Classification

Cristian López Del Alamo
clopezd@utec.edu.pe
GINIA - Research group

2024

Objective: In this class, the objective is to understand what classification is from the perspective of machine learning, what binary classification is and how it works, and finally to implement logistic regression.

Classification

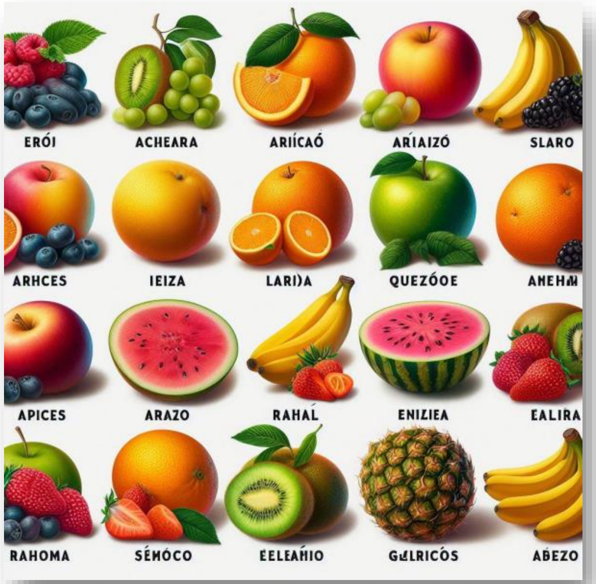
- Aprendizaje Supervisado.
- Proceso de Entrenamiento
- Datos de Entrenamiento.

- Proceso de Testing

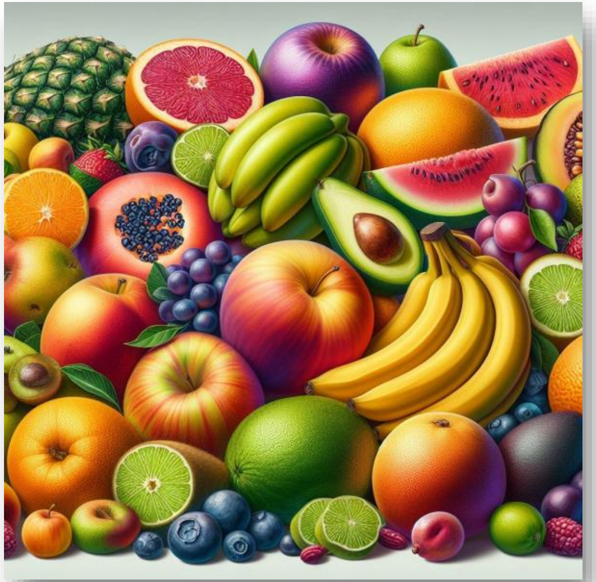
- Proceso de Inferencia

Classification

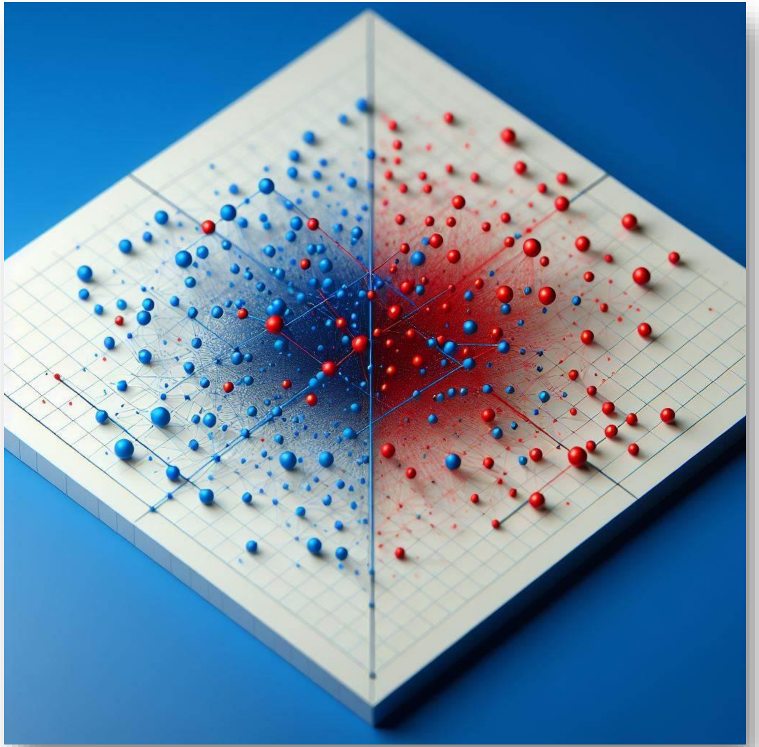
$$TrainingData = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$$

