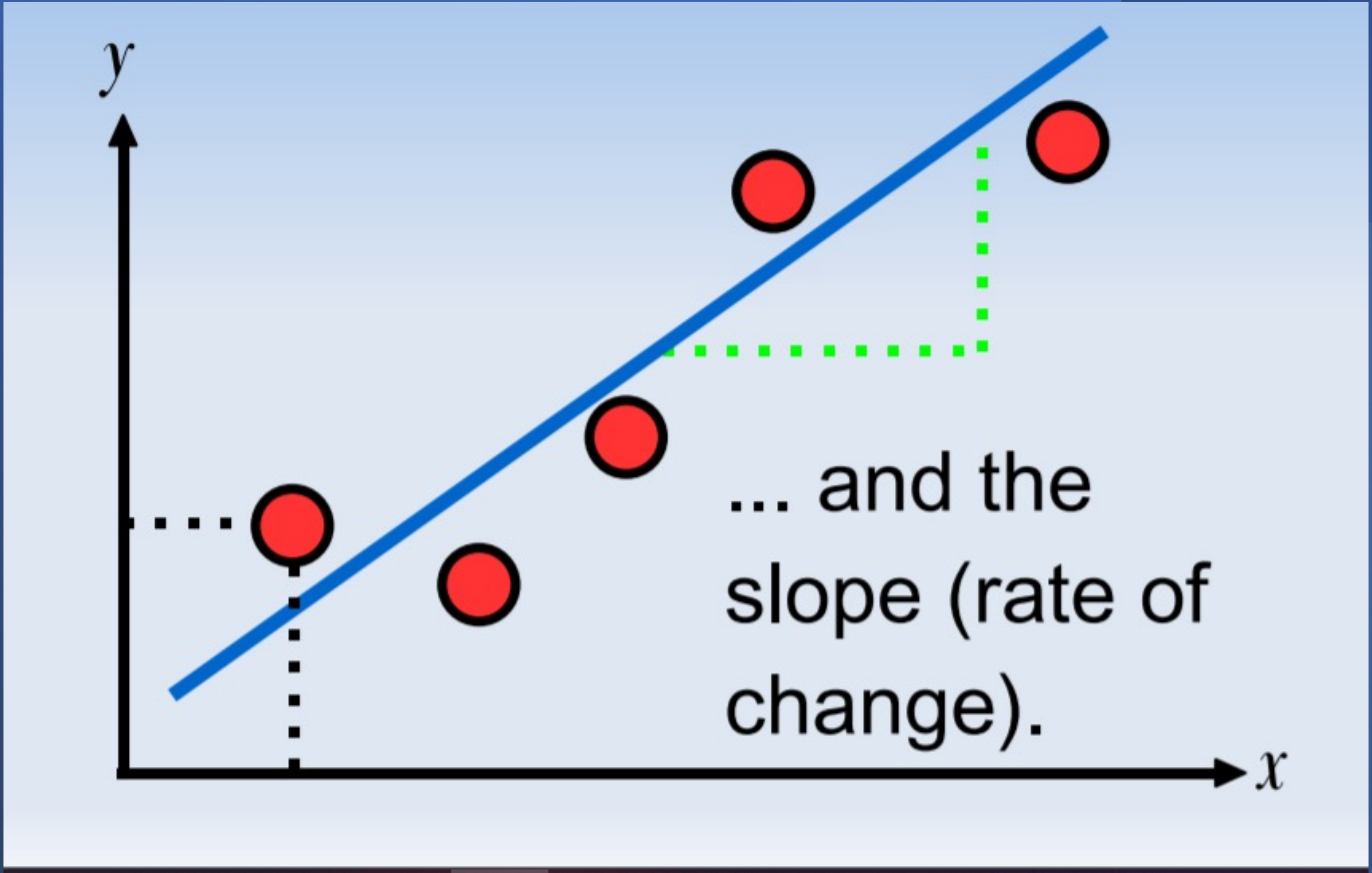

y
Salary
x
Experience


y
x
Fit a line.
Is this one good enough?


y
x
Or is this one
better?


y
x
The best fit is found by minimizing the total error.


y
x
Can now
predict other
values...


y
... and the slope (rate of change).
x

Dependent & Independent variable

Independent variables are regarded as inputs to a system and may take on different values freely.

