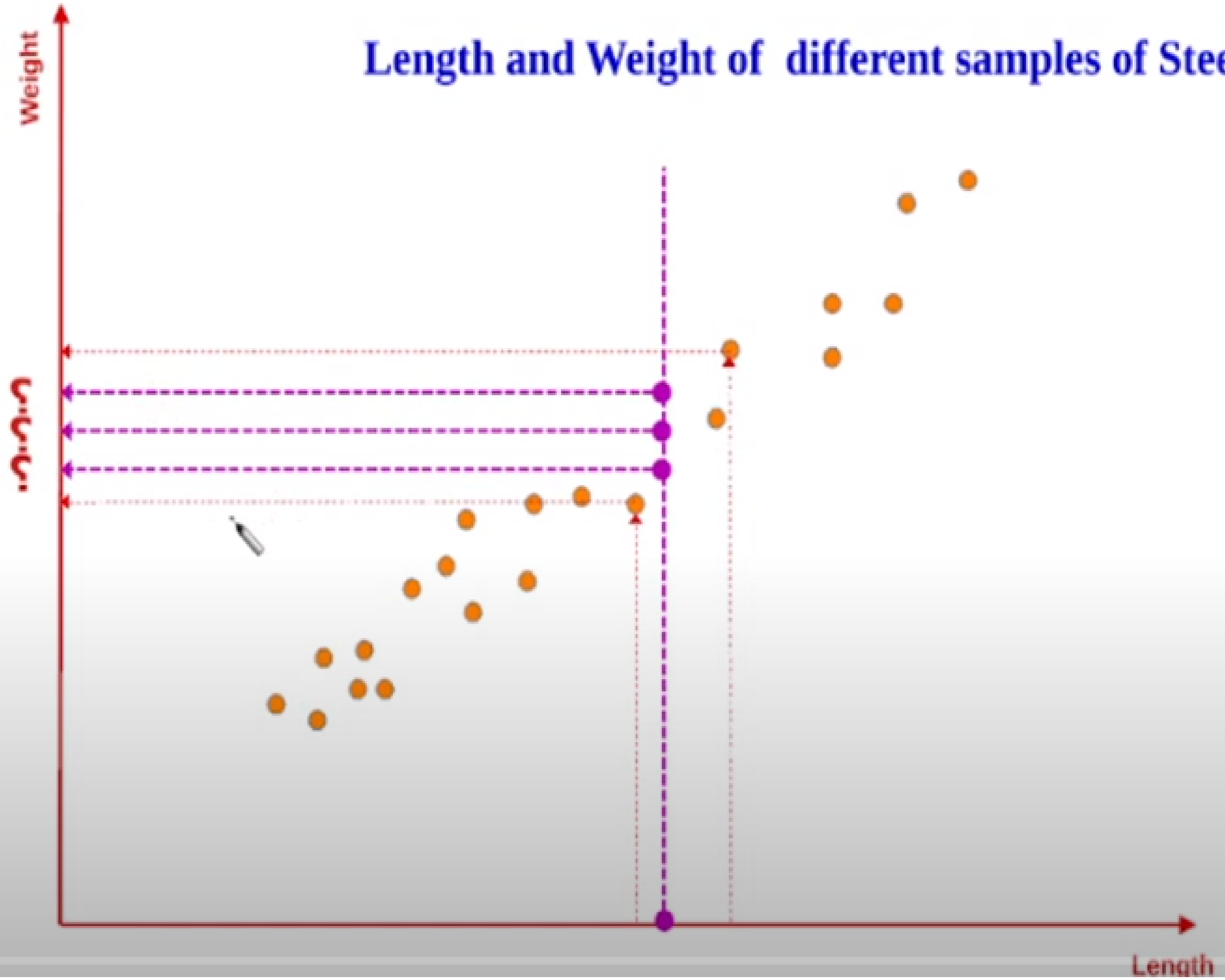
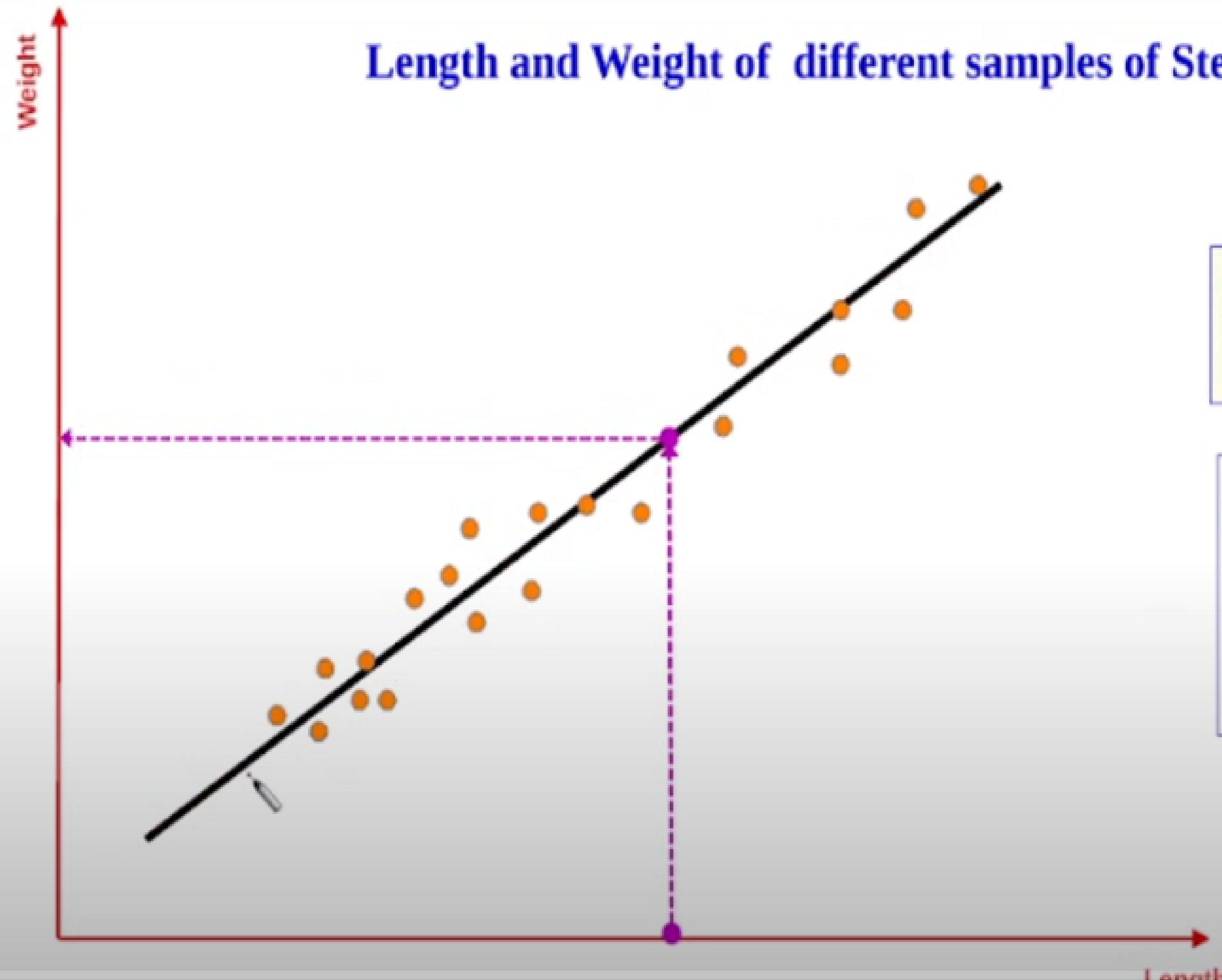
Length and Weight of different samples of Steel Bars