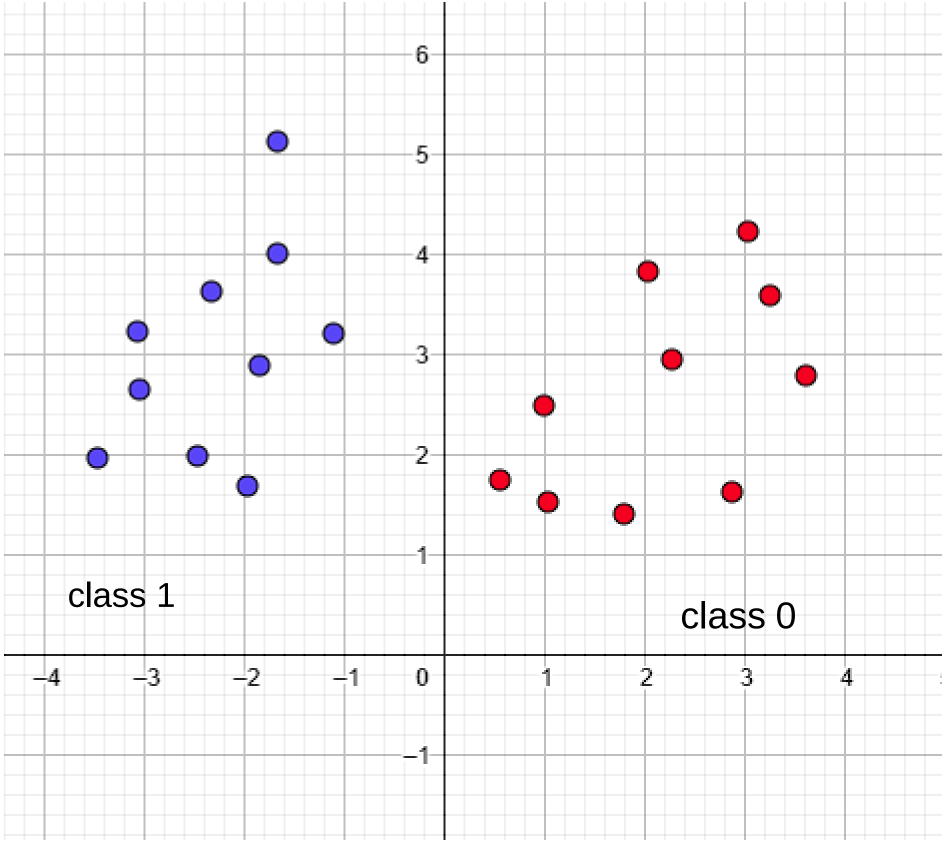


$$TestData = \{x_1, x_2, ..., x_n\}$$

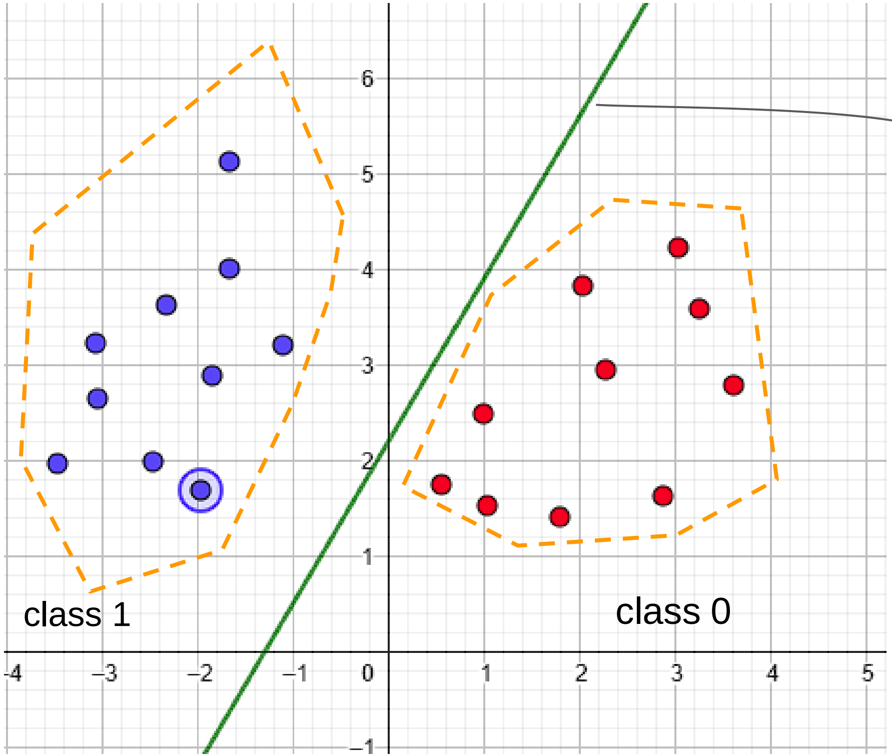


Binary Classification: Logistics Classification



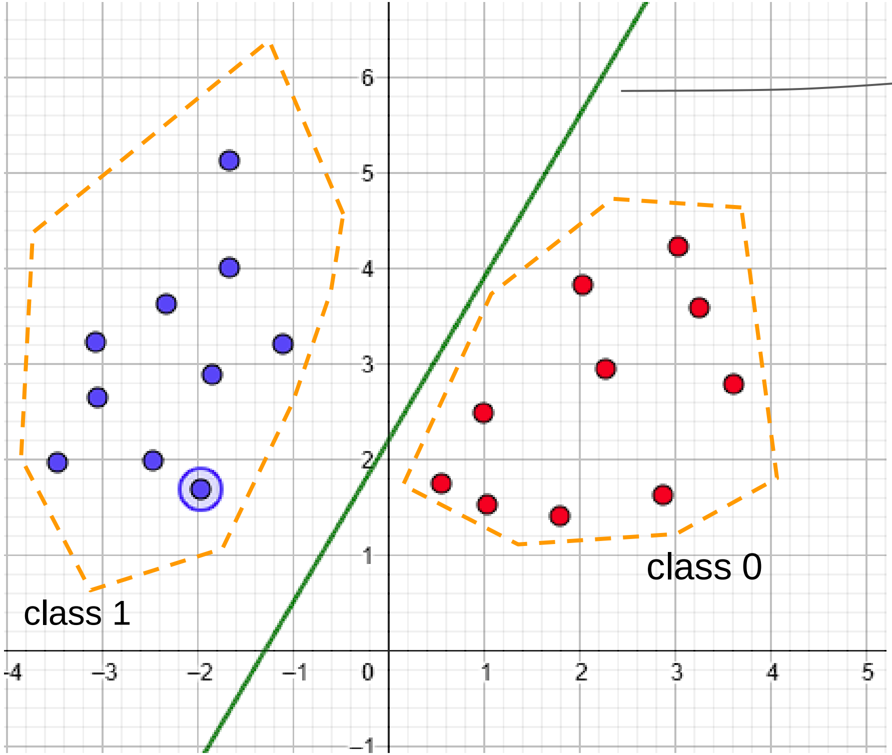

6
5
4
3
2
1
0
−1
−4
−3
−2
−1
1
2
3
4
class 1
class 0

Binary Classification



$$h(x_i) = w_0 + x_1 w_1$$

Binary Classification

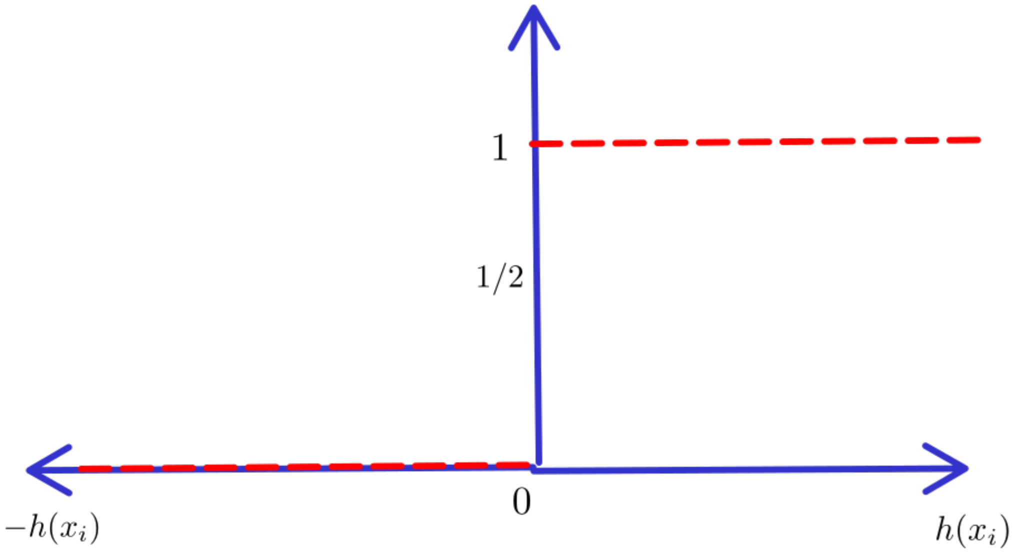


$$\boxed{h(x_i)} = w_0 + x_1 w_1$$

$$s(x_i) = \frac{1}{1 + e^{\boxed{-h(x_i)}}}$$

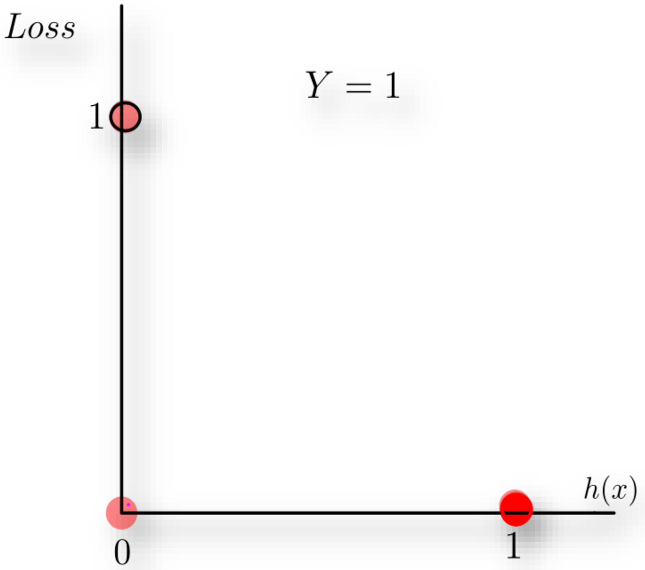
Logistic Function

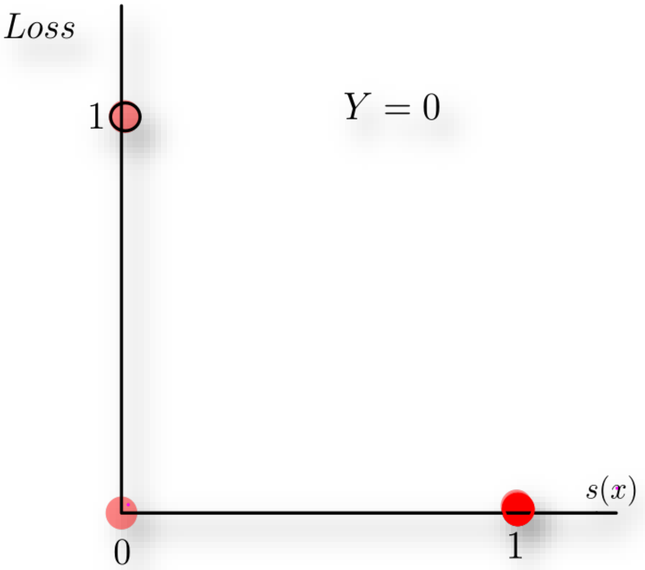
$$s(x_i) = \frac{1}{1 + e^{\boxed{-h(x_i)}}}$$

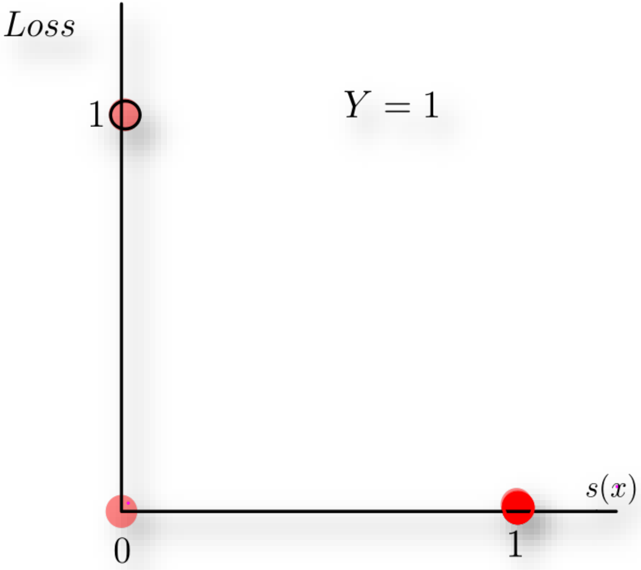


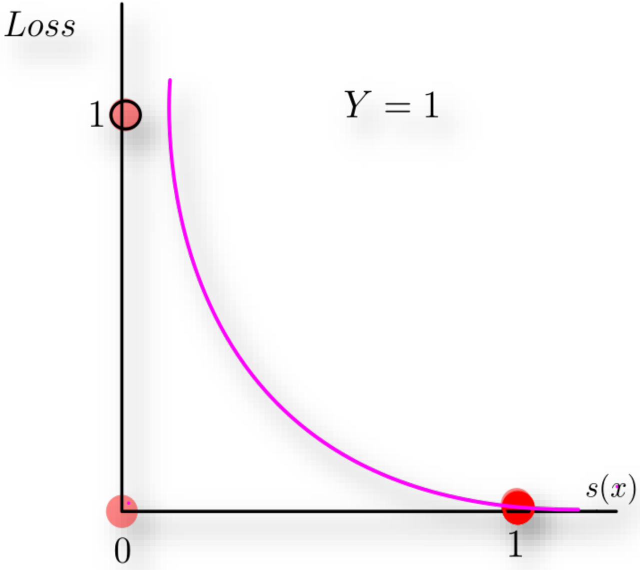
1. Hypothesis : $$s(x_i) = \frac{1}{1 + e^{-h(x_i)}}$$

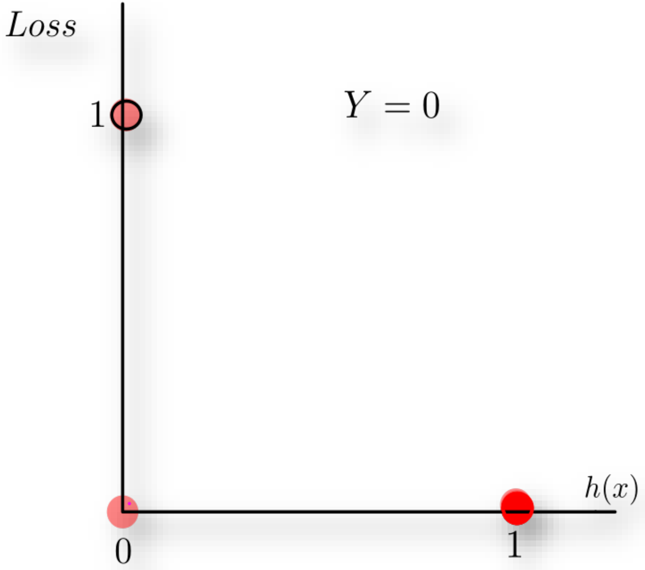
What would be the error function?

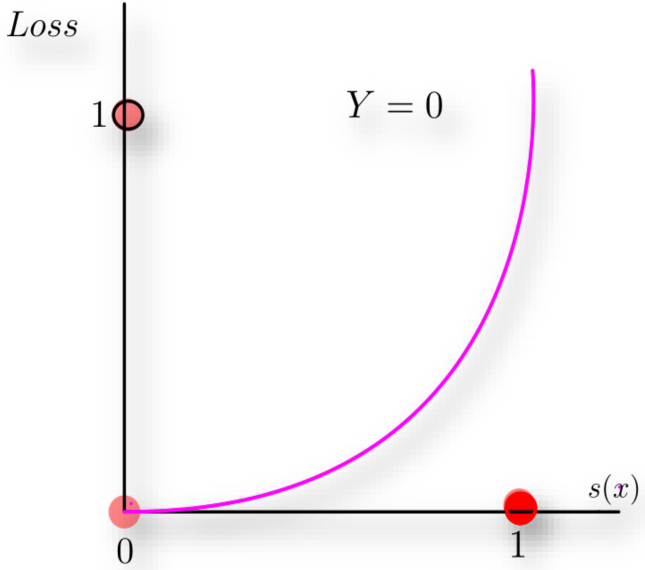

$Loss$
$Y = 1$
1
0
1
$h(x)$

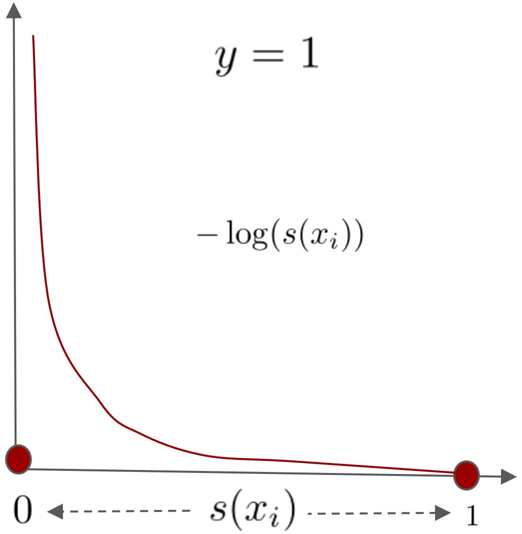

$Loss$
$Y = 0$
1
0
1
$s(x)$

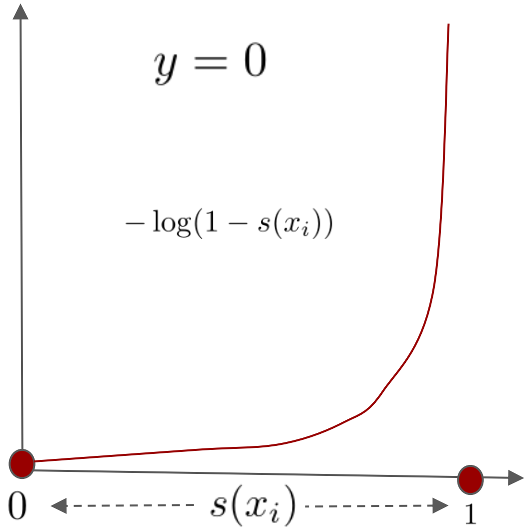

$Loss$
$Y = 1$
1
0
1
$s(x)$


$Loss$
$Y = 1$
1
0
1
$s(x)$


$Loss$
$Y = 0$
1
0
1
$h(x)$


$Loss$
$Y = 0$
1
0
1
$s(x)$





$$\mathcal{L} = -\sum_{i=1}^{n} (y_i \log(s(x_i)) + (1 - y_i) \log(1 - s(x_i)))$$

Si $y_i = 1 \Longrightarrow \log(s(x_i))$

Si $y_i = 0 \Longrightarrow \log(1 - s(x_i))$

2. Loss Function :

$$\mathcal{L} = -\sum_{i=1}^{n} (y_i \log(s(x_i)) + (1 - y_i) \log(1 - s(x_i)))$$

Hipótesis $$s(x_i) = \frac{1}{1 + e^{-h(x_i)}}$$

Loss $$\mathcal{L} = -\sum_{i=1}^{n}(y_i \log(s(x_i)) + (1 - y_i) \log(1 - s(x_i)))$$

