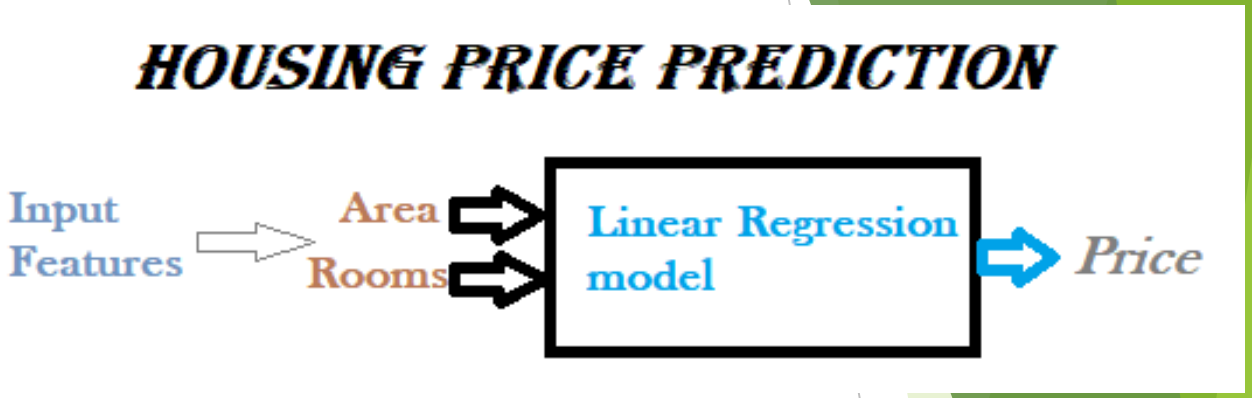
Machine Learning and Neural Networks

-Ranjith MS

Machine Learning

- What is machine learning, where it is used?
- Types of Machine learning
- How to Implement Machine learning models
- Optimization algorithms

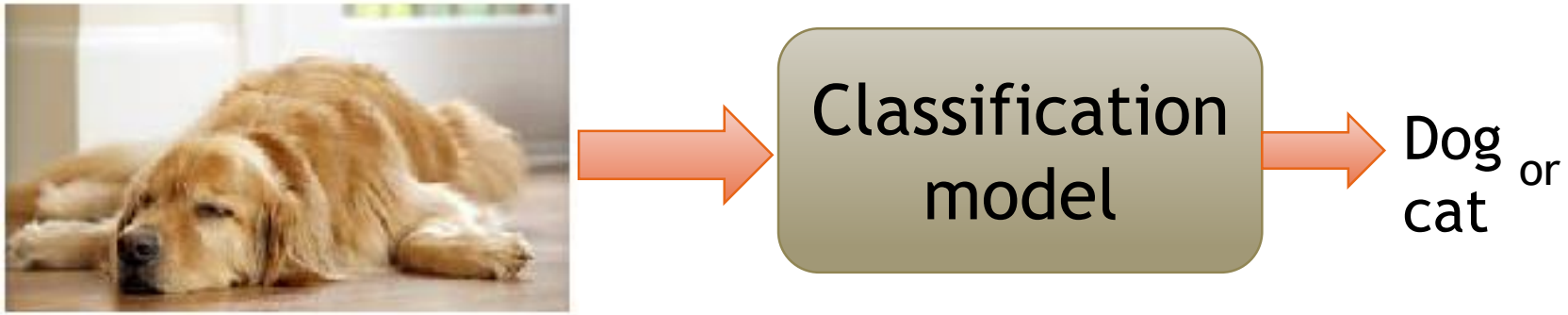


Data Visualization

- Seaborn
- Plotly
- Matplotlib

- ***Neural Networks and Tensorflow***
- What are neural networks and why?.
- Use of deep learning framework
- Implementation of neural network (digit Recognition)

- ***Creating dataset with google images to build Classification model***



Neural Style Transfer

Content Image | Style Image | Generated Image

Content Image | Style Image | Generated Image

Prerequisites

- Familiarity with programming language (preferably with python)
- Basics of linear algebra (eqn. of a straight line, derivatives) <Not mandatory>

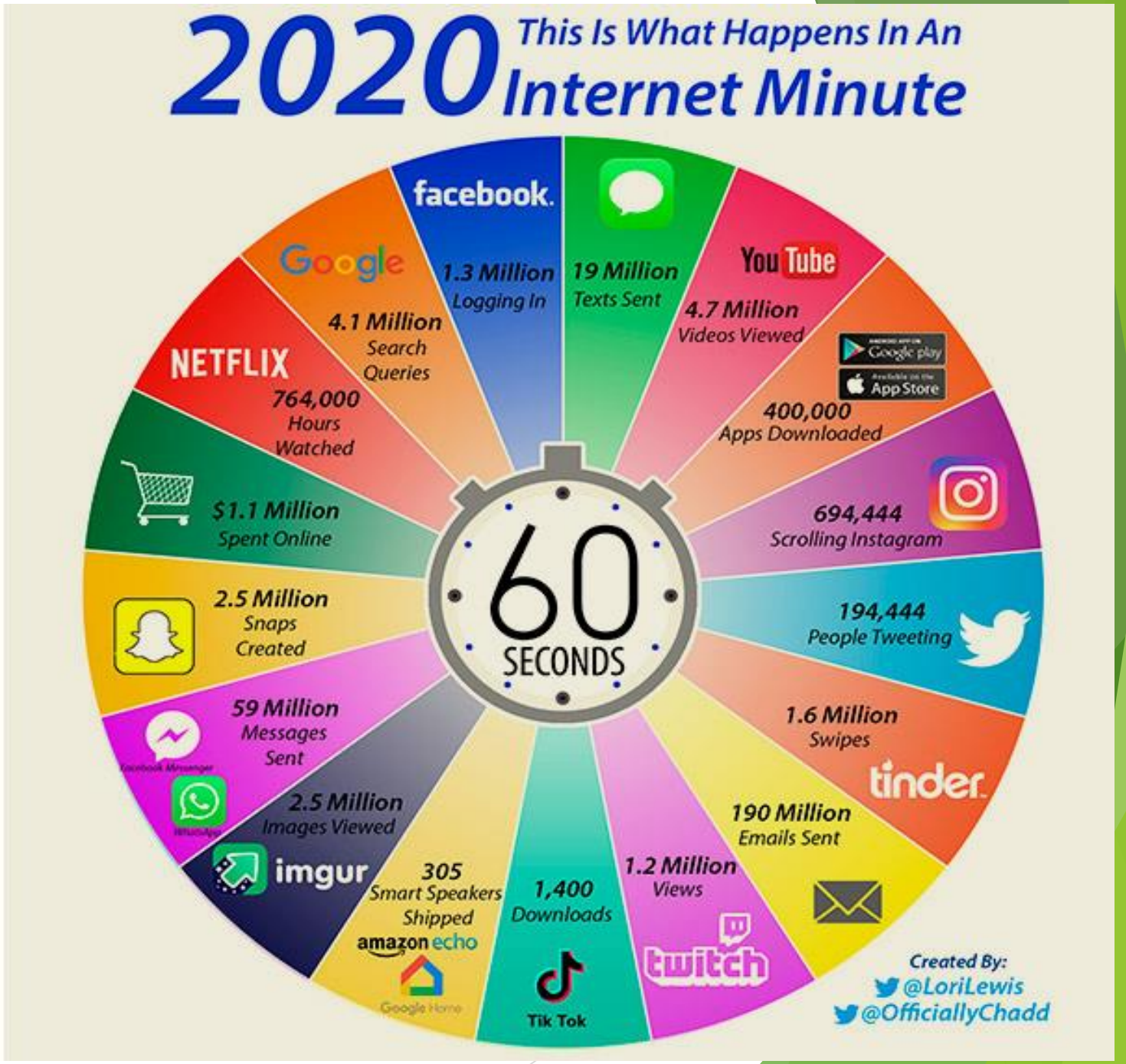
Outline:

- What's Machine Learning??..
- Supervised and Unsupervised Learning
- Hypothesis Function
- Gradient Descent Algorithm
- Feature Scaling and Mean Normalization
- Underfitting and Overfitting
- Back Propagation
- Hands On: Housing Price Prediction

What's Machine Learning or Artificial Intelligence and Where is it used???.....

- web search
- Facebook recognizes your friends in your picture
- Spam filter in emails
- self-driving cars
- Gaming Control

Need of Machine learning and AI



COSMIC SKILLS


HTML + CSS + JS + PHP
= Mode 1
HTML5 + Bootstrap + React JS + Laravel
= Mode 2
Mode 2 + Machine
Learning + Deep Learning

Question: What is the Relation between X and Y ??...

X	Y
2	2
-3	-3
5	5
6	6
0	0
1.3	1.3
10	10

- X=Y

Question: What is the Relation between X and Y ??...

X	Y
2	-4
3	-6
-2	4
5	-10
0	0

- $Y=-2X$

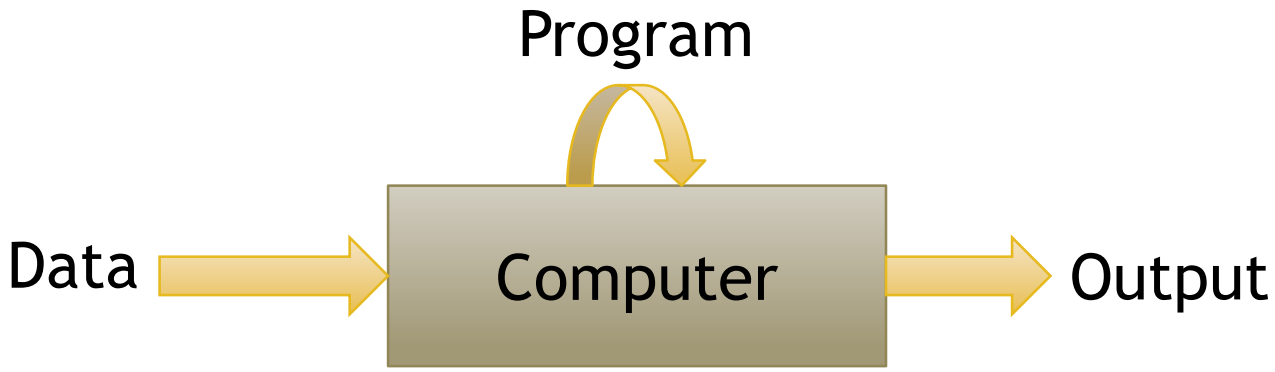
Ranjith MS

Now!!...
What's the relation between X and Y???...

X	Y
2	4
7	19
-5	-17
10	28
15	43

- $Y=3X-2$

Machine Learning

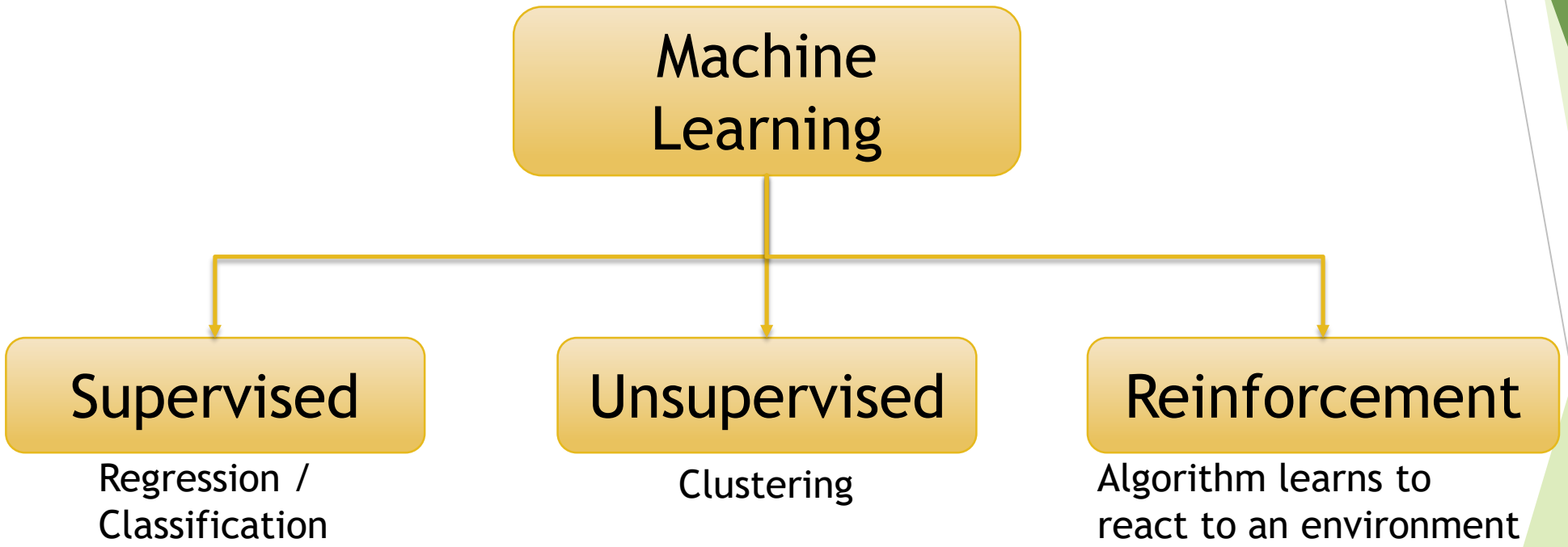


Programming

So What's Machine Learning??...

