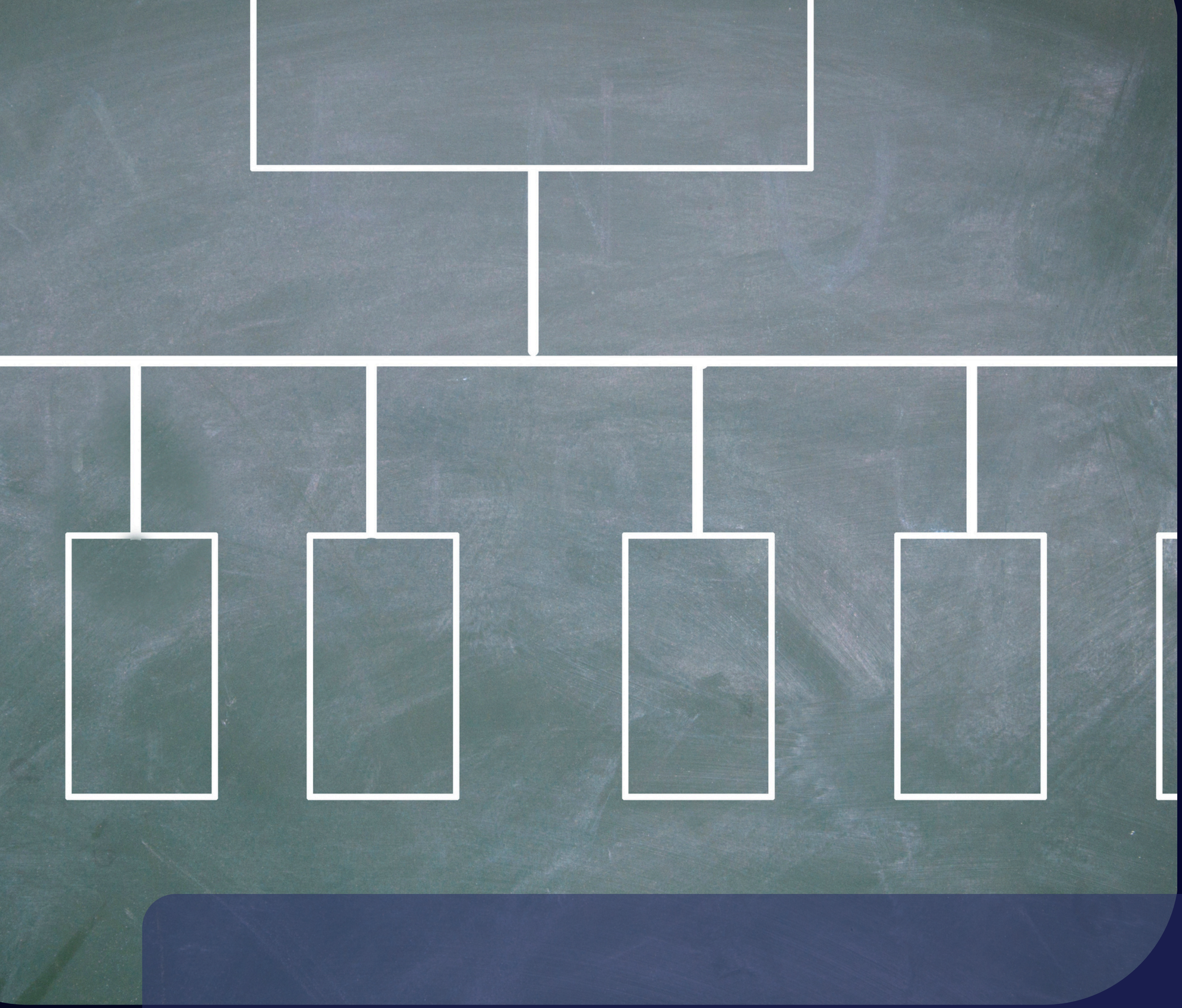
Inteligencia Artificial
Primer Semestre 2025

SVM

Máquinas de Soporte Vectorial



Contenido

En el vasto mundo del aprendizaje automático, donde los modelos se enfrentan al desafío de clasificar datos complejos o realizar predicciones precisas, las Máquinas de Vectores de Soporte (SVM) se erigen como una herramienta elegante y poderosa. Este algoritmo no solo busca trazar la línea que mejor separa los datos, sino que lo hace maximizando el margen entre las clases, garantizando un enfoque robusto y eficiente.

1. **Introducción**

   ¿Por qué necesitamos SVM? Comparación con regresión logística.

2. **Fundamentos**

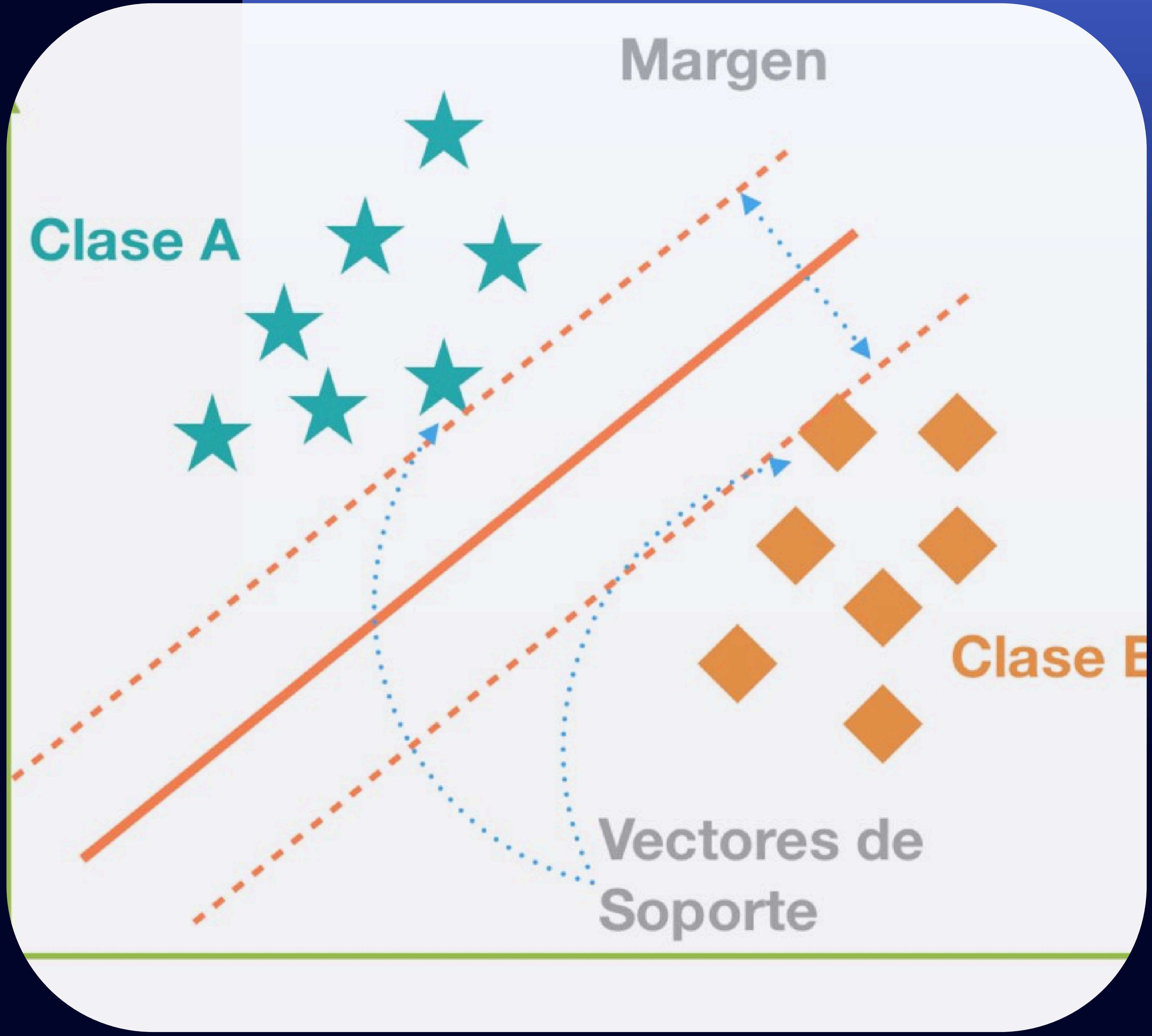
   Concepto de hiperplano de separación.
   Márgenes y soporte vectorial.
   Función de costo y optimización en SVM.

3. **Clasificación para modelos linealmente no separables**

   Kernel Trick y Transformación de Espacios.

SVM

Un Support Vector Machine (SVM) es un modelo de aprendizaje supervisado que se utiliza para resolver problemas de clasificación y regresión. En particular, en clasificación, SVM busca una hiperplano que separe las clases de manera óptima.



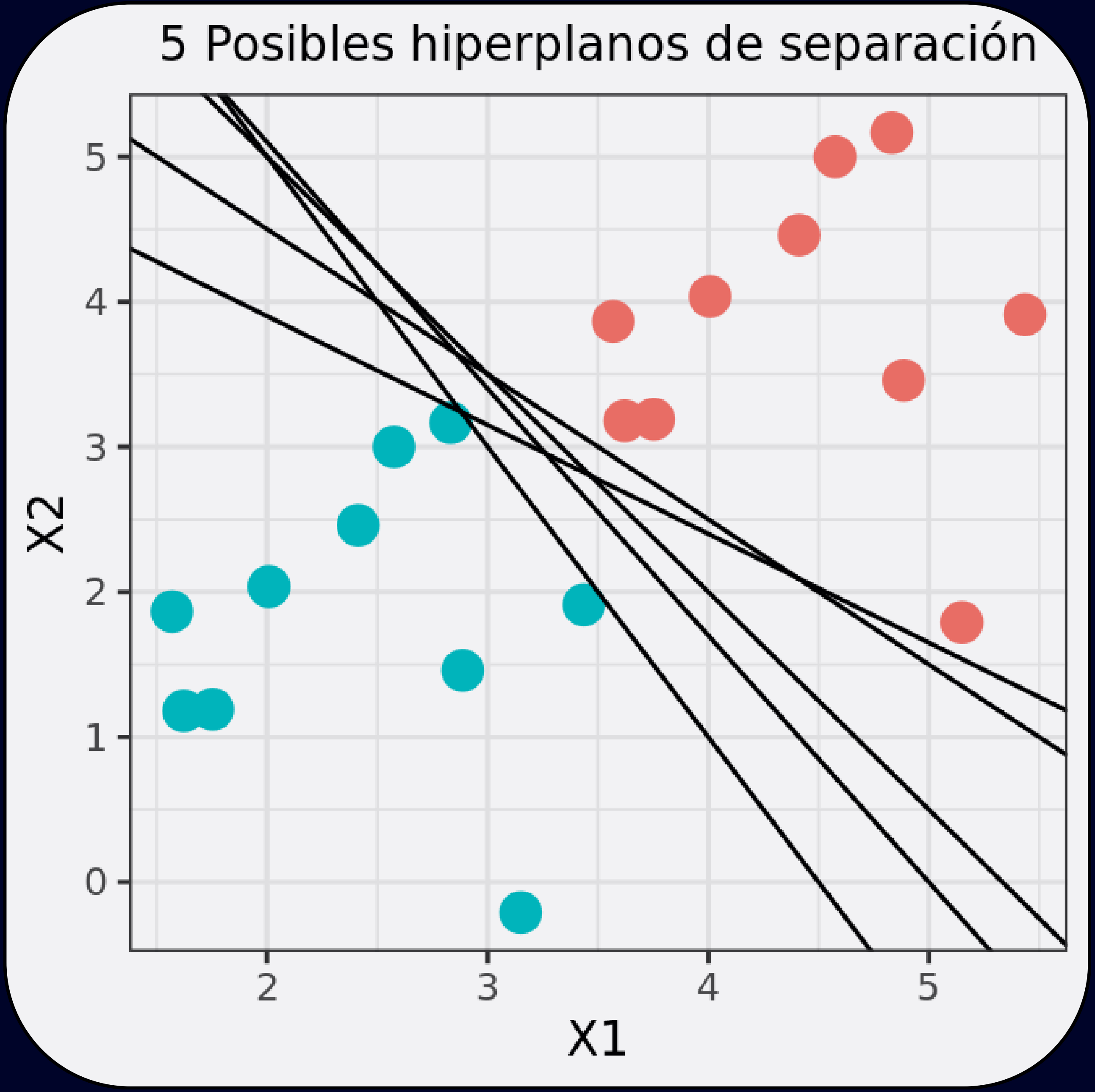
1. **Eficiencia en Espacios de Alta Dimensión**
   Funciona bien con datos donde el número de características es mayor que el número de muestras.

2. **Uso de Kernels para Datos No Lineales**
   Puede transformar datos no linealmente separables en espacios de mayor dimensión para hacerlos separables.

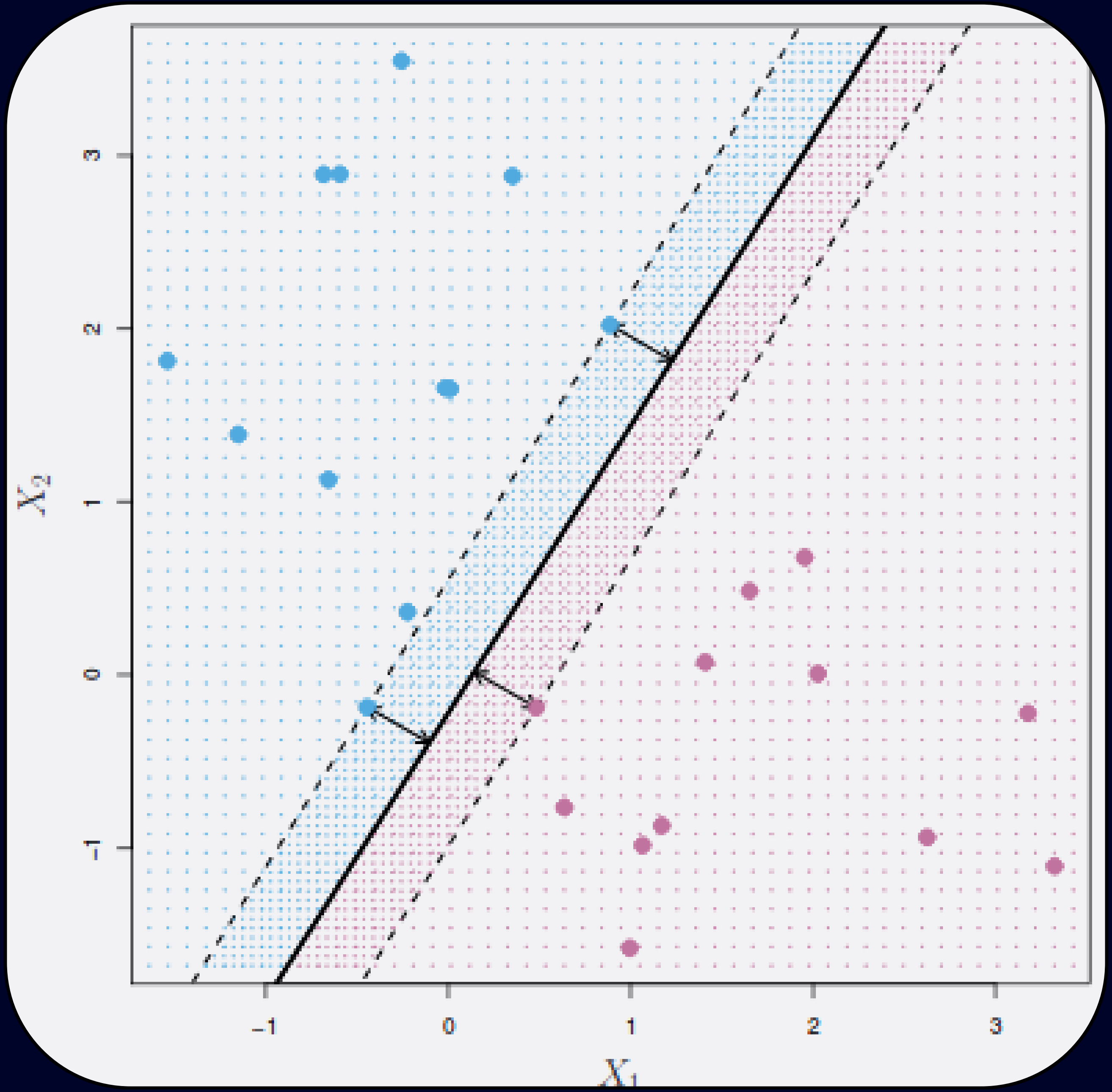
3. **Robustez ante Sobreajuste**
   Con un buen ajuste de hiperparámetros (C y gamma), evita el sobreajuste en datasets pequeños o ruidosos.

4. **Buena Generalización**
   Maximiza el margen de separación entre clases, lo que mejora su capacidad de generalización en nuevos datos.

