

# Tema 5: Clasificación



Universidad  
Francisco de Vitoria  
**UFV** Madrid

*Alberto Nogales*

*alberto.nogales@ufv.es*

*Curso 2020-2021*

# Índice

- Definición del problema
- Clasificación basada en reglas.
- Árboles de decisión y sus algoritmos.
- Clasificación basada k-nearest-neighbors.
- Teorema de Bayes y sus clasificadores.
- Support Vector Machines.

# Introducción

A partir de un conjunto de datos con una clase asignada. Estudiar las diferencias entre un grupo y otro, sus características peculiares. Usarla para clasificar nuevos datos con las clases anteriores.

## Aplicaciones:

1. Clasificar tipo de tumores según sus características.
2. Anticipar el fraude bancario (tipos de clientes).
3. Categorizar clientes según su comportamiento (ofertas).
4. Clasificar el tráfico de una zona de la ciudad.

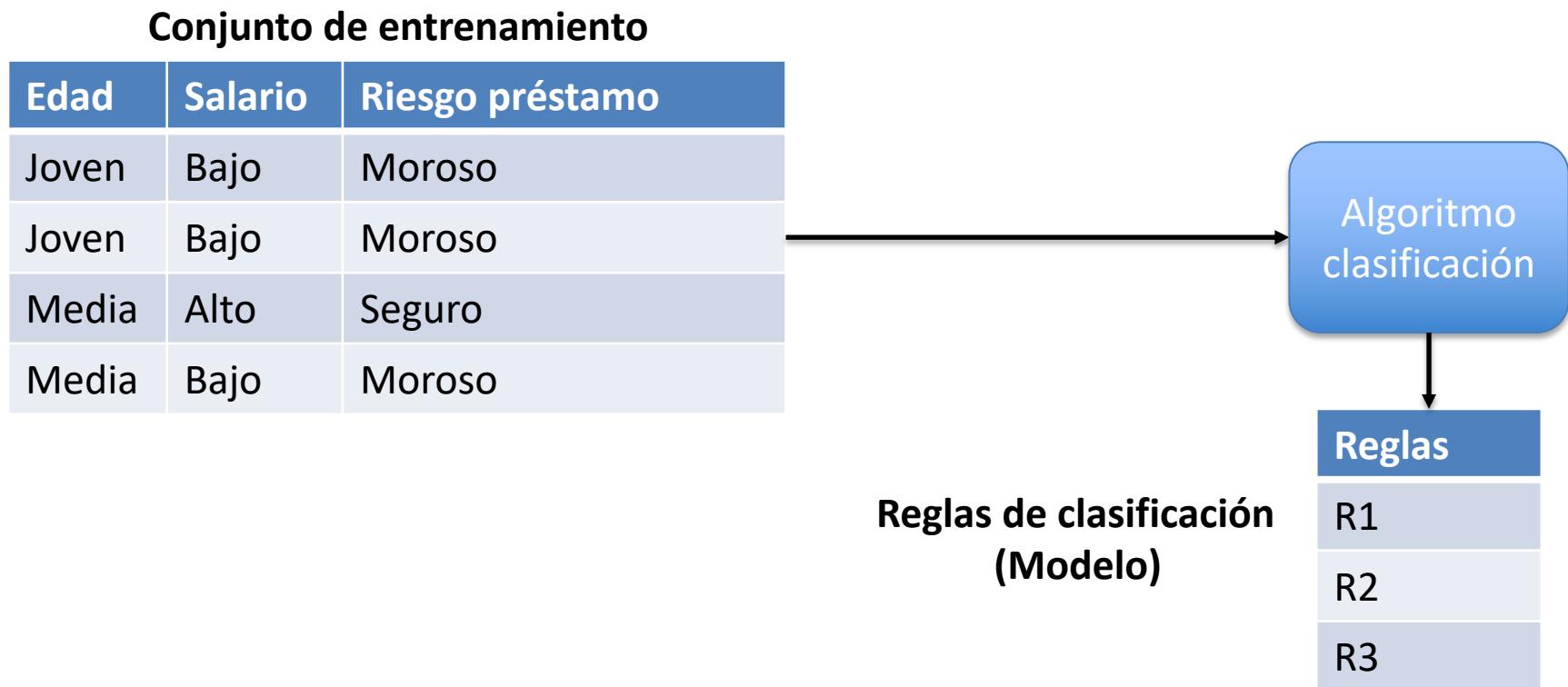
# Introducción

## Métodos:

1. Modelos basados en reglas.
2. Árboles de decisión.
3. Algoritmo k-nearest-neighbors.
4. Redes bayesianas.
5. Support vector machines.

