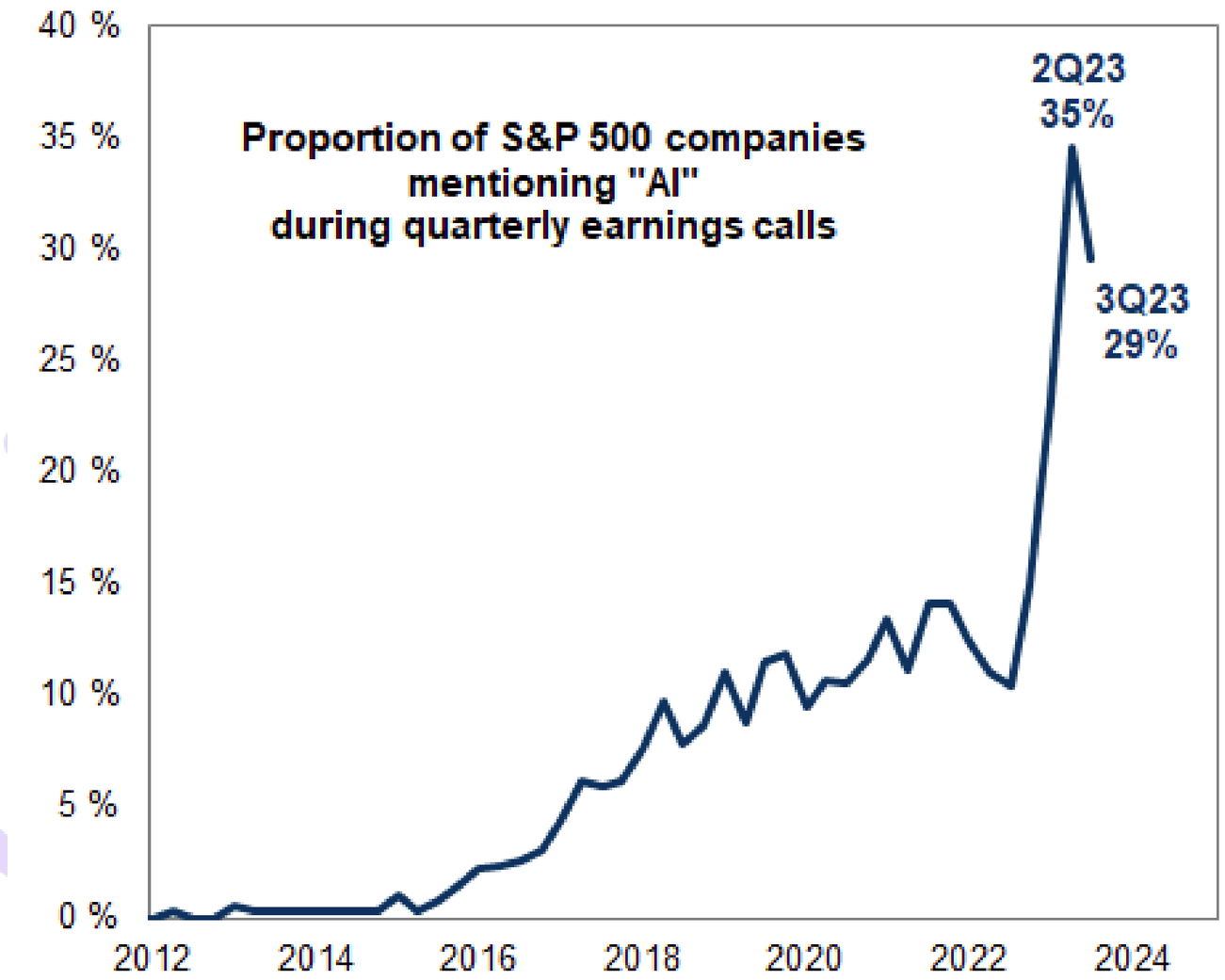
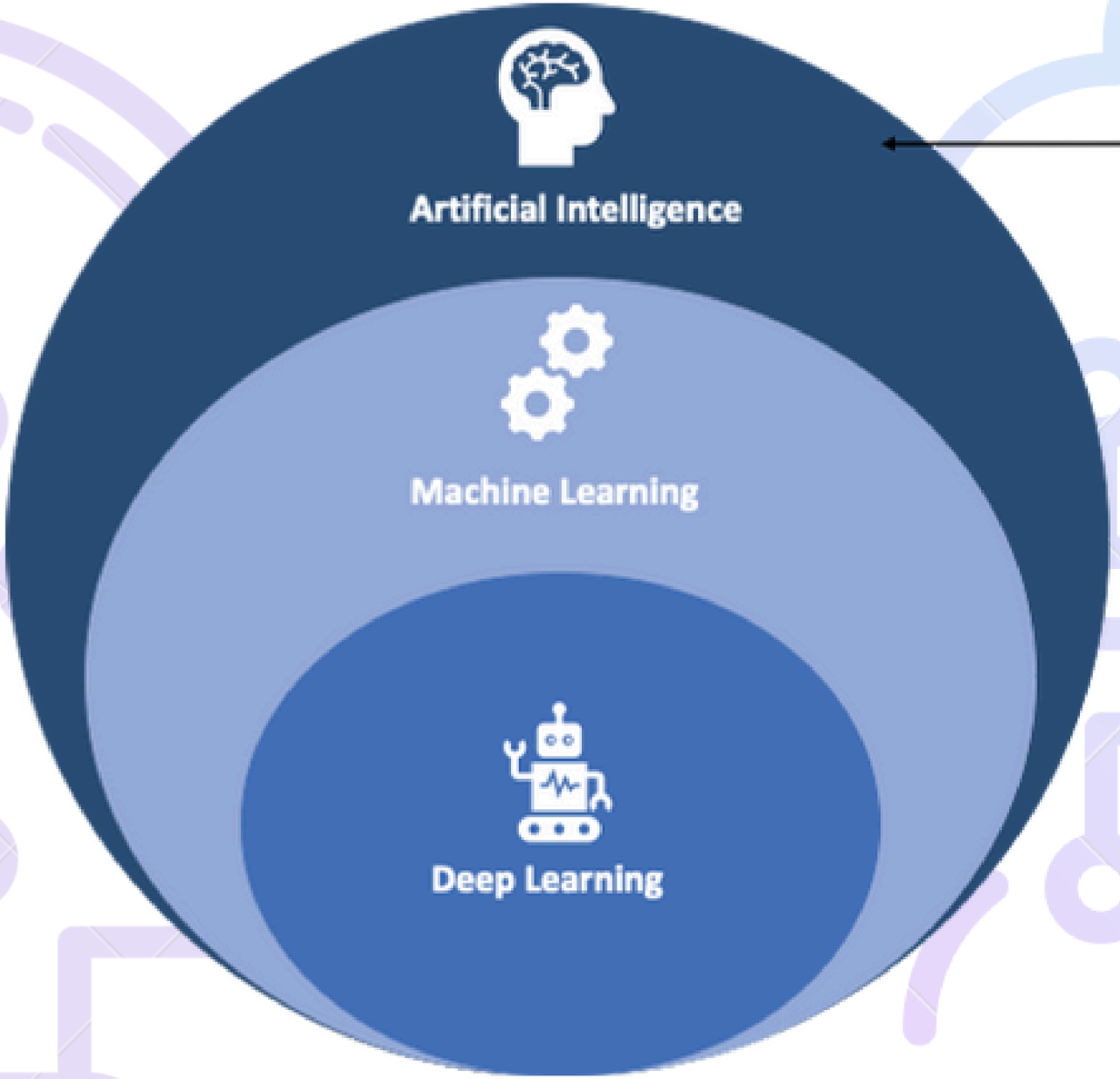
Machine Learning Sup'Com Workshop #1

