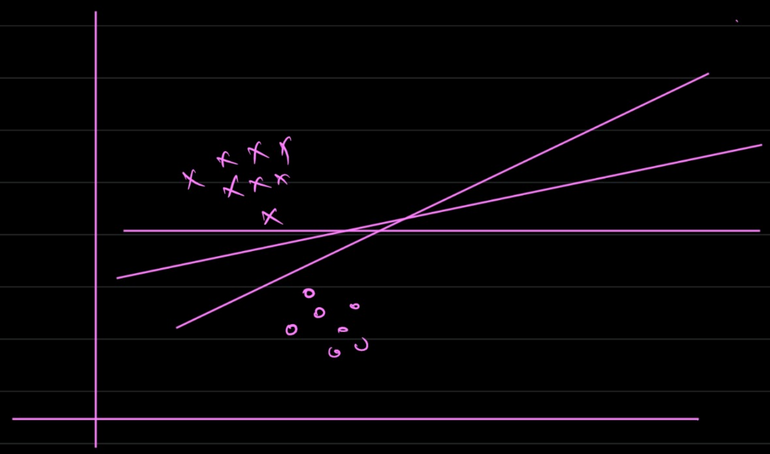
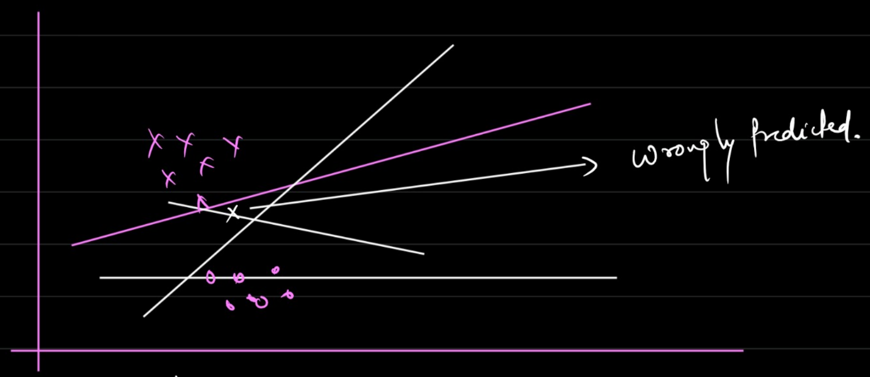
Support Vector Machines

