

# Módulo 2

## Procesado de Imagen y Visión por Computador

Grado en Ingeniería en Stmas. Telecomunicación



### Introducción al reconocimiento de patrones



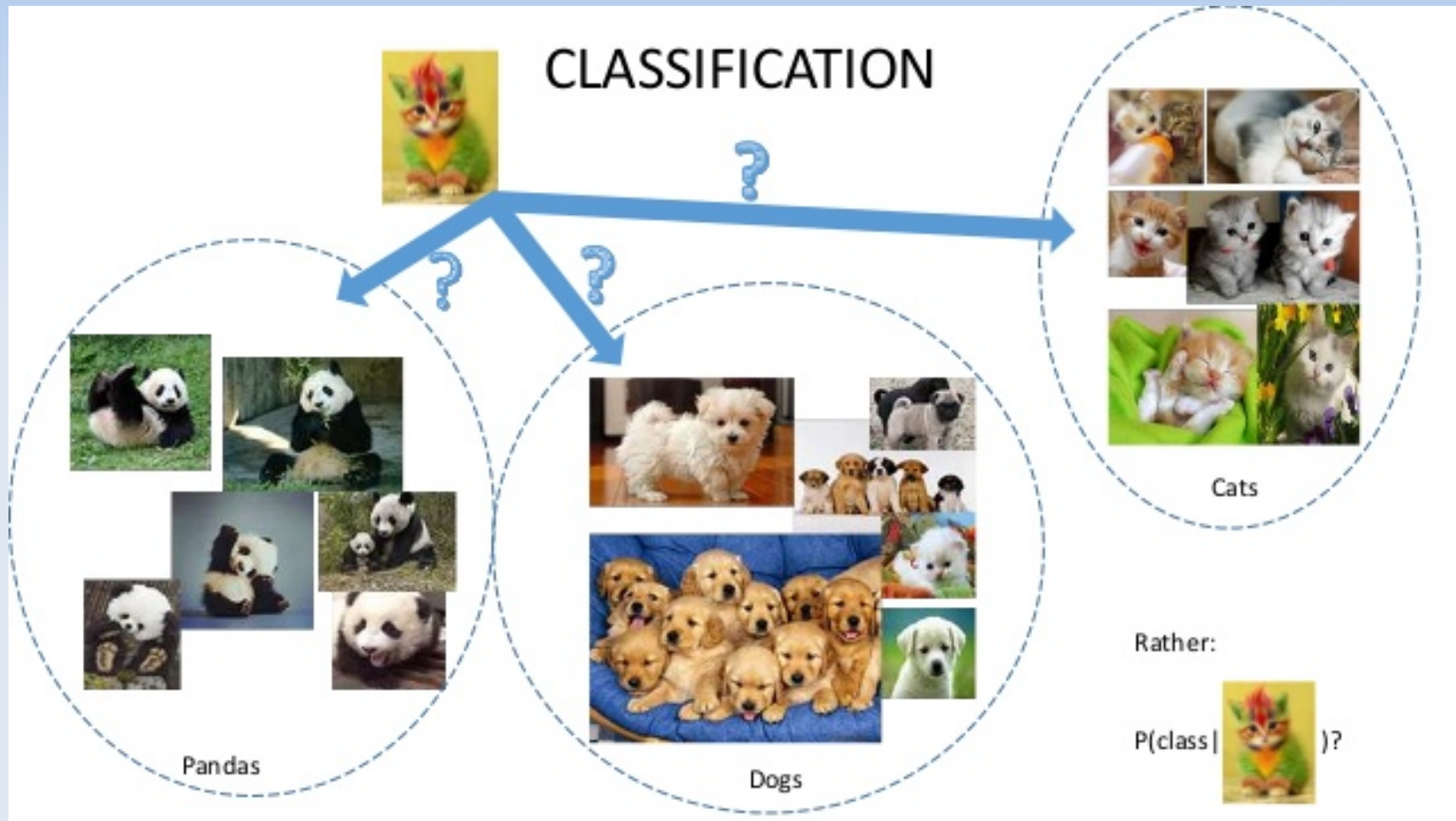
Universidad  
de Alcalá

# Reconocimiento de patrones

- ¿En qué consiste? En clasificar objetos en categorías o clases.
- Objetos se describen mediante vectores de atributos.
- Ejemplo con 3 clases y vectores de dos dimensiones.



# Reconocimiento de patrones



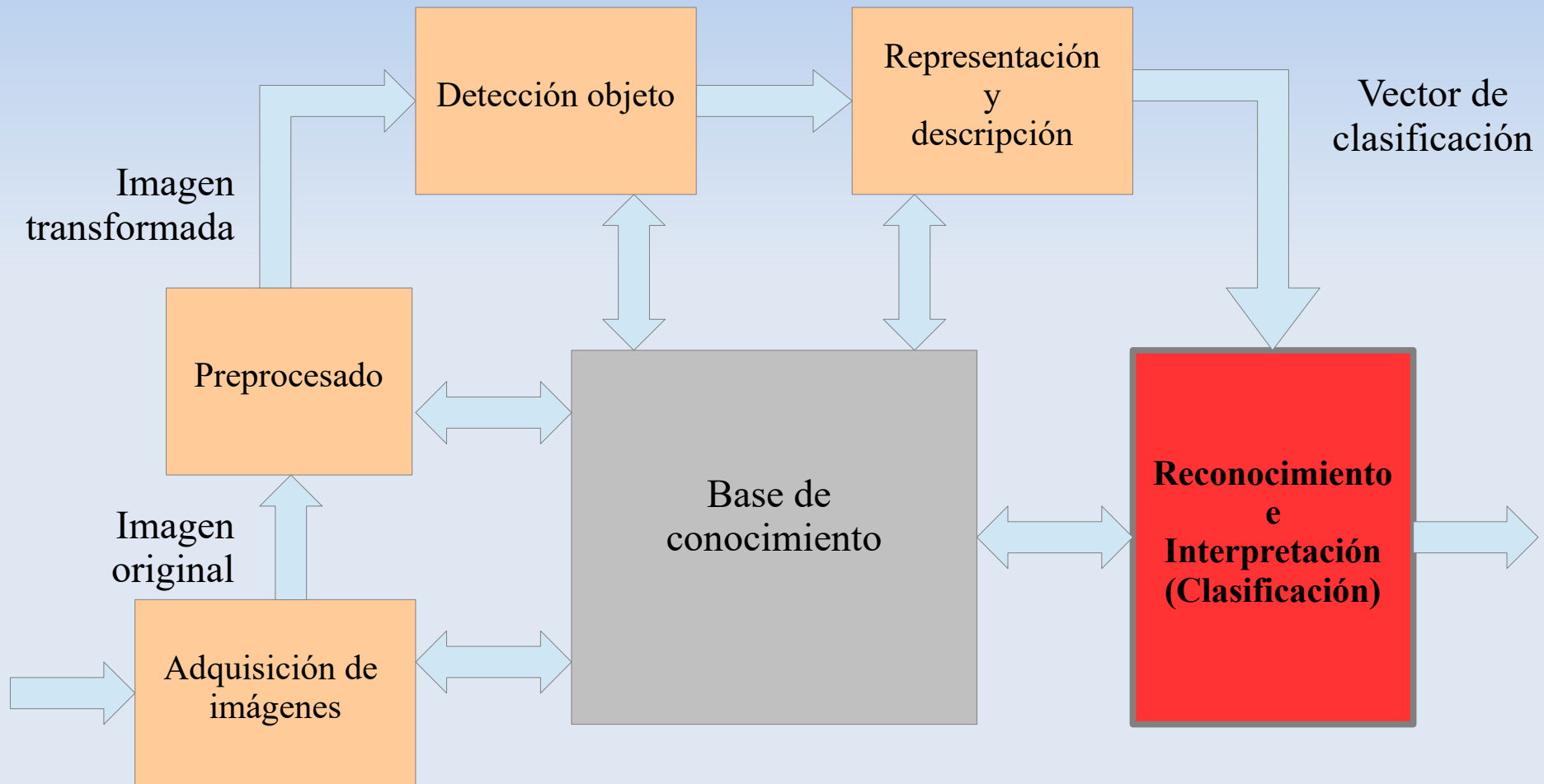
# Reconocimiento de patrones

- Los objetos (patrones) pueden ser de naturaleza variada:
  - Imágenes (visión artificial)
  - Audio (reconocimiento de voz)
  - Señales de sensores (ejemplo: ECG)
  - Variables económicas



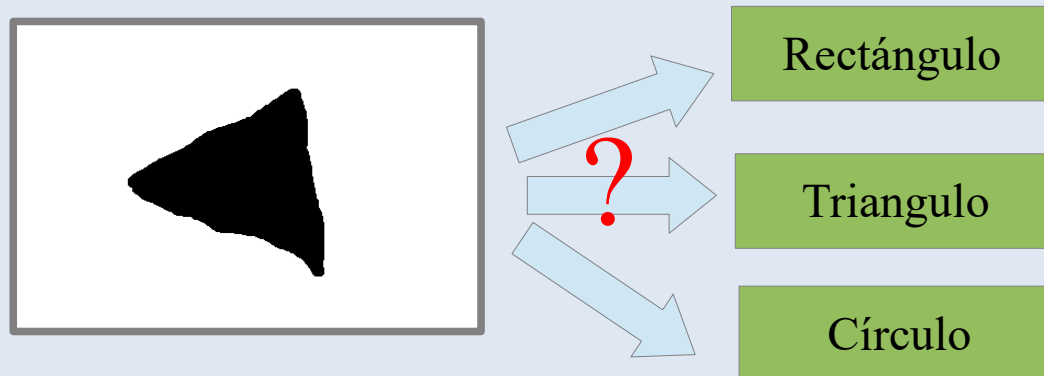
# Reconocimiento de patrones

- Esquema completo de visión artificial



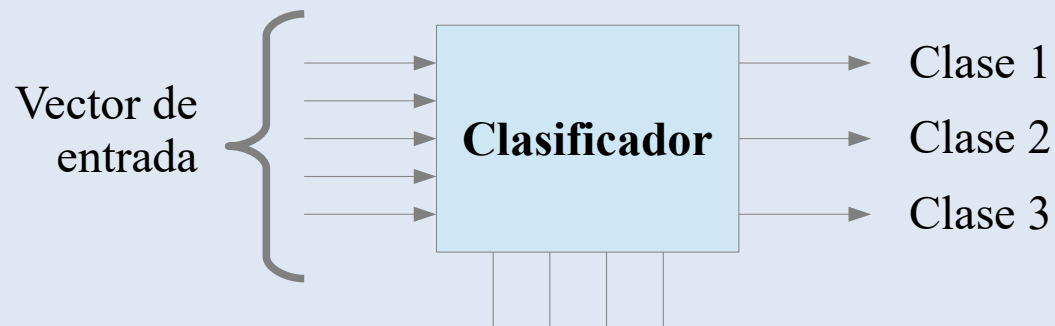
# Reconocimiento de patrones

- Objetivo:
  - Dado un vector de características (patrón) y un número de clases determinado.
  - Determinar la clase a la que pertenece.
  - Ejemplo: clasificar forma de la figura