Model
I/P: Length
O/P: Weight

Length and Weight of different samples of Steel Bars



Model
I/P: Length
O/P: Weight

***Model* (Line)**

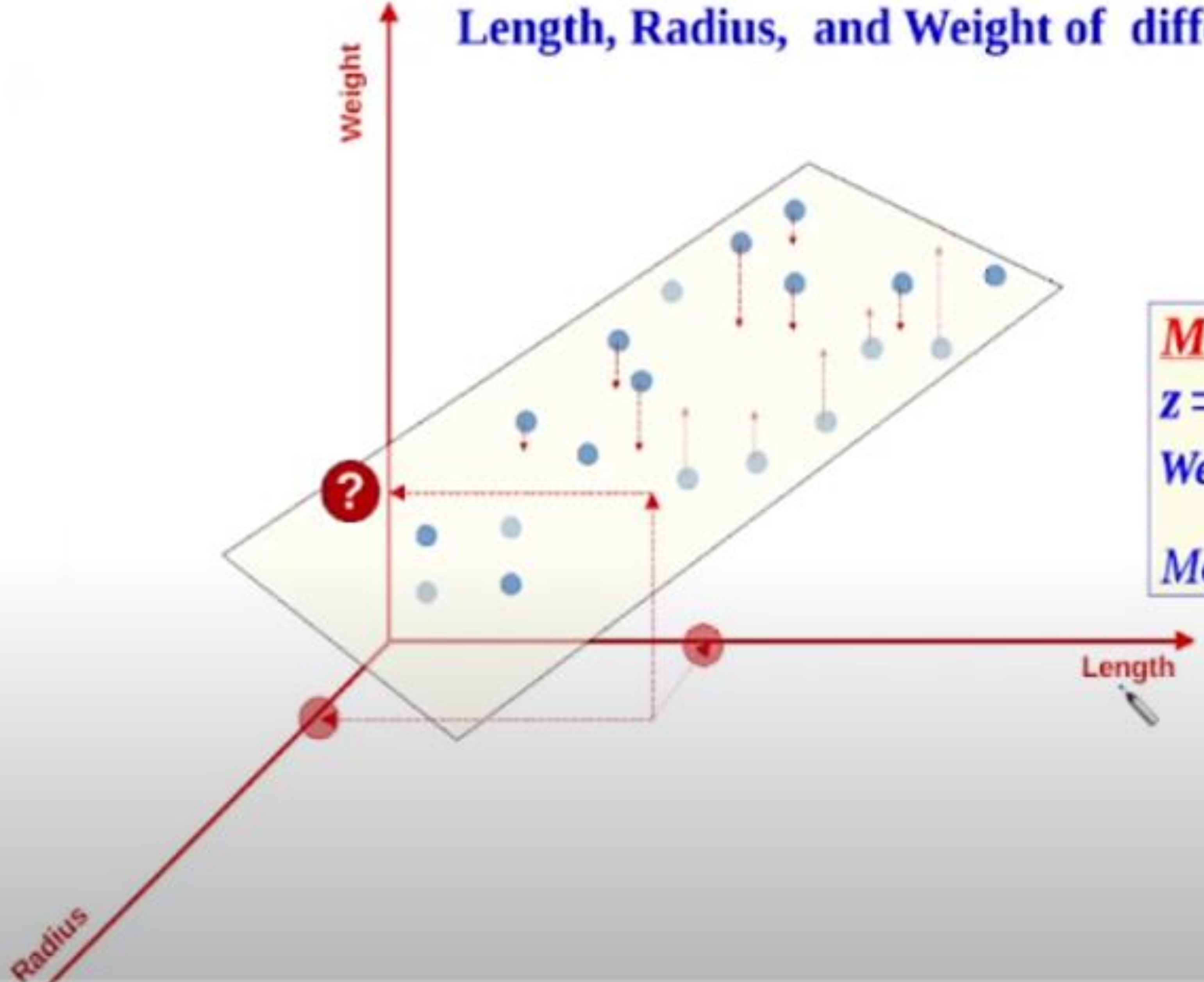
$y = a\,x + b$

$Weight = a * Length + b$

Model Parameters : $\boldsymbol{a}$, $\boldsymbol{b}$

Regression – Two Dimensional Case

Length, Radius, and Weight of different samples of Steel Bars



Model
I/P: Length, Radius
O/P: Weight

***Model* (Plane)**

$$z = a\,x + b\,y + c$$

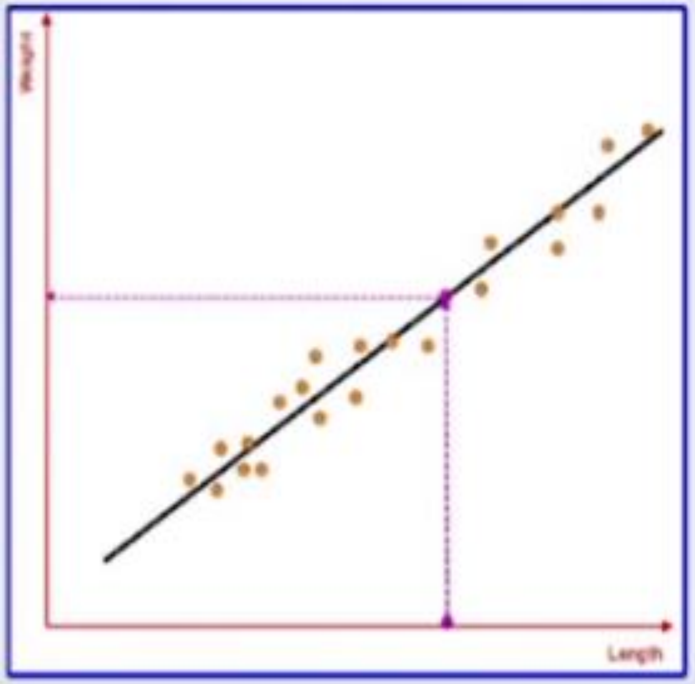
$$Weight = a * Length + b * Radius + c$$