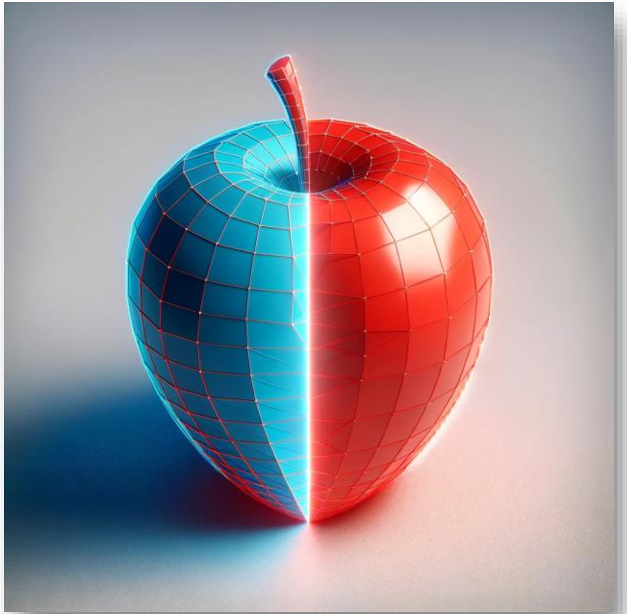
Derivadas $$\frac{\partial L}{w_j} = \frac{1}{n}\sum_{i=1}^{n}(y_i - s(x_i))(-x_{ij})$$

```python
def train(X, Y, epochs, alfa,lam):
    np.random.seed(2001)
    W = np.array([np.random.rand() for i in range(X.shape[1])])
    L = Error(X,W,Y,lam)
    loss = []
    for i in range(epochs):
        dW = derivada(X, W, Y,lam)
        W = update(W, dW, alfa)
        L = Error(X, W,Y,lam)
        loss.append(L)
        if i%10000==0:
            print(L)
    return W, loss
```

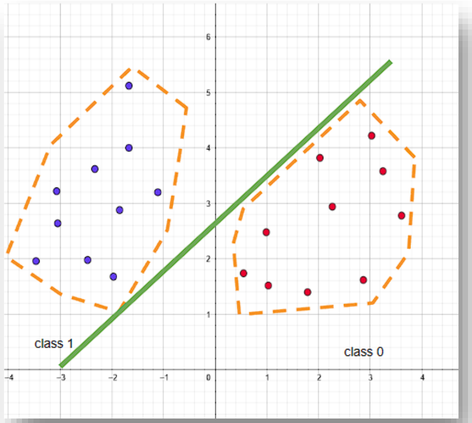
Teamwork Time

SVM

Support Vector Machines

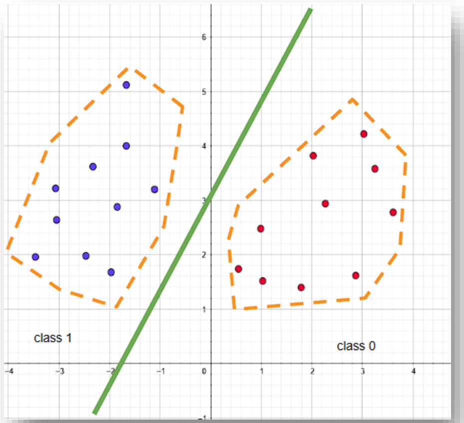


What is the best line that separates both groups?



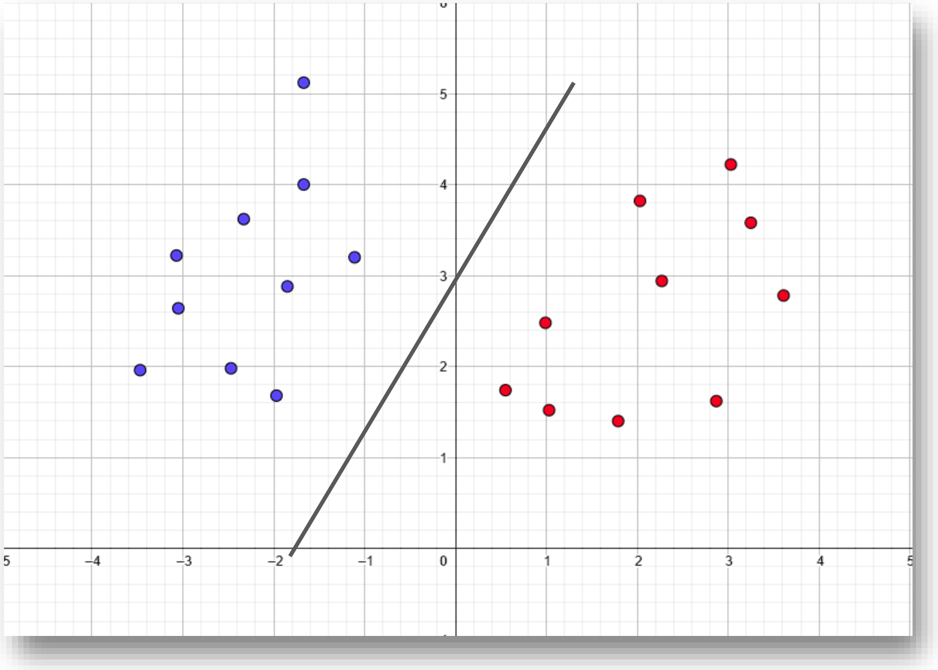
solution 1

solution 2

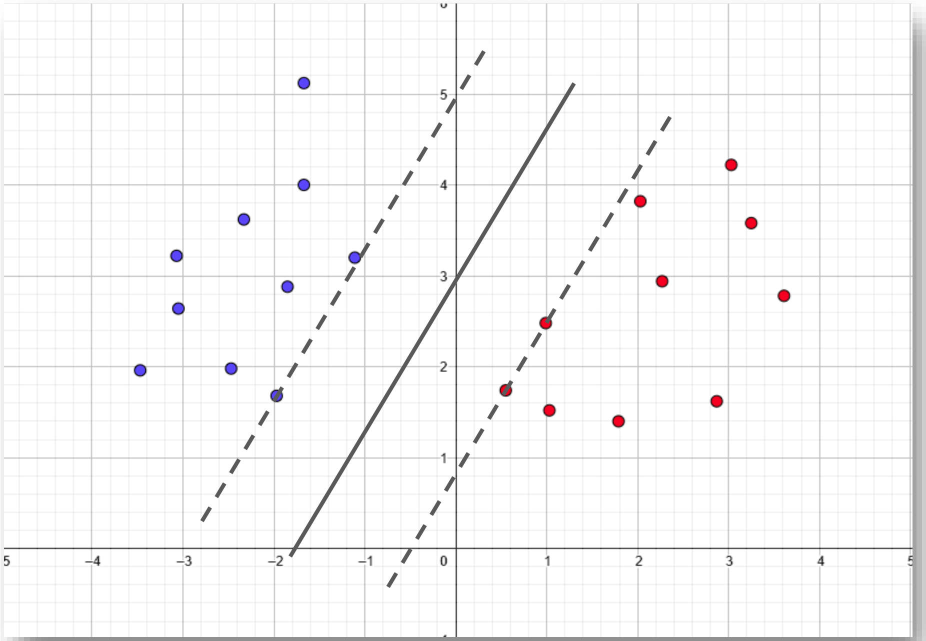


solution 3

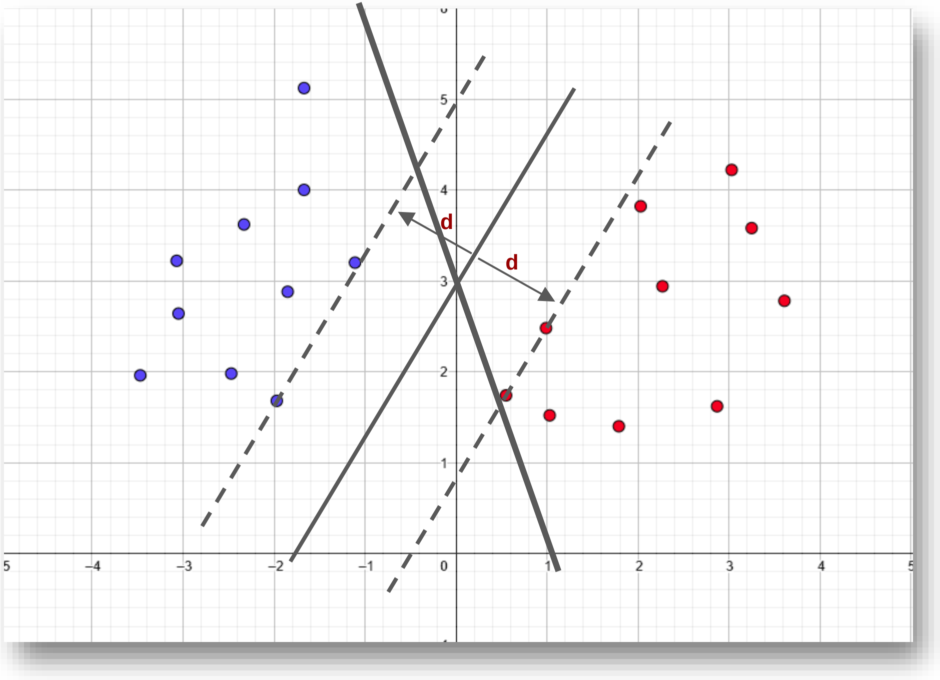
Try to find the best line that best separates both groups.



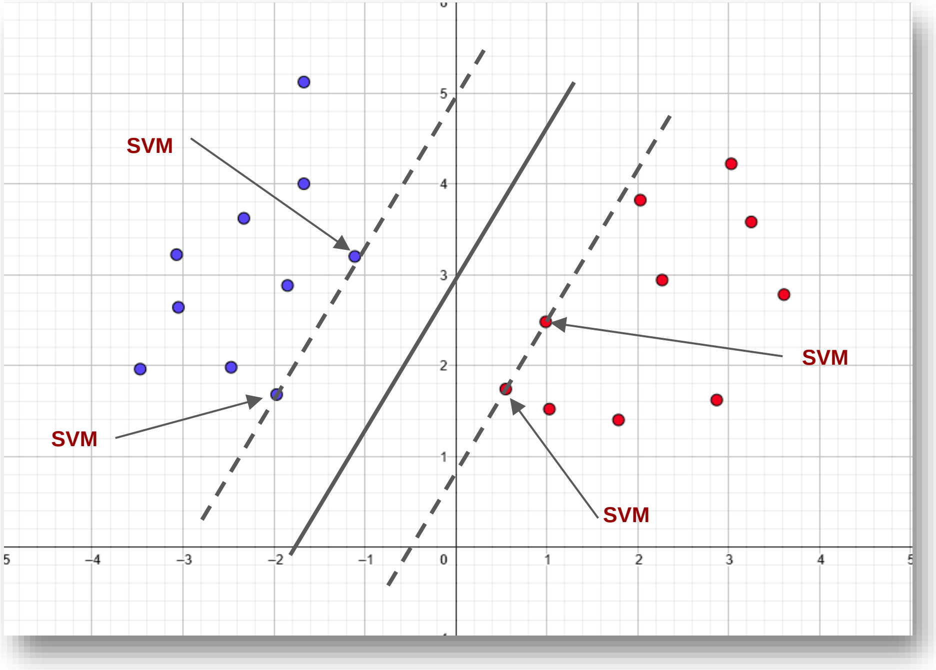
Try to find the best line that best separates both groups.



Maximize the distance d so that both classes are as separated as possible.

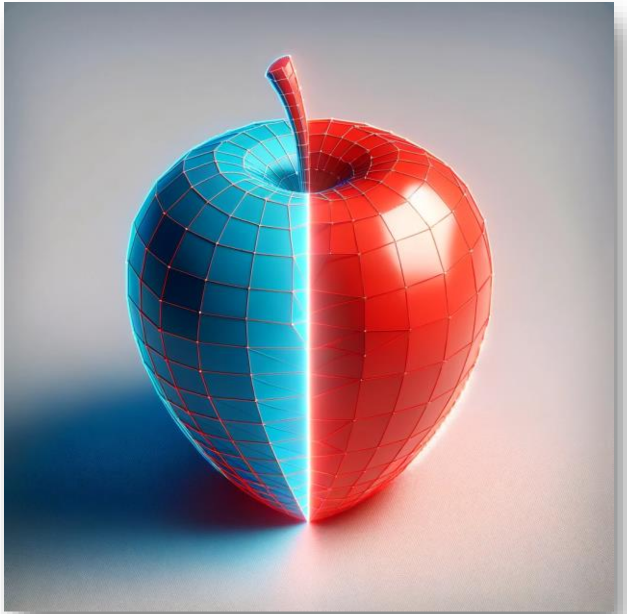


Who are the support vector machines?