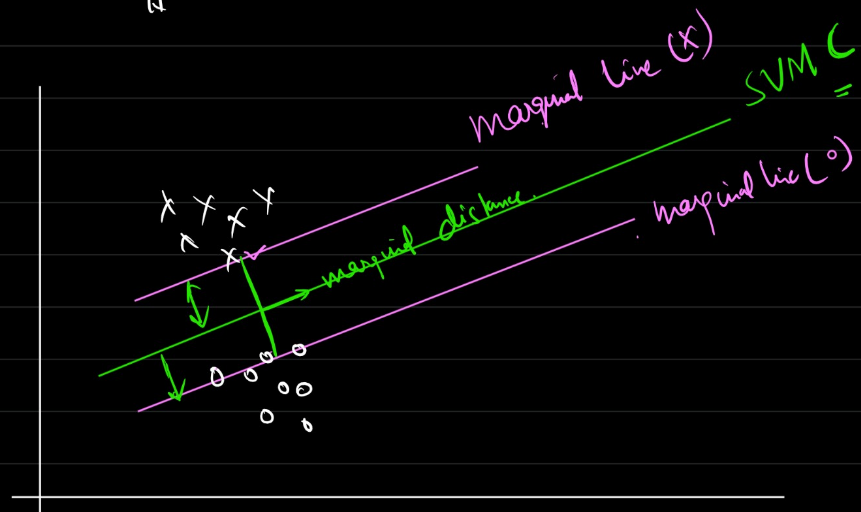
logistic Reg

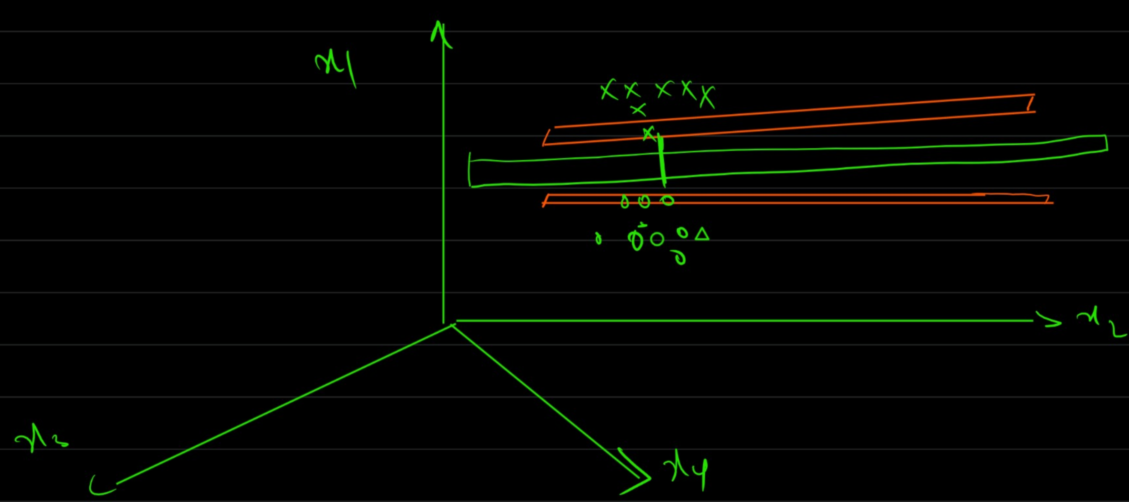


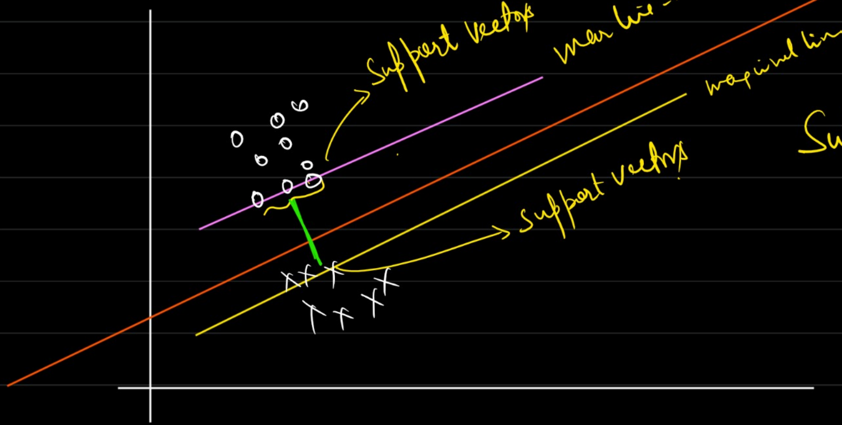


||

SVM Classifier







Support vectors

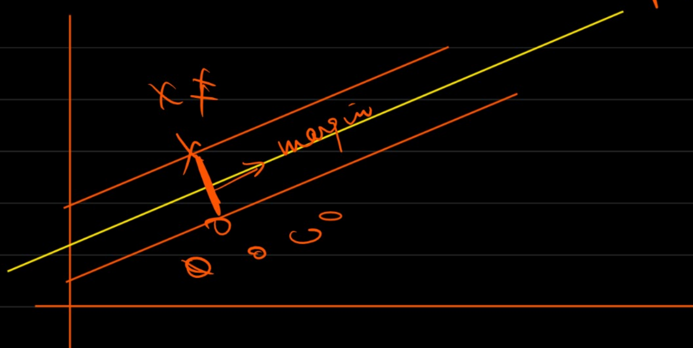
→ nearest data point of either class

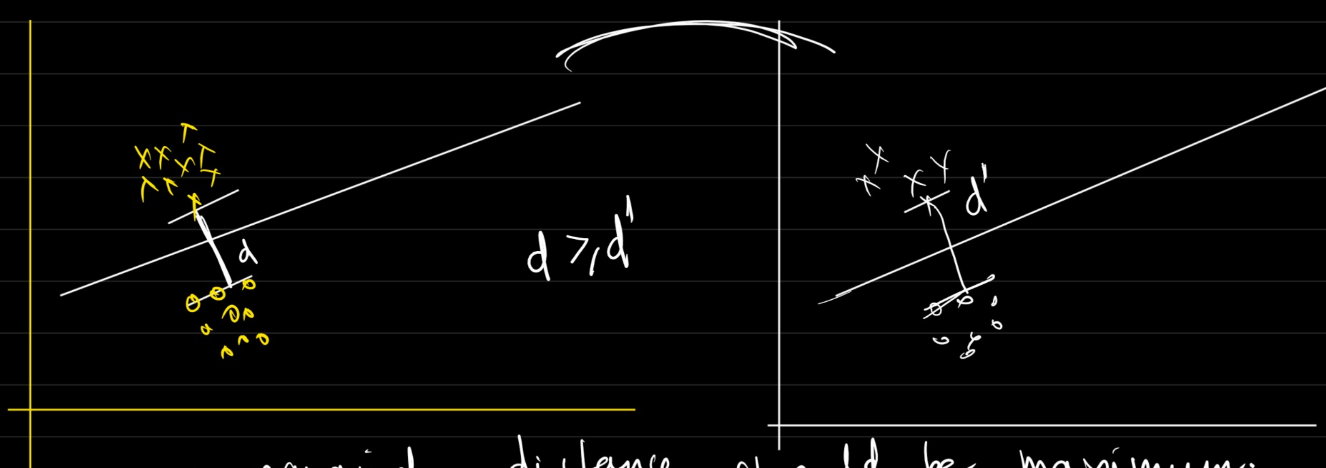
→ It is called Support vectors because these neared datapoints helps to create the right classifier.

→ There are no limits on Support vector

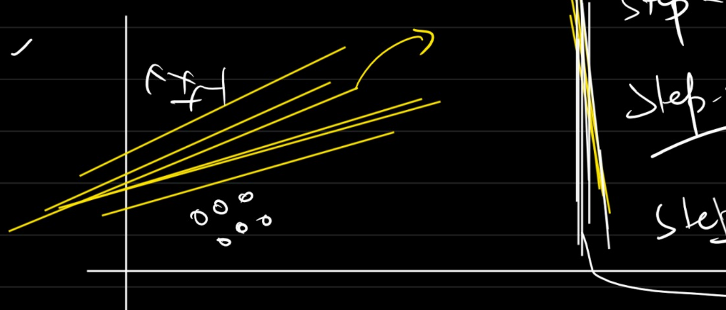
→ atleast no of support vectors = 2.

* SVC is also known as margin classifier





✓ * marginal distance should be maximum.

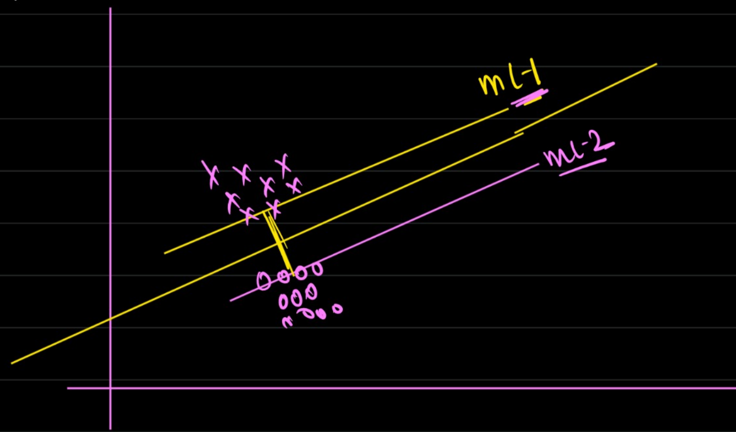


Step-1 → find out all possible classifier

Step-2 → find out margin of each classifier

Step 3 → select the classifier which has maximum margin.

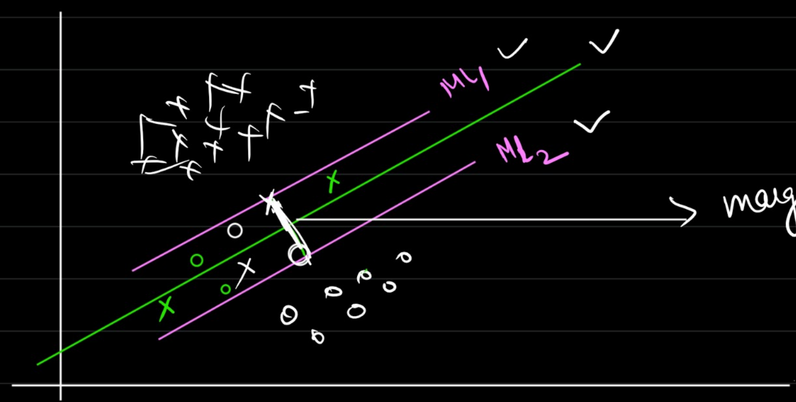
* Maths SVM C



Hard margin classifier

⇓

None of the datapoints are misclassified/overlapping

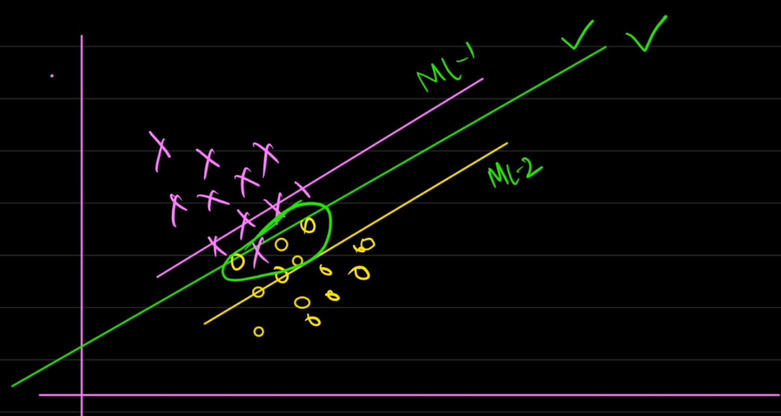


→ marginal distance will be small.