- **Machine learning is a science of getting computers to learn without being explicitly programmed.**

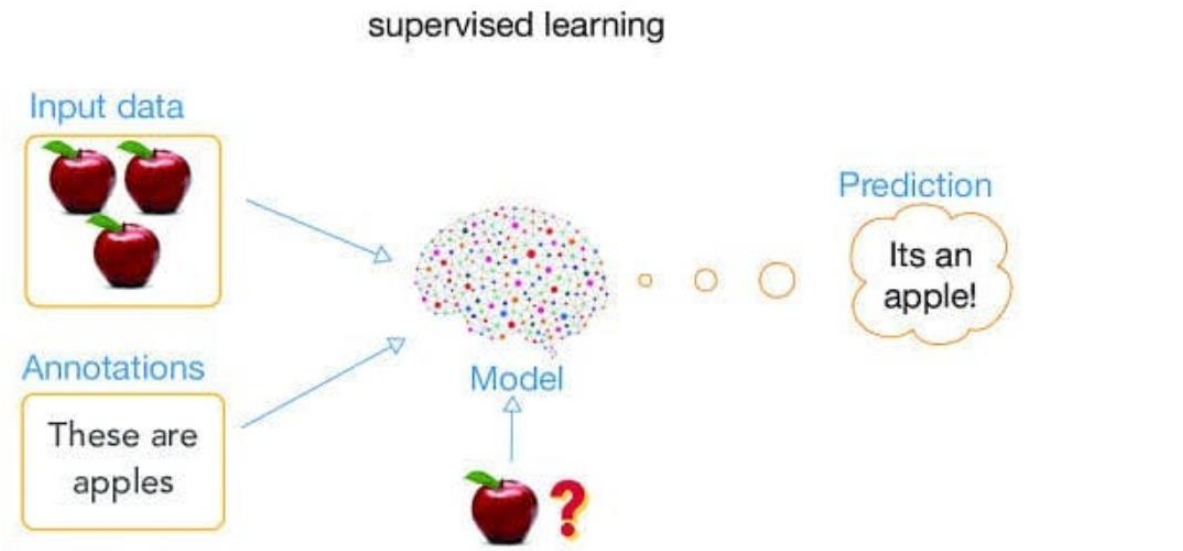
Types of Machine learning

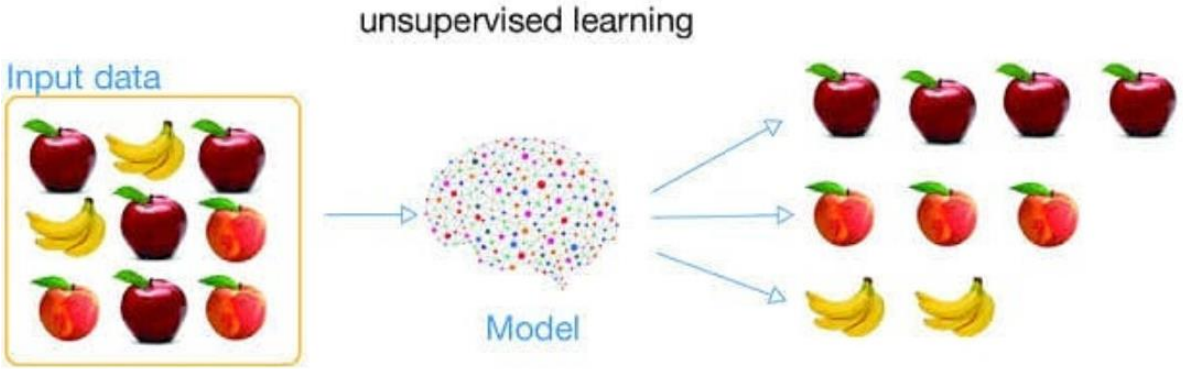


Supervised learning

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Eg : Predicting the house price given the size of the house based on the previous data.


supervised learning
Input data
Prediction
Its an apple!
Annotations
These are apples
Model


unsupervised learning
Input data
Model

Classification of Supervised Learning

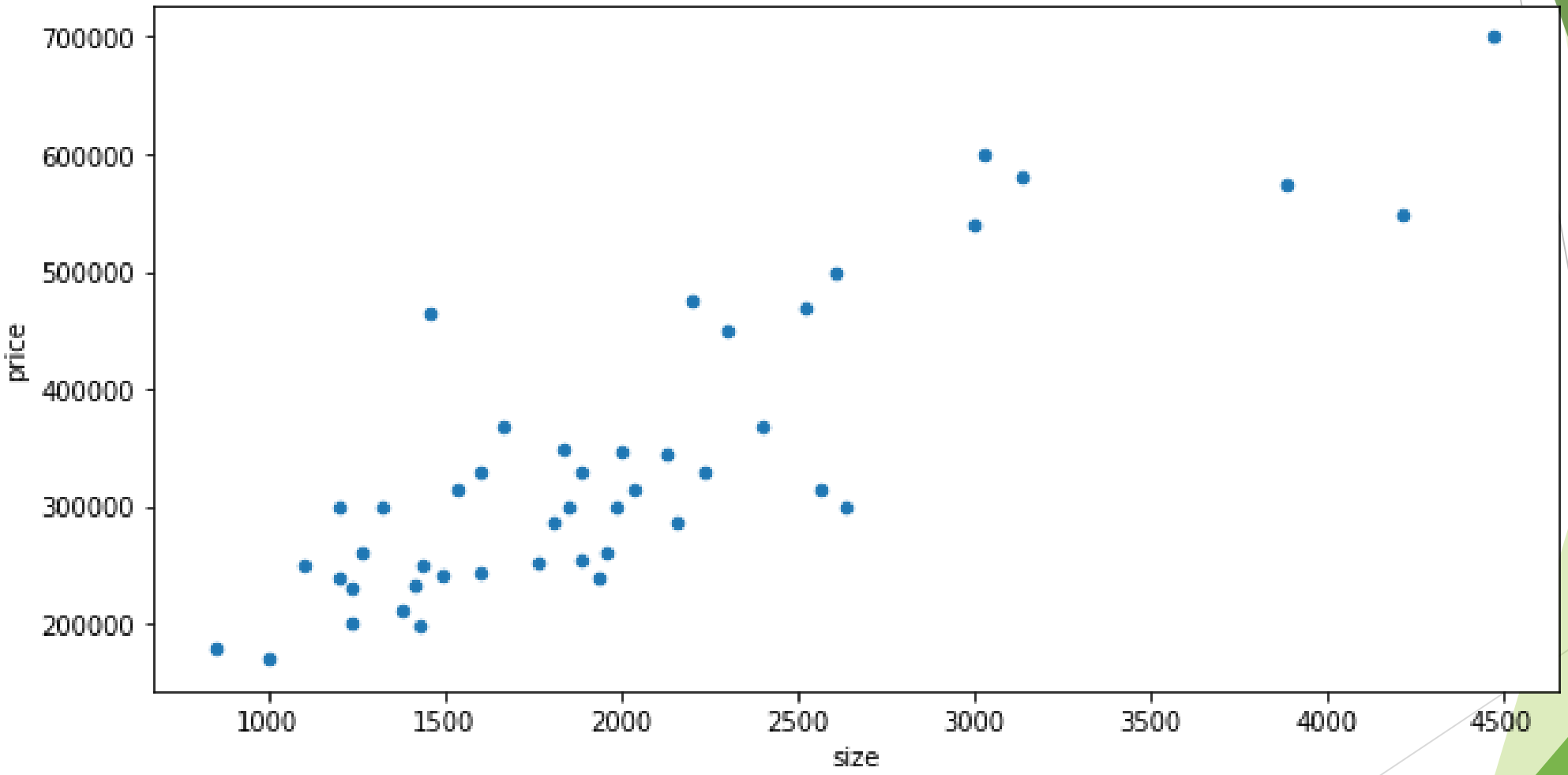
Regression

- A regression problem is when the output variable is a real value, such as “dollars” or “weight”.
- Eg: Given a picture of a person, we have to predict their age on the basis of the given picture

Classification

- A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
- Eg: Given a patient with a tumor, we have to predict whether the tumor is malignant or benign.

Regression


700000
600000
500000
400000
300000
200000
price
1000
1500
2000
2500
3000
3500
4000
4500
size

Training Set:

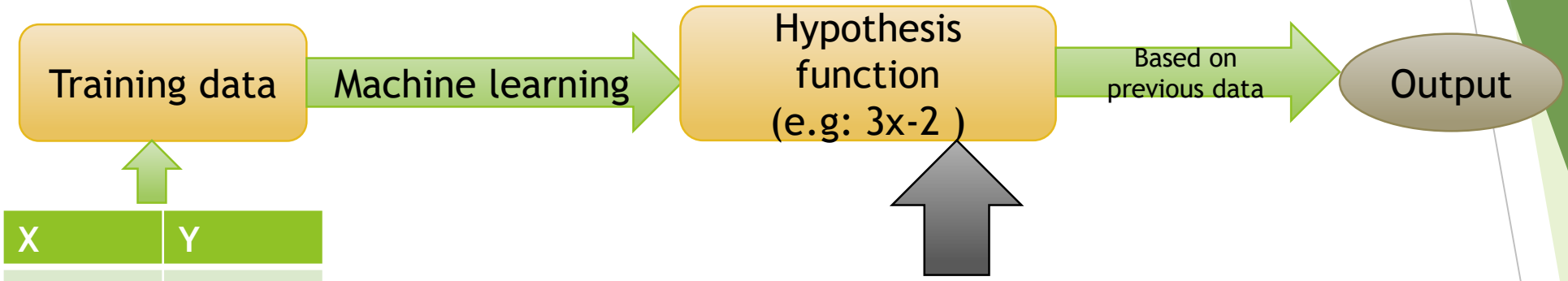
Notation:

m=Number of Training Examples

x’s = “input” variable / features

Y’s = Output/Target Variables

Steps

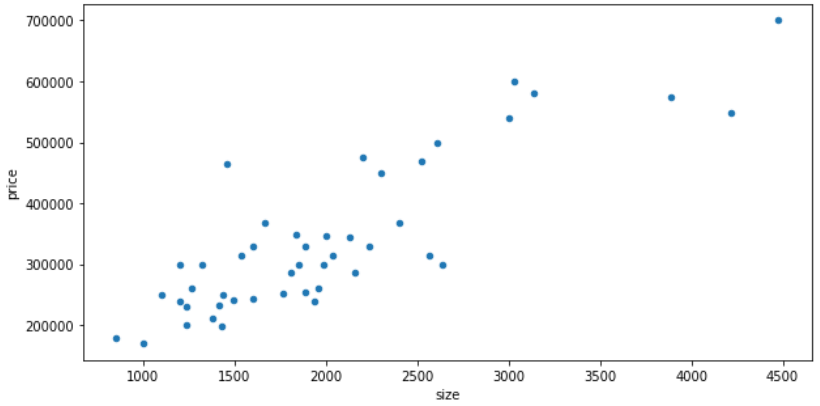


X	Y
2	4
7	19
-5	-17
10	28
15	43

Value I want to predict

- Gained an experience on data

Hypothesis function



ML

Hypothesis function
(e.g.: y=3x-2)