Introduction + Regression

The Hype Market



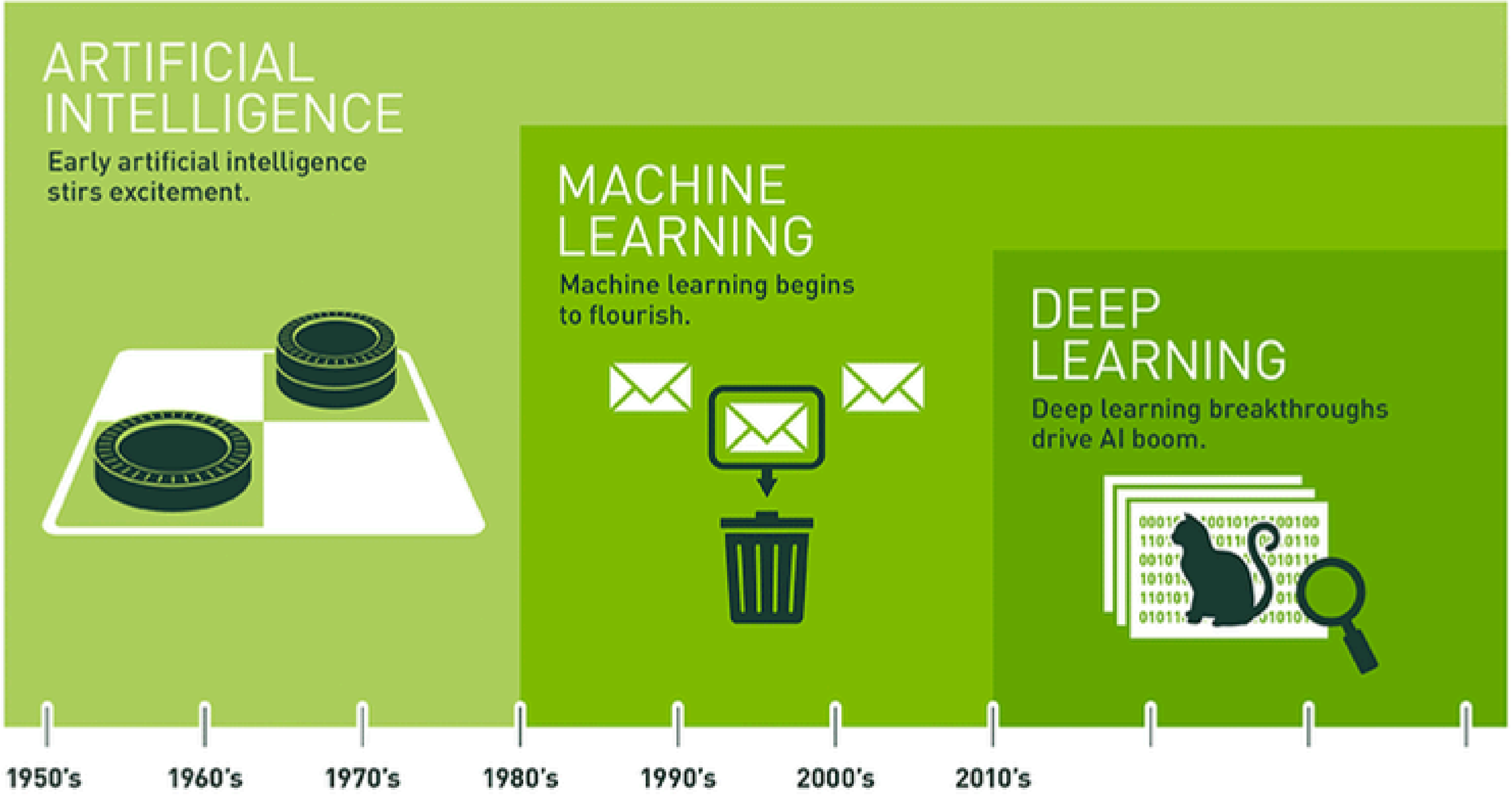
Source: FactSet, Goldman Sachs Global Investment Research



Artificial Intelligence
Technology in which computers behave like humans and have learning capabilities

Machine Learning
Technology in which computers improves its algorithms through learning experience

Deep Learning
Type of machine learning algorithms using complex neural networks



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Data

Structured Data	Unstructured Data
• Consistent data types and formats	• Varied data types and formats
• Easily searchable and analyzed	• Difficult to search and analyze
• Well-defined schema and relationships	• No predefined schema or relationships
• Stored in databases or spreadsheets	• Stored in documents, emails, social media, etc.
• Examples: Sales transactions, financial statements	• Examples: Text documents, images, videos

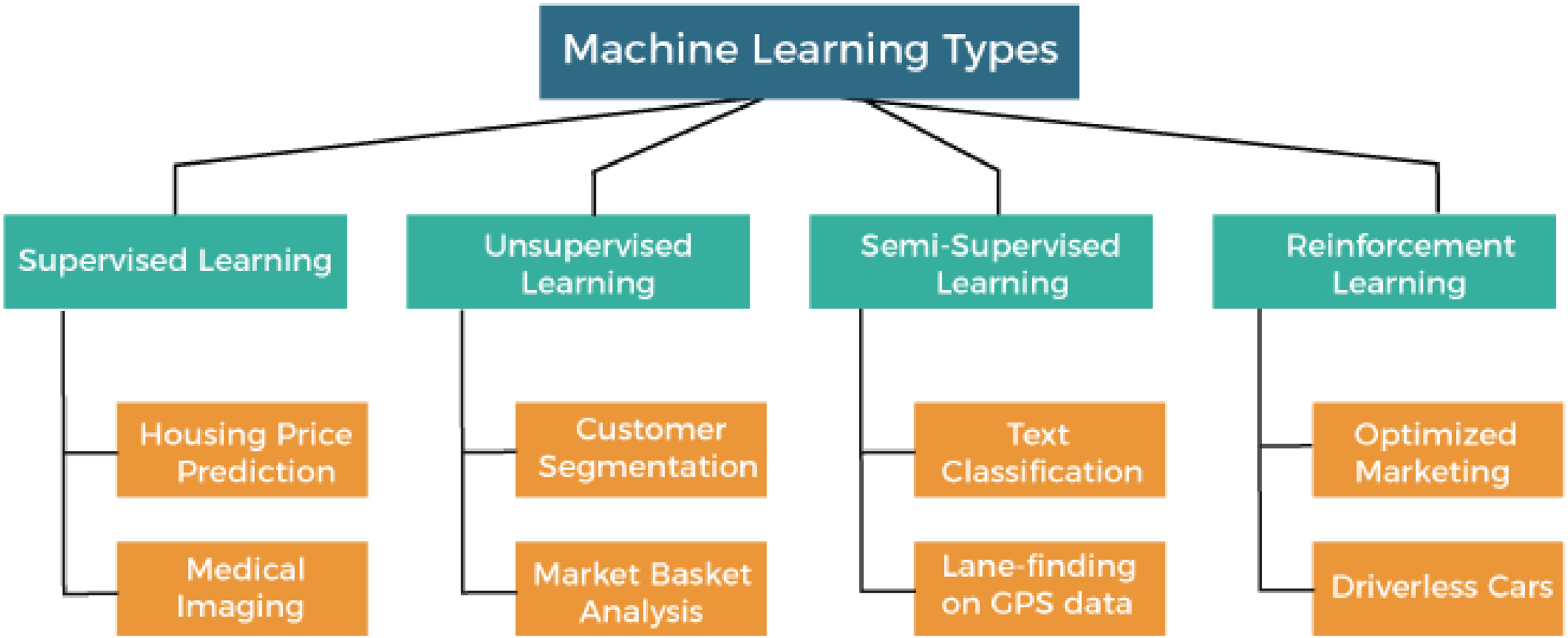
When to use machine learning ?

So..my expectation ?