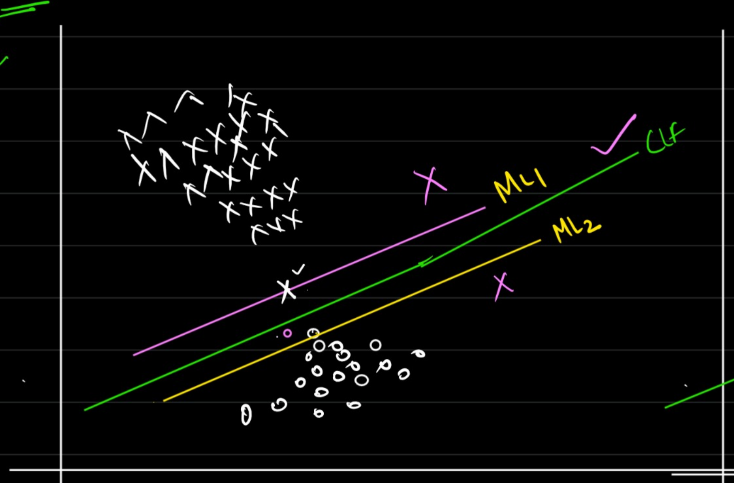
Soft margin

⇓

Some datapoints are allowed to be misclassified (error).

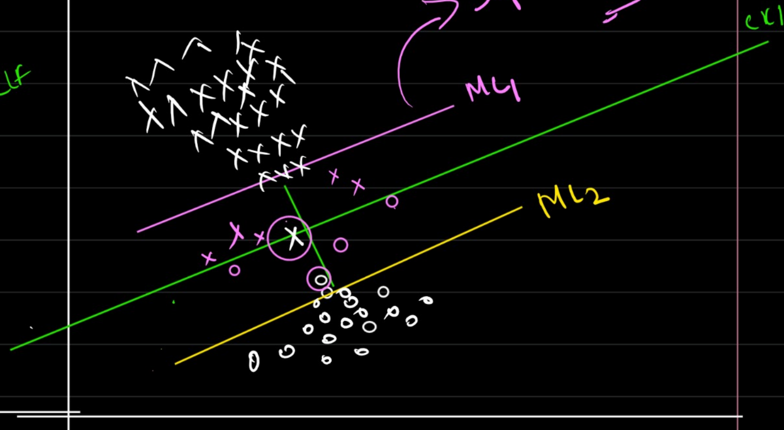


Scen-1

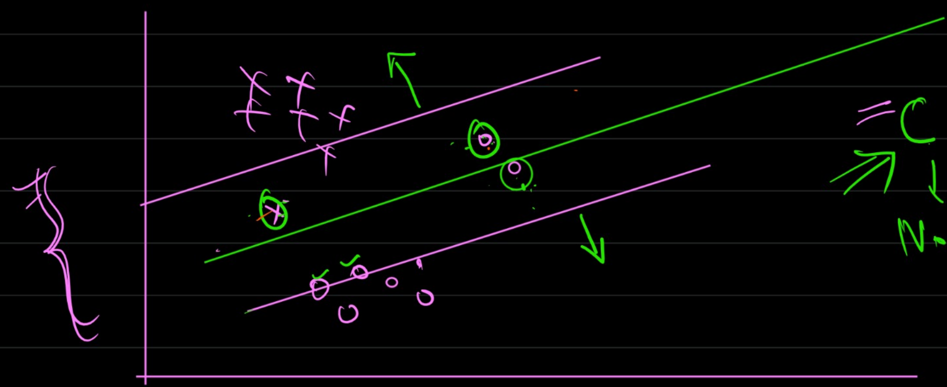


Scen-2

→ Soft margin



→ You are ready to misclassify some of the dp's in order to have good testing accuracy it's a soft margin classifier.



$C = \{1, 2, 3, 4, 5 \ldots 100\}$

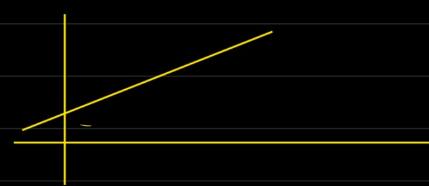
$C = 3$

No. of dp's so sacrifice so that best fit line is not overfitting

* Mathematical explanation

① Equation of line

$y = mx + c, \quad \theta_0 + \theta_1 x$



$ax + by + c = 0$

$by = -ax - c$

$y = \underbrace{\frac{-a}{b}}_{m} x \underbrace{- \frac{c}{b}}_{c}$

$y = mx + c$

In 3d

$y = \theta_0 + \theta_1 x$

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots \theta_n x_n$ → hyperplane

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

$\theta_1, \theta_2 \text{ —— } \theta_n \Rightarrow$ weights

b – bias

w – weights

$\theta_0 \to b$ — bias

$y = b + \boxed{w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4}$

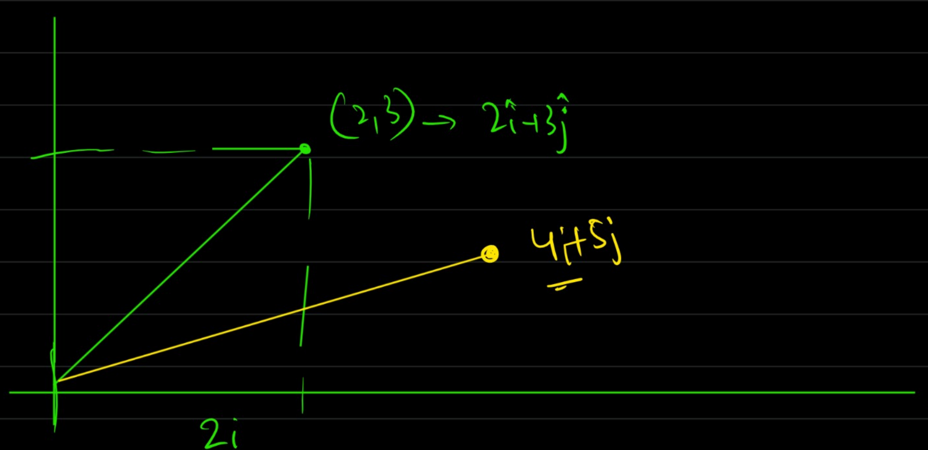
$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$w^T \cdot x$$