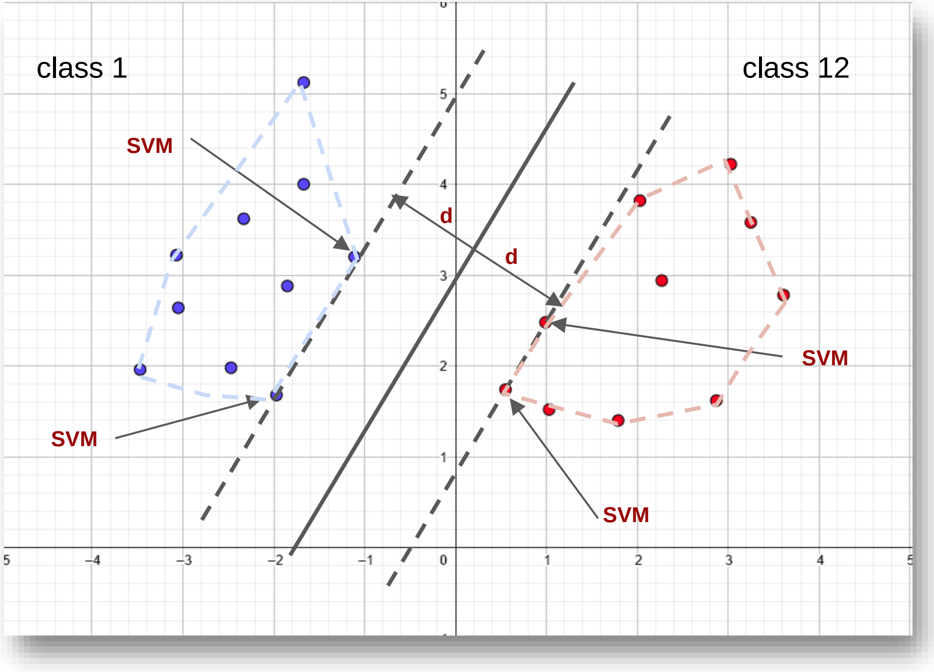


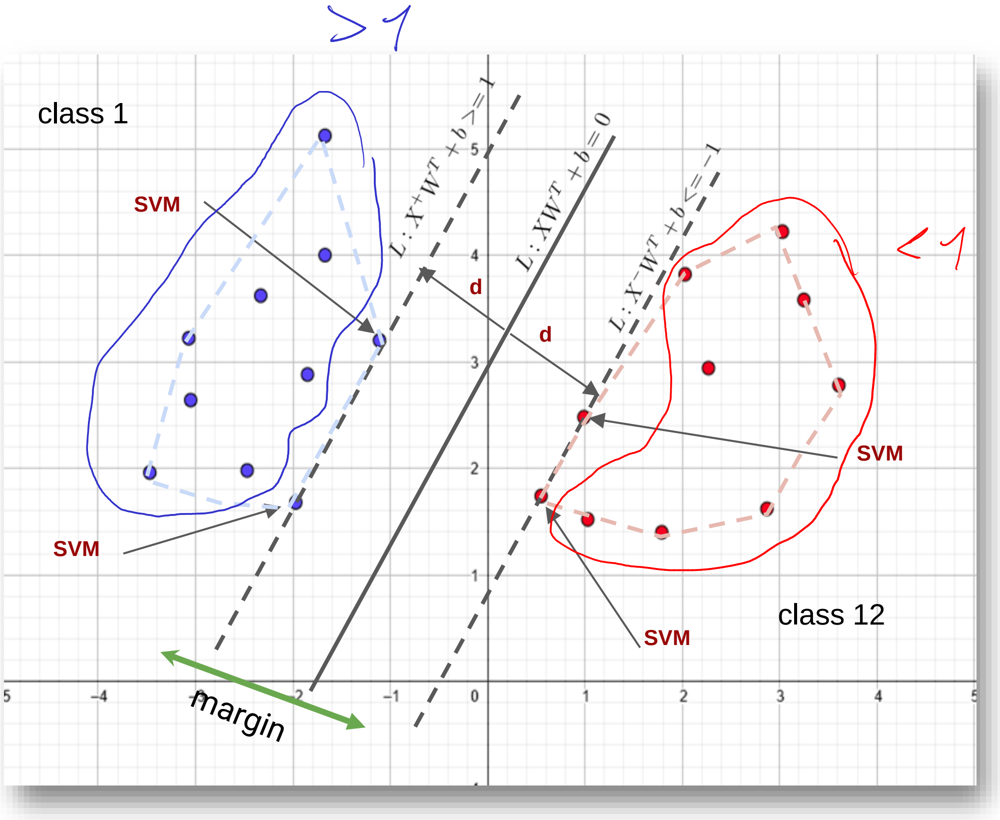
HARD SVM

Hard Support Vector Machines



The objective is to find the support vector machines through which the lines pass that are at a maximum distance d from the line that separates both groups.





Objective: Maximize 2d subject to 2 constraints.

maximize

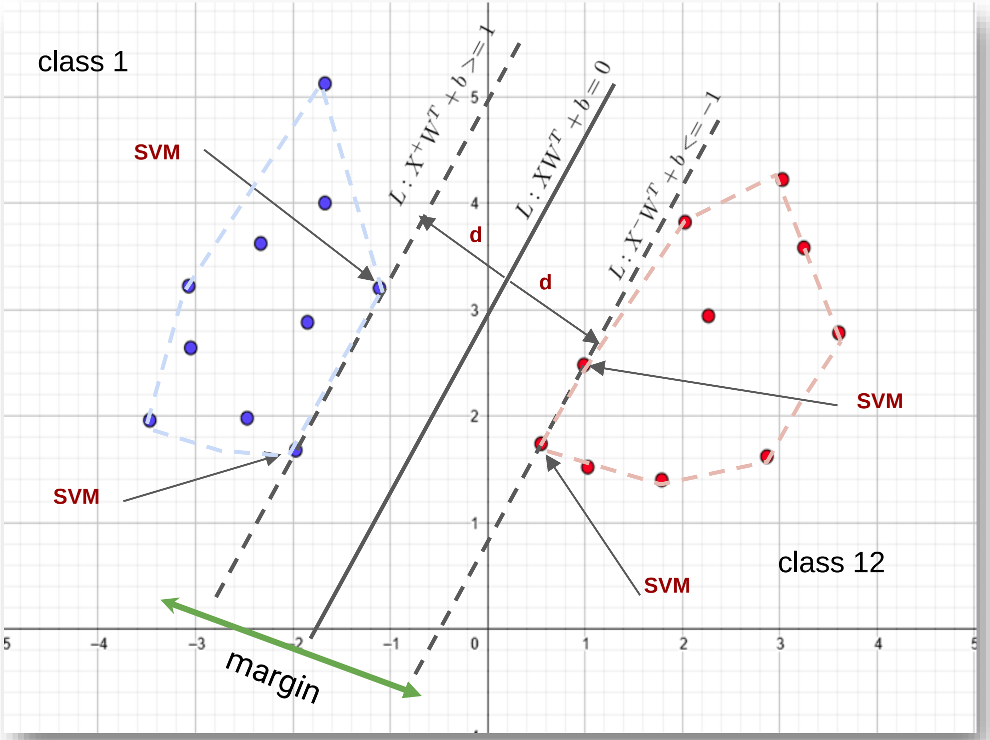
$max(2d)\ s.t$

$X^+W^T + b >= 1$ (+)

$X^-W^T + b <= -1$ (−

x^- : Labeled dataset -1

x^+ : Labeled dataset +1



Objective: Maximize 2d subject to 2 constraints.

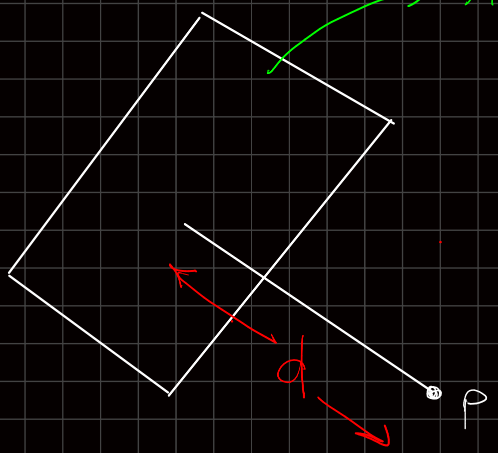
$$max(2\boxed{d})\ s.t$$

$$Y(X^{-}W^{T} + b) >= 1$$

Distance between a point and a plane

$x_1 w_1 + x_2 w_2 + b = 1$

Plane: $ax + bx + c = 0$



$$d = \frac{|x_1 w_1 + x_2 w_2 + b|}{\sqrt{w_1^2 + w_2^2}}$$

$$d = \frac{1}{\|w\|}$$

max $2d$ such that

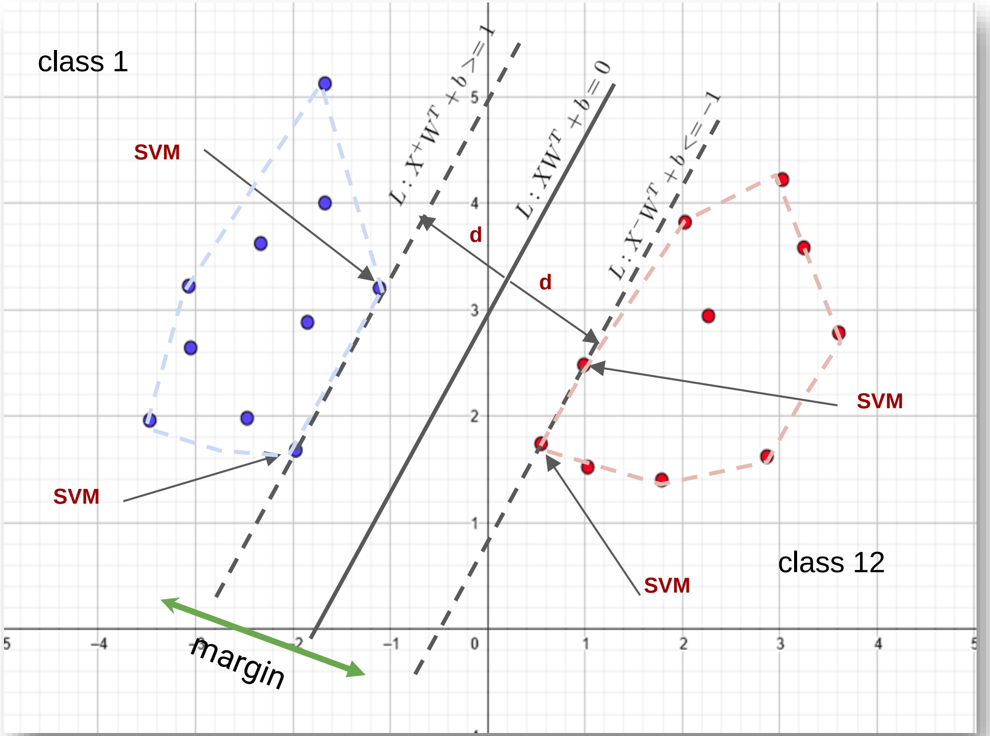
$x^+ w^t + b \geq 1$ $\quad y = 1$

$x^- w^t + b \leq -1 \rightarrow y = -1$ $\Rightarrow$

max $2d$ such that

$y(xw^t + b) \geq 1$

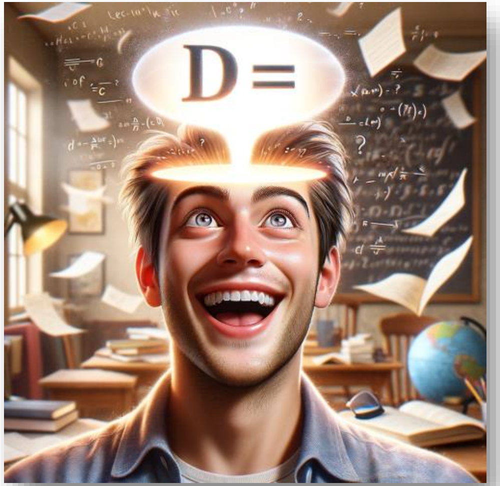
$$\boxed{\max \frac{2}{\|w\|} \quad \text{s.t} \quad y(xw^t + b) \geq 1}$$



