



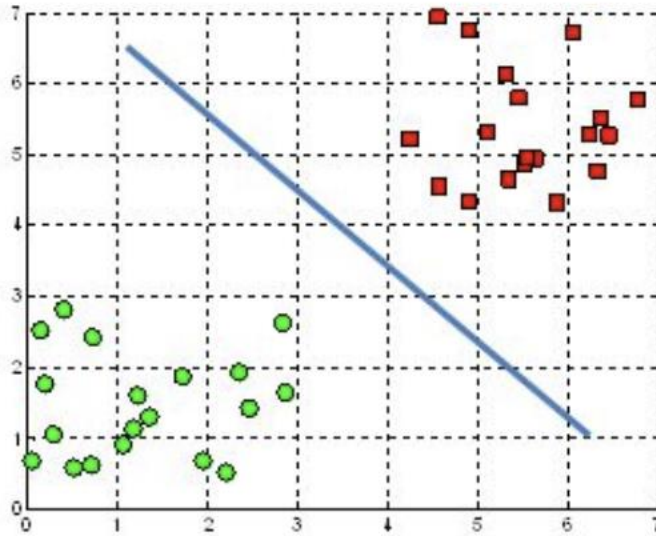
MACHINE LEARNING

Support Vector Machines

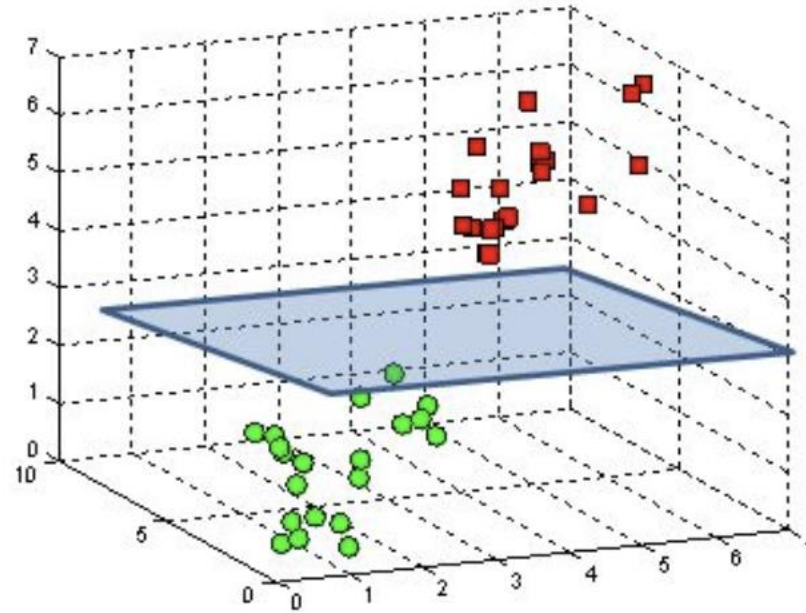
PhD. César Astudillo | Facultad de Ingeniería