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \underline{w_1 x_1 + w_2 x_2 + \cdots w_4 x_4 + b}$$

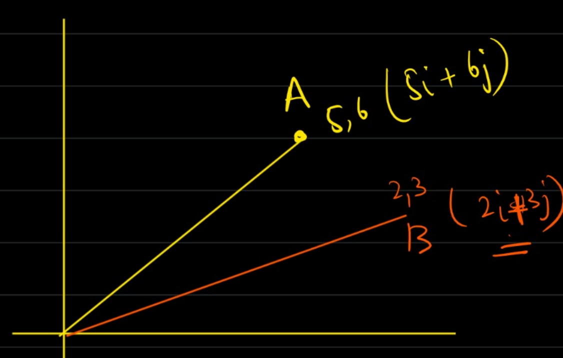
$$\Rightarrow \boxed{y = w^T x + b}$$

②



③ Vector subtraction

$\vec{A} - \vec{B}$



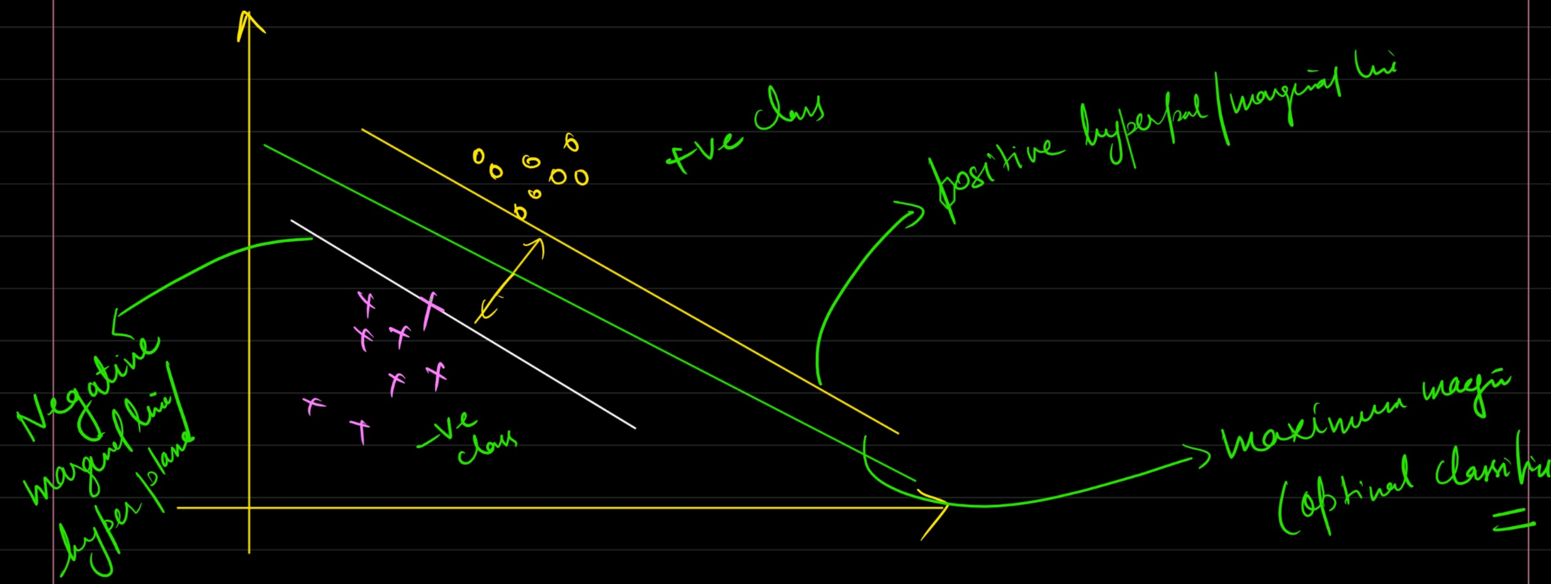
$$\vec{A} - \vec{B} = (5-2)\hat{i} + (6-3)\hat{j}$$
$$= 3i + 3j$$

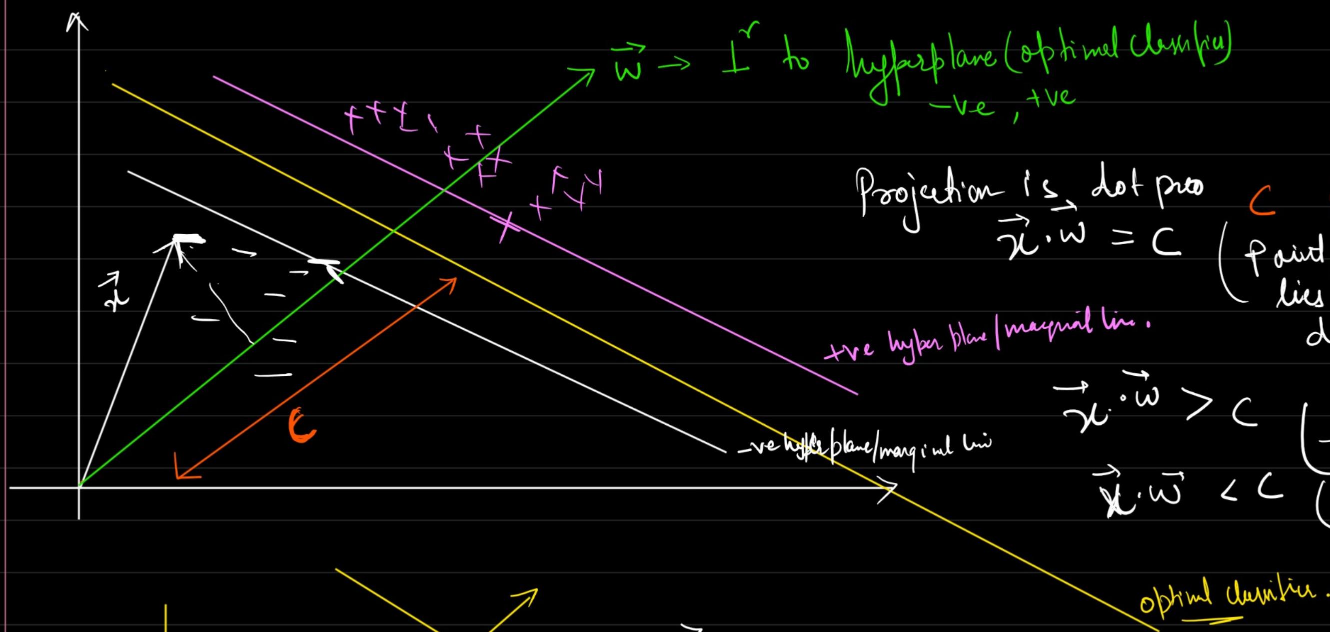
④ Dot Product of two vectors.