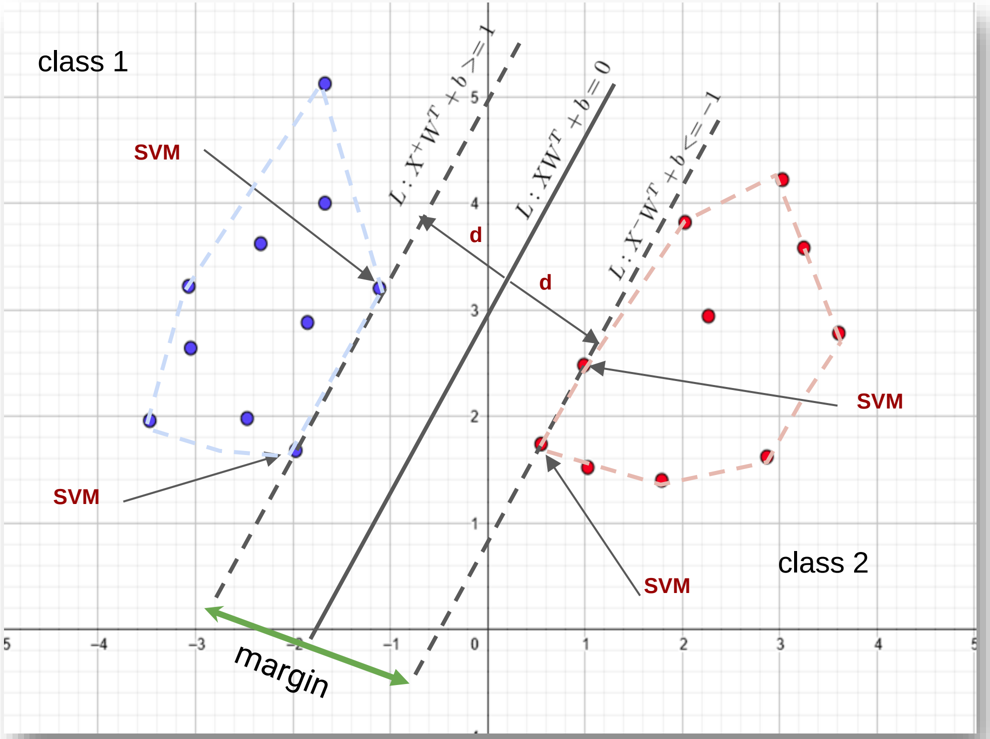
Objective: Maximize 2d subject to 2 constraints.

$$max(2\boxed{d})\ s.t$$

$$Y(X^{-}W^{T} + b) >= 1$$

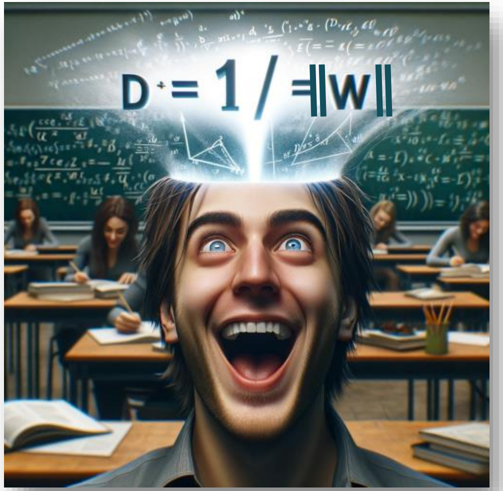


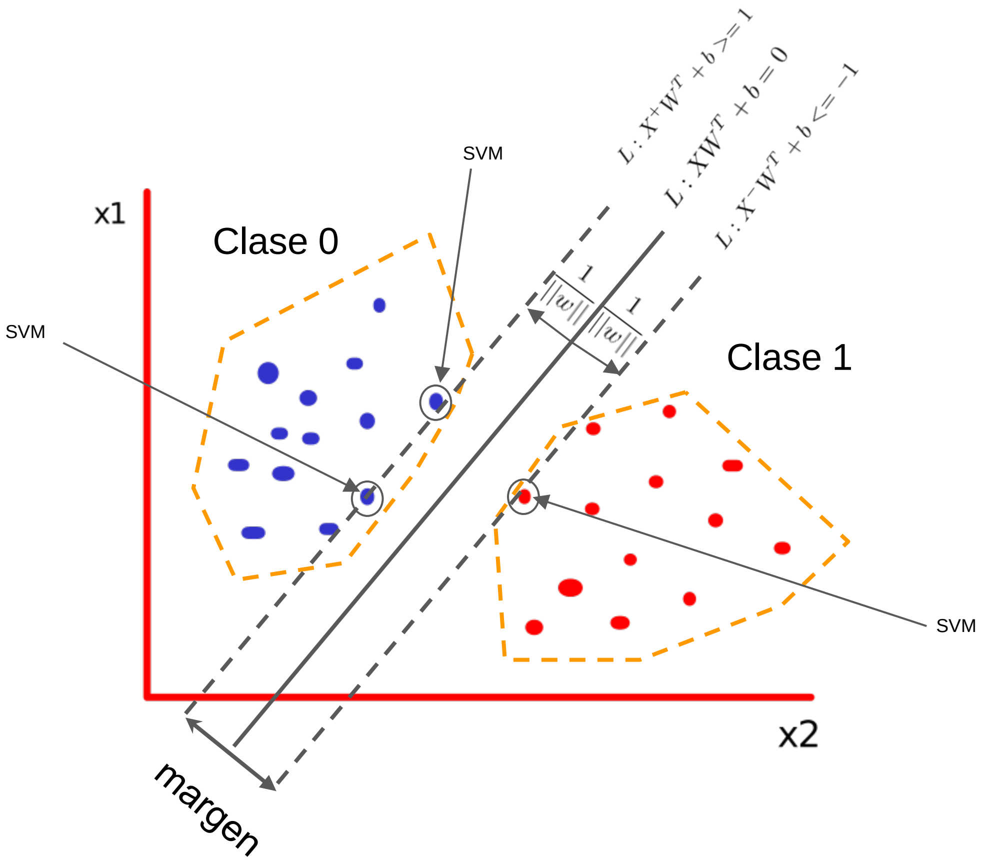
Objective: Maximize 2d subject to 2 constraints.



$$max(2\boxed{d})\ s.t$$

$$Y(X^{-}W^{T} + b) >= 1$$



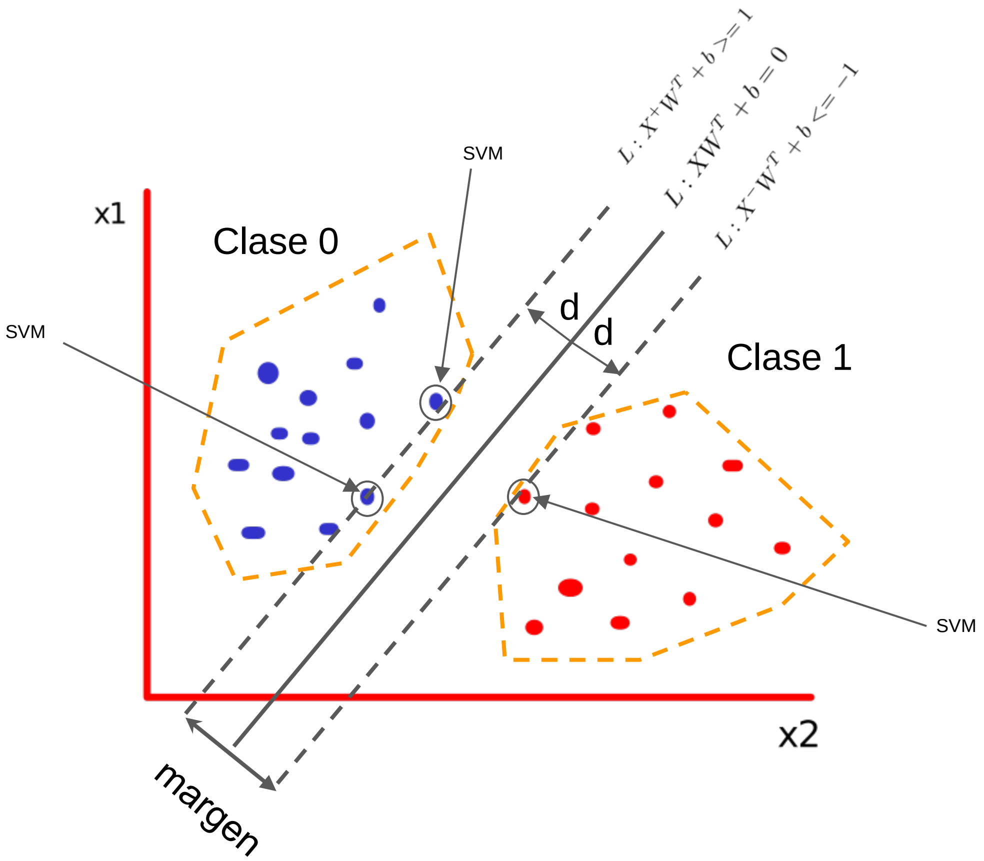


$$max(2d)\ s.t$$

$$Y(X^{-}W^{T} + b) >= 1$$

$$d = \frac{1}{||w||}$$

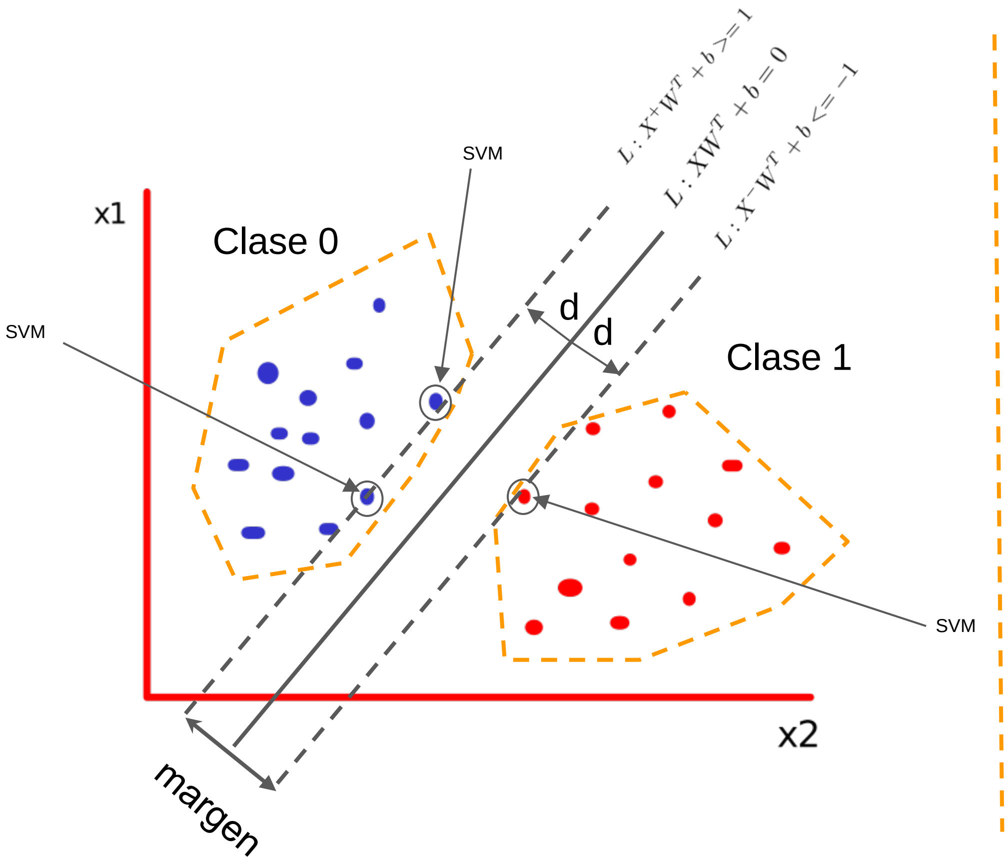
¿Demuestre que d = $\frac{1}{||w||}$?