Machine learning engineer

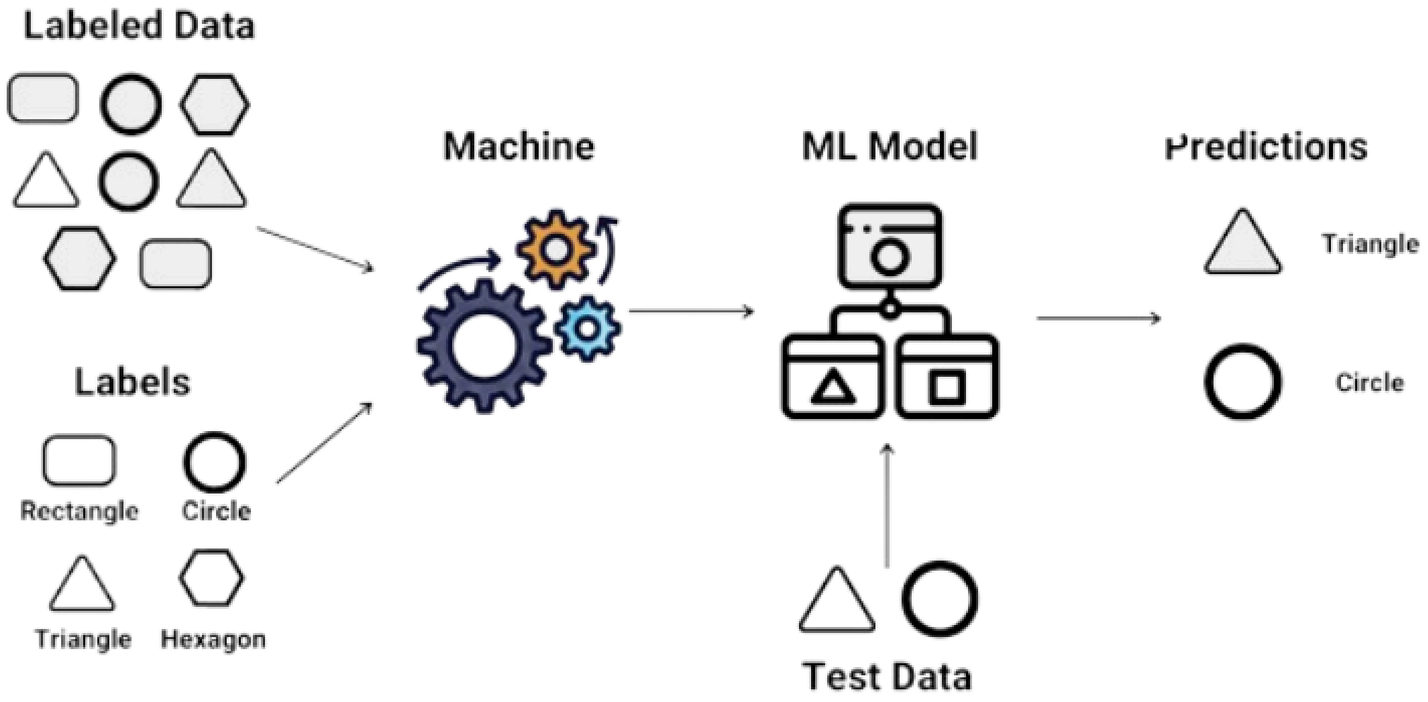
Data Scientist / Analyst

Data Engineer


Machine Learning Types
Supervised Learning
Unsupervised Learning
Semi-Supervised Learning
Reinforcement Learning
Housing Price Prediction
Medical Imaging
Customer Segmentation
Market Basket Analysis
Text Classification
Lane-finding on GPS data
Optimized Marketing
Driverless Cars

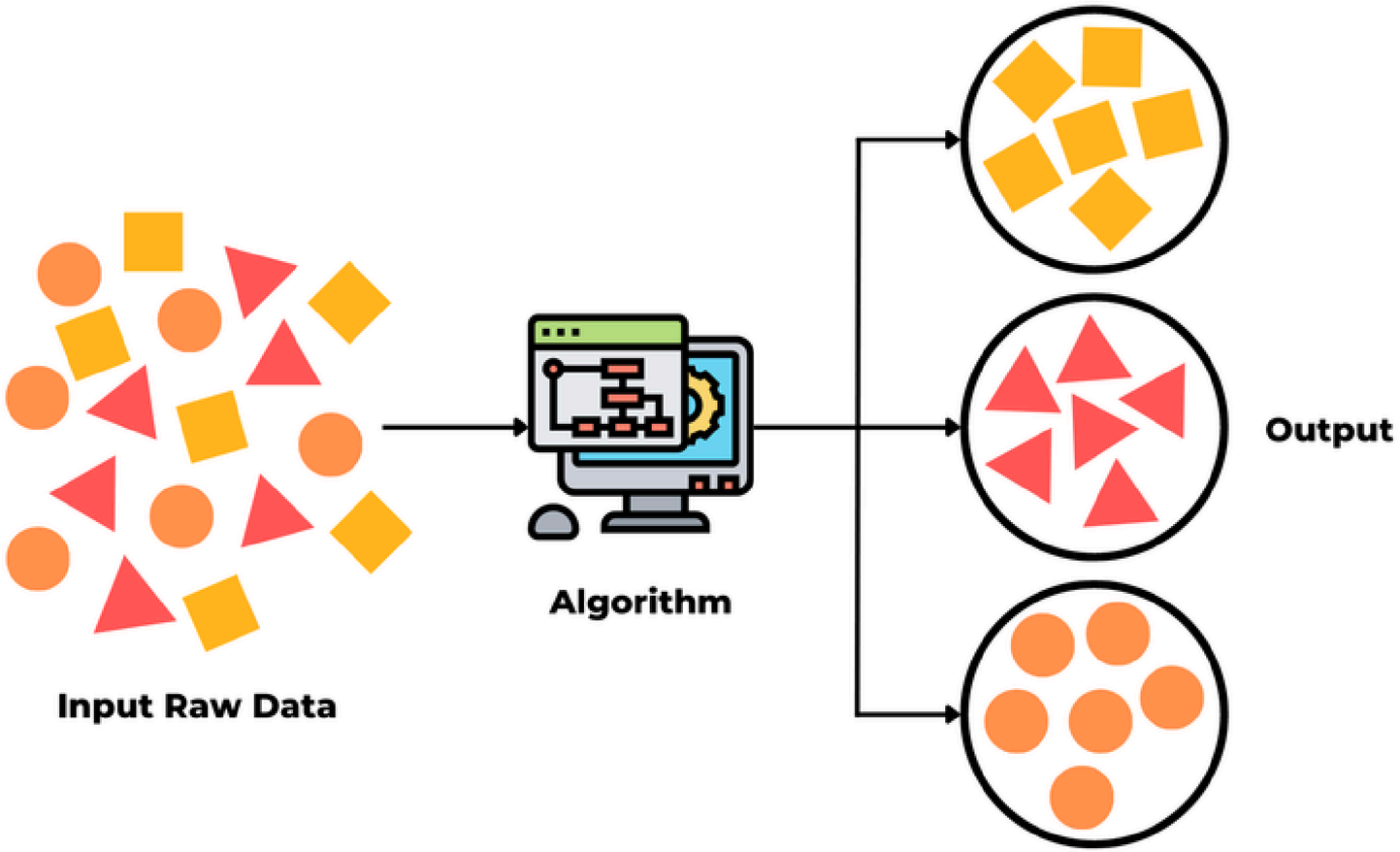
Supervised Learning

i have (x , y) learning f(x)



Unsupervised Learning

i only have x..what do i do ?



Tools we'll be using

jupyter

Keras

PyTorch

Programming languages

python™

R

C++

julia

(if you hate yourself)

(Brief talk about kaggle)

kaggle™

(Brief talk about Hugging Face)

Hugging Face

(Brief talk on Research papers)