# Definiciones

Paso 1.- Aprendizaje/Entrenamiento: se construye un clasificador describiendo un conjunto de clases.

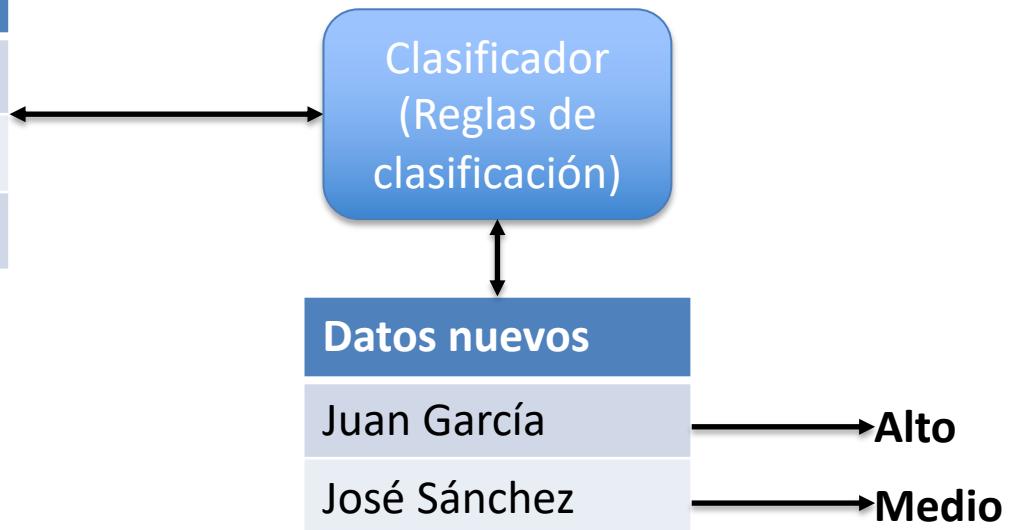


# Definiciones

Paso 2.- Validación y clasificación: donde el clasificador se prueba y usa para asignar clases.

Conjunto de validación

Edad	Salario	Riesgo préstamo
Mayor	Alto	Medio
Joven	Alto	Seguro
Media	Normal	Medio



# Definiciones

- Métodos directos:
  - Extraen las reglas directamente de los datos.
  - Ej: Sequential covering, RIPPER, etc.
- Métodos indirectos:
  - Extraen las reglas de otros métodos de clasificación como los arboles de decisión.
  - Ej: ID3, C4.5, etc.

# Métodos basados en reglas

- Objetivo
- Definiciones del problema
- Método Sequential Covering
- Algoritmo PRISM.

# Definición del problema

- **Objetivo:** dado un conjunto de registros (conjunto de entrenamiento). Cada registro es un conjunto de atributos de los cuales uno indica una clase. Encontrar un modelo para el atributo de clase en función del resto de atributos. Este modelo será usado para clasificar nuevos registros.

Edad	Estación	País
17	Invierno	Austria
23	Verano	Italia
13	Verano	Grecia
24	Invierno	Inglaterra
25	Verano	Italia

If (Edad $\geq$ 18) AND (Estación=Verano)  
Then  
País = Italia

# Definiciones

- Registro: representada por un vector de atributos de dimensión n,  $X=(x_1, x_2, \dots, x_n)$  que describe n medidas  $A_1, A_2, \dots, A_n$
- Etiqueta de atributo de clase: clase categórica a la que pertenece el registro.
- Conjunto de entrenamiento: registros usado para construir el modelo.
- Clasificador: modelo construido para obtener una etiqueta.
- Conjunto de test: registros para medir la calidad del modelo.

# Definiciones

Edad	Estación	País
17	Invierno	Austria
23	Verano	Italia
13	Verano	Grecia
24	Invierno	Inglaterra
25	Verano	Italia

Registro: {17, Invierno, Austria}

- Vector de atributos (Edad, Estación, País)
- Dimensión: 3.

# Definiciones: clasif. basada en reglas

- Reglas: Si (Condición) entonces Y  
Condición es una conjunción de atributos  
Y es la etiqueta que se le asigna a la clase
- Una regla cubre el registro X, si los atributos del registro satisfacen la condición (antecedente) de la regla.
- Cobertura de una regla: fracción de registros (registros cubiertos dividido el total de registros) que satisface el antecedente de una regla.
- Precisión de una regla: fracción de registros que satisface tanto el antecedente como el consecuente de una regla (sobre aquellos que satisfacen el antecedente).
- Una regla es perfecta cuando su precisión es 1.

