ED5340 - Data Science: Theory and Practise

L18 - Linear Regression: Multivariate

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

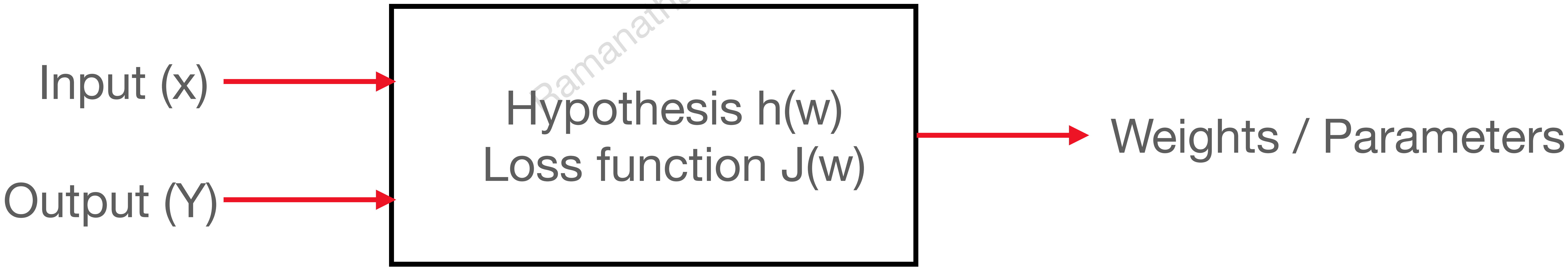
Linear Regression

Univariate

- Ground truth data - Input feature / output $(\mathbf{x}, \mathbf{y})$ are the knowns
- Use a model / hypothesis as $h(w)$
- Develop an error / cost / loss function $J(w) = J(\mathbf{y}, \bar{\mathbf{y}}) = J(\mathbf{y}, h(w))$
- The weights are identified by
    - min $J(w)$
- Essentially, ML problem is now reduced to an optimization problem.
- Weights are identified using Optimization.

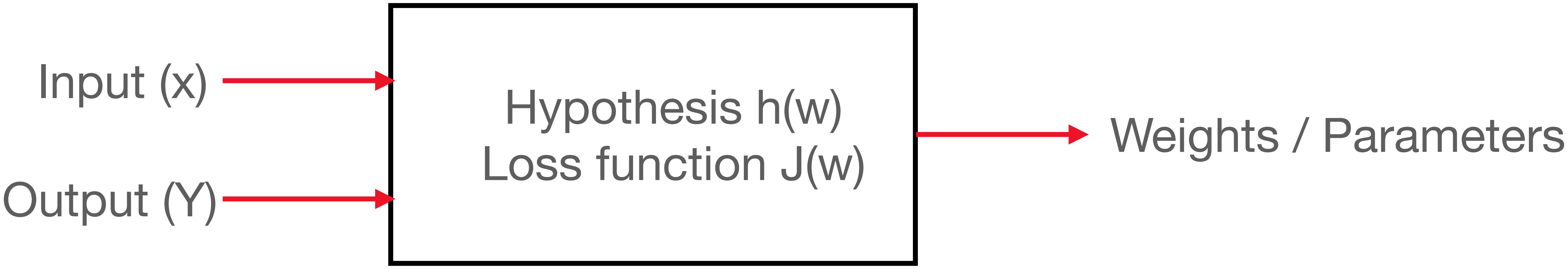
Linear Regression
Univariate case

- Ground truth data - Input feature / output $(\mathbf{x}, \mathbf{y})$ are the knowns
- Use a model / hypothesis as $h(w)$ and cost function $J(w)$
-



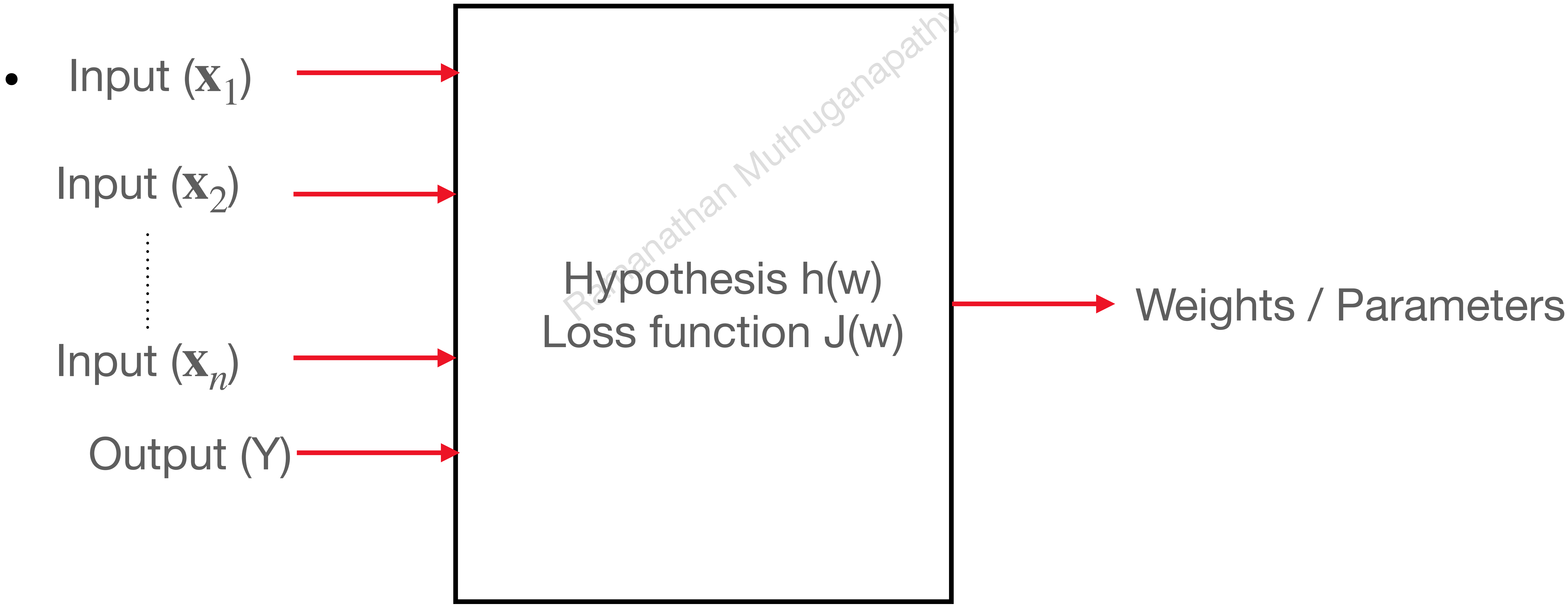
Linear Regression
Multivariate case

- Ground truth data - Input feature / output $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots \mathbf{x}_n, \mathbf{y})$ are the knowns - n features
- $\mathbf{x}_1$ - Size, $\mathbf{x}_2$ - BA, $\mathbf{x}_3$ - Distance to school, $\mathbf{x}_4$ - To hospital, $\mathbf{x}_5$ - maintenance etc..
- Use a model / hypothesis as $h(w)$ and cost function $J(w)$
-



Linear Regression

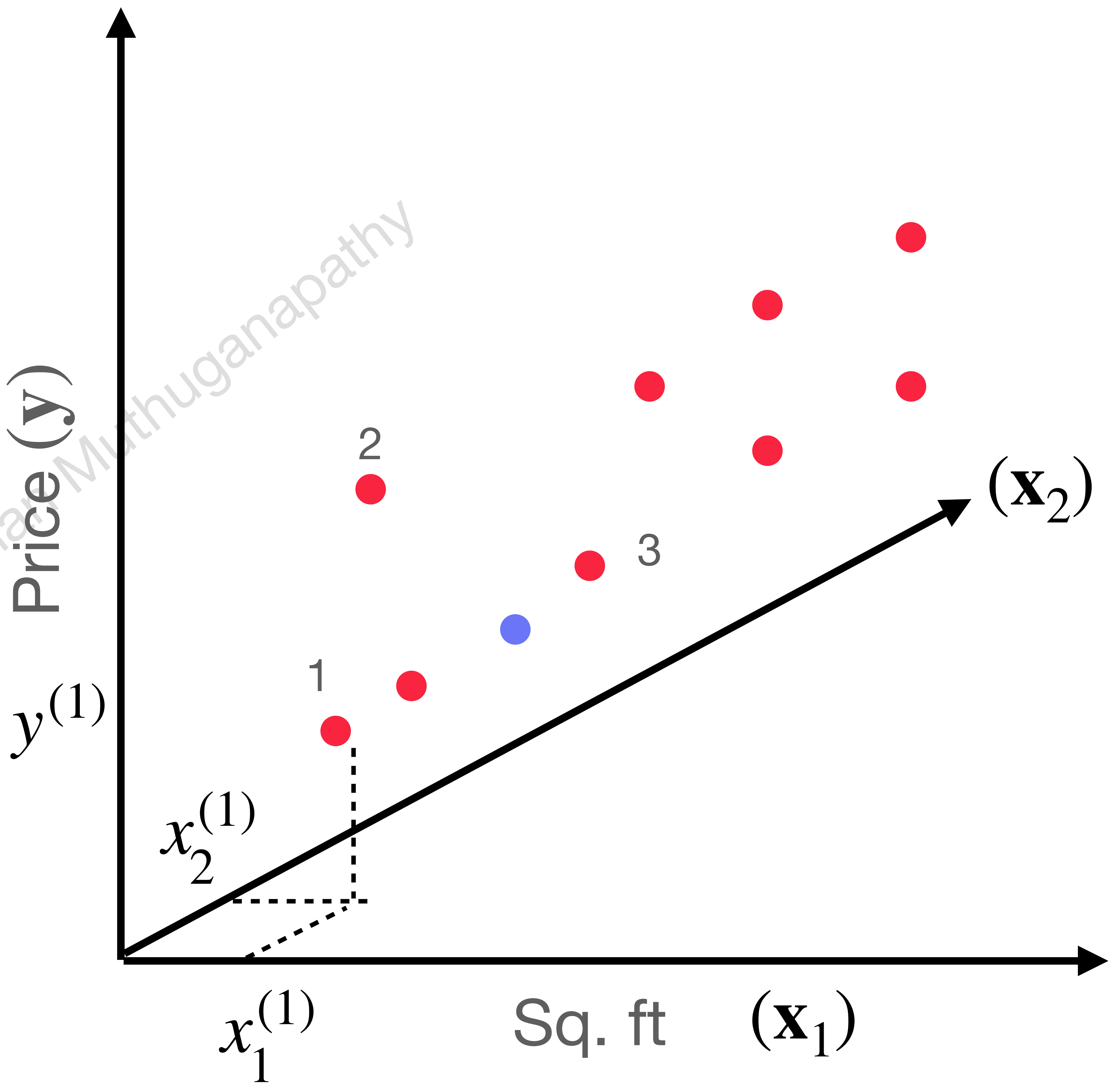
Multivariate case (n features)