arXiv

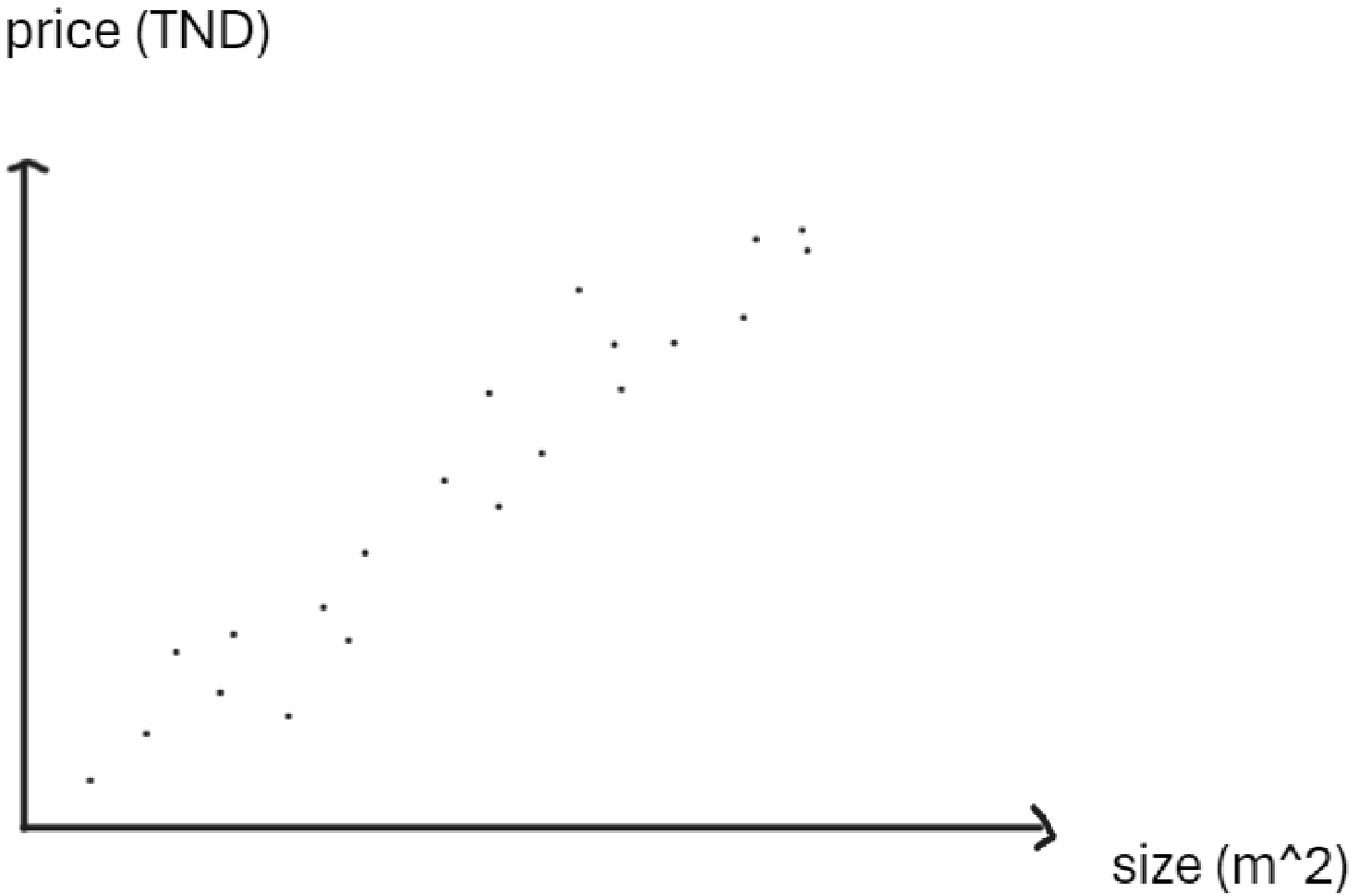
Questions ?