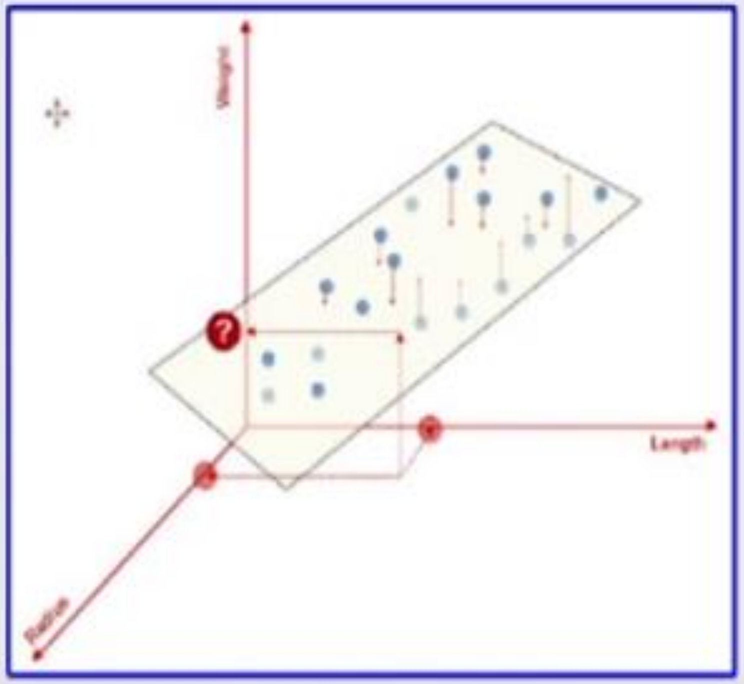
Model Parameters : a, b, c

Input Vector: 1-D

Input Vector: 2-D

Input Vector: d-D





General Case

Input: $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_d \end{pmatrix}$ **Output:** (y)

Model
I/P: Length [x]
O/P: Weight [y]

Model
I/P: Length, Radius [x,y]
O/P: Weight [z]

Model
I/P: $[X_1, X_2, X_3, \ldots X_d]$
O/P: [Y]

