what is supervised learning?

→ supervised learning is a type of machine learning in which an algorithm learns to makes prediction and decision based on labeled train data.

The Algorithm is provided with a dataset that includes input output pairs

X( ) = Inputs
(feature or attributes of data

Y( ) = Outputs
(corresponding target values (labels)

A function learn $h: X \rightarrow y$

CamScanner

Training set

↓

learning algorithm

↓

$x \longrightarrow h \longrightarrow y$

Predicted

Living area of house

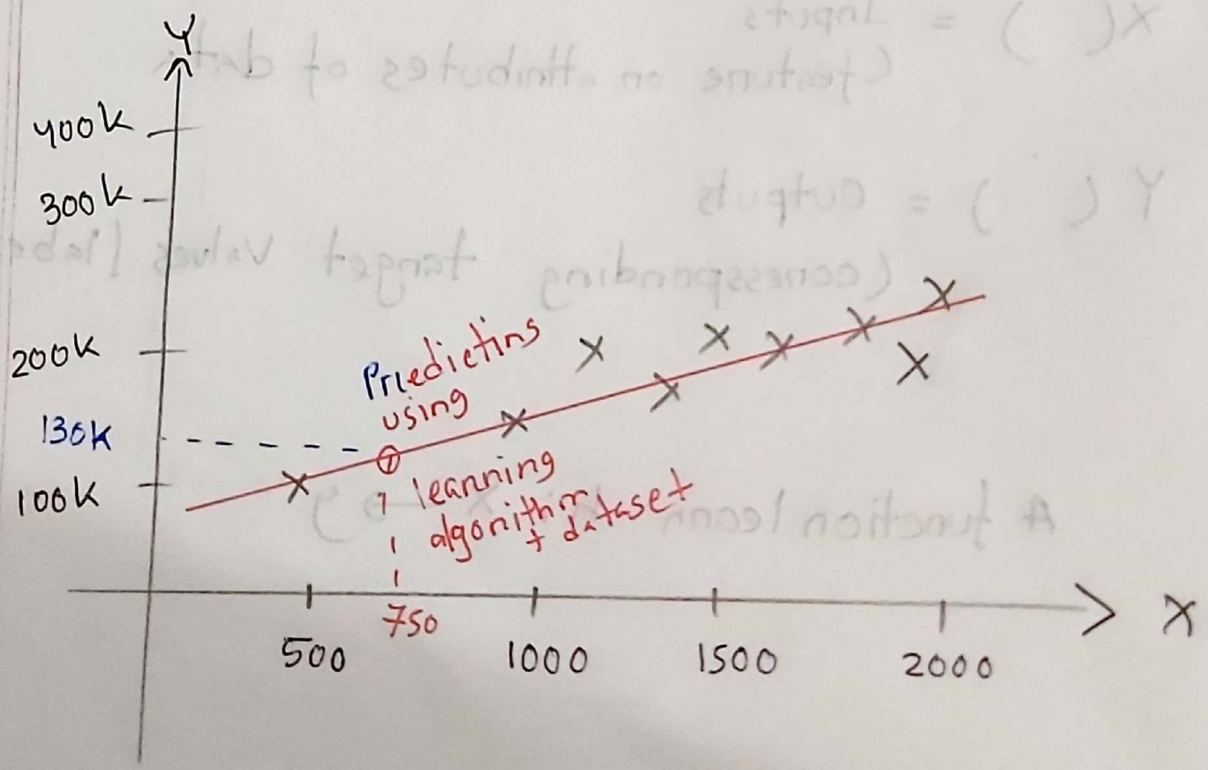
Hypothesis

predicted price of house

$h(x)$

is a good predictor for corresponding y

Predict prize of living area (750 feet^2)



Regression

It is a supervised learning task where the goal is to predict a continous numerical or value a real number.

Example :- Predicting House price

Input feature (x) : square foot of house

Output (Y) : Predict the selling price which is continous numerical value

Classification

Classification is a supervised learning task where the goal is to categorize or label data into predefined classes or category.

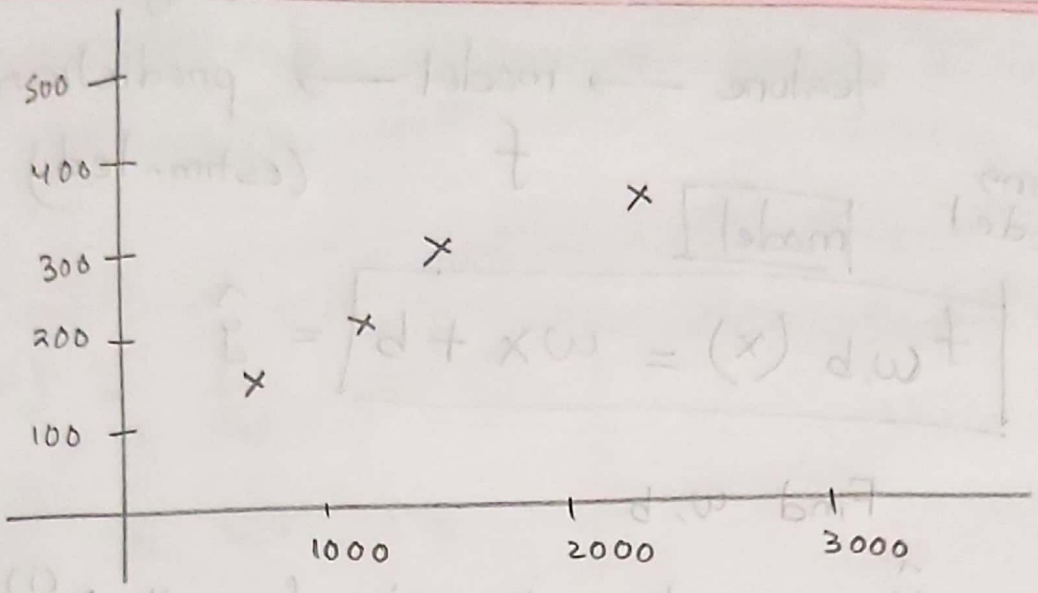
Example :- Email spam detection

Input(X) :- The content and attributes of email (text, sender, subject)

