Supervised Machine Learning

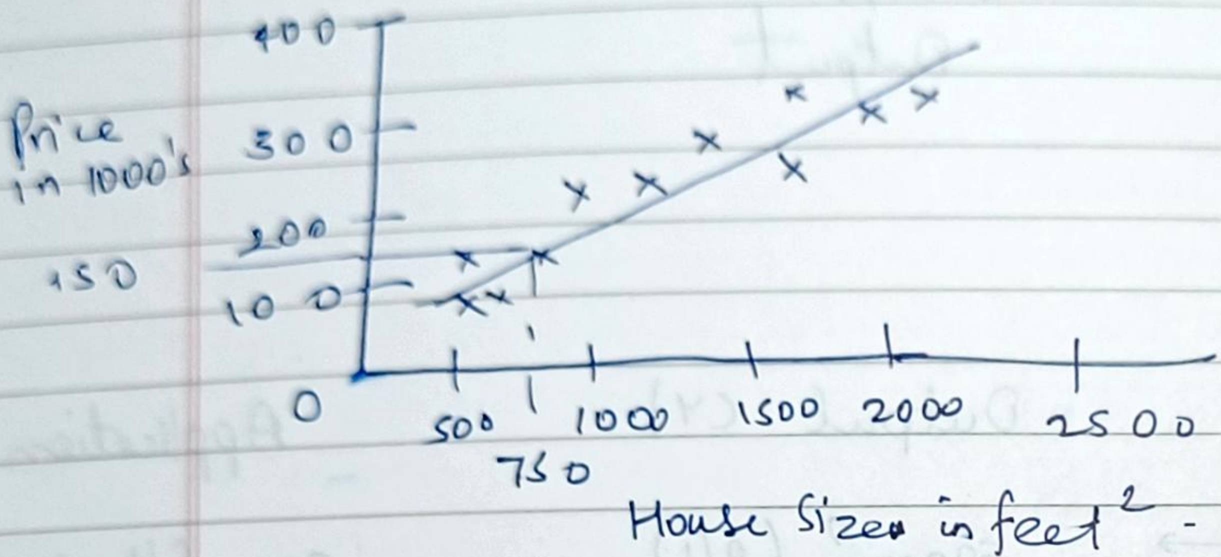
X ⟹ Y.

Input ⟶ Output

Examples

Input (x)		Output (Y)	Application
email	⟶	spam? (0/1)	Spam filtering
audio	⟶	text transcripts	speech recognition
English	⟶	Spanish	machine Translation
ad, user info	⟶	click (0/1)	online advertising
image, radar info	⟶	position of other cars	self-driving cars
image of phone	⟶	defect? (0/1)	visual inspection.

→ Regression : Housing price prediction

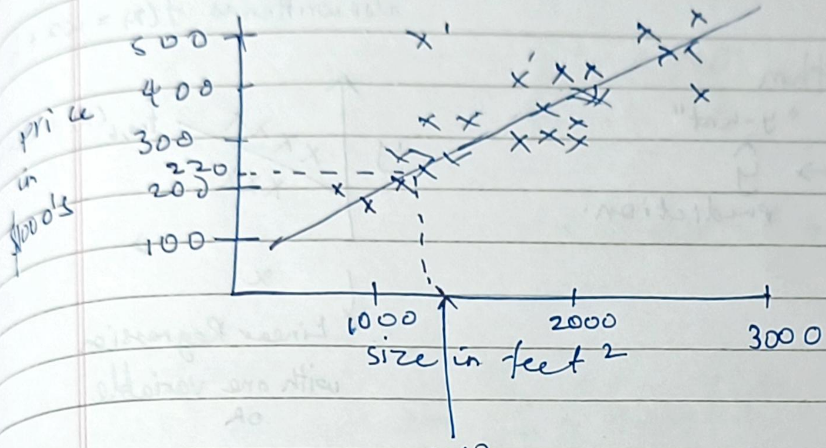


Regression : Predict a number → infinitely many possible outputs

→ Classification : Breast Cancer Detection

Linear Regression with one variable

House Sizes & Prices



supervised Learning Model

<u>Terminology</u>

Training set : Data used to train the model.

x size in feet	price in $1000's
2104	400
1416	232
1534	315
852	178
⋮	⋮
3210	870

<u>Notation</u>

x = "input variable feature

y = "output" variable "target" variable

m = number of training examples

(x, y) = single training example.

$x^{(1)} = 2104$, $y^{(1)} = 400$

$(x^{(1)}, y^{(1)}) = (2104, 400)$

(x^i, y^i)

$(x^{(i)}, y^{(i)}) = i^{th}$ training example.

training set
↓
Learning algorithm
↓

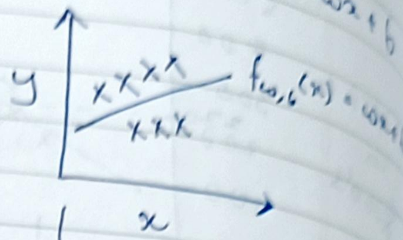
$x \rightarrow \boxed{f} \rightarrow \hat{y}$ "y-hat"

feature — function — prediction

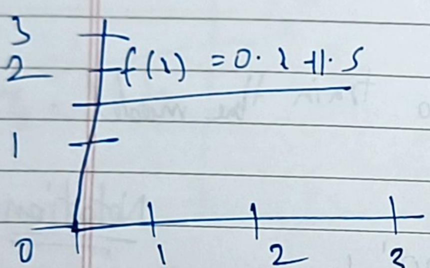
model

$f_{w,b}(x) = wx + b$

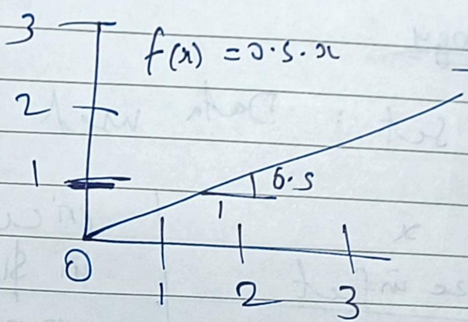
also written as $f(x) = wx + b$



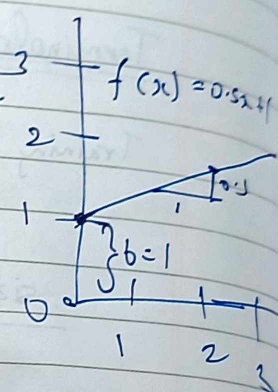
Linear Regression with one variable
OR
Univariate Linear Regression



