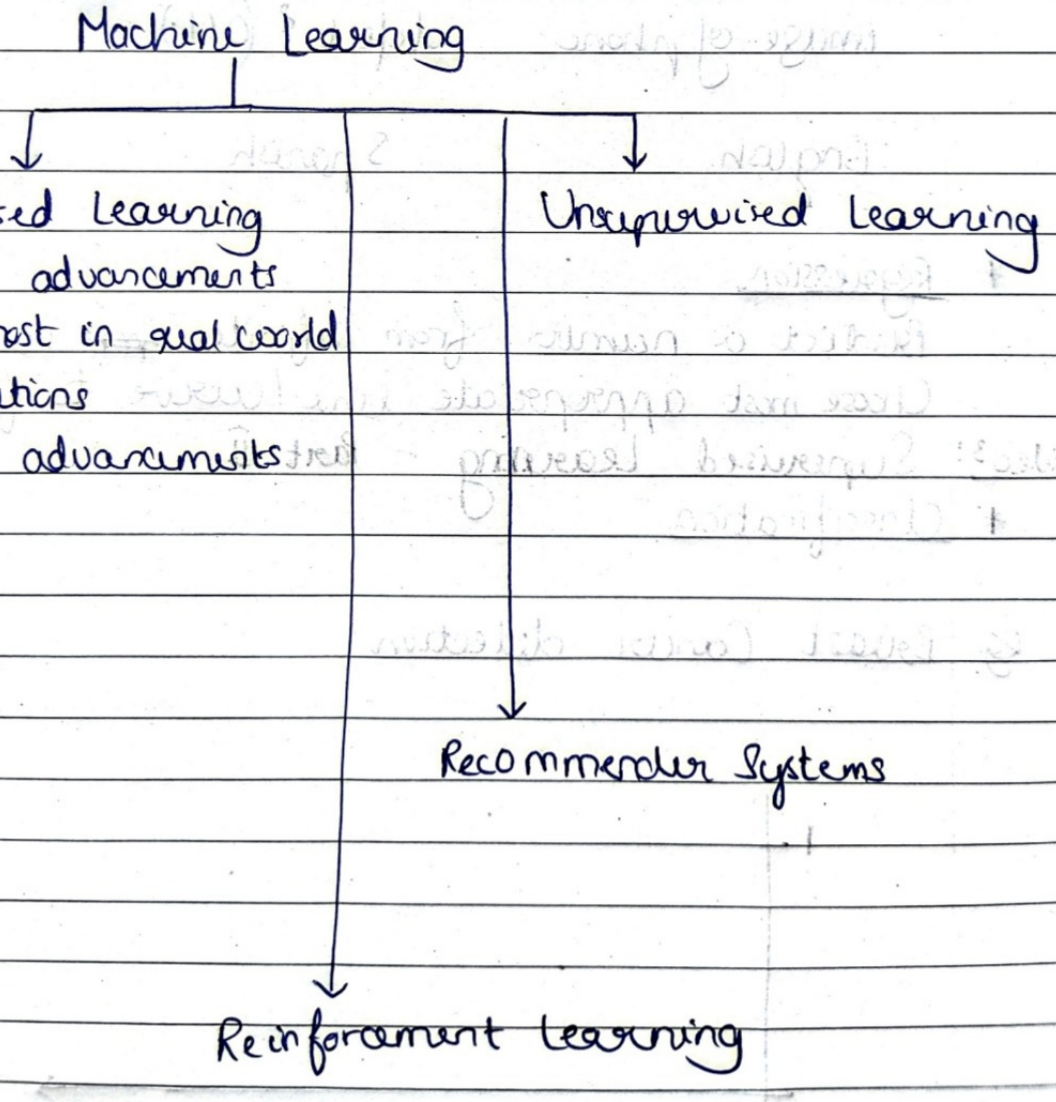
SUPERVISED ML: REGRESSION & CLASSIFICATION

Week 1: Supervised vs Unsupervised Machine Learning

Video 1: What is machine learning?

The more opportunities you give a learning algorithm to learn, the better it will perform.