Hyperplanes in 2D and 3D

A hyperplane in \mathbb{R}^2 is a line



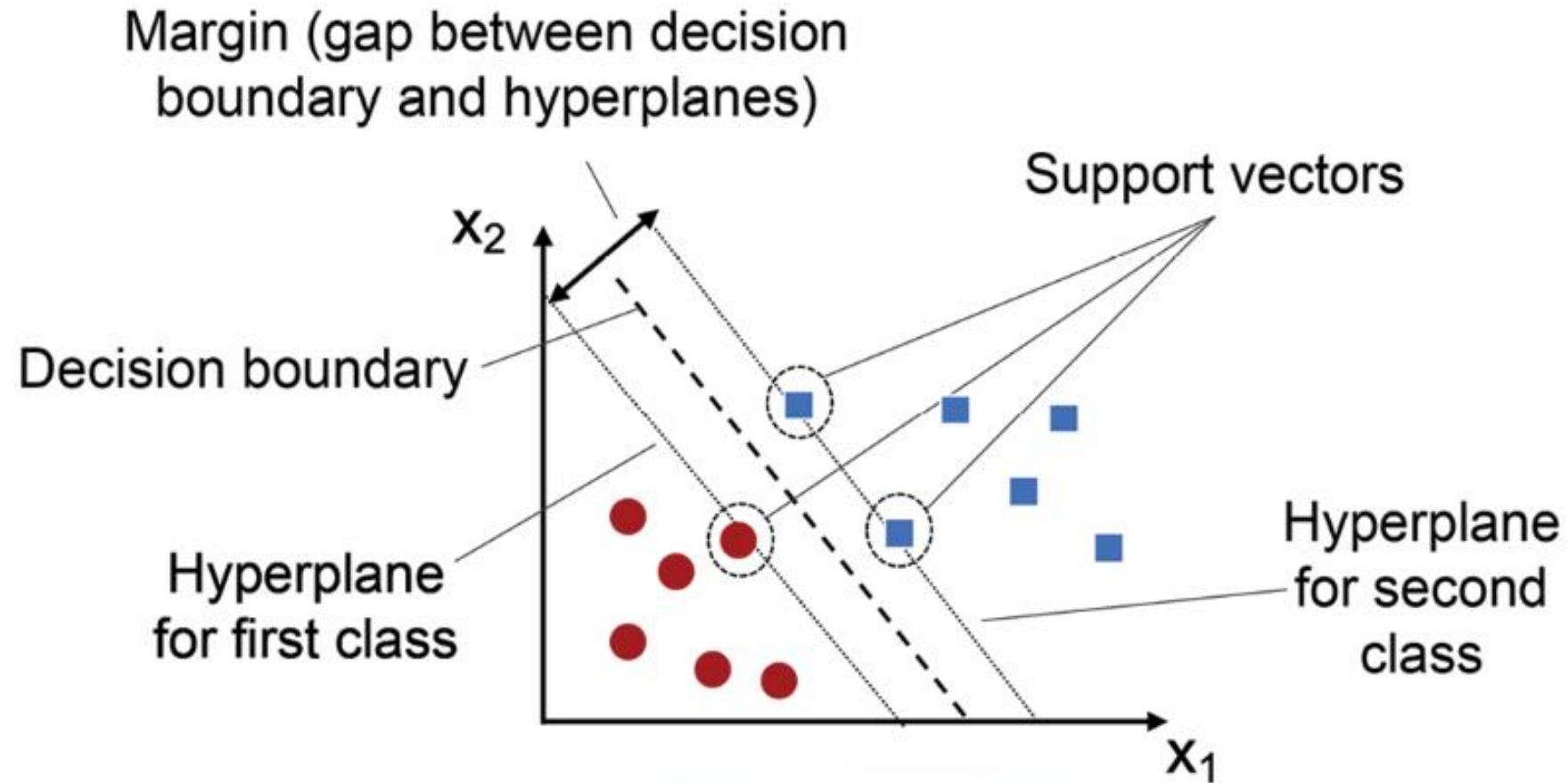
A hyperplane in \mathbb{R}^3 is a plane



¿Qué es un Hiperplano?

- Un **hiperplano** es una superficie de dimensión $d-1$ que divide un espacio de dimensión d en dos regiones.
- En un espacio bidimensional ($d=2$), es una línea recta que separa dos clases.
- En un espacio tridimensional ($d=3$), es un plano.

SVM



¿Cómo se Vinculan los Hiperplanos con SVM?

- En **SVM (Support Vector Machines)**, un **hiperplano** es la frontera de decisión que separa las clases en el espacio de características.
- El objetivo de SVM es encontrar el **hiperplano óptimo**, es decir, aquel que **maximiza el margen** entre las clases, lo que mejora la capacidad de generalización del modelo.
- Los **vectores de soporte** son los puntos de datos más cercanos al hiperplano. Estos puntos son cruciales, ya que definen la orientación y posición del hiperplano.
- Si los datos no son linealmente separables, SVM utiliza la técnica de **kernels** para transformar los datos a un espacio de mayor dimensión donde un hiperplano puede separarlos.