- Knowledge of how output varies with input
- Knowledge is gained from previous data

Analogous to how humans learn

- E.g.,: A baby learns to walk by falling down many times (experience)

Previously,

X	Y
2	4
7	19
-5	-17
10	28
15	43

$$Y=3X-2$$

X	Y
2	4.3
7	19.1
-5	-17.5
10	27.8
15	43.12

Prediction 1	Prediction 2
4	3.1
19	17.32
-17	-11.85
28	31.21
43	48.29

$$h_\Theta(x) = \Theta_0 + \Theta_1 x$$

or

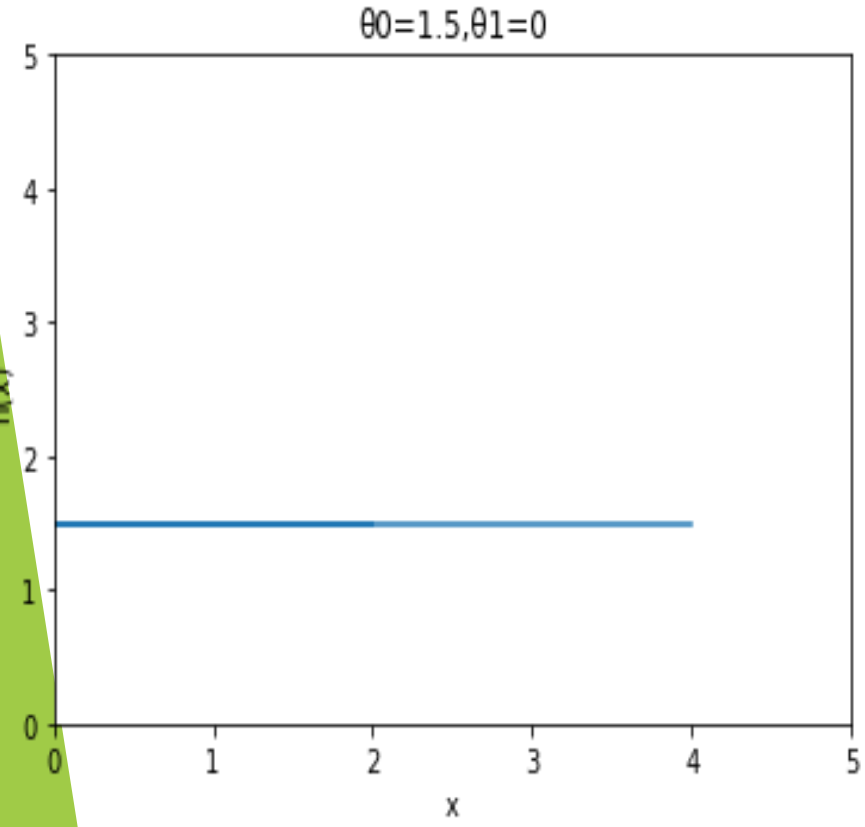
$$h_w(x) = w_0 + w_1 x$$

COSMIC SKILLS

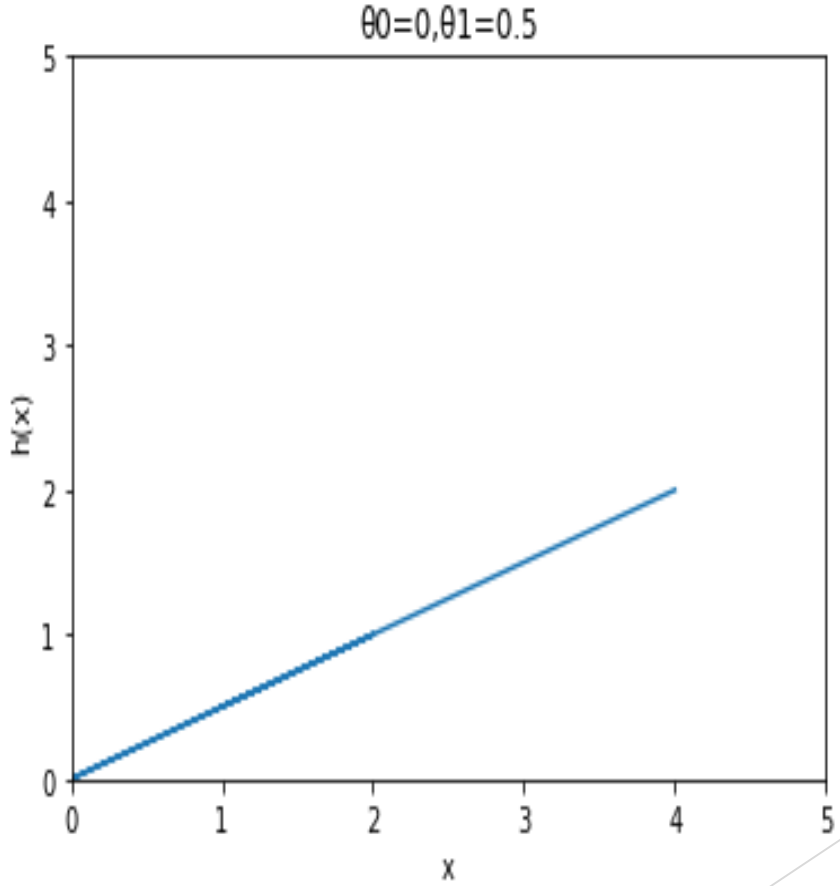
How To Choose Parameters???....

$$h_\Theta(x) = \Theta_0 + \Theta_1 x \qquad h_\Theta(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2$$