Linear Regression

Supervised Leaning

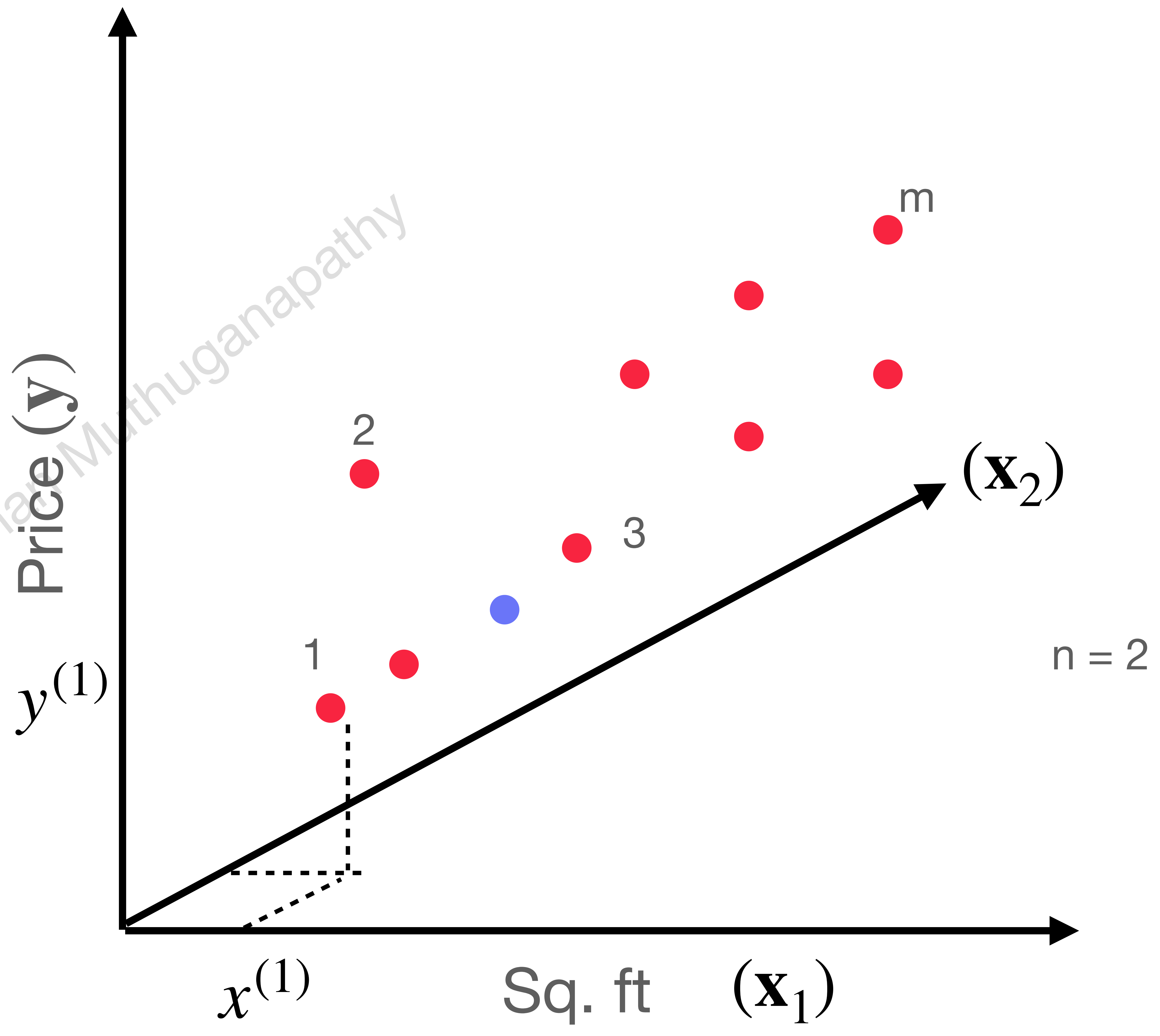
- Each feature will have m training samples
- $\mathbf{x}_1 = (x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, \ldots. x_1^{(m)})$
- $\mathbf{x}_2 = (x_2^{(1)}, x_2^{(2)}, x_2^{(3)}, \ldots. x_2^{(m)})$
- $\mathbf{x}_n = (x_n^{(1)}, x_n^{(2)}, x_n^{(3)}, \ldots. x_n^{(m)})$
- $\mathbf{y} = (y^{(1)}, y^{(2)}, y^{(3)}, \ldots. y^{(m)})$



Linear Regression

n dimensional coordinates

- Each training sample will come from n features (n dimensional coordinates)
- $(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \ldots . x_n^{(1)})$
- $(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \ldots . x_n^{(2)})$
- $(x_1^{(m)}, x_2^{(m)}, x_3^{(m)}, \ldots . x_n^{(m)})$
- $\mathbf{y} = (y^{(1)}, y^{(2)}, y^{(3)}, \ldots . y^{(m)})$



Sample data

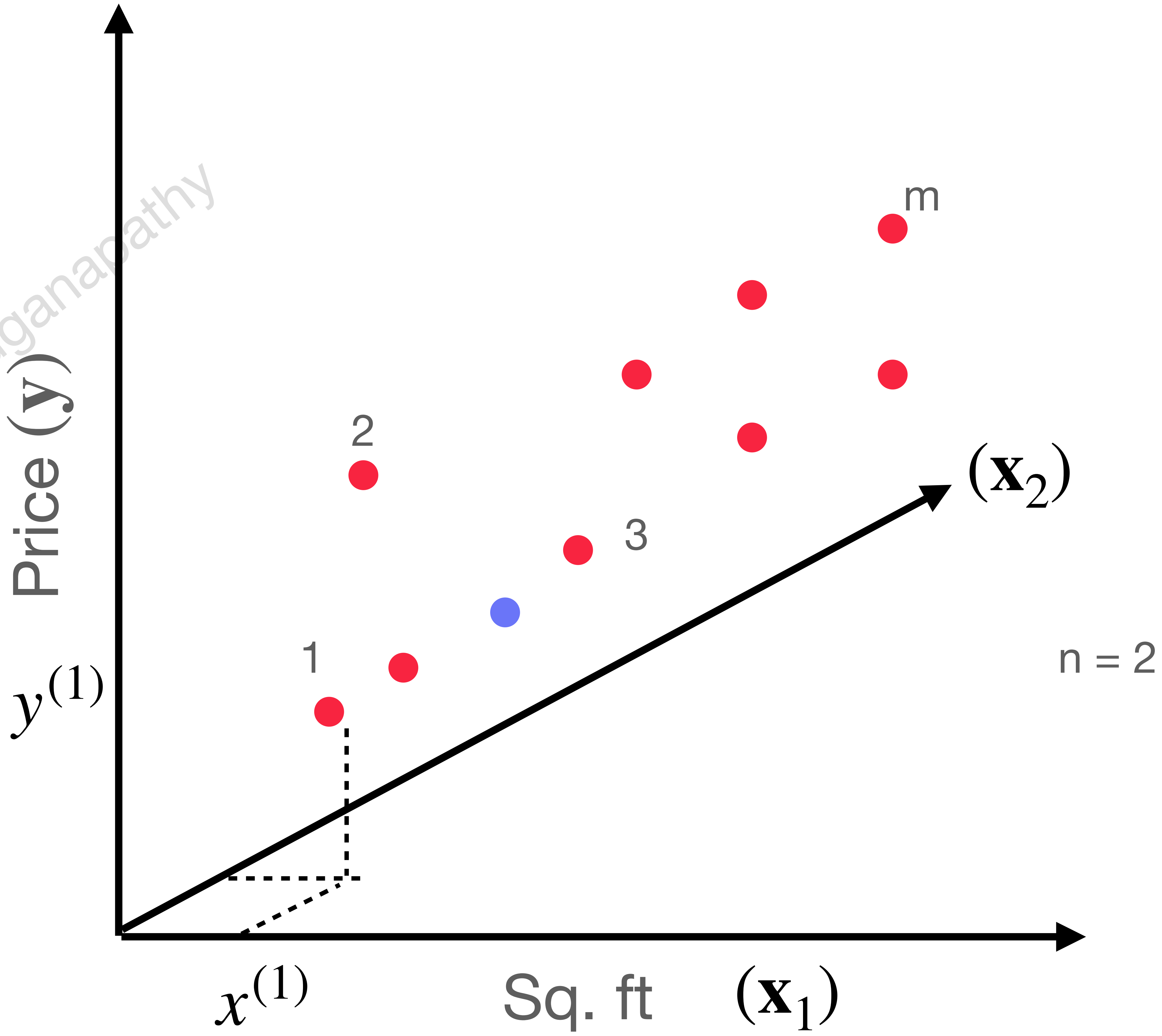
m samples each with n features

sample number	Size $\mathbf{x}_1$	BA $\mathbf{x}_2$	Maintanence $\mathbf{x}_3$	$y^{(i)}$ Price
1 $(x_1^{(1)}, x_2^{(1)}, x_3^{(1)})$	1000	800	2.5	70
2 $(x_1^{(2)}, x_2^{(2)}, x_3^{(2)})$	1500	1250	3.0	85
m $(x_1^{(m)}, x_2^{(m)}, x_3^{(m)})$	800	400	3.5	55