***Model* (Line)**

$y = a\,x + b$

$Weight = a * Length + b$

***Model* (Plane)**

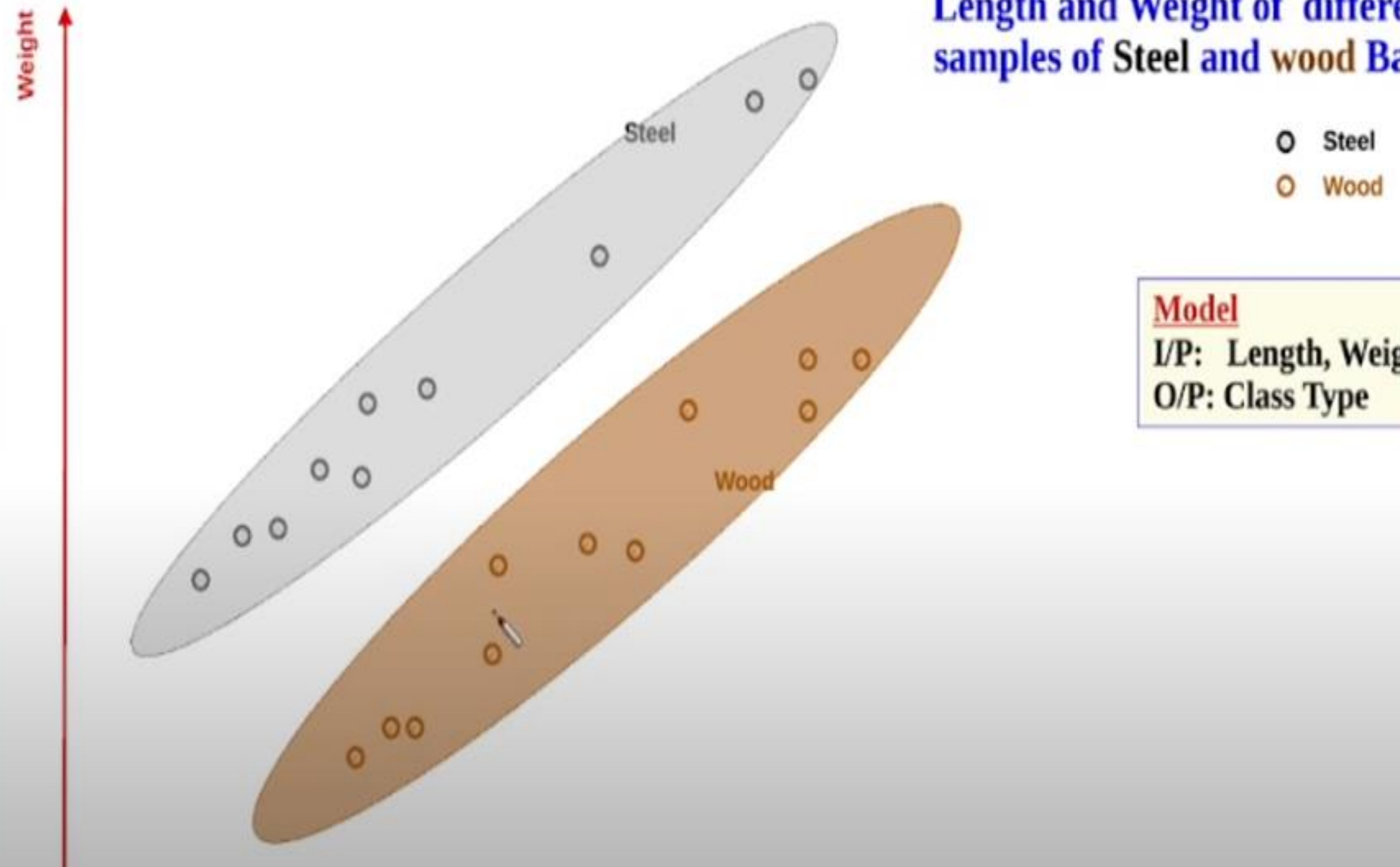
$z = a\,x + b\,y + c$

$Weight = a * Length + b * Radius + c$

***Model* (Hyperplane)**

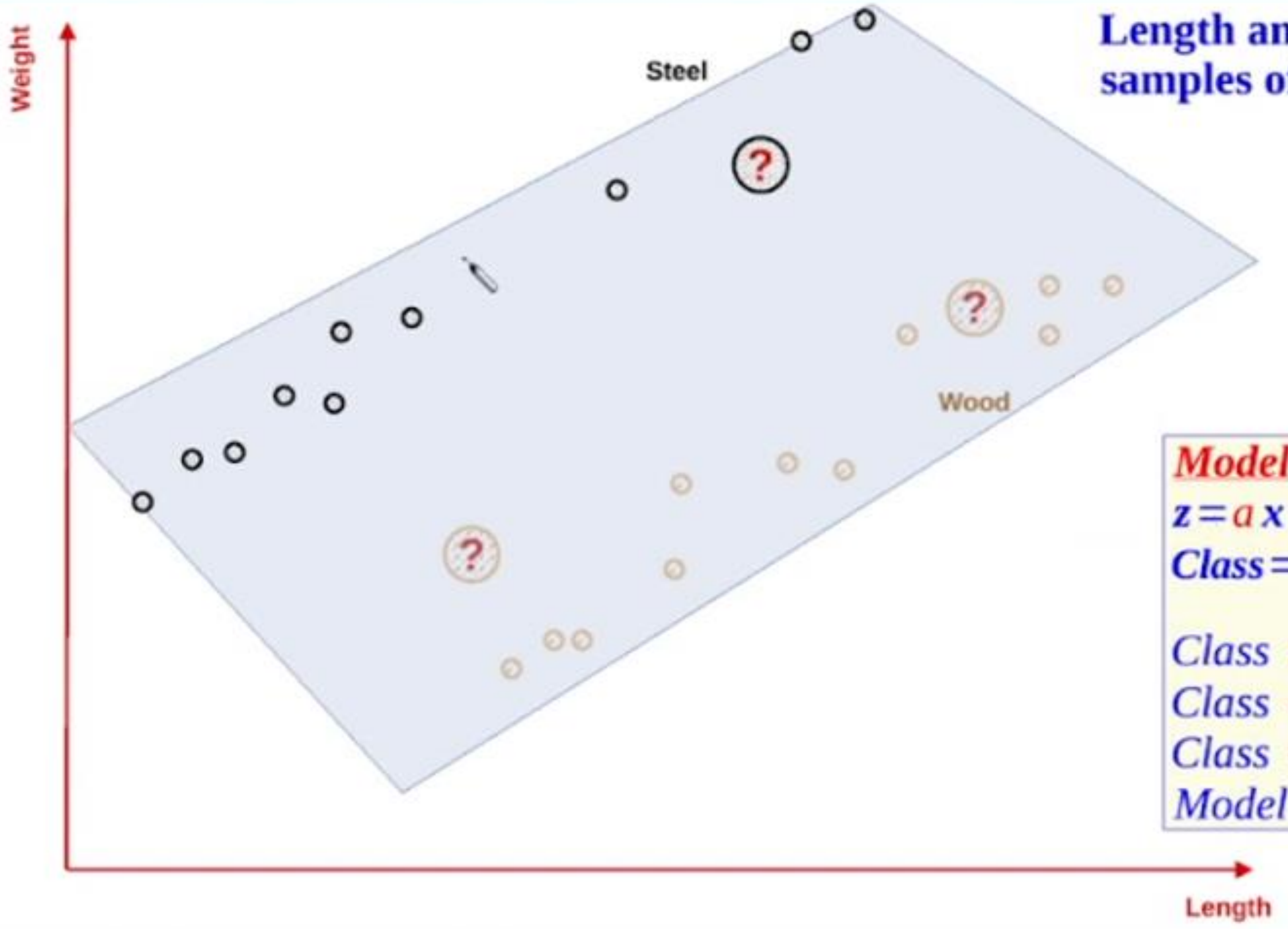
$y = w_1\,x_1 + w_2\,x_2 + w_3\,x_3 + \ldots + w_d\,x_d + w_{d+1}$

Length and Weight of different samples of Steel and wood Bars



Model
I/P: Length, Weight
O/P: Class Type

Classification – Two Dimensional Case



Length and Weight of different samples of Steel and wood Bars

o Steel
o Wood

Model
I/P: Length, Weight
O/P: Class Type

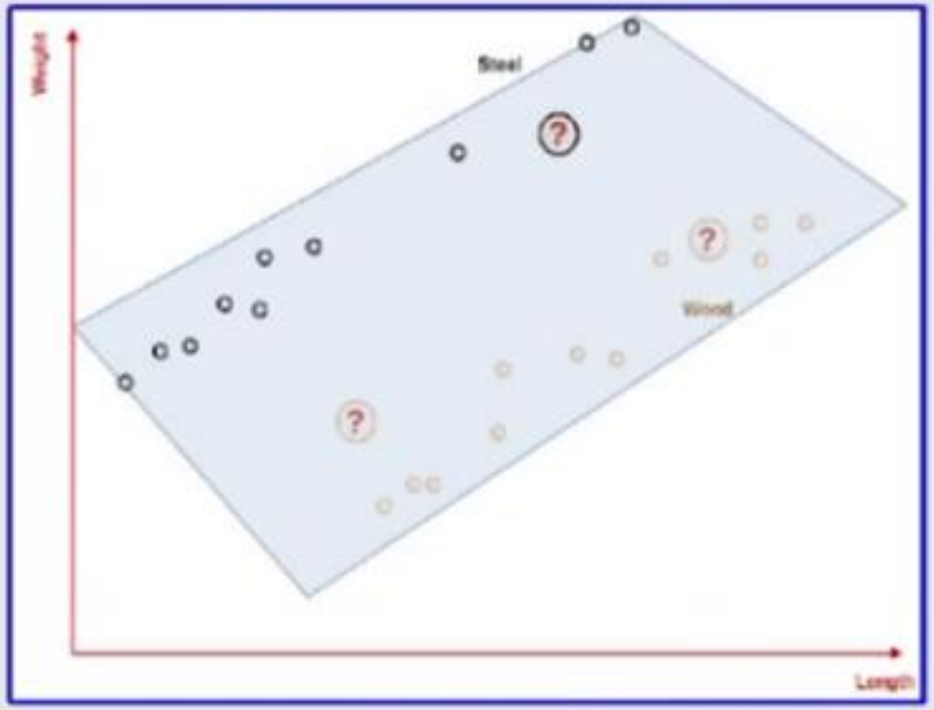
***Model* (Plane)**

$z = a\,x + b\,y + c$

$Class = a * Length + b * Weight + c$

$Class > 0$ Steel
$Class < 0$ Wood
$Class = 0$ Undefined
$Model\ Parameters: a, b, c$

Input Vector: 2-D



Model
I/P: Length, Weight
O/P: Class Type

***Model* (Plane)**

$z = a\,x + b\,y + c$

$Class = a * Length + b * Weight + c$

$Class > 0$ Steel $\quad$ $Class < 0$ Wood

$Class = 0$ Undefined

Input Vector: d-D

General Case

Input: $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_d \end{pmatrix}$ **Output:** (y)

Model
I/P: $[X_1, X_2, X_3, \ldots . X_n]$
O/P: $[y]$

***Model* (Hyperplane)**

$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_n x_d + w_{d+1}$

$y > 0 \Rightarrow$ Class 1 $\quad$ $y < 0 \Rightarrow$ Class 2 $\quad$ $y = 0 \Rightarrow$ Undefined

$Model\ Parameters: w_1, w_2,, w_3, \ldots,, w_{d+1}$

Same Equation for Linear Regression and Classification

Model

I/P: $(x_1 \quad x_2 \quad x_3 \quad ..... \quad x_n)$

O/P: (y)

General Case - Regression

Model

$$y = w_1 x_1 + w_2 x_2 + w_2 x_2 + \ldots + w_n x_d + b$$

Model Parameters : $w_1, w_2, w_3, \ldots w_d, b$

b is called **"bias"** term as it is independent of x

Input: $\underline{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \ldots \\ \ldots \\ x_d \end{pmatrix}$ $\quad \underline{W} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ \ldots \\ \ldots \\ W_d \end{pmatrix}$ Output: (y)

General Case - Classification

Model

$$y = w_1 x_1 + w_2 x_2 + w_2 x_2 + \ldots + w_d x_d + b$$

$y > 0 \Rightarrow$ Class 1 $\quad y < 0 \Rightarrow$ Class 2 $\quad y = 0 \Rightarrow$ Undefined

Model Parameters : $w_1, w_2, w_3, \ldots, w_d, b$

Vector Notation for Linear Regression and Classification (One Sample)

Model

I/P: $(x_1 \quad x_2 \quad x_3 \quad ..... \quad x_n)$ **O/P:** (y)

Model

$y = w_1 x_1 + w_2 x_2 + w_2 x_2 + \ldots + w_d x_d + b$

$y = \underline{W}^T \underline{X} + b$

$$y = \begin{pmatrix} w_1 & w_2 & .... & w_d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_d \end{pmatrix} + b$$

Input: $\underline{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_d \end{pmatrix}$ $\underline{W} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ \cdots \\ \cdots \\ W_d \end{pmatrix}$ **Output:** (y)

