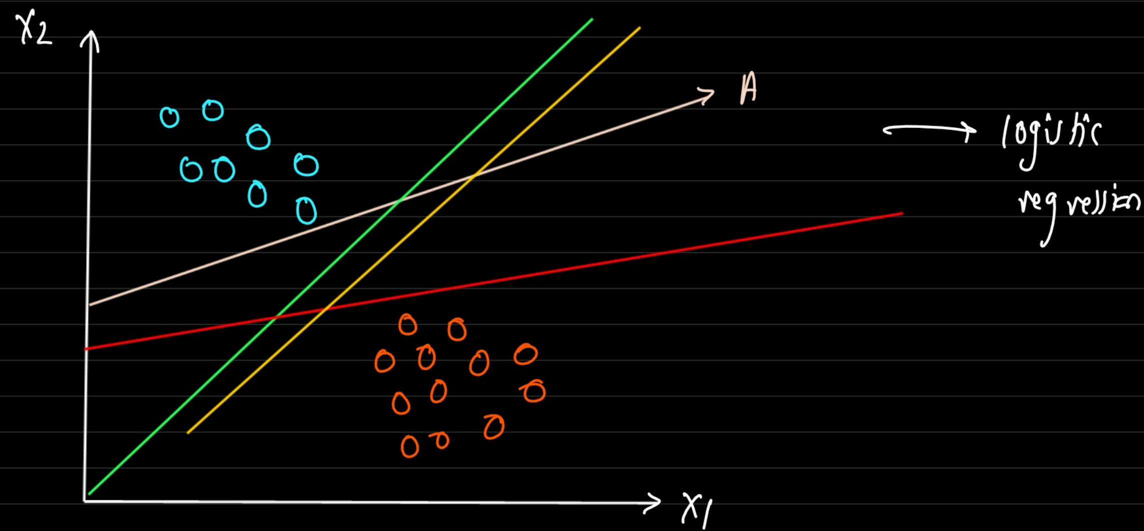
25-10-2025

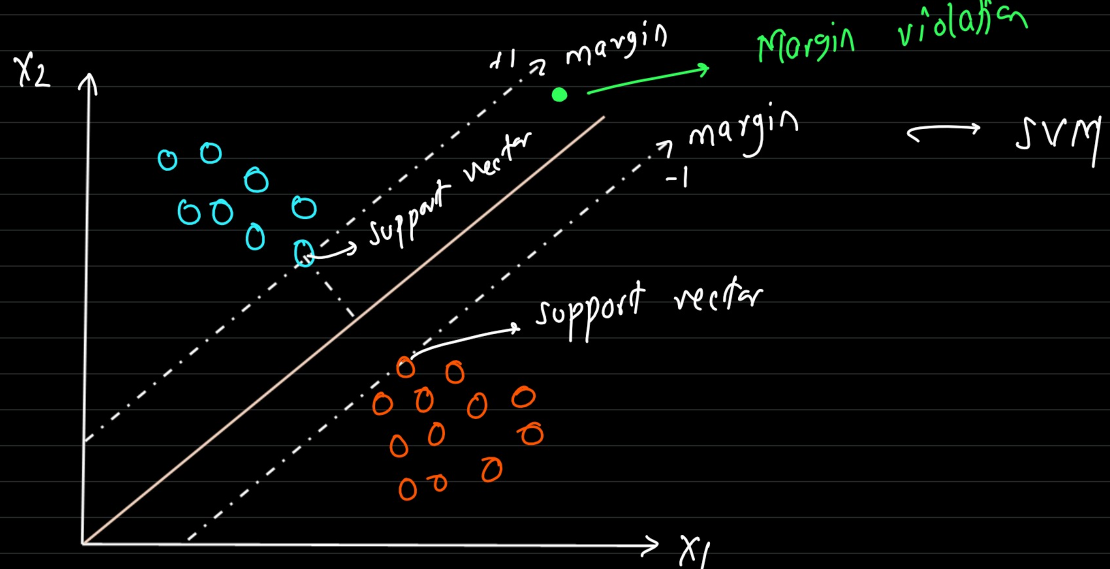
Agenda:

SVM → Support Vector Machines

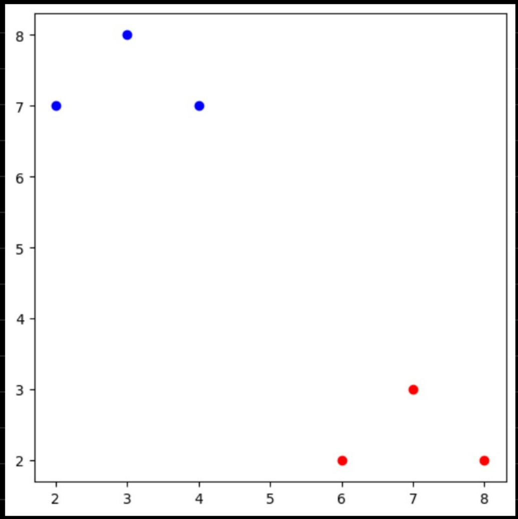
→ Supervised ML Algo

→ Classification & Regression





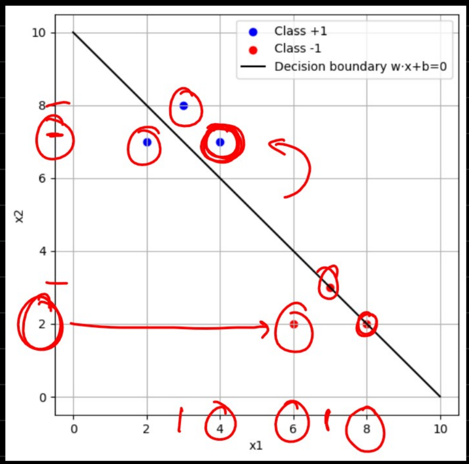
point	x_1	x_2	Class - y
A	2	7	+1
B	3	8	+1
C	4	7	+1
D	6	2	-1
E	7	3	-1
F	8	2	-1



$w_1 = 1, \quad w_2 = 1, \quad b = -10$

$w_1 x_1 + w_2 x_2 + b = 0 \longrightarrow \quad w \cdot x + b = 0$

$\longrightarrow$ eqn of line



point	x_1	x_2	Class - y	$w \cdot x + b$
A	2	7	+1	$1 \times 2 + 1 \times 7 - 10 = -1$
B	3	8	+1	+1
C	4	7	+1	+1
D	6	2	-1	-2
E	7	3	-1	0
F	8	2	-1	0

$w_1 \cdot x_1 + w_2 \cdot x_2 + b$

Interpret:

$w \cdot x + b > 0$: point lies above the line $\longrightarrow$ class +1

$w \cdot x + b < 0$: point lies below the line $\longrightarrow$ class -1

$w \cdot x + b = 0$: point is exactly on the boundary

Score is directly proportional to the distance of a point from the decision boundary

A point x_1, score $+0.1$

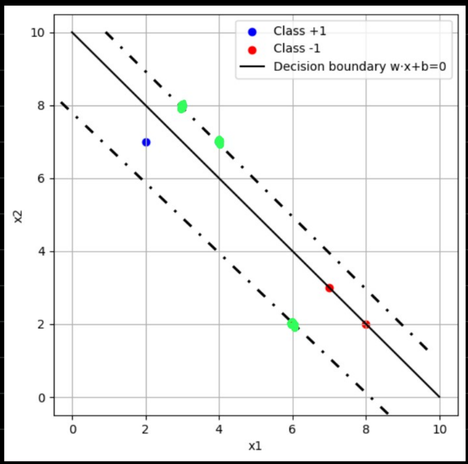
x_2, score $+100$

SVM doesn't just want a line where score > 0, it wants to fit the line to be good such that every point is at least some minimum distance away from the boundary.