Linear Regression

Goal: Approximation that fits the data

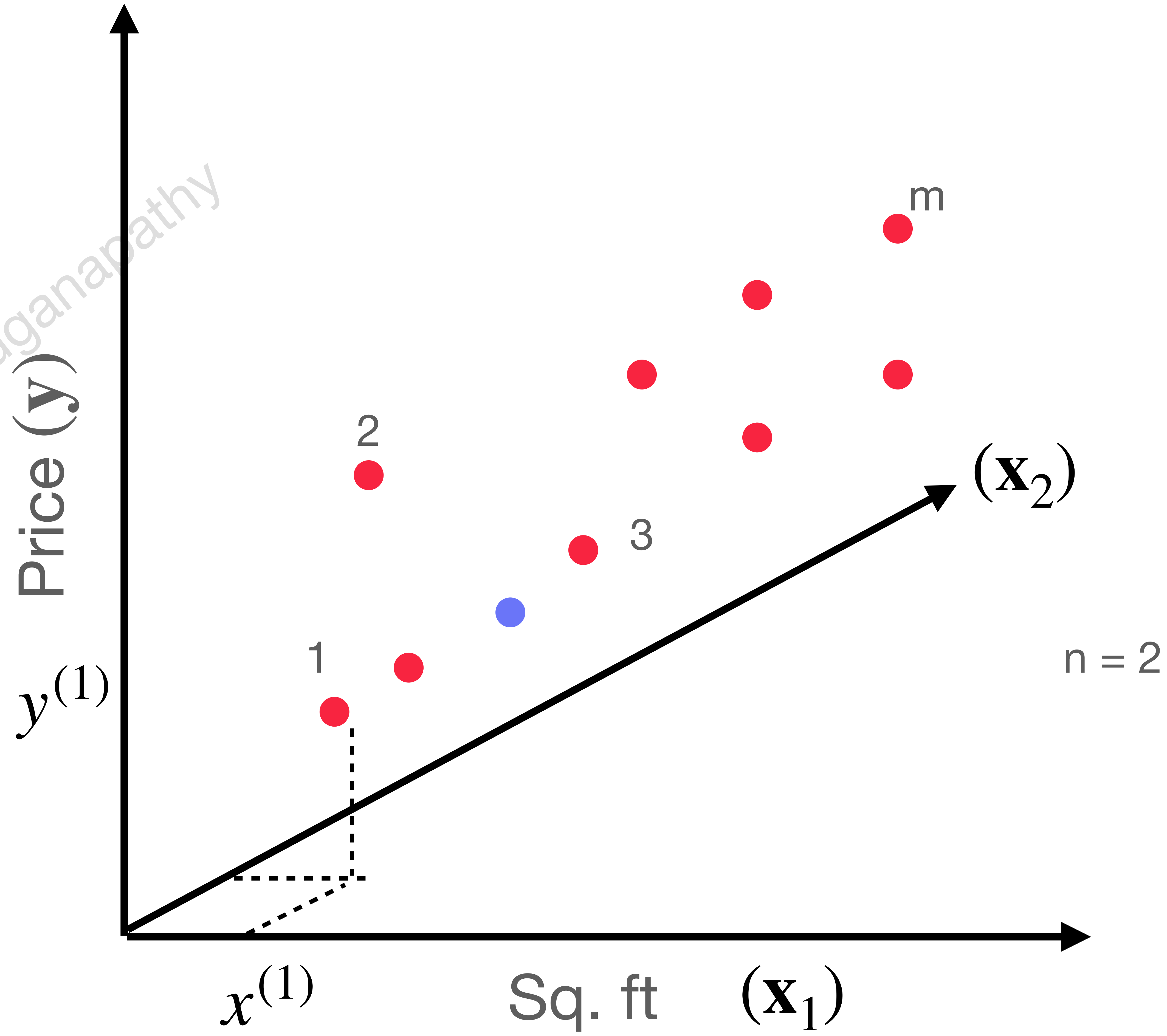
- Look at the data, hyperplane fit is carried out!
- $h_w(x) = w_0 + w_1x_1 + w_1x_2 + w_3x_3$
- Goal: Determine weights $(w_0, w_1, w_2, \ldots\ldots, w_n)$



Linear Regression

Generalized form

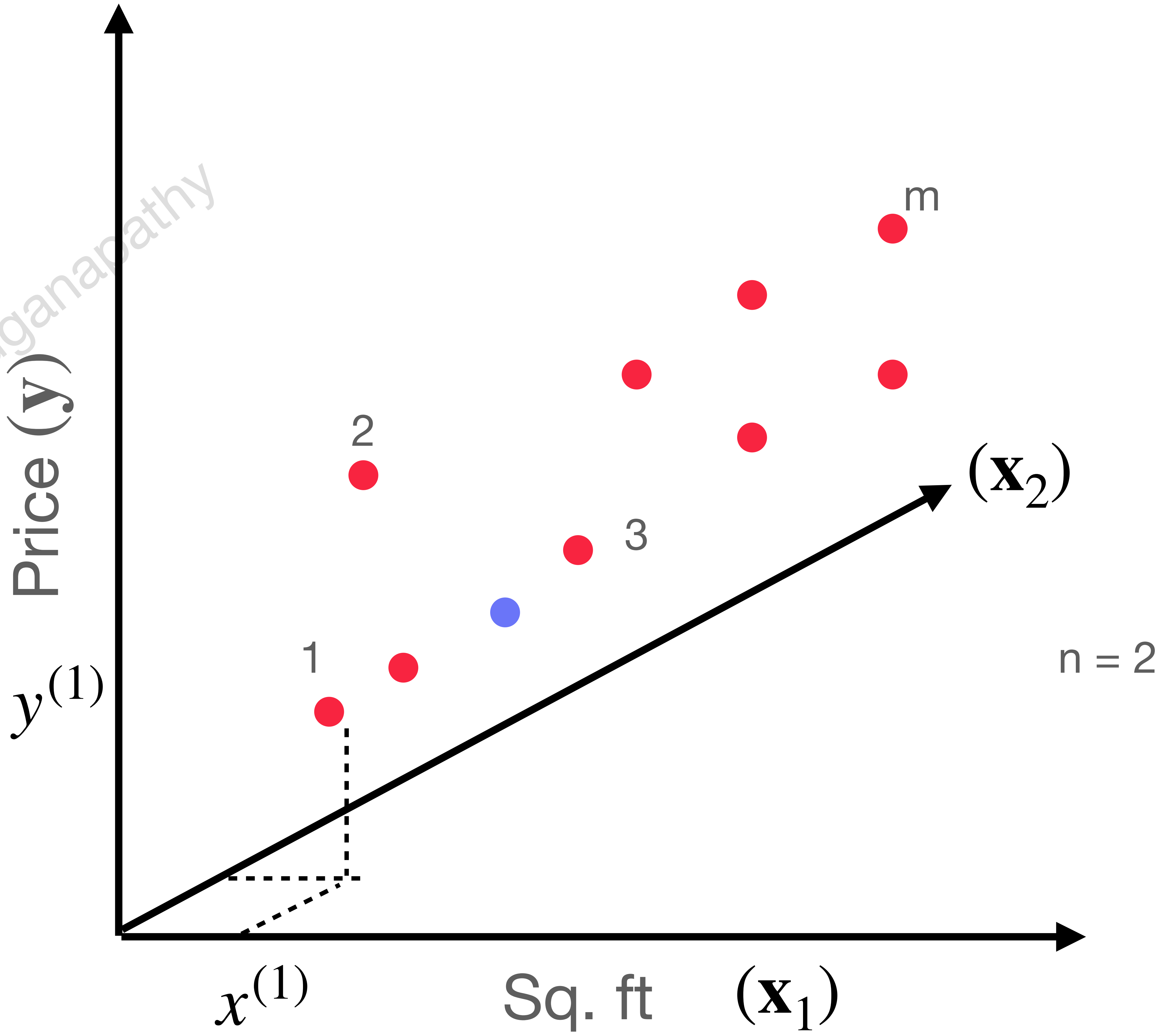
- Look at the data, hyperplane fit is carried out!
- $h_w(x) = w_0x_0 + w_1x_1 + w_1x_2 + w_3x_3$
- $x_0 = 1$
- $h_w(x) = \mathbf{w}^T\mathbf{x}$ (or $= \mathbf{x}^T\mathbf{w}$)