SVM en Problemas de Alta Dimensión

- SVM es efectivo en espacios con muchas variables (alta dimensionalidad).
- La búsqueda del hiperplano óptimo no se ve afectada negativamente por un número elevado de atributos.
- Ejemplo: Datos genéticos o imágenes con miles de píxeles como variables.

SVM en Problemas de Datos Complejos

- SVM puede manejar relaciones no lineales mediante el uso de kernels.
- Los kernels transforman los datos a un espacio de mayor dimensión donde se vuelven separables.
- Ejemplo: Detección de patrones en datos biométricos o análisis de texto con miles de palabras clave.

Kernel Trick

¿Qué es un Kernel?

- Un kernel es una función matemática que transforma los datos originales a un espacio de mayor dimensión donde las clases pueden ser separables.
- Utilizado cuando los datos no son linealmente separables en su espacio original.
- Evita la necesidad de realizar transformaciones manuales de los datos.
- Permiten encontrar un hiperplano óptimo incluso con datos complejos.

¿Qué es el Kernel Trick?

- **Definición:**

- El **Kernel Trick** (truco del kernel) es una técnica matemática utilizada en SVM para manejar datos no linealmente separables.
- Permite proyectar los datos a un espacio de mayor dimensión donde las clases puedan separarse con un hiperplano.

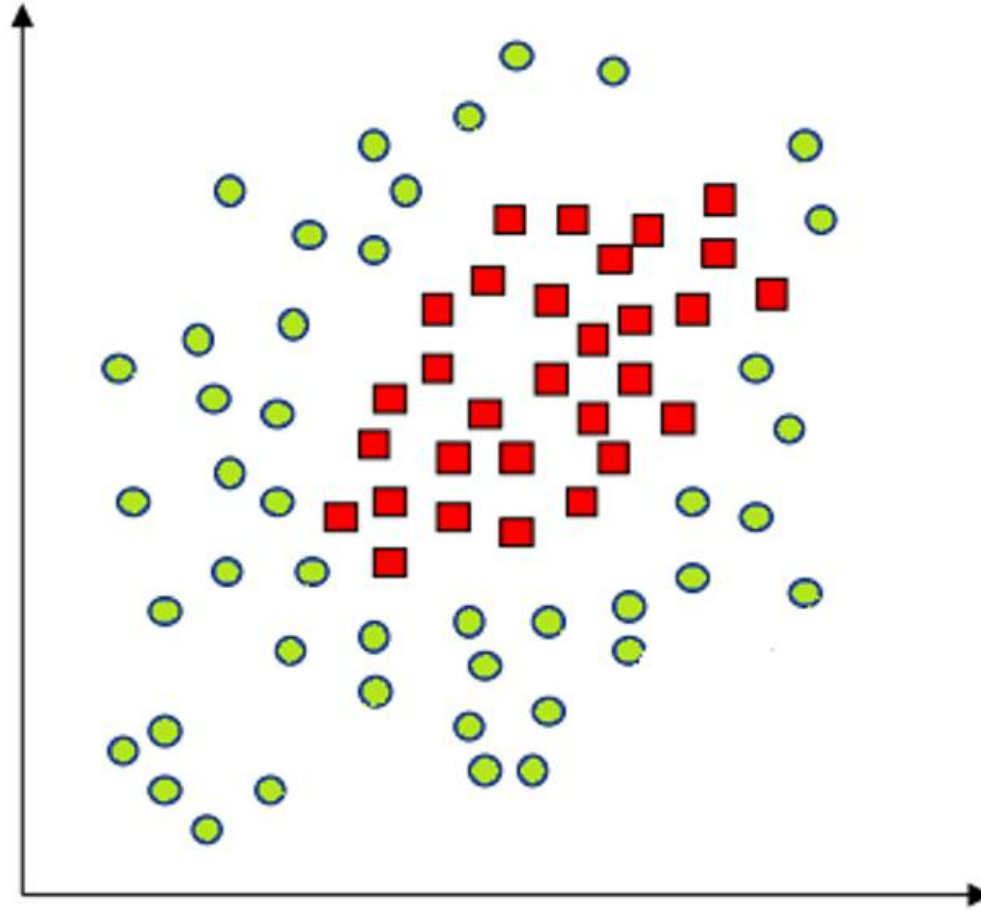
- **Idea Clave:**

- En lugar de transformar directamente los datos, el kernel calcula el producto escalar entre los datos en un espacio de alta dimensión **sin necesidad de transformar los datos explícitamente**.