# Definiciones: clasif. basada en reglas

A devolver	Estado sentimental	Salario	Ayuda
Si	Soltero	125k	No
No	Casado	100k	No
No	Soltero	70k	No
Si	Casado	120k	No
No	Divorciado	95k	Si
No	Casado	60k	No
Si	Divorciado	220k	No
No	Soltero	85k	Si
No	Casado	75k	No
No	Soltero	90k	Si

Regla: Si (Estado\_sentimental=Soltero) entonces Ayuda=No

Cubre los registros: 1, 3, 8 y 10.

La cobertura es 4/10 y la precisión 2/4.

# Clasificación basada en reglas

## Ejemplo (clientes)

Salario	Edad	Nº de hijos	Oferta
Alto	23	0	Tecnología
Bajo	45	4	Alimentación
Medio	35	2	Ocio
Alto	66	3	Ocio

R1: Si (Salario=Alto) AND (Edad<35) Entonces (Oferta=Tecnología)

R2: Si (Salario=Medio) AND (Num\_hijos>3) Entonces (Oferta=Alimentación)

R3: Si (Edad>65) Entonces (Oferta=Ocio)

# Clasificación basada en reglas

## Funcionamiento:

R1: Si (Salario=Alto) AND (Edad<35) Entonces (Oferta=Tecnología)

R2: Si (Salario=Medio) AND (Num\_hijos>3) Entonces (Oferta=Alimentación)

R3: Si (Edad>65) Entonces (Oferta=Ocio)

Cliente	Salario	Edad	Nº de hijos	Oferta
Miguel Pérez	Alto	31	1	?
María Díaz	Medio	72	4	?

Miguel Pérez activa la regla R1 se clasifica como Tecnología

María Díaz activa la regla R3 se clasifica como Ocio

# Clasificación basada en reglas

## Propiedades de las reglas:

- Reglas de exclusión mutua:
  - Dos reglas no pueden ser activadas por el mismo registro.
  - Asegura que cada registro es cubierto por máximo una regla.
- Reglas exhaustivas:
  - Existe una regla por cada combinación de valores en los atributos.
  - Asegura que cada registro es cubierto por al menos una regla.

# Clasificación basada en reglas

## Método Sequential Covering:

1. Se empieza con la regla vacía
2. Crear una regla usando la función Learn-One-Rule
3. Eliminar del set de entrenamiento los registros cubiertos por la regla
4. Repetir el paso 2 y 3 hasta que se cumple el criterio de parada

# Clasificación basada en reglas

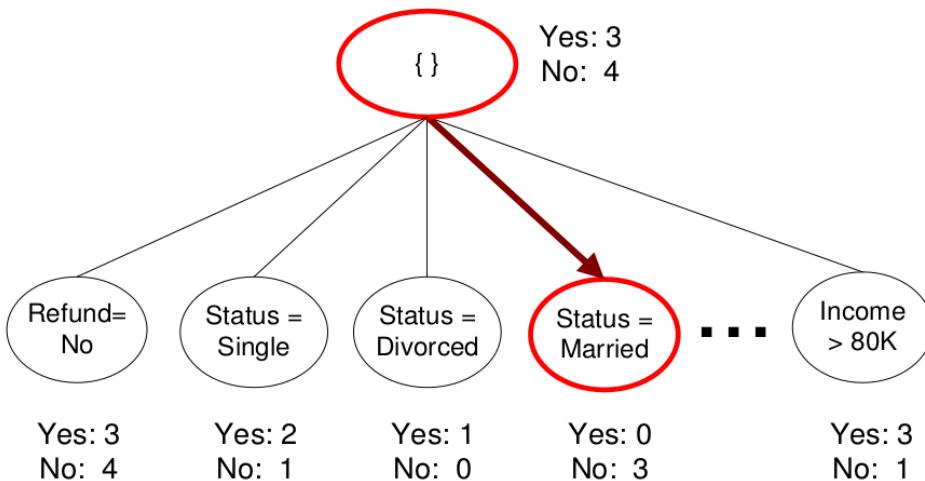
Learn one rule:

- De genérica a específica: Empezar con la hipótesis que más generaliza y luego ir especializándola paso a paso.
- De específica a genérica: Empezar con un conjunto de hipótesis lo más específicas posibles y luego ir generalizando por pasos.