Linear Regression

Generalized form

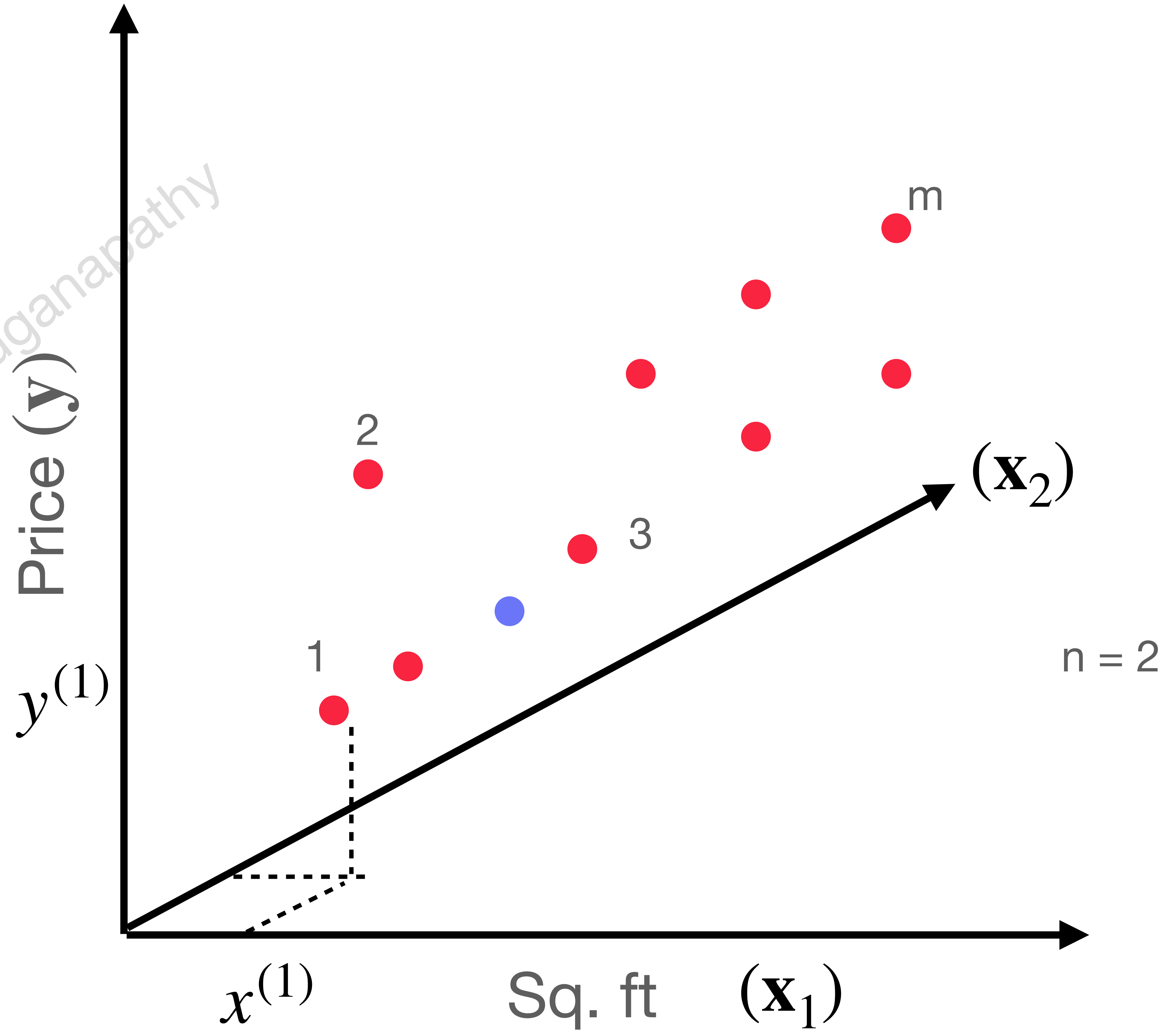
- $h_w(x) = \mathbf{w}^T\mathbf{x}$ (or $= \mathbf{x}^T\mathbf{w}$)
- $\mathbf{w}^T = [w_0 \quad w_1 \quad w_2 \quad w_3]$
- $\mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}$



Linear Regression

Generalized form

- $$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ x_1^{(1)} & x_1^{(2)} & x_1^{(3)} \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} \\ x_3^{(1)} & x_3^{(2)} & x_3^{(3)} \end{bmatrix}$$



Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Linear Regression
Predicted values

- $\mathbf{y} = (y^{(1)}, y^{(2)}, y^{(3)}, \ldots . y^{(m)})$
- $\bar{\mathbf{y}} = (\bar{y}^{(1)}, \bar{y}^{(2)}, \bar{y}^{(3)}, \ldots . \bar{y}^{(m)})$
- $\bar{y}_j^{(i)} = h_w(x_j^{(i)}) = w_0 x_j^{(i)} + w_1 x_j^{(i)} + w_2 x_j^{(i)} + w_3 x_j^{(i)}$
- Goal: Determine weights (w_0, w_1, w_2, w_3)
- i - 1 to m (training samples), j = 1 to n (features)

Linear Regression

Minimize the cost function

- $J(w) = \sum_{i=1}^{m} \frac{1}{2m}(w_0 x_j^{(i)} + w_1 x_j^{(i)} + w_2 x_j^{(i)} + w_3 x_j^{(i)} - y^{(i)})^2$
- min $J(w)$

Linear Regression

Gradient descent

- Find $\nabla J(w_0, w_1, w_2, w_3) = \left(\frac{\partial J}{\partial w_0}, \frac{\partial J}{\partial w_1}, \ldots\right)$
- $\frac{\partial J}{\partial w_0} = \frac{1}{m}\sum_{i=1}^{m}(w_0x_0^{(i)} + w_1x_1^{(i)} + w_2x_2^{(i)} + w_3x_3^{(i)} - y^{(i)})x_0^{(i)}$
- $\frac{\partial J}{\partial w_1} = \frac{1}{m}\sum_{i=1}^{m}(w_0x_0^{(i)} + w_1x_1^{(i)} + w_2x_2^{(i)} + w_3x_3^{(i)} - y^{(i)})x_1^{(i)}$
- $\frac{\partial J}{\partial w_2} = \frac{1}{m}\sum_{i=1}^{m}(w_0x_0^{(i)} + w_1x_1^{(i)} + w_2x_2^{(i)} + w_3x_3^{(i)} - y^{(i)})x_2^{(i)}$
- $\frac{\partial J}{\partial w_3} = \frac{1}{m}\sum_{i=1}^{m}(w_0x_0^{(i)} + w_1x_1^{(i)} + w_2x_2^{(i)} + w_3x_3^{(i)} - y^{(i)})x_3^{(i)}$

Linear Regression

Gradient descent update

- $\dfrac{\partial J}{\partial w_j} = \dfrac{1}{m}\sum_{i=1}^{m}(w_0x_0^{(i)} + w_1x_1^{(i)} + w_2x_2^{(i)} + w_3x_3^{(i)} - y^{(i)})x_j^{(i)}$
- $w_j^{k+1} = w_j^k - \alpha_k \dfrac{\partial J}{\partial w_j}$

Linear Regression

Gradient descent update

- $\dfrac{\partial J}{\partial w_j} = \dfrac{1}{m}\sum_{i=1}^{m}(h_w(x) - y^{(i)})x_j^{(i)}$
- $w_j^{k+1} = w_j^k - \alpha_k \dfrac{\partial J}{\partial w_j}$

