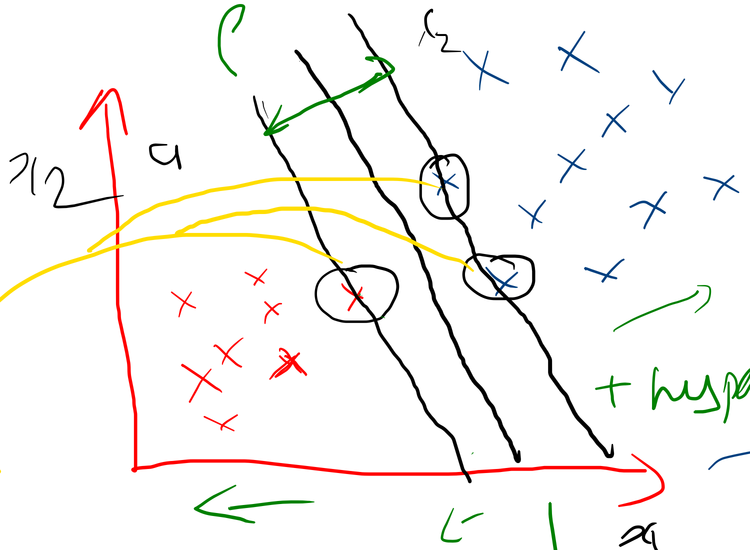
Support vectors



+ hyperplane

hyperplane

optimum hyperplane

Support vector ?

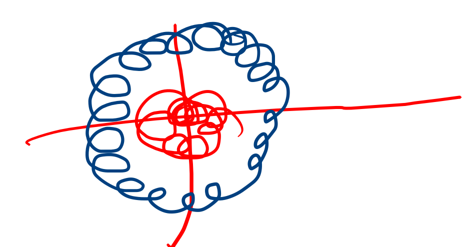
↓

hyperplane

↳ decision boundary

non separable

svm



Hung

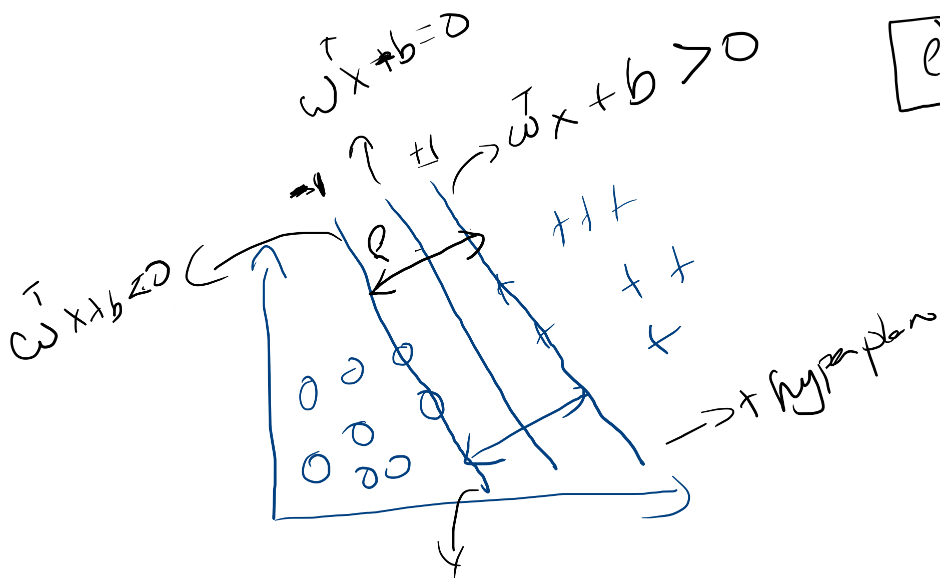
2d ↳ 3P

↳ hyperplane

$x \Rightarrow \phi(x)$

↳ higher dimen

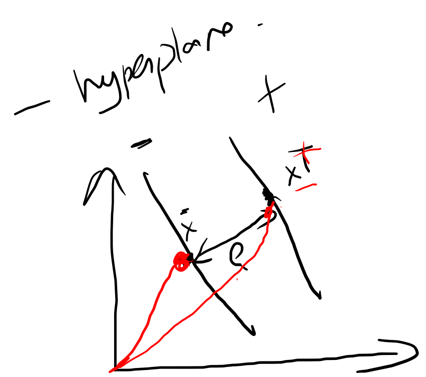
Svm < nonlinear / Linear



$\boxed{\rho}$ -> margin between hyperplane

$w^Tx^+ + b = +1$ -> (1)