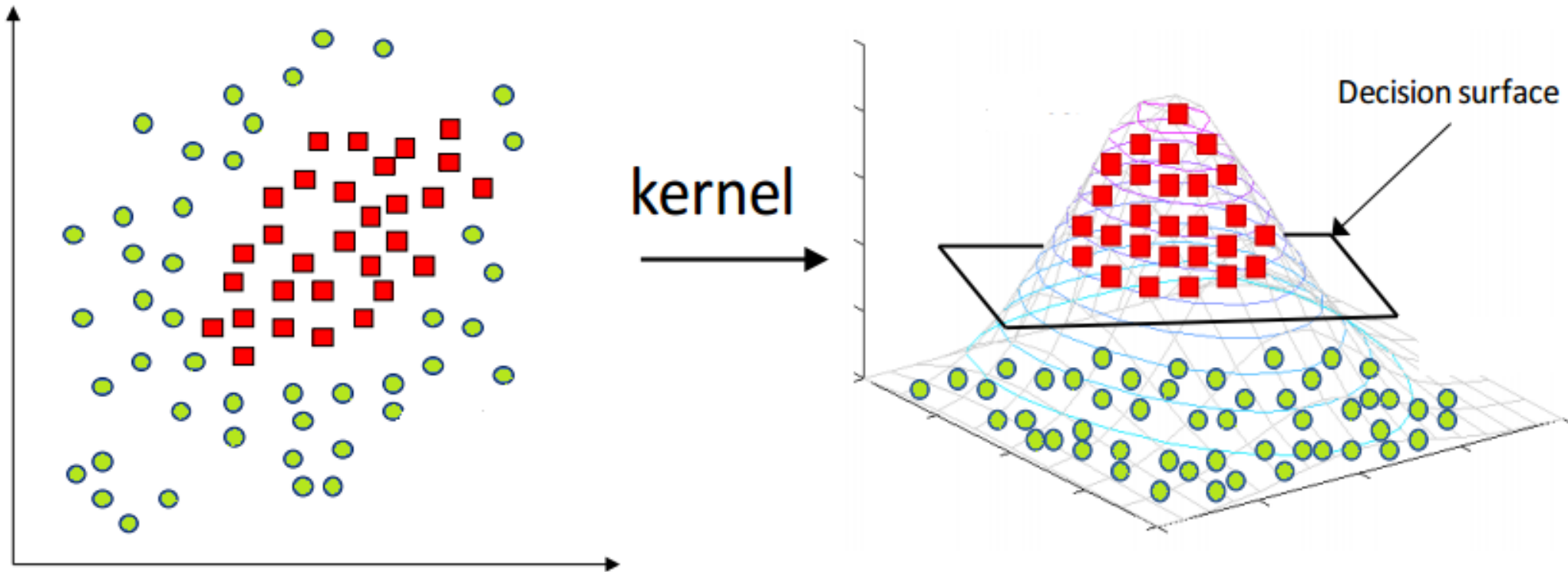
¿Cómo Funciona el Kernel Trick?

1. **Datos Originales:** Los puntos no son separables linealmente en el espacio original.
2. **Proyección Implícita:** Se utiliza una función kernel para mapear los datos a un espacio de mayor dimensión.
3. **Separación Lineal:** En este nuevo espacio, un hiperplano puede separar las clases.