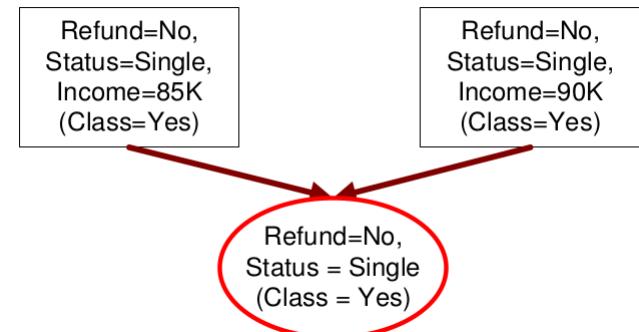
# Clasificación basada en reglas

Learn one rule:

(1) Genérico a específico



(2) Específico a genérico



# Clasificación basada en reglas

Algoritmo PRISM:

Para cada clase C:

    Sea E el conjunto de entrenamiento

    Mientras E tenga registros de C:

        Crea una regla R con antecesor vacío y clase C

        Hasta que R sea perfecta o no haya atributos A:

            Para cada atributo A no incluido en R y cada valor v

                Añadir la condición  $A=v$  al antecesor de R

                Selecciona el par  $A=v$  que maximice la precisión p/t

                (en caso de empate, selecciona el que tenga mayor p)

                Añade  $A=v$  a R

Learn  
One  
Rule

    Elimina de E los registros cubiertos por R

# Clasificación basada en reglas

Ejemplo (Decidir tipo de lentes):

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended Lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

# Clasificación basada en reglas

1) Con uno de los valores del atributo de clase.  
Planteamos una regla.

Si ? Entonces Recommended\_lens="hard"

2) Generar posibles reglas candidatas con cada atributo y sus valores.

Para Age

Age="young"

Age="pre-presbyopic"

Age="presbyopic"

# Clasificación basada en reglas

3) Calcular la precisión para todas las reglas candidatas. Clase Recommended\_lens="hard".

Atrib. 1	{ Age="young" Age="pre-presbyopic" Age="presbyopic"	2/8 1/8 1/8
Atrib. 2	{ Spectacle prescription="myope" Spectacle prescription="hypermetrope"	3/12 3/12
Atrib. 3	{ Astigmatism="no" Astigmatism="yes"	0/12 4/12
Atrib. 4	{ Tear prod. rate="reduced" Tear prod. rate="normal"	0/12 4/12

# Clasificación basada en reglas

4) Se añade la regla con mayor precisión. **OJO!!! Hay empate.**

Si **Astigmatism=“yes”** Entonces  
**Recommended\_lens=“hard”**

5) Se reduce el set de datos eligiendo aquellas instancias que cubren la regla (antecedente). La precisión de la regla es 4/12 hay que seguir refinando.

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

# Clasificación basada en reglas

6) Se vuelve al paso inicial.

**Si Astigmatism=“yes” AND ? Entonces  
Recommended\_lens=“hard”**

7) Calcular la precisión para todas las reglas candidatas.

Atrib. 1	{	Age=“young”	2/4
		Age=“pre-presbyopic”	1/4
		Age=“presbyopic”	1/4
Atrib. 2	{	Spectacle prescription=“myope”	3/6
		Spectacle prescription=“hypermetrope”	1/6
Atrib. 3	{	Tear prod. rate=“reduced”	0/6
		Tear prod. rate=“normal”	4/6

# Clasificación basada en reglas

6) Se vuelve a añadir la que maximice la precisión y se repite hasta que se encuentre una regla perfecta.

**Si Astigmatism=“yes” AND Tear\_prod\_rate=“normal**  
**Entonces Recommended\_lens=“hard”**

Precisión=4/6

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None

# Clasificación basada en reglas

7) Se repite lo anterior

Si **Astigmatism**=“yes” AND  
**Tear\_prod\_rate**=“normal” AND ? Entonces  
    **Recommended\_lens**=“hard”

Posibles reglas:

**Age**=“young”

2/2

**Age**=“pre-presbyopic”

1/2

**Age**=“presbyopic”

1/2

**Spectacle prescription**=“myope”

3/3

**Spectacle prescription**=“hypermetrope”

1/3

Hay empate, se elije la que tenga mayor p.

# Clasificación basada en reglas

8) Se para porque la precisión es 1,0 que indica que son reglas perfectas.