Feature scaling

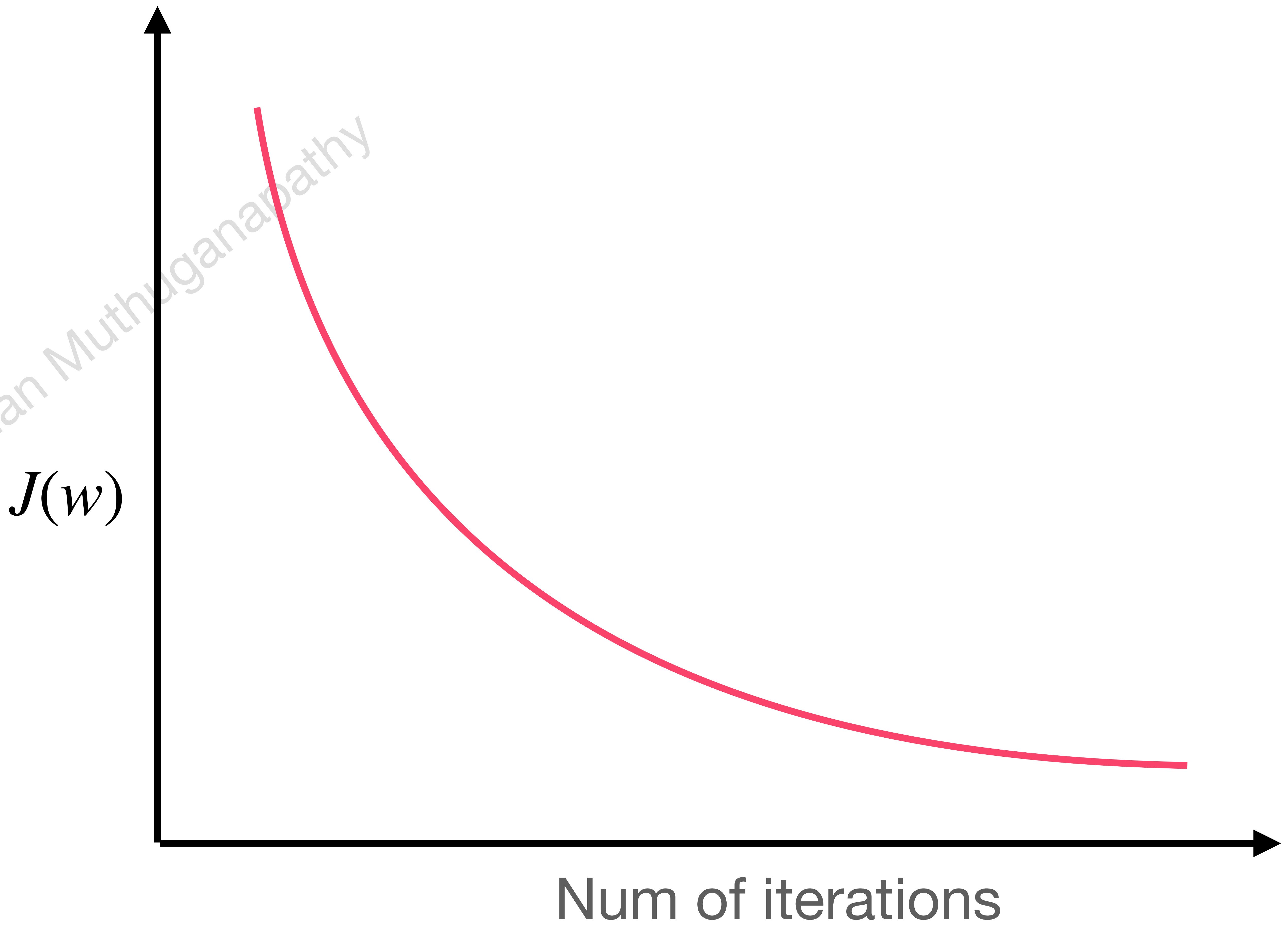
Normalising the features

- Features are on a similar scale
- Normalise the features (0 to 1)
- Normalise the features (-1 to 1) - around that range
- Normalising over mean

S. Gradient Descent

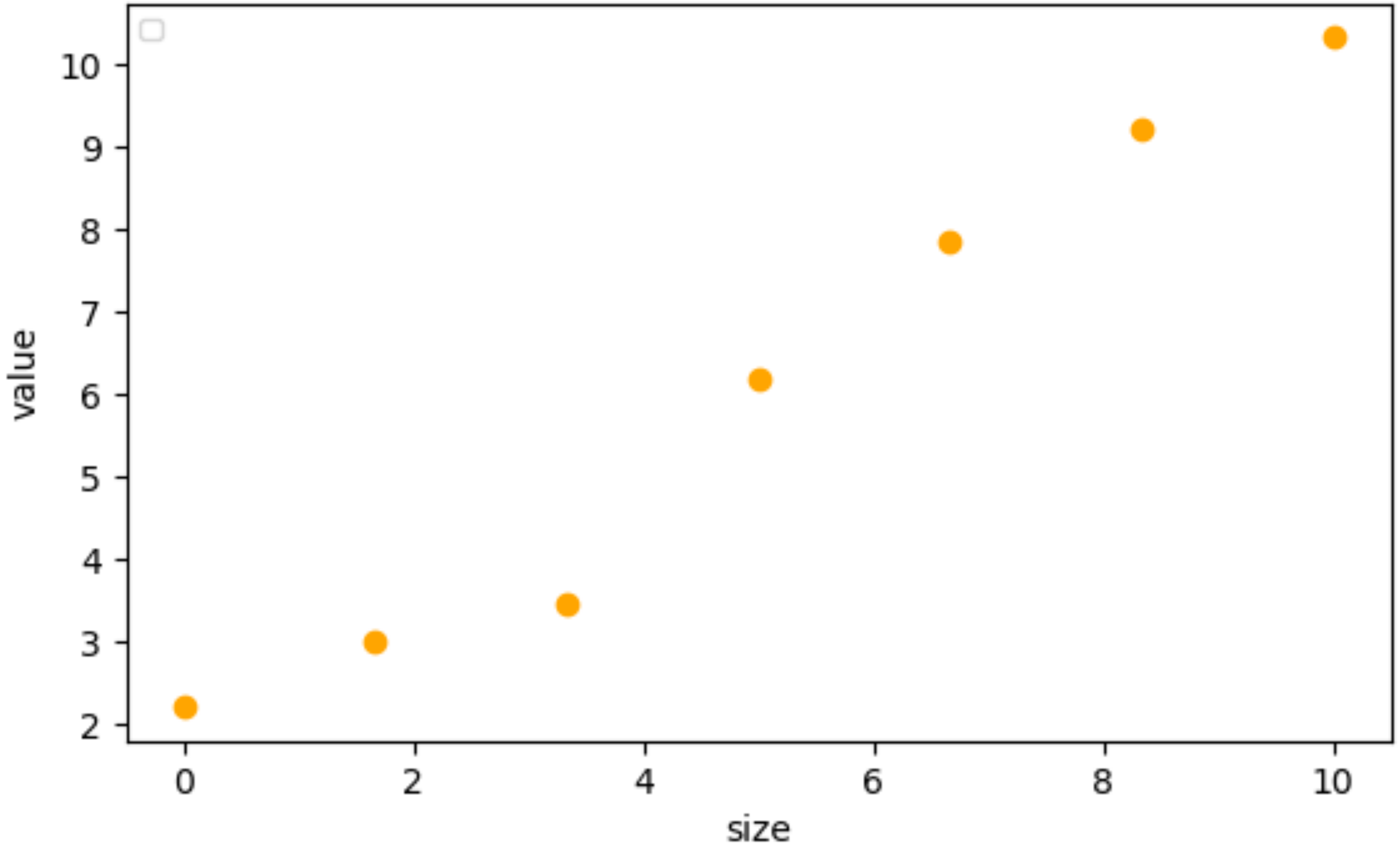
Debugging

- Plot k (iteration) vs J (Cost)
- α_k too small, slow convergence
- α_k too large, could also be another issue
- Use line search to find α_k