Regresión vs SVM

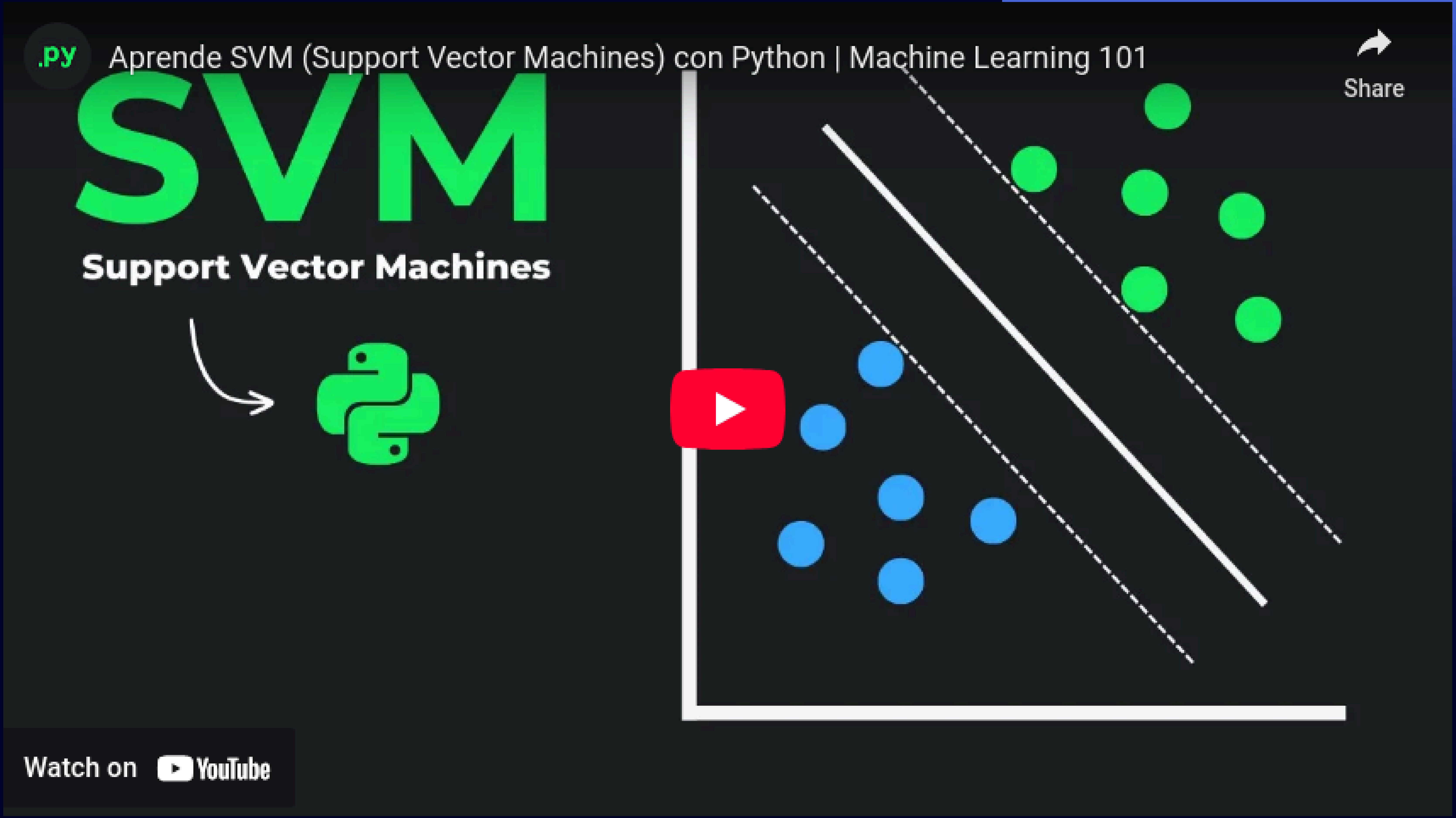


Una regresión Lineal encuentra muchos hiperplanos de separación



Un SVM encuentra el hiperplano de separación óptimo

Característica	**SVM**	**Regresión Logística**
Tipo de Modelo	Clasificador basado en márgenes	Clasificador basado en probabilidad
Función de Decisión	Encuentra el hiperplano con el mayor margen entre clases.	Modela la probabilidad de pertenencia a una clase.
Manejo de No Linealidad	Puede usar trucos de kernel para transformar datos no lineales.	Solo funciona bien en problemas linealmente separables (a menos que usemos transformaciones de características).
Robustez ante Outliers	Más resistente gracias a la maximización del margen y a las slack variables.	Más sensible a outliers, ya que optimiza la verosimilitud.
Velocidad de Entrenamiento	Más lento en grandes volúmenes de datos, especialmente con kernels complejos.	Generalmente más rápido y eficiente en grandes datasets.
Interpretabilidad	Difícil de interpretar cuando se usan kernels.	Fácil de interpretar, ya que los coeficientes representan la importancia de cada variable.
Uso en Alta Dimensión	Funciona bien con datos de muchas dimensiones (ejemplo: texto, imágenes).	Puede sufrir en dimensiones altas si no se aplica regularización adecuada.

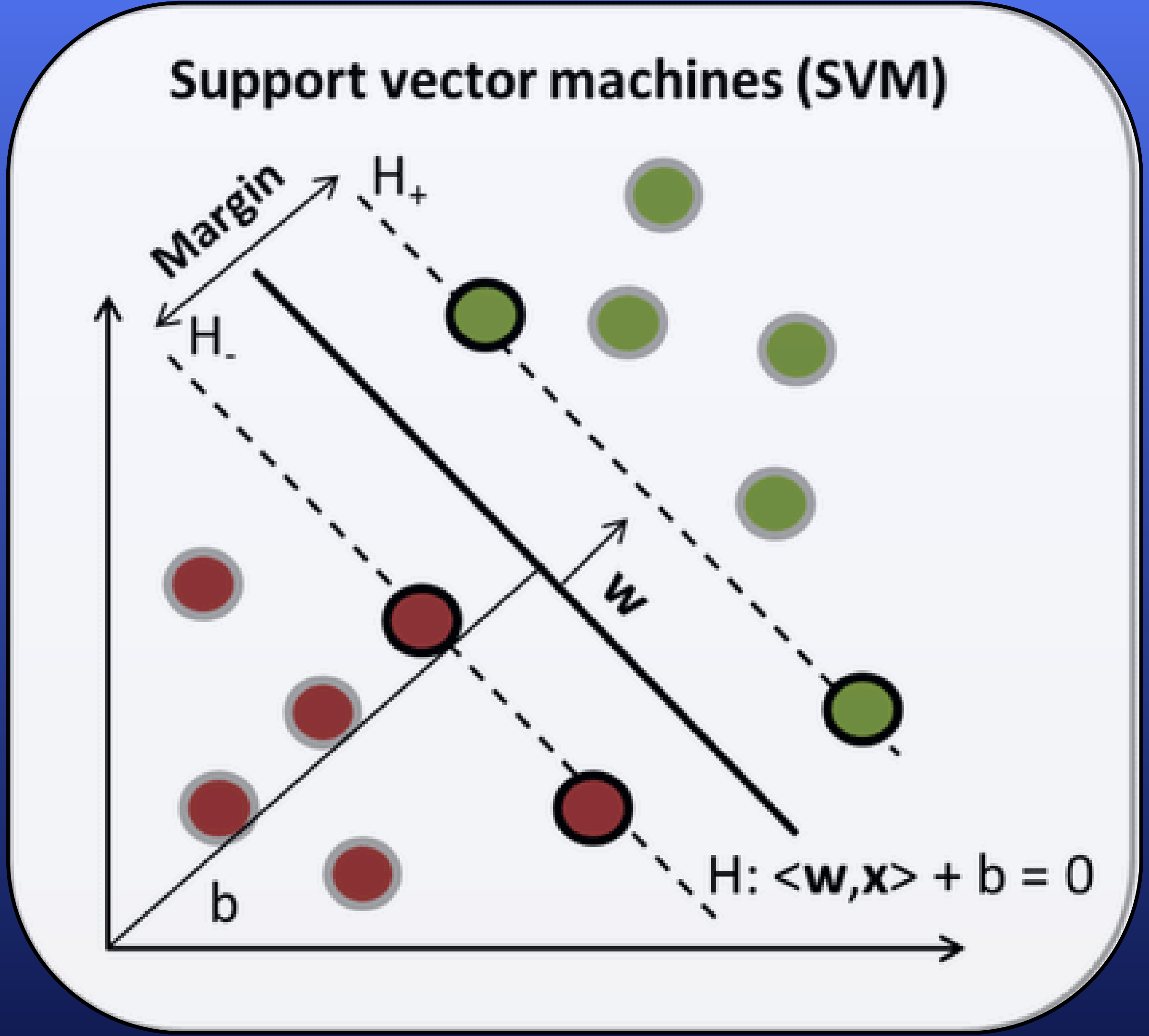


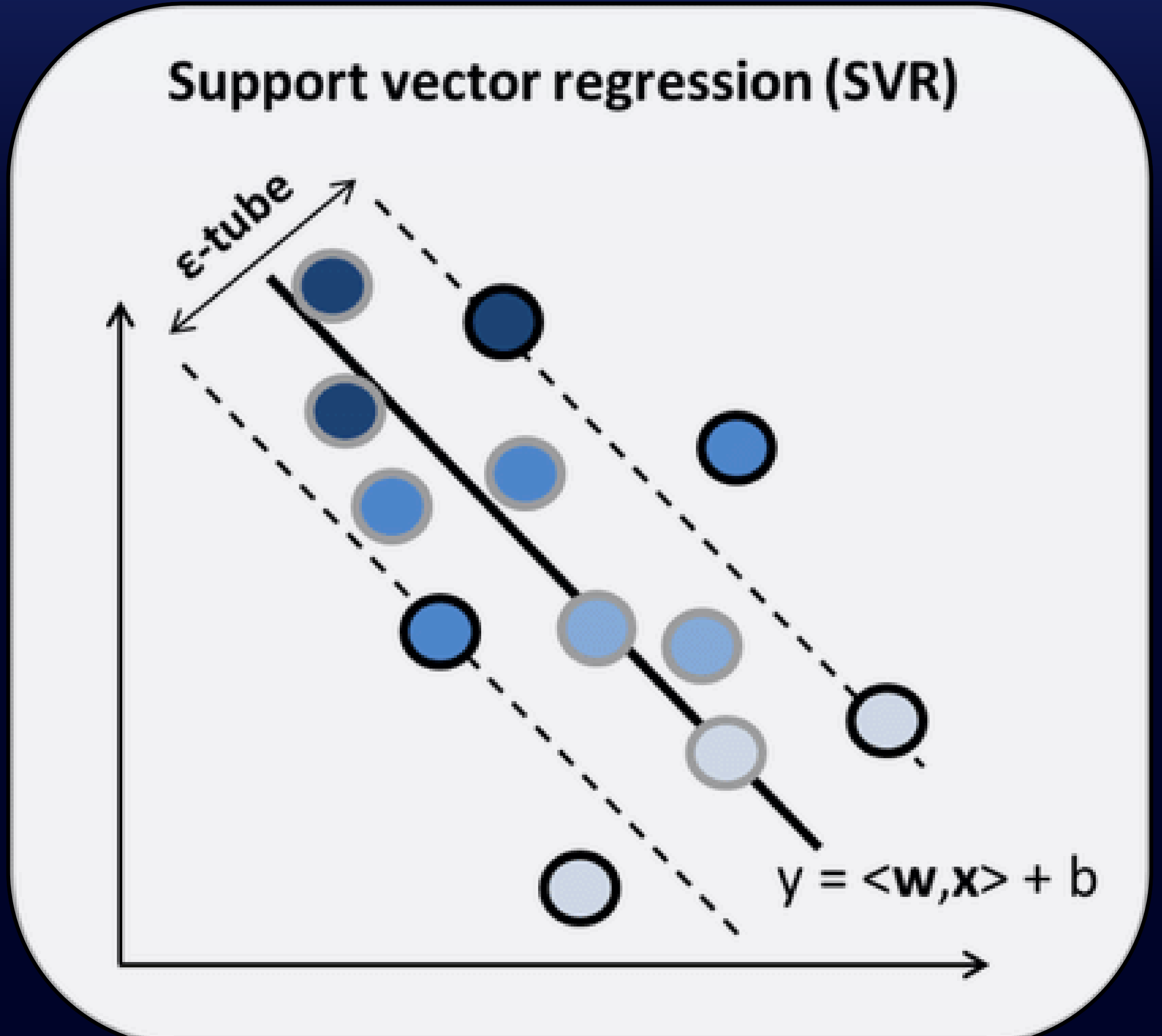
.py
Aprende SVM (Support Vector Machines) con Python | Machine Learning 101
Share
SVM
Support Vector Machines
Watch on YouTube

- ¿Cuándo usar SVM?
  - Cuando los datos no son linealmente separables y queremos usar kernels.
  - Cuando hay pocos datos y muchas características (ejemplo: bioinformática, texto).
- ¿Cuándo usar Regresión Logística?
  - Cuando queremos una interpretación clara de los coeficientes.
  - Cuando los datos son linealmente separables o queremos una probabilidad de clasificación.

Si el problema es lineal y queremos rapidez e interpretabilidad, usamos Regresión Logística.
Si el problema es más complejo, con patrones no lineales, SVM puede ser la mejor opción gracias a los kernels.

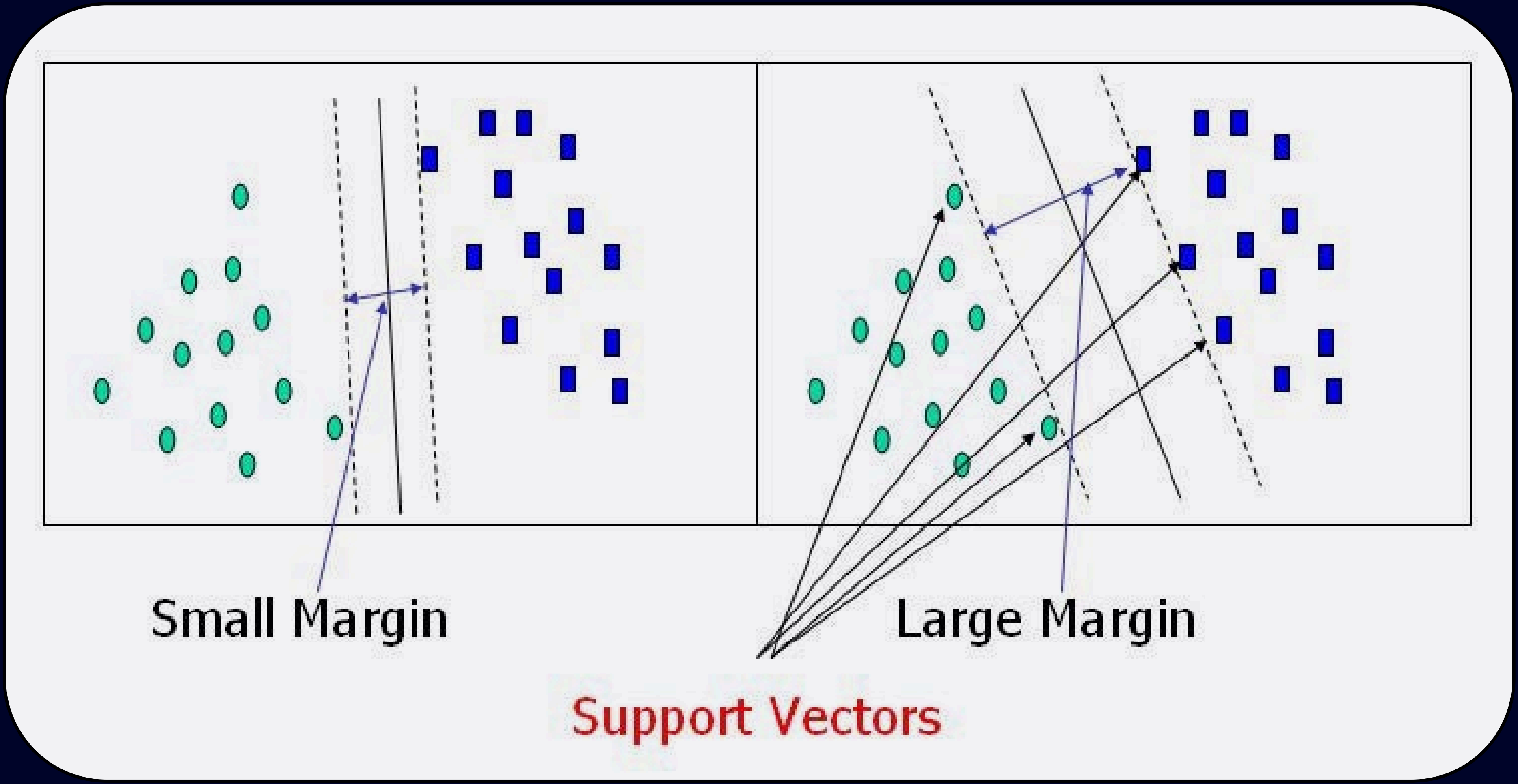
Característica	SVM (Support Vector Machine)	SVR (Support Vector Regression)
Propósito	Clasificación (asigna etiquetas a clases)	Regresión (predice valores continuos)
Función Objetivo	Encuentra el hiperplano óptimo con máximo margen para separar clases.	Encuentra un hiperplano óptimo que minimice el error dentro de un margen de tolerancia (ϵ\epsilonϵ).
Manejo de Datos No Lineales	Usa kernels para transformar datos no lineales en un espacio de mayor dimensión.	También usa kernels para modelar relaciones no lineales en regresión.
Concepto Clave	Maximización del margen entre clases.	Minimización del error dentro de una tolerancia ϵ\epsilonϵ (tubo de regresión).
Ejemplo de Uso	Detección de spam, reconocimiento facial, clasificación de imágenes.	Predicción de precios de casas, análisis financiero, series de tiempo.





Máxima Marginalidad en SVM

El margen es la distancia entre el hiperplano óptimo y los puntos más cercanos de cada clase. En SVM, buscamos maximizar este margen para mejorar la capacidad de generalización del modelo.



1. **Soporte Vectorial**
   Los vectores de soporte son los puntos de datos más cercanos al hiperplano de decisión.

2. **Hiperplano Óptimo**
   Esta condición garantiza que los vectores de soporte estén en los bordes del margen y que los datos estén correctamente clasificados dentro del margen máximo.

3. **Función de costo**
   Encuentra un hiperplano óptimo que maximice el margen entre clases y minimice los errores de clasificación.