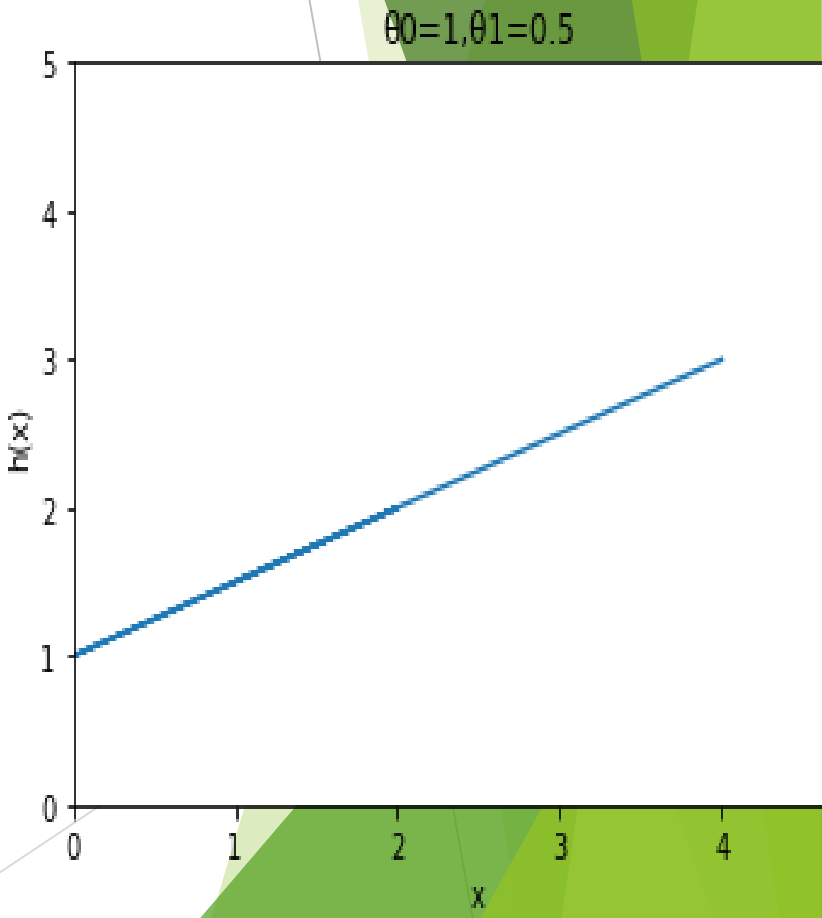
$\Theta_0=1.5, \Theta_1=0$



$\Theta_0=0, \Theta_1=.5$



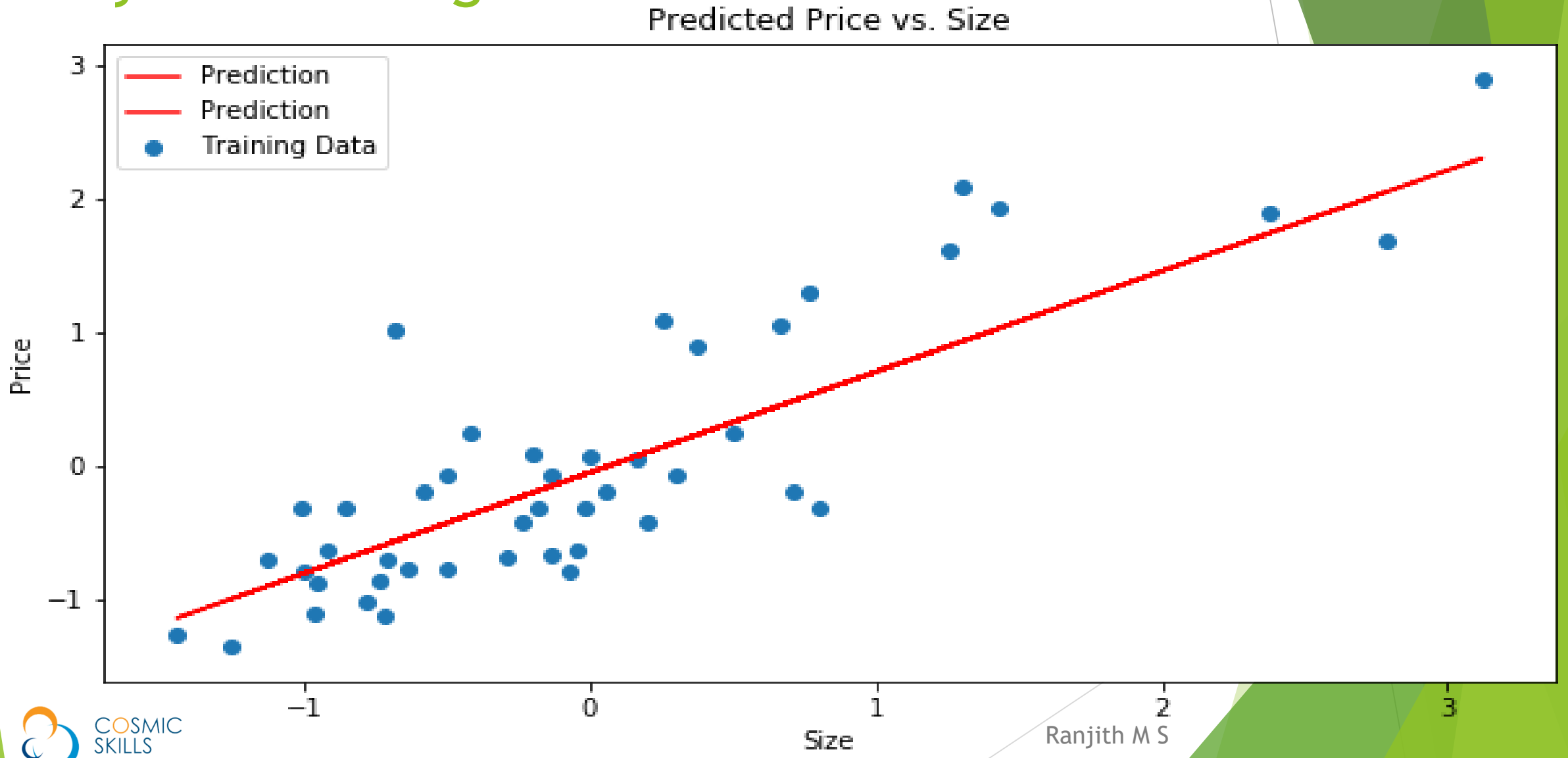
$\Theta_0=1, \Theta_1=.5$



COSMIC SKILLS

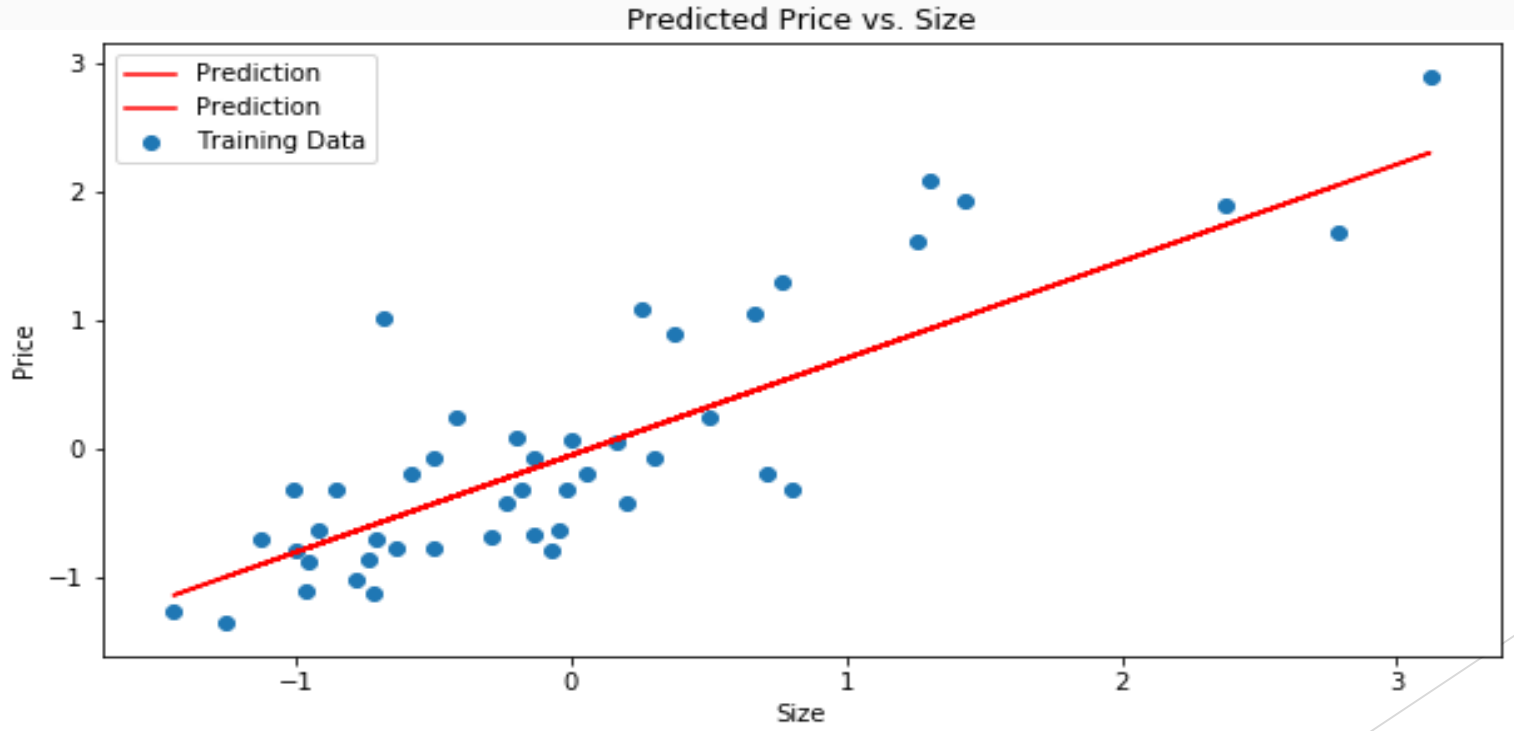
Ranjith M S

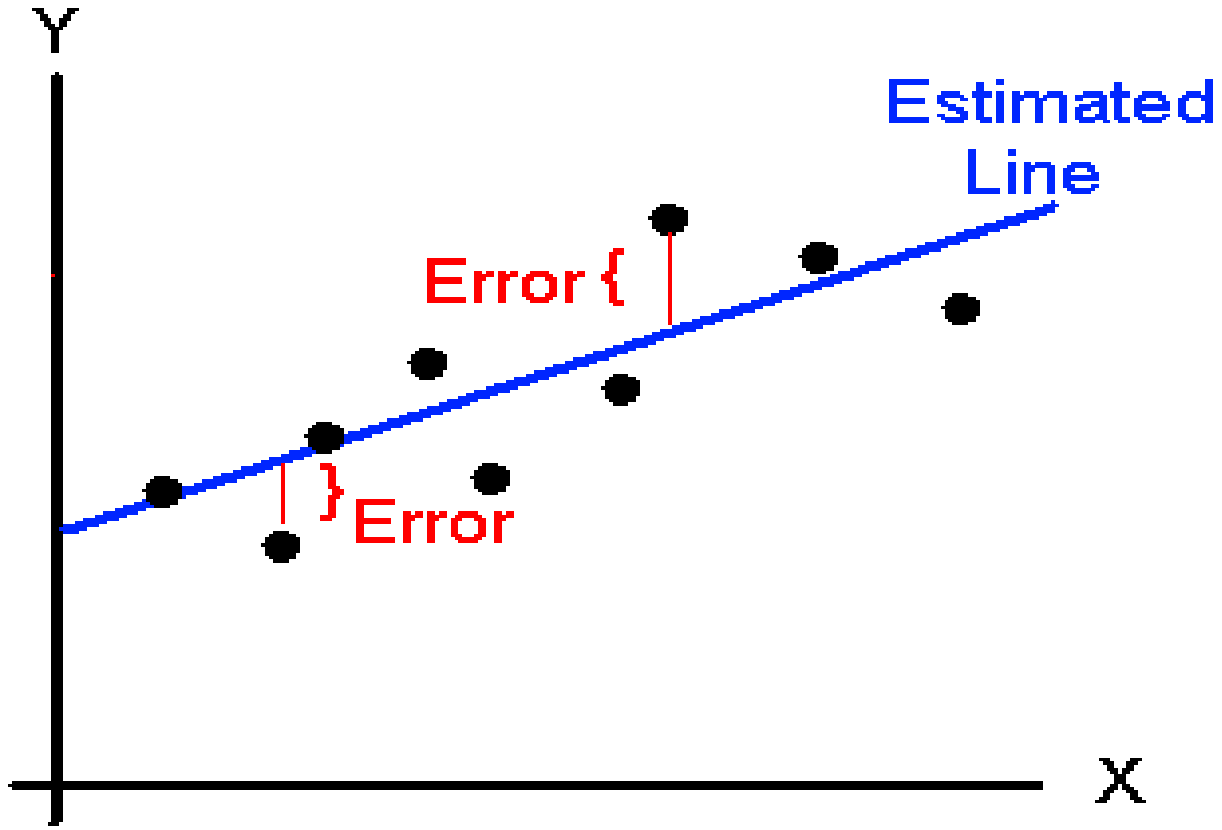
Choose parameters so that h(x) is close to y for training data



Cost function [Squared error function]

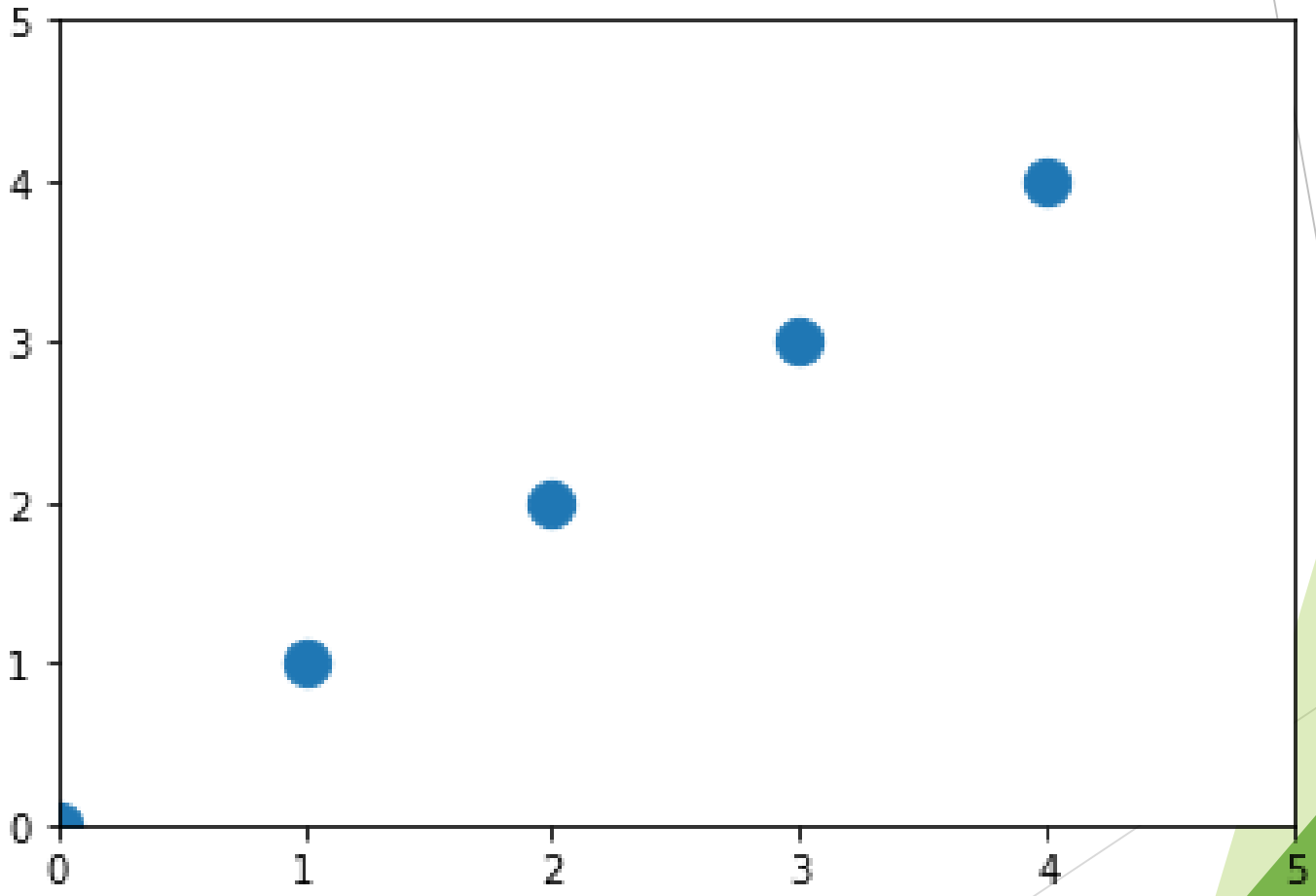
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$