$w^Tx^- + b = -1$ -> (2)



$\boxed{x^+ = x^- + \rho w}$

$w^T(x^- + \rho w) + b = +1$

$\rho w^Tw + \boxed{w^Tx^- + b} = +1$

$\rho w^Tw + (-1) = +1$

$\rho w^Tw = +1+1 = 2.$

$$\boxed{\rho = \frac{2}{w^Tw}}$$

margin

margin = $C = \frac{2}{\boxed{w^T w}} = \frac{2}{\|w\|}$ sum max

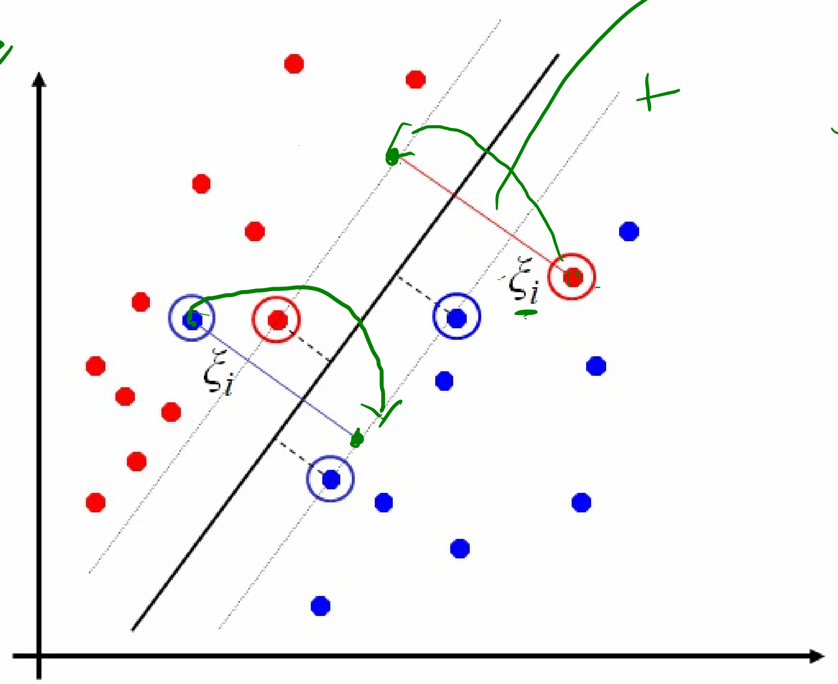
↳ margine

$\min w^T w$ = $\frac{\partial L}{\partial \theta} = 0$ → Loss → min cost

Cost

Slack → misclassification in SVM

Training



misclassification

Soft margin

Slack variable

$p_1 \xrightarrow{d} p_2$

$\boxed{p_1 + d = p_2}$

overfitting

$$C = \frac{2}{\boxed{w^T w + C \sum \xi_i}}$$

c = 0.01
0.01
0.001
0.015
0...

c = 100 training

2 + 2 = 5

Non-linear separable will be handled

svm → high dimension [time taken will be huge]