PAGE NO.
DATE :

Video 2: Supervised Learning - Part I

* They learn to map from input to output.
* You give your learning algorithm different examples (input x & desired output y)

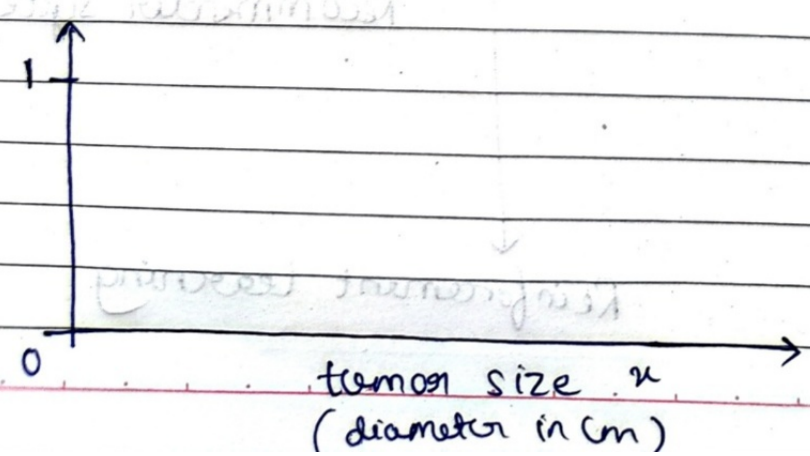
input	output (y)	application
email	spam? (0/1)	spam filtering
ad, user info	click? (0/1)	online ads
image of phone	defect? (0/1)	visual inspection
English	Spanish	machine translation

* **Regression**
  Predict a number from infinite ~~no~~ numbers.
  Choose most appropriate line/curve to fit data.

Video 3: Supervised Learning - Part II

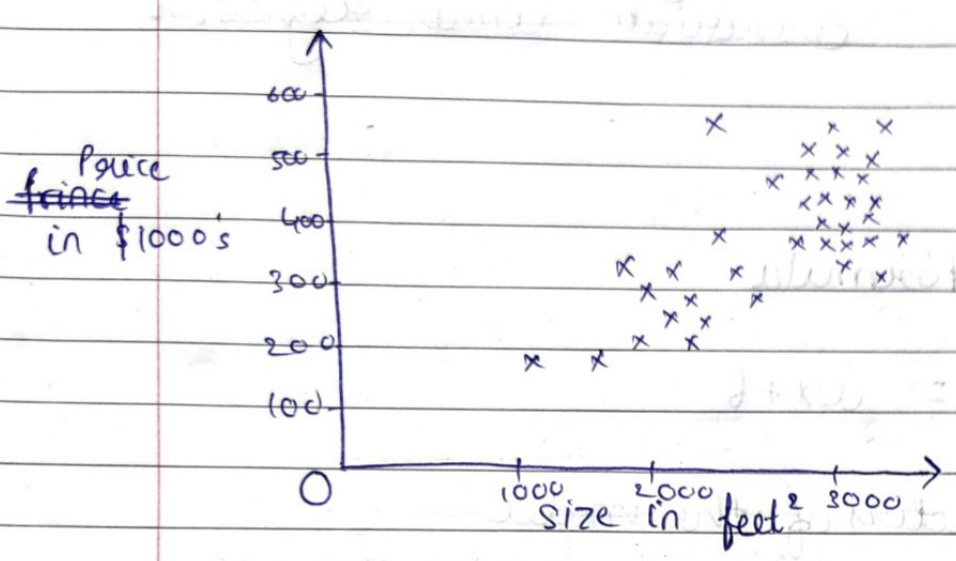
* **Classification**

eg. Breast Cancer detection



Week 1: Regression Model

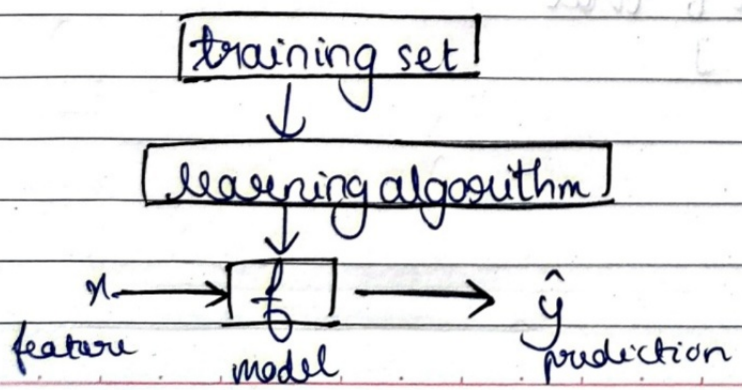
Video 1: Linear Regression Model - Part (I)



Training set : Data used to train model
notation : x
output / target variable : Price of house
notation : y
m - total training examples