Polynomial Regression

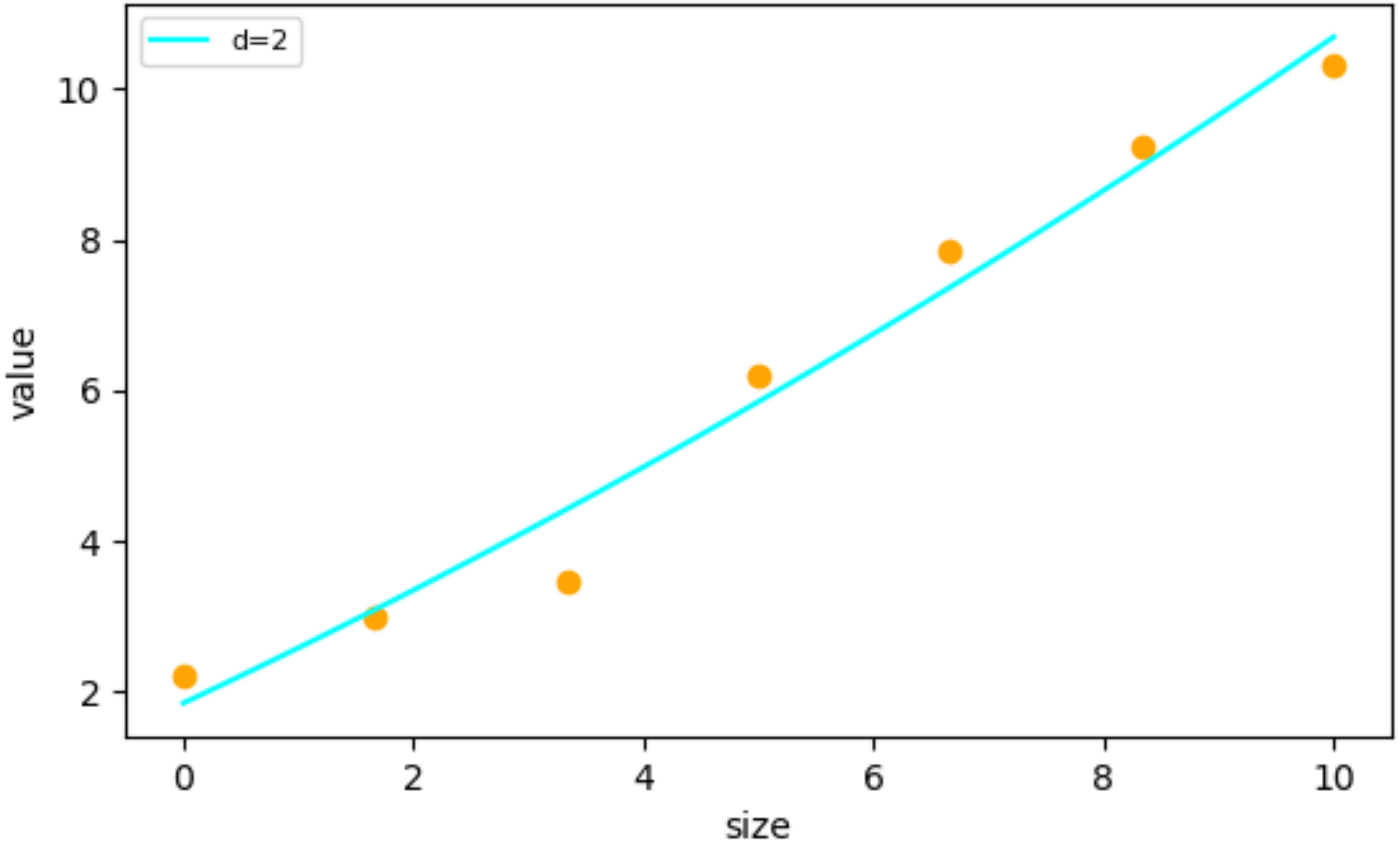
Polynomial Regression



- Imagine the data points

Polynomial Regression

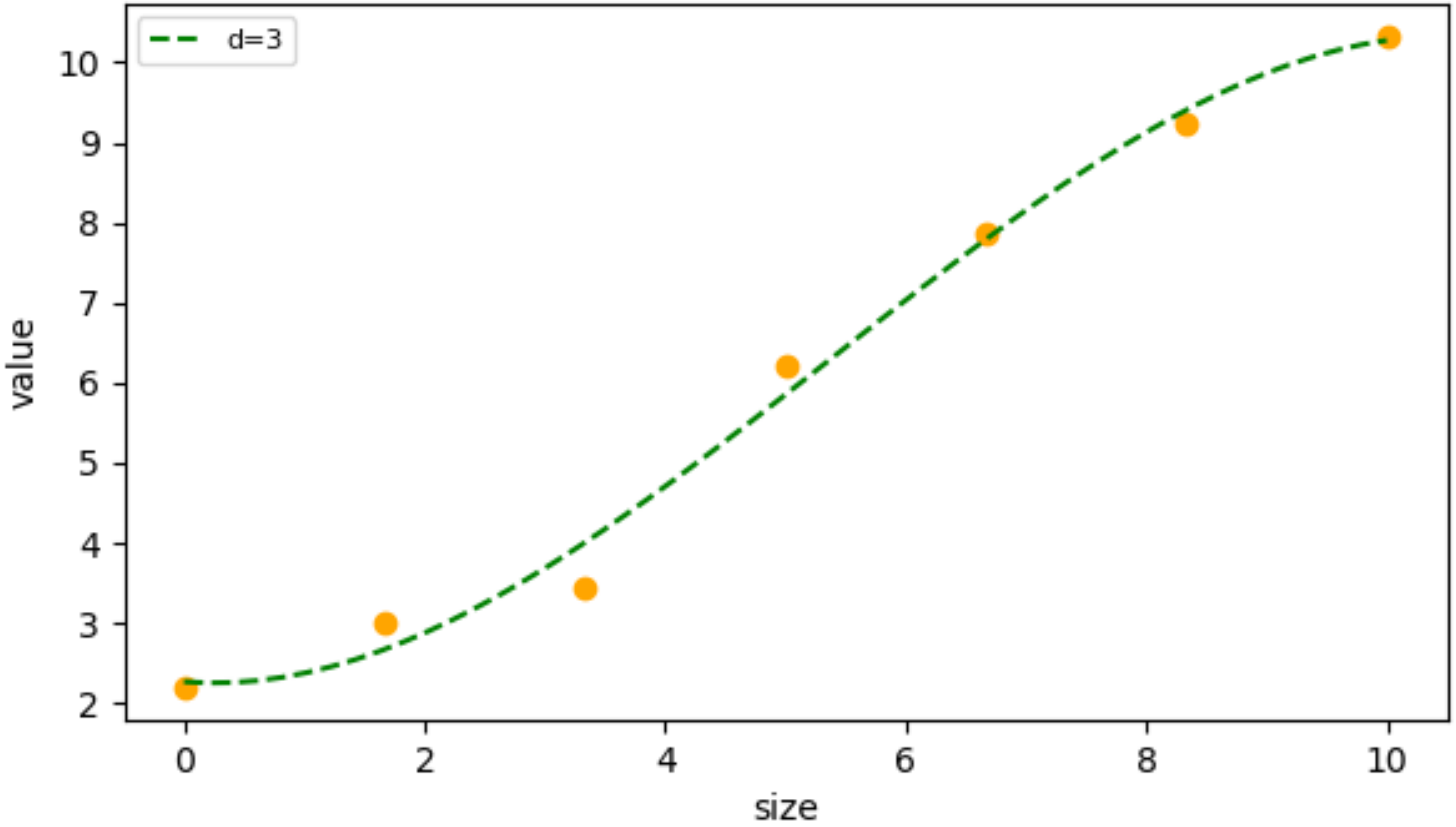
d = 2



- $h_w(x) = w_0 + w_1 x + w_2 x^2$

Polynomial Regression

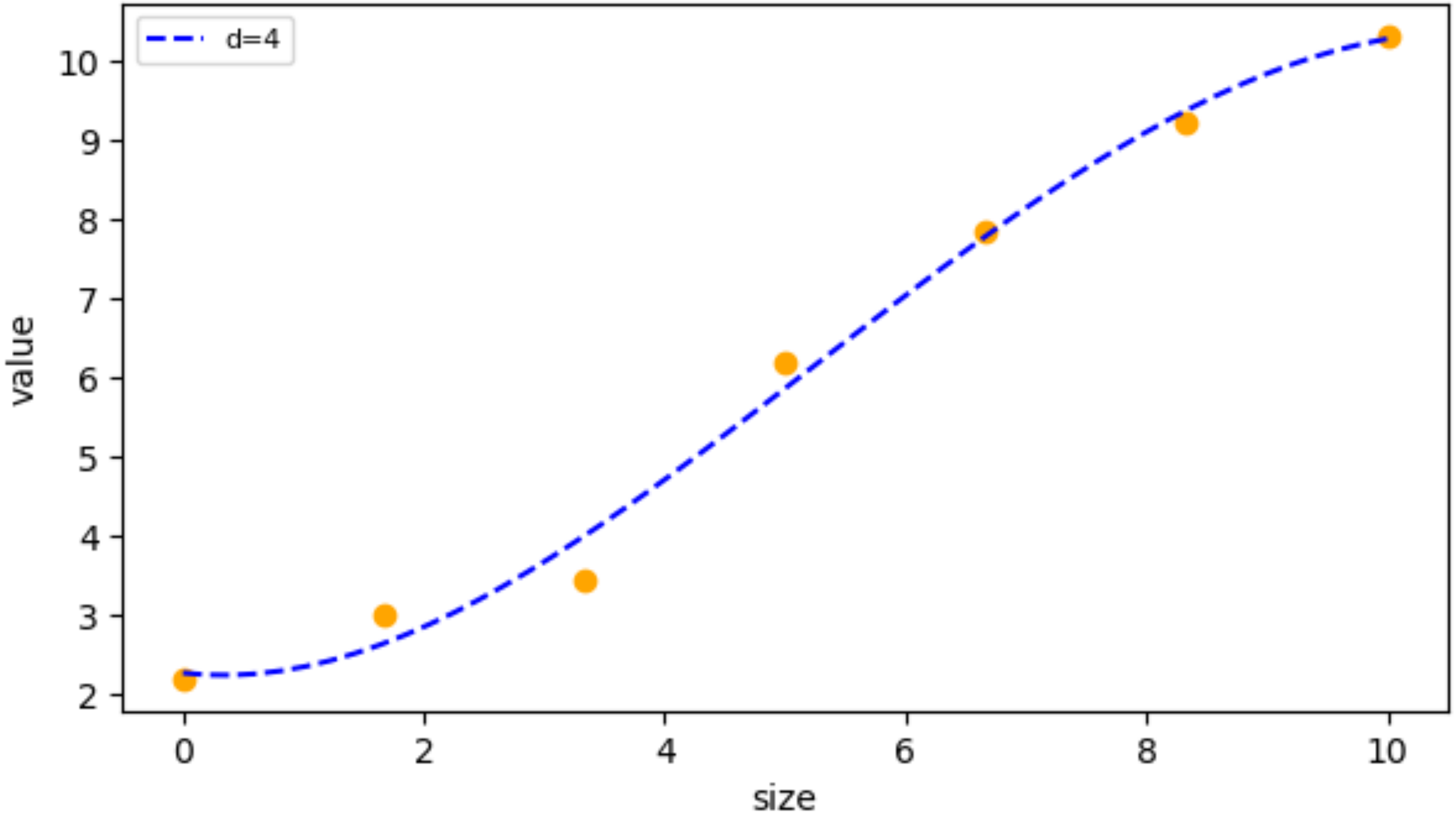
d = 3



- $h_w(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$

Polynomial Regression

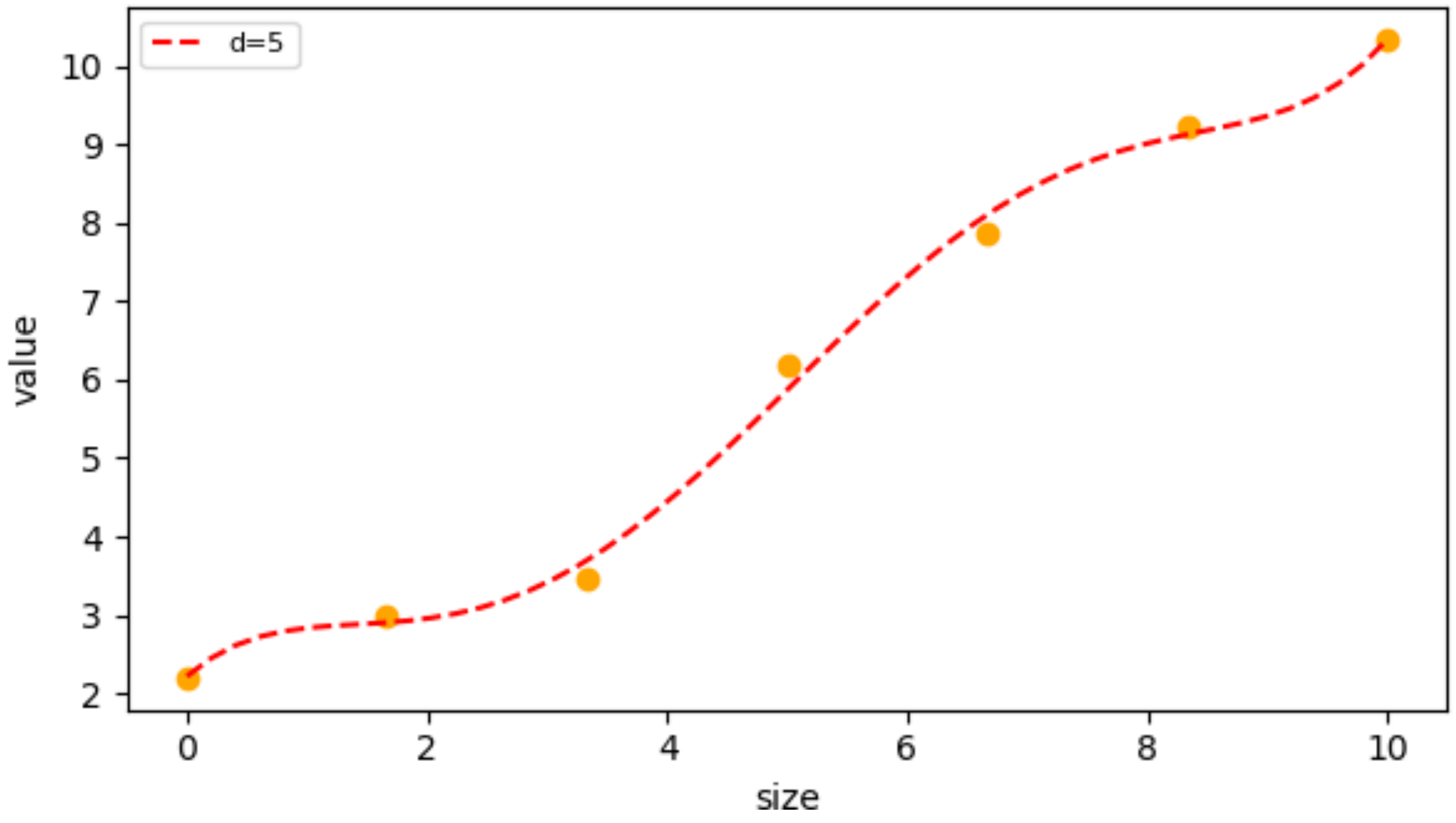
d = 4



- $h_w(x) = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4$