Starting with supervised learning

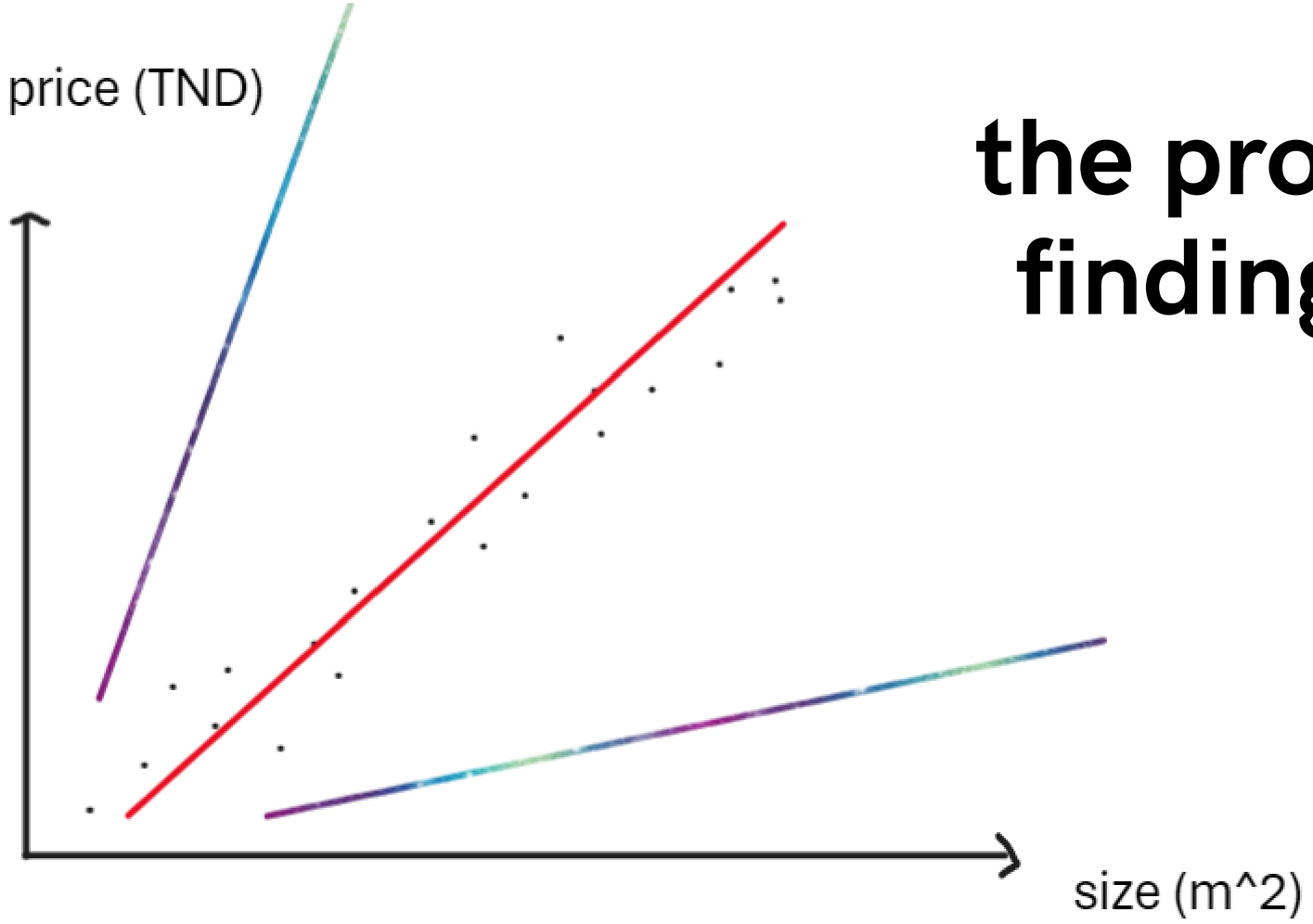
regression

classification

Classical regression example : Housing price prediction



f(x)=w*x+b



the problem is essentially finding the best w and b

The meaning of best w and b

THE COST FUNCTION

$$\frac{1}{2m}\sum_{1}^{m}(h(x^{(i)})-y^{(i)})^2$$

Mean Square Error

$$mae = \frac{\sum_{i=1}^{n} abs\left(y_i - \lambda(x_i)\right)}{n}$$

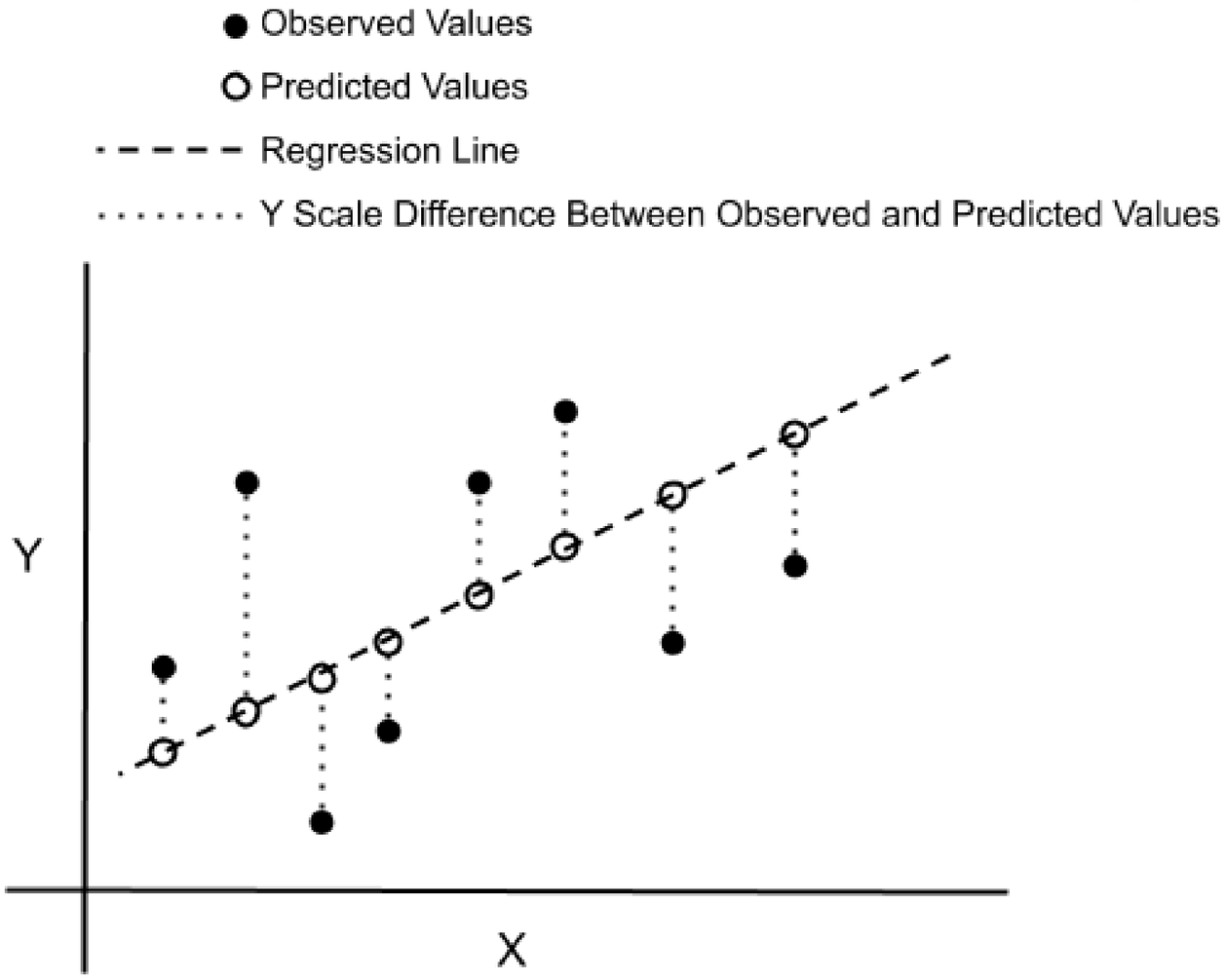
Mean absolute Error

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1-y_i) \cdot log(1-p(y_i))$$

cross entropy loss

Mean Square Error

$$\frac{1}{2m}\sum_{1}^{m}(h(x^{(i)})-y^{(i)})^2$$



The problem becomes finding the miniumum of this loss ?

Consistently finding the minimum?

The gradient descent algorithm

$$J(x) = \frac{1}{2m}\sum_{i=1}^{m}(f(x_i) - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(w \cdot x_i + b - y_i)^2$$

initialze random w and b

1Calculate the derivatives

2)Update the parameters

go back to 1)

*(it is J(w,b) not J(X))

Calculating the derivatives

$$J(x) = \frac{1}{2m}\sum_{i=1}^{m}(w \cdot x_i + b - y_i)^2$$

$$\frac{\partial J(x)}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}(w \cdot x_i + b - y_i) \cdot x_i$$

$$\frac{\partial J(x)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(w \cdot x_i + b - y_i)$$

2)Update the parameters

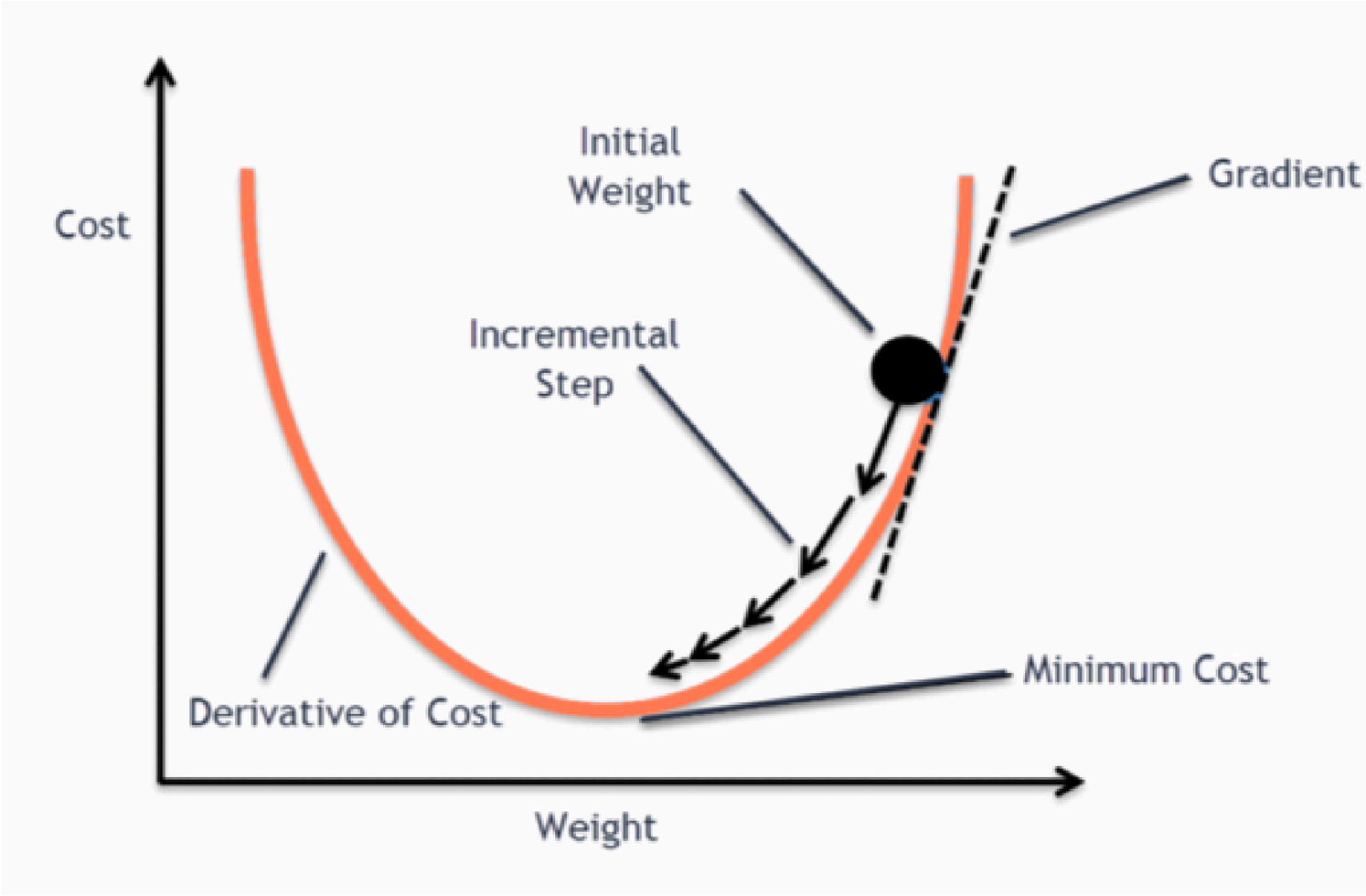
$$w_{new} = w_{old} - \lambda \cdot \frac{\partial J(x)}{\partial w}$$