Vector Notation for Linear Regression and Classification (One Sample)

Model

I/P: $(x_1 \quad x_2 \quad x_3 \quad \ldots\ldots \quad x_n)$ **O/P:** (y)

Input: $\underline{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \ldots \\ \ldots \\ x_d \end{pmatrix}$ $\underline{W} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ \ldots \\ \ldots \\ W_d \end{pmatrix}$ **Output:** (y)

Model

$$y = w_1 x_1 + w_2 x_2 + w_2 x_2 + \ldots + w_n x_n + b$$

$$y = \underline{W}^T \underline{X} + b$$

$$y = \begin{pmatrix} w_1 & w_2 & \ldots & w_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix} + b \Rightarrow y = \begin{pmatrix} b & w_1 & w_2 & \ldots & w_d \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \ldots \\ x_d \end{pmatrix} \Rightarrow \boxed{y = \underline{W}^T \underline{X}}$$

Note:
"b" may be added at begin of the vector
or at end of the vector
Same for "1" added to input vector

$y = \underline{W}^T \underline{X}$
Where $\underline{X}$ is Augmented vector
$\underline{W}$ is $d+1$ dimension

Model Parameter is $\underline{w}$

Matrix Notation for Linear Regression and Classification (N- Samples)

N Training Samples

$$\begin{array}{cccc} X_1 & X_2 & \dots & X_n \\ \begin{pmatrix} x_{11} \\ x_{12} \\ .. \\ x_{1d} \end{pmatrix} & \begin{pmatrix} x_{21} \\ x_{22} \\ .. \\ x_{2d} \end{pmatrix} & \begin{pmatrix} \dots \\ \dots \\ .. \\ \dots \end{pmatrix} & \begin{pmatrix} x_{n1} \\ x_{n2} \\ .. \\ x_{nd} \end{pmatrix} \\ y_1 & y_2 & \dots & y_n \end{array}$$

Remember

$$y = \underline{W}^T \underline{X} = \underline{X}^T \underline{W}$$

In General: $N >>> d$

$$+\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}_{N \times 1} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1d} \\ 1 & x_{21} & x_{22} & \dots & x_{2d} \\ 1 & \dots & \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nd} \end{pmatrix}_{N \times d+1} \begin{pmatrix} 1 \\ w_2 \\ w_2 \\ \dots \\ w_d \end{pmatrix}_{d+1 \times 1}$$

$$\underline{Y} = X\,\underline{W}$$

Where: $X_{N \times d+1}$ is a **matrix**

$\underline{Y}_{N \times 1}$ is labels **vector**

Calculation of Model Parameter "W"

$$\underline{Y} = X\,\underline{W}$$

Where: $X_{N\,x\,d+1}$ is a **matrix** $\Rightarrow$ **Given**

$\underline{Y}_{Nx1}$ is labels **vector** $\Rightarrow$ **Given**

$\underline{W}$ is Model Parameter $\Rightarrow$ **Required**

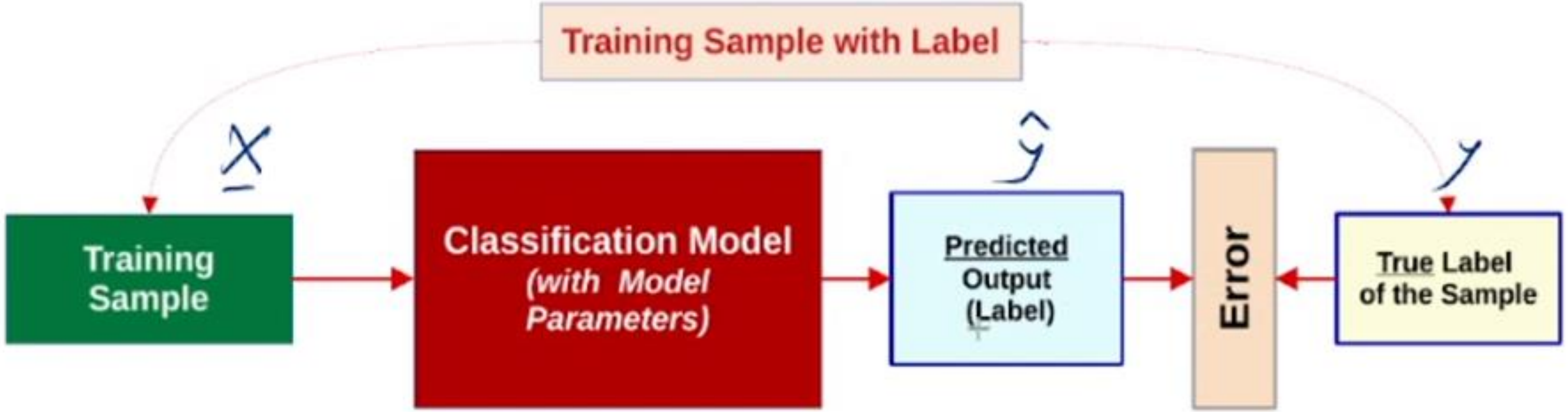
$$\underline{W} = X^{-1}\underline{Y}$$

$$\underline{W} = \left(X_{N\,x\,d}\right)^{-1}\underline{Y}$$

$\boldsymbol{X}_{N\,x\,d}$ is non-square matrix

There is NO Inverse

How to obtain the BEST model parameters (W)



Try to obtain model parameters by minimizing the Error (Loss Function)

Note:

There are three terms that are commonly used interchangeably *(though there is a difference)*

Loss function , **Error** function, and **Objective** function (to be discussed later)

Concept of LOSS Function

Let $\underline{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}_{N \times 1}$ **Original Labels** from Training Samples

Let $\underline{\hat{Y}} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \dots \\ \hat{y}_n \end{pmatrix}_{N \times 1}$ **Predicted Labels** from Model $\hat{Y} = \boldsymbol{X}\,\underline{W}$

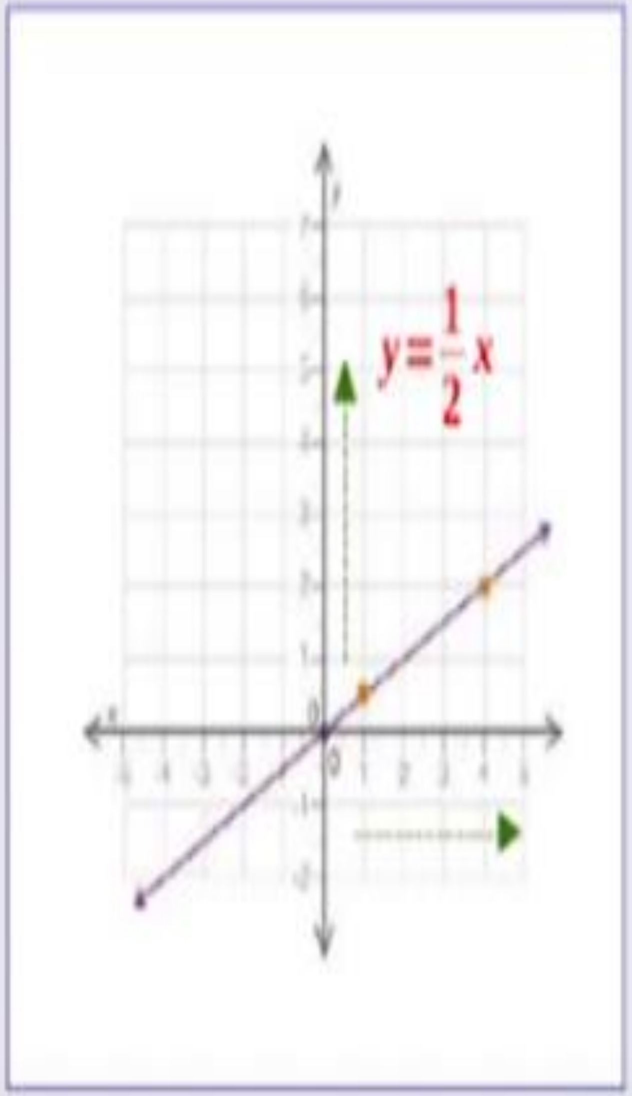
Objective is $\underline{\hat{Y}} = \underline{Y}$