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

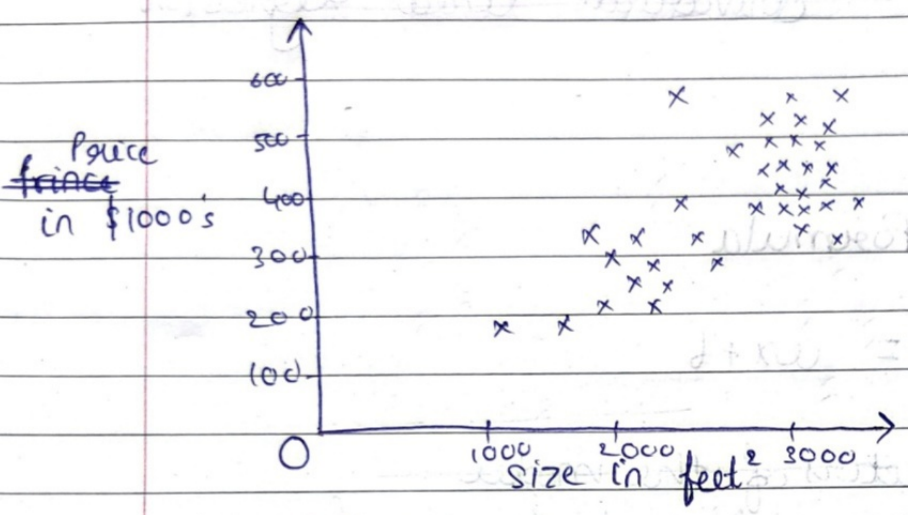
Video 2: Linear Regression Model - Part (II)



PAGE NO.
DATE: / /

Week 1: Regression Model

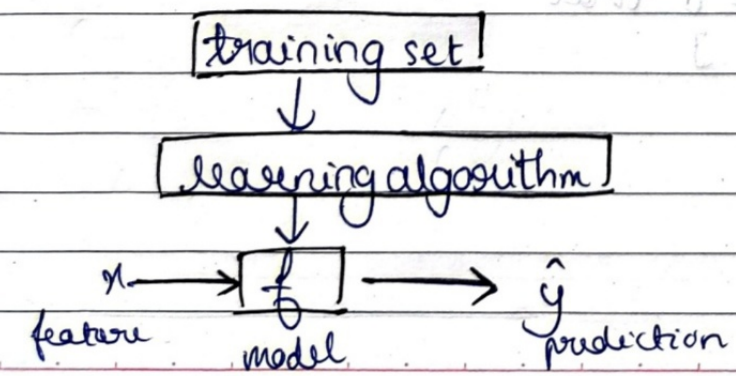
Video 1: Linear Regression Model - Part (I)



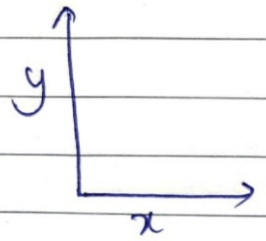
Training set : Data used to train model
notation : x
output / target variable : Price of house
notation : y
m - total training examples

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

Video 2: Linear Regression Model - Part (II)



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



linear regression with one variable - univariate linear regression

Video 3: Cost function Formula

Model : $f(x) = wx + b$

w, b – parameters of the model
they can adjusted during training
weights

cost function is used to check how well a line fits the training data.

It takes the prediction $\hat{y}$ and compares it with y.

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

(squared error cost function)
used for linear regression

to find : w & b that minimise J

PAGE NO.
DATE: / /

Video 4: Cost Function Intuition

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

parameters: w, b

cost function:

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

goal:

minimize $J(w,b)$

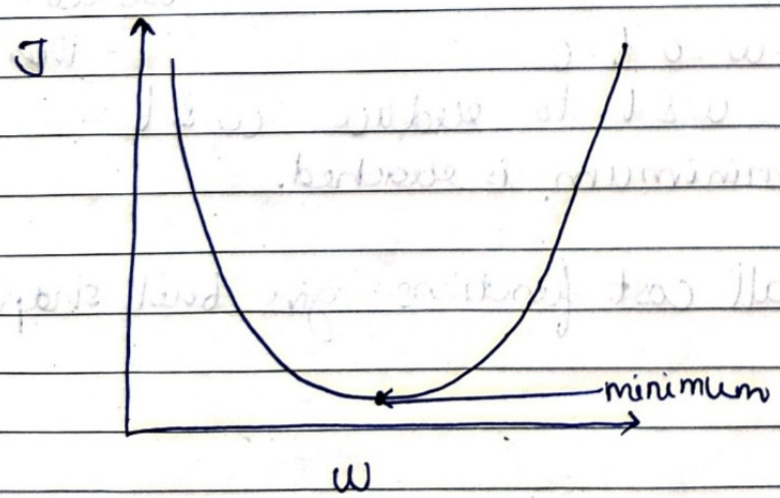
simplified:

$f_w(x) = wx \qquad b = \phi$

parameter : w

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

$\therefore$ minimise $J(w)$



Video 5: Visualising cost function

* we use contour plots to visualise

Video 6: Visualisation examples

(gradient descent is used to fine tune parameters

Week 1: Train the model with gradient descent

Video 1: Gradient Descent

We have : $J(w,b)$
We want : $\min_{w,b} J(w,b)$

gradient descent can be used to minimise any function.

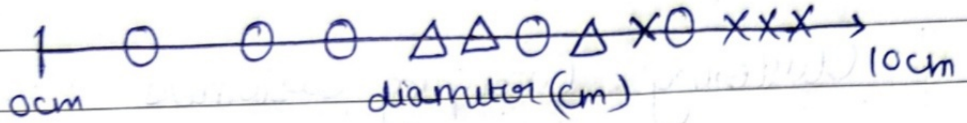
Outline :

- start with $w=0$ $b=0$
- keep changing w & b to reduce w & b
- ~~until~~ until minimum is reached.

we can have more than one minimum

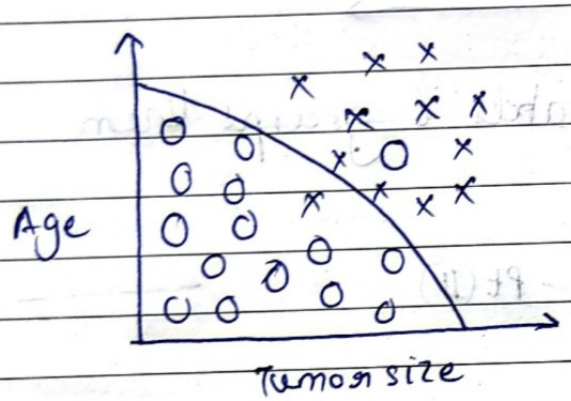
Note: Not all cost functions give bowl shaped graph.

O - benign
X - malignant type 1
Δ - malignant type 2



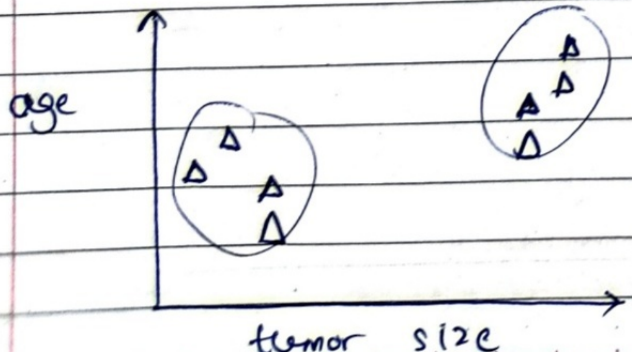
classification - gives difinite numbers-predicts category
regression - gives infinitely many numbers

Two or more inputs



Video 4: Unsupervised Learning - Part 1

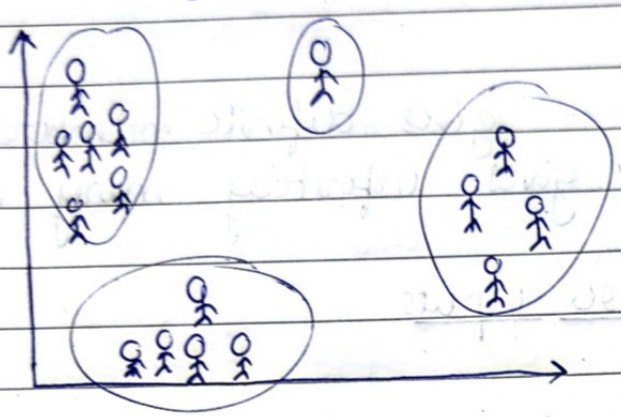
* data isn't associated with output labels.
* we need to find some pattern in the data.



Clustering : Google news

* Bunches today related or similar articles having similar words.

Clustering : Grouping customers



* takes data without labels & groups them.