positive points ($y_i = +1$) must have a score $\geq +1$

negative points ($y_i = -1$) must have a score ≤ -1

The points that are exactly +1, -1 are the points that have the lowest acceptable confidence. These are support vectors.



formulas (2):

logistic regression:

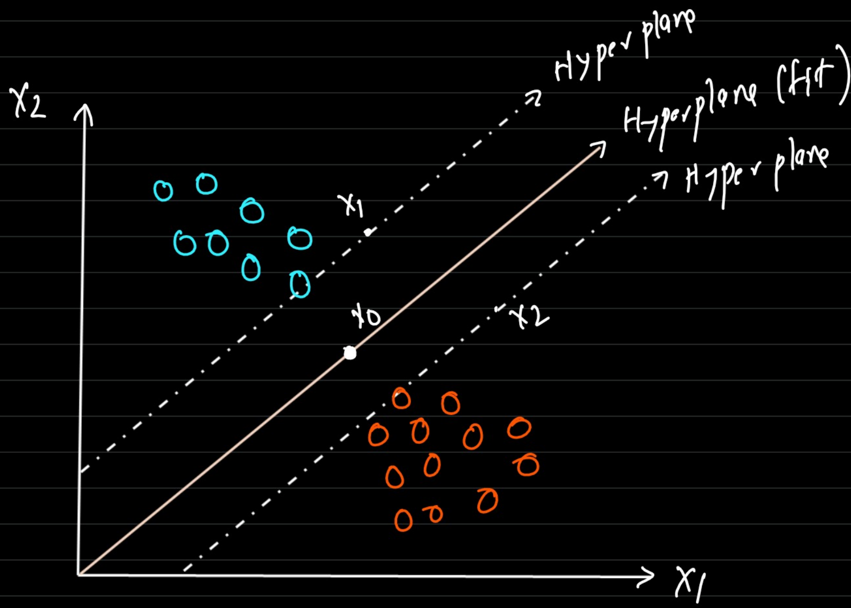
(1) $z = mx + b$

(2) $\hat{y} = \text{sigmoid}(z)$

(3) calculate loss

(4) g = compute gradient from loss function

(5) m_new = m_old − $\eta \cdot$ gradient



x_0 :

$w \cdot x_0 + b = 0$

x_1 :