$$max \frac{2}{||w||} \quad s.t$$

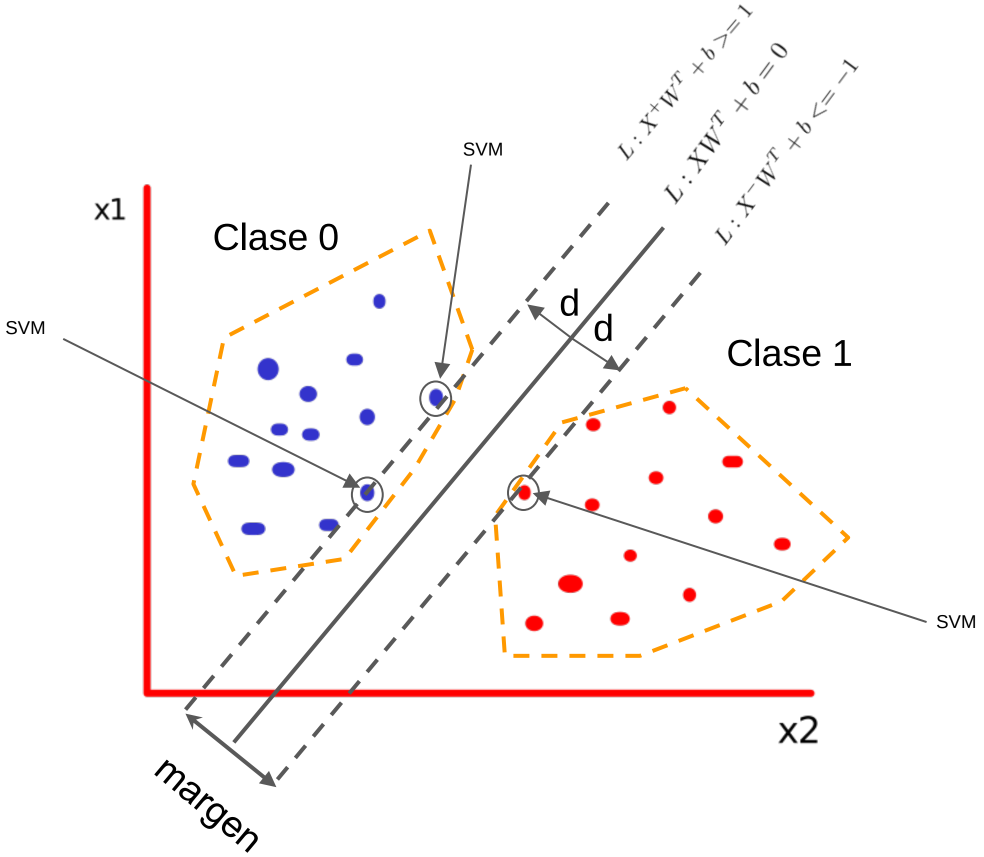
$$Y(X^{-}W^{T} + b) >= 1$$

¿Queremos maximizar?



$$min \frac{||w||}{2} \quad s.t$$

$$Y(X^{-}W^{T} + b) >= 1$$



$$min \frac{||w||}{2} \quad s.t$$

$$Y(X^{-}W^{T} + b) >= 1$$

$$min \frac{||w||^{2}}{2} \quad s.t$$

$$Y(X^{-}W^{T} + b) >= 1$$

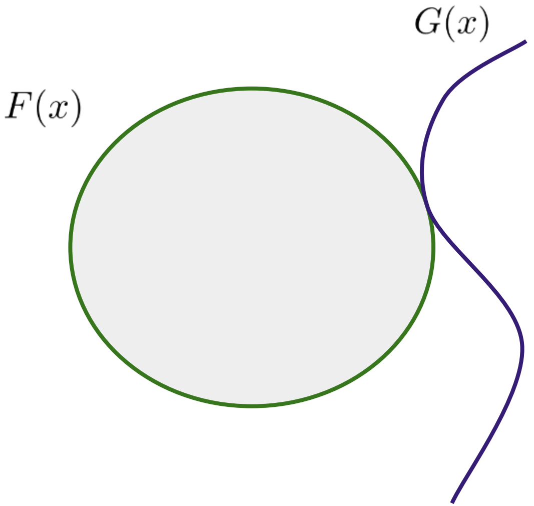
How do we solve this equation?

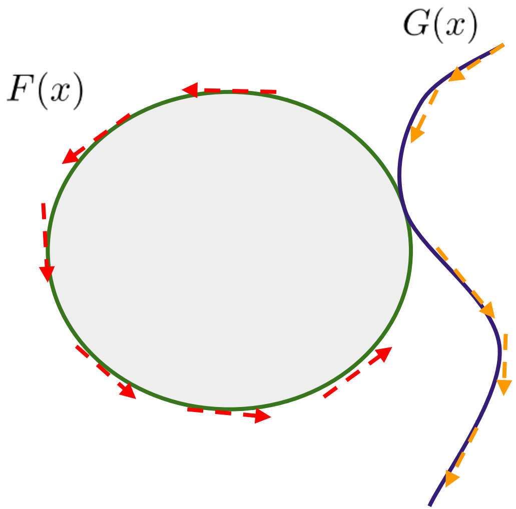
$$min \frac{||w||^2}{2} \quad s.t \;\; y_i(x_i w^t + b) >= 1 \quad \forall i; \;\; 1 <= i <= n$$

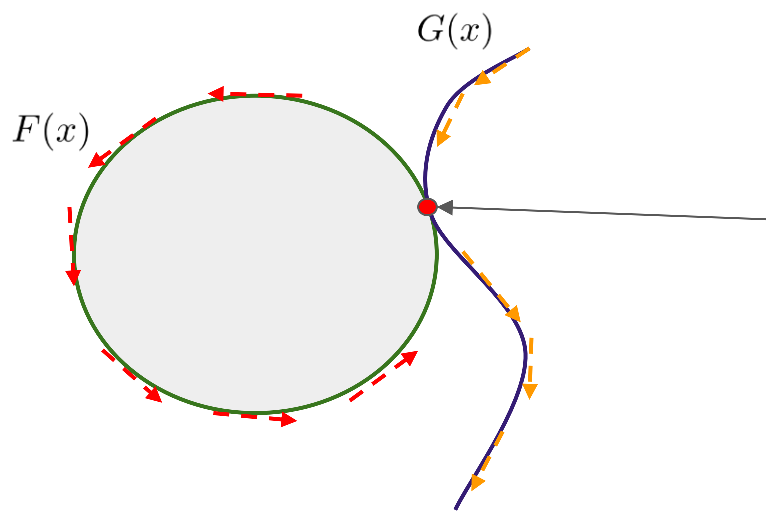
How do we solve this equation?

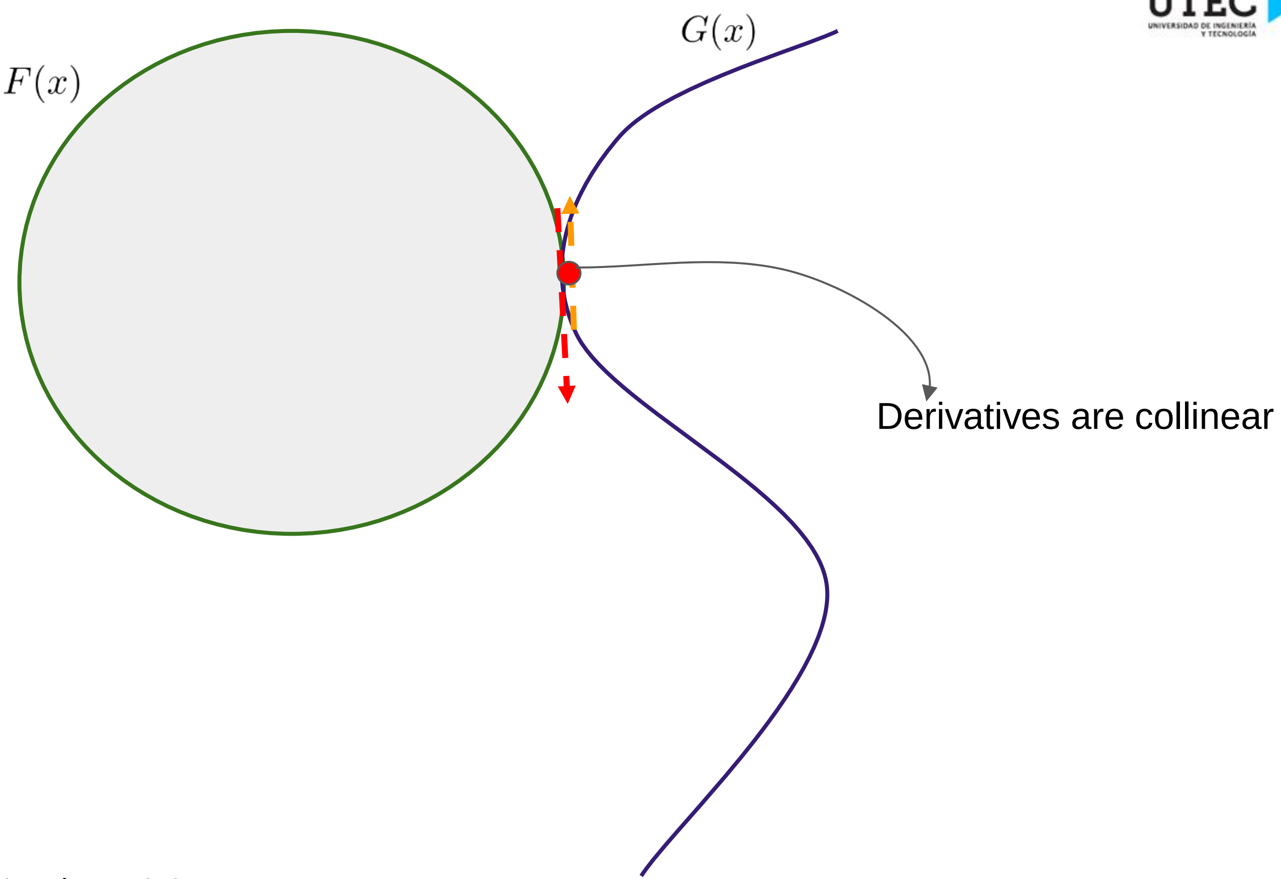
LAGRANGE

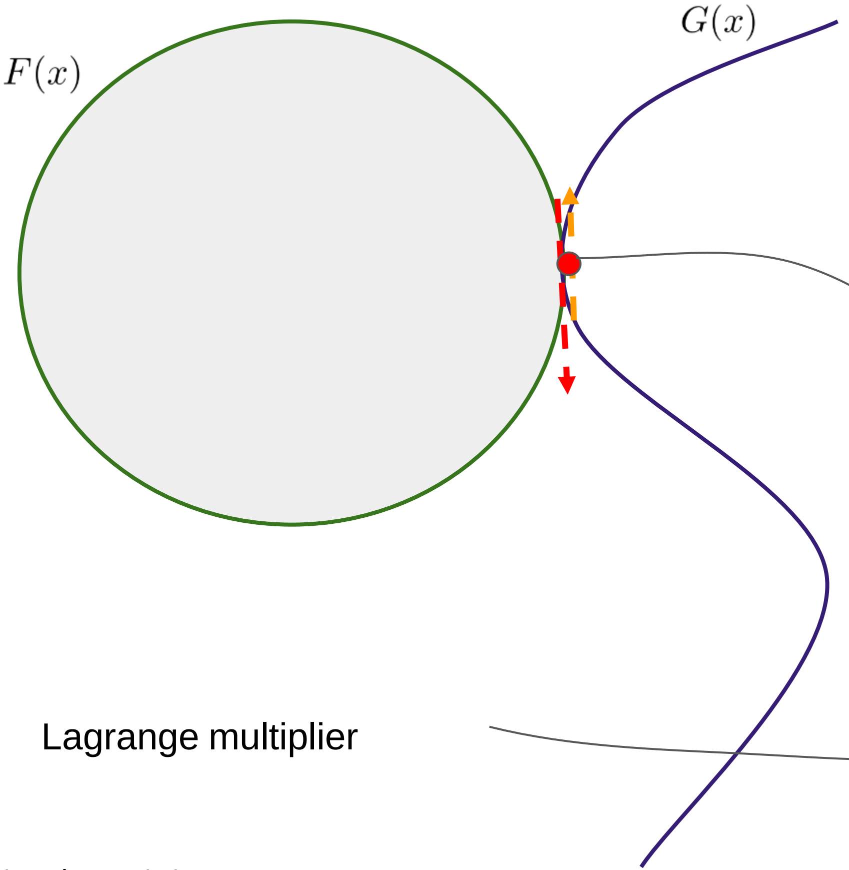
Joseph-Louis Lagrange was a prominent 18th-century mathematician and astronomer, born in Italy and later naturalized French. He is known for his revolutionary contributions to analytical mechanics, number theory, and mathematical analysis. His work, especially "Mécanique Analytique", is fundamental to modern physics and mathematics, influencing the development of theoretical physics and engineering. Lagrange points, named after him, highlight his influence on optimization and polynomial equations.