$$b_{new} = b_{old} - \lambda \cdot \frac{\partial J(x)}{\partial b}$$

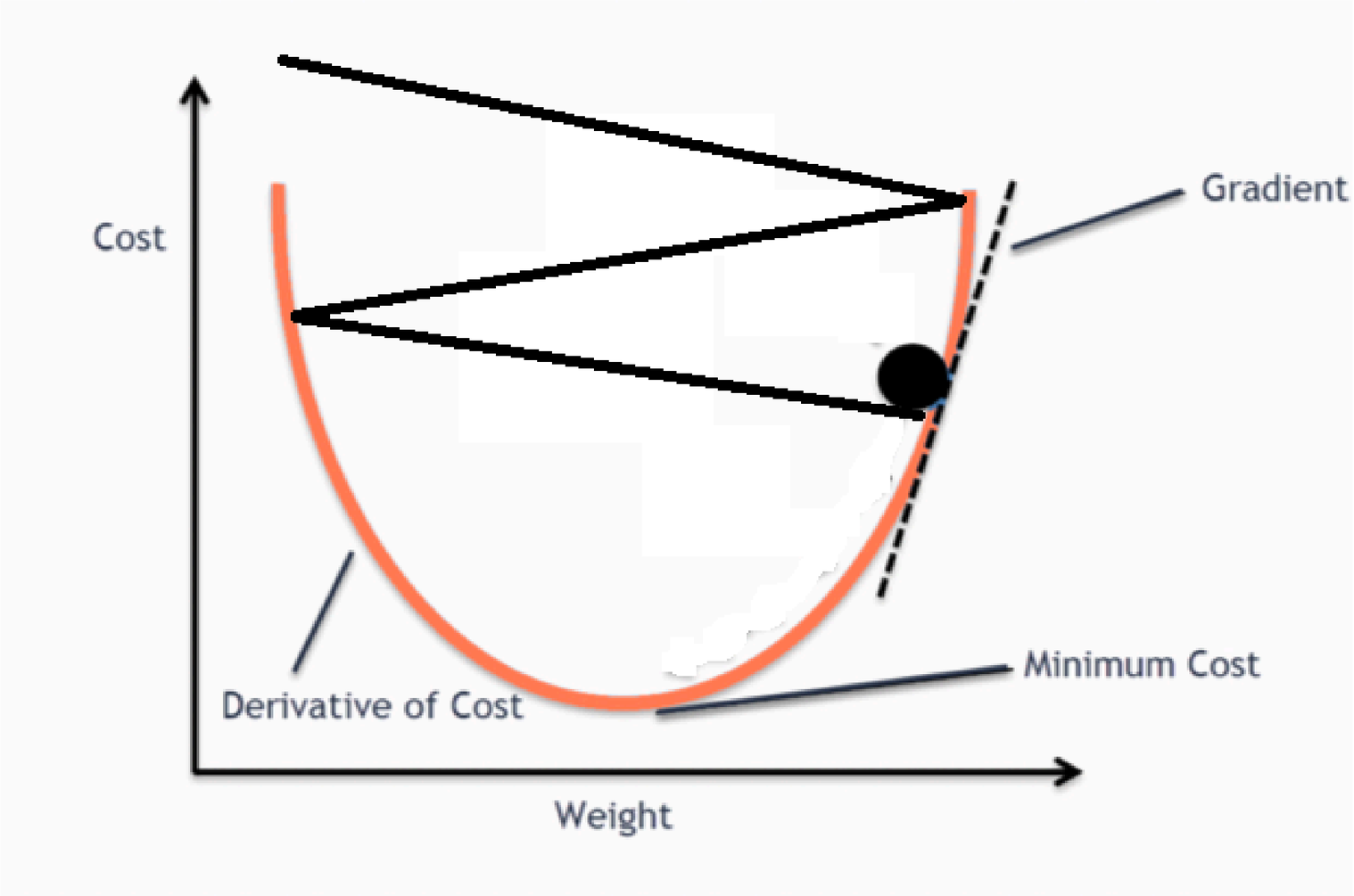
Lambda is called the learning rate

it is referred to as a hyper parameter

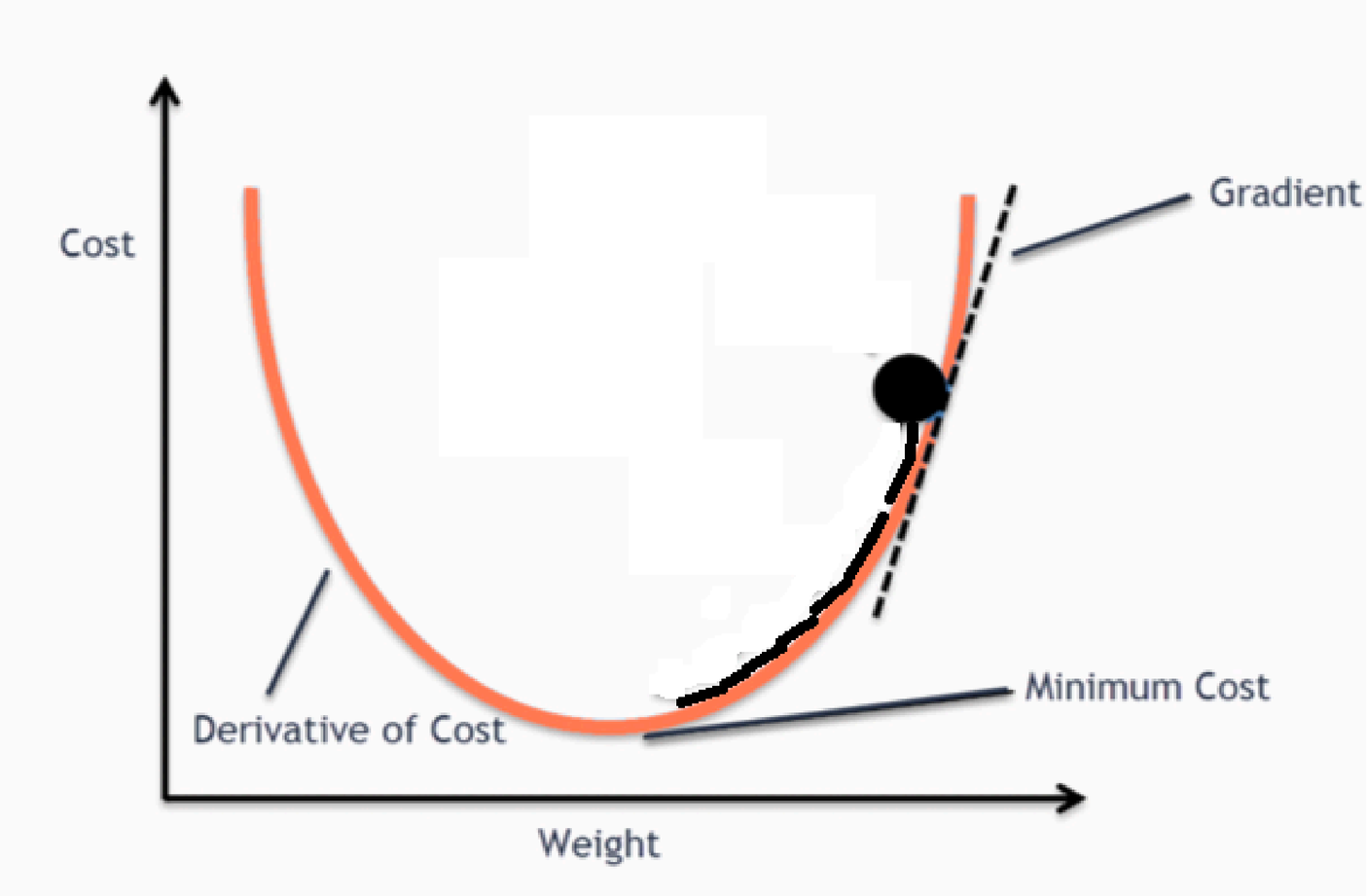
what are we doing ?