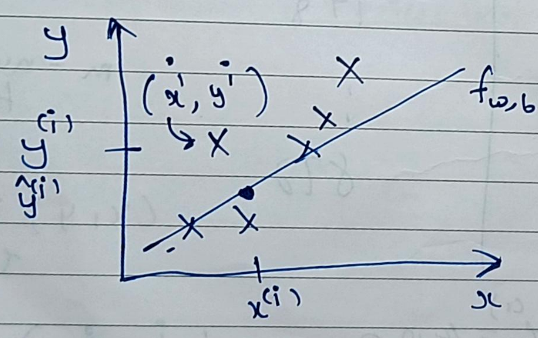
$w = 0$
$b = 1.5$



$w = 0.5$
$b = 0$



$w = 0.5$
$b = 1$



$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$

$f_{w,b}(x^{(i)}) = wx^{(i)} + b$

* Cost Function : Squared Error Cost Function

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(\underbrace{\hat{y}^{(i)} - y^{(i)}}_{\text{error}} \right)^2$$

m = number of training examples

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{(w,b)}(x^{(i)}) - y^{(i)} \right)^2$$

simplified

model : → $f_w(x) = wx$ $\quad b = \phi$

$f_{w,b}(x) = wx + b$

parameters : → w

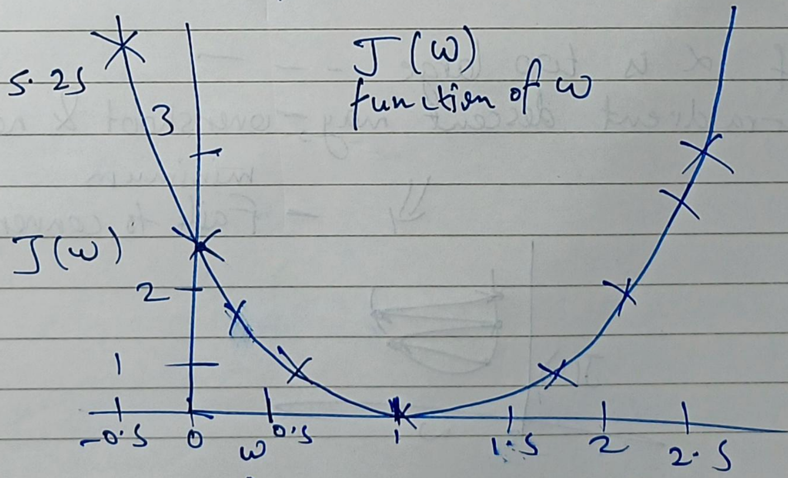
w, b

cost function : → $J_{(w)} = \frac{1}{2m} \sum_{i=1}^{m} \left(\underbrace{f_w(x^{i})}_{\to\, wx^{(i)}} - y^{i} \right)^2$

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

goal :

$\underset{w,b}{\text{minimize}}\ J(w,b)$ → $\underset{w}{\text{minimize}}\ J(w)$



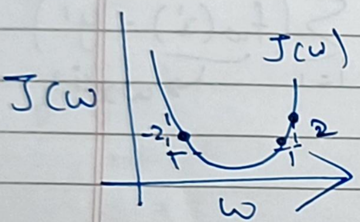
→ Gradient Descent Algorithm [helps us get w's accurate value]

$$temp\_w = w - \alpha \frac{d}{dw} J(w,b)$$

$$temp\_b = b - \alpha \frac{d}{db} J(w,b)$$

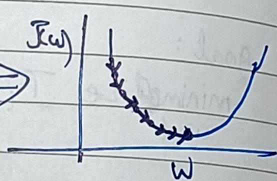
$$w = w - \alpha \underbrace{\frac{d}{dw} J(w)}_{> 0}$$

$$w = w - \alpha \cdot (\text{positive number})$$



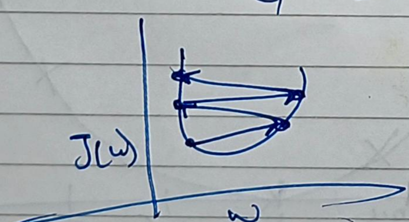
$$w = w - \alpha \cdot (\text{negative number})$$

If α is too small - -
Gradient descent may be slow ⟹



If α is too large - - - -
Gradient descent may overshoot & never reach the minimum

⇓ - Fails to converge, diverge



Linear regression model.

$$f_{(w,b)}(x) = wx + b$$

Cost f^n

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m} f_{w,b}\left[x^i - y^i\right]^2$$

Gradient Descent Algorithm

repeat untill convergence {

$$w = w - \alpha\left(\frac{\partial}{\partial w} J(w,b)\right) \rightarrow \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}(x^i) - y^i\right) x^i$$

$$b = b - \alpha\left(\frac{\partial}{\partial w} J(w,b)\right) \rightarrow \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}(x^i) - y^i\right)$$