$G(x)$
$F(x)$


$F(x)$
$G(x)$


$G(x)$
$F(x)$

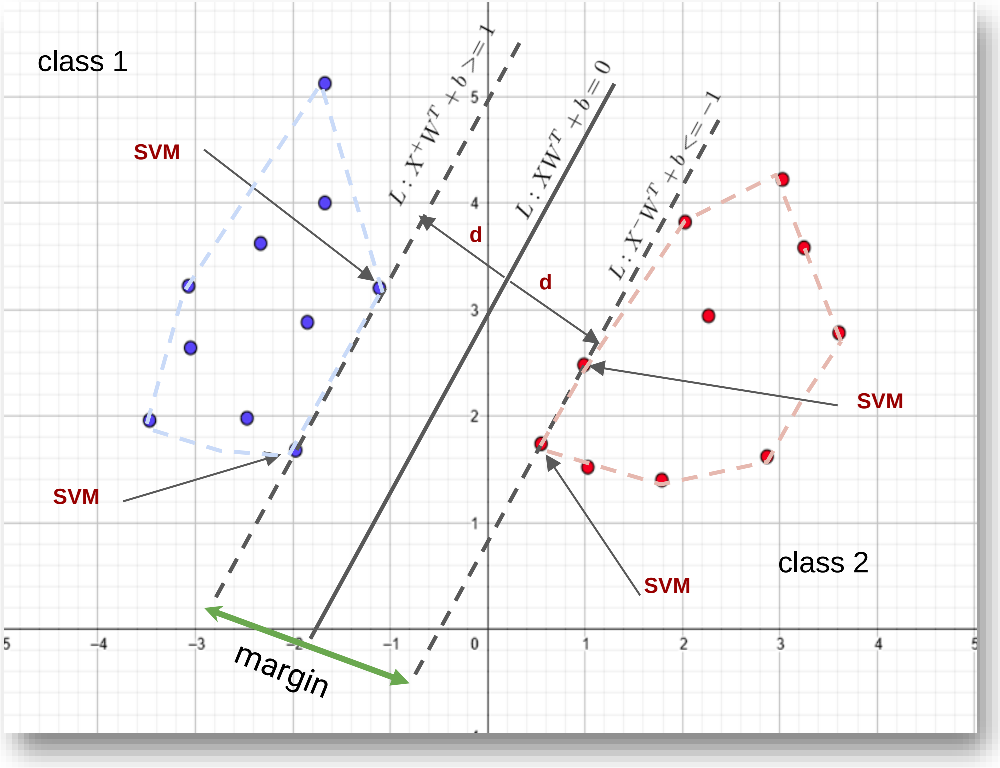

$F(x)$
$G(x)$
Derivatives are collinear



Derivatives are collinear

$$\frac{\partial F(x)}{\partial x} = \lambda \frac{\partial G(x)}{\partial x}$$

Lagrange multiplier



Objective: Maximize 2d subject to this constraints.

$$min\frac{||w||^2}{2} \quad s.t \;\; y_i(x_iw^t + b) >= 1 \quad \forall i; \;\; 1 <= i <= n$$

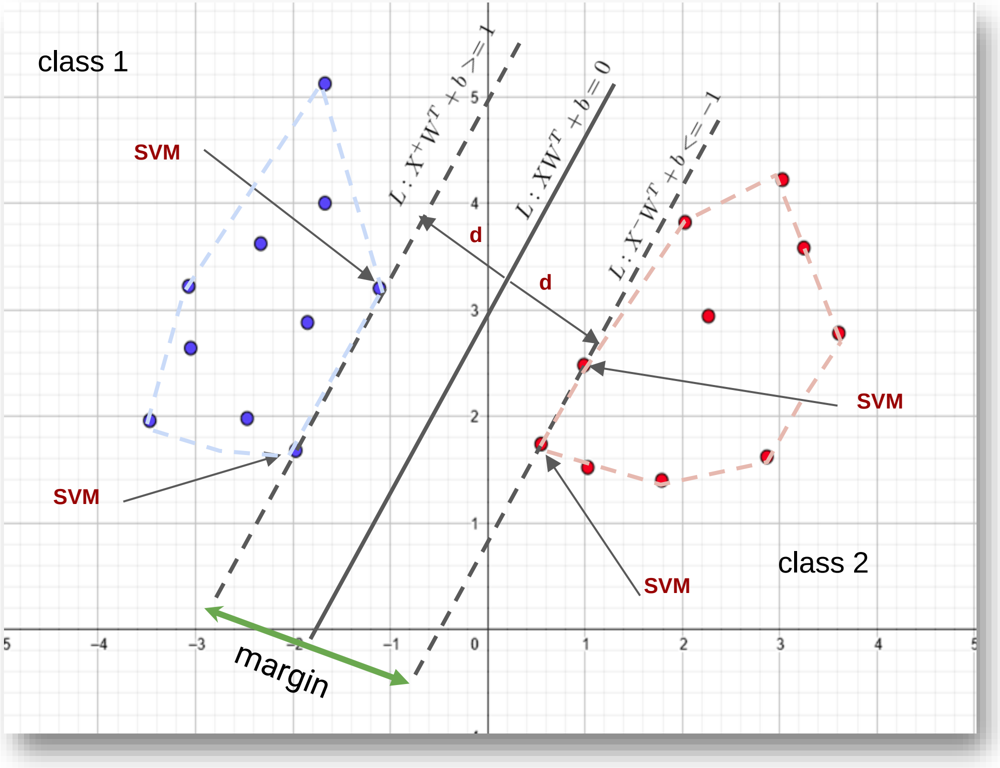
Lagrangian

$$\mathcal{L}(w, b, \lambda) = \frac{||w||^2}{2} - \sum_{i=0}^{n} \lambda_i(y_i(w^tx_i + b) - 1))$$

Solve the Lagrangian.

$$\frac{\partial\mathcal{L}(w, b, \lambda)}{\partial w} = \frac{||w||^2}{2} - \sum_{i=0}^{n} \lambda_i(y_i(w^tx_i + b) - 1))$$

$$\frac{\partial\mathcal{L}(w, b, \lambda)}{\partial b} = \frac{||w||^2}{2} - \sum_{i=0}^{n} \lambda_i(y_i(w^tx_i + b) - 1))$$



Objective: Maximize 2d subject to this constraints.

$$min\frac{||w||^2}{2} \quad s.t \;\; y_i(x_i w^t + b) >= 1 \quad \forall i; \;\; 1 <= i <= n$$

Lagrangian

$$\mathcal{L}(w,b,\lambda) = \frac{||w||^2}{2} - \sum_{i=0}^{n} \lambda_i(y_i(w^t x_i + b) - 1))$$

Solve the Lagrangian.

$$\frac{\partial \mathcal{L}(w,b,\lambda)}{\partial w} = \frac{||w||^2}{2} - \sum_{i=0}^{n} \lambda_i(y_i(w^t x_i + b) - 1))$$

$$\frac{\partial \mathcal{L}(w,b,\lambda)}{\partial b} = \frac{||w||^2}{2} - \sum_{i=0}^{n} \lambda_i(y_i(w^t x_i + b) - 1))$$

Objective: Maximize 2d subject to this constraints.

$$min\frac{||w||^2}{2} \quad s.t \;\; y_i(x_i w^t + b) >= 1 \quad \forall i; \quad 1 <= i <= n$$

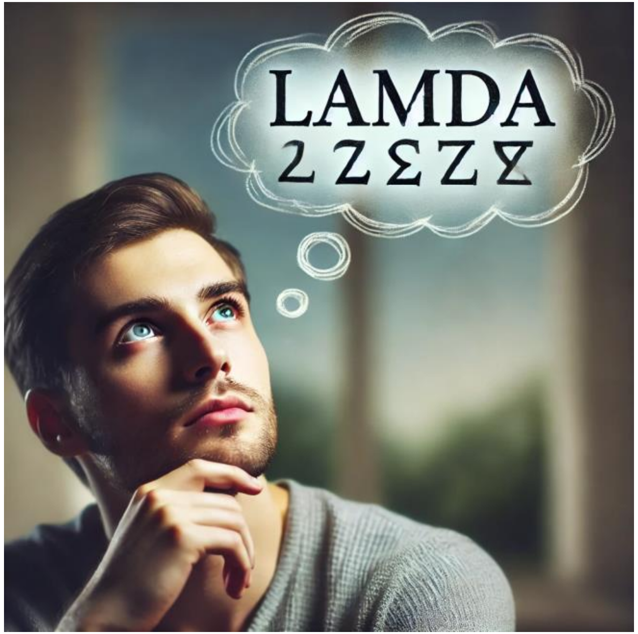
Lagrangian

$$\mathcal{L}(w,b,\lambda) = \frac{||w||^2}{2} - \sum_{i=0}^{n} \lambda_i (y_i(w^t x_i + b) - 1))$$

Solve the Lagrangian.

$$\frac{\partial \mathcal{L}(w,b,\lambda)}{\partial w} = \mathbf{w} - \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i = 0 \;\Rightarrow\; \mathbf{w} = \sum_{i=1}^{n} \boxed{\lambda_i} y_i \mathbf{x}_i$$

$$\frac{\partial \mathcal{L}(w,b,\lambda)}{\partial b} = -\sum_{i=1}^{n} \lambda_i y_i = 0 \;\Rightarrow\; \sum_{i=1}^{n} \boxed{\lambda_i} y_i = 0$$

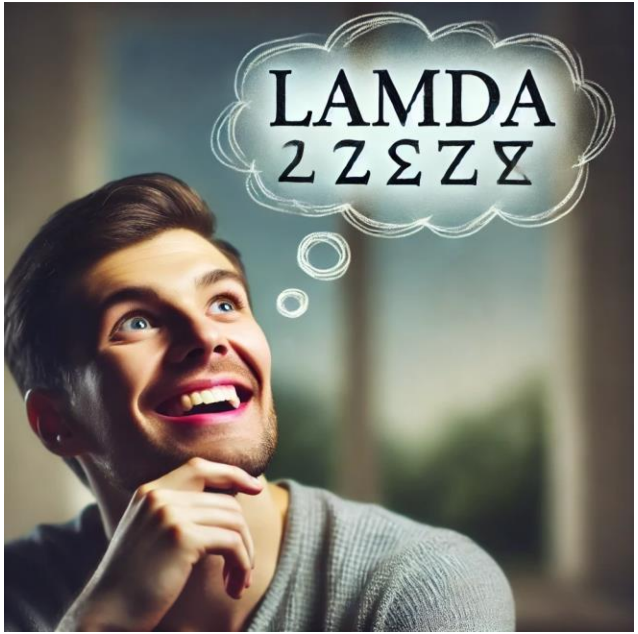


Formulating the Dual Problem

$$\boxed{\mathbf{w}} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i \quad \Rightarrow \quad L(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2} \|\boxed{\mathbf{w}}\|^2 - \sum_{i=1}^{n} \lambda_i \left[y_i (\boxed{\mathbf{w}}^\top \mathbf{x}_i + b) - 1 \right]$$

replace

$$L(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i^\top \mathbf{x}_j)$$



The dual problem is then:

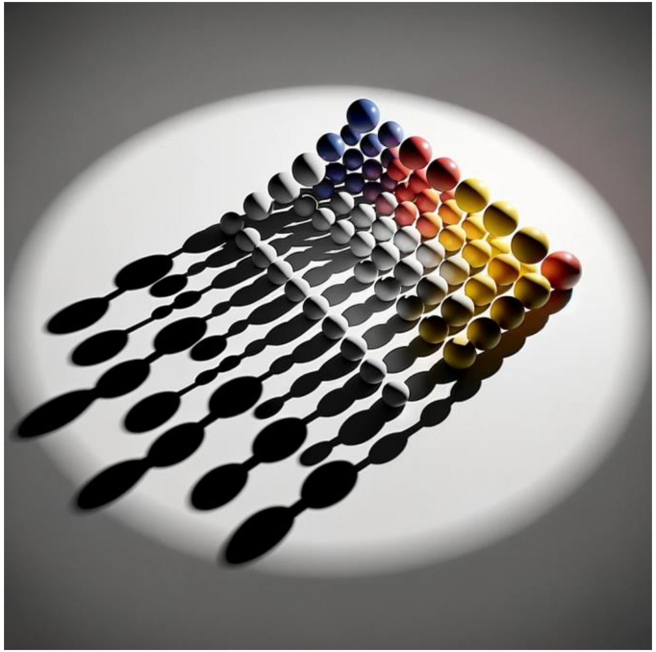
Maximize $$L(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \boxed{(\mathbf{x}_i^\top \mathbf{x}_j)}$$ with respect to **λ**

Subject to $$\sum_{i=1}^{n} \lambda_i y_i = 0, \quad \lambda_i \geq 0, \quad \forall i$$

The dual problem is a **convex quadratic optimization problem** that can be solved using standard quadratic programming (QP) algorithms. **Solving it** yields the optimal **values of λi.**

SVM KERNELS

If you can’t see what’s happening in your dimension, go to a higher dimension to see reality.