X → $\phi(x)$ → high dimension

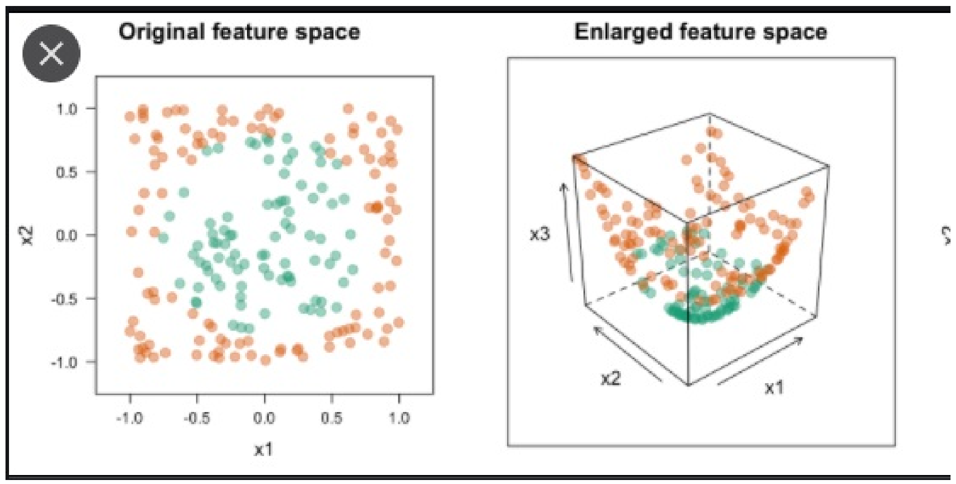
X → kernel (x_1, x_2)

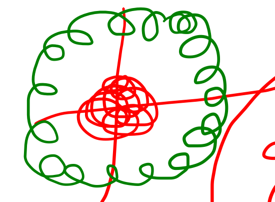
- , a kernel function *implicitly* maps data to a high-dimensional space (without the need to compute each φ(**x**) explicitly).

$\phi(x)$

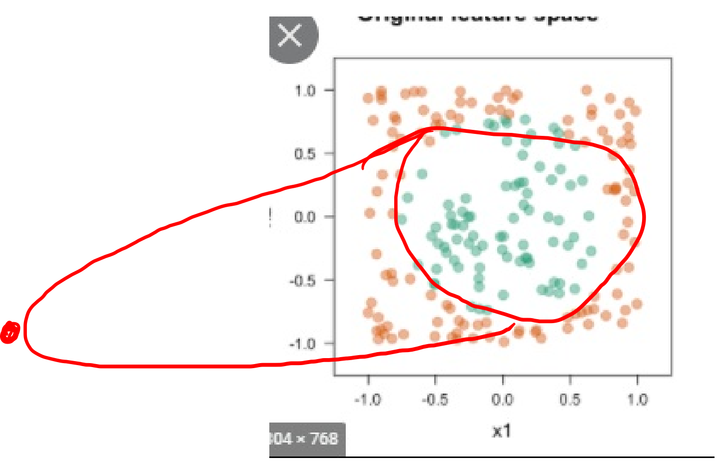
x

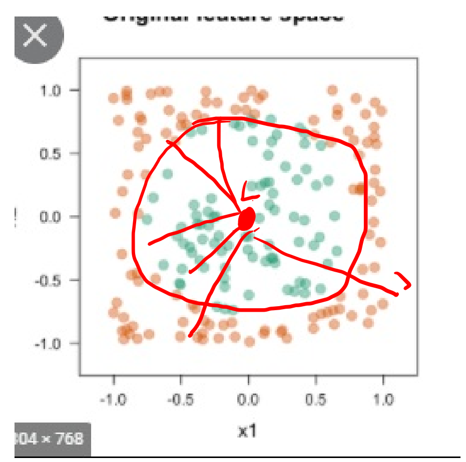
Kernel





$K(x, x_i)$





$$RBF = e^{-\gamma \| x_i - x_j \|^2}$$

$$= e^{\infty} \rightarrow 0$$

$$= e^{0} \rightarrow 1$$

ML pipline template

data → preprocess

- └→ EDA
  - uni
  - multivariant
- └→ feature selection
- └→ feature Transformation
  - stand, minmax
  - Label encodin
  - Binning
  - Label binarize
- → train/test split
- └→ list ML model
  - └→ train
- └→ Best model
  - └→ grid

compare

Roc, Auc

preci, recal,

Accur, F1 score

evaluate

- → good
- → bad