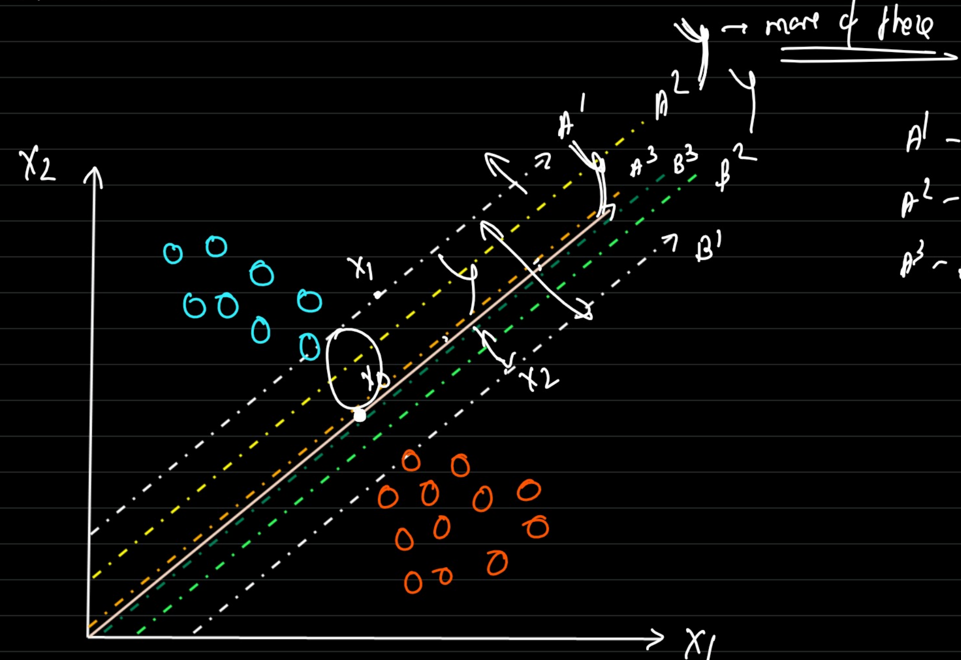
$w \cdot x_1 + b = 1$

x_2 :

$w \cdot x_2 + b = -1$

x_1 & x_2 distance:

$$\frac{|b_1 - b_2|}{\|w\|} = \begin{matrix} b_1 = 1 \\ b_2 = -1 \end{matrix} \Rightarrow \frac{2}{\|w\|} \rightarrow$$ width between 2 classes



$A^1 - B^1 \rightarrow$ support vector

$A^2 - B^2 \rightarrow$

$A^3 - B^3 \rightarrow x^0$

maximize : $\dfrac{2}{\|w\|}$; minimize : $\dfrac{\|w\|}{\cancel{2}}$

Optimization algo generally work better with minimization problem.

decrease in $||w||$, increase $\frac{2}{||w||}$

→ $||w||$

→ if we divide by 2, it cancels out the 2 in the derivative later, just for convenience.

→ minimize $\boxed{\frac{1}{2}||w||^2}$

<u>Next:</u>

x_i belongs to class $+1$: (positive)

$$w \cdot x_i + b \geq 1$$

x_i belongs to class -1 : (negative)

$$w \cdot x_i + b \leq -1$$

Combine both into one expression:

step 1: multiply both cases by y_i.

case 1: $y_i : +1$

$$y_i(w \cdot x_i + b) = (+1)(w \cdot x_i + b) = w \cdot x_i + b \geq 1$$

$$y_i(w \cdot x_i + b) \geq 1$$

case 2 : $y_i = -1$

$$y_i(w \cdot x_i + b) = (-1)(w \cdot x_i + b) = -w \cdot x_i - b \leq -1$$

rewriting above eqn :