SVM Conceptos

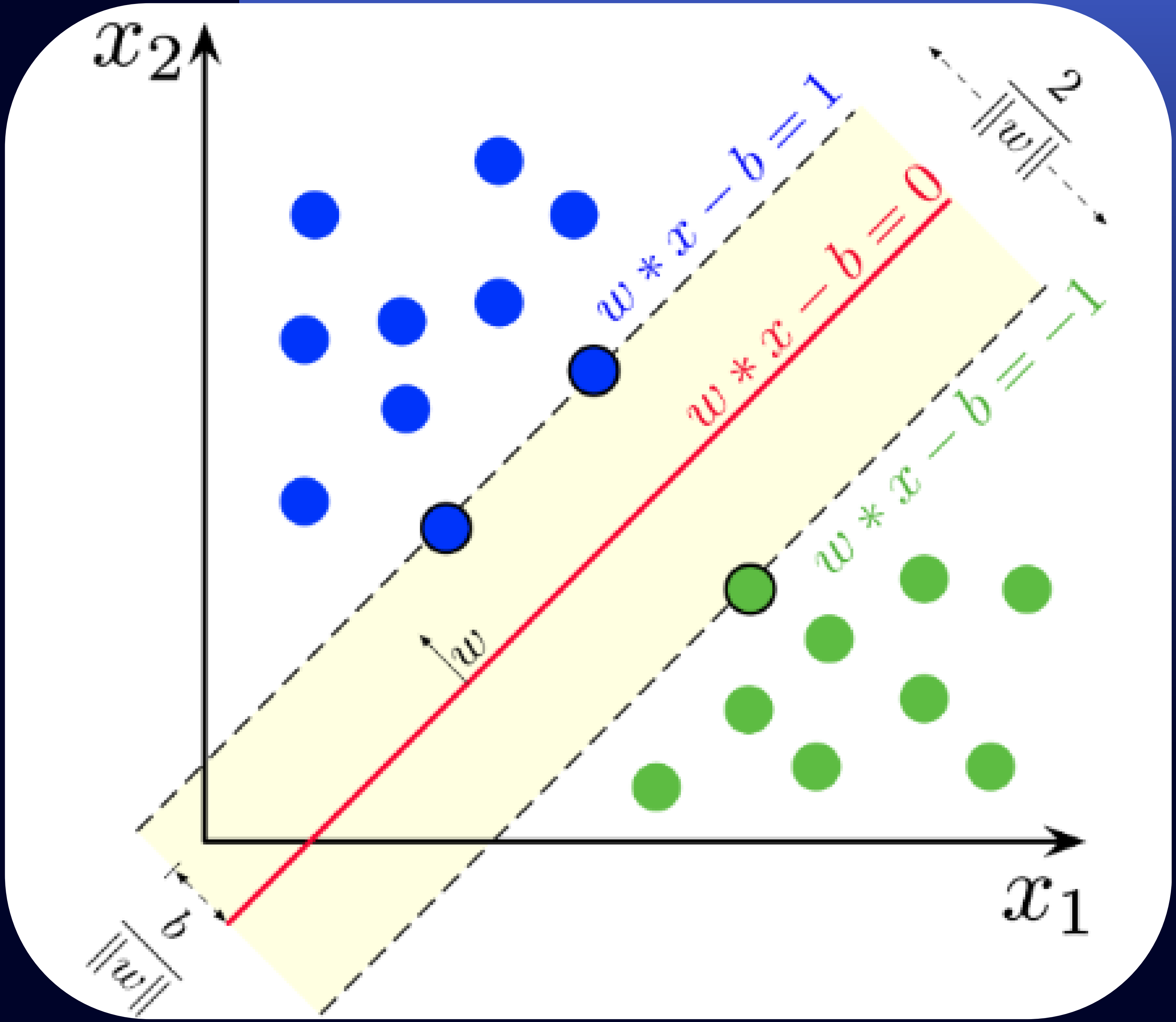
- **Hiperplano**

  Un hiperplano en un espacio de n dimensiones es una ecuación de la forma:

$$w \cdot x + b = 0$$

El vector ***x*** representa un punto en el espacio de entrada, es simplemente el conjunto de datos de entrada, con n dimensiones:

$$x = (x_1, x_2, ..., x_n)$$



norma euclidiana $\|(x_1,\ldots,x_n)\| = \sqrt{x_1^2+\cdots+x_n^2}$

SVM Conceptos

Hiperplano

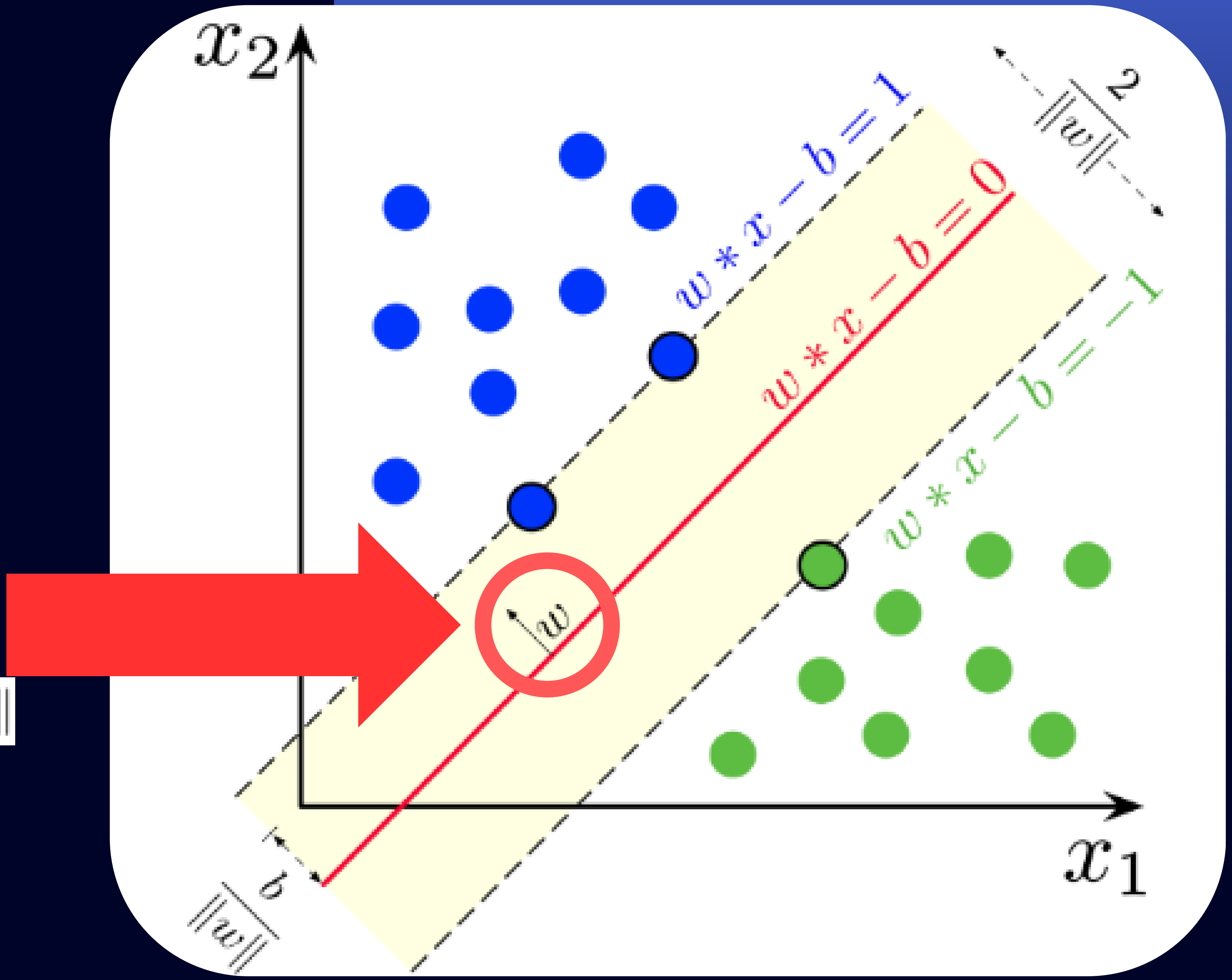
El vector de pesos **w** es un vector normal (perpendicular) al hiperplano de separación. Define la dirección y orientación del hiperplano en el espacio.

$$w=(w_1,w_2,\ldots,w_n)$$

Se obtiene resolviendo un problema de optimización en SVM, busca minimizar la norma $\|w\|$ (norma del vector) mientras mantiene las restricciones de clasificación.

Se usan métodos matemáticos como:

- Multiplicadores de Lagrange
- Método del gradiente



una norma en un espacio vectorial es un operador que permite definir una noción de "longitud" o "tamaño" de cualquier vector.

norma euclidiana : $\|(x_1, \ldots, x_n)\| = \sqrt{x_1^2 + \cdots + x_n^2}$

SVM Conceptos

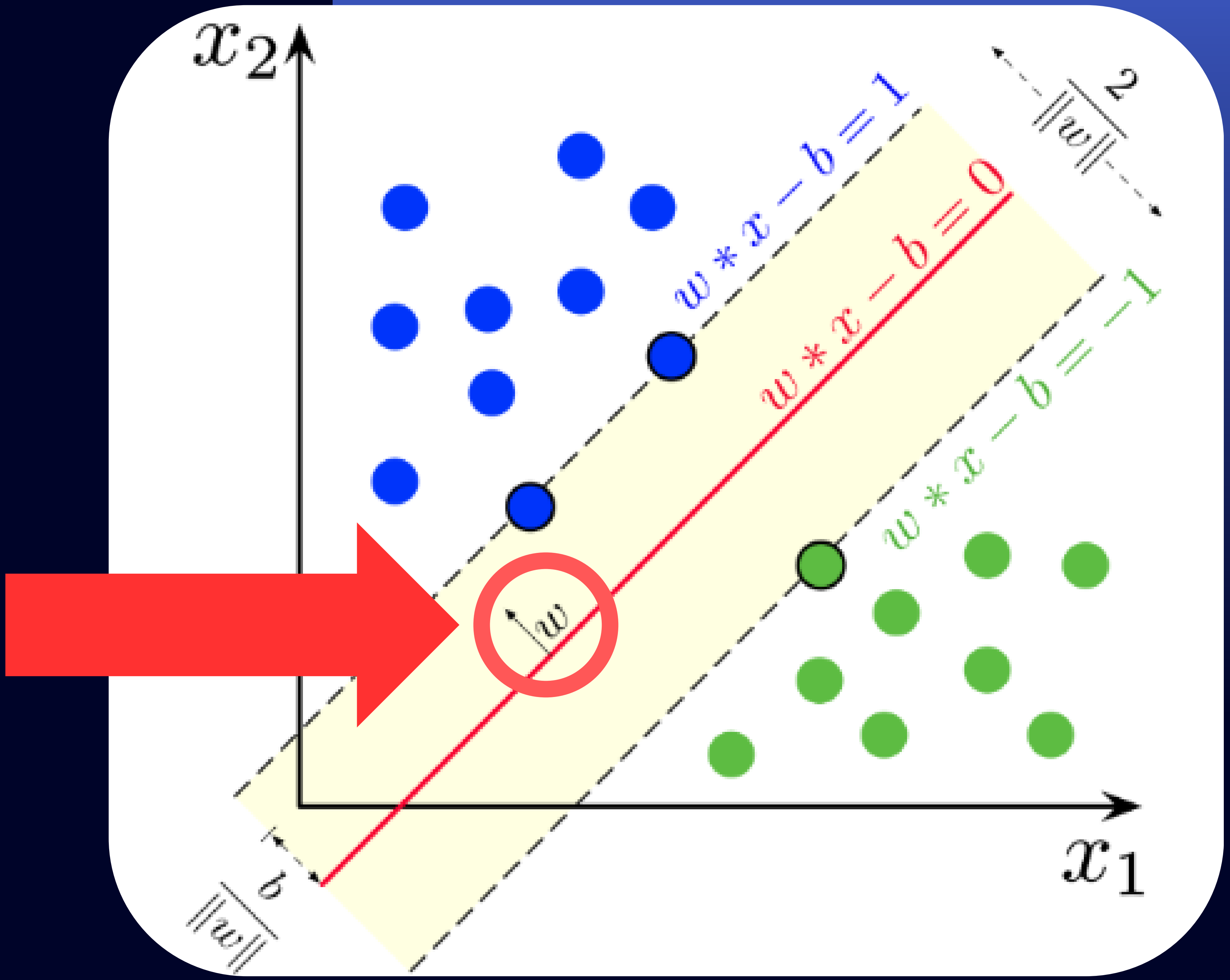
- Norma $\|w\|$

  Si tienes un vector de pesos w=(w1,w2,…,wn), su norma Euclidiana se calcula así:

  $$\|\mathbf{w}\| = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$$

  Supongamos que w=(3,4)

  $$\|\mathbf{w}\| = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$$



una norma en un espacio vectorial es un operador que permite definir una noción de "longitud" o "tamaño" de cualquier vector.


.py
Gradiente Descendente desde cero con Python | Deep Learning 101
Share
Gradiente Descendente
el algoritmo para entrenar una IA
Watch on YouTube

SVM Conceptos

Hiperplano

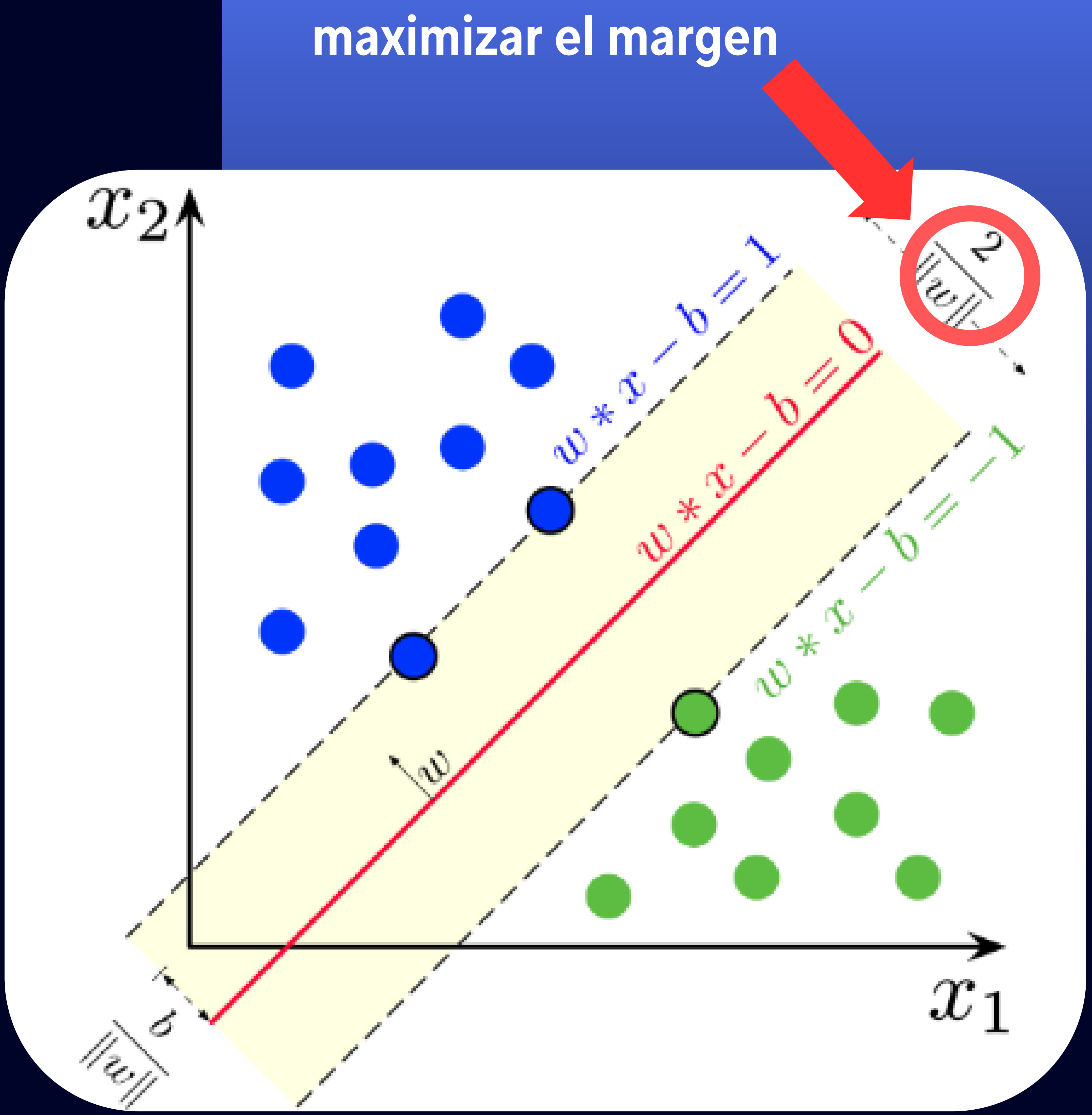
El objetivo es encontrar el hiperplano que maximiza el margen.