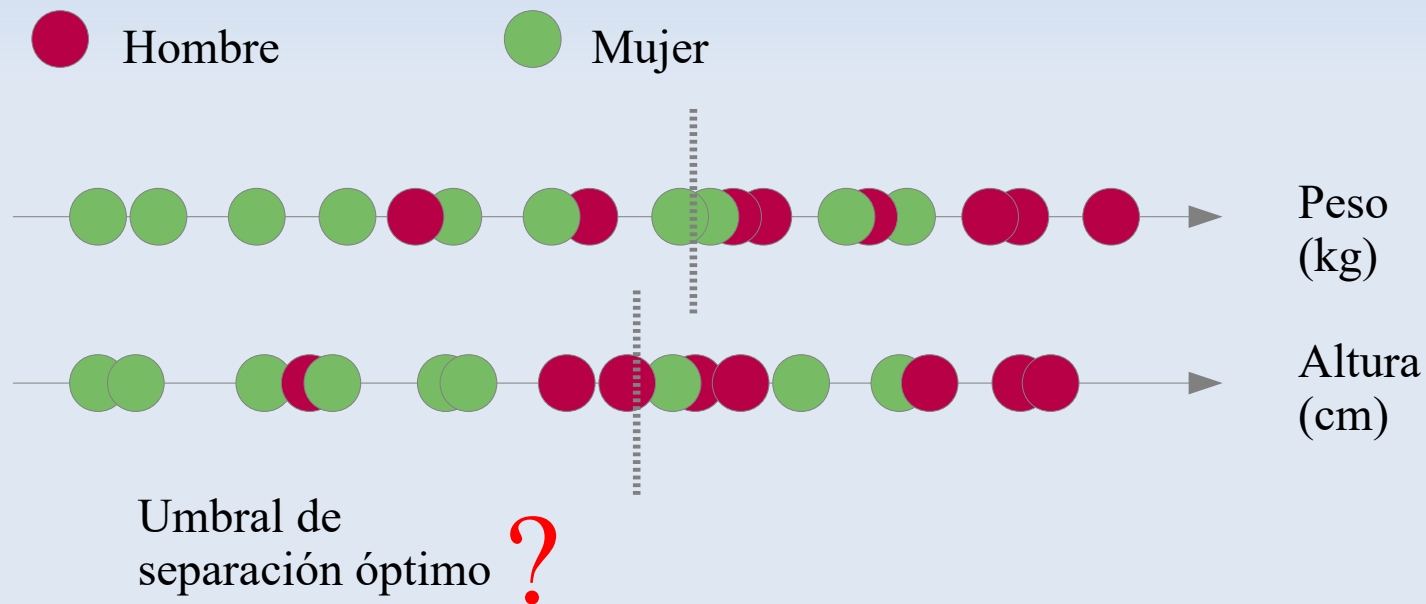
# Reconocimiento de patrones

- Estrategia:
  - Extraer características del vector (patrón) que mejor diferencian los vectores de una clase del resto.
  - Objetivo: ¿cuáles y cuántas? Características útiles e incorreladas.
  - En imágenes: características robustas a transformaciones (traslación, tamaño, orientación, ruido, etc.)



# Reconocimiento de patrones

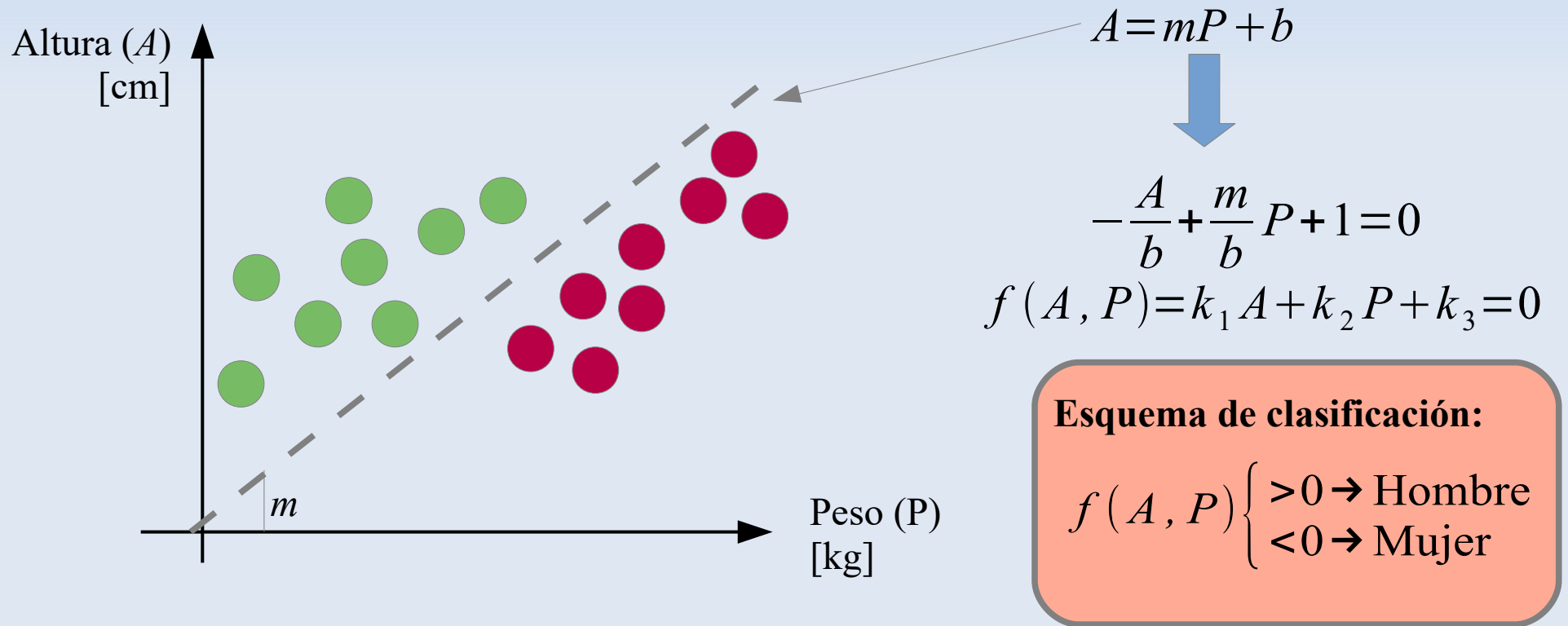
- Vector de características n-dimensional:
  - Ejemplo: clasificación hombres-mujeres a partir de peso y estatura.





# Reconocimiento de patrones

- Representación en dos dimensiones:
  - Las características muestran cierta correlación



# Reconocimiento de patrones

- Frontera de decisión:

$$f(A, P) = k_1 A + k_2 P + k_3 = 0$$

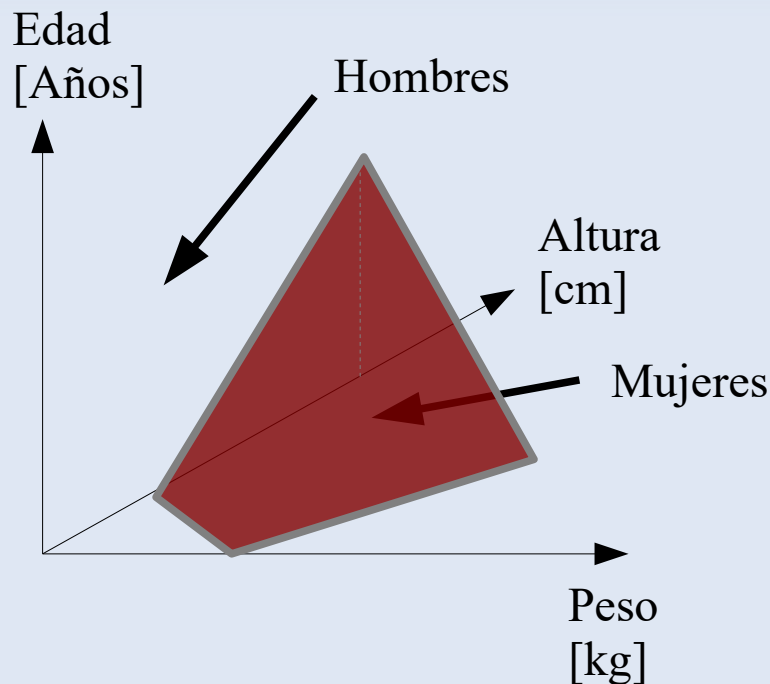


$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = 0$$

$$\left\{ \begin{array}{l} \text{Vector de características : } \mathbf{x} = [A, P]^T = [x_1, x_2]^T \\ \text{Vector de pesos : } \mathbf{w} = [k_1, k_2]^T = [w_1, w_2]^T \\ \text{Umbral : } w_0 \end{array} \right.$$

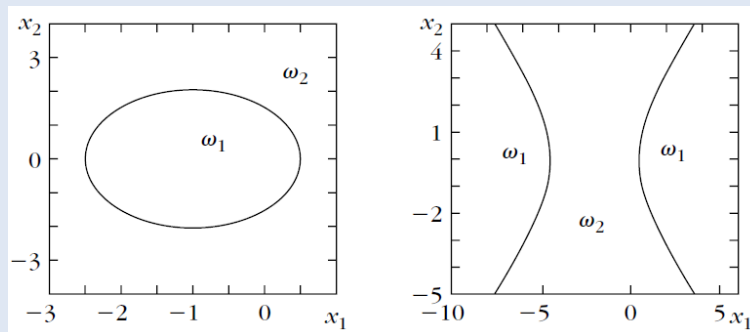
# Reconocimiento de patrones

- Posibilidad de añadir más características:
  - Añadir edad (vector de 3 dimensiones)
  - Añadir más características: (vector de n-dimensiones)



# Reconocimiento de patrones

- Frontera de decisión depende de la dimensionalidad del problema:
  - Punto: vector de 1 característica
  - Recta (curva): vector de 2 características
  - Plano (superficie): vector de 3 características
  - Hiperplano (hipersuperficie): vector de 4 o más características



Fuente: Pattern Recognition, S. Theodoridis, K. Koutroumbas. Elsevier.