Dependent variables are those values that change as a consequence of changes in other values in the system.

Independent variable is also called as predictor variable and it is denoted by X.

Dependent variable is also called as response variable and it is denoted by Y.

Hypothesis

- *Our goal is to learn a function h(x) which is a “good” predictor for the given training set.*

  Parameters -

  θ_0 *: Intercept ,* θ_1 *: Slope*

- *Hypothesis Function :* $h(x) = \theta_0 + x.\theta_1$

  (straight line) $Y = C + x.b$

The Perfect Model!

Your Prediction	Label
4	6
5	3
9	8
10	11

Error = (4-6) + (5-3) + (9-8) + (10-11) = (-2) + (2) + (1) + (-1) = 0   Voila!!!

Actual Error (RMSE)= $(\sqrt{[(-2)^2 + (2)^2 + (1)^2 + (-1)^2]})/4 = \sqrt{10} =$ 3.16   Awww!!

Cost Function

We can measure the accuracy of our hypothesis function by using a cost function.

Cost : $\textbf{Cost Function: } J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x) - y)^2$

This function is also called as the 'Squared Error Function' or 'Mean Squared Error' cost function.

Aim: To minimize $J(\theta_0 , \theta_1)$

Notations

No. of training examples: *m*

Input Variable : $X^{(i)}$

Output Variable: $y^{(i)}$

Training Example: $(x^{(i)}, y^{(i)})$

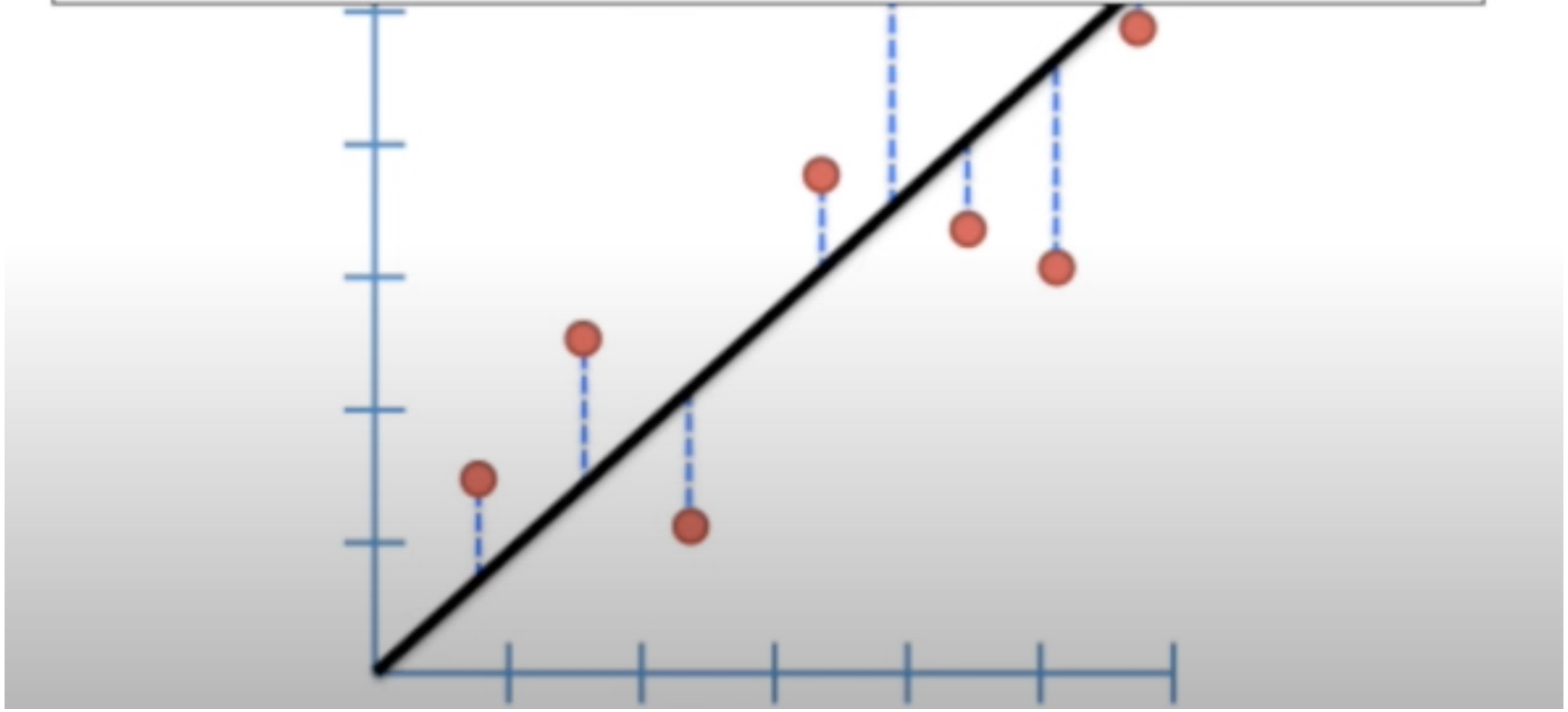
Training Set: List of m training examples $(x^{(i)}, y^{(i)})$