If $\underline{\hat{Y}} \neq \underline{Y}$ $\Rightarrow$ There is an ***Error (Loss)***

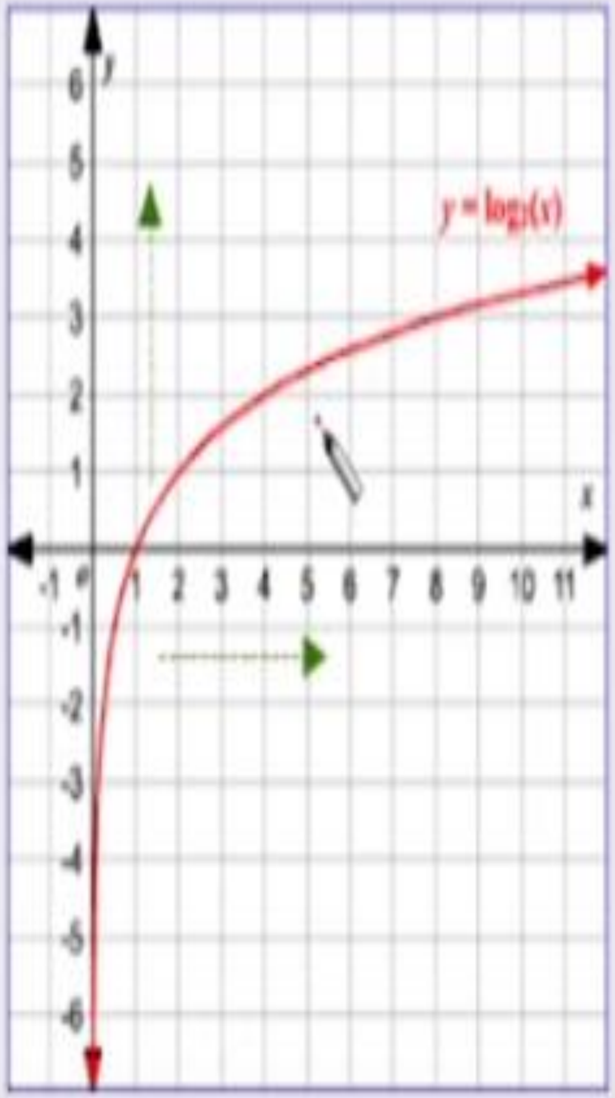
Define **Loss** function as squared distance between $\underline{\hat{Y}}$ and $\underline{Y}$

Remember
$\|a\|_2 = a^T a$

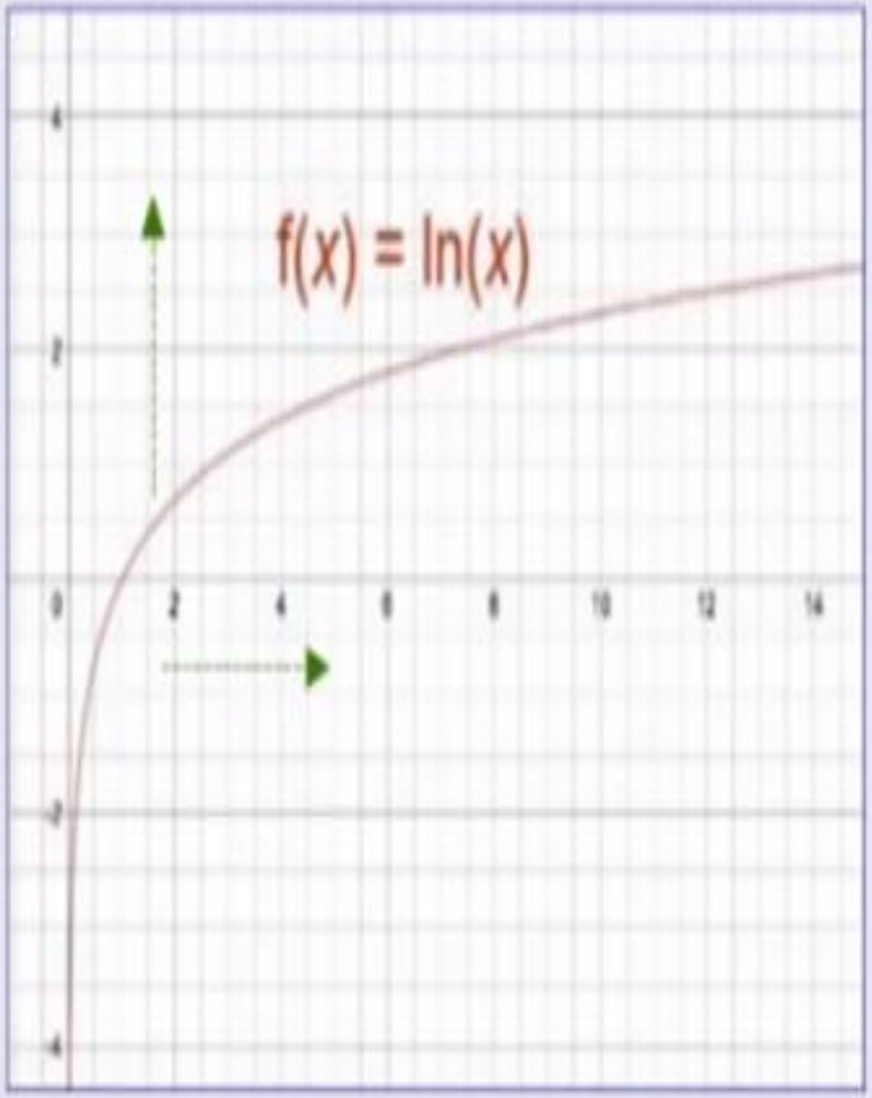
$$Loss = \|\underline{Y} - \underline{\hat{Y}}\|_2 = (\underline{Y} - \underline{\hat{Y}})^T (\underline{Y} - \underline{\hat{Y}})$$