Video 4: Unsupervised Learning - Pt (II)

(i) Clustering

(ii) Anomaly Detection - finds unusual data points.
application - fraud detection

(iii) Dimensionality Reduction - compress data using fewer numbers.

* Steepest descent should be followed.

Video 2: Implementing Gradient Descent Algorithm

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

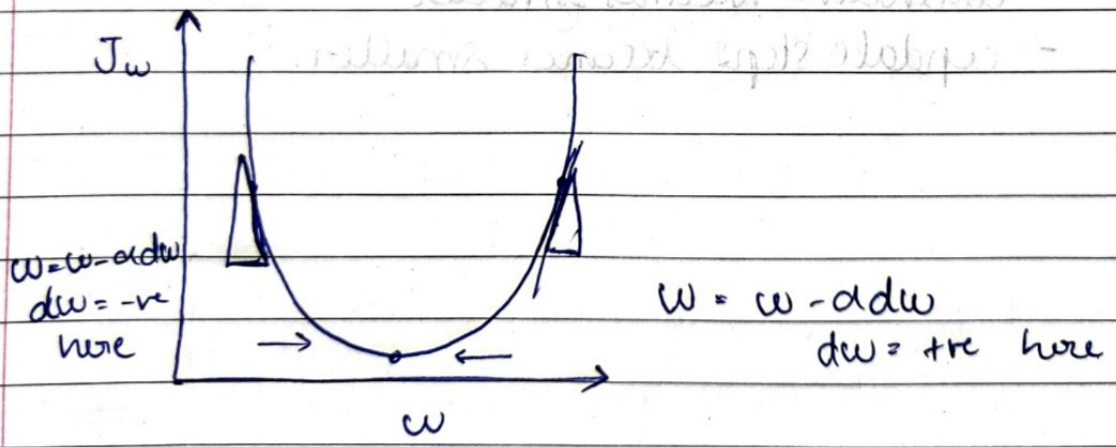
simultaneously updated

α - learning rate
- it is a small number
- it controls how steeply you will go down the slope.

same for b

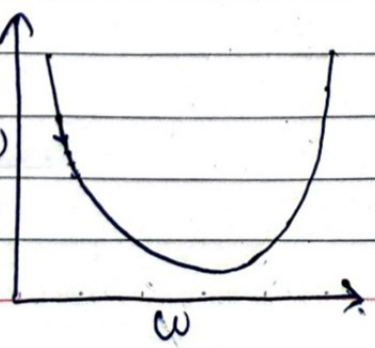
$\frac{d}{dw} J(w,b)$ - tells in which direction you want to take your steps.

Video 3: Gradient Descent Intuition

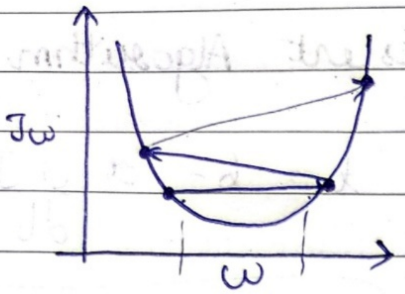


Video 4: Learning Rate

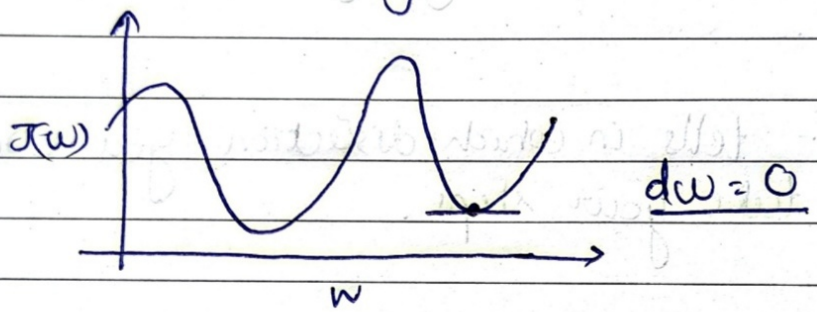
If α is too small:
steps are minuscule
slow.



If α is too large: results in overshoting



If you are already at local minimum, w remains unchanged.



Near a local minimum,
- derivative becomes smaller
- update steps becomes smaller.

5) Gradient Descent for Linear Regression

DATE : / /

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 algorithm

repeat until convergence {

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

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

Video 6: Running Gradient Descent

Batch gradient descent - entire batch is processed