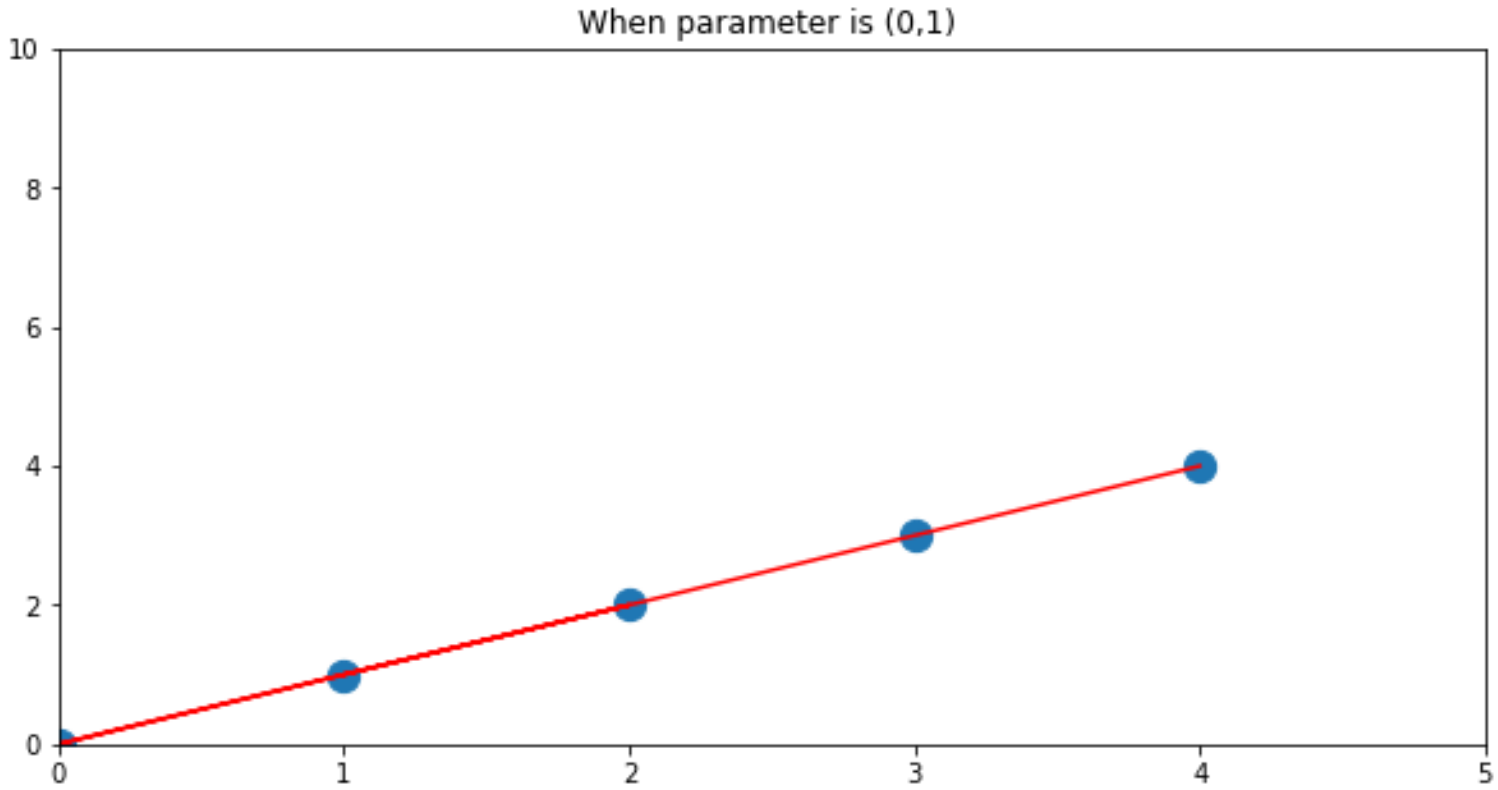
Cost/Loss Function

Let x=[0,1,2,3,4]
y=[0,1,2,3,4]



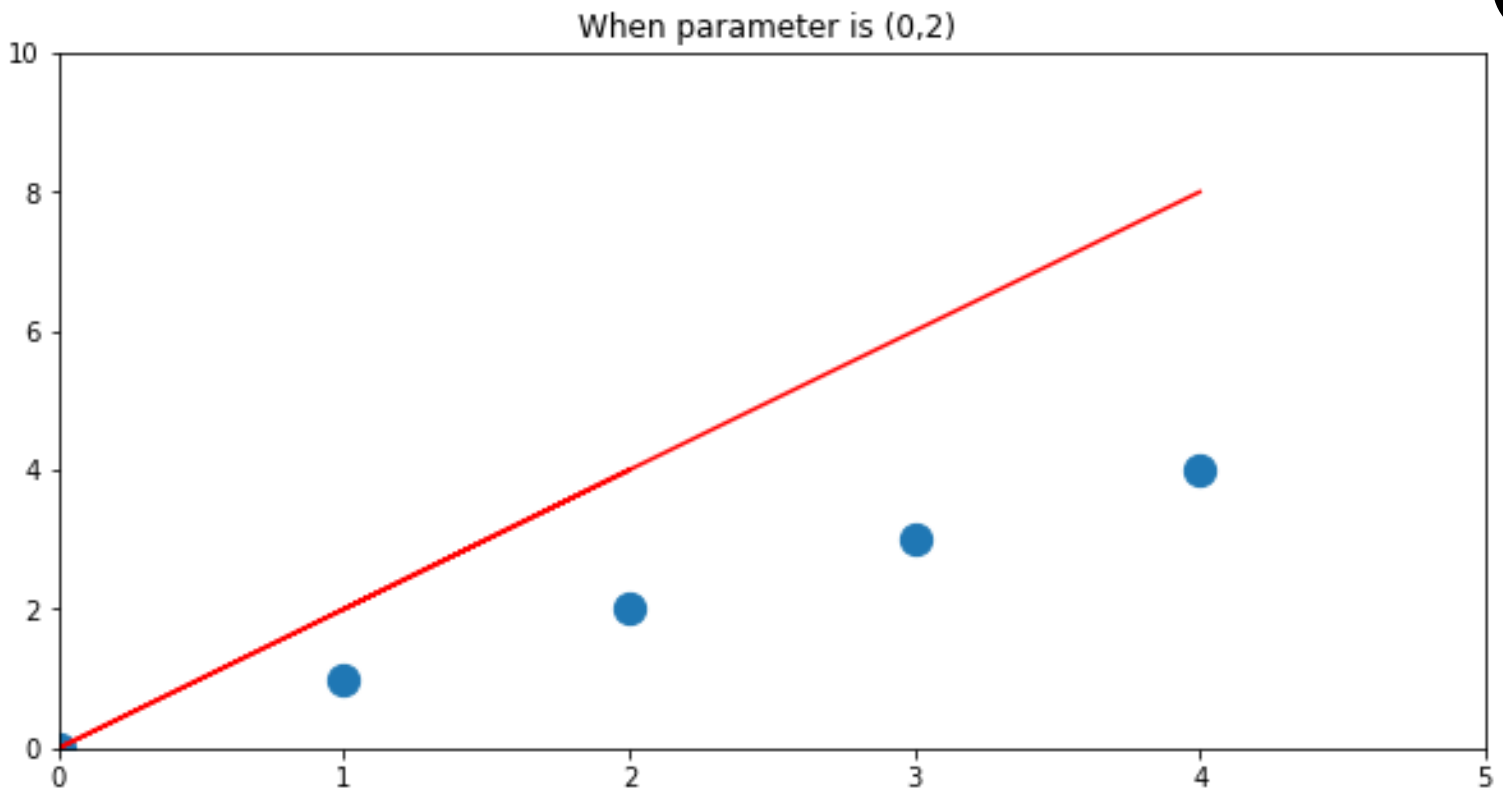
COSMIC SKILLS

Let us assume $\Theta_0=0$
What is the cost when $\Theta_1=1$???....



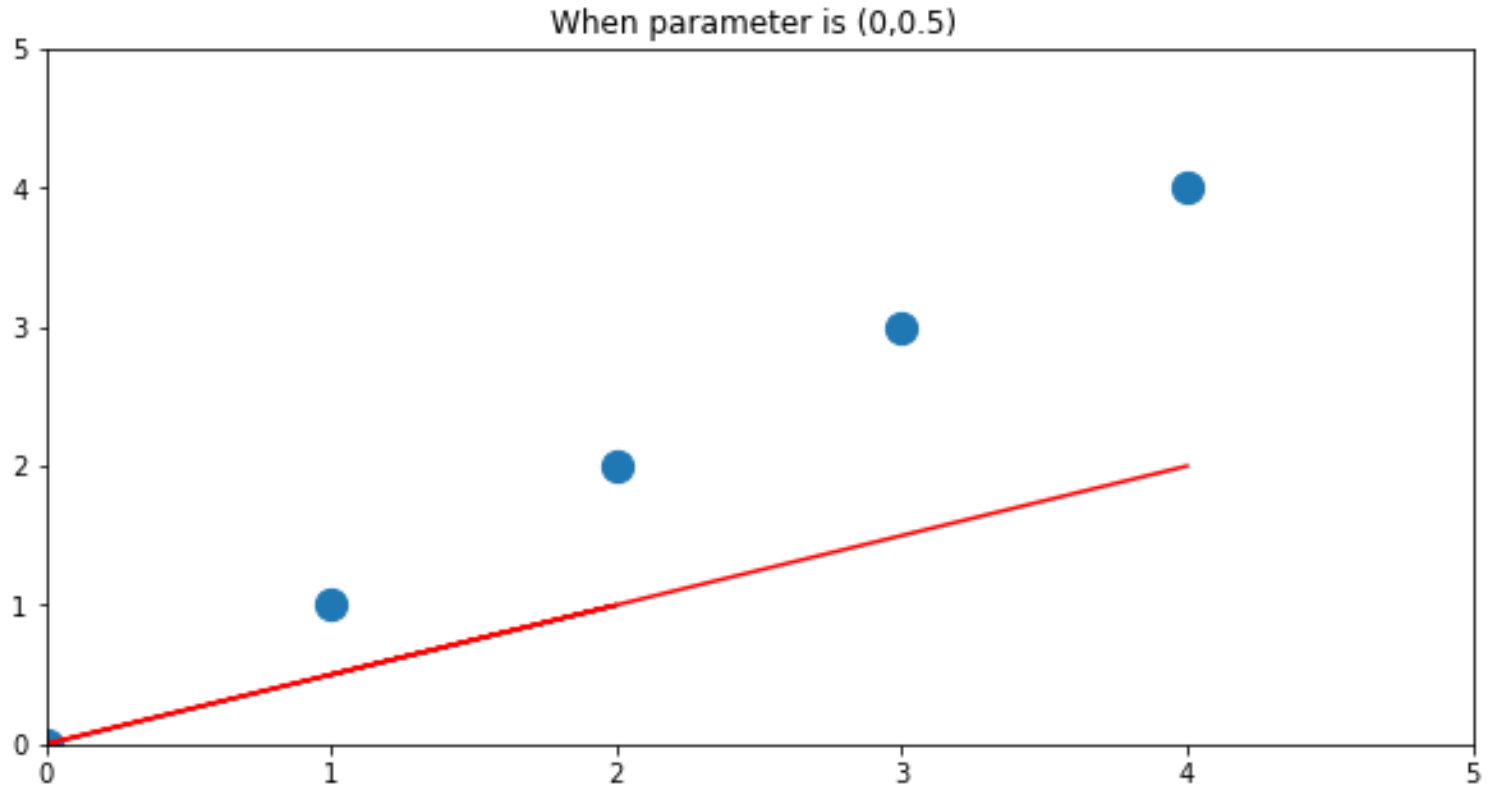
Cost = 0

What is the cost when Θ_1=2 ???....