y = 1/2 x

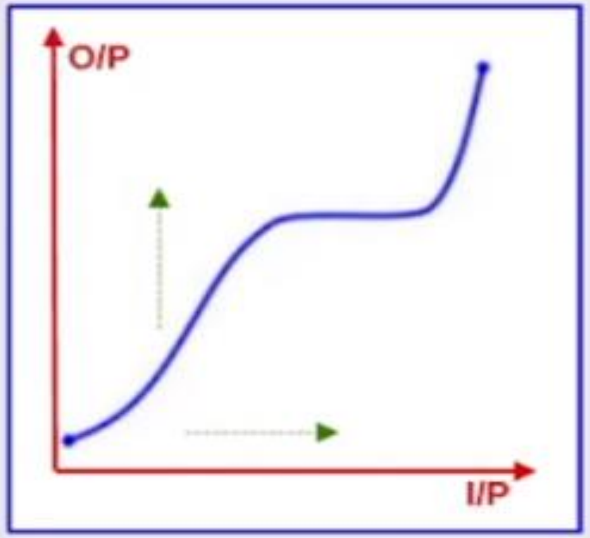


y
x
6
5
4
3
2
1
-1
-2
-3
-4
-5
-6
-1 1 2 3 4 5 6 7 8 9 10 11

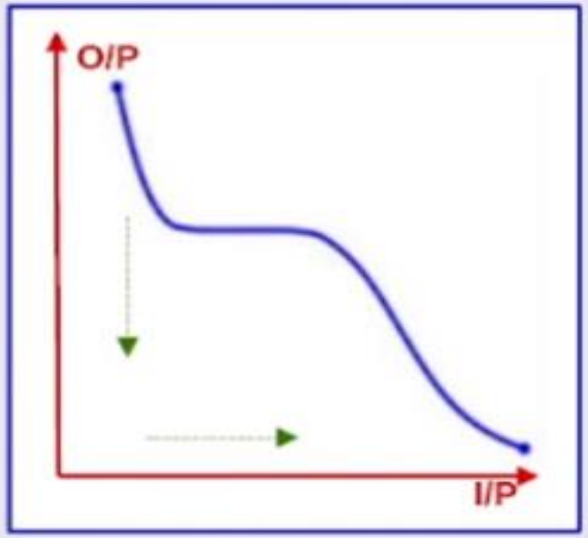


f(x) = ln(x)
0 2 4 6 8 10 12 14

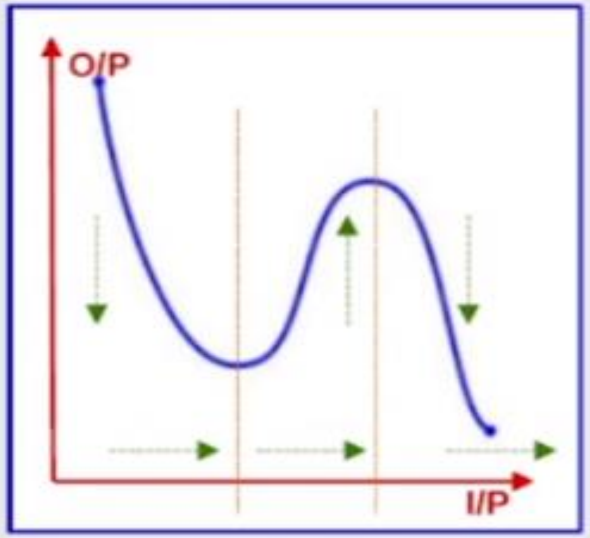
Monotonic Functions



Monotonically Increase



Monotonically Decrease

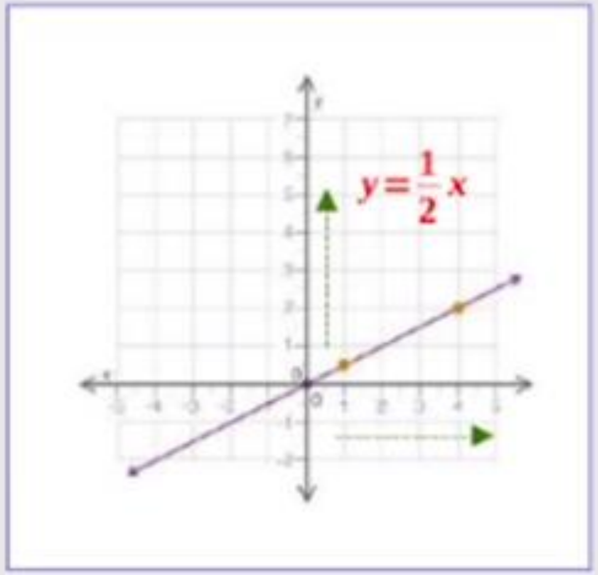


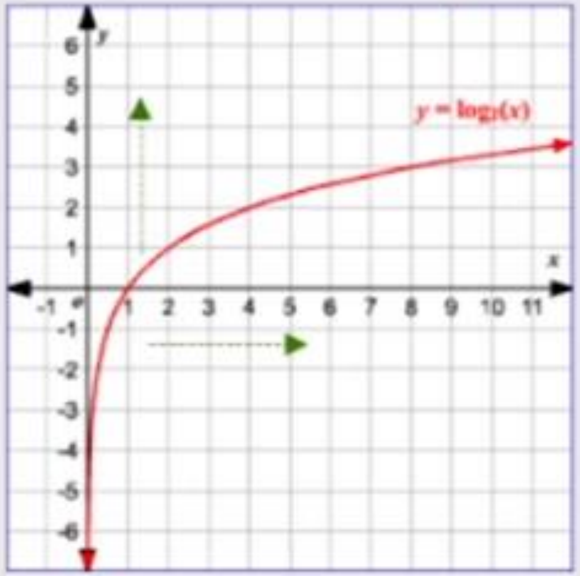
Not Monotonic

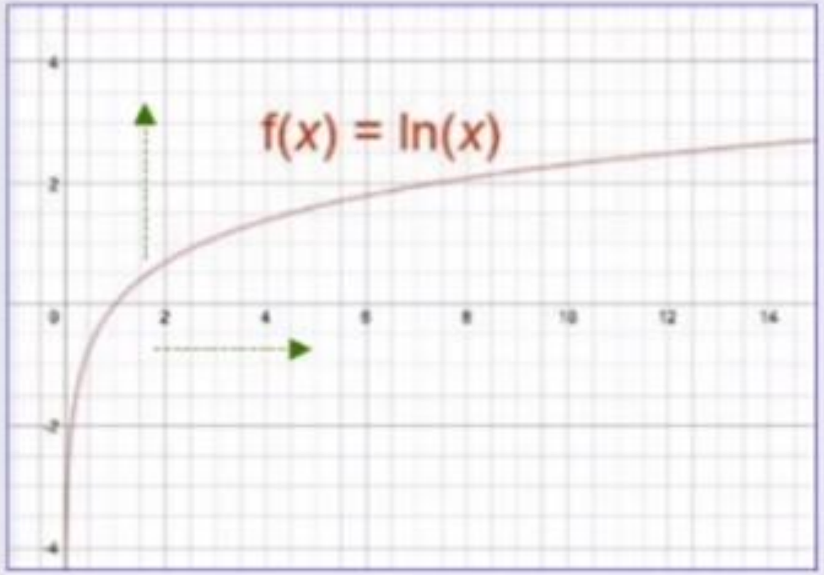
For Monotonically **increase** function: When Input **increases** → Output **increases**

For Monotonically **decrease** function: When Input **increases** → Output **decreases**

Some Monotonically increase Functions







Apply monotonically increase function Φ on Loss function $L(x) \quad \Rightarrow \quad \Phi(L(x))$

when $L(x)$ *is minimized* $\quad \Rightarrow \quad \Phi(L(x))$ is minimized

Conclusion: *instead of using* ***Loss*** *function* $L(x)$ *during training*, one can use $\Phi(L(x))$ where Φ *is any monotonically increase function*