Polynomial Regression

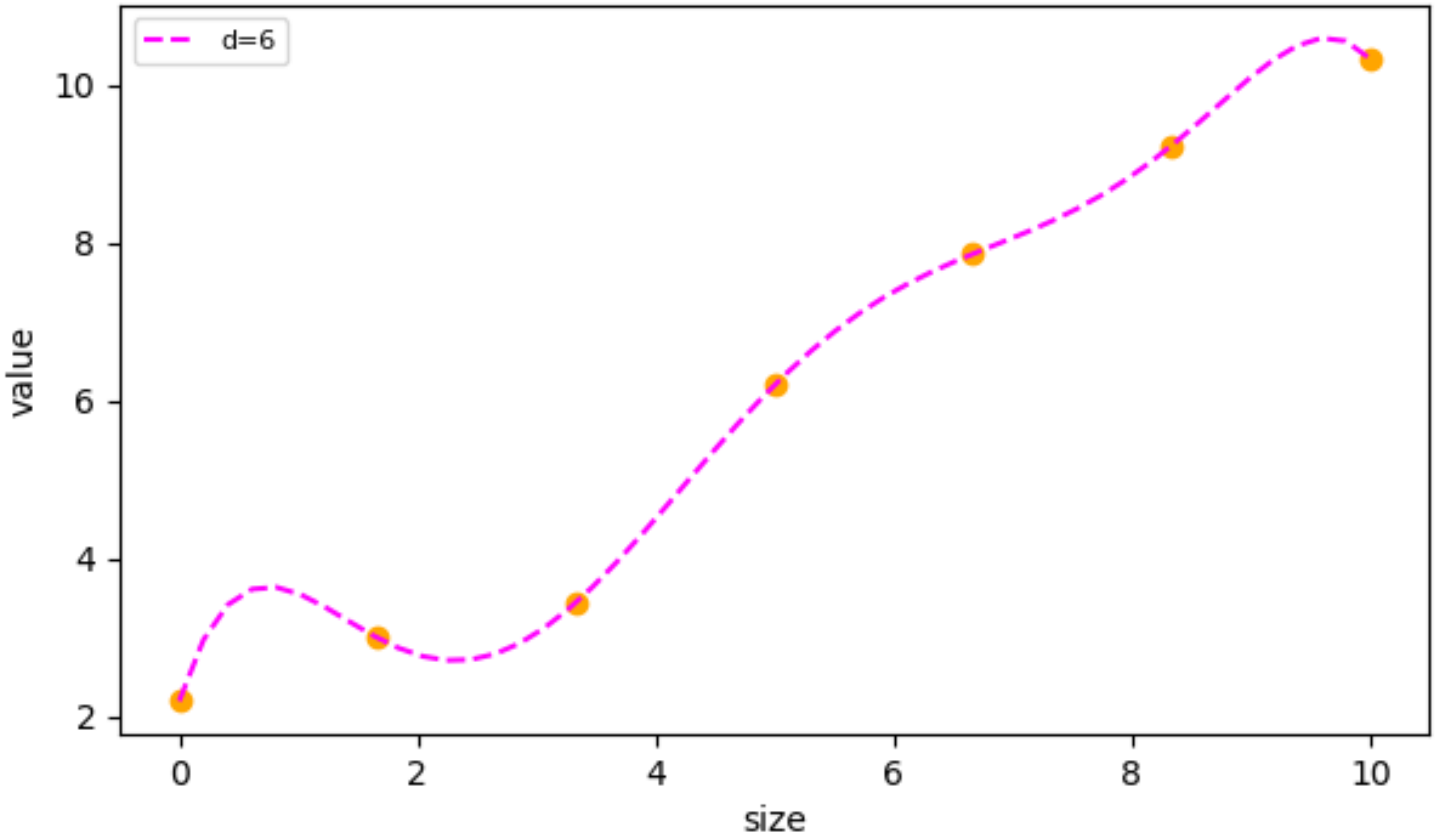
d = 5



- $h_w(x) = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4 + w_5x^5$

Polynomial Regression

d = 6



- $h_w(x) = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4 + w_5x^5 + w_6x^6$

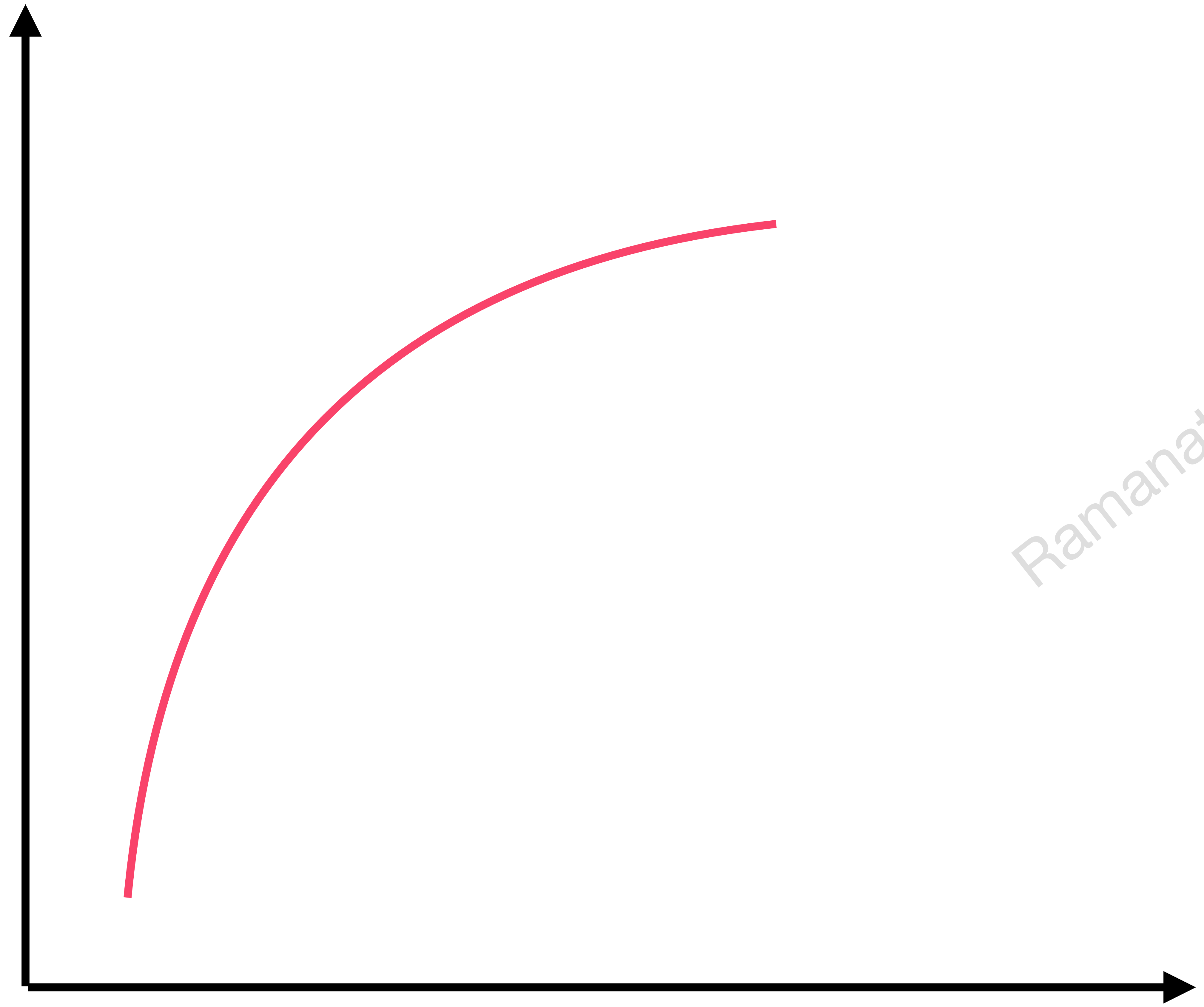
Polynomial Regression

d = 6

- The fit is seemingly the best!
- Oscillations starts for higher degree polynomials
- Given the data, find the weights using optimisation