# Reconocimiento de patrones

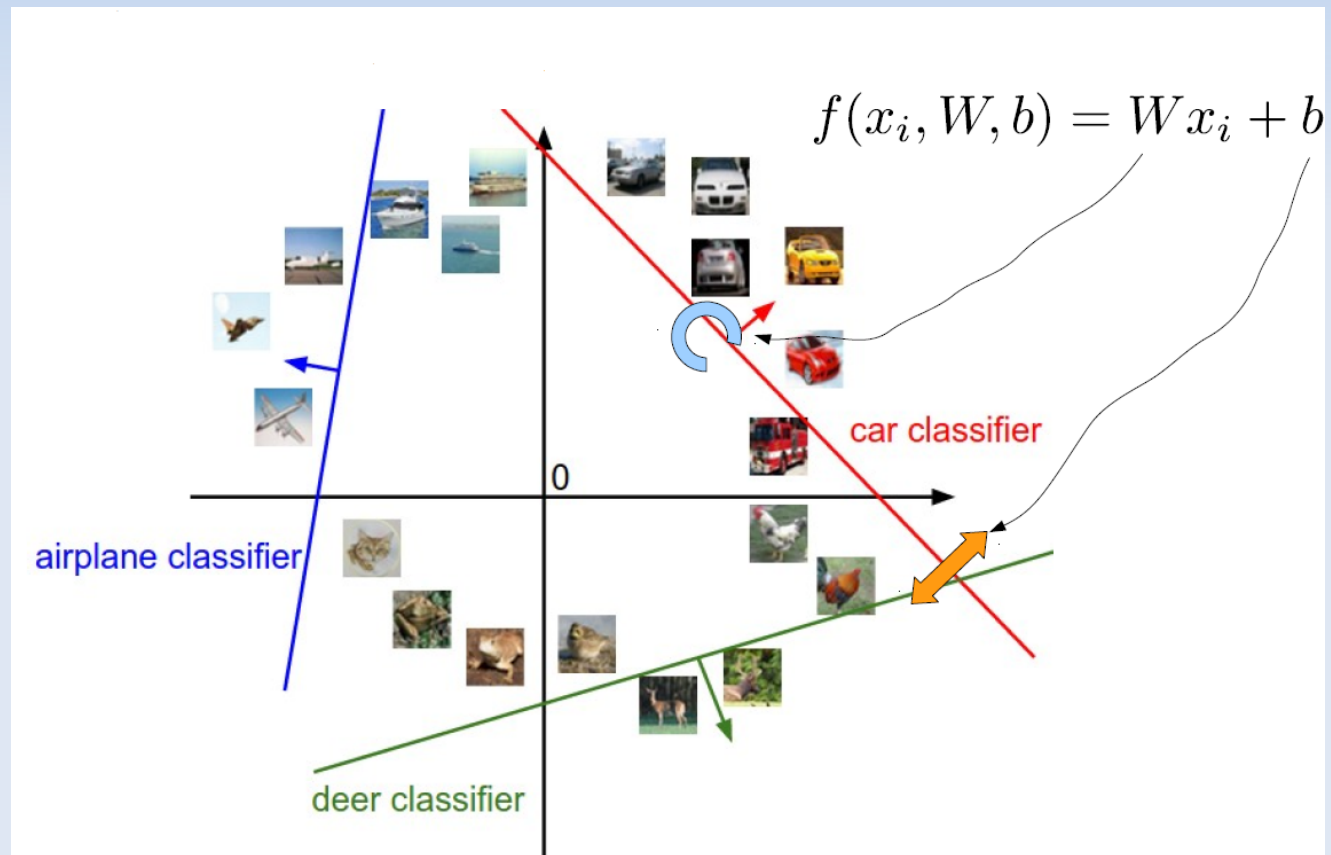
- Etapas:
  - **Entrenamiento:** búsqueda de la frontera de decisión entre clases
    - Se necesitan vectores (muestras) preclasificadas manualmente (aprendizaje supervisado).
  - **Test:** decidir clase de nuevas muestras diferentes a las de entrenamiento. Comprueba la fiabilidad del sistema.
  - **Funcionamiento normal:** una vez diseñado el clasificador.

Fuente: Pattern Recognition, S. Theodoridis.

Procesado de Imagen y Visión por Computador

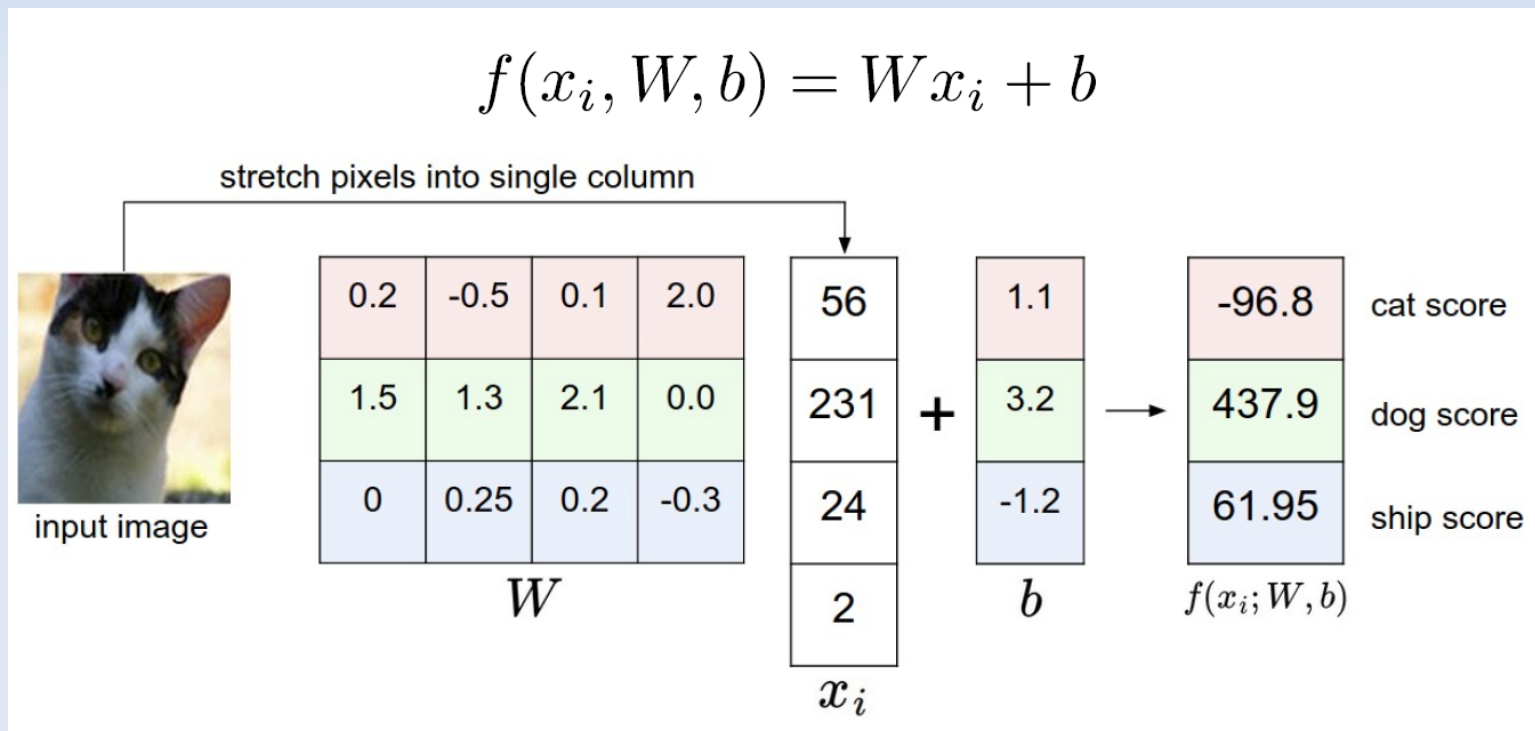
# Reconocimiento de patrones

- Imágenes pueden verse como puntos de alta dimensionalidad.

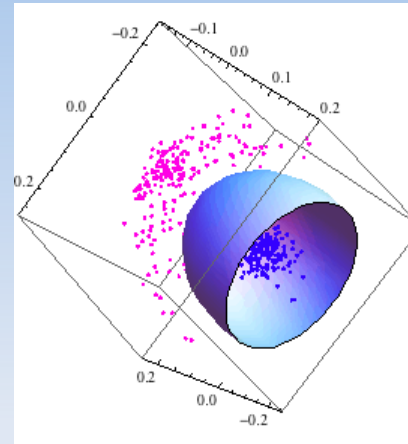
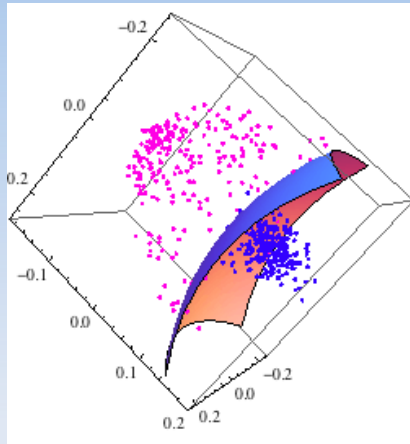


# Reconocimiento de patrones

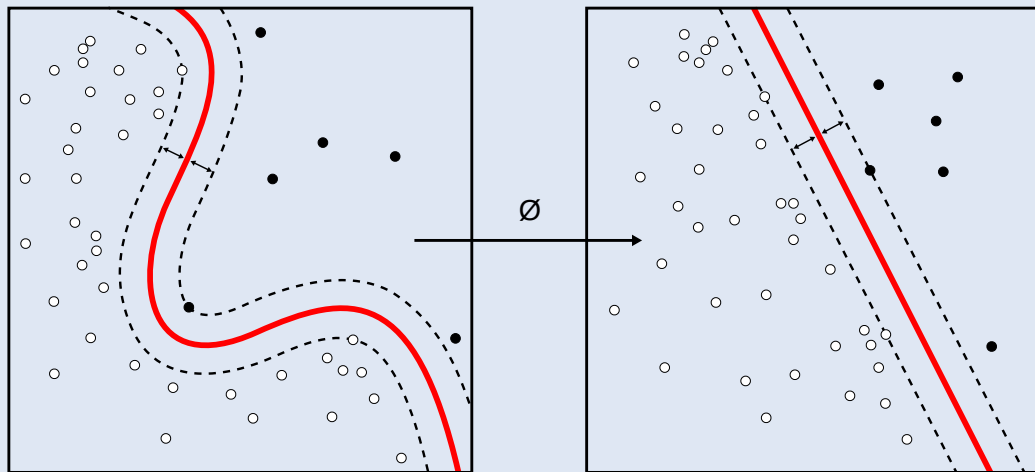
- Imagen puede ser codificada como un vector  $x_i$  concatenando sus píxeles.



# Reconocimiento de patrones



Fuente: <http://www.mathematica-journal.com/2011/07/fisher-discrimination-with-kernels/>