Kernel trick: ejemplo



Kernel trick: ejemplo



Kernel Lineal

- **Definición:** Separa las clases con un hiperplano lineal.
- **Fórmula:** $K(x, y) = x \cdot y$
- **Aplicaciones:**
 - Datos con clases linealmente separables.
 - Problemas de texto y datos con alta dimensionalidad.
- **Ventaja:** Simple y eficiente en datos lineales.
- **Desventaja:** No funciona bien con datos no linealmente separables.

Kernel Polinomial

- Definición: Extiende la separación lineal utilizando funciones polinomiales.
- Fórmula: $K(x, y) = (x \cdot y + c)^d$.
- Aplicaciones: Datos con relaciones cuadráticas o cúbicas entre variables.
- Ventaja: Modela relaciones no lineales de bajo orden.
- Desventaja: Menos eficiente con datos de alta dimensionalidad.

Kernel RBF (Radial Basis Function) o Gaussiano

- Definición: Mapea los datos a un espacio infinito dimensional.
- Fórmula: $K(x, y) = \exp(-\gamma \|x-y\|^2)$.
- Aplicaciones: Problemas con datos altamente no lineales.
- Ventaja: Altamente flexible y efectivo para problemas complejos.
- Desventaja: Requiere ajuste cuidadoso del parámetro γ .