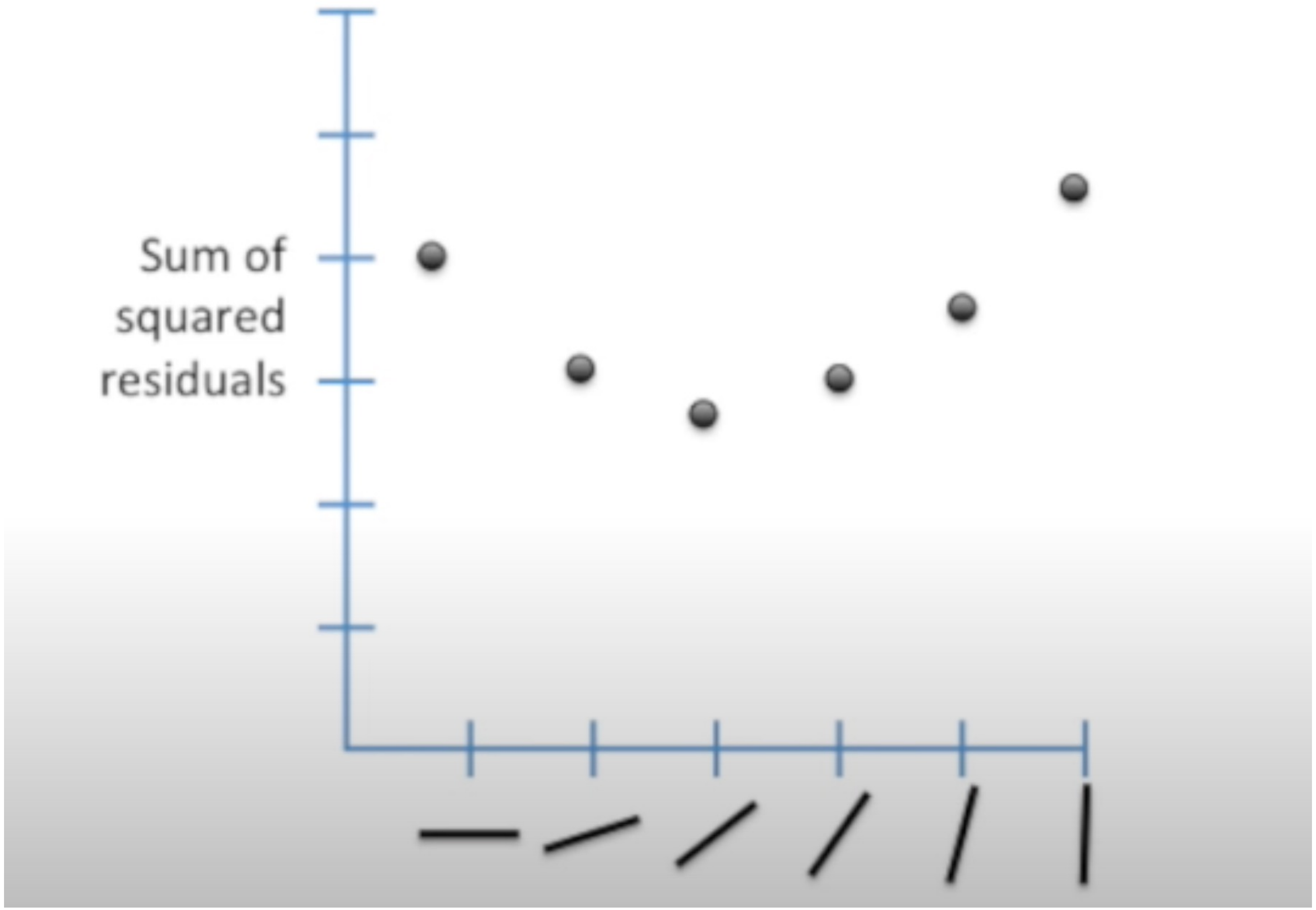
where $i = 1,2,3.......m-1, m$ *where* X ϵ R, Y ϵ R