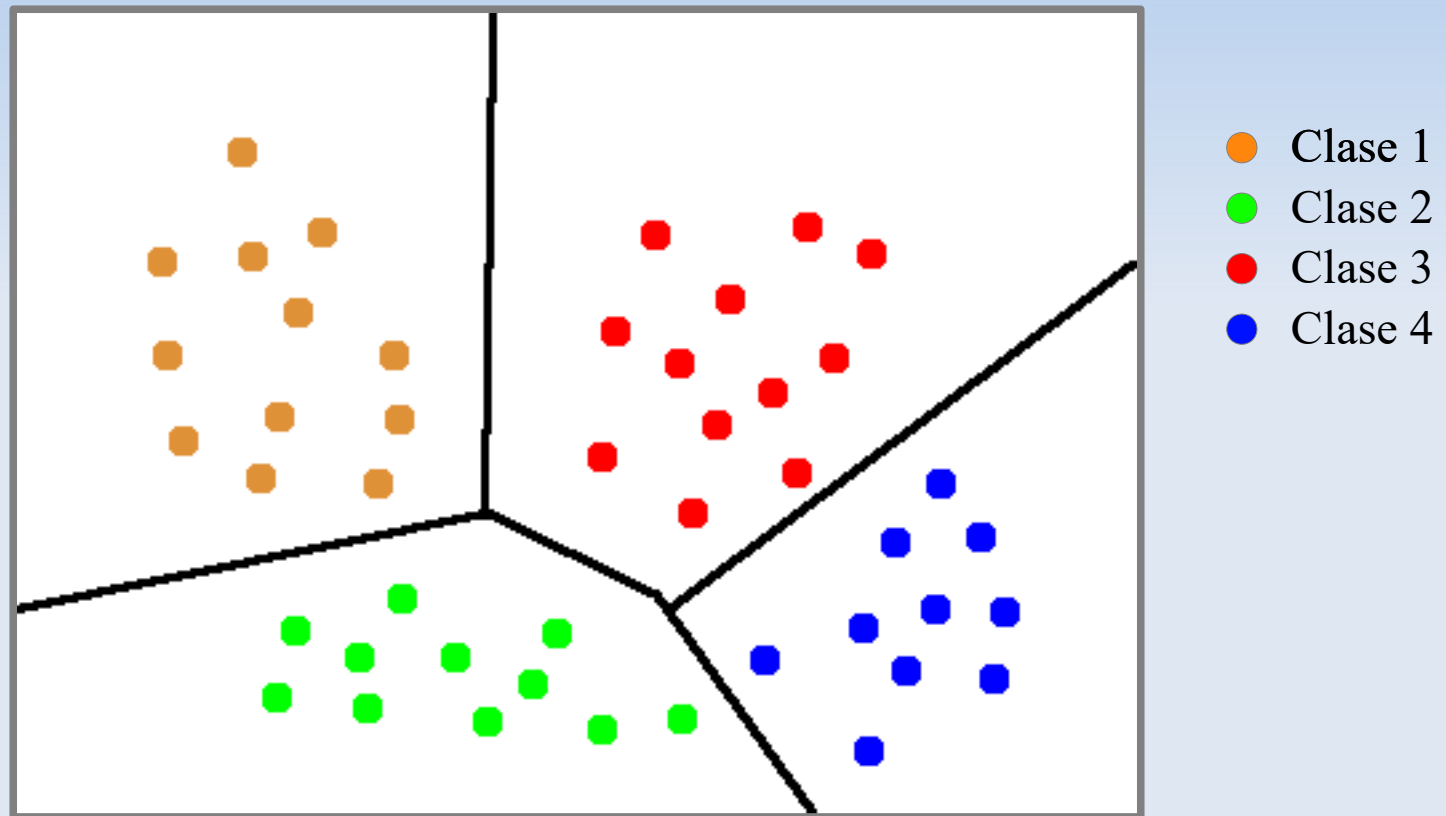


Fuente: [https://en.wikipedia.org/wiki/Support\\_vector\\_machine](https://en.wikipedia.org/wiki/Support_vector_machine)



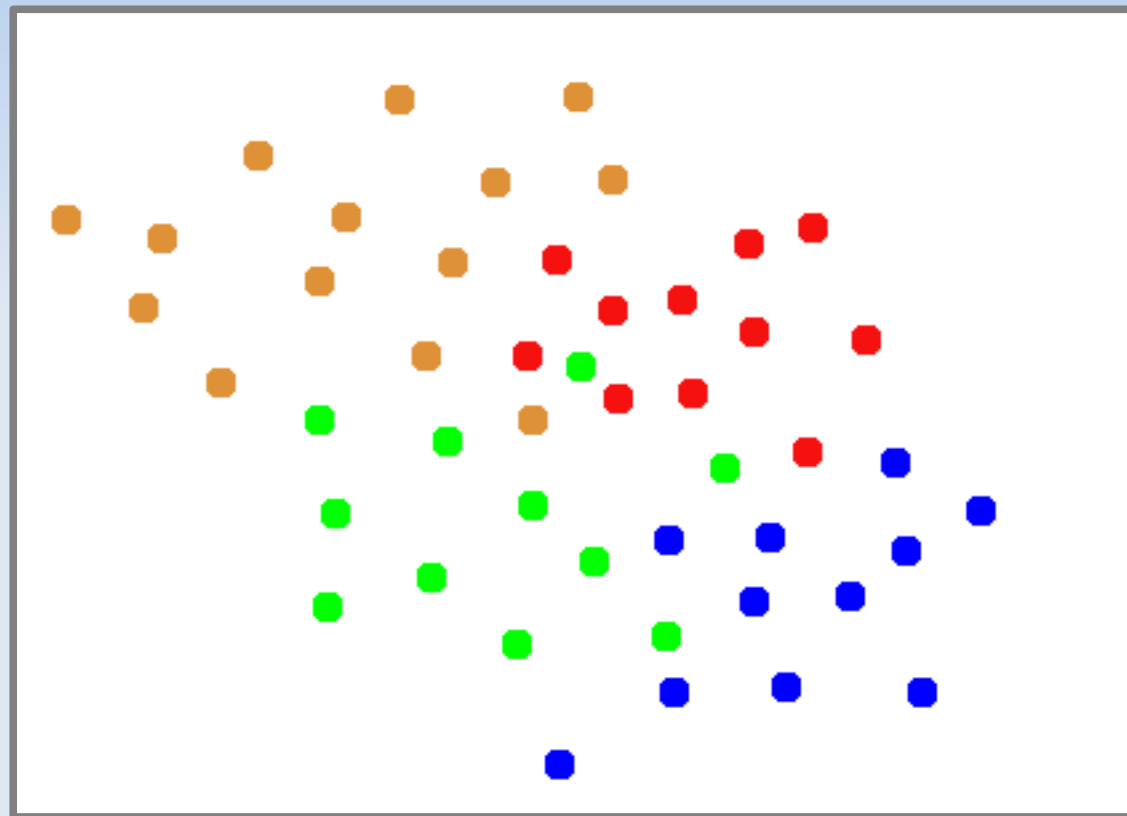
# Reconocimiento de patrones

- Muestras separables:



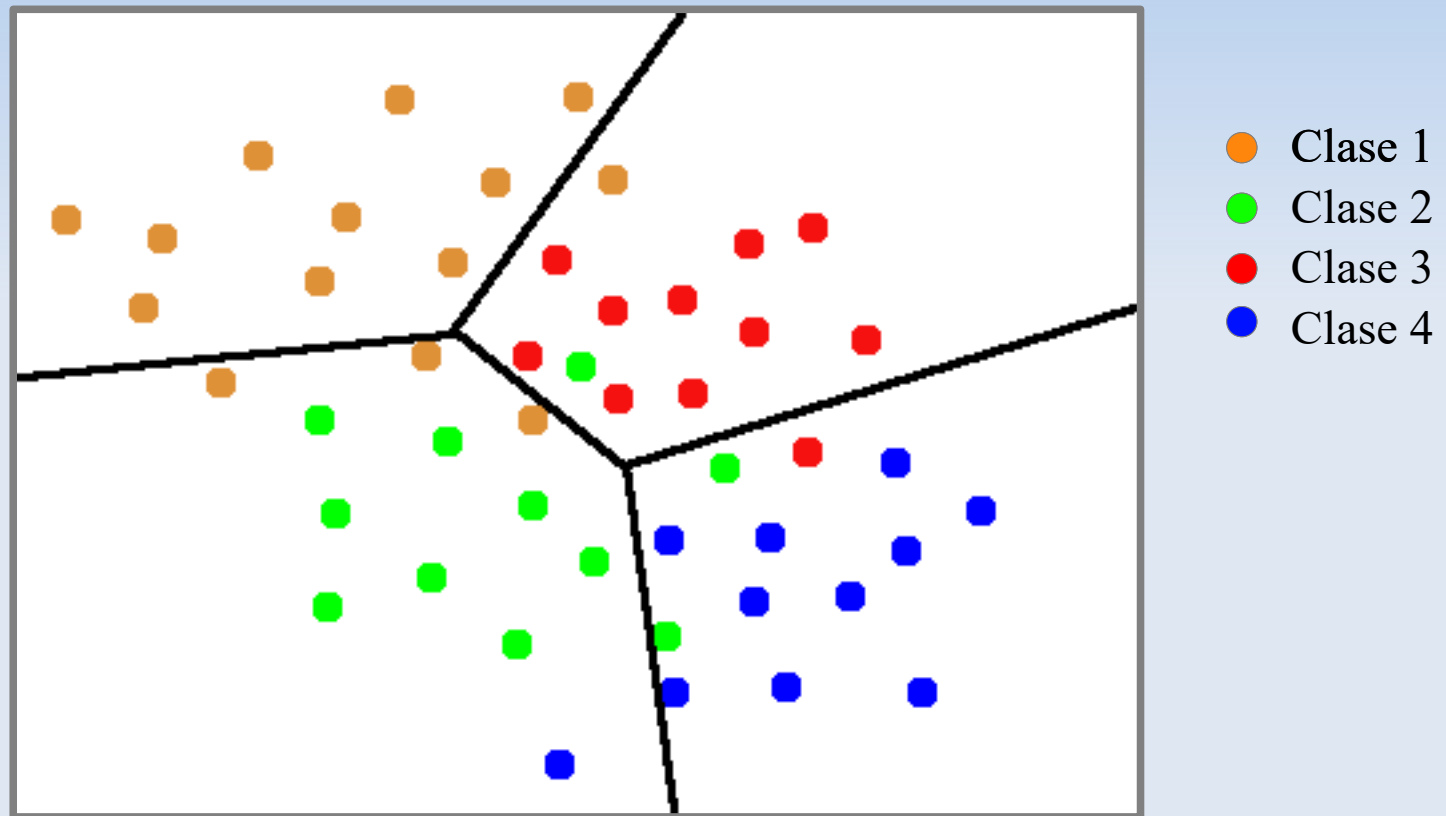
# Reconocimiento de patrones

- Muestras no separables:



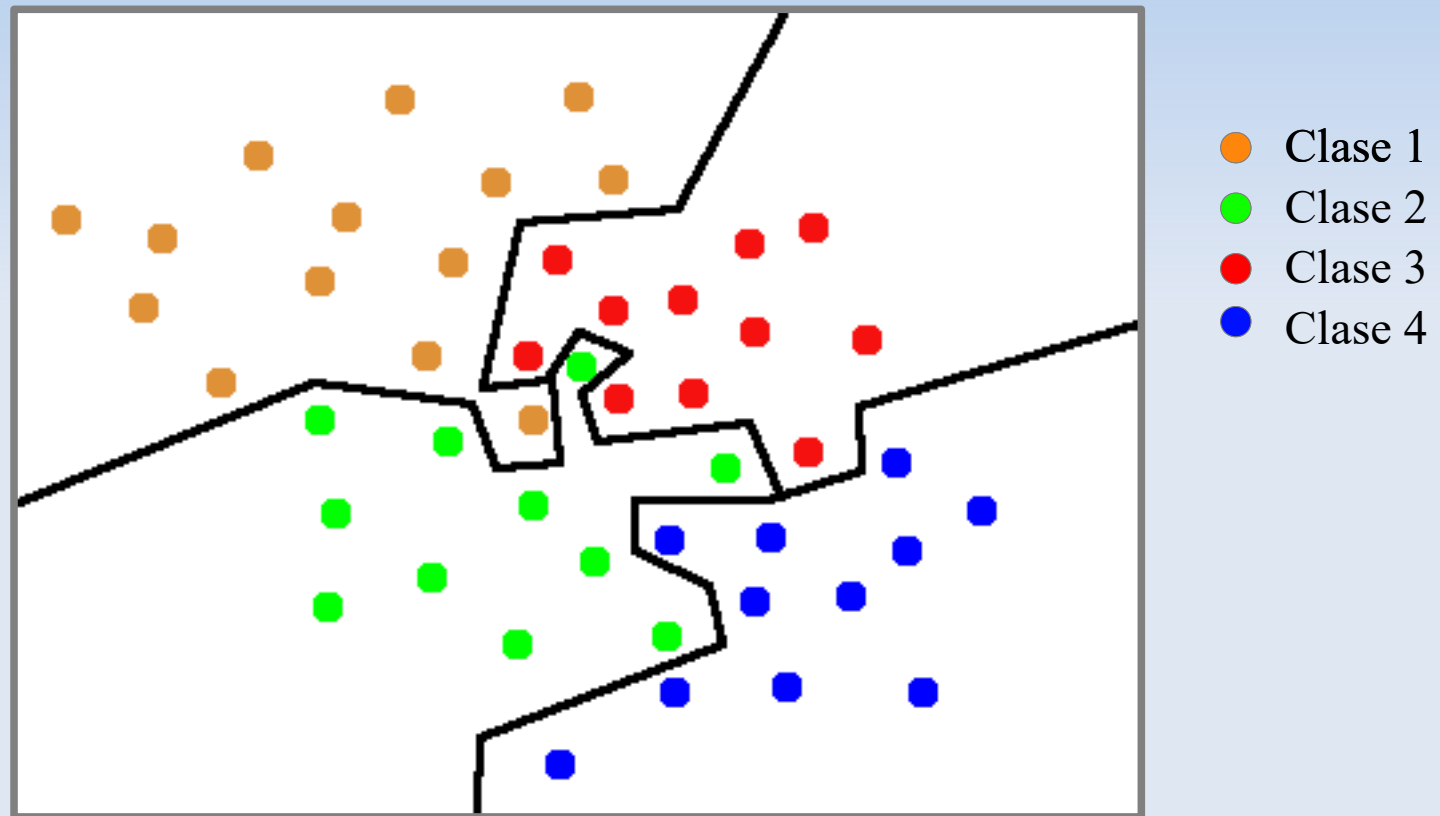
# Reconocimiento de patrones

- Muestras no separables (posible solución):