$w \cdot u = |w| \cos\theta \cdot |u|$

↓

dot product means projection of u on w.





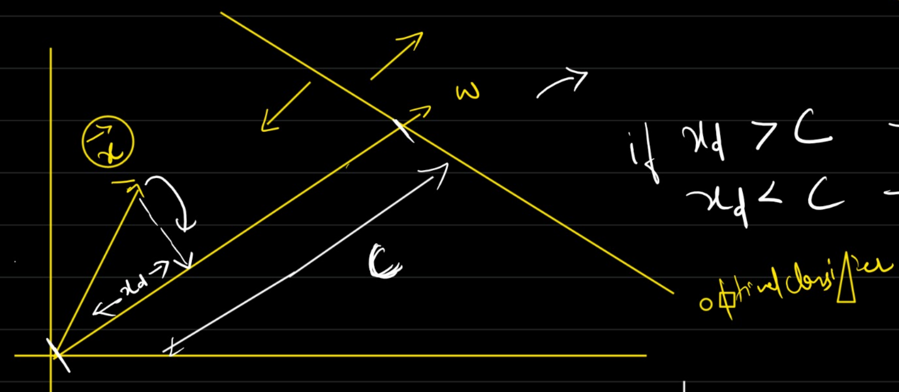
Projection is dot pro

$\vec{x} \cdot \vec{w} = C$ (point lies on decision boundary)

C is distance of optimal

$\vec{x} \cdot \vec{w} > C$ (+ve class)

$\vec{x} \cdot \vec{w} < C$ (negative class)



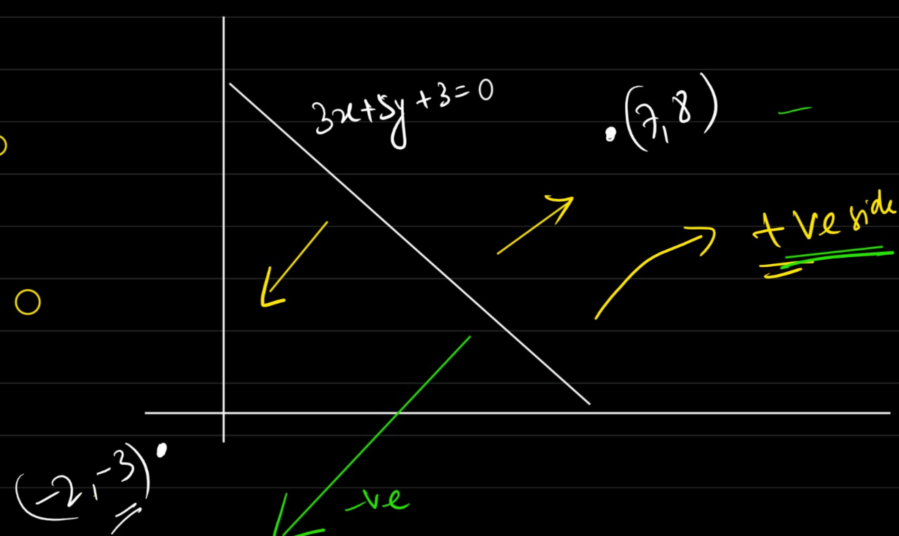
if $x_d > C \to$ +ve class

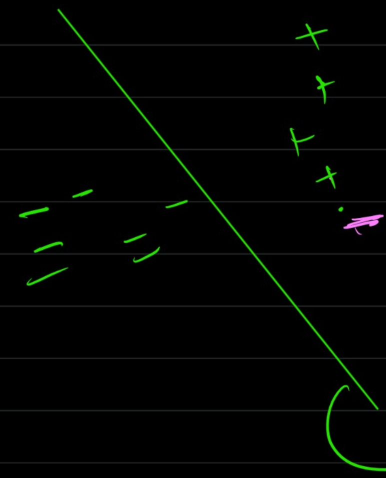
$x_d < C \to$ -ve class.

for (7,8)

$3 \times 7 + 5 \times 8 + 3 \to \geq 0$

$-2 \times 7 + -3 \times 7 + 3 < 0$



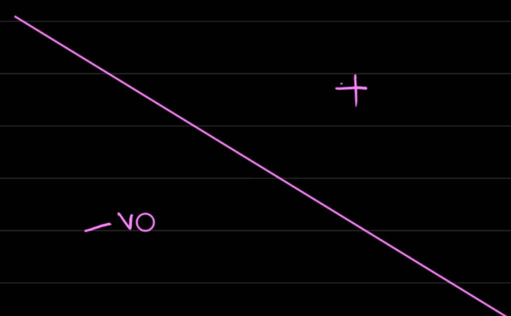


$$\vec{x}\cdot\vec{w} \geq c$$

$$\vec{x}\cdot\vec{w} - c \geq 0 \qquad (-c = b)$$

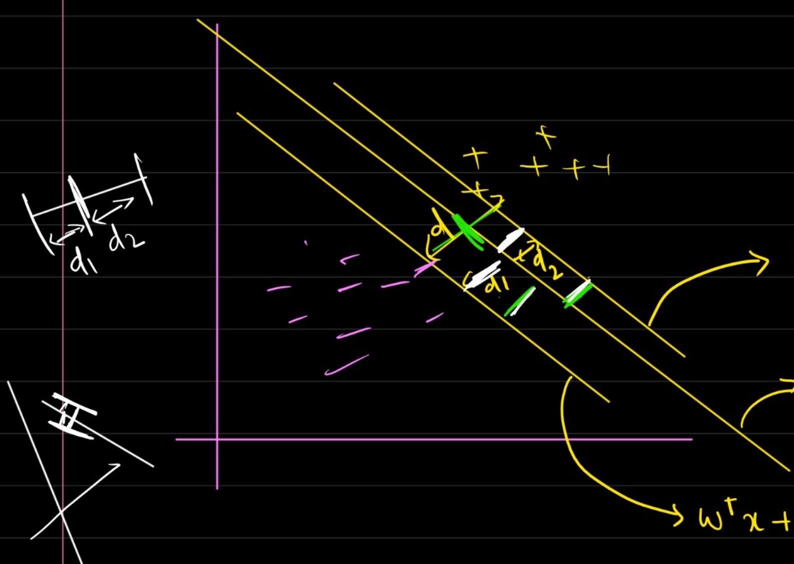
$$w^T x + b = 0$$

$$\vec{x}\cdot\vec{w} + b \geq 0$$



$$w^T * x + b = 0$$

$$y = \begin{cases} +1 & \text{if } \vec{x}\cdot\vec{w} + b \geq 0 \\ -1 & \text{if } \vec{x}\cdot\vec{w} + b < 0 \end{cases}$$