Output (Y) : A binary classification label either "spam" or "not spam"

CamScanner

Linear Regression Model



Training data set

m number of training	X = input variables / features Size in feet^2	Y = output / target variable prize
1	2104	400
2	1416	232
3	1534	315
4	852	178
⋮	⋮	⋮
47	3210	870

(x, y) = single training example

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

tanget

feature $\longrightarrow$ model $\longrightarrow$ prediction

f (estimated)

Learning model

$\boxed{\text{model}}$

$$\boxed{f_{w,b}(x) = wx + b} = \hat{y}$$

Find w, b

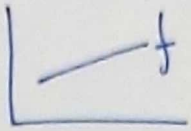
$\hat{y}$ is close to $y^{(i)}$ for all $x^{(i)}, y^{(i)}$

$\boxed{\text{cost function}}$

(squared error cost function)

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

that is adjusted as the model learns data



w.b = coefficient or weights

parameter

m = number of training example

y(i) = ith target prediction

$$\boxed{\begin{aligned} \hat{y}^{(i)} &= f_{w,b}(x^{(i)}) \\ &= wx^{(i)} + b \end{aligned}}$$

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

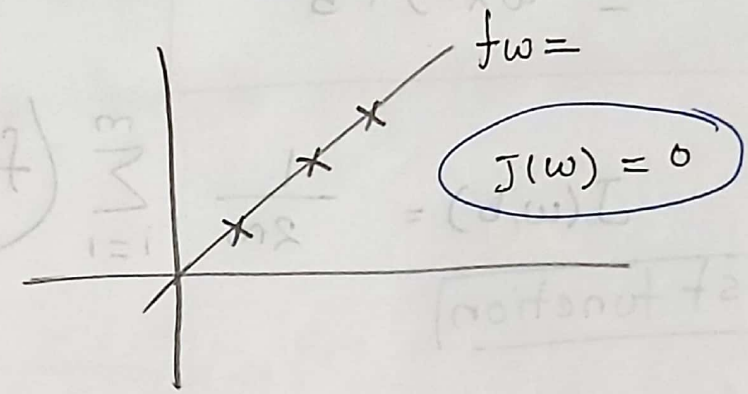
cost function

$\boxed{\text{Goal is to minimize } J(w,b)}$

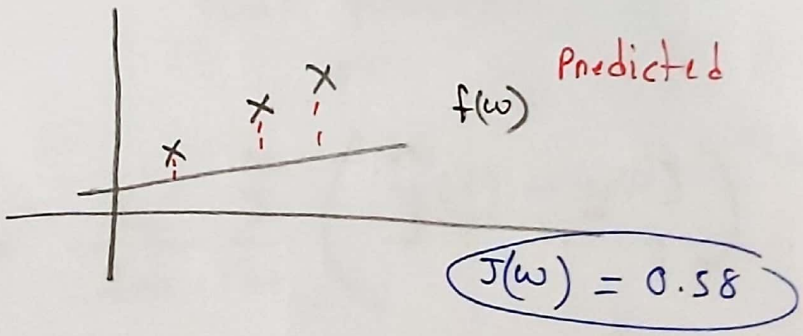
simplified

$$\boxed{f_w(x) = wx} \qquad b = 0$$

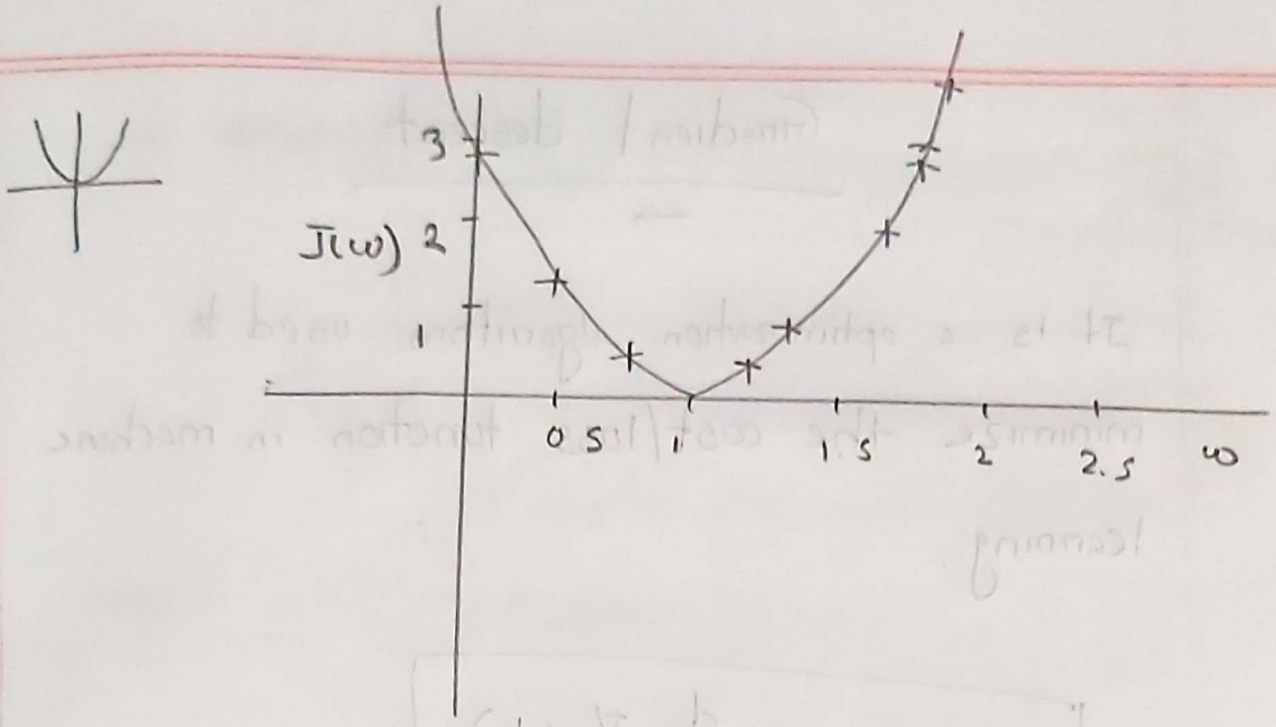
$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_w(x^{(i)}) - y^{(i)} \right)^2$$