We will use $\frac{1}{2}L(x)$ instead of $L(x)$ during minimization (to Simplify Math operations)

Minimization of Loss function to obtain BEST model parameters

$$Loss = \frac{1}{2}(\underline{Y} - \hat{\underline{Y}})^T(\underline{Y} - \hat{\underline{Y}}) = \frac{1}{2}(\underline{Y} - X\,\underline{W})^T(\underline{Y} - X\,\underline{W})$$

$$Loss = \frac{1}{2}[\underline{Y}^T\underline{Y} + (X\underline{W})^T(X\underline{W}) - \underline{Y}^T(X\underline{W}) - (X\underline{W})^T\underline{Y}]$$

$$Loss = \frac{1}{2}[\underline{Y}^T\underline{Y} + (X\underline{W})^T(X\underline{W}) - 2\,(X\underline{W})^T\underline{Y}]$$

Remember

X is a matrix, $\underline{W}$ is a vector,

$X\underline{W}$ is a vector

$$\underline{Y}^T(X\underline{W}) = (X\underline{W})^T\underline{Y}$$

$$\frac{\partial Loss}{\partial \underline{W}} = 0$$ Get Best W that minimizes the loss

Minimization of Loss function to obtain BEST model parameters

$$Loss = \frac{1}{2}(\underline{Y} - \hat{\underline{Y}})^T(\underline{Y} - \hat{\underline{Y}}) = \frac{1}{2}(\underline{Y} - X\,\underline{W})^T(\underline{Y} - X\,\underline{W})$$

$$Loss = \frac{1}{2}[\underline{Y}^T\underline{Y} + (X\,\underline{W})^T(X\,\underline{W}) - \underline{Y}^T(X\,\underline{W}) - (X\,\underline{W})^T\underline{Y}]$$

$$Loss = \frac{1}{2}[\underline{Y}^T\underline{Y} + (X\,\underline{W})^T(X\,\underline{W}) - 2\,(X\,\underline{W})^T\underline{Y}]$$

Remember
X is a matrix, $\underline{W}$ is a vector,
$X\,\underline{W}$ is a vector

$$\underline{Y}^T(X\,\underline{W}) = (X\,\underline{W})^T\underline{Y}$$

$\frac{\partial Loss}{\partial \underline{W}} = 0$ Get Best W that minimizes the loss

$$\frac{\partial Loss}{\partial \underline{W}} = 0 = \frac{1}{2}[(X\,\underline{W})^T(X) + (X)^T(X\,\underline{W}) - 2\,(X)^T\underline{Y}]$$

Remember

$$\underline{X}^T(X\,\underline{W}) = (X\,\underline{W})^T\underline{X}$$

$$\frac{\partial Loss}{\partial \underline{W}} = 0 = \frac{1}{2}[2X^T X\,\underline{W} - 2\,X^T\underline{Y}] \Rightarrow (X^T X)\underline{W} = X^T\underline{Y} \Rightarrow \boxed{\underline{W} = (X^T X)^{-1}X^T\underline{Y}}$$

Minimum Sum Squared Error (MSSE) Technique
(Least Squares)

$$\underline{W} = \left(X_{N\,x\,d}\right)^{-1} \underline{Y}$$

$X_{N\,x\,d}$ is non-square matrix

There is NO Inverse

$$\underline{W} = \left(X^T X\right)^{-1} X^T \; \underline{Y}$$

Pseudo Inverse of Non-Square Matrix

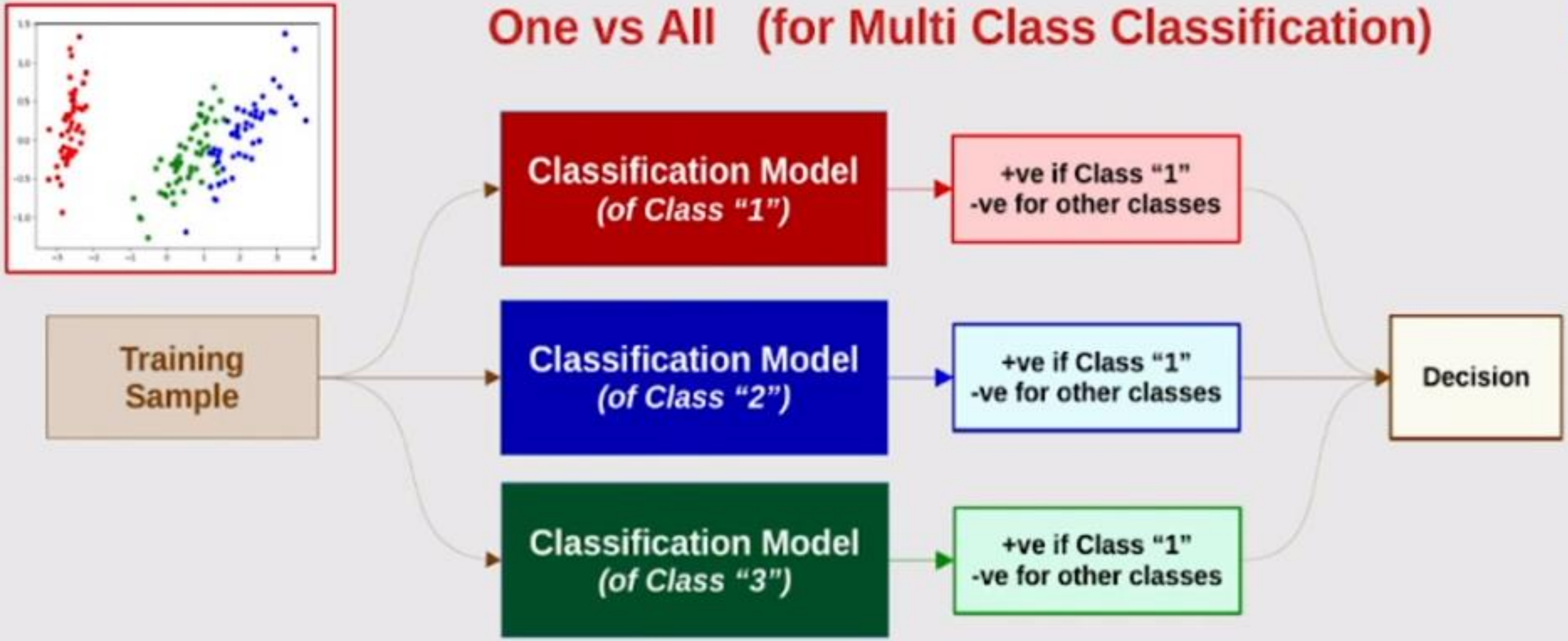
$$\left(X_{N\,x\,d}\right)^{-1} = \left(X^T X\right)^{-1} X^T$$

Remember

$$\left(X^T_{d\,x\,N} X_{N\,x\,d}\right)^{-1} = (\ldots)^{-1}_{d\,x\,d}$$

In General: $d \lll N$

One vs All (for Multi Class Classification)



Decision Rules

Number of Positive output (Only One) → Input is properly Classified

Number of Positive output (More than One) → can NOT define the class

NO Positive output → Not one of the classes (new class)