Hypothesis : h:X→Y ; h(x)

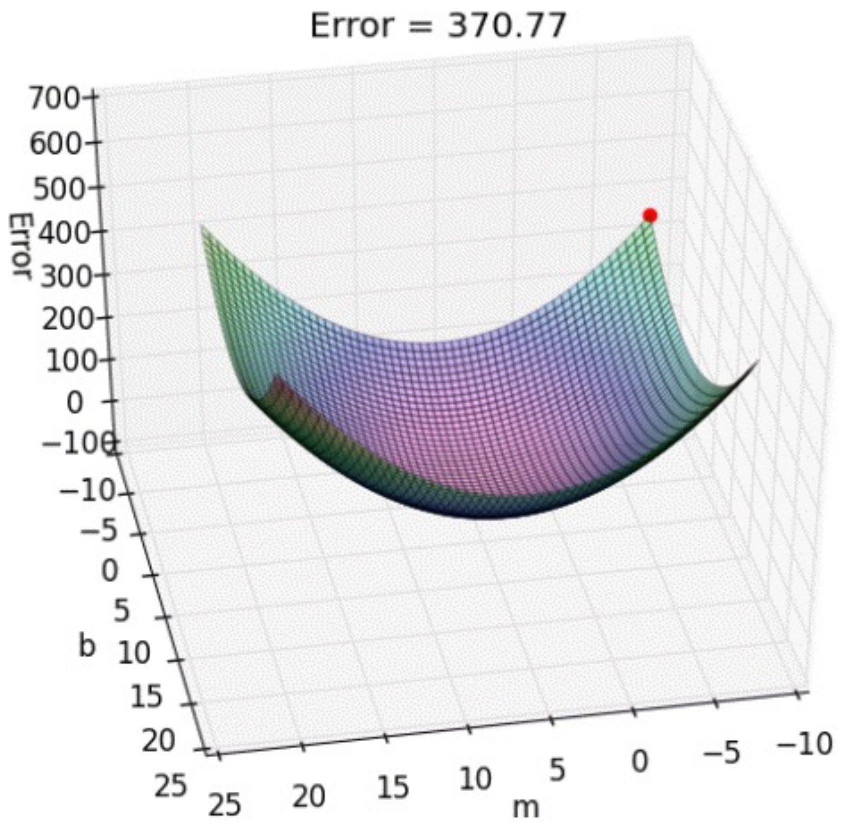
Sum of squared residuals = $((a^*x_1 + b) - y_1)^2 + ((a^*x_2 + b) - y_2)^2 + \ldots$