$$w^T x + b = 1$$

$$w^T x + b = 0$$

$$w^T x + b = -1$$

for +ve class = +1

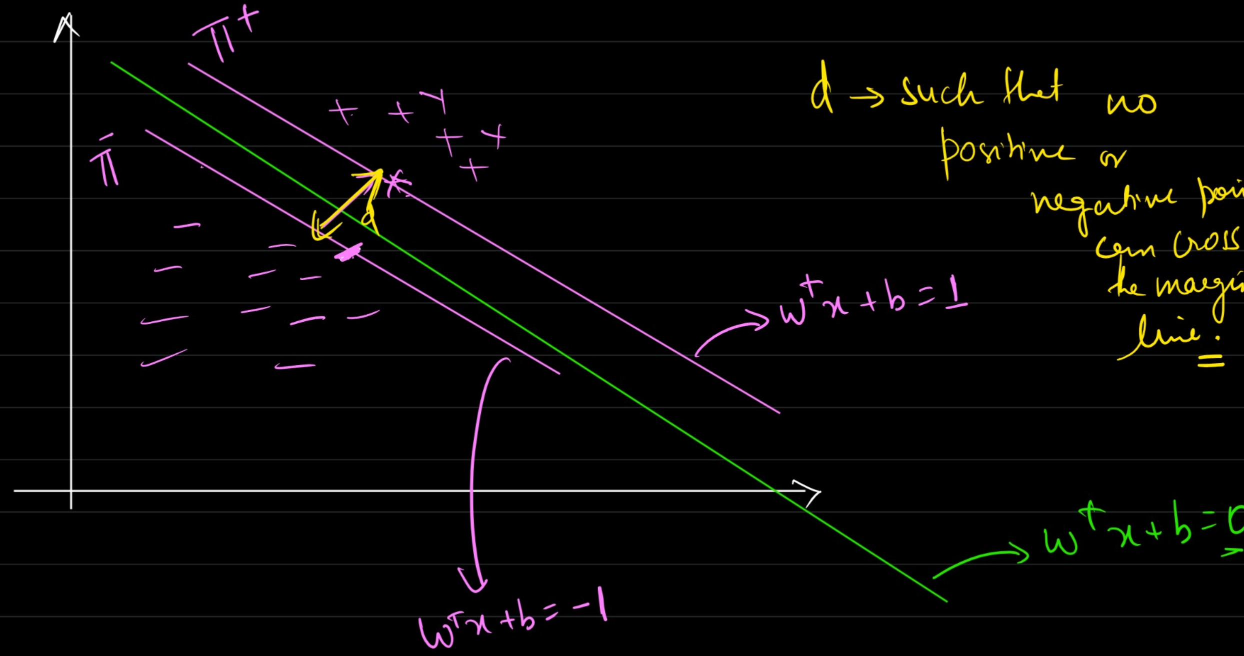
for negative class = -1

Why both equal? (both 1)

d_1 & d_2 should be equidistant.

Why only 1?

why not? -2, +2; -3, 3; -4, 4



$d \rightarrow$ such that no positive or negative point can cross the margin line.

$+$ for +ve class dps $\rightarrow$ $\vec{w} \cdot \vec{x} + b \geq 1$

$-$ve for -ve class dps $\rightarrow$ $\vec{w} \cdot \vec{x} + b \leq -1$

$+1$

-1

We want to maximize 'd' such this constraints holds true.

$$\Downarrow$$

$$\Rightarrow y_i(\vec{w} \cdot \vec{x} + b) \geq 1$$

(equals to mean support vector of either of the class.

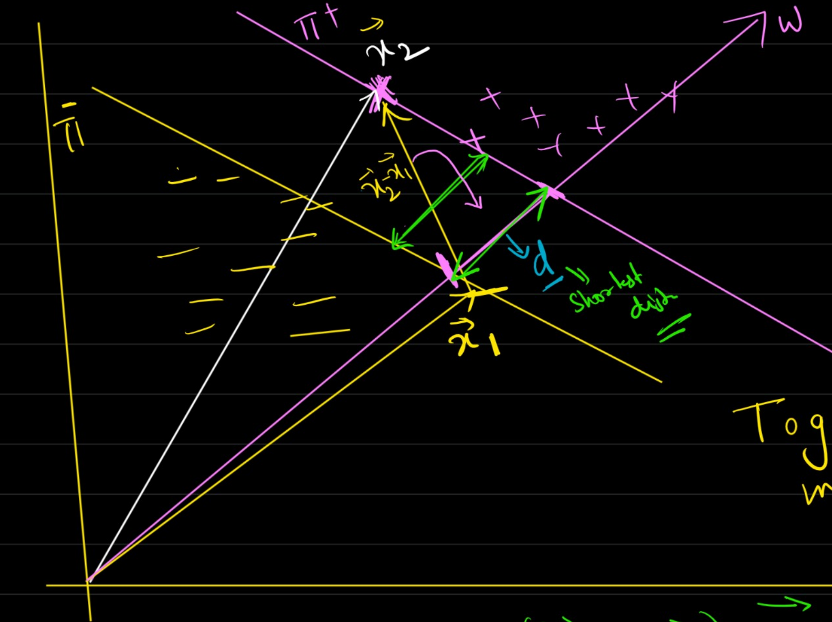
for +ve class $\rightarrow$ $y_i \cdot (\vec{w} \cdot \vec{x} + b) \geq 1$

$y_i = +1$, $(\vec{w} \cdot \vec{x} + b)$ +ve

for -ve class $y_i (\vec{w} \cdot \vec{x} + b) \leq -1$

$$+1 \; (\vec{w} \cdot \vec{x} + b) \geq +1$$

$$\hookrightarrow y_i(\vec{w} \cdot \vec{x} + b) \geq 1 ,$$



maximise d

Such that

$y_i(\vec{w}\cdot\vec{x} + b) \geq 1$

To get shortest distance $\vec{x}_2 - \vec{x}_1$ b/w the marginal plane, we need projection of $\vec{x}_2 - \vec{x}_1$ on unit vector $\vec{w}$ to get d.

$$d = (\vec{x}_2 - \vec{x}_1)\cdot\frac{\vec{w}}{\|w\|}$$

rough: $y_i = 1$
$y_i(wx + b) = 1 \Rightarrow 1(wx+b) = 1$
$wx + b = 1$
$\Rightarrow wx_1 = 1 - b$
$y_i(wx+b) = 1$
$-1(wx_2+b) = 1 \Rightarrow -wx_2 - b = 1$
$-wx_2 = 1 + b$
$wx_2 = \ldots$

$$= \frac{x_2\cdot\vec{w} - \vec{x}_1\cdot\vec{w}}{\|w\|}$$

→ where x_1 and x_2 are support vectors, they lie on marginal hyperplane.