**Si `Astigmatism="yes"` AND `Tear_prod_rate="normal"` AND  
`Spectacle_prescription="myope"`**

Entonces `Recommended lens="hard"`

**Si `Age="young"` AND `Astigmatism="yes"` AND  
`Tear_prod_rate="normal"`**

Entonces `Recommended lens="hard"`

9) Repetir para hasta que no haya registros. Para ello quitar aquellos que cumplen con las reglas generadas.

# Clasificación basada en reglas

Ejemplo (Climatología):

Día	Vista	Temperatura	Humedad	Viento	Paraguas
1	Soleado	Alta	Alta	No	No
2	Soleado	Alta	Alta	Sí	No
3	Nublado	Alta	Alta	No	Sí
4	Lluviosos	Media	Alta	No	Sí
5	Lluviosos	Baja	Normal	No	Sí
6	Lluviosos	Baja	Normal	Sí	No
7	Nublado	Baja	Normal	Sí	Sí
8	Soleado	Media	Alta	No	No
9	Soleado	Baja	Normal	No	Sí
10	Lluviosos	Media	Normal	No	Sí
11	Soleado	Media	Normal	Sí	Sí
12	Nublado	Media	Alta	Si	Si
13	Nublado	Alta	Normal	No	Sí
14	Lluviosos	Media	Alta	Sí	No

# Clasificación basada en reglas

1.1) Si ? Entonces Paraguas="Sí"

Vista="Soleado"	2/5
Vista="Nublado"	4/4
Vista="Lluvioso"	3/5

Cómo es perfecta, generamos una regla.

Regla: Si Vista="Nublado Entonces Paraguas="Sí"  
Quitamos los registros que cubren la regla.

# Clasificación basada en reglas

Ejemplo (Climatología):

Día	Vista	Temperatura	Humedad	Viento	Paraguas
1	Soleado	Alta	Alta	No	No
2	Soleado	Alta	Alta	Sí	No
3	Nublado	Alta	Alta	No	Sí
4	Lluviosos	Media	Alta	No	Sí
5	Lluviosos	Baja	Normal	No	Sí
6	Lluviosos	Baja	Normal	Sí	No
7	Nublado	Baja	Normal	Sí	Sí
8	Soleado	Media	Alta	No	No
9	Soleado	Baja	Normal	No	Sí
10	Lluviosos	Media	Normal	No	Sí
11	Soleado	Media	Normal	Sí	Sí
12	Nublado	Media	Alta	Sí	Sí
13	Nublado	Alta	Normal	No	Sí
14	Lluviosos	Media	Alta	Sí	No

# Clasificación basada en reglas

1.2) Si ? Entonces Paraguas=“Sí”

Vista=“Soleado”	2/5
Vista=“Lluvioso”	3/5
Temperatura=“Alta”	0/2
Temperatura=“Media”	3/5
Temperatura=“Baja”	2/3
Humedad=“Alta”	1/5
Humedad=“Normal”	4/5
Viento=“Sí”	1/4
Viento=“No”	4/6

# Clasificación basada en reglas

2) Si **Humedad=“Normal” AND ? Entonces Paraguas=“Sí”**

Vista=“Soleado”	2/2
Vista=“Lluvioso”	2/3
Temperatura=“Alta”	0/0
Temperatura=“Media”	2/2
Temperatura=“Baja”	2/3
Viento=“Sí”	1/2
Viento=“No”	3/3

**Si **Humedad=“Normal” AND Viento=“No”****

**Entonces Paraguas=“Sí”**

# Clasificación basada en reglas

3) Otras reglas:

**Si Temperatura=“Media” AND Humedad=“Normal”**

Entonces Paraguas=“Sí”

**Si Vista=“Lluvioso” AND Viento=“No” Entonces**

Paraguas=“Sí”

**Si Vista=“Soleado” AND Humedad=“Alta” Entonces**

Paraguas=“No”

**Si Vista=“Lluvioso” AND Viento=“Sí” Entonces**

Paraguas=“No”

# Clasificación basada en reglas

Ejemplo (compras):

Acceso	Gasto	Vivienda	Última compra	Tipo de cliente
Transp. Público	Alta	Comprada	Libro	1
Transp. Público	Alta	Alquilada	Disco	2
Transp. Público	Baja	Visitante	Libro	1
A píe	Baja	Alquilada	Libro	1
Transp. Público	Normal	Alquilada	Libro	2
Coche	Baja	Alquilada	Libro	2

# Clasificación basada en reglas

Ejemplo (lentes):