$$-(w \cdot x_i + b) \geq 1$$

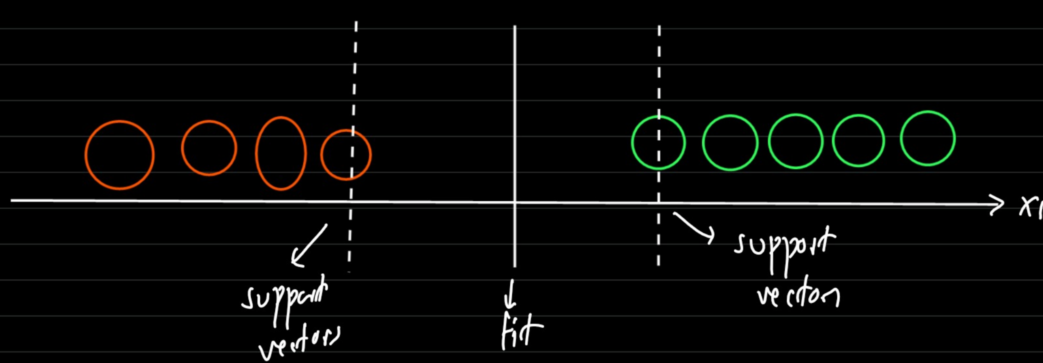
$$\boxed{y_i(w \cdot x_i + b) \geq 1}$$

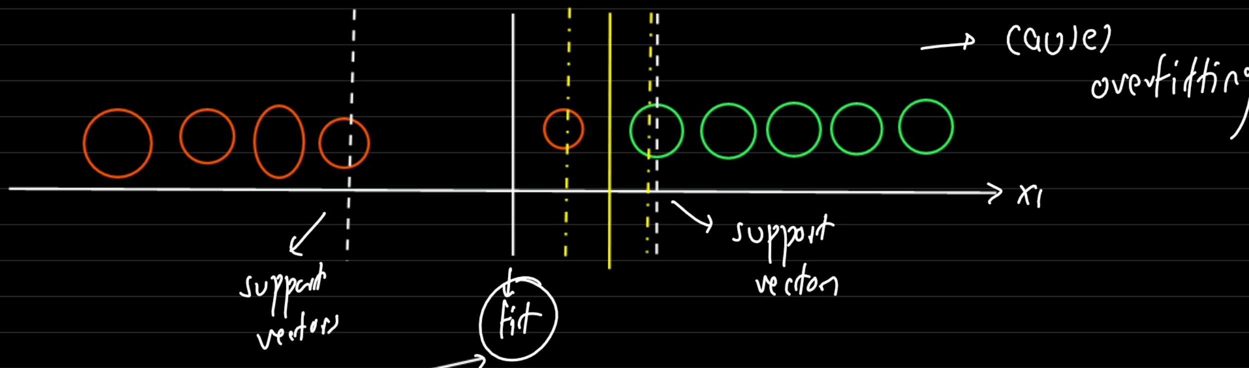
SVM:

we want to minimize , subject to

$$\boxed{\frac{1}{2}\|w\|^2} \qquad \boxed{y_i(w \cdot x_i + b) \geq 1}$$

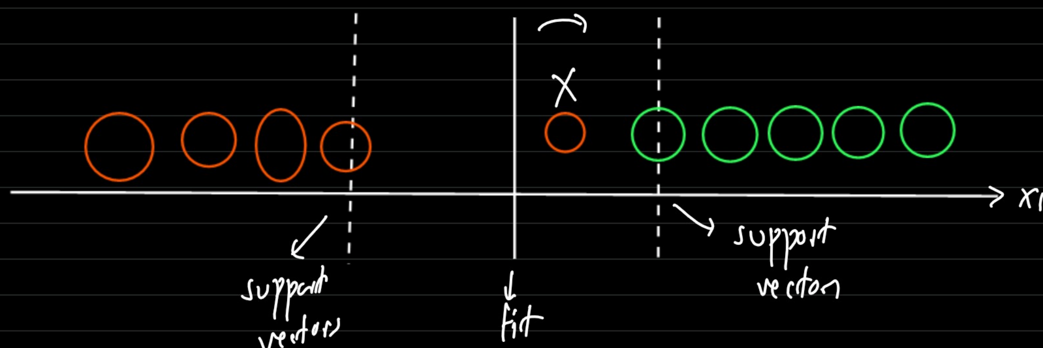
: Hard margin version (perfect seperation)





SVM

Soft margin SVM



: Incorporate small penalty for that mistake
: Acceptable margin of error is allowed.

$y_i = +1$
$w \cdot x_i + b = -50$
$\max(0, 1+51)$
$\to \underline{51}$

cost (w, b) : $\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i (w \cdot x_i + b))$

$\frac{1}{2} \|w\|^2$ ↓ makes the margin as wide as possible

$C \sum_{i=1}^{n} \max(0, 1 - y_i (w \cdot x_i + b))$
↓
Hinge loss
↓
Penalizing points that violates margin
↓
Acts like Regularization

C → How strongly we want to penalize margin violation..

$c = 0.1, 1$ → willing to accept wider margin but more missclassification.