Para esto, se resuelve el siguiente problema de optimización:

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

Esto se hace porque:

- **minimizar $\|w\|^2$ equivale a maximizar el margen $\frac{2}{\|w\|}$**
- Usar $\|w\|^2$ en lugar de $\|w\|$ facilita la optimización, ya que se evita la raíz cuadrada



SVM Conceptos

Restricciones

Los puntos más cercanos al hiperplano (vectores soporte) deben cumplir:

- $w \cdot x_{+} + b = +1$ (para clase +1)
- $w \cdot x_{-} + b = -1$ (para clase -1)

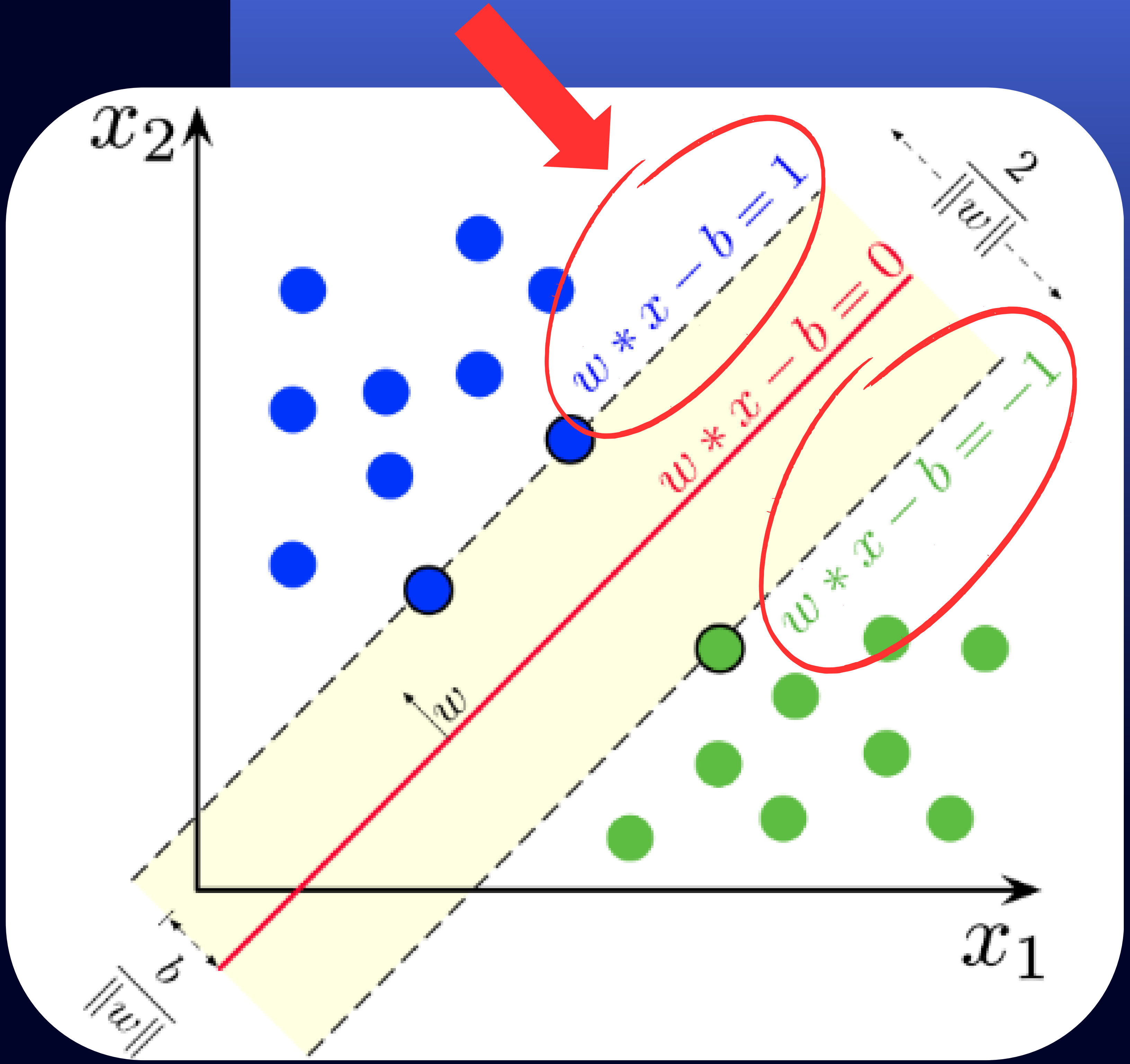
Queremos que todos los puntos cumplan:

- Si $y_i = +1$: $\mathbf{w} \cdot \mathbf{x}_i + b \geq +1$
- Si $y_i = -1$: $\mathbf{w} \cdot \mathbf{x}_i + b \leq -1$

Esto se puede compactar en una sola restricción:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \qquad \forall \; i$$

Restricciones

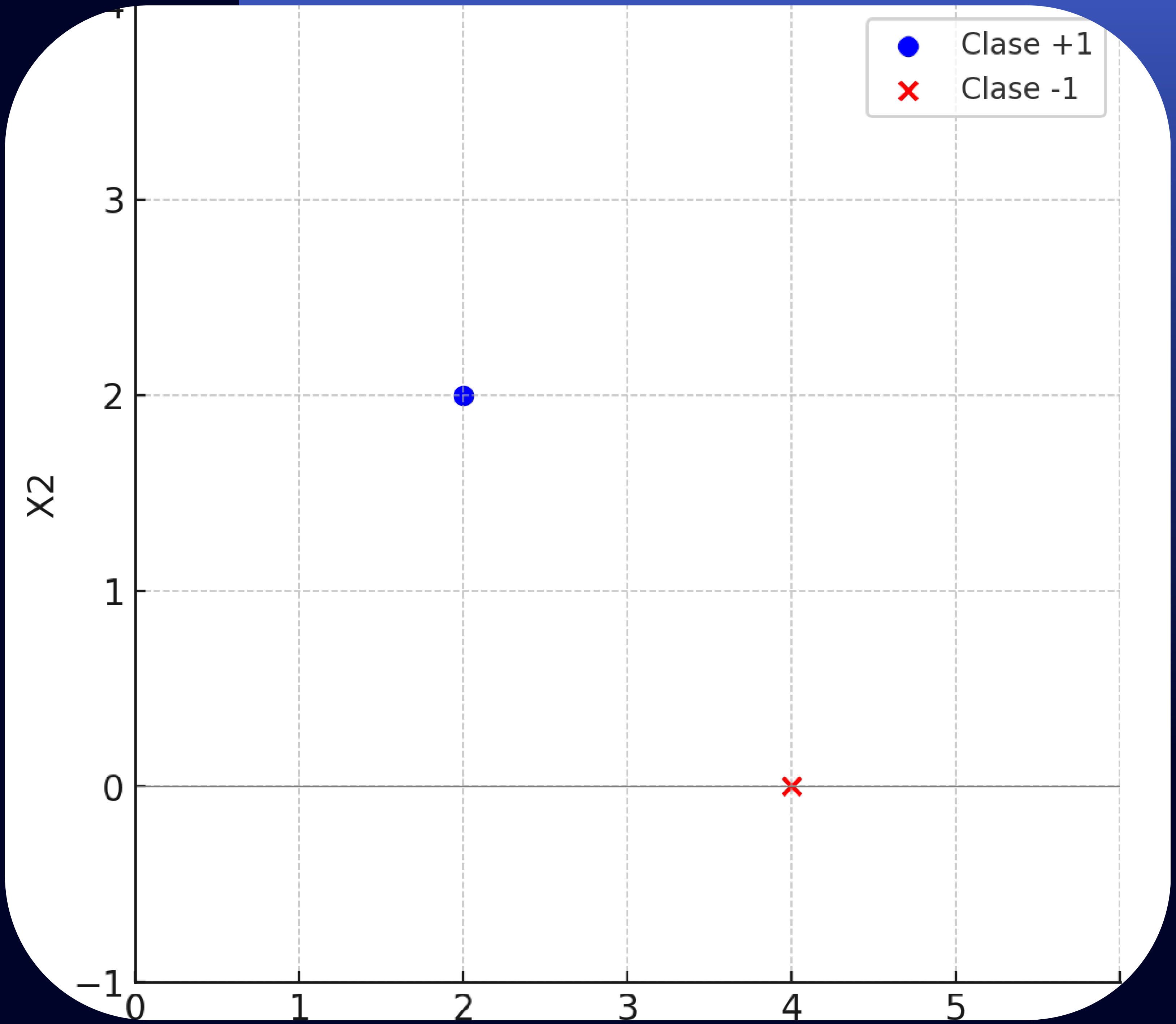


SVM Conceptos

- Ejemplo Demostrativo - Escogiendo puntos para Vectores de Soporte

X_1	X_2	Y
2	2	1
4	0	-1

Entonces, tomando en consideracion la ecuacion: **$y_i(wx_i+b) \geq 1$**,que representa la restricción de optimización que asegura que todos los puntos de entrenamiento queden más allá o exactamente sobre sus líneas de soporte correspondientes



SVM Conceptos

- Ejemplo Demostrativo - Plantear las restricciones de optimización

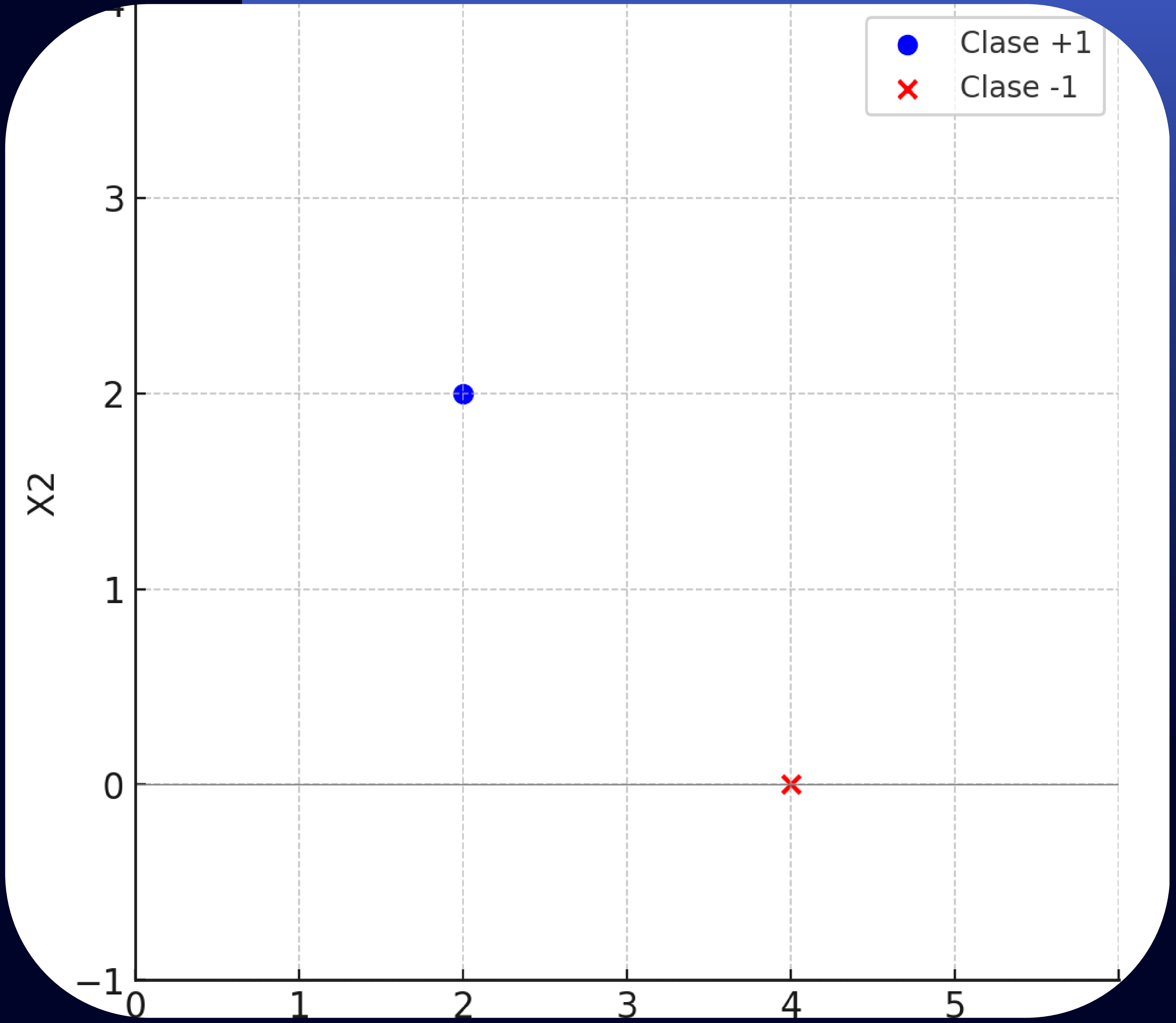
Si **$y_i(wx_i+b) \geq 1$** entonces usando los puntos para vectores de soporte, definimos las restricciones:

Para $x_1=2$, $x_2=2$, $y=+1$

- **$(2w_1+2w_2+b) \geq 1$**

Para $x_1=4$, $x_2=0$, $y=-1$

- $-(4w_1+0w_2+b) \geq 1 \Rightarrow$ **$4w_1+b \leq -1$**



SVM Conceptos

- Ejemplo Demostrativo - Resolver el sistema de ecuaciones

Dadas las restricciones anteriores, los vectores de soporte son los que cumplen con la igualdad:

- $\mathbf{2w_1+2w_2+b=1}$
- $\mathbf{4w_1+b=-1}$

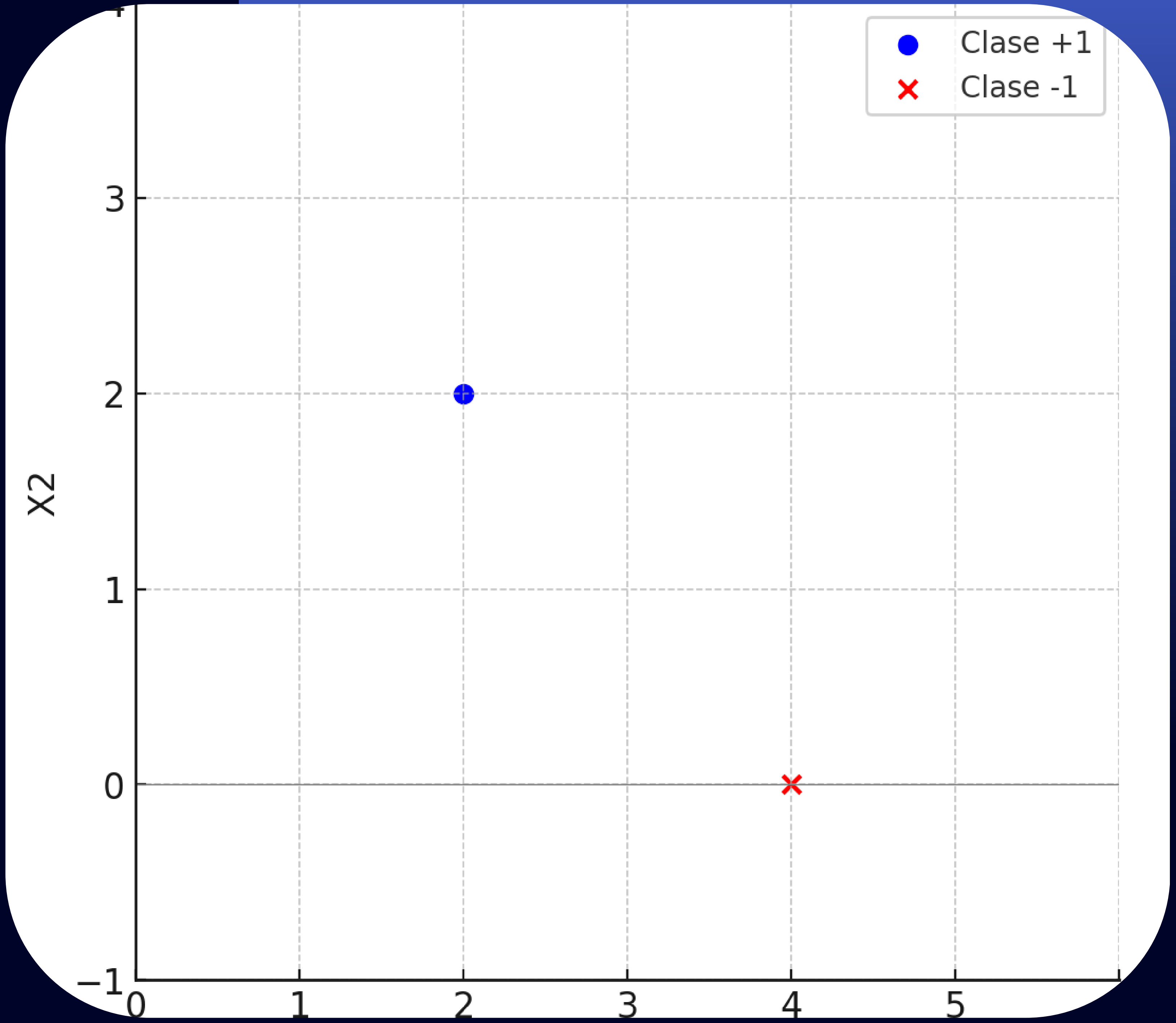
Estas ecuaciones son un sistema de 2 ecuaciones con 3 incógnitas (w_1,w_2,b)

De la segunda ecuación:

- $b=-1-4w_1$

Sustituimos en la primera:

$2w_1+2w_2+(-1-4w_1)=1$



SVM Conceptos

- Ejemplo Demostrativo - Resolver el sistema de ecuaciones

$2w_1+2w_2+(-1-4w_1)=1$

Entonces

$$-2w_1+2w_2-1=1$$

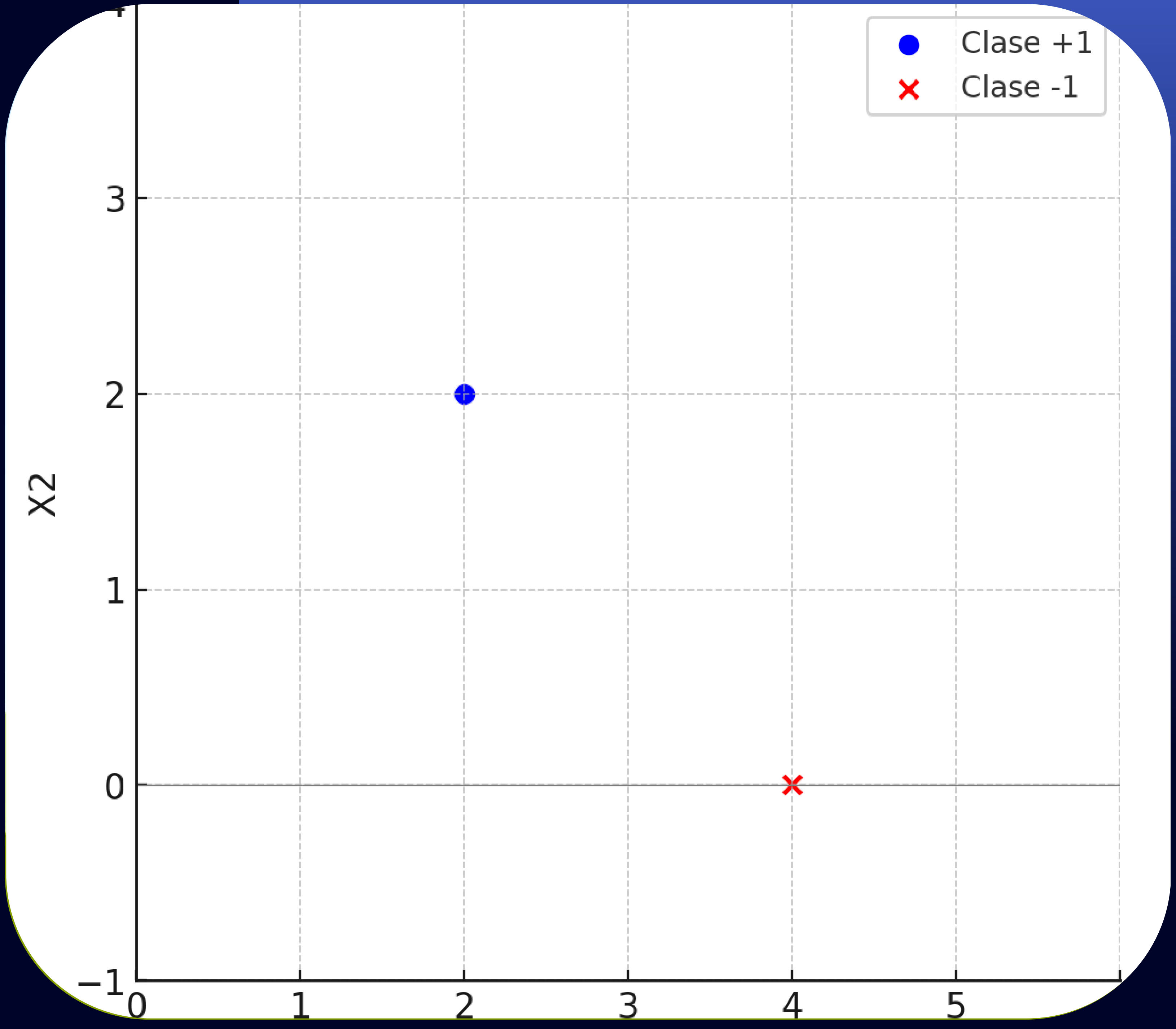
$$-2w_1+2w_2=2 \Rightarrow -w_1+w_2=1 \Rightarrow w_2=w_1+1$$

Expresamos la norma:

Queremos minimizar $\|\mathbf{w}\|^2 = w_1^2 + w_2^2$

Reemplazamos w_2 en $\|\mathbf{w}\|^2 = w_1^2 + (w_1+1)^2$

Expandimos: $w_1^2+(w_1^2+2w_1+1) \Rightarrow 2w_1^2+2w_1+1$



SVM Conceptos

Ejemplo Demostrativo - Optimizar

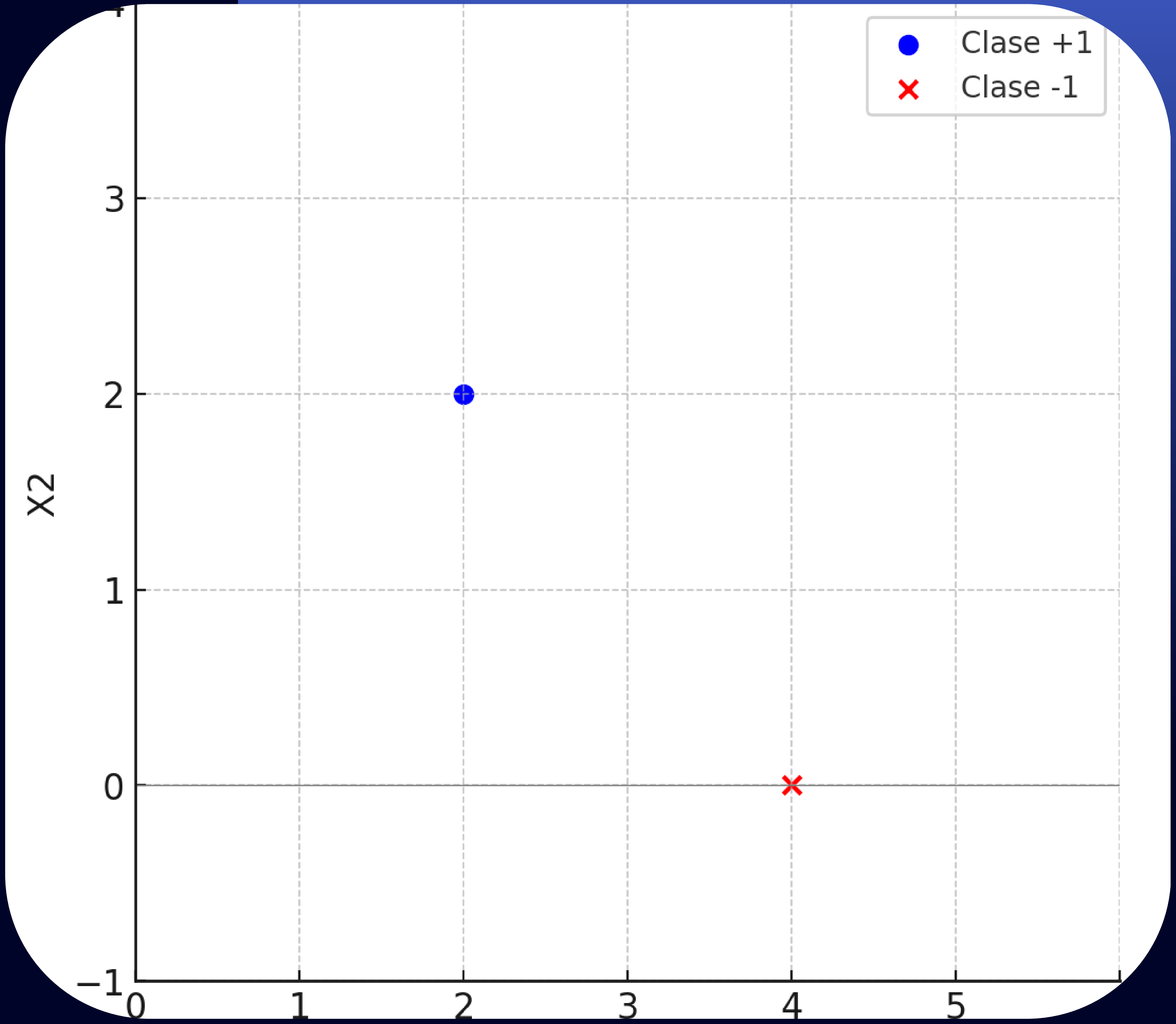
Obtuvimos $\|\mathbf{w}\|^2$: $2w_1^2+2w_1+1$
Para minimizar esta función cuadrática, necesitamos derivar respecto a w_1:

$$\frac{d}{dw_1}(2w_1^2+2w_1+1)=4w_1+2$$

e igualamos a cero, para encontrar el minimo de w_1:

- $4w_1+2=0 \Rightarrow w_1=-1/2$

Dado w_1, calculamos w_2:
$w_2=w_1+1=(-1/2)+1=1/2$



SVM Conceptos

Sesgo

El sesgo ***b*** es un escalar que ajusta la posición del hiperplano.
Si b=0, el hiperplano pasa por el origen.
Si b≠0, el hiperplano se desplaza.

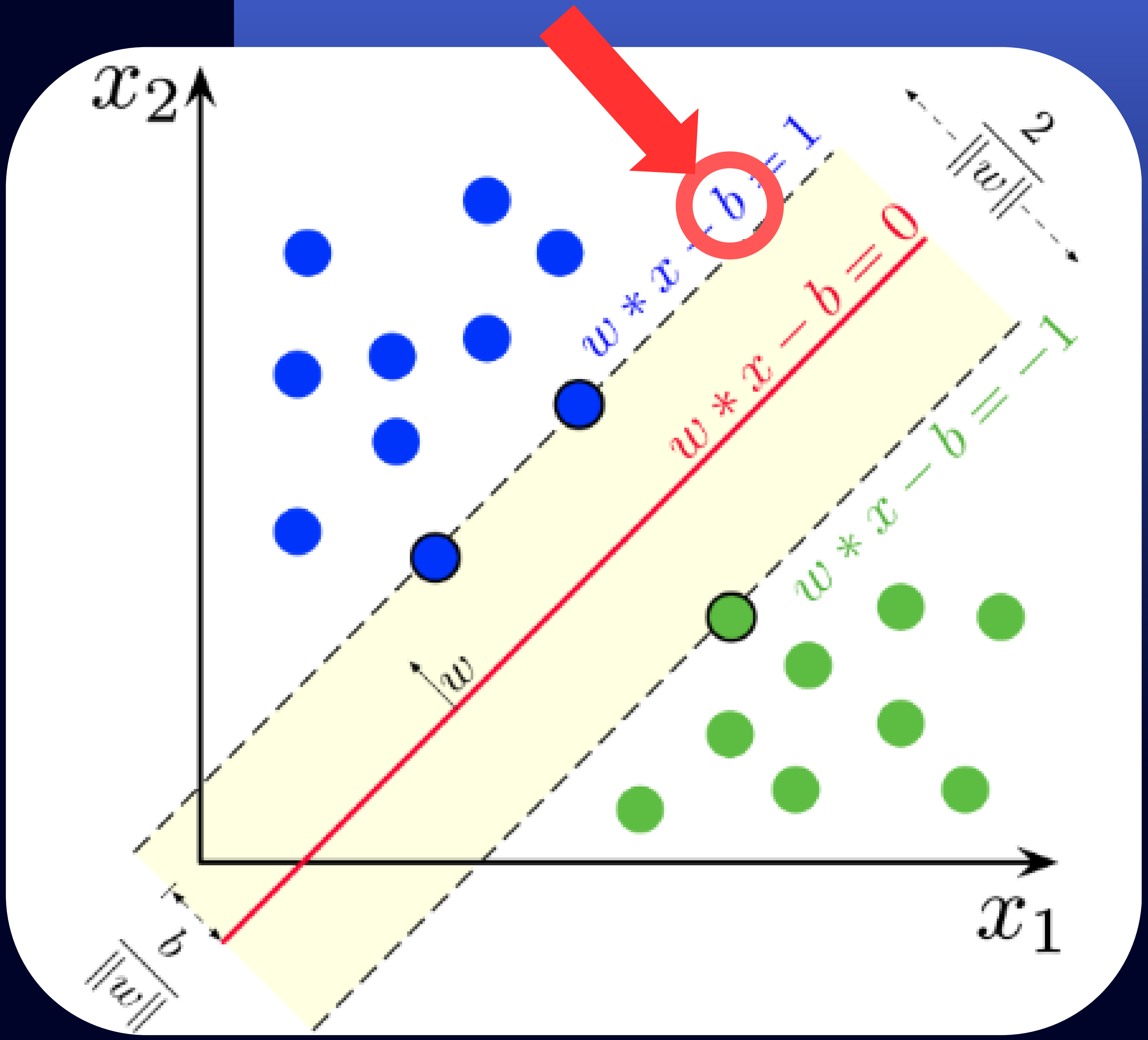
Se calcula después de obtener el vector de pesos w

Para calcular b:

1. Se obtiene el vector de pesos w
2. Se elige un vector de soporte X_i
3. Se usa la fórmula:
    a. $b = Y_i - w \cdot X_i$

donde Y_i es es la etiqueta de la clase

Sesgo o Bias



SVM Conceptos

- Ejemplo Demostrativo - Determinar el sesgo

Finalmente, sustituimos en b:
$b=-1-4w_1=-1-4(-1/2)=-1+2=\underline{1}$

Entonces la ecuación del hiperplano es:

$$\underline{w_1x_1+w_2x_2+b=0}$$

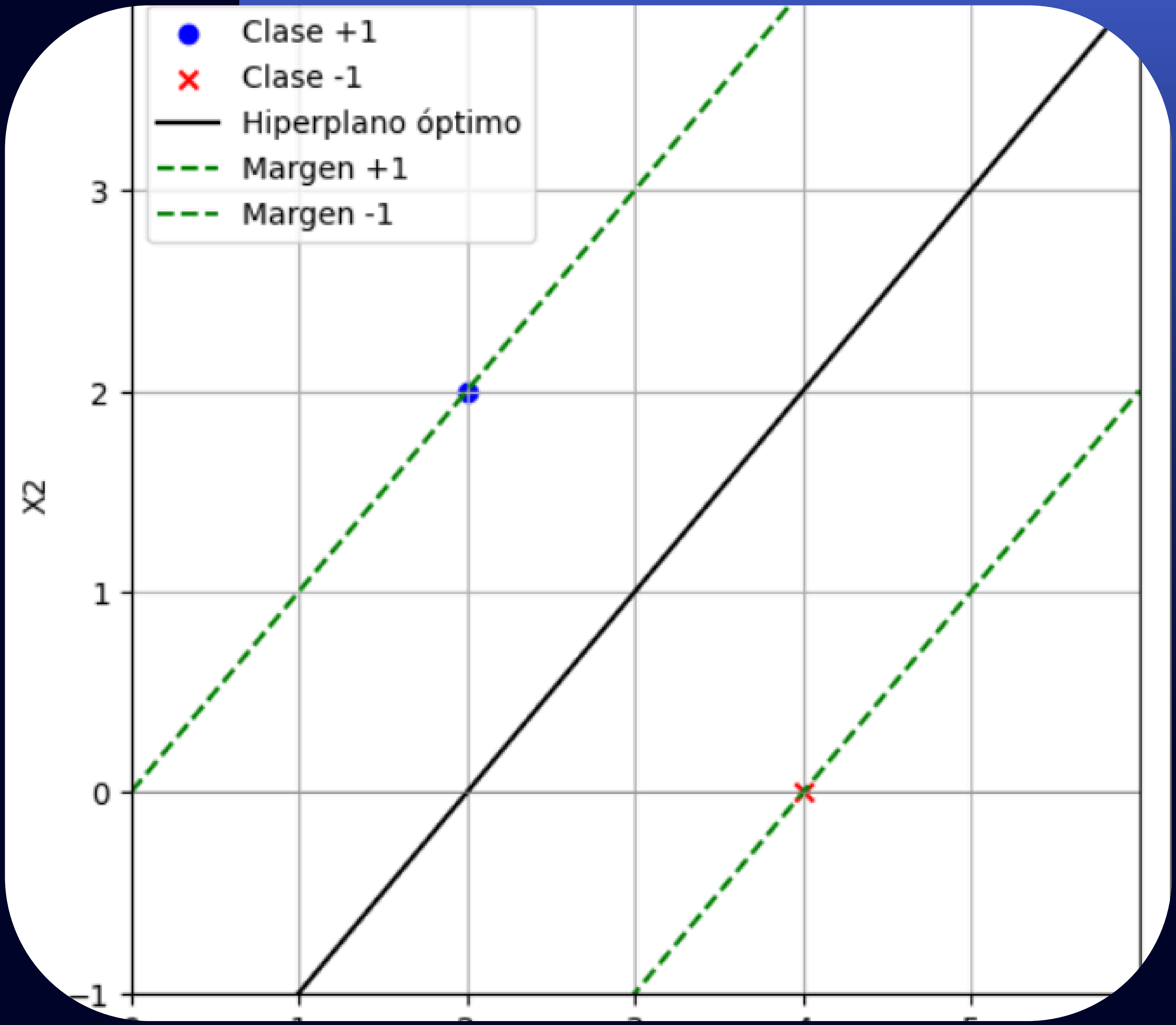
$$\underline{(-1/2)x_1+(1/2)x_2+1=0}$$

La norma de w es:

$$\|\mathbf{w}\| = \sqrt{\left(-\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2} = \sqrt{\frac{1}{4}+\frac{1}{4}} = \sqrt{\frac{1}{2}} = \frac{1}{\sqrt{2}}$$

El ancho del margen es:

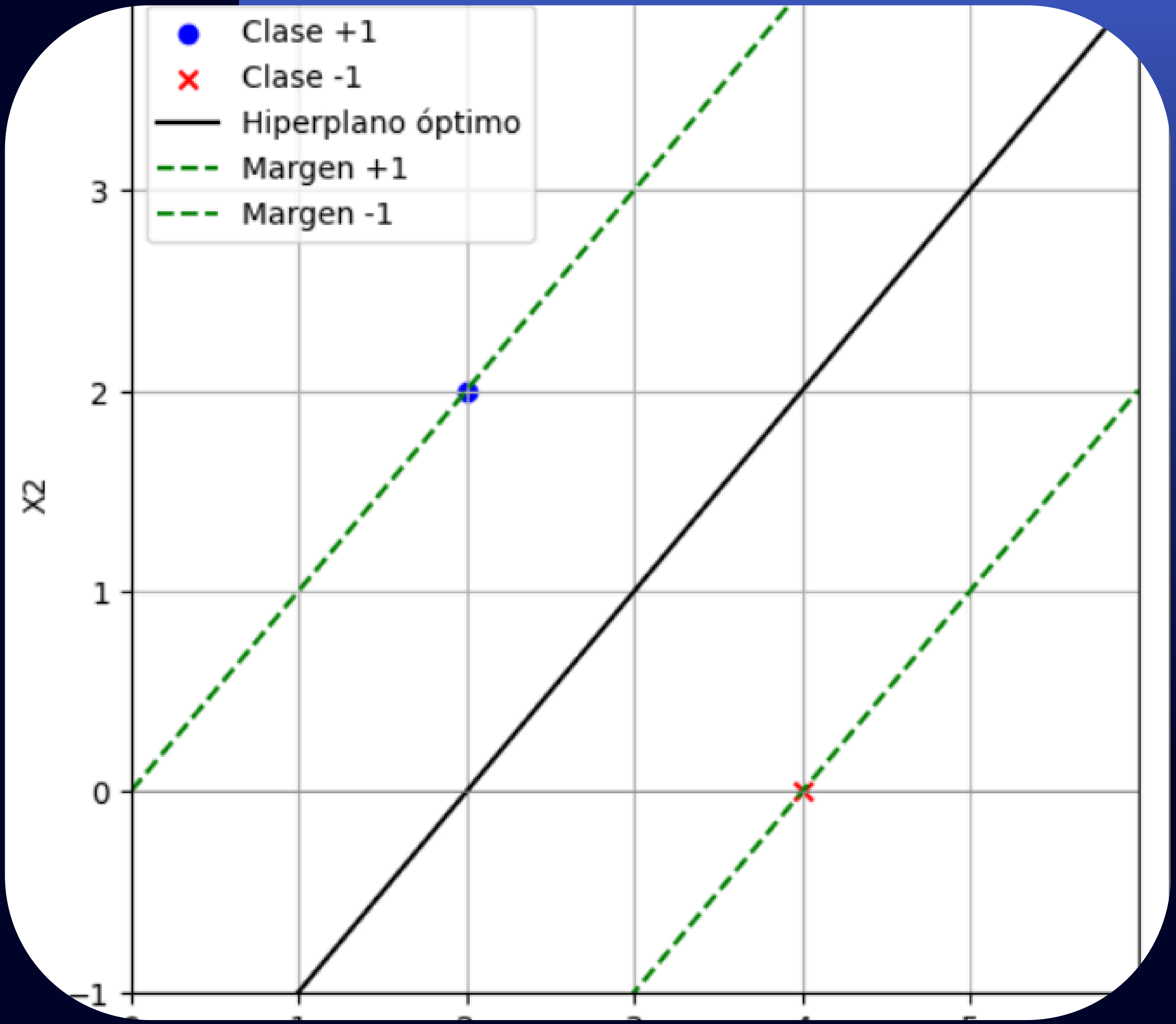
$$\text{Margen} = \frac{2}{\|\mathbf{w}\|} = 2\sqrt{2}$$



SVM Conceptos

- Resumen

Variable	Valor
w_1	-1/2
w_2	1/2
b	1
$\|\mathbf{w}\|$	$\frac{1}{\sqrt{2}}$
Margen	$2\sqrt{2}$



Algoritmo SVM como Clasificador

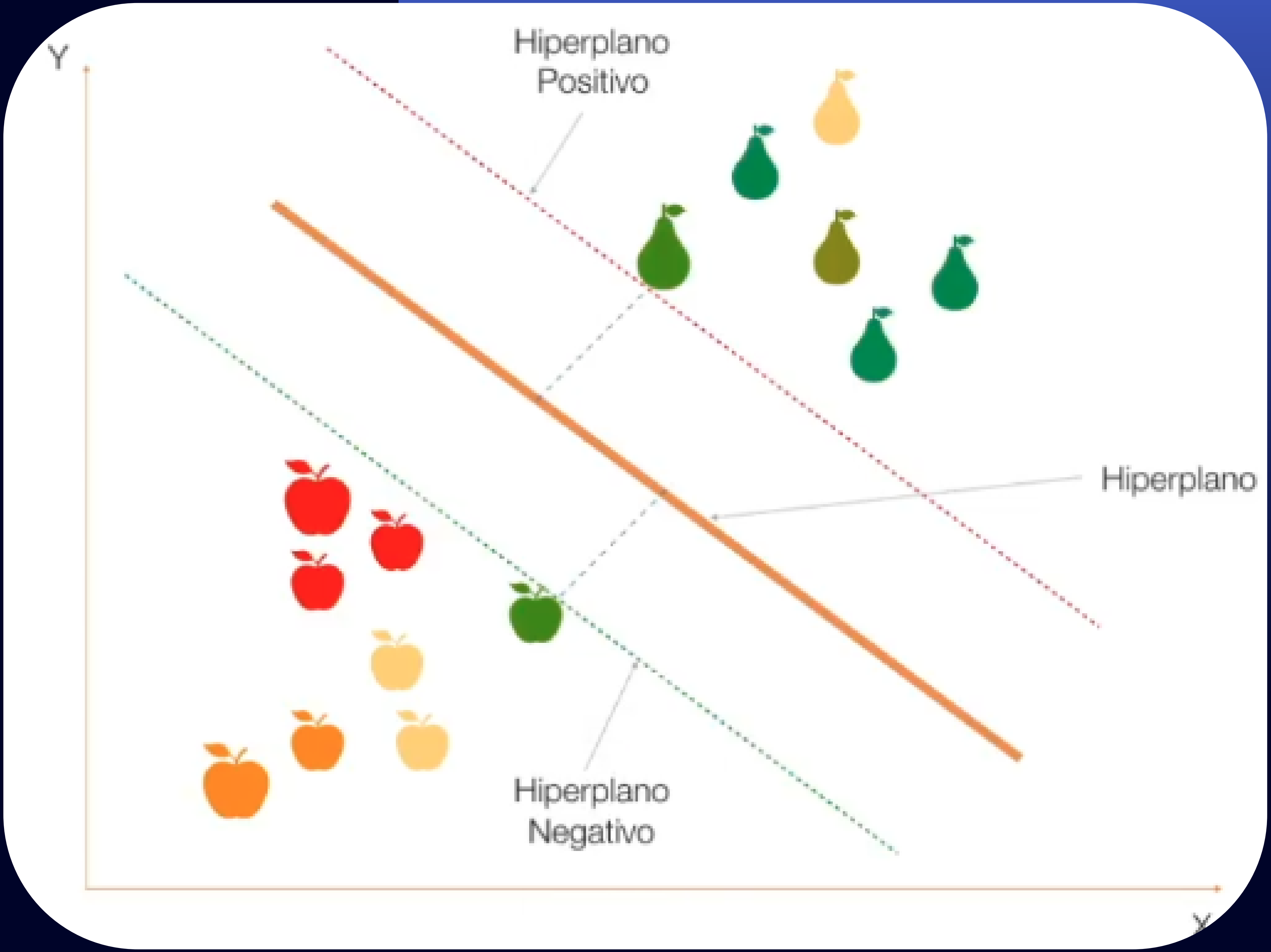
Clasificar Nuevos Puntos

Una vez entrenado el modelo, se puede clasificar un nuevo punto de datos (x_1,x_2…, x_n) usando la función de decisión:

$$\text{Clase} = \text{sign}(w_1 x_1 + w_2 x_2 + b)$$

donde la función **sign** tiene la siguiente lógica:

- Si el argumento es negativo, la función devuelve -1.
- Si el argumento es cero, la función devuelve 0.
- Si el argumento es positivo, la función devuelve 1.



Algoritmo SVM como Clasificador

- **Prediccion, usando el ejemplo anterior**

Dado $x_1=1$ y $x_2=3$, predecir Y,

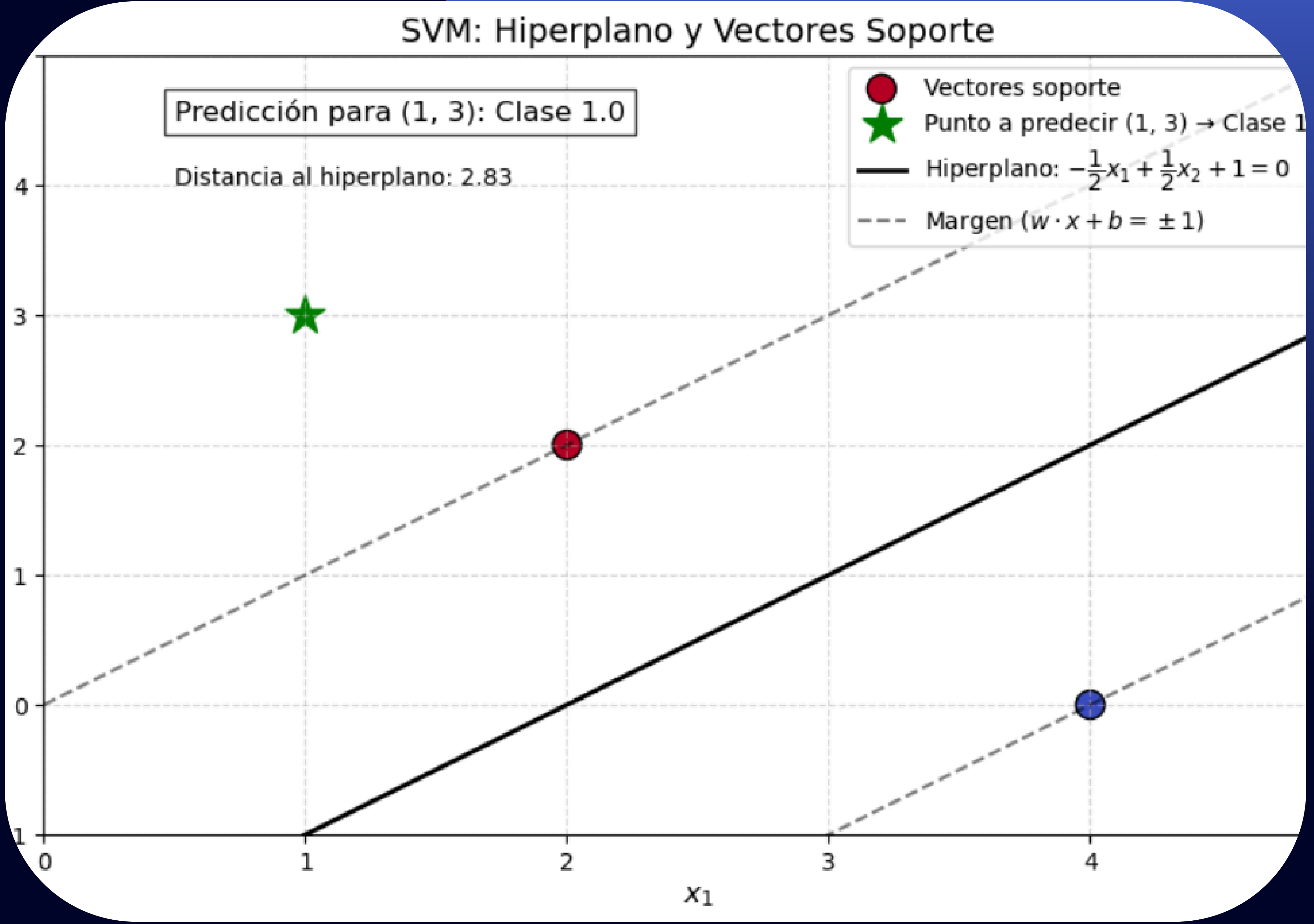
Sabemos que: **$(-1/2)x_1+(1/2)x_2+1$**

Entonces: (-0.5)(1)+(0.5)(3)+1= 2

Dado:

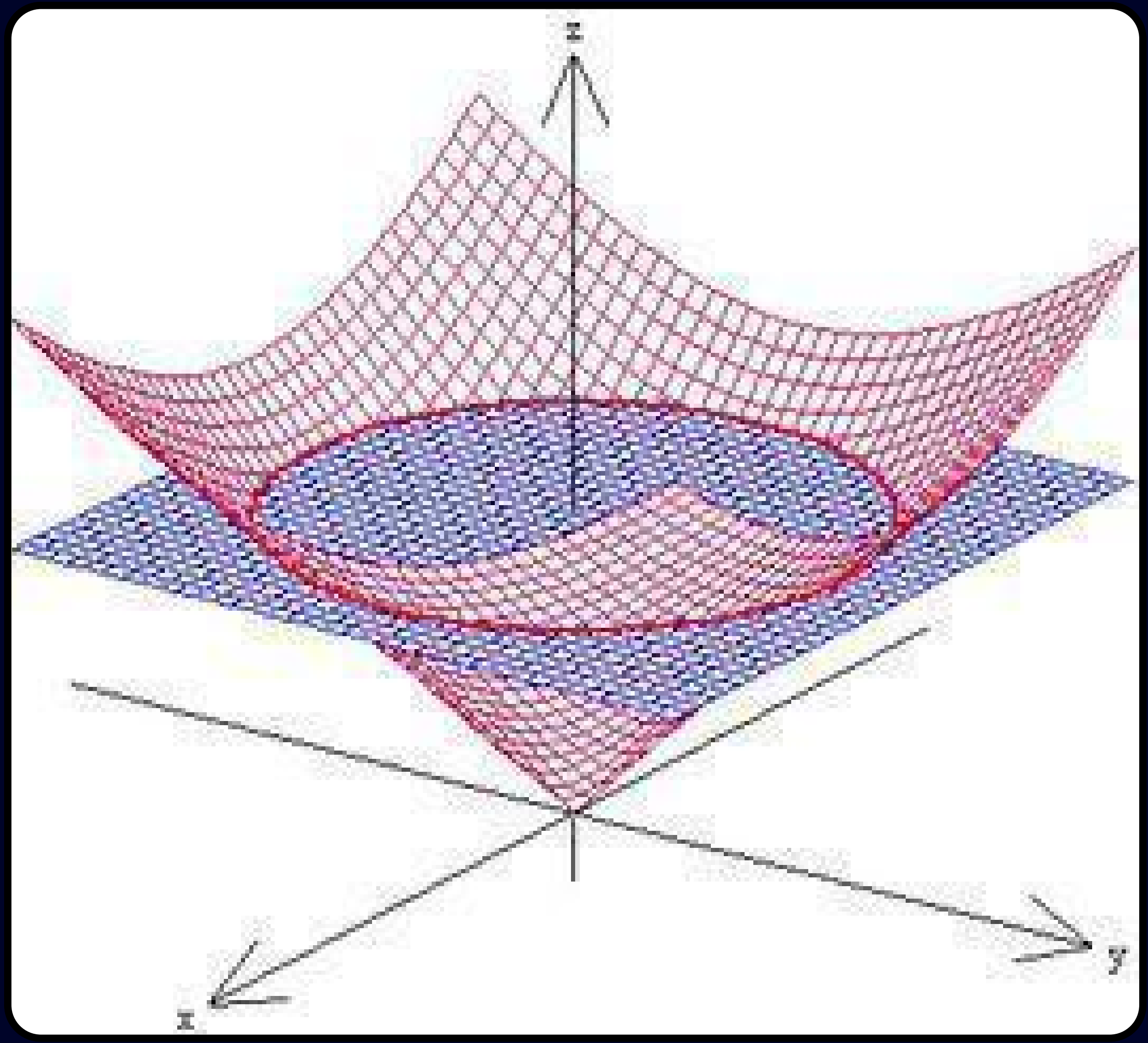
$$\hat{y} = \begin{cases} +1, & \text{si } f(\mathbf{x}) > 0, \\ -1, & \text{si } f(\mathbf{x}) < 0. \end{cases}$$

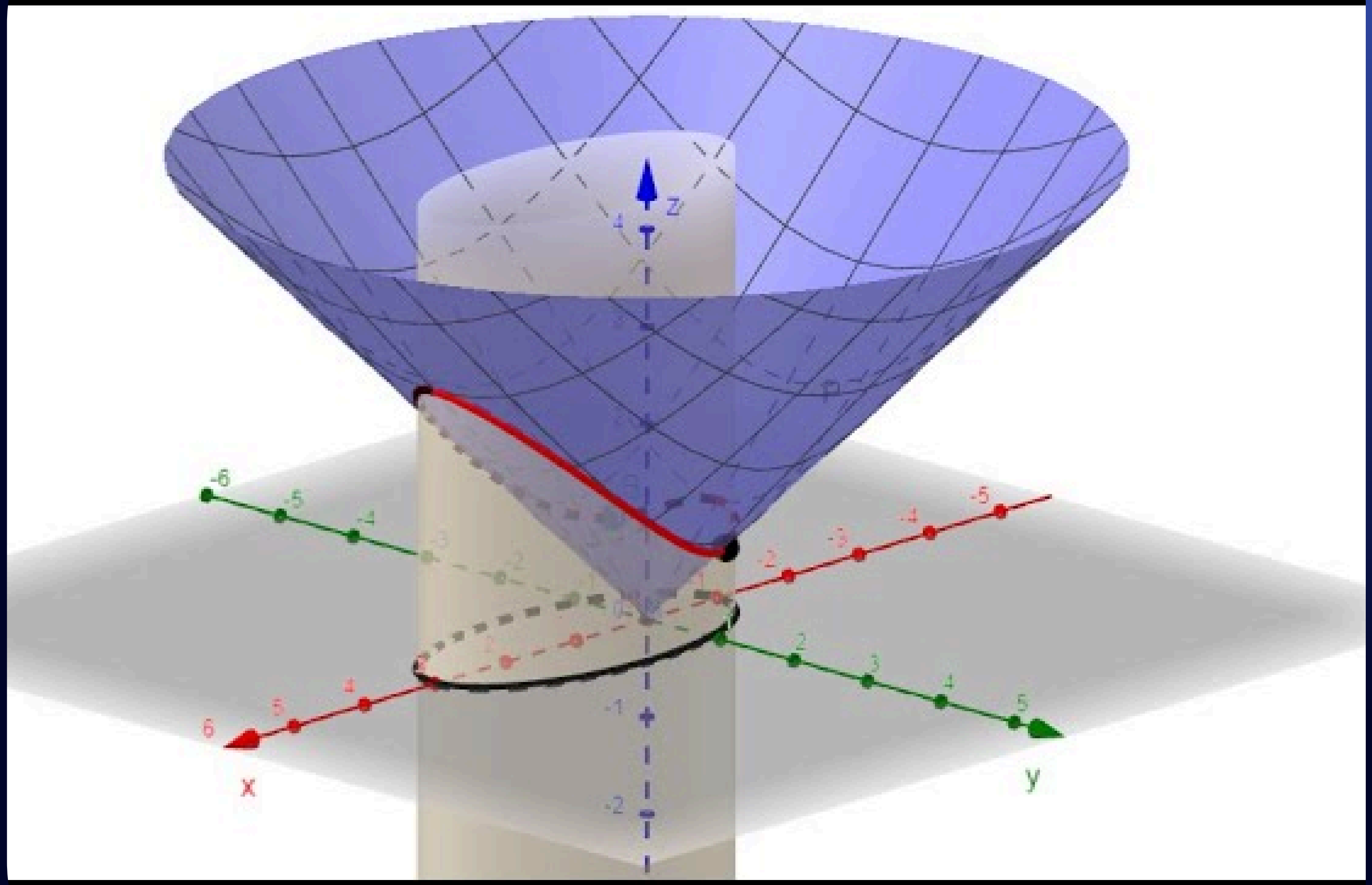
f(x)=2>0 entonces "*y*" predicha es = +1



Algoritmo SVM como Clasificador

- Esto es factible cuando tenemos un sistema de dos variables (2 dimensiones), para sistemas superiores se debe utilizar multiplicadores de Lagrange.