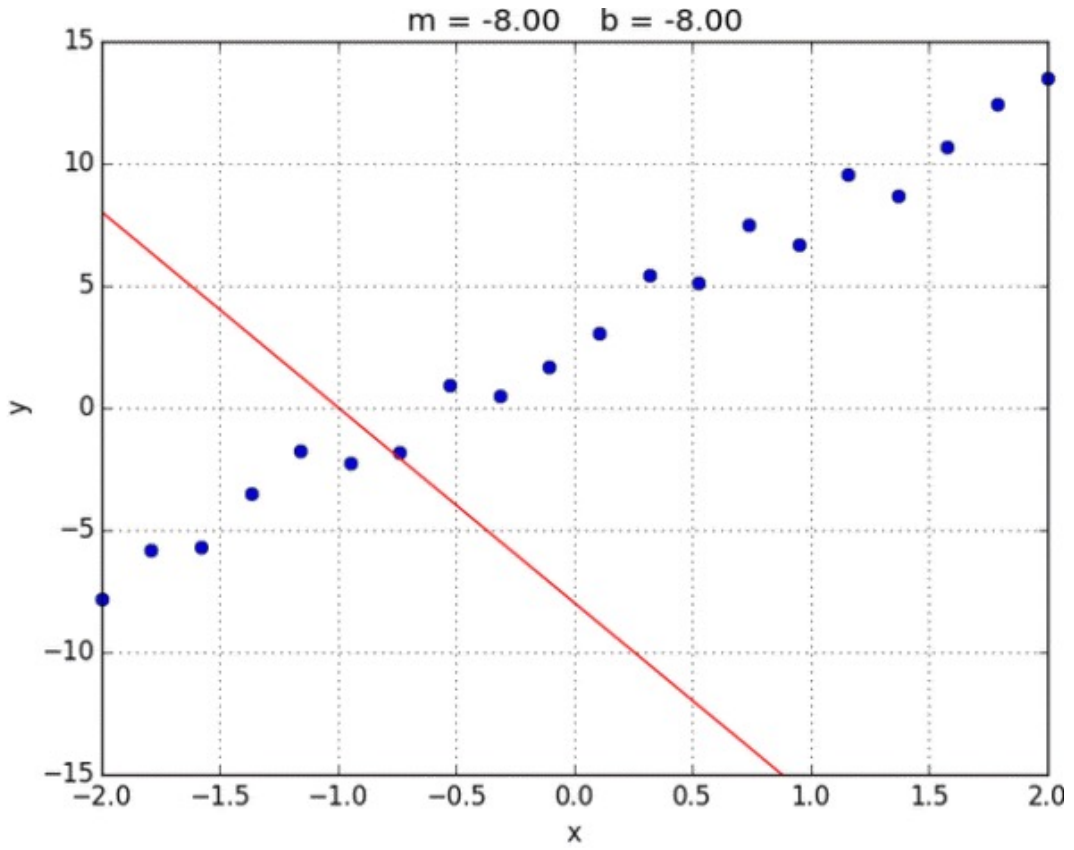
Since we want the line that will give us the smallest sum of squares, this method for finding the best values for "*a*" and "*b*" is called "Least Squares".




Sum of
squared
residuals

Gradient Descent





Optimization

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

Where j = 0,1 denotes the feature index number.

$$\theta_1 = \theta_1 - \alpha \cdot \frac{\partial}{\partial \theta_1} \left(\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x) - y)^2 \right)$$

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

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

$$\theta_1 = \theta_1 - \frac{2\alpha}{2m} \cdot ((\theta_0 + \theta_1 x) - y) \cdot (x)$$

$$\theta_1 = \theta_1 - \frac{\alpha}{m} \cdot h_\theta(x) \cdot x$$

Linear Regression with multiple variables

Where 'n' is the No. of features

$$h(x) = \theta_0 + x_1.\theta_1 + x_2.\theta_2 + \ldots\ldots\ldots + x_n.\theta_n$$

OR

$$h(x) = \theta^T X$$

Gradient Descent:

$$\theta_j = \theta_j - \frac{\alpha}{m} \cdot h_\theta(x) \cdot x_j$$