$c = 1000$ → leads to narrower margin & result in classifying everything correctly.

point	x_1	x_2	class - y
A	2	7	+1
B	3	8	+1
C	4	7	+1
D	6	2	-1
E	7	3	-1
F	8	2	-1

$w = [0, 0], b = 0$

$\eta = 0.01$

$C = 1$

For each sample (x_i, y_i):

if $y_i(w \cdot x_i + b) \geq 1$:

No loss: $w = w - \eta w$, $w = w$

$b = b$

if $y_i(w \cdot x_i + b) < 1$:

loss incurred: $w = w - \eta(w - C y_i x_i)$

$b = b + \eta C y_i$

start:

$A(2, 7), y = +1$

→ $y(w \cdot x + b)$

→ $1(2 \cdot 0 + 7 \cdot 0 + 0)$

$= 1(0 + 0) = 0 < 1$

$w = 0 - 0.01(0 - 1 * (+1) * [2, 7])$ $\quad w = w - \eta(w - C y_i x_i)$

$= [0.02, 0.07]$ $\quad b = b + \eta C y_i$

$b = 0 + 0.01 * (+1) = 0.01$

$w = [0, 0]$, $b = 0$

w_new = $[0.02, 0.07]$, b_new = $[0.01]$

---

w_final = $[-0.28, 0.16]$, bf = $[-0.03]$

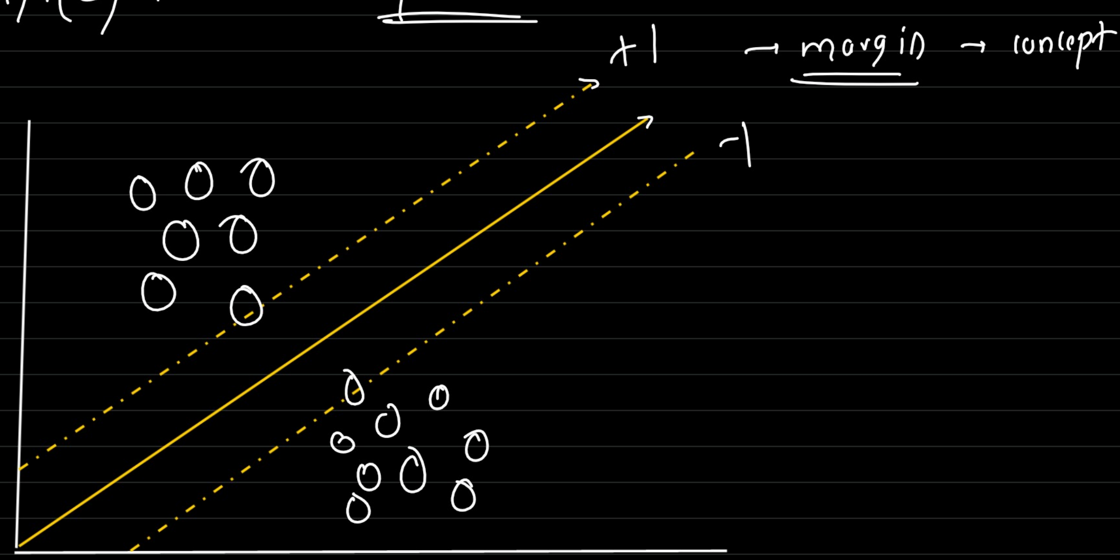
step 1: $z = w \cdot x + b$

$\hat{y} = \text{sign}(z)$

$x = (2, 7)$

$z = (-0.28) * 2 + 0.16 * 7 - 0.03 = 0.53$

$\text{sign}(z) \to +1 \to$ <u>positive</u>



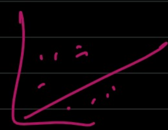
margin is not a learable parameter, it's a concept

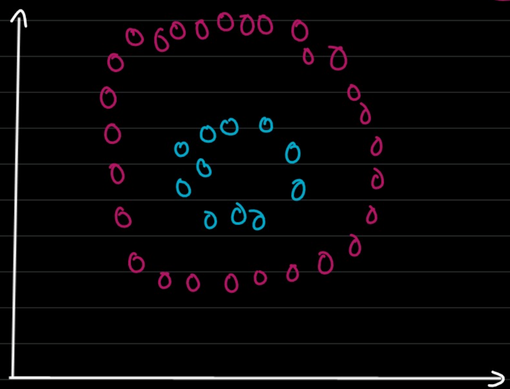
→ It helps us measure how far the support vectors (closest point) are from the seperating line.