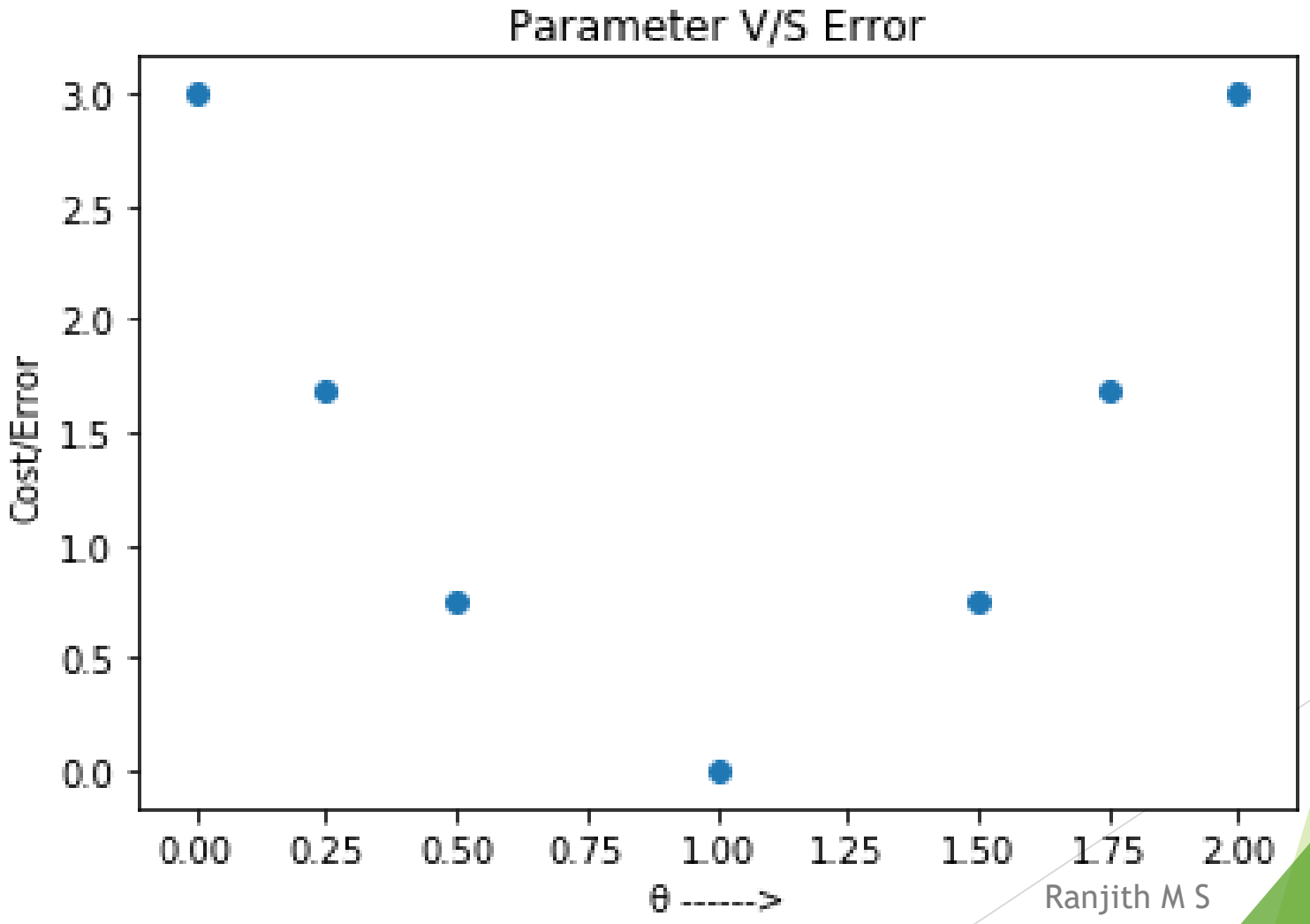
Cost = 3.0

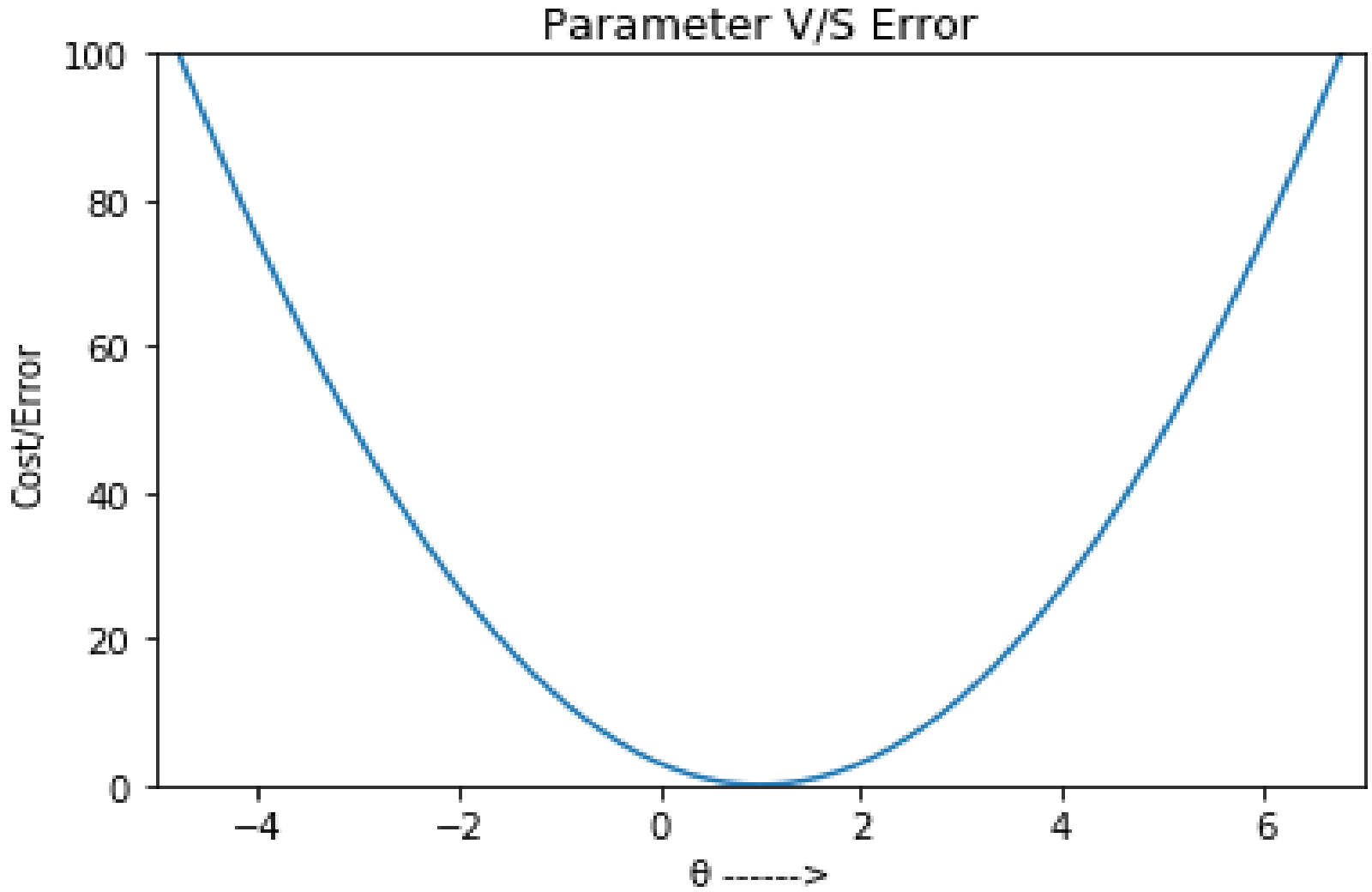
What is the cost when Θ_1=0.5 ???....



