Non-Linear SVM

→ For non-linear datasets in SVM, the commonly used kernels are:

- Polynomial kernel:

  → $K(x, x') = (x^T x' + c)^d$ or $K(x, y) = (x^T y + c)^d$

⇒ (In the formula nothing to do with output it; have 5 inputs x and y both are considered inputs in "poly" and "RBF")

- RBF (Radial Basis Function) or Gaussian Kernel:

  → $K(x, x') = \exp(-\gamma \|x - x'\|^2)$ → $K(x, x') = e^{-\gamma \|x - x'\|^2}$

  → or $K(x, y) = e^{-\gamma \|x - y\|^2}$

- Sigmoid Kernel:

  → $K(x, x') = \tanh(\gamma x^T x' + r)$

⇒ These kernels allow SVM to separate data that is not linearly separable by transforming it into a higher-dimensional space.

*- Code

```
from sklearn import svm

model = svm.SVC(kernel='poly', degree=2, coef0=1,
                C=1)
```

1 - coef0 = 1

→ This sets the constant c in the formula :- $(x^T x' + c)^d$

→ If c is large, it makes the kernel focus more on higher-degree terms, affecting the shape / flexibility of the boundary usually keep 0 or 1 for stability

2 - degree = 2

→ This sets the degree of the polynomial kernel

- For degree 2, the kernel computes $(x^T x' + c)^2$, which creates a quadratic (curved) decision boundary

- Higher degrees (eg 3, 4) make even more complex boundaries

- Degree = 2: Quadratic boundary (parabolic curve)
- Degree = 3: Cubic boundary (even more flexible)
- Degree = 4: or higher: Increasingly complex boundaries

⇒ so, if your data requires a more complex decision ~~boundary~~ surface, try higher degrees, regardless of the original input dimension.

★ Polynomial kernel - Indepth math

x_1 x_2 y :- we have 2 inputs
↳ x_1 x_2
and one ~~inp~~ output
↳ y

dot products

Polynomial kernel formula :

→ $K(x, x') = (x^T x' + c)^d$ (vector operations)

or

→ $K(x, x') = (x^T x' + 1)^d$

Transpose operation

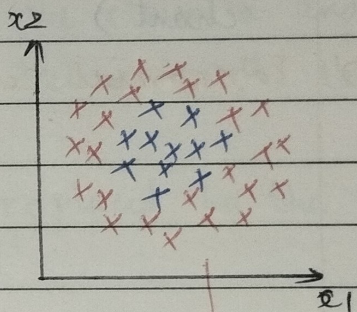
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

$$= \begin{bmatrix} x_1^2 & x_1 x_2 \\ x_1 x_2 & x_2^2 \end{bmatrix}$$

-This are imp components

★ { $x_1 x_2$ → repeated feature (we take only one)

x_1	x_2	y	x_1^2	x_2^2	$x_1 x_2$



↓

Non-Linear dataset (2 Dimensional)

↓

its impossible to create a hyperplane and with to respect to margin plane

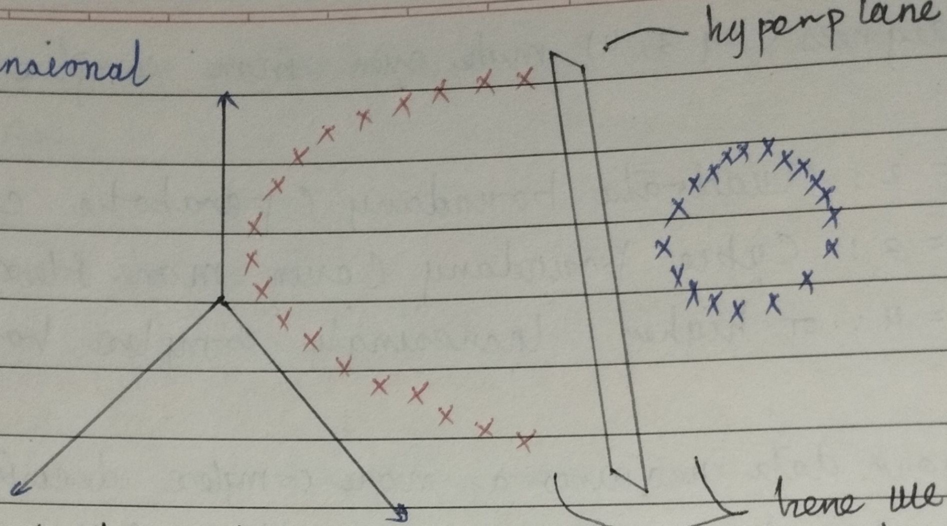
↓

we use SVM kernel techniques to tranform this data points into higher dimension such that → it helps to easily create a hyperplane & margin plane

⇒ Output Next Page

Transformation

3-Dimensional



here we create

→ marginal plane at one of the nearest support vectors

if we take the three datasets x_1^2, x_2^2, x_1 & x_2
(This will ~~plot~~ plot 3 Dimensional chart)

⇒ Polynomial kernel :- Important components
(x_1^2, x_2^2, x_1x_2)

(after this we can create a normal hyperplane using kernel = 'linear')

⇒ $K(x, y) = (x_1 y_1 + x_2 y_2 + x_3 y_3 + x_4 y_4 + c)^d$

→ (If I have 4 inputs)