# Reconocimiento de patrones

- Muestras no separables (otra posible solución):



# Reconocimiento de patrones

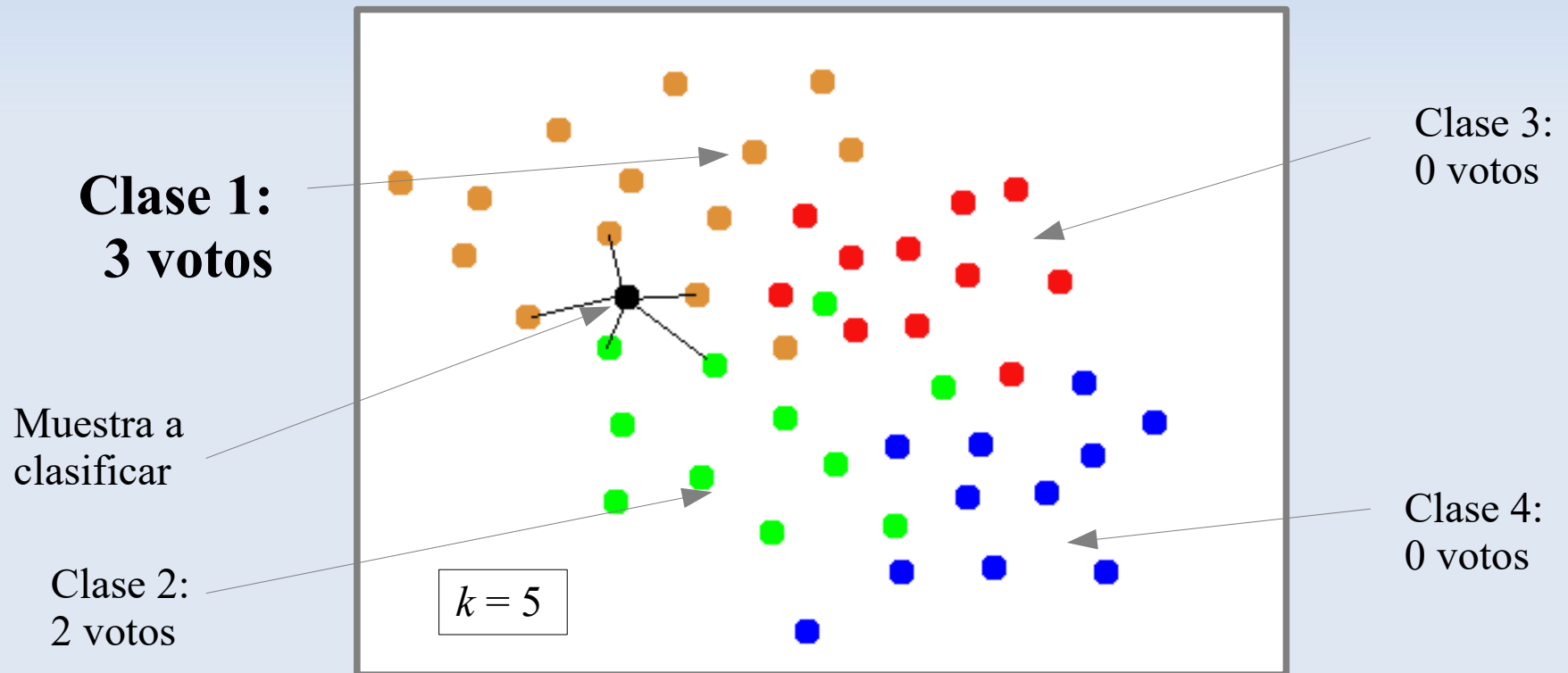
- Clasificación automática: ¿Qué clasificador es mejor?
  - **Generalización**
    - Capacidad de obtener buenos resultados con muestras diferentes (no vistas) a las del entrenamiento.
  - **Sobre-entrenamiento.**
    - Falta de generalización: solo es capaz de clasificar datos similares a los vistos en el entrenamiento.

# Tipos de clasificadores

- Estadísticos:
  - Intentan minimizar la probabilidad de error a nivel estadístico de acuerdo a la regla de Bayes (clasificador óptimo)
  - Inconveniente: requieren estimación de densidad de probabilidad (¿cómo se distribuyen las características en cada clase?)
  - Tipos:
    - Clasificador gaussiano
    - K-vecino más próximo (k-nearest Neighbour)
    - ...

# Tipos de clasificadores

- Ejemplo: **k-vecino más próximo**
  - Busca los  $k$  patrones con menor distancia.
  - Selección por votación mayoritaria



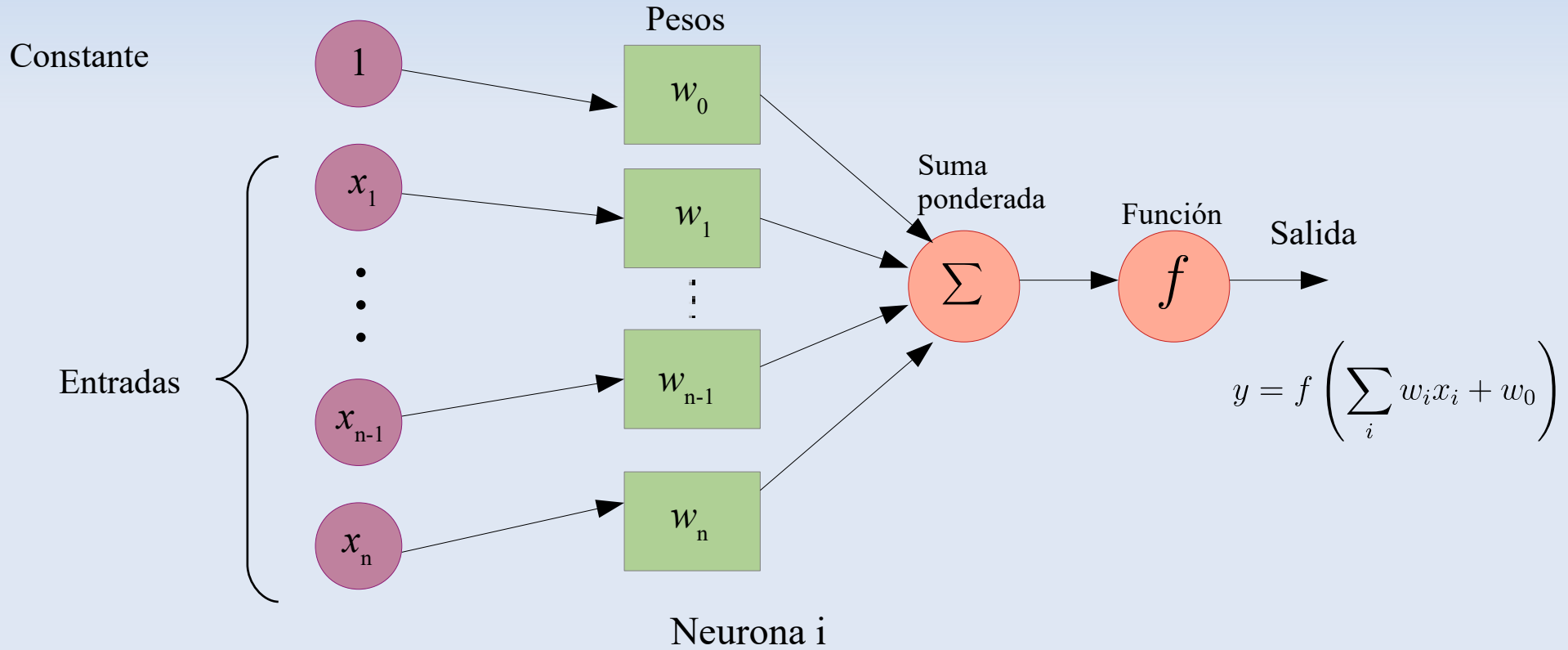
# Tipos de clasificadores

- Inteligencia artificial:
  - No es necesario considerar suposiciones
  - Método de aprendizaje iterativo
  - Minimiza el error sobre conjunto de datos de entrenamiento.
  - Tipos:
    - Redes neuronales (NN: *neural networks*)
      - Perceptrón multicapa
      - Redes RBF
    - Máquinas de vectores soporte (SVM: *support vector machines*).
    - ...



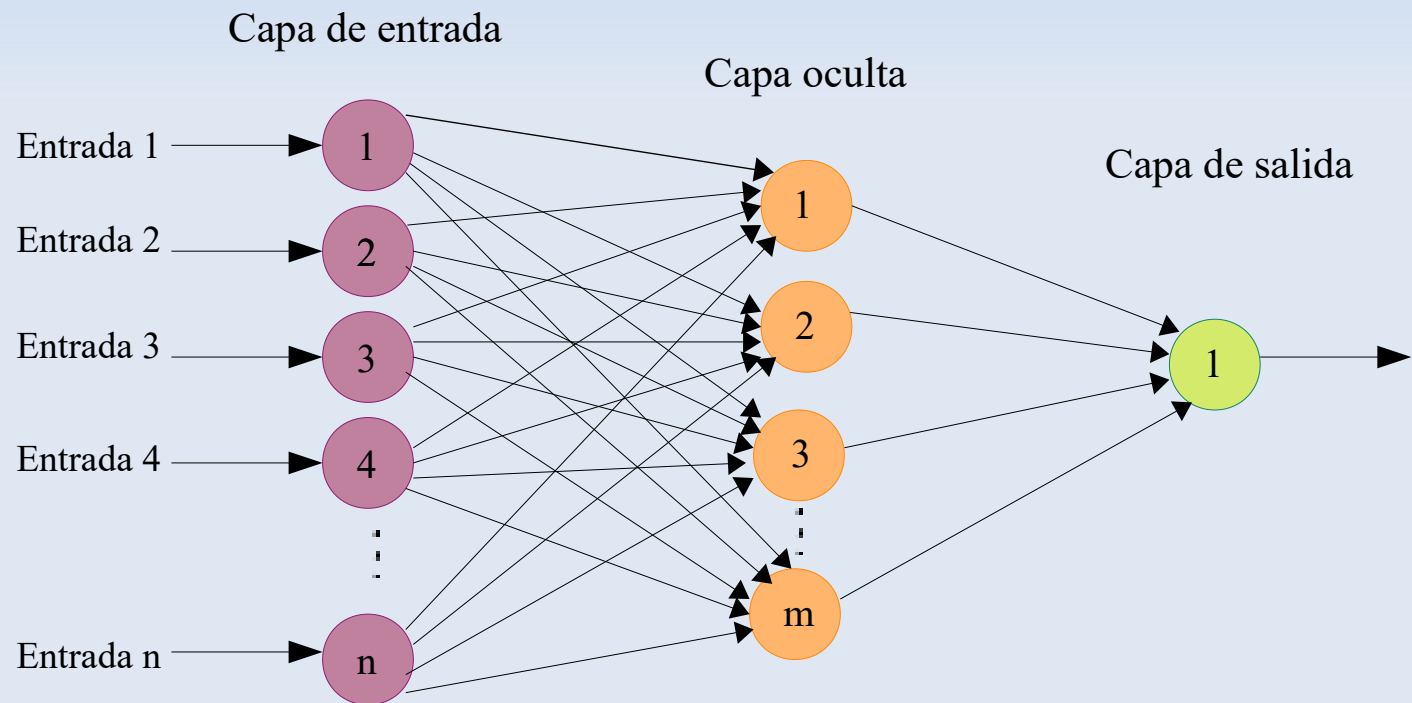
# Tipos de clasificadores

- Redes neuronales:
  - Intentan emular el funcionamiento del cerebro humano. Su unidad básica es la neurona.