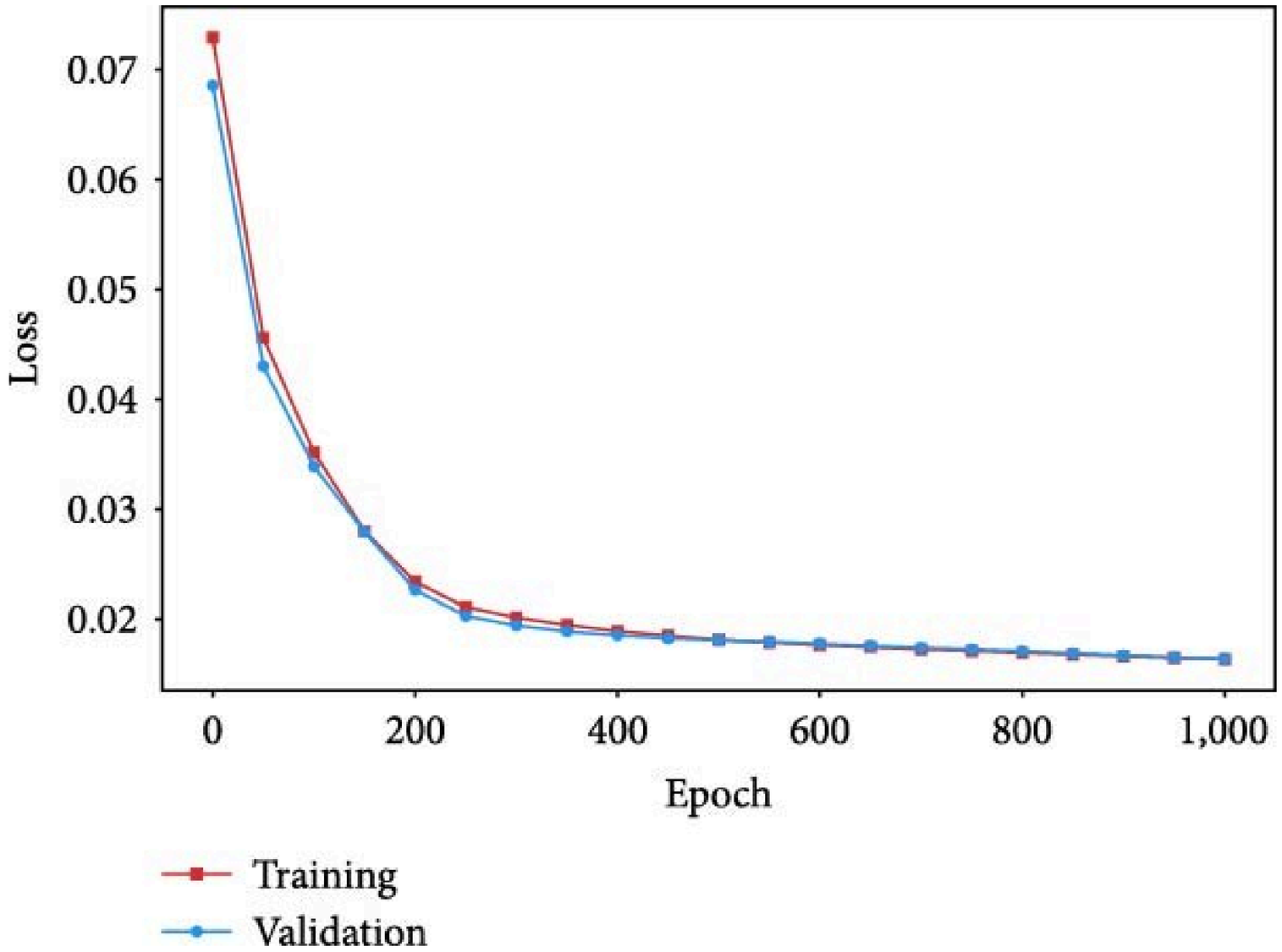
choosing a big lambda



choosing a small lambda

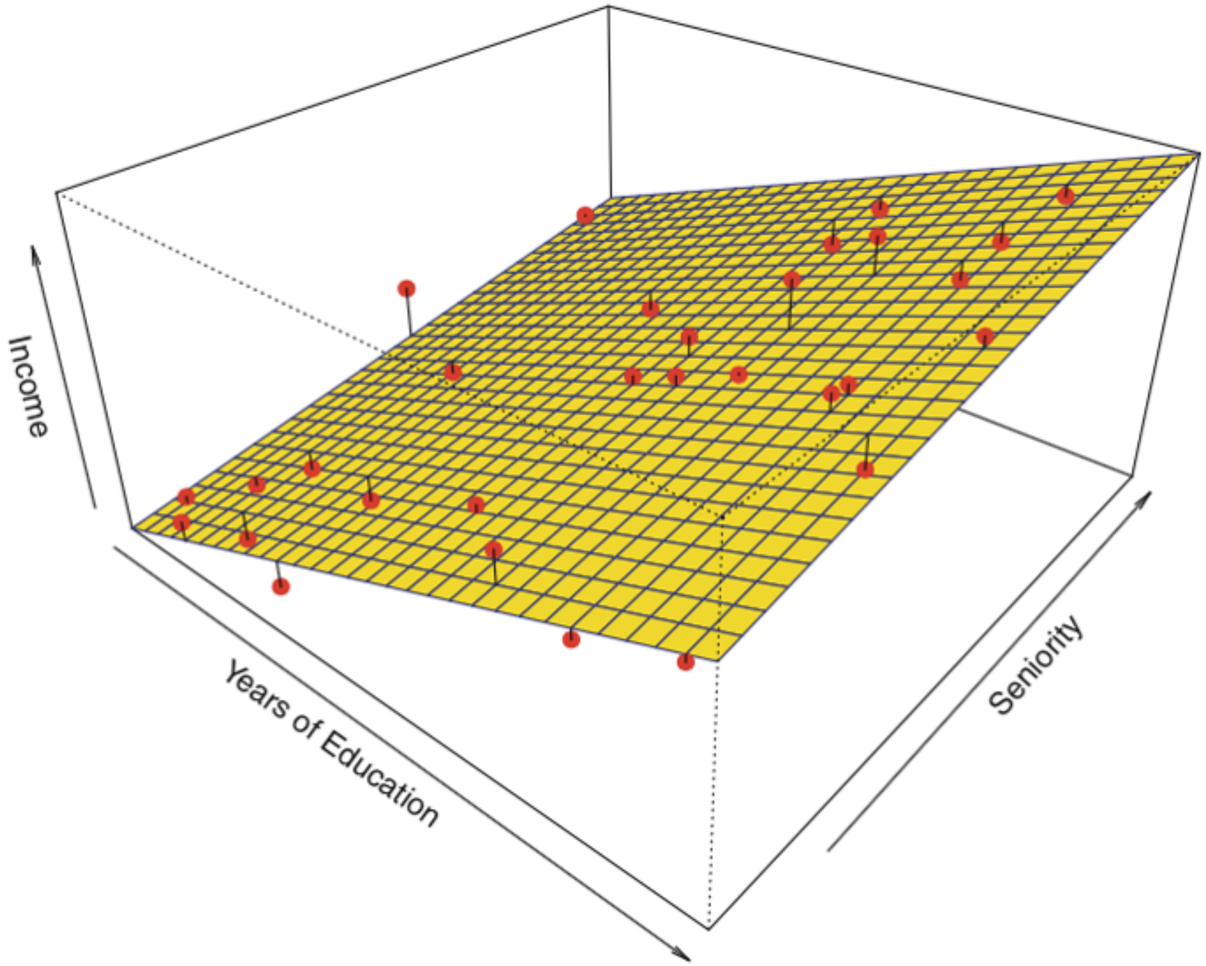


Typical loss graph



The bigger picture

f(x1,x2) = w1*x1+w2*x2+b



$$\mathrm{MSE}(w_1, w_2, b) = \frac{1}{2n} \sum_{i=1}^{n} (w_1 x_{1i} + w_2 x_{2i} + b - z_i)^2$$

1. Derivative with respect to w_1:

$$\frac{\partial \mathrm{MSE}}{\partial w_1} = \frac{1}{n} \sum_{i=1}^{n} (w_1 x_{1i} + w_2 x_{2i} + b - z_i)\, x_{1i}$$

2. Derivative with respect to w_2:

$$\frac{\partial \mathrm{MSE}}{\partial w_2} = \frac{1}{n} \sum_{i=1}^{n} (w_1 x_{1i} + w_2 x_{2i} + b - z_i)\, x_{2i}$$

3. Derivative with respect to b:

$$\frac{\partial \mathrm{MSE}}{\partial b} = \frac{1}{n} \sum_{i=1}^{n} (w_1 x_{1i} + w_2 x_{2i} + b - z_i)$$

1. Update w_1:

$$w_{1_{\text{new}}} = w_{1_{\text{old}}} - \lambda \cdot \frac{1}{n} \sum_{i=1}^{n} \left(w_1 x_{1i} + w_2 x_{2i} + b - z_i\right) x_{1i}$$

2. Update w_2:

$$w_{2_{\text{new}}} = w_{2_{\text{old}}} - \lambda \cdot \frac{1}{n} \sum_{i=1}^{n} \left(w_1 x_{1i} + w_2 x_{2i} + b - z_i\right) x_{2i}$$

3. Update b:

$$b_{\text{new}} = b_{\text{old}} - \lambda \cdot \frac{1}{n} \sum_{i=1}^{n} \left(w_1 x_{1i} + w_2 x_{2i} + b - z_i\right)$$