$w = 1$



$w = 0.5$



∴ Goal of linear regression

minimize $J(w)$

$J(w)$ is minimum = model fits the dataset perfectly

CSE

Gradient descent

It is a optimization algorithm used to minimize the cost/loss function in machine learning.

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

α = Learning rate

It is a parameter in machine learning and optimization algorithm (gradient descent) that determines the size of the steps taken during updates in training proces

is like balance between moving fast and overshooting best solution

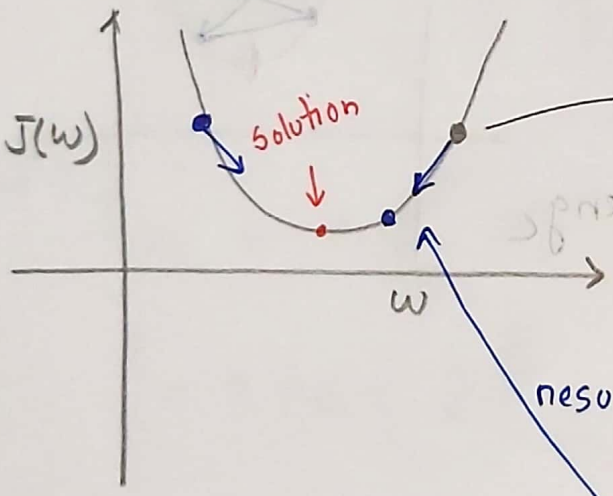
if learning rate high, jump over lowest point, wont reach best solution

small, get stuck with tiny steps. long time to reach

repeat untill convergence

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

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



moving towand to minimize

doing gradient descent at this point

result

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

$-\alpha$ · positive numbe

pos

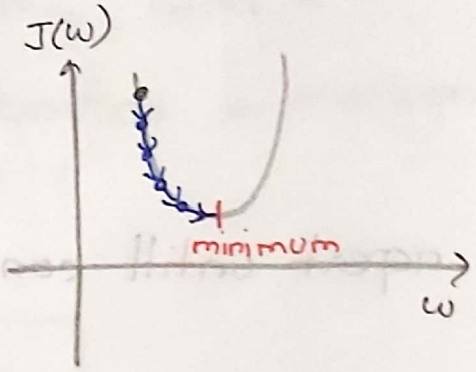
= decreasing w

CSE

$$\boxed{w = w - \alpha \frac{d}{dw} J(w)}$$

If α is too small

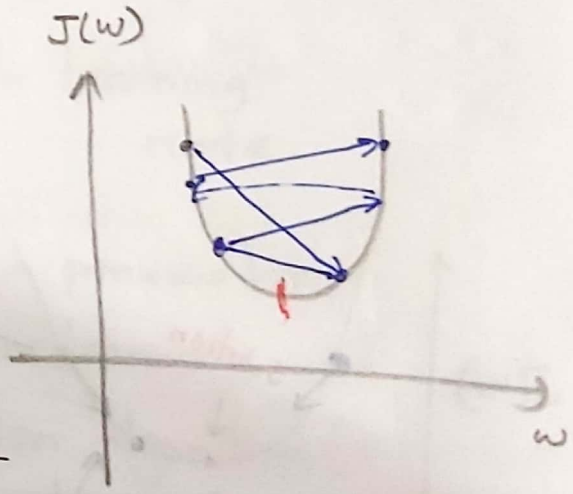
Gradient descent is too small



if α is too large

Gradient descent may overshoot and never reach the minimum

→ fail to converge, diverge



Linear regression model

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

cost function

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

Gradient descent Alogonithm

repeat untill convergence

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

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

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

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

CS CamScanner

CSE

what is machine learning?

Machine Learning is subset of AI that focus on development of algorithm and statistic models allowing system to learn and improve from experience without being explicitly programmed.