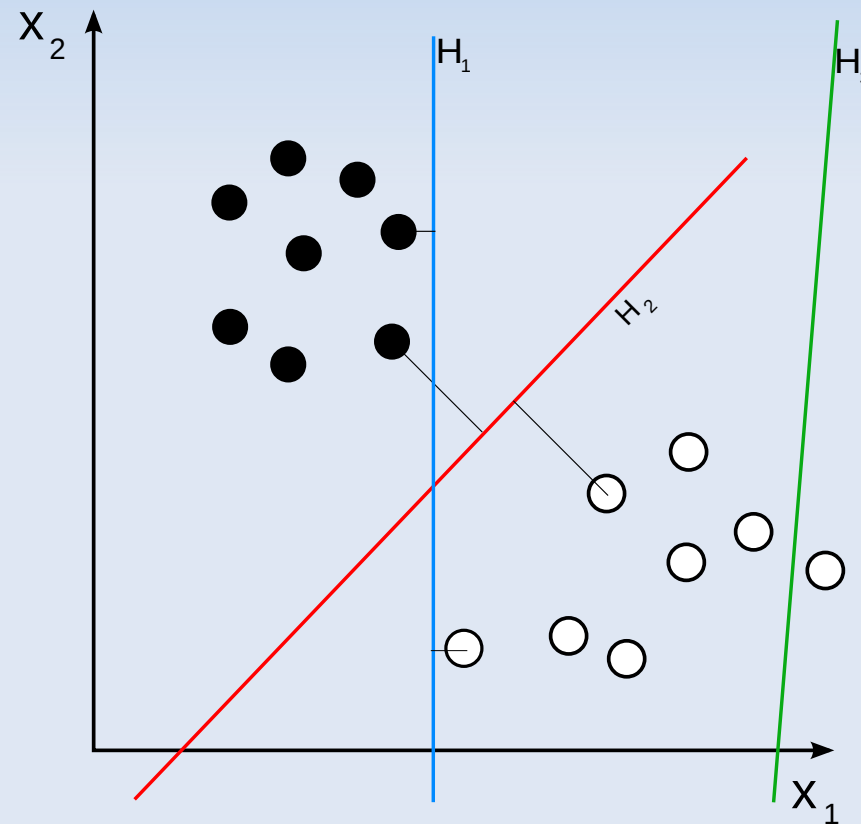
# Tipos de clasificadores

- Redes neuronales:
  - Las neuronas se agrupan en capas para obtener funciones de clasificación complejas.



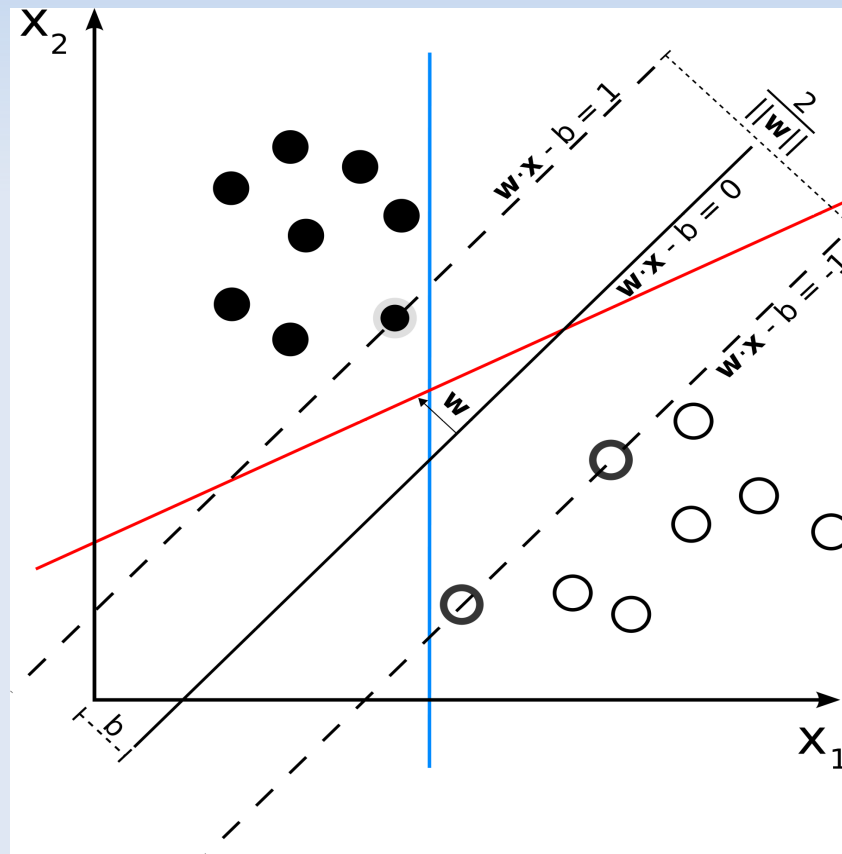
# SVM

- Busca la frontera de decisión con más margen de distancia a las muestras.



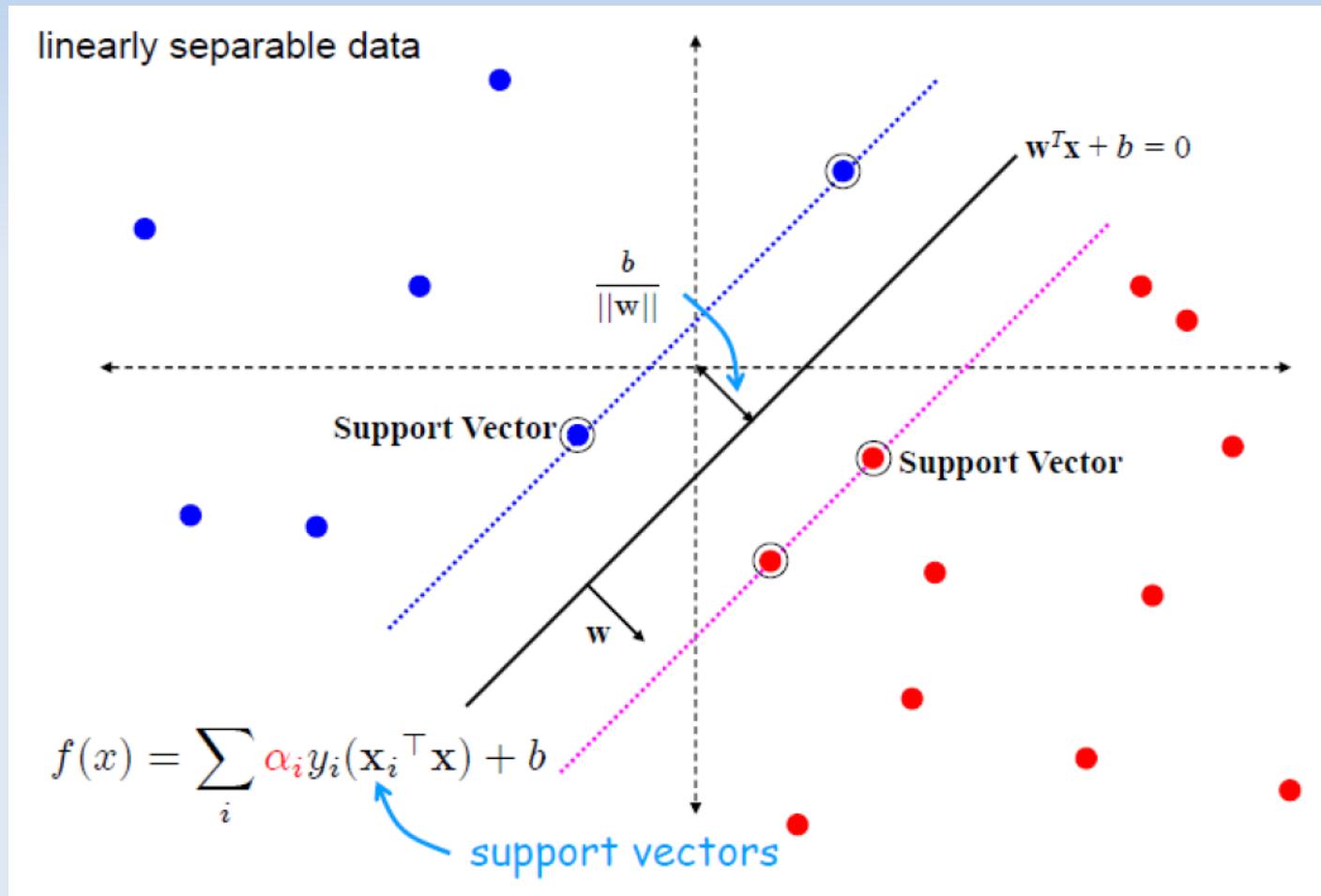
# SVM

- Busca la frontera de decisión con más margen de distancia a las muestras.