⇓

Constraint $y_i(\vec{w}\cdot\vec{x} + b) = 1$

Putting ② & ①

$$\frac{(-b-1)-(1-b)}{\|w\|} = \frac{-2}{\|w\|} = d$$

for +ve class $y_i = 1$
$1 \times (wx_1 + b) = 1$
$wx_1 = 1 - b$ ①

for −ve class $y_i = -1$
$-1(wx_2 + b) = 1$
$-wx_2 - b = 1$
$-b = 1 + wx_2$
$-b - 1 = wx_2$ ②

$$\boxed{\underset{w,b}{\text{maximise}}\ \frac{2}{\|w\|} \quad \text{such that} \quad y_i(\vec{w}\cdot\vec{x} + b) \geq 1}$$

magnitude of coefficients

⇓

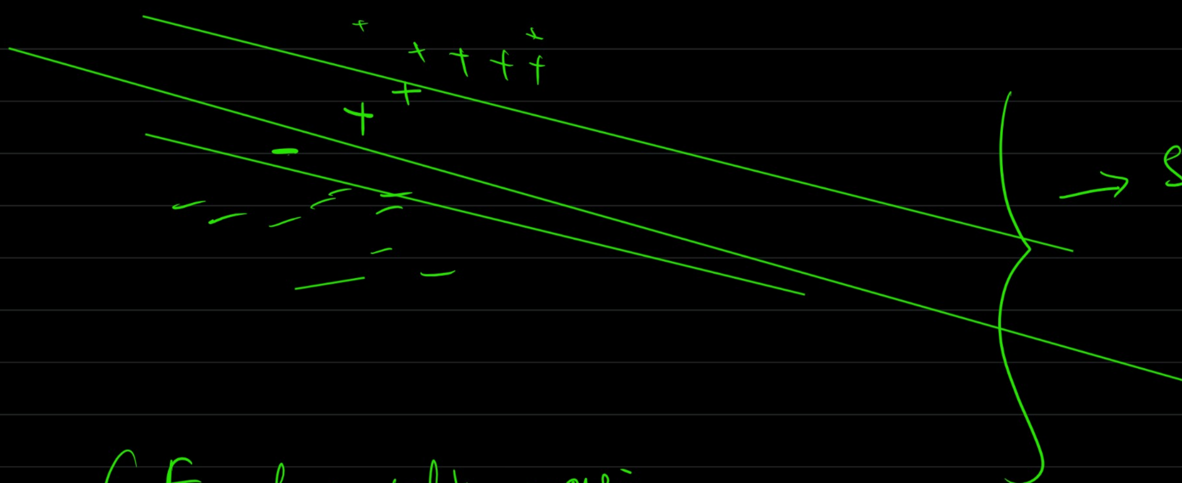
⇒ { modified CF

minimise $\frac{\|w\|}{2}$ by varying w & b

$y_i(wx_i + b) \geq 1$ }

$\Downarrow$ hard margin

max $f(x) \iff \min\left(\frac{1}{f(x)}\right)$



$\rightarrow$ soft margin

CF for soft margin

$$\min_{w,b} \underbrace{\frac{\|w\|}{2}}_{\text{SVM error}} + C \sum \xi_i$$

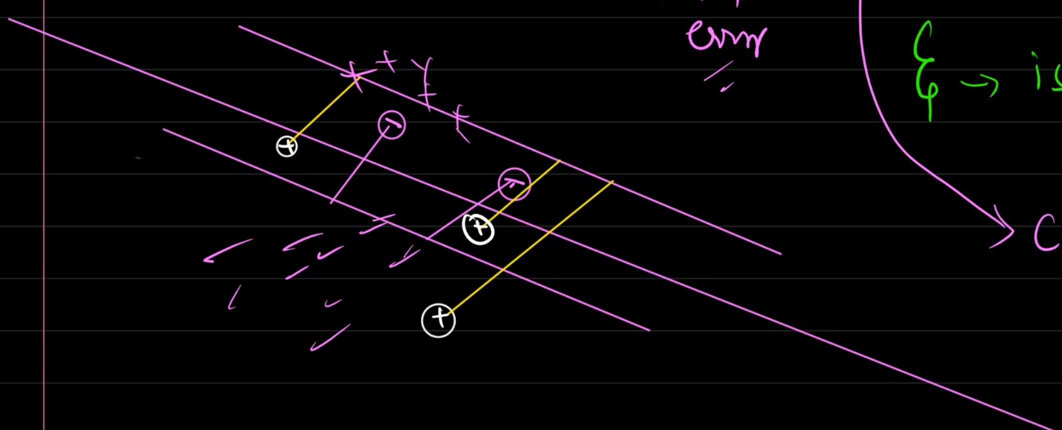
ξ → zeta

ξ_i → mis classification error — new margin

C → no. of misclassified dp's (hyper parameter)

$\xi \rightarrow$ is the distance of all misclassified dp's due to new margin

$C = \frac{1}{\lambda}$



such that

$$y_i(w^T x_i + b) \geq 1$$

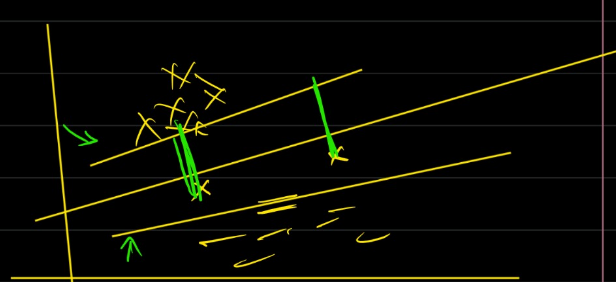
Support Vector Regression

SVC

$$\min_{w,b} \frac{\|w\|}{2} + \underbrace{C \sum_{i=1}^{n} \xi_i}_{\text{hinge loss}}$$

C → no. of misclassified dp's

ξ → zeta



$\varepsilon \rightarrow$ Epsilon
ξ – Zeta
ψ – Shi
η – eta

SVR

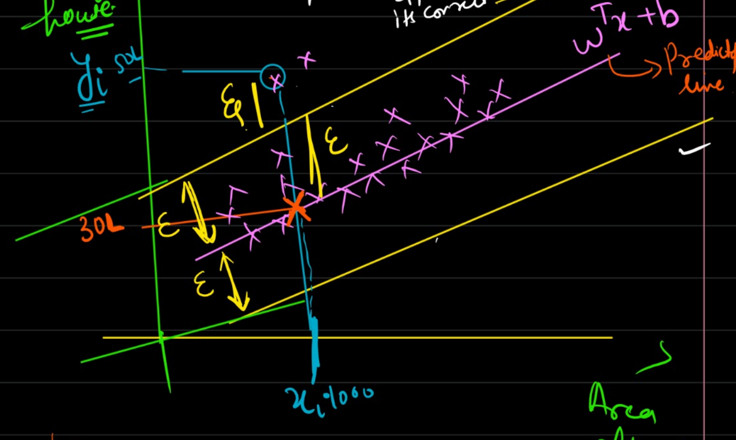
$\downarrow$

→ we want all dp's in surrounding of best fit line

$\Downarrow$

$w^Tx + b + \varepsilon$ to $w^Tx + b - \varepsilon$

Zeta → ξ is the distance of dp's from its correct marginal plane.