Demostracion cuando tenemos Xi Parametros

- Cuando tenemos X_i y Y_i

$$x_i \in \mathbb{R}^d, y_i \in \{+1, -1\}$$

Nuevo problema de optimizacion, removiendo restricciones, introduciendo un multiplicador de Lagrange ($\alpha_i \geq 0$) para cada restricción

$$y_i(w \cdot x_i + b) - 1 \geq 0$$

Entonces:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(w \cdot x_i + b) - 1 \right]$$

donde N, es el numero de vectores de soporte.

Demostracion cuando tenemos Xi Parametros

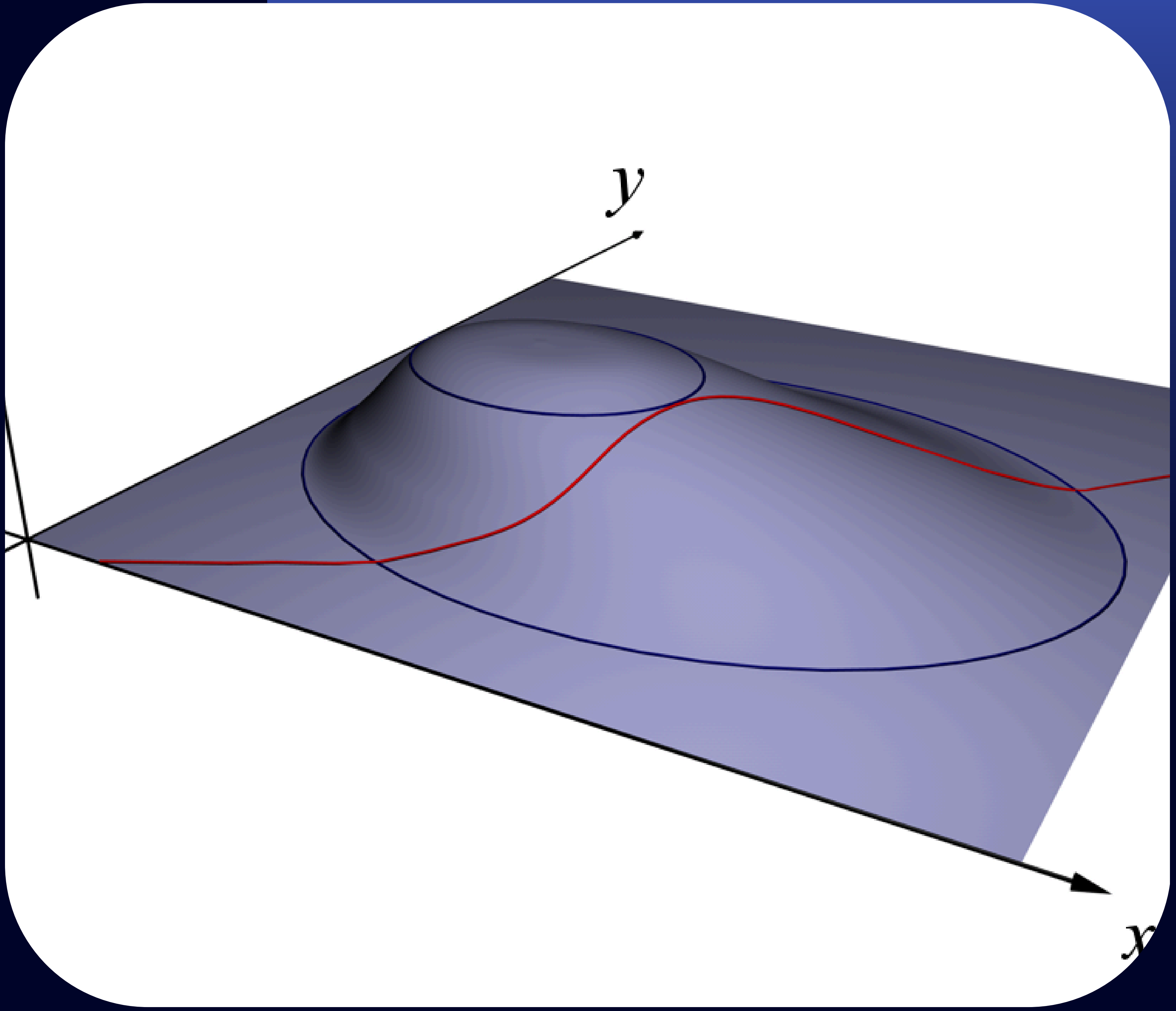
Para obtener el minimo de la funcion:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i (w \cdot x_i + b) - 1 \right]$$

derivamos con respecto w y b igualando a 0

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \implies w = \sum_{i=1}^{N} \alpha_i \, y_i \, x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{N} \alpha_i \, y_i = 0.$$



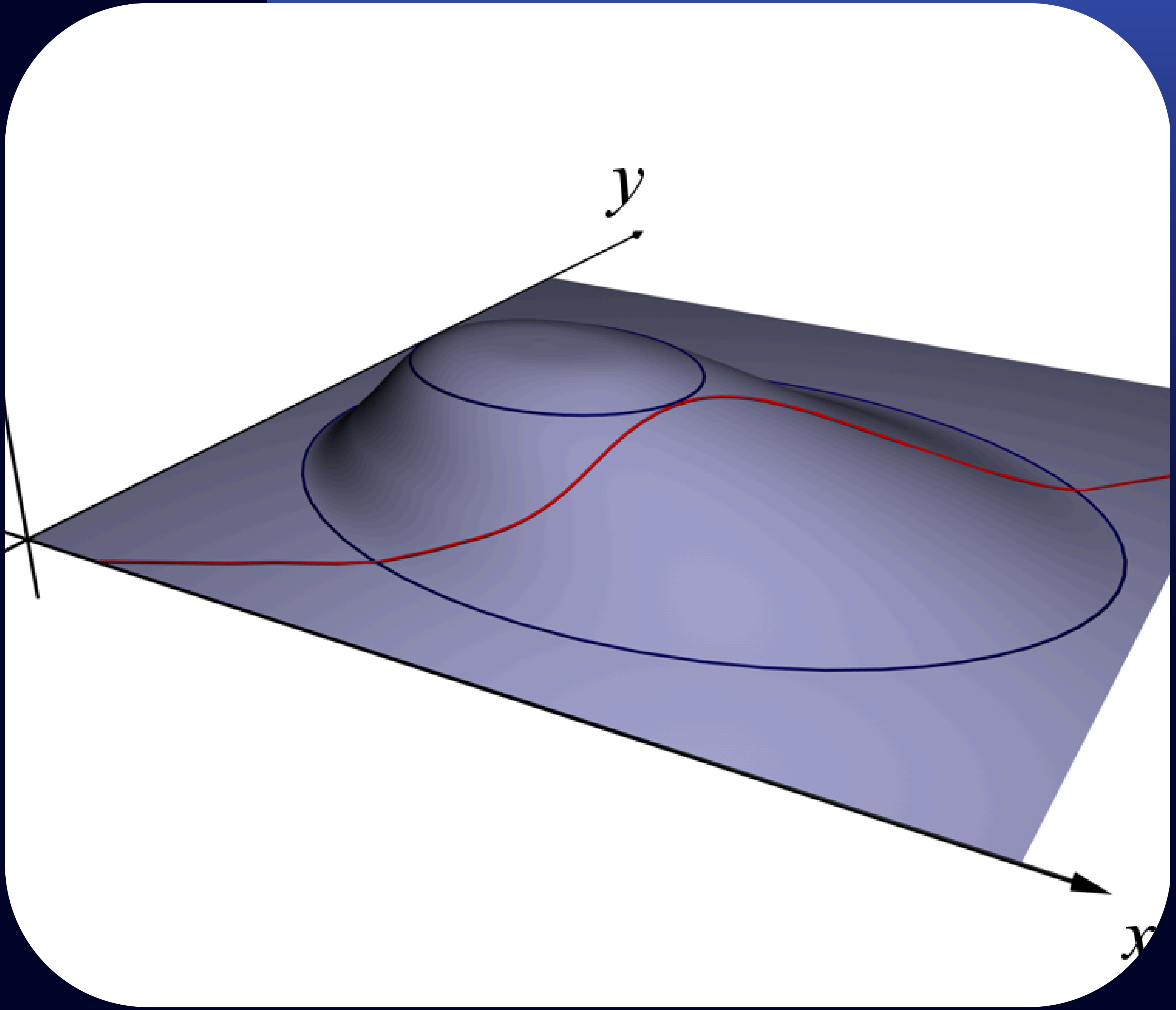
Demostracion cuando tenemos Xi Parametros

- Reemplazamos w en de la funcion de Lagrange:

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(w \cdot x_i + b) - 1\right]$$

entonces

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{N} \alpha_i \; - \; \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \, \alpha_j \, y_i \, y_j \, (x_i \cdot x_j)$$




MÁQUINAS DE SOPORTE VECTORIAL: ¡explicación COMPLETA!
Share
¿escalador?
¿embalador?
Watch on YouTube

Ejemplo Practico

X_1	X_2	**Clasificación**
1	2	A
2	3	A
3	3	B
6	5	B
7	8	B

Suponiendo que hemos encontrado los siguientes valores después del entrenamiento:

$\mathbf{w_1=1}$

$\mathbf{w_2=1}$

$\mathbf{b=-4}$

¿El hiperplano es?

$$\mathbf{x_1+x_2-4=0}$$

Para el punto nuevo **(4,3)**

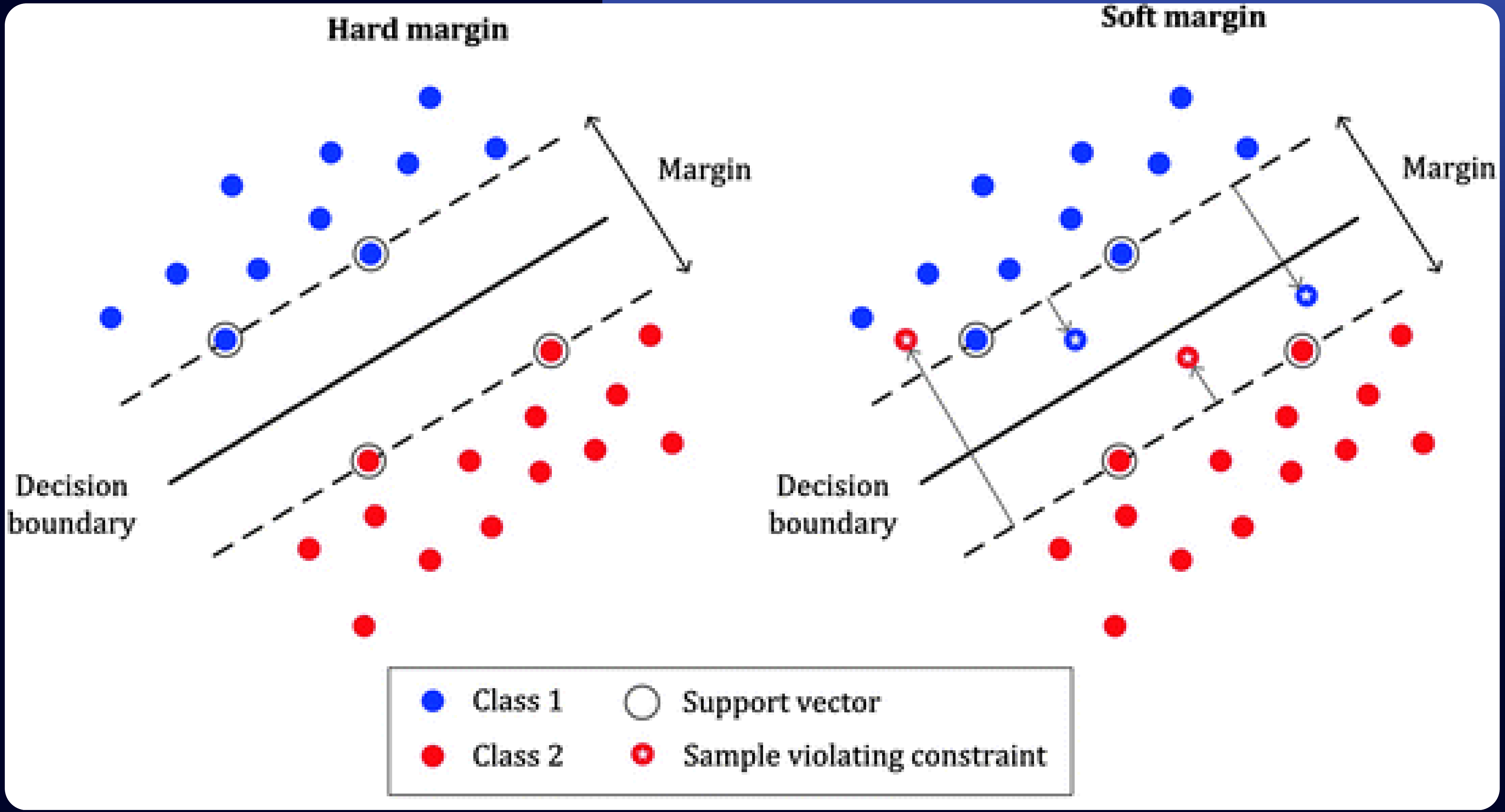
Clase$=sign(\mathbf{1\times4+1\times3-4})=sign(\mathbf{3})=\mathbf{1}$ -> **Clase B**

SVM - Margen Blando

El margen blando permite a la SVM cometer algunos errores de clasificación a cambio de lograr una mejor generalización del modelo.

En margen blando, se permiten:

1. Puntos dentro del margen (penalizados).
2. Algunos puntos mal clasificados (también penalizados).



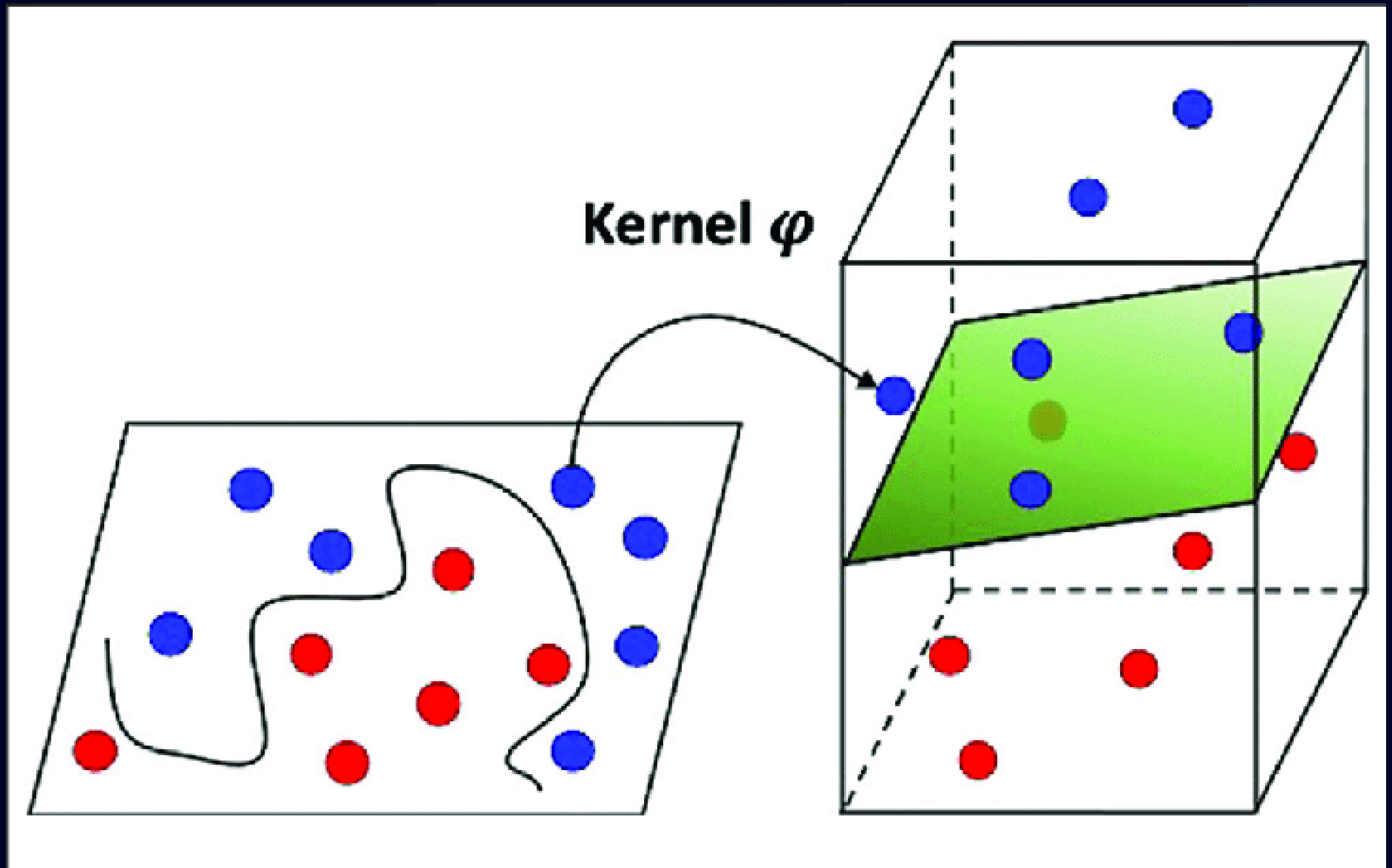
Esto se logra introduciendo variables de holgura ξ_i (x_i) para cada punto mal clasificado o dentro del margen.