Types

1. Supervised learning
2. Unsupervised learning
3. Reinforcement

CS CamScanner

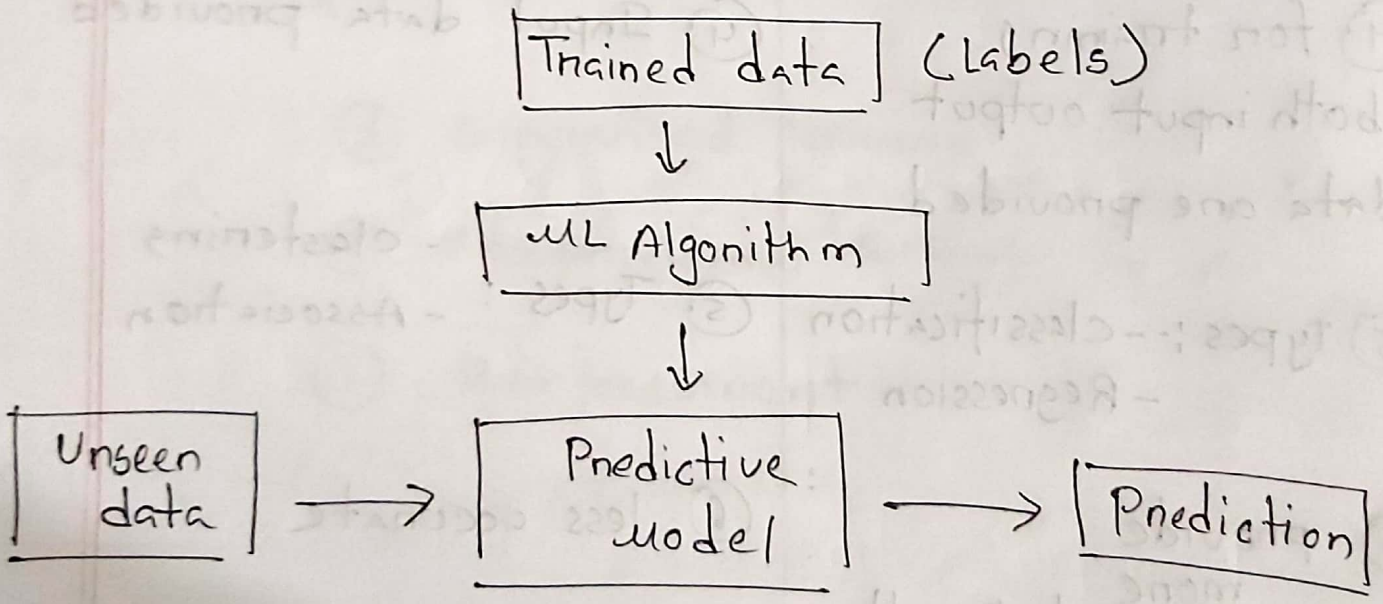
Supervised learning	Unsupervised learning
① Algorithms are trainned using labeled data	① unlabeled data
② Take direct feedback of output prediction	② doesn't take feedback
③ model predicts the outputs	④ model finds hidden patterns in data
④ for training both input, output data are provided	④ Input data provided only
⑤ Types :- - classification - Regression	⑤ Types : - clustering - Association
⑥ provide more accurate Result	⑥ less accurate

CamScanner

① The goal of supervised learning is to train model so that it can predict output when new data given

② The goal of unsupervised is to find hidden patterns and useful insights from unknown dataset

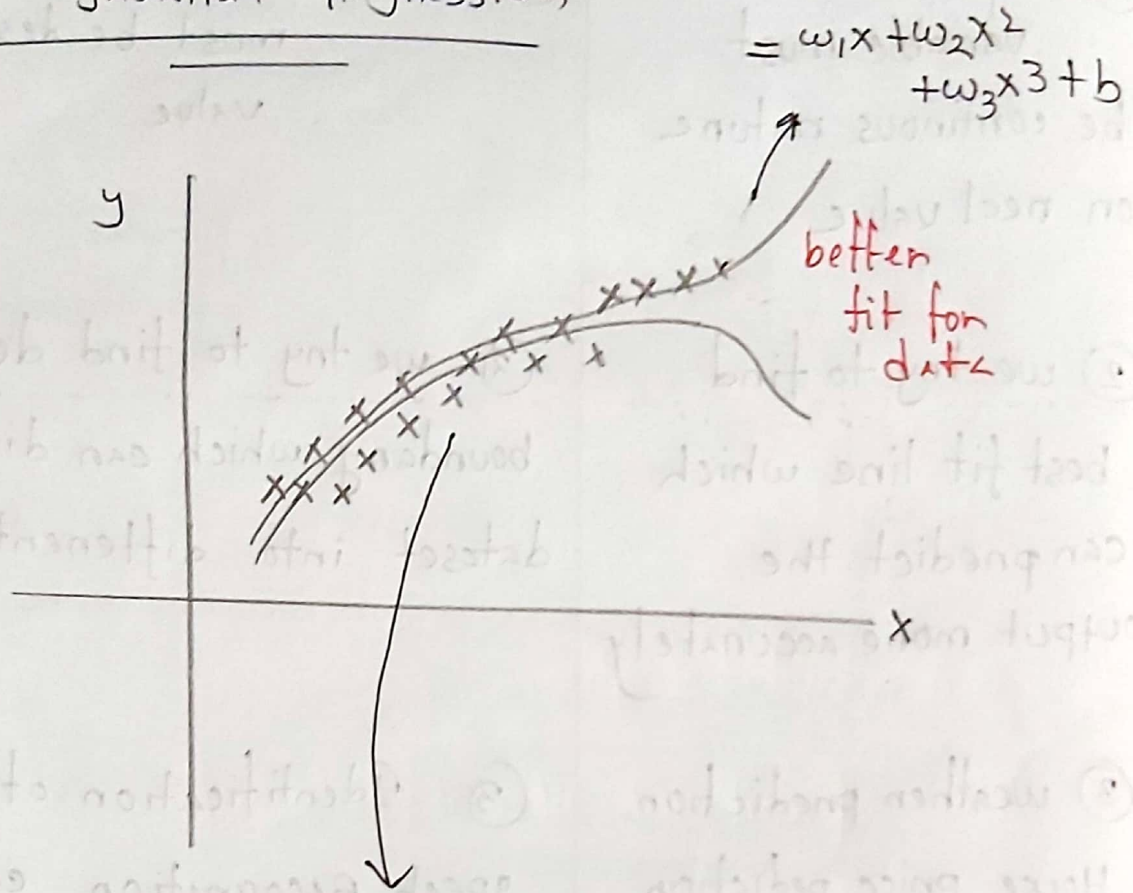
— o —



CSE

Regression	classification
① output variable must be continous nature on real value	② output variable must be descrete value
② we try to find best fit line which can predict the output more accurately	② we try to find decision boundary which can divide dataset into different classes
③ weather prediction, House price pediction	③ Identification of spam, speech Recognition, cancel Identification
④ type :- linear nonlinear	④ type :- Binary Multiclass