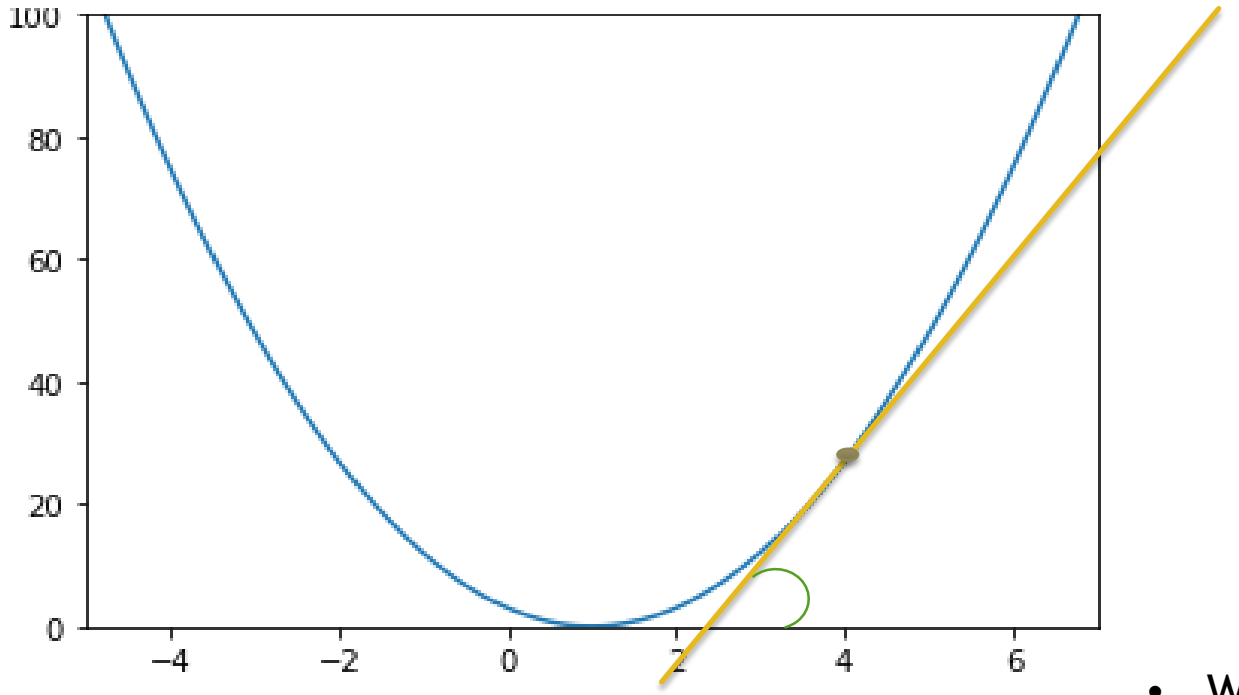
Cost =0.75

How Error varies with θ




Parameter V/S Error
100
80
60
40
20
0
Cost/Error
−4
−2
0
2
4
6
θ ------>

Derivative of a function

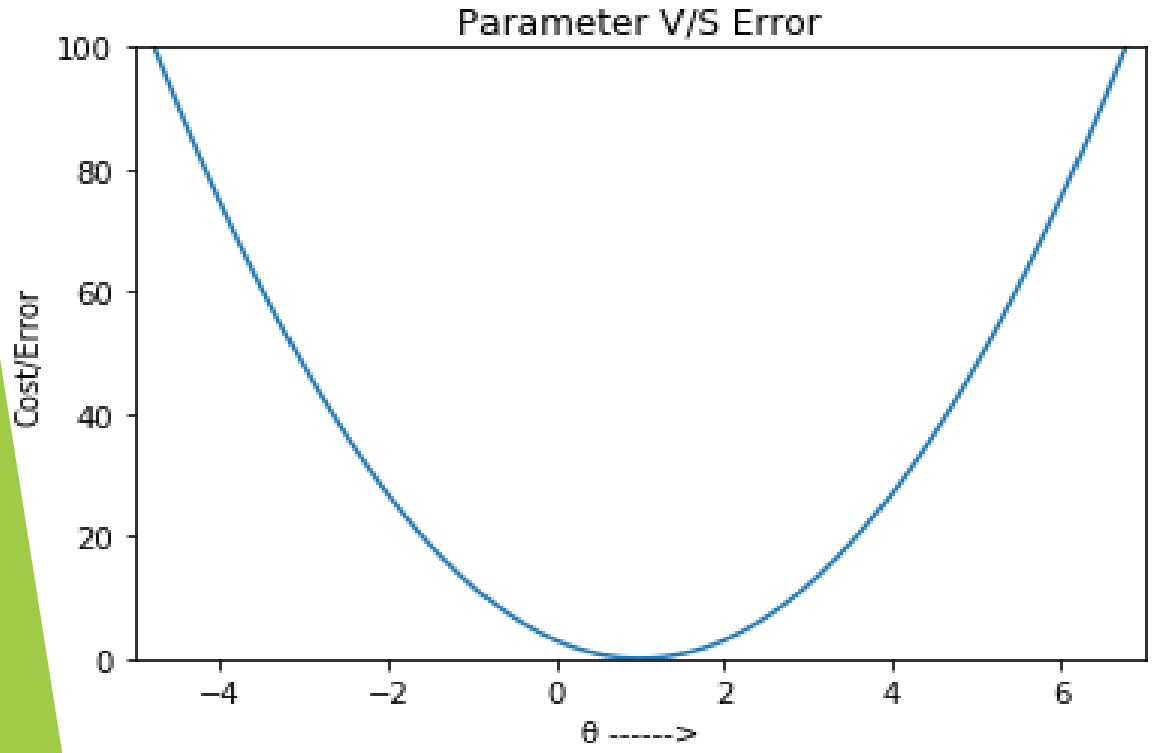


Positive if increasing
Negative if decreasing

- Write a tangent
- Angle of tangent with x-axis
- tan of the angle

Gradient Descent

- It is a method to find the values of Θ such that J(Θ) will be minimum



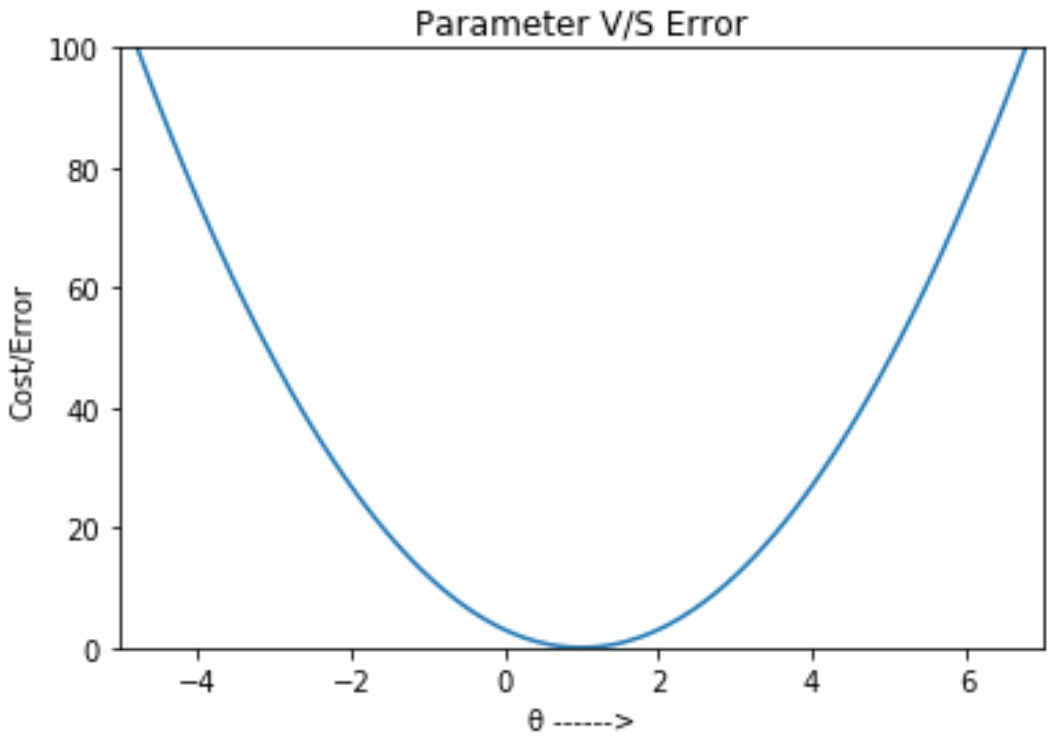
Gradient Descent

- What is the derivative of an increasing Function ??....
- Positive

- What is the derivative of an decreasing Function ??....
- Negative

Gradient descent

- Initialize the weights randomly
- Update the weights based on the derivative (i.e, derivative of cost function w.r.t parameter)
- Decrease the initialized/previous weight if derivative is positive
- Increase the initialized/previous weight if the derivative is negative
- Continue the process until you reach minima



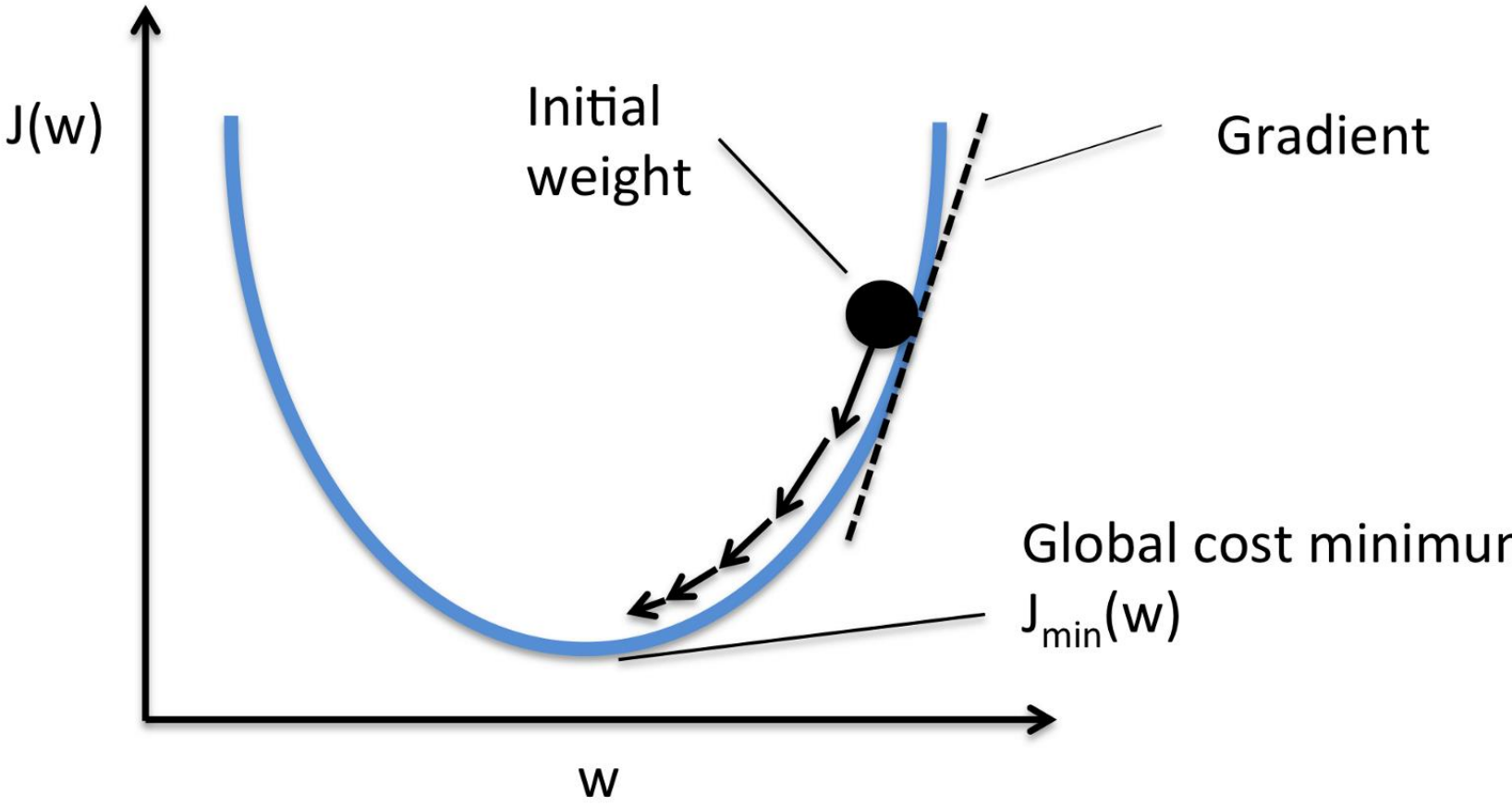
COSMIC SKILLS

Gradient descent

Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

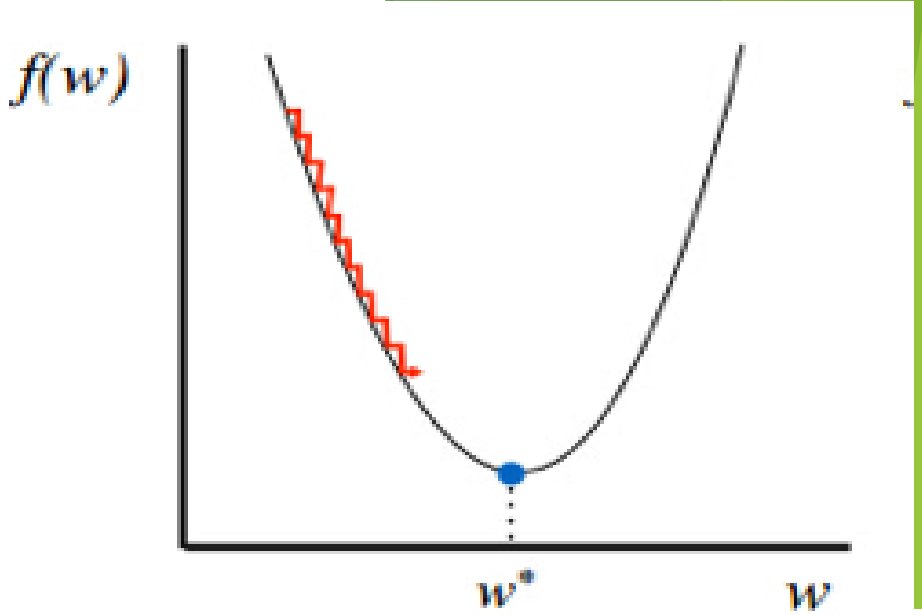
}



The size of each step is determined by the parameter α, which is called the learning rate.

Debugging gradient descent

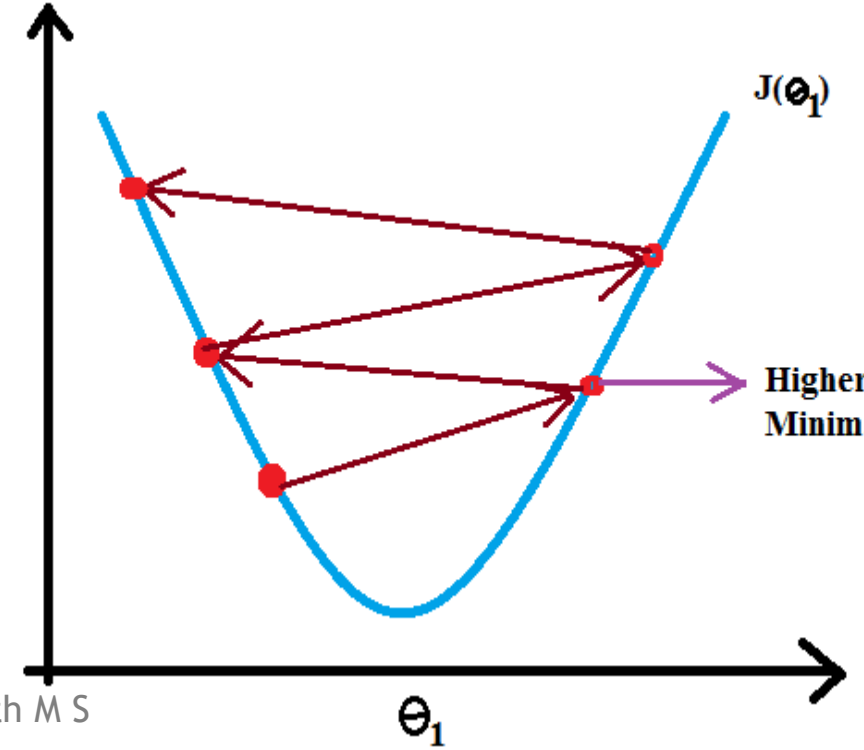
- If α is too small then the gradient descent can be slow
- If α is too large then the gradient descent may overshoot the minima and may fail to converge