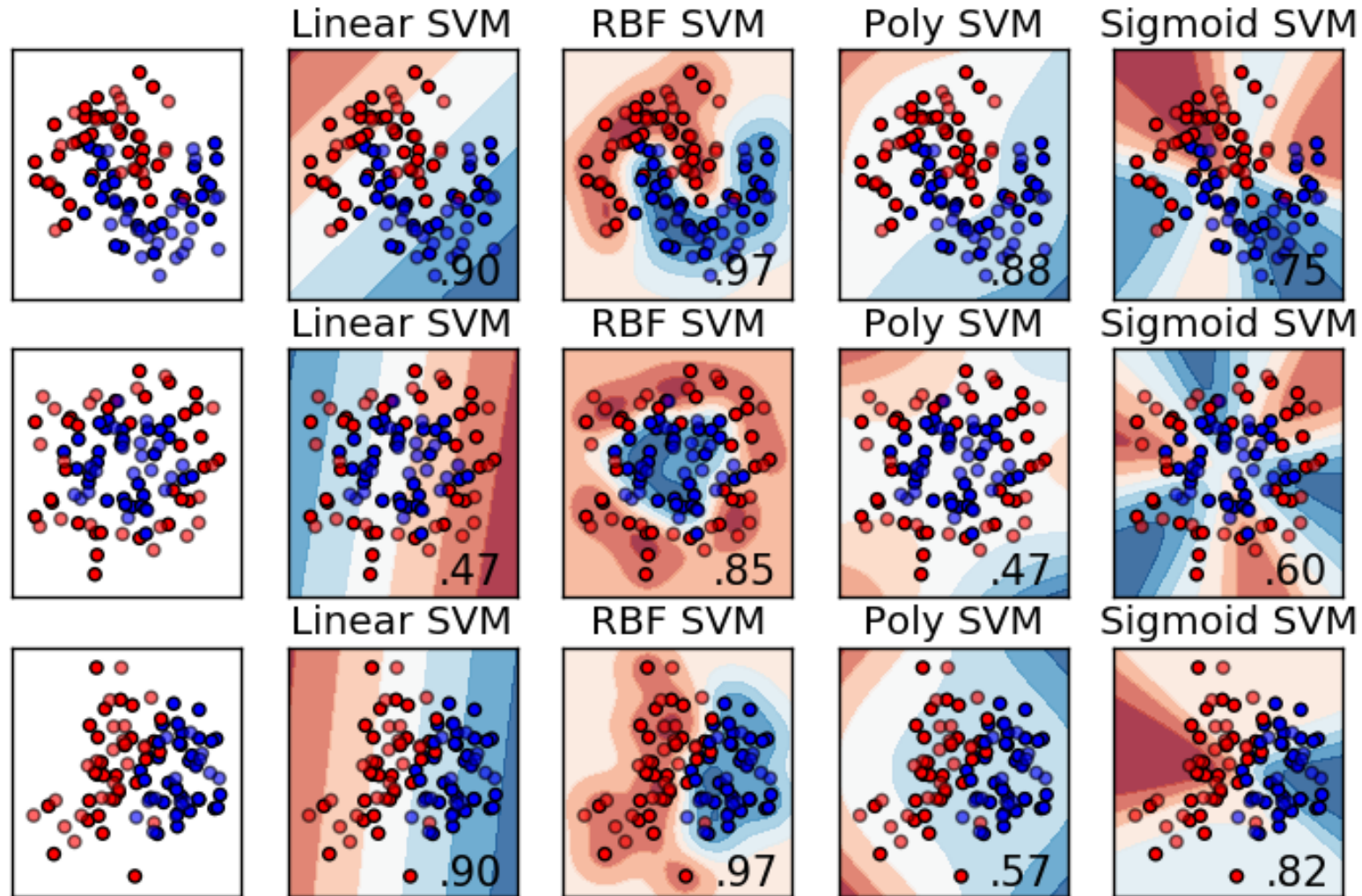
Kernel Sigmoide

- Definición: Basado en la función sigmoide, similar al comportamiento de una neurona en redes neuronales.
- Fórmula: $K(x, y) = \tanh(\alpha \cdot x \cdot y + c)$.
- Aplicaciones:
- Problemas de clasificación similares a redes neuronales.
- Ventaja: Adecuado para problemas complejos con fronteras no lineales.
- Desventaja: Menor uso en la práctica comparado con RBF.

Comparación General de los Kernels

- Lineal: Datos simples y lineales.
- Polinomial: Datos con relaciones no lineales simples.
- RBF: Datos complejos con fronteras curvas.
- Sigmoide: Problemas inspirados en redes neuronales.
- Importancia de la Normalización:
- Los datos deben ser normalizados para un mejor desempeño de los kernels.

Ejemplos de SVM usando diferentes Kernel



Conclusiones respecto a los Kernels

- Los kernels permiten que SVM se adapte a diferentes problemas de clasificación.
- La elección del kernel debe basarse en la naturaleza de los datos.
- Es importante realizar una validación cruzada para ajustar los hiperparámetros correctamente.

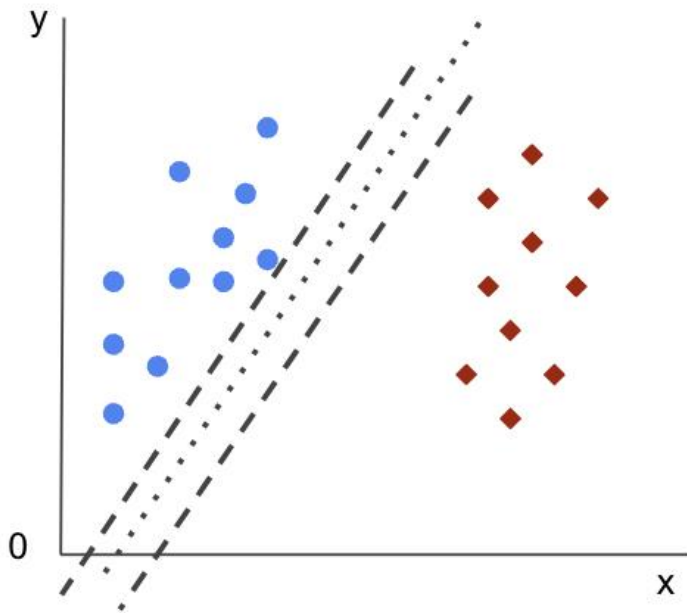
Parámetro C en SVM

¿Qué es el Parámetro C en SVM?