Polynomial Regression

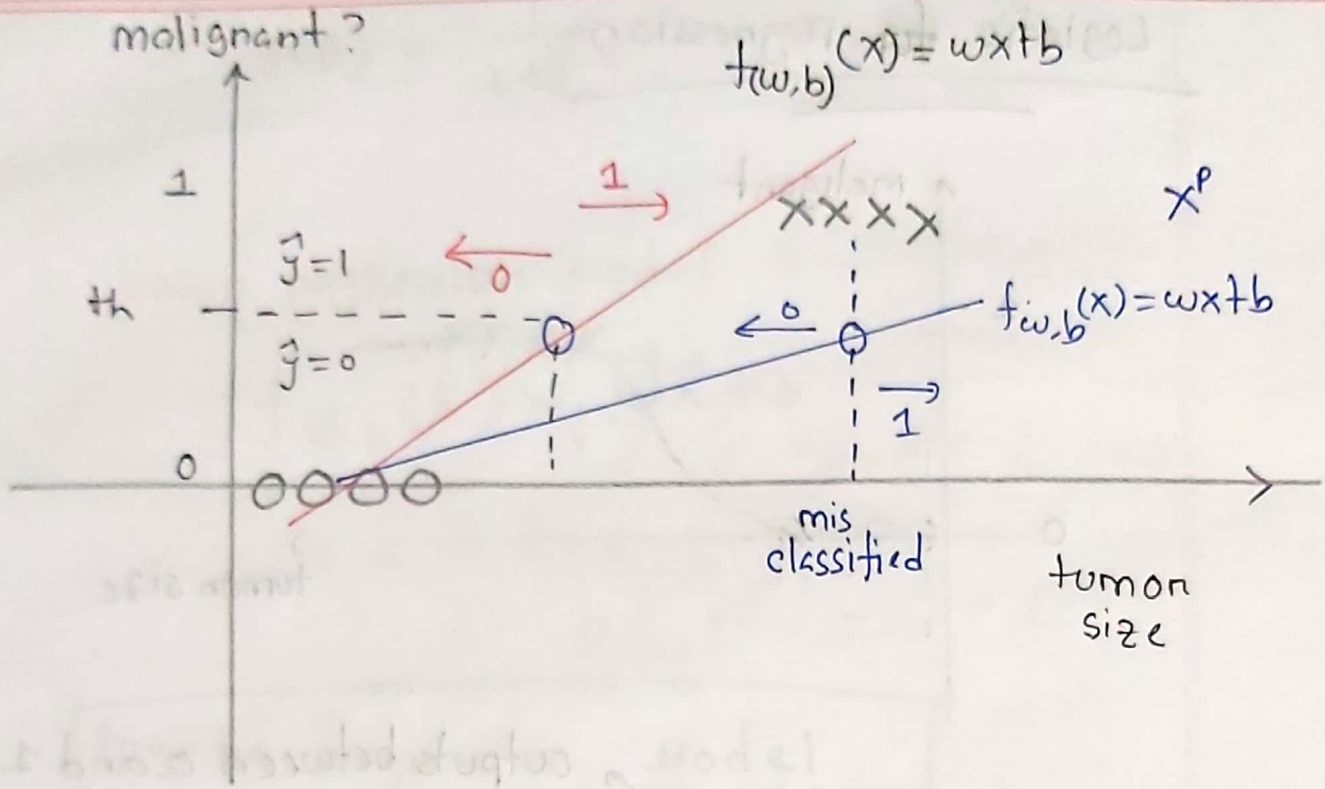


$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + b$$

what is cost function?

In machine learning cost function refers to mean squared error between the predicted outputs and true labels

classification



threeslold (th)