Repeat until convergence {

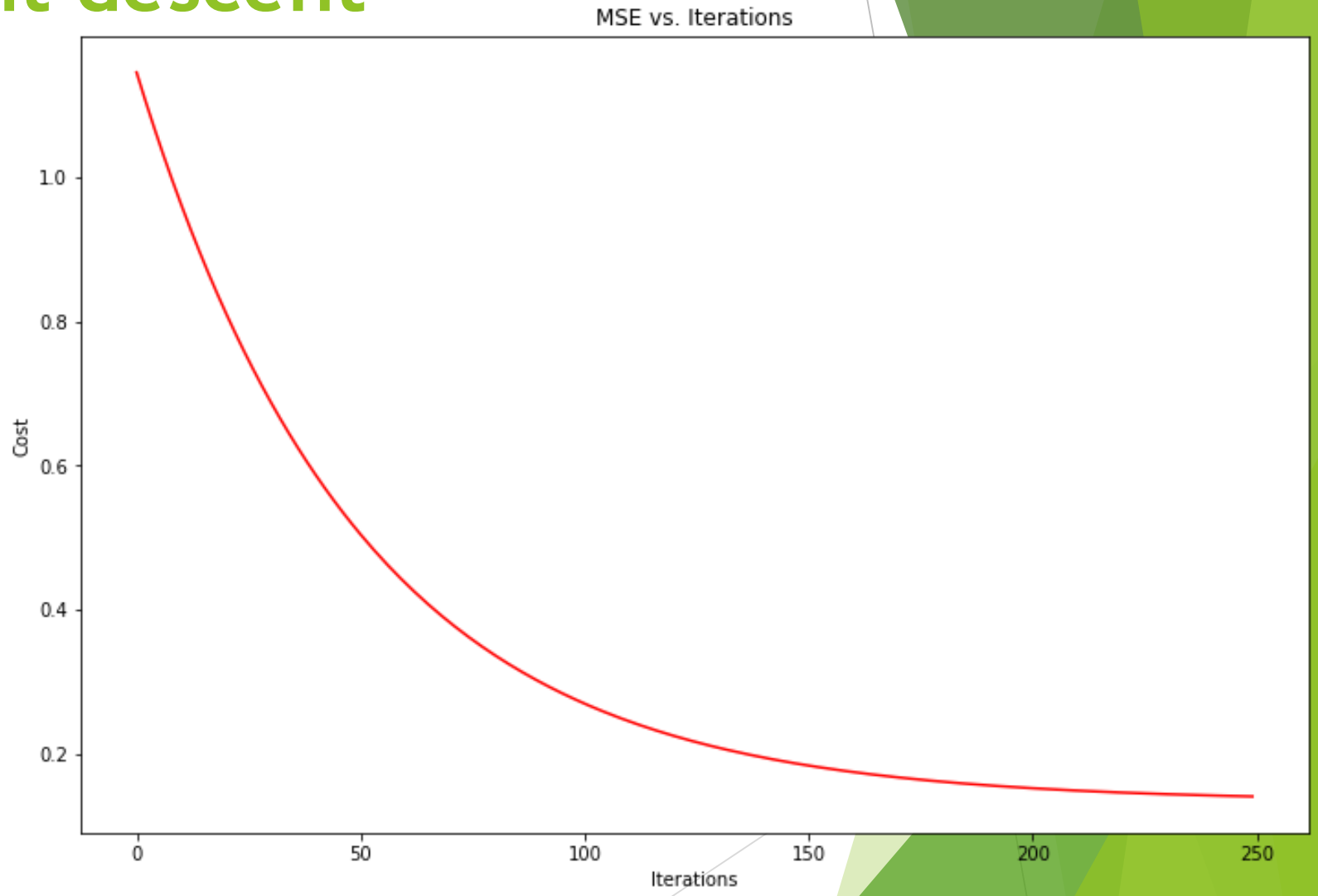
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}



Debugging gradient descent

- Now plot the cost function, J(θ) over the number of iterations of gradient descent. If J(θ) ever increases, then you probably need to decrease α.



Feature Scaling

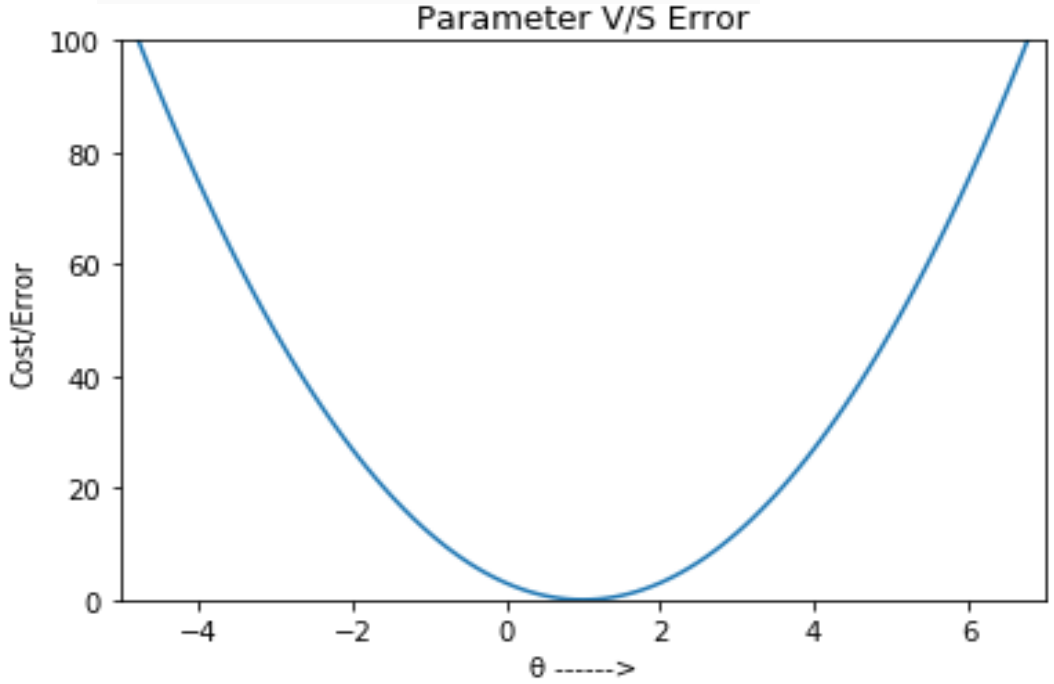
$$x_i := \frac{x_i - \mu_i}{s_i}$$

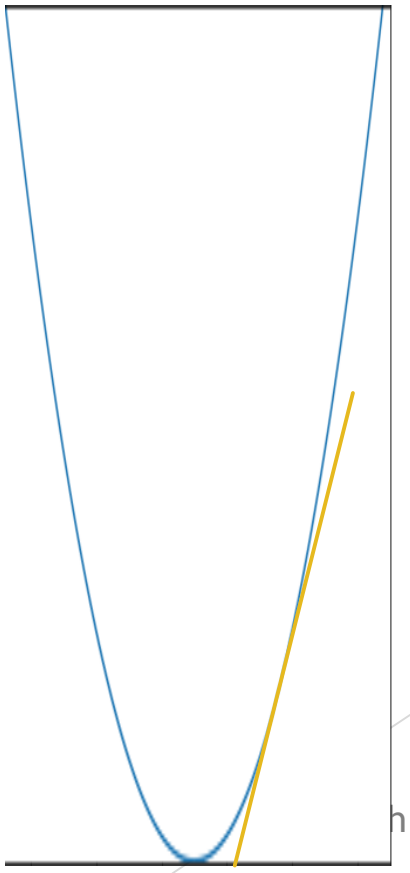
Mean =0
Sd = 1

Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

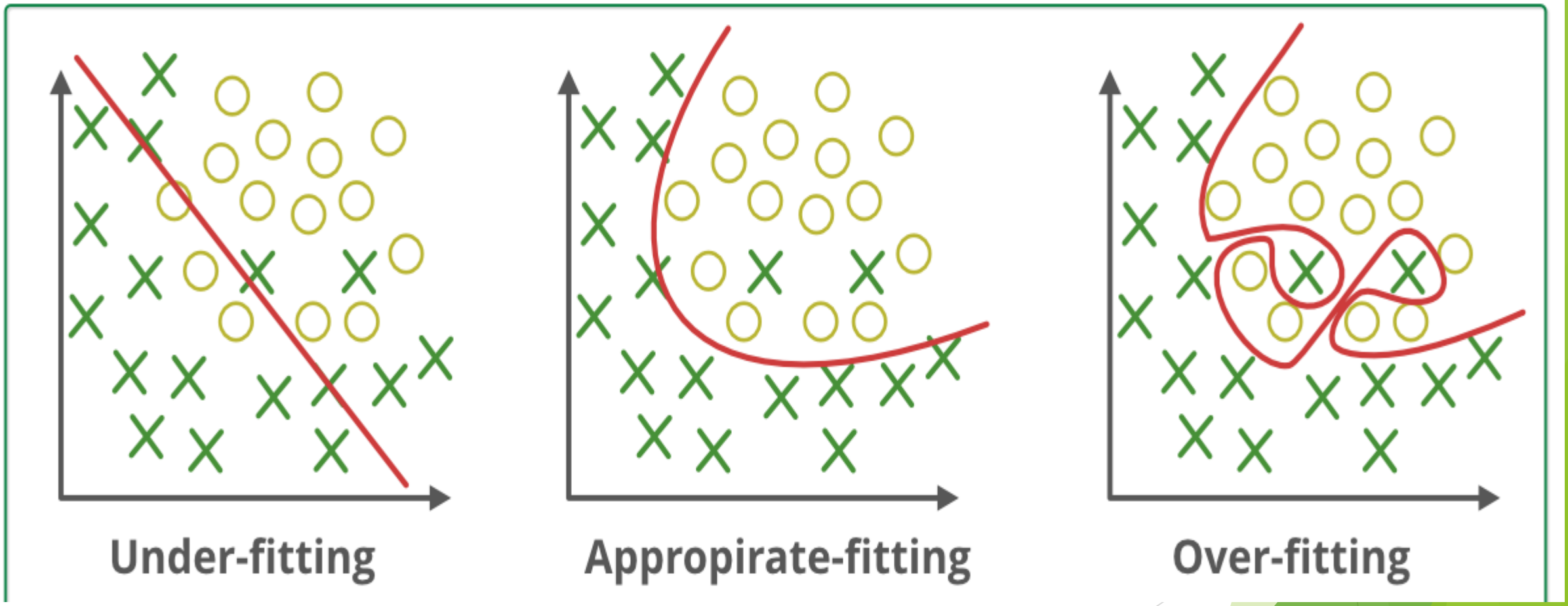




X	Y
2	4.3
7	19.1
-5	-17.5
10	27.8
15	43.12

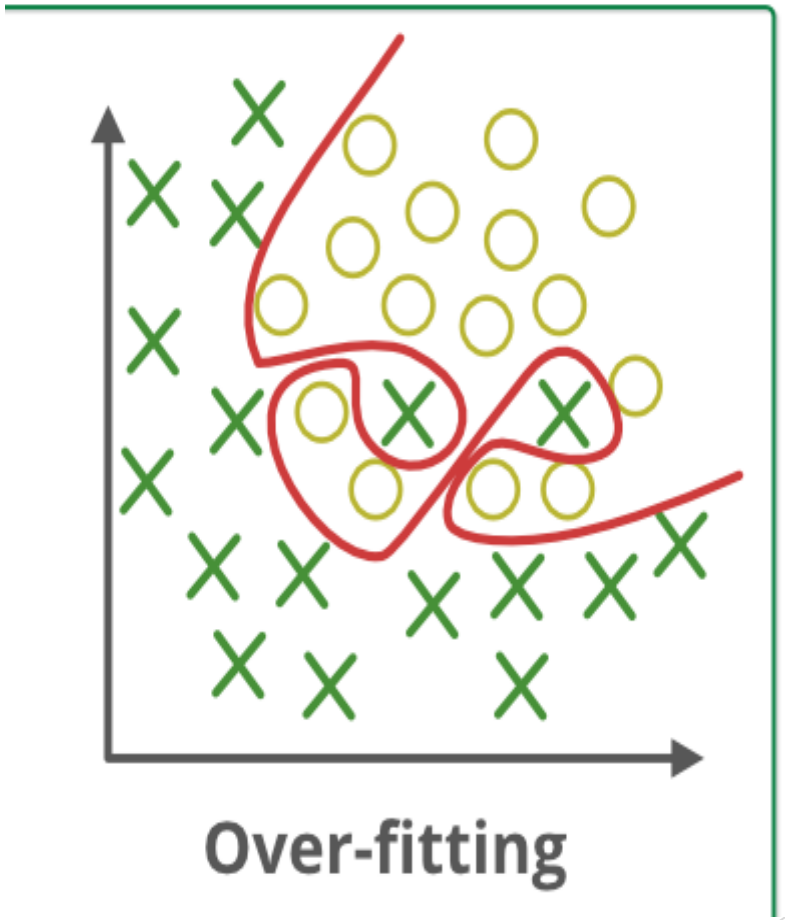
COSMIC SKILLS

How to choose the curve??.....



Ways to reduce overfitting

- Dropout
- Regularization



Training a model