$y_{act} - y_{pred} = \text{Error} = \varepsilon + \xi$

$$CF = \min_{w,b} \frac{\|w\|}{2} + C \sum_{i=1}^{n} (\xi_i + \varepsilon) \ \text{min}$$

Constraints

$$|y_i - y_{pred}| \leq \varepsilon + \xi$$

$$|y_i - w^T x_i| \leq \varepsilon + \xi$$

* All the dp's in between of two marginal plane

$\xi \rightarrow$ is the distance of dp from its correct marginal plane

$\Downarrow$

ξ should be minimum

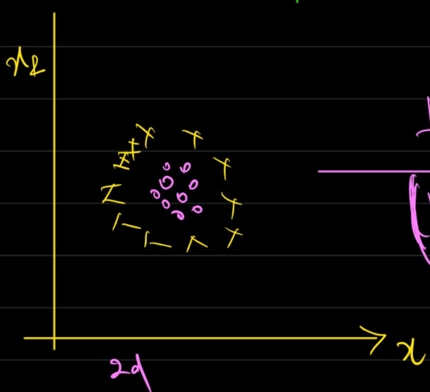
n_class = 2.

$\Downarrow$

SVM — Kernel trick

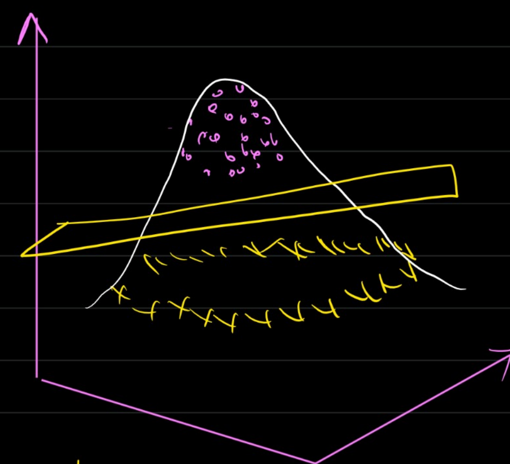
SVM, SVC
- Math
- Intuition

Kernel.



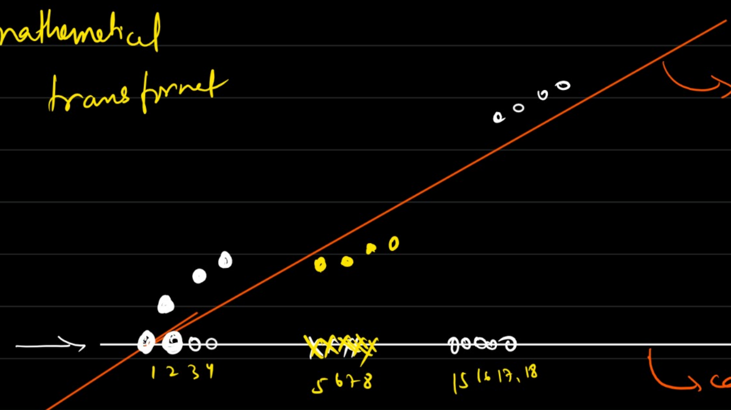
transformation
(Using mathematical formulae)

2d — 3d



$f(x) \Longrightarrow$ Kernel. $\Rightarrow$ Kernel transformation

mathematical transformat



best fit classifier.

cannot segregate

x	y
2	Yes
3	No
4	Yes
	No
5	Yes

$Z = x^2$

Idea:- → To change the data from 1d to 2d, 2d–3d or increase the dimension by mathematical transformation to distinguish b/w the class

Kernel trick
why?

$\Rightarrow$ You are not sending the data in higher dimension. You are using mathematical transformation to achieve it.

① Polynomial Kernel

$$f(x_1, x_2) = (X_1^T \cdot X_2 + c)^d$$

$x_1, x_2 \rightarrow$ two feature.

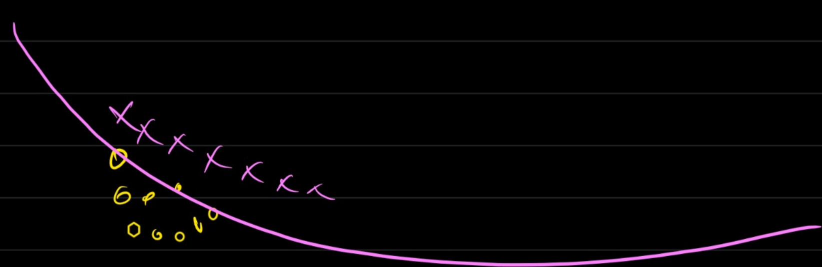
d is degree of polynomial

$c \rightarrow$ Constant.

$$X_1^T \cdot X_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot [x_1 \ x_2]$$

$$\begin{bmatrix} X_1^2 & X_1 X_2 \\ X_1 X_2 & X_2^2 \end{bmatrix}$$

$$X_1\ X_2 \Longrightarrow X_1, X_2, X_1^2, X_2^2, X_1 X_2$$



degree-2

② radial basis feature (rbf)

$\longrightarrow$ Creates non-linear combination of features to lessy feature higher dimension

$$f(x_1, x_2) = e^{-\frac{\|(x_1 - x_2)\|^2}{2\sigma^2}}$$

$\|x_1 - x_2\| \rightarrow$ Euclidean distance b/w x_1 and x_2

σ — variance.

③ Sigmoid Kernel

$$f(x) = \frac{1}{1+e^{-x}}$$

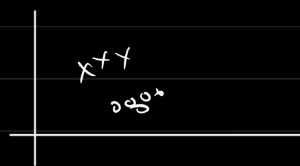
* How to choose right kernel.

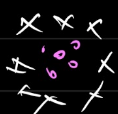
⇓

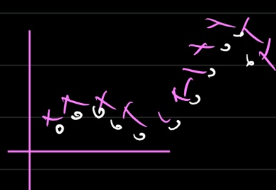
Hyperparameter tuning.

Tricks

↳ distinguishable → linear kernel



✓ →  → Rbf kernel

✓ →  → Polynomial kernel.

* Bessel kernel → $f(x,y) = \dfrac{J_{v+1}(\sigma\|x-y\|)}{\|x-y\|^{-n(v+1)}}$

* ANOVA kernel → $f(x,y) = \sum_{k=1}^{n} \exp\left(-\sigma\left(x^k - y^k\right)^2\right)^d$

predict forecast

time series

Shaped

name | id | duration | more

Call count, avg cost . . .

month

na

time series → Regression

Predict the demand

SARIMAX

time

Demand

Jan

Feb

March

April

dec

Exogenous

Regressio

featu

JAN

Mar

Mar

Ap

Apr

MA

Septh