if $f_{w,b}(x) < 0.5$

$\hat{y} = 0$

if $f_{w,b}(x) \geq 0.5$

$\hat{y} = 1$

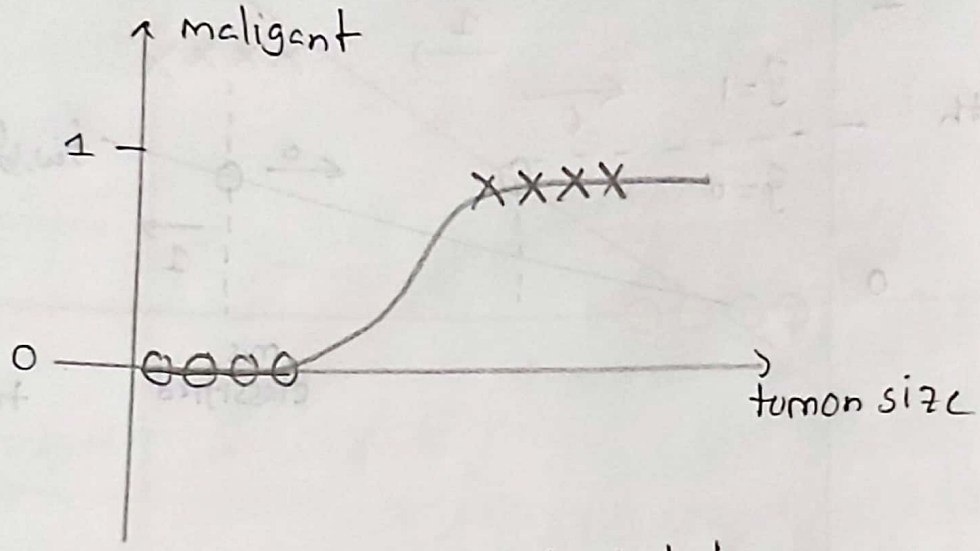
now if P occurs

model will change

that's why for above example linear Regression is not a good choice

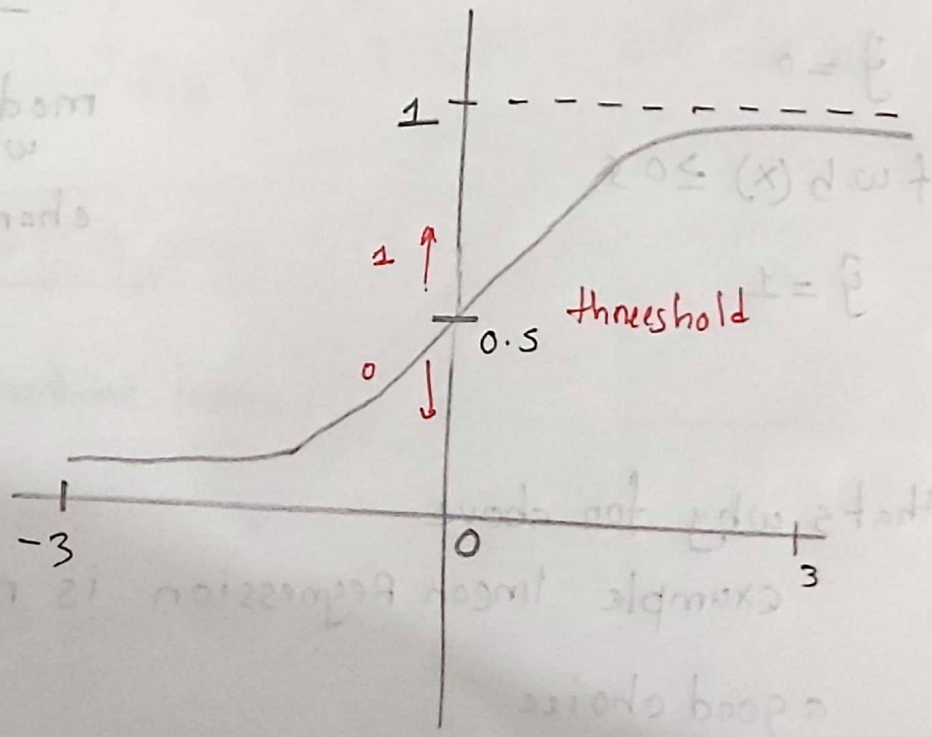
we need classification here (decision boundary)

Logistic Regression



outputs between 0 and 1

$\boxed{\text{Sigmoid / logistic function}}$



sigmoid function

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1$$

$\therefore$ linear regression model

$$f_{\vec{w},b}(\vec{x}) = \vec{w}\cdot\vec{x} + b$$

$$\therefore \quad z = \vec{w}\cdot\vec{x} + b$$

$\therefore$ $\boxed{\text{logistic regression model}}$

$$f_{\vec{w},b}(\vec{x}) = g(z)$$

$$= \frac{1}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

CamScanner

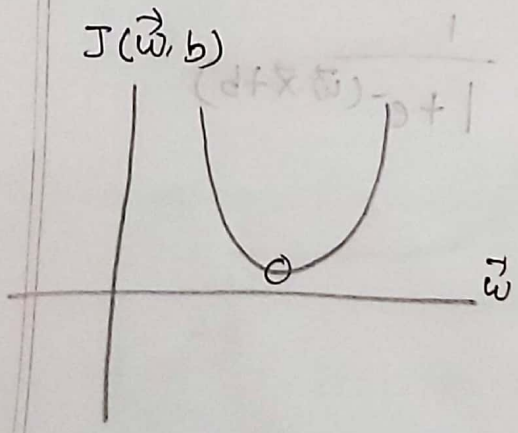
Cost function for logistic Regression

linear Regression

$$f_{\vec{w},b} = \vec{w}\cdot\vec{x} + b$$

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

which is convex

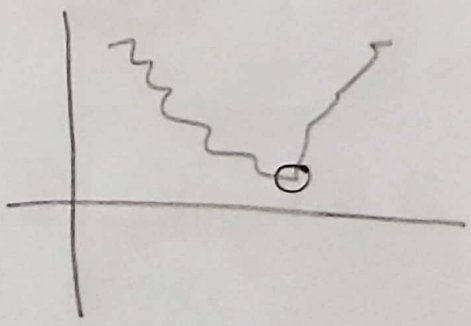


logistic negression

$$f_{\vec{w},b} = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

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

which is non convex



↓ convex logistic negression cost function

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \underbrace{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right)}_{\text{Loss function}}$$

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) & \boxed{\text{if } y^{(i)} = 1} \\ -\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right) & \boxed{\text{if } y^{(i)} = 0} \end{cases}$$

convex :- now can reach a global minimum

simplified cost function

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)})$$

$$= -y^{(i)} \log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) - (1-y^{(i)})\left(\log(1-f_{\vec{w},b}(\vec{x}^{(i)}))\right)$$