Optimal Parameters in SVM

Optimal Model Parameters

$$w^* = \sum_{i=0}^{n} \alpha_i y_i x_i$$

$$b^* = \frac{1}{svm} \sum_{i=0}^{svm} (y_k - x_k w*)$$

Increasing Dimension: Kernels

$$w* = \sum_{i=0}^{svm} \alpha_i y_i \boxed{\phi(x_i)})$$

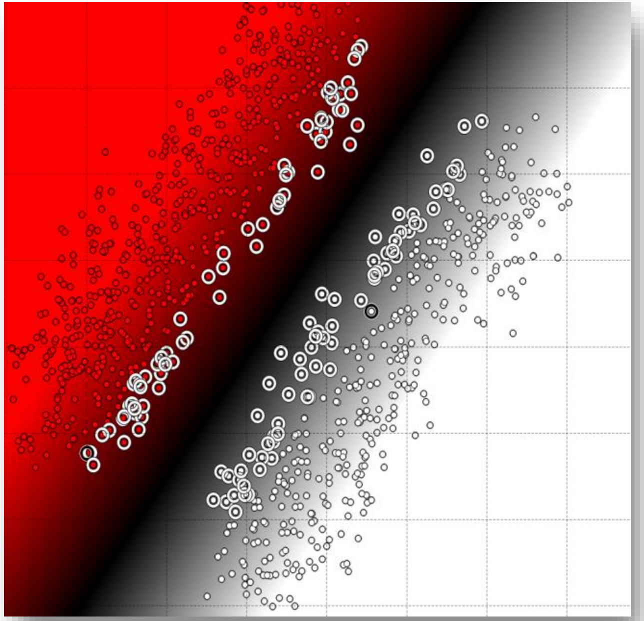
This function maps x_i to a higher-dimensional space

$$b^* = \frac{1}{svm} \sum_{k=0}^{svm} (y_k - \sum_{i=0}^{n} \alpha_i y_i \boxed{K(x_i, x_k)})$$

Kernel: When the dataset is not linearly separable

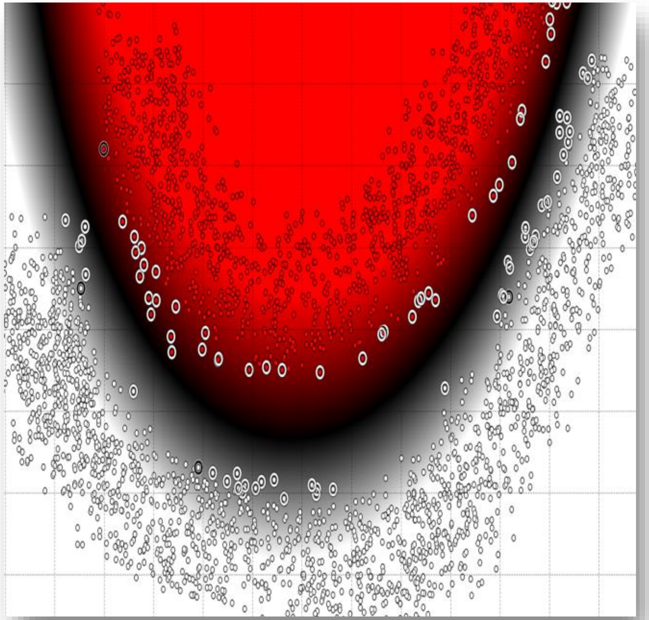
Linear kernel

$$K(x_i, x_k) =< \phi(x_i), \phi(x_k) >$$



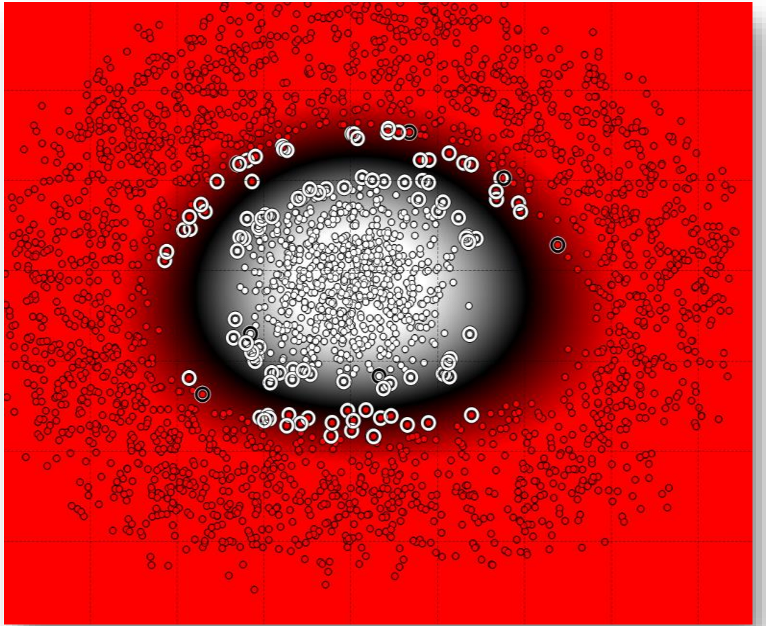
Polynomial kernel

$$K(x_i, x_k) = (< \gamma\phi(x_i), \phi(x_k) > +r)^d$$



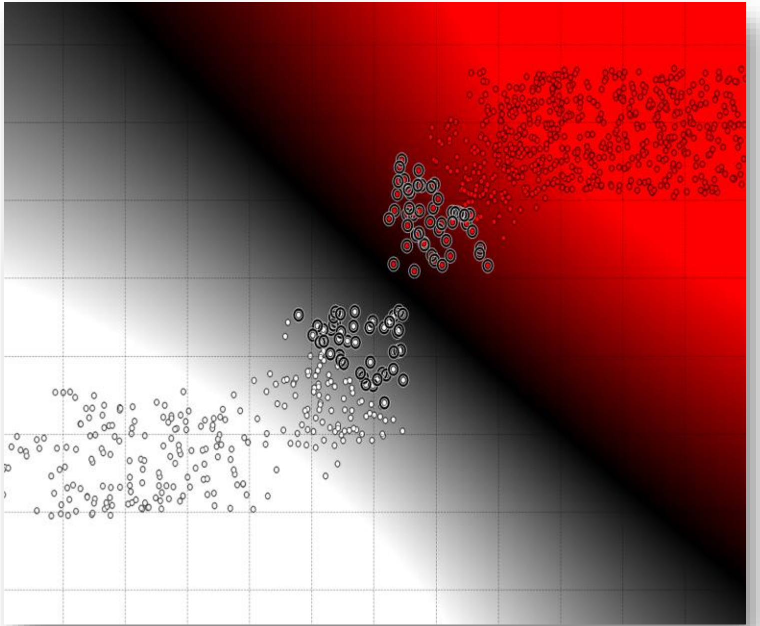
Radial Basis Function (RBF)

$$K(x_i, x_k) = exp(-\gamma \|\phi(x_i) - \phi(x_k)\|^2)$$



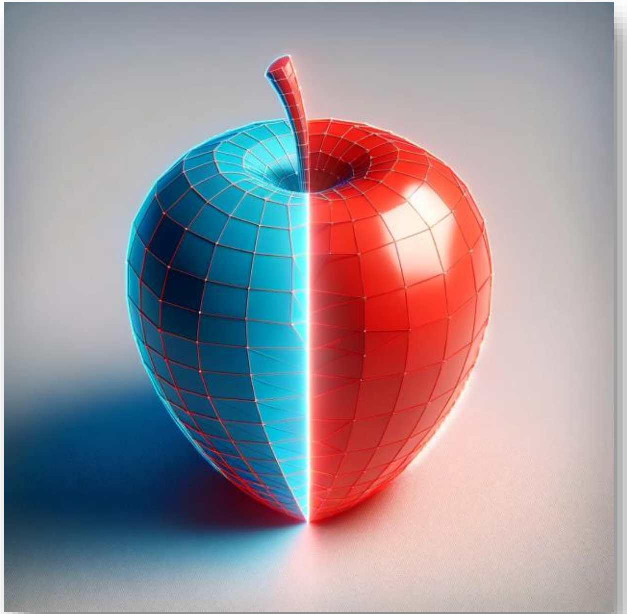
Sigmoid Kernel

$$K(x_i, x_k) = \tanh(\gamma < x_i, x_k > +r)$$



SOFT SVM

SOFT Support Vector Machines



- Hipótesis

$$(x_i * w^t + b))$$

- Loss

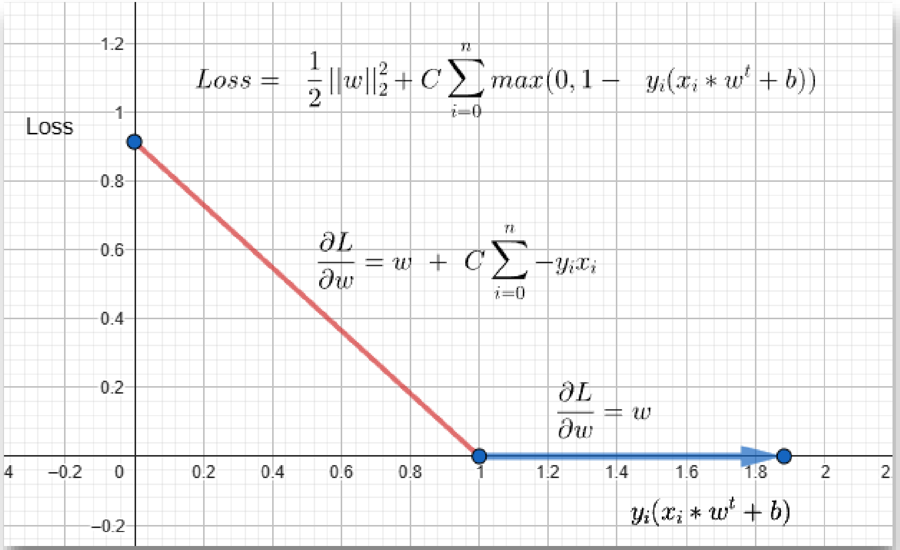
$$\frac{1}{2}||w||_2^2 + C\sum_{i=0}^{n} max(0, 1 - y_i(x_i * w^t + b))$$

- Derivadas

$$Si\ y_i(x_i w^t + b) < 1$$
$$\frac{\partial L}{\partial w} = w + C\sum_{i=0}^{n} -y_i x_i$$
$$else$$
$$\frac{\partial L}{\partial w} = w$$

- Update

$$Si\ y_i(x_i w^t + b) < 1$$
$$w = w - \alpha(w + C\sum_{i=0}^{n} -y_i x_i)$$
$$else$$
$$w = w - \alpha * w$$




UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA
INGENIERÍA
MECATRÓNICA
BIOINGENIERÍA
CIENCIA DE
LA COMPUTACIÓN
AMBIENTAL