margin width $= \dfrac{2}{\|w\|} \to \dfrac{1}{2}\|w\|^2$

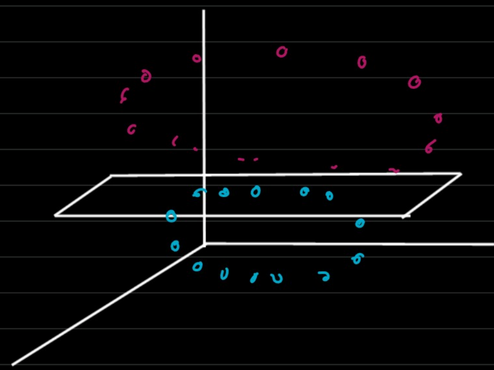
Actual parameters remain the same as logistic regression.

SVM kernels :





Non-linear Data



Non-linear Data (2D) → Transformation (3D) (manual way) (the long way) $2D \to \phi(x) \to 3D$ → kernel function (The shortcut) $(K(x_1, x_2))$ → To solve nonlinear data

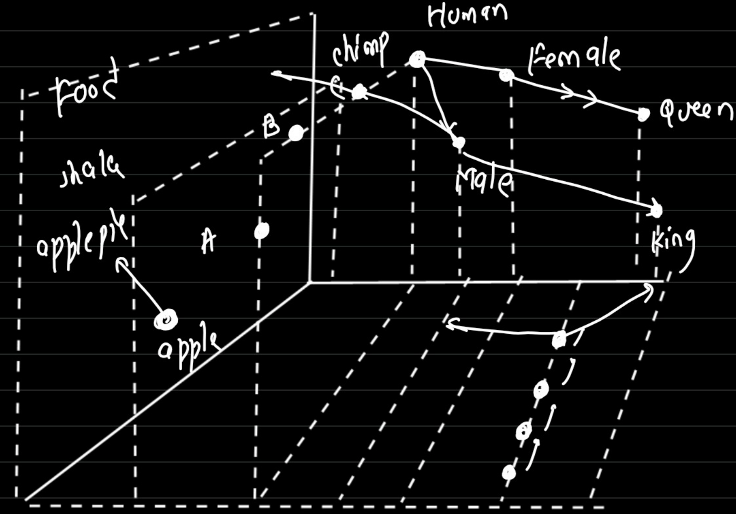
$x_1 (2,3) \longrightarrow x_{1\_new} (x_{1}.a, x_{1}.b)$

Kernel Name	Transformation (ϕ)	Kernel function ($K(x_1, x_2)$)
1. linear	$\phi(x) = x$	$K(x_1, x_2) = x_1 \cdot x_2$
2. polynomial (Degree 2)	$\phi(x) = (x_1^2, x_2^2, x_1 \cdot x_2, x_2 \cdot x_1, \sqrt{2}x_1, \sqrt{2}x_2)$ (6D)	$K(x_1, x_2) = (x_1 \cdot x_2 + 1)^2$
3. Radial Basic function (RBF) / Gaussian	$\phi(x)$ = Infinite dimension map	$K(x_1, x_2) = e^{(-\gamma \cdot \lVert x_1 - x_2 \rVert^2)}$

NLP → embedding → vector

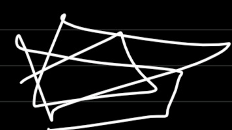
A, Hello, Human, Queen,

Hello → 0
Human → 1
Queen → 2 } encoding



king - apple → opposite

king - Queen → near to each other
(-) (;)
large number



A B → similarity score

SVM, kernel = " "