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

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

Gradient descent for logistic regression

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

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

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}.\vec{x}+b)}}$$

Underfitting

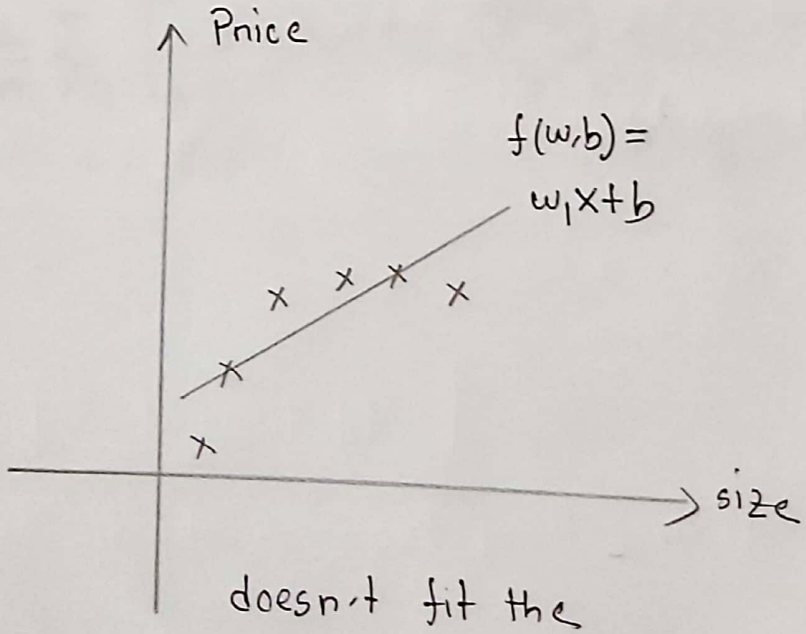
→ when model is not able to capture the underlying trend of data

→ didnot learn enough from trained data

→ High bias → (toosimple, inconnect prediction)

→ Low variance (doesn't adapt trained data)

→ Model has too few parameters / feature



doesn't fit the training set well

CSE

overfitting

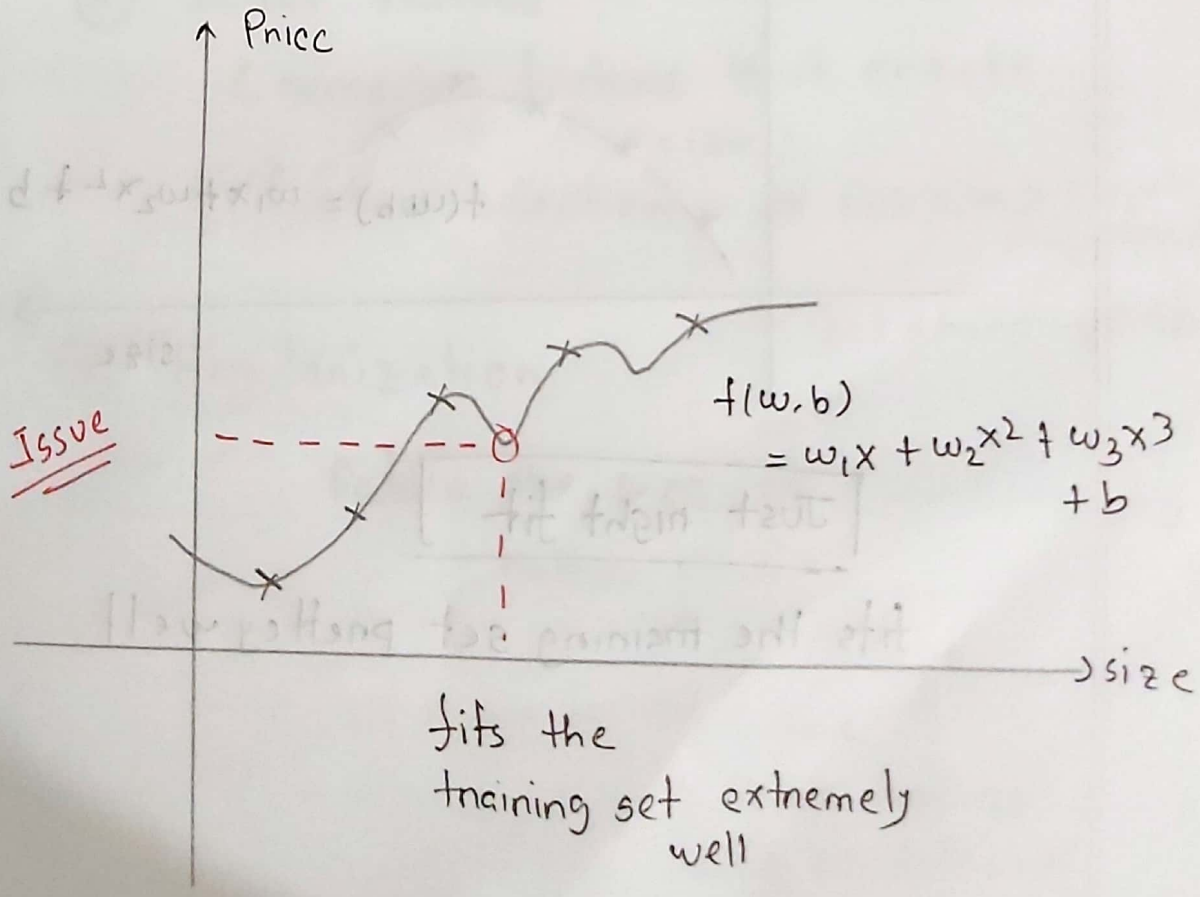
→ when model tries to coven all the data points / more than required data in dataset (unnecessary - irrelevant data)

→ caching noises and inaccurate values

→ Low bias (very complex, random fluctuations)

→ High variance (change prediction significantly)