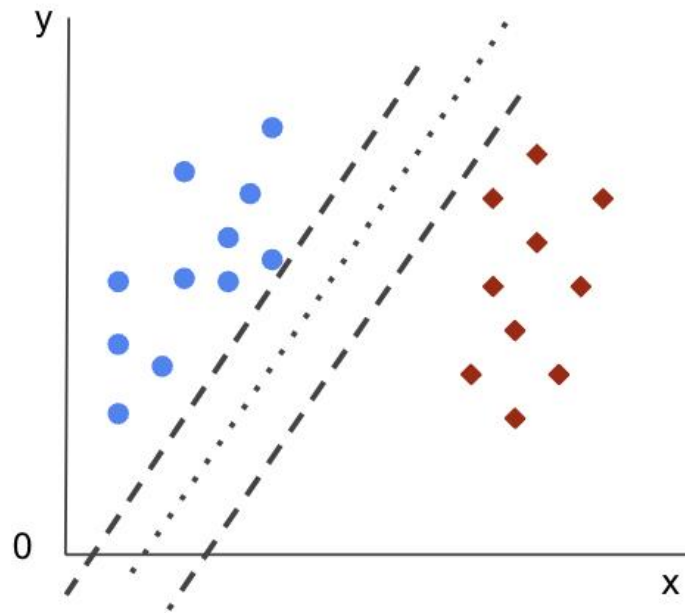
- El parámetro C controla la tolerancia al error en SVM.
- Es un hiperparámetro clave que ajusta el balance entre un margen grande y la correcta clasificación de los puntos.
- C afecta directamente la complejidad del modelo y su capacidad de generalización.

Ejemplo de diferentes valores para C

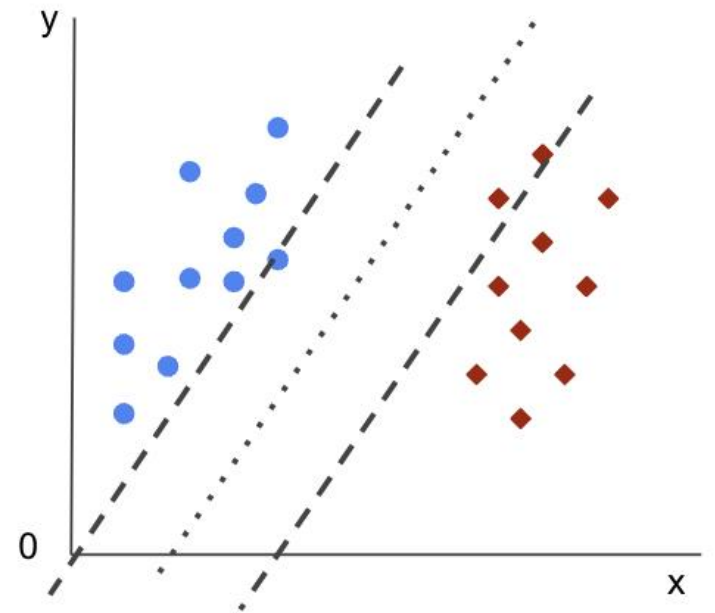
$C = 100$



$C = 10$



$C = 1$



Comparación: C Grande vs. C Pequeño

C Grande

- Penaliza más los errores de clasificación.
- Intenta clasificar todos los puntos correctamente.
- Resulta en un margen más estrecho.

C Pequeño

- Permite más errores de clasificación.
- Busca un margen más amplio.
- Mejora la capacidad de generalización, pero puede aumentar el error.

Efecto Visual del Parámetro C

- C Grande: Margen pequeño, más riesgo de sobreajuste.
- C Pequeño: Margen amplio, menos riesgo de sobreajuste pero más error en entrenamiento.
- La selección óptima de C depende del nivel de ruido y complejidad del dataset.

Cómo Seleccionar el Valor de C

- Usar validación cruzada para probar diferentes valores de C.
- Un valor medio suele ser ideal para datasets con ruido moderado.
- Evitar valores extremos sin probar múltiples opciones.
- Considerar el equilibrio entre sesgo y varianza.



MACHINE LEARNING

Support Vector Machines

PhD. César Astudillo | Facultad de Ingeniería