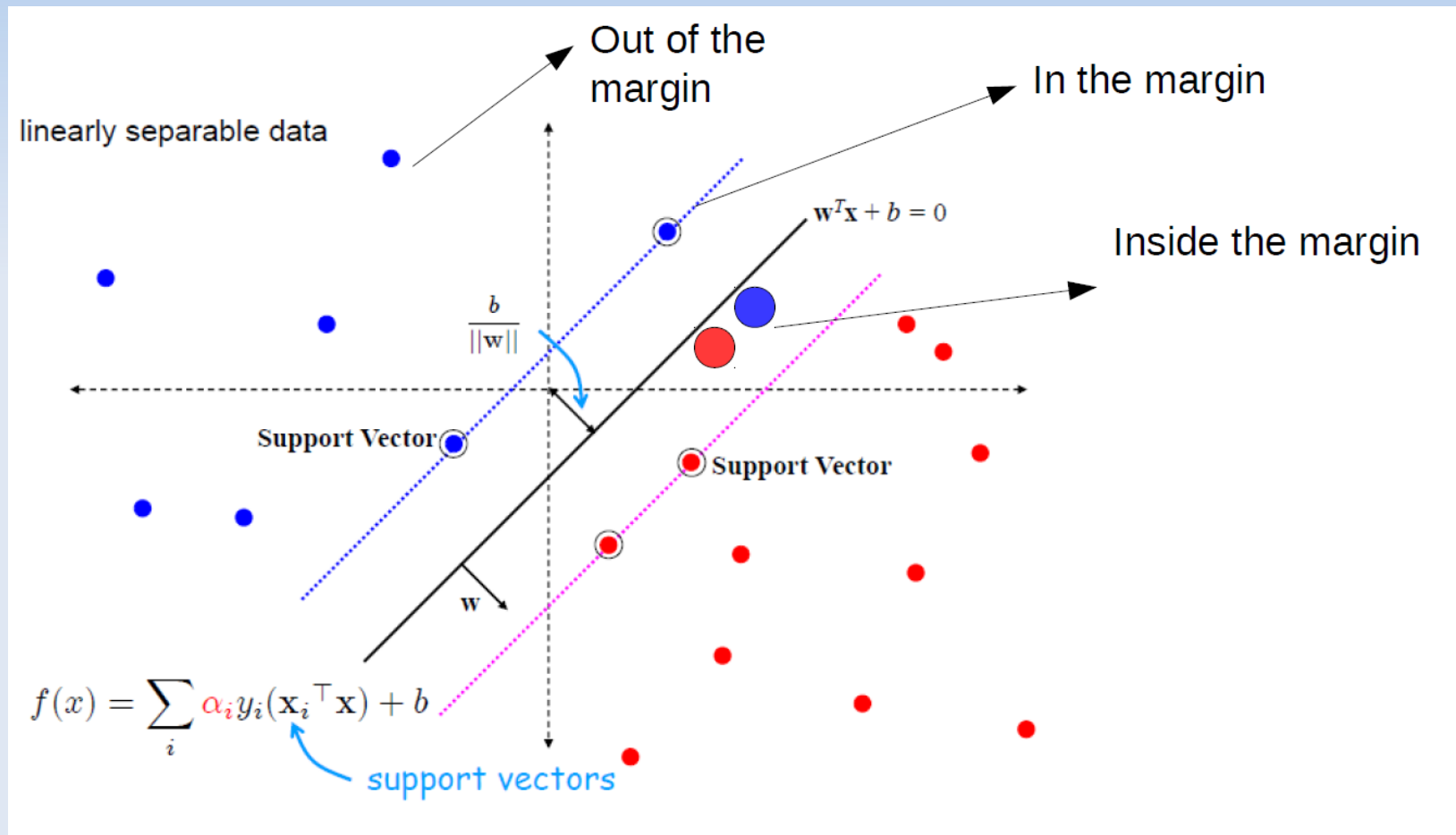
# SVM

- Clases separables linealmente



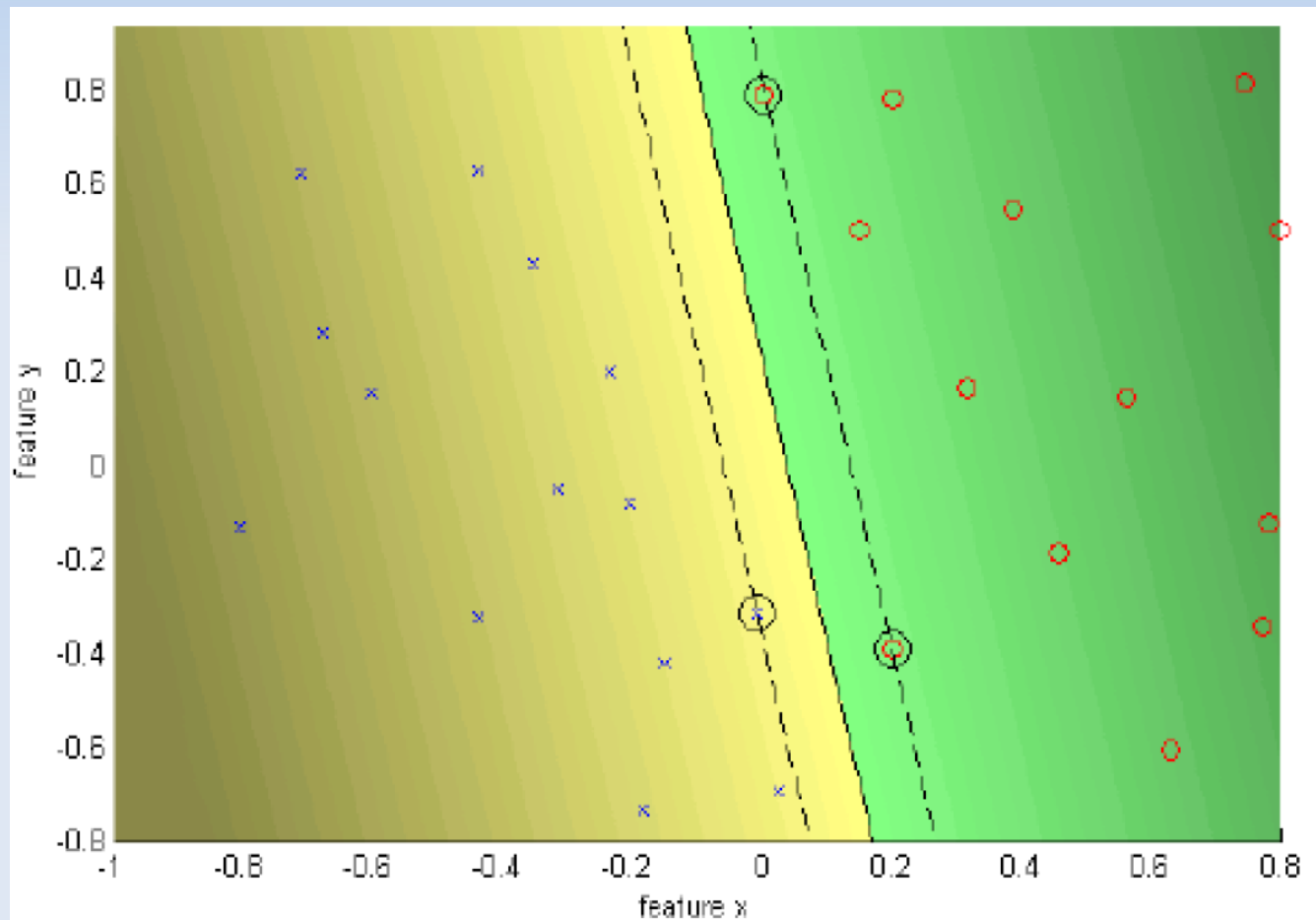
# SVM

- Clases no separables linealmente



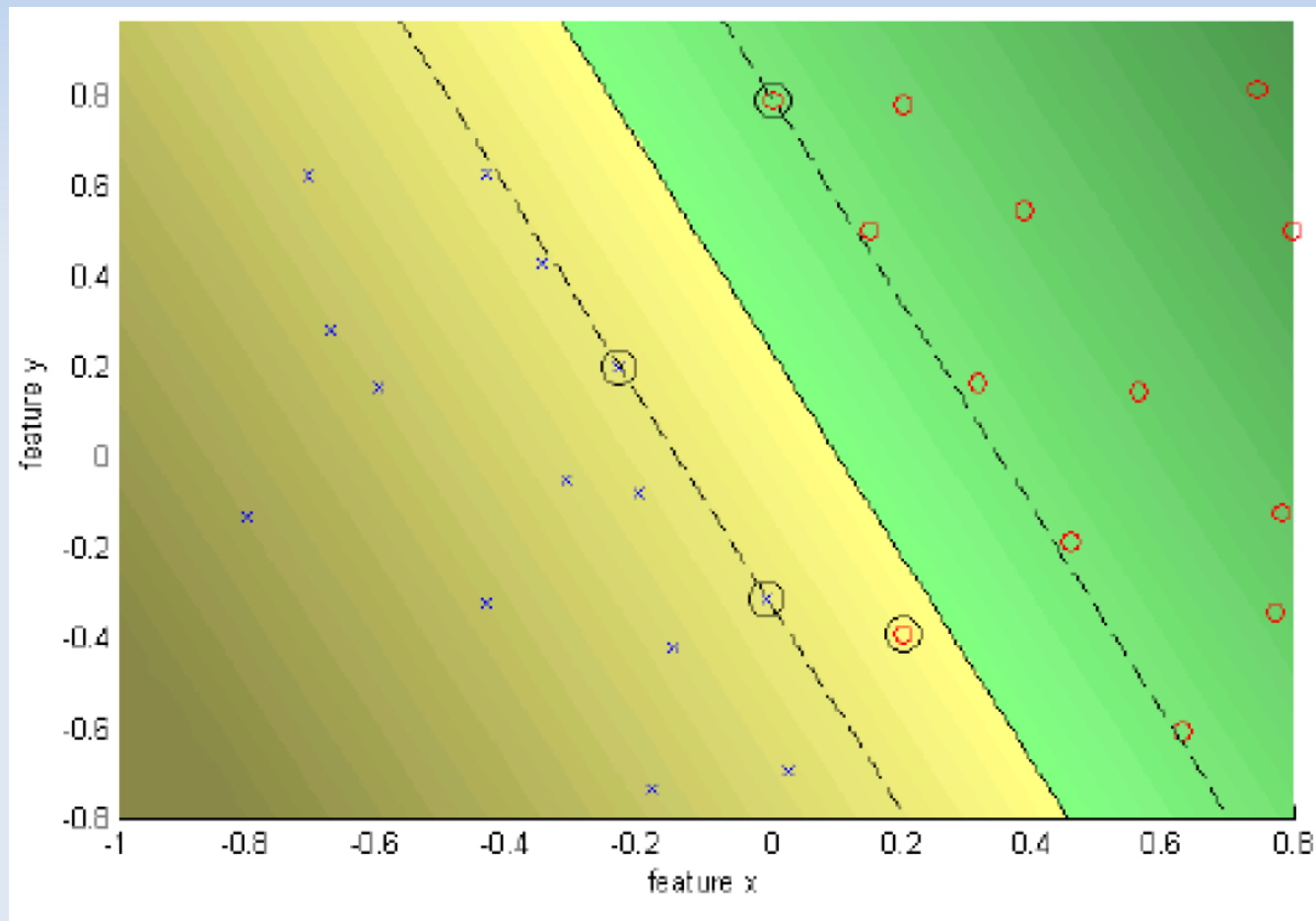
# SVM

- $C=\text{inf}$ , margen 'duro'