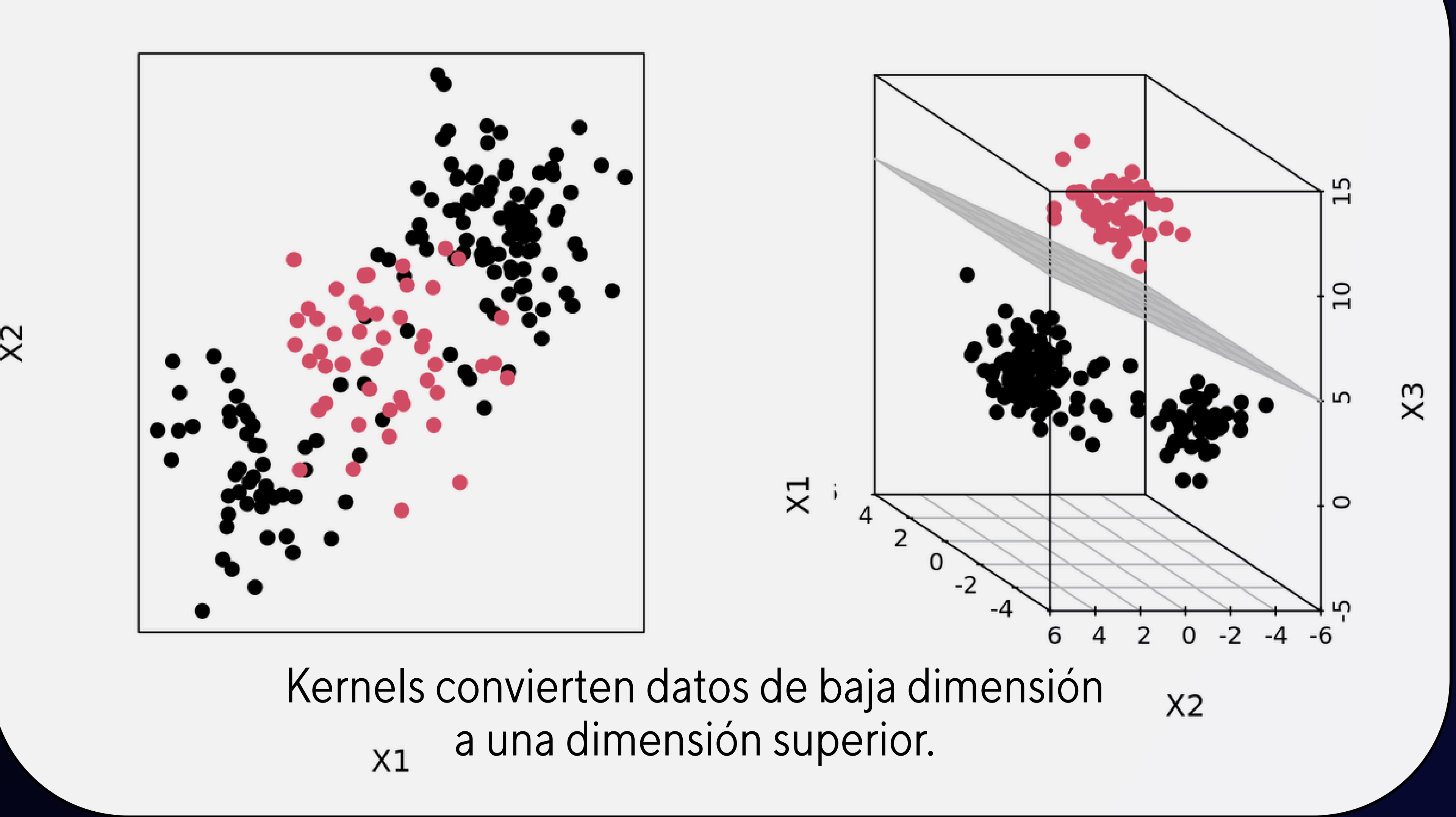
Aumento de la dimensión

La dimensión de un conjunto de datos puede transformarse combinando o modificando cualquiera de sus dimensiones. Por ejemplo, se puede transformar un espacio de dos dimensiones en uno de tres aplicando la siguiente función:

$$f(x_1, x_2) = \left({x_1}^2, \sqrt{2x_1, x_2}, {x_2}^2\right)$$

Esta es solo una de las infinitas trasformaciones posibles, ¿Cómo saber cuál es la adecuada? Es aquí donde los kernel entran en juego. Un kernel (K) es una función que devuelve el resultado del dot product entre dos vectores realizado en un nuevo espacio dimensional distinto al espacio original en el que se encuentran los vectores. Aunque no se ha entrado en detalle en las fórmulas matemáticas empleadas para resolver el problema de optimización, esta contiene un dot product.





Kernel Trick y Transformación de Espacios

Muchos problemas de clasificación no son lineales. El kernel trick permite transformar datos no lineales en espacios lineales, donde SVM puede encontrar un hiperplano óptimo.

1. **Kernel Lineal**

   Para datos separables

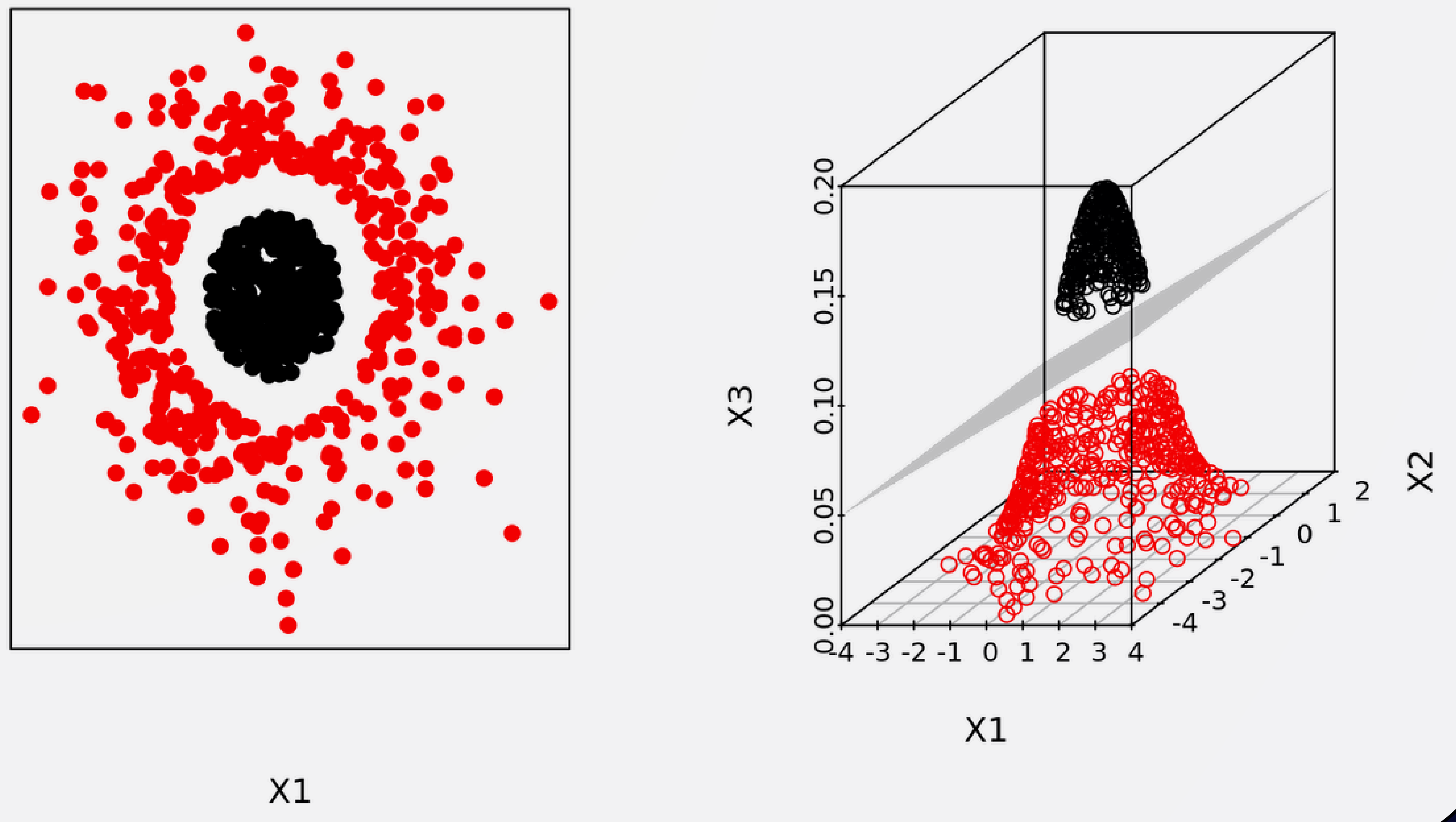
2. **Kernel Polinomial**

   Permite curvas en la separación

3. **Kernel Radial Basis Function (RBF)**

   Popular para datos no lineales

Kernels convierten datos de baja dimensión
a una dimensión superior.



Kernel Trick y Transformación de Espacios

Imagina un conjunto de datos donde los puntos de una clase forman un círculo rodeado por otra clase. En este caso, ninguna línea recta podrá separar ambas clases en el espacio original.

1. **Kernel Lineal**

   La función del kernel es simplemente:

   $$K(x_i, x_j) = x_i \cdot x_j$$

2. **Kernel Polinomial**

   Función del kernel:

   $$K(x_i, x_j) = (x_i \cdot x_j + c)^d$$

   El grado d define la complejidad del modelo (mayores valores permiten curvas más complejas).

3. **Kernel Radial Basis Function (RBF)**

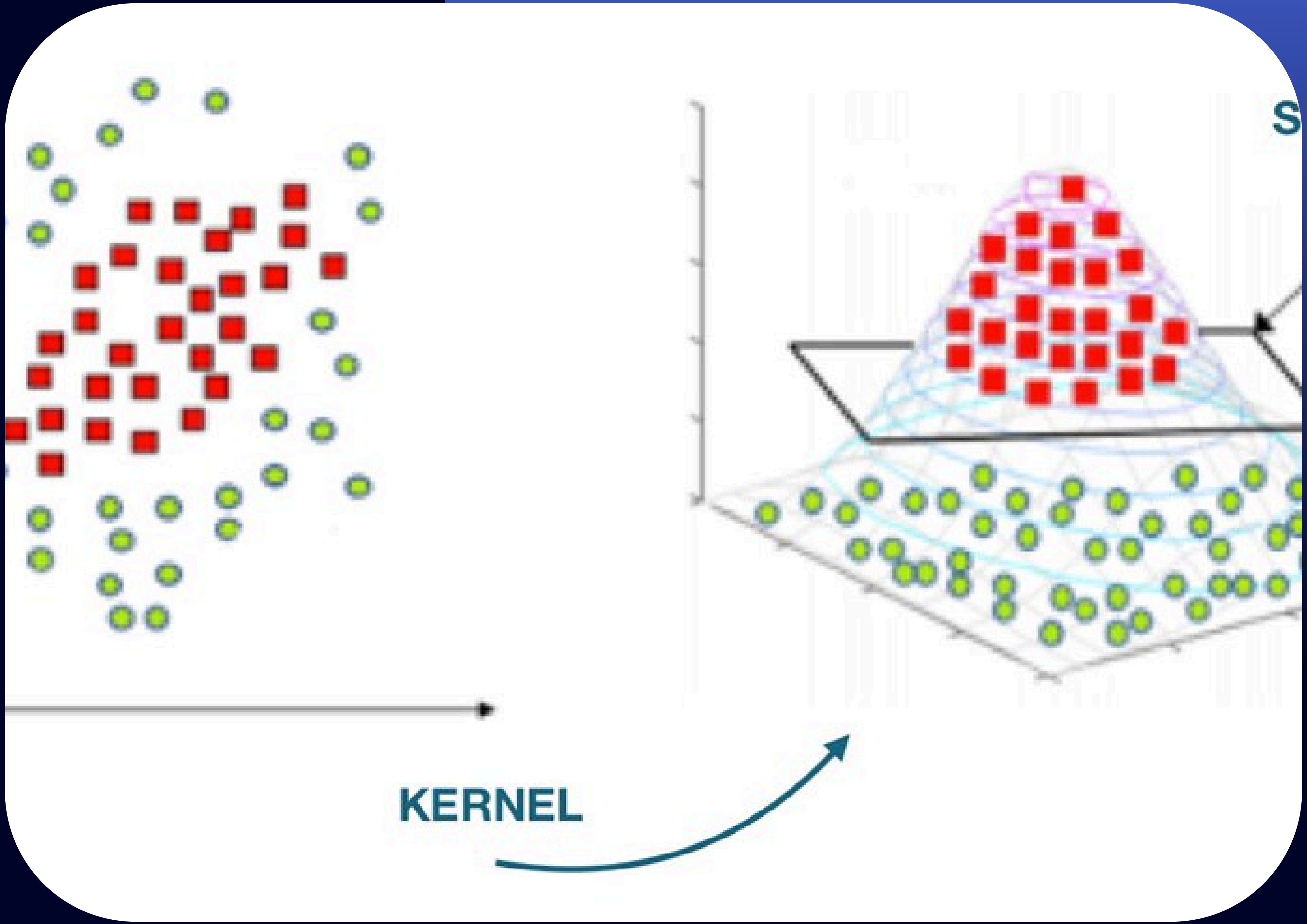
   Función del kernel:

   $$K\left(x_i, x_j\right) = exp(-\gamma ||x_i, x_j||^2)$$

Kernel Radial

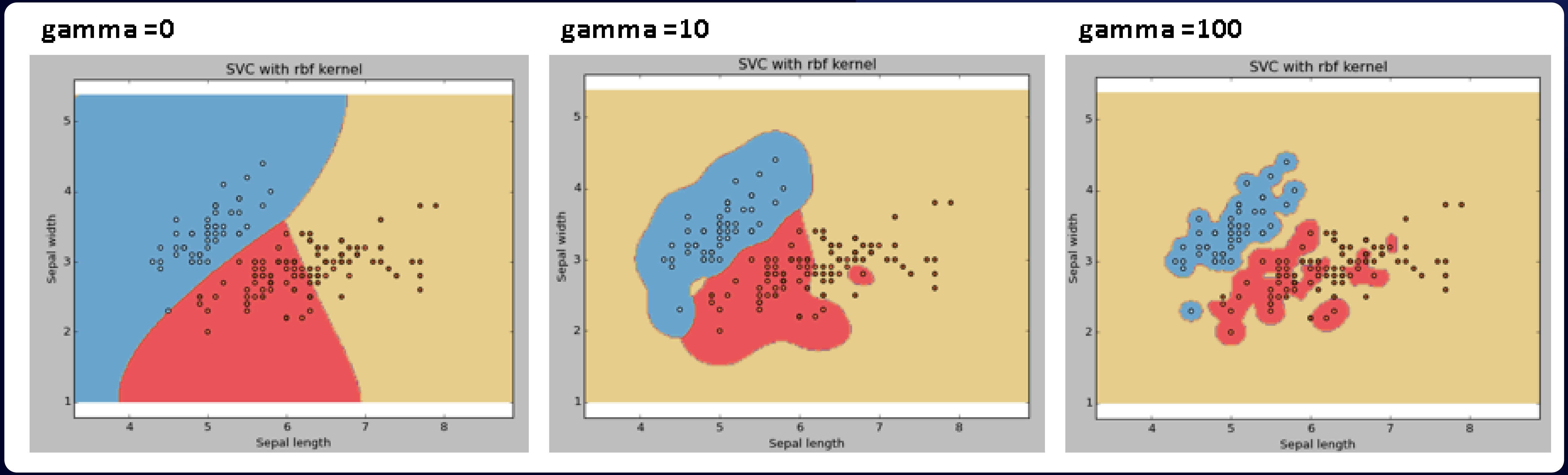
- El kernel radial, también conocido como RBF (Radial Basis Function), es el tipo de kernel más utilizado en máquinas de soporte vectorial (SVM) cuando el problema no es linealmente separable. Permite que el SVM encuentre fronteras de decisión no lineales al transformar los datos a un espacio de mayor dimensión.

$$K(x_i, x_j) = e^{-\gamma||x_i, x_j||^2}$$



Kernel Radial

- γ *(gamma)* define la "anchura" de la influencia de cada punto de entrenamiento.
  - γ grande: influencia muy local → puede sobreajustar (overfitting).
  - γ pequeño: influencia amplia → puede subajustar (underfitting).



Aplicación del Kernel a la Ecuación de Optimización

El lagrangiano original para SVM (con kernel lineal) es:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]$$

Al aplicar las condiciones de optimalidad, se reemplaza w · xi con el kernel K(xi,xj) es:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

Sujeto a:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C \quad \forall i$$