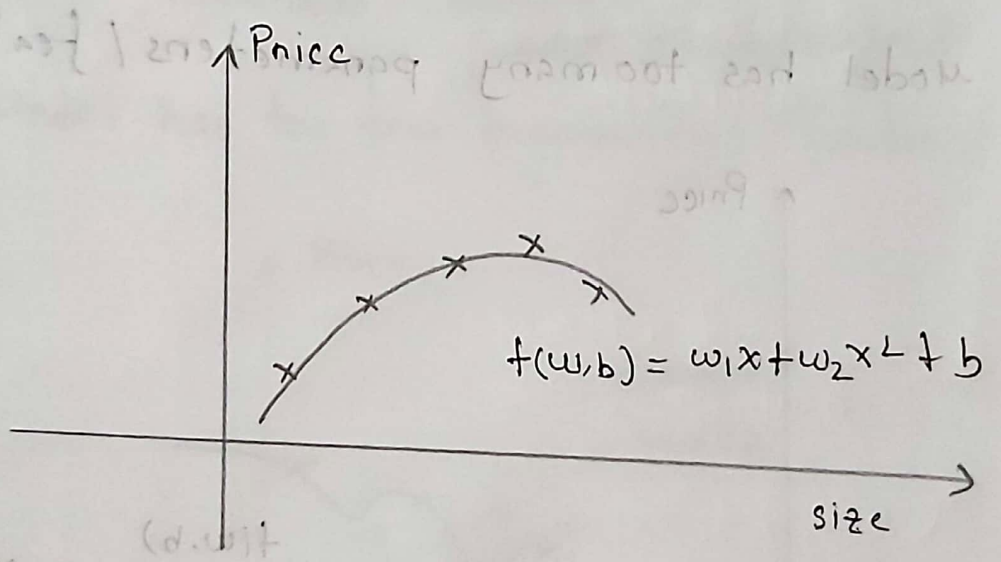
→ Model has too many parameters / feature



fits the training set extremely well

Bias → represents how much the prediction of model differs from true values

Variance → measures how much the model's prediction fluctuate or vary when trained on different subsets of data



Just right fit

fits the training set pretty well

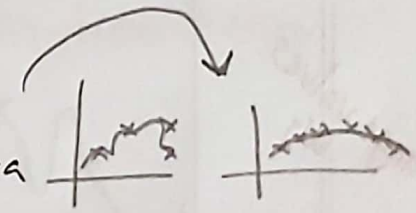
CSF

what is generalization?

refers to model ability to make accurate predictions on new, unseen data that has not been trained on.

CSF

How to solve overfiting?



1. Collect more trained data
2. Select feature to include/exclude (removing feature that create noise)

   feature selection

   ✗ $f() = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 \ldots$

   ✓ $f() = w_1 x + w_2 x^2 + b$

3. Regularization

   Reduce the size of parameter w_j rather than elimination

   → takes all feature

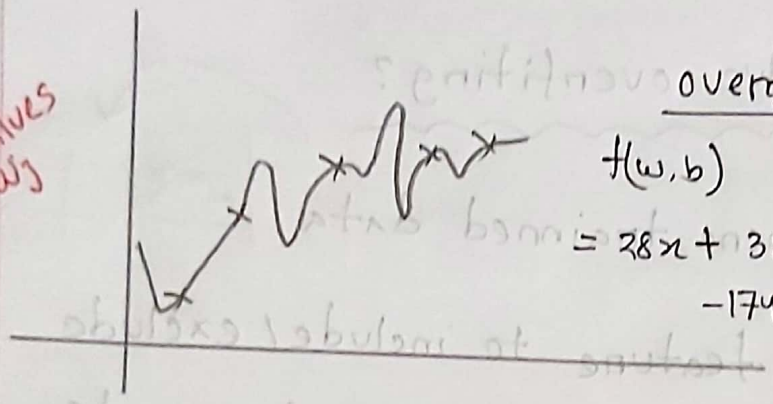
   → but restrict them having large effect on dataset

OSF

Regularization

refers to regularize / reduce the magnitude of feature/parameter by keeping the same number of feature.

large values w_j



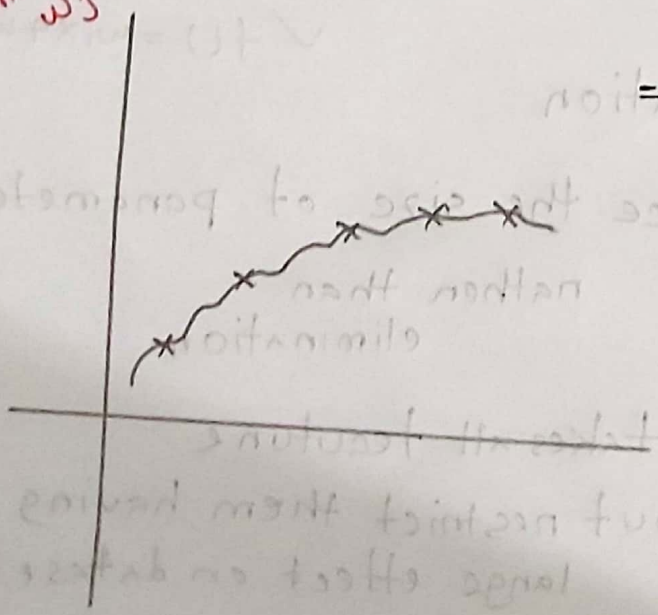
overfiting

$f(w,b)$

$= 28x + 385x^2 + 39x^3 - 174x^4 + 100$

after doing Regularization

small values w_j



$= 13x + (-.23x^2) + 0.000014x^3 - .0001x^4 + 10$

cost function of Regularization :-

$$J(\vec{w}, b) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \underbrace{\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2}_{\text{Regularization term}}\right]$$

Regularization parameter

Gradient descent

$$w_j = \underbrace{w_j\left(1 - \alpha\frac{\lambda}{m}\right)}_{\text{Shrinking } w_j} - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}$$