# SVM

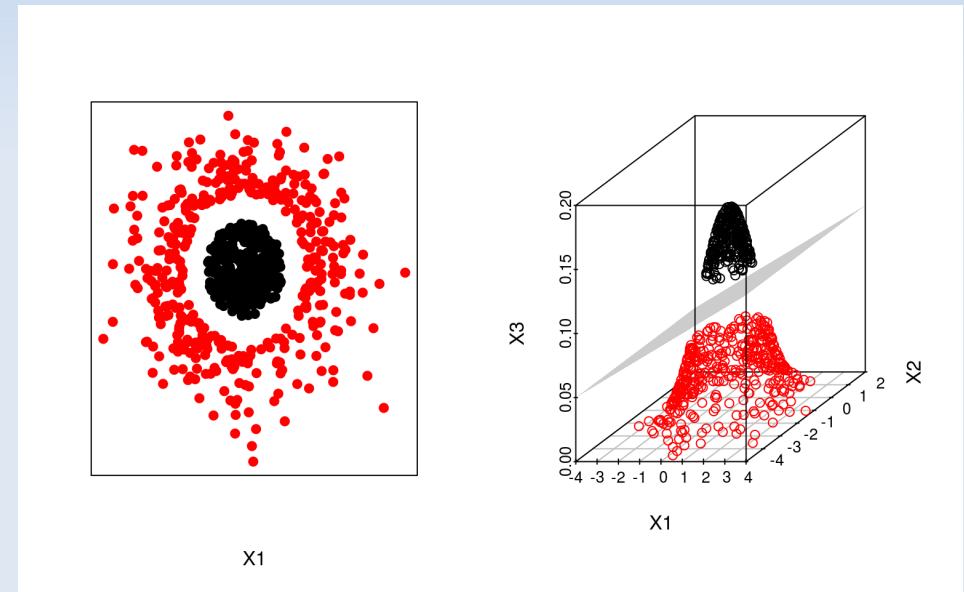
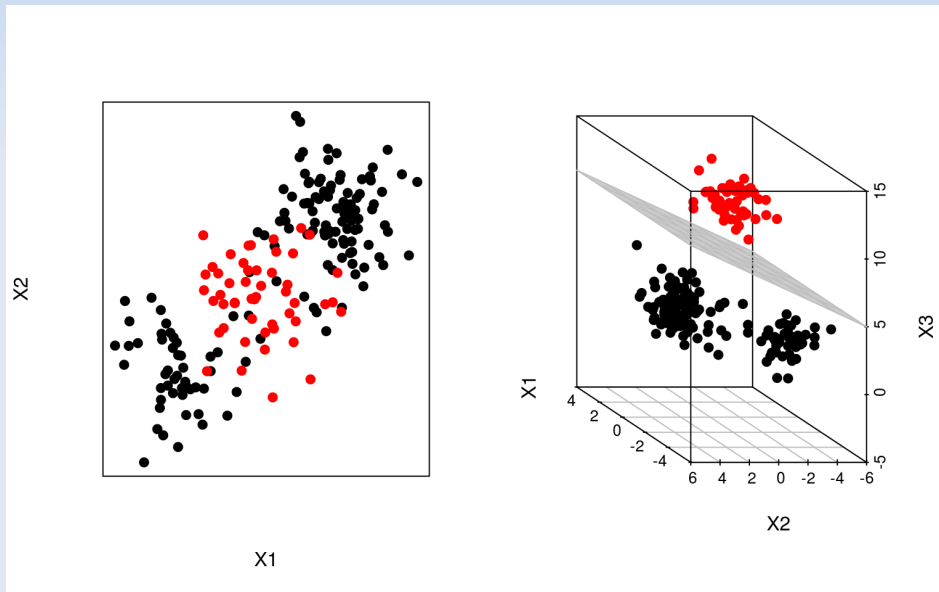
- $C=10$ , margen 'blando'





# SVM

- Clasificación no lineal ('truco' del kernel):



# SVM

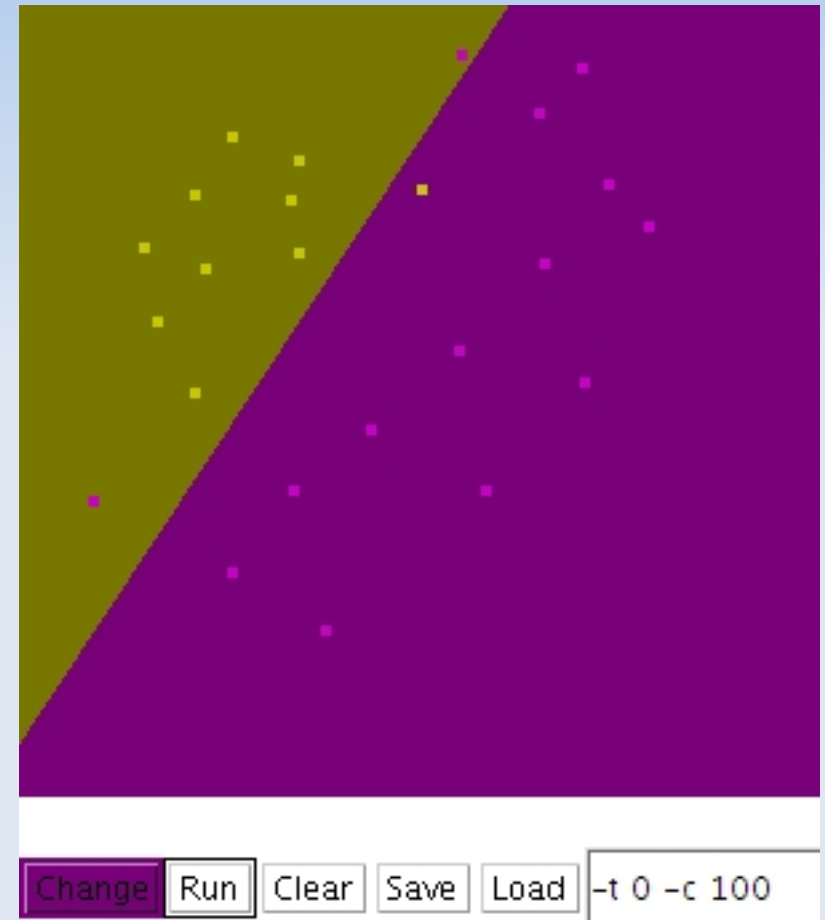
- Clasificación no lineal (‘truco’ del kernel):

$$f(\mathbf{x}) = \text{signo} \left( \sum_{i=1}^{N_{SV}} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \right)$$

- $k$  es el kernel que permite separar los datos.
- Ejemplos de kernel:
  - Gaussiano o RBF:  $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$
  - Polinómico:  $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + c)^d$

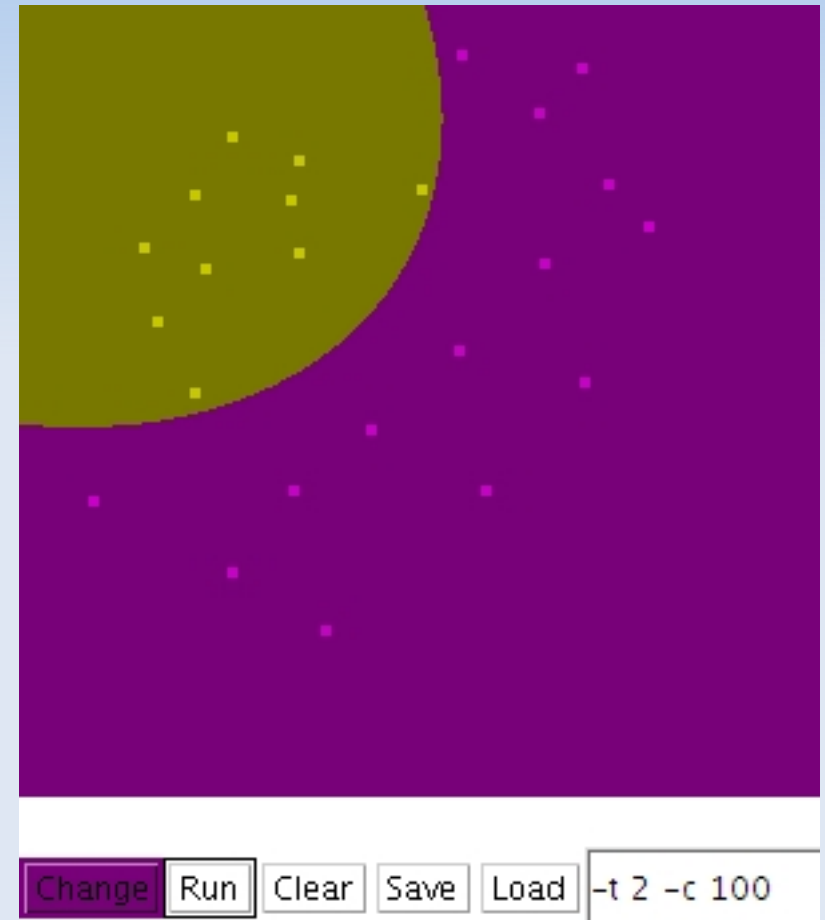
# SVM

- SVM lineal:
  - Frontera es una línea recta.
  - Pequeña complejidad computacional
  - Baja precisión de clasificación



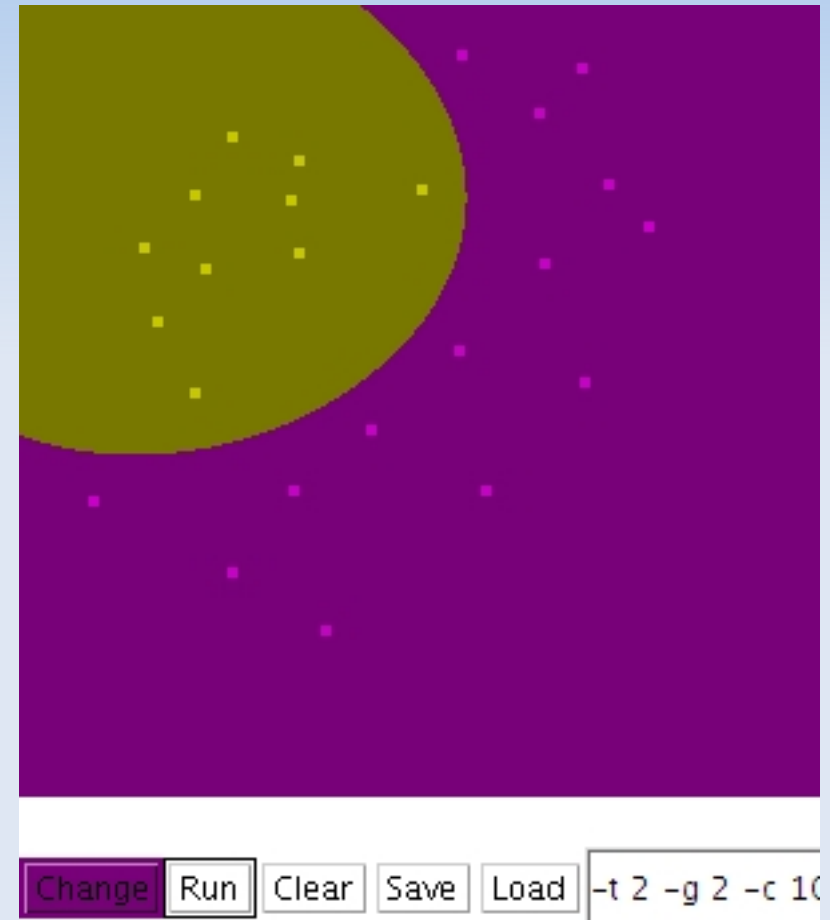
# SVM

- SVM con kernel gaussiano (1):
  - Frontera es una curva:
    - Mayor complejidad computacional
    - Alta precisión de clasificación



# SVM

- SVM con kernel gaussiano (2):
  - Frontera es una curva más compleja:
    - ¿Posible sobre-entrenamiento?



# Referencias

- Referencias:
  - Pattern Recognition and Machine Learning. Cristhoper M. Bishop. Ed: Springer.
  - Pattern Recognition. S. Theodoridis. Ed: Elsevier.
  - <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
  - [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_ml/py\\_svm/py\\_svm\\_basics/py\\_svm\\_basics.html#svm-understanding](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_svm/py_svm_basics/py_svm_basics.html#svm-understanding)
  - <https://www.learnopencv.com/support-vector-machines-svm/>