General case

$$f_{\mathbf{w},b}(\mathbf{x}) = w_0x_0 + w_1x_1 + \ldots + w_{n-1}x_{n-1} + b$$

$$f_{\mathbf{w},b}(\mathbf{x}^{(i)}) = \mathbf{w} \cdot \mathbf{x}^{(i)} + b$$

$$\frac{\partial J(\mathbf{w}, b)}{\partial w_j} = \frac{1}{m}\sum_{i=0}^{m-1}(f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})x_j^{(i)}$$

$$\frac{\partial J(\mathbf{w}, b)}{\partial b} = \frac{1}{m}\sum_{i=0}^{m-1}(f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})$$

$$w_j = w_j - \alpha\frac{\partial J(\mathbf{w}, b)}{\partial w_j} \qquad \text{for j = 0..n-1}$$

$$b = b - \alpha\frac{\partial J(\mathbf{w}, b)}{\partial b}$$

More generality

$$h_{\mathbf{w}}(x) = w_0 + w_1 x + w_2 x^2$$

$$J(w_0, w_1, w_2) = \frac{1}{2m} \sum_{i=1}^{m} (h_{\mathbf{w}}(x_i) - y_i)^2$$

Derivative with respect to w_0 (the intercept term):

$$\frac{\partial J(w_0, w_1, w_2)}{\partial w_0} = \frac{1}{m} \sum_{i=1}^{m} (h_{\mathbf{w}}(x_i) - y_i)$$

Derivative with respect to w_1 (the coefficient for x):

$$\frac{\partial J(w_0, w_1, w_2)}{\partial w_1} = \frac{1}{m} \sum_{i=1}^{m} (h_{\mathbf{w}}(x_i) - y_i)\, x_i$$

Derivative with respect to w_2 (the coefficient for x^2):

$$\frac{\partial J(w_0, w_1, w_2)}{\partial w_2} = \frac{1}{m} \sum_{i=1}^{m} (h_{\mathbf{w}}(x_i) - y_i)\, x_i^2$$

1. For w_0 (intercept term):

$$w_0 = w_0 - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} (h_{\mathbf{w}}(x_i) - y_i)$$

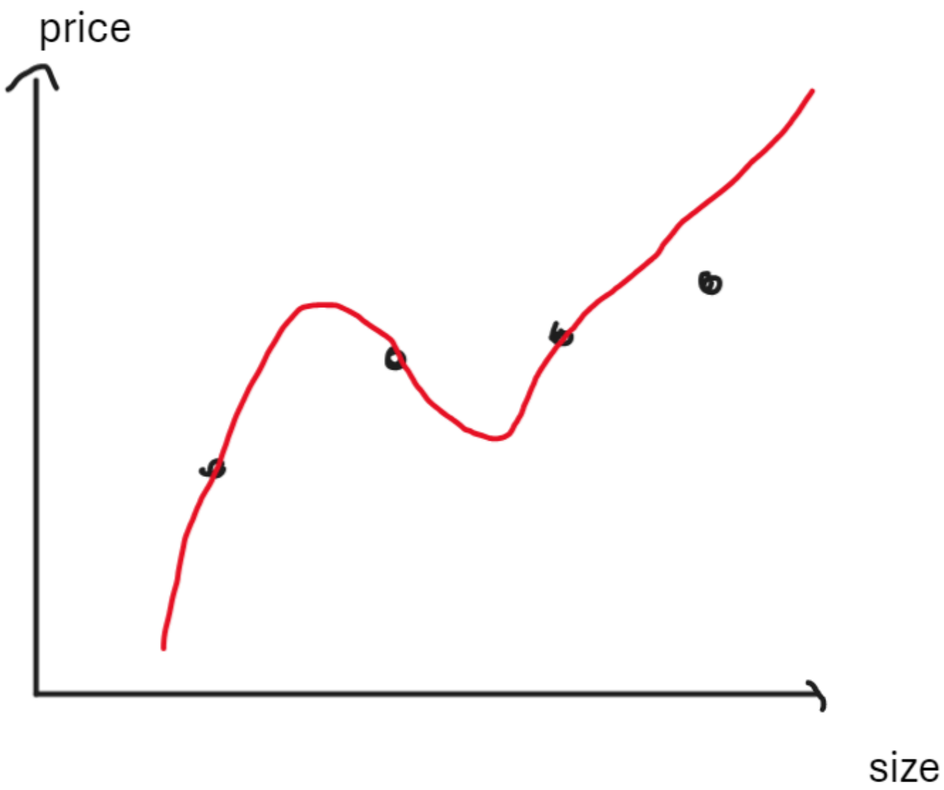
2. For w_1 (linear term coefficient):

$$w_1 = w_1 - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} (h_{\mathbf{w}}(x_i) - y_i)\, x_i$$

3. For w_2 (quadratic term coefficient):

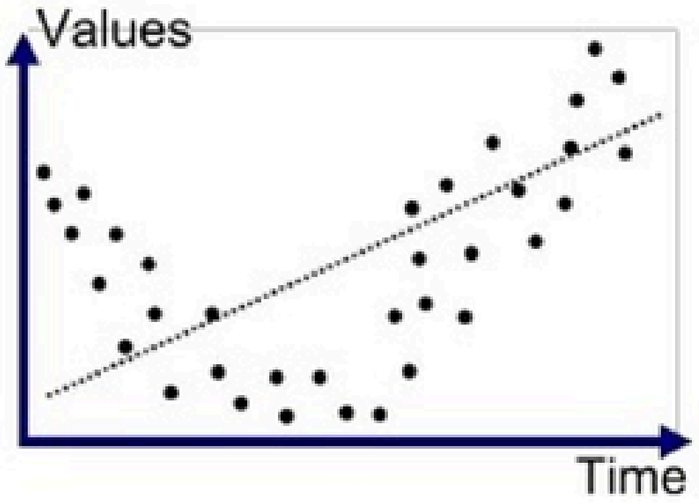
$$w_2 = w_2 - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} (h_{\mathbf{w}}(x_i) - y_i)\, x_i^2$$

Going back to our case

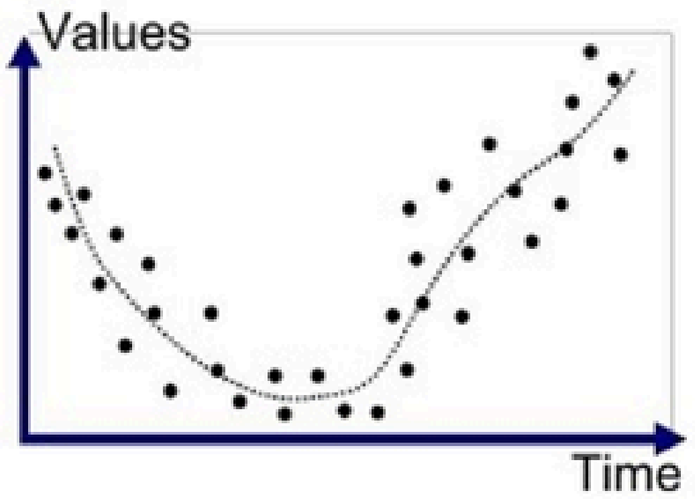


is low loss <=> good model?

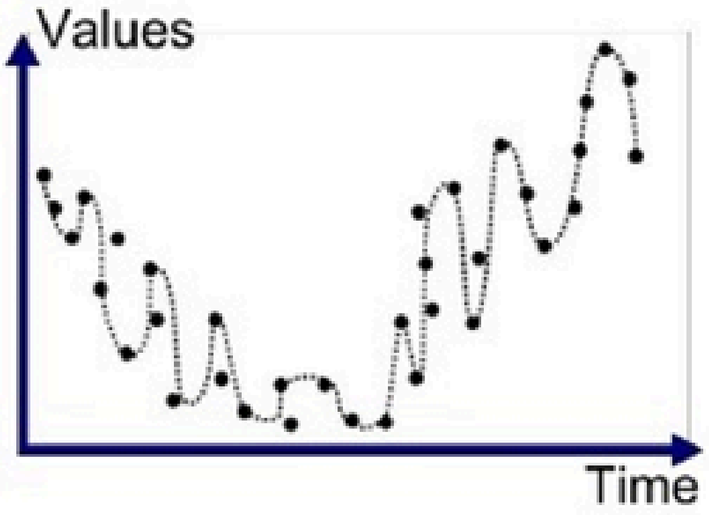
Overfitting



Underfitted



Good Fit/Robust



Overfitted

How to fix over fitting in this case ?

Regularization

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$