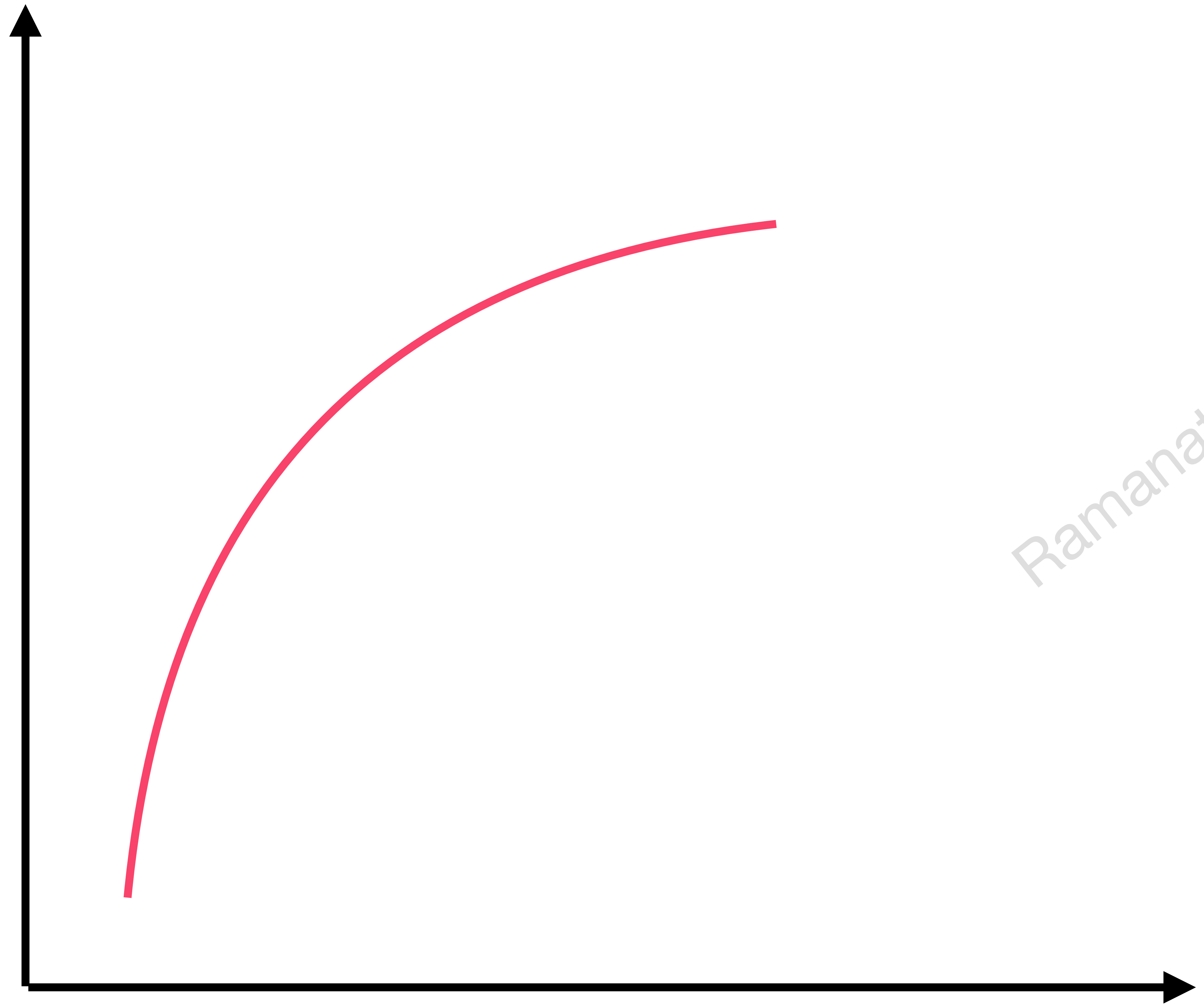
Other issues…

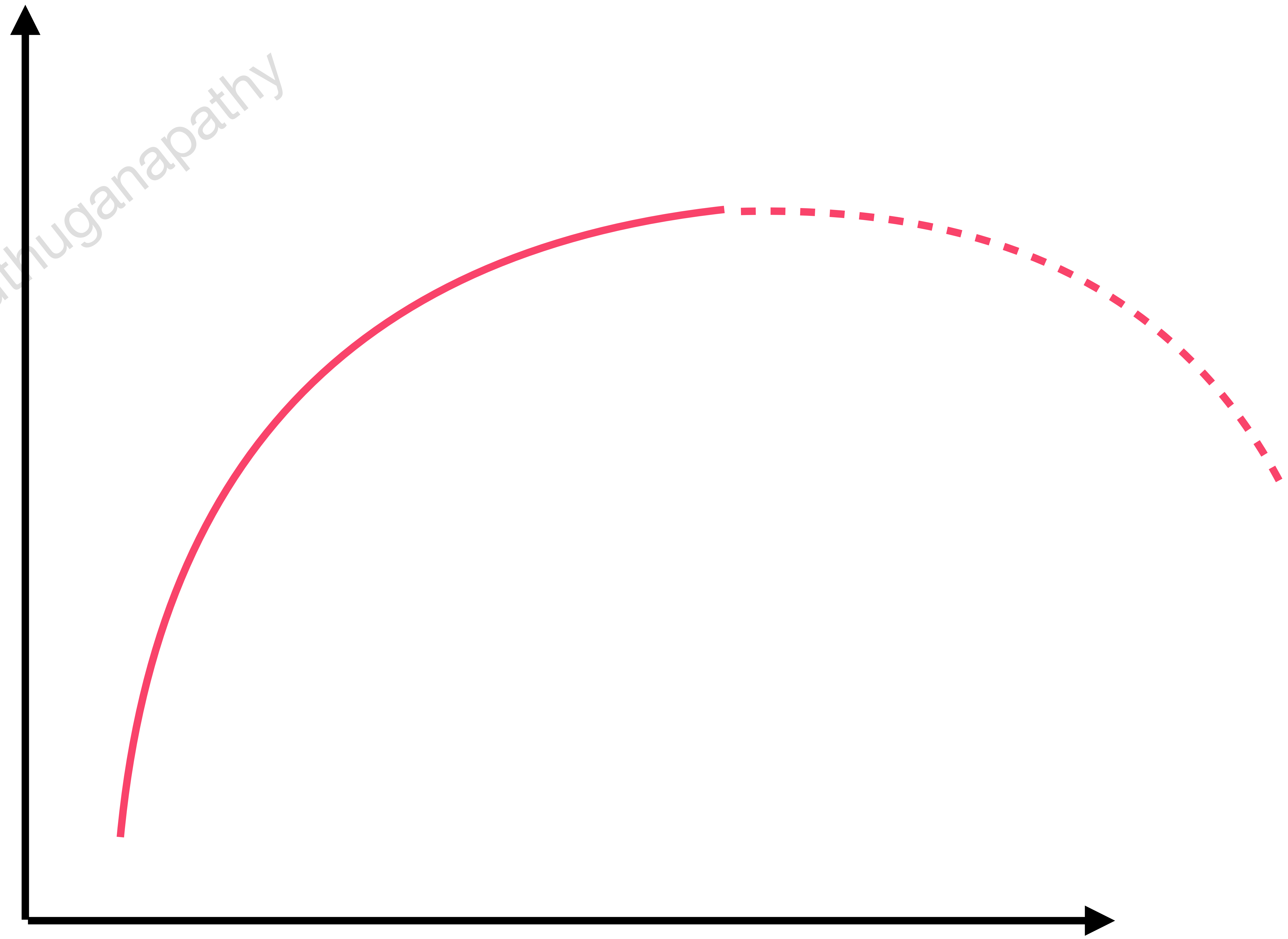
E.g. Quadratic



Other issues…

E.g. Quadratic





Various possible models

- $h_w(x) = w_0 + w_1x + w_2x^2$
- $h_w(x) = w_0 + w_1x + w_2x^2 + w_3x^3$
- $--$
- $--$-
- $-$-
- $h_w(x) = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4 + w_5x^5 + w_6x^6$

Various possible models - multivariate

Two features

- $h_w(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2$
- $h_w(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2$
- $--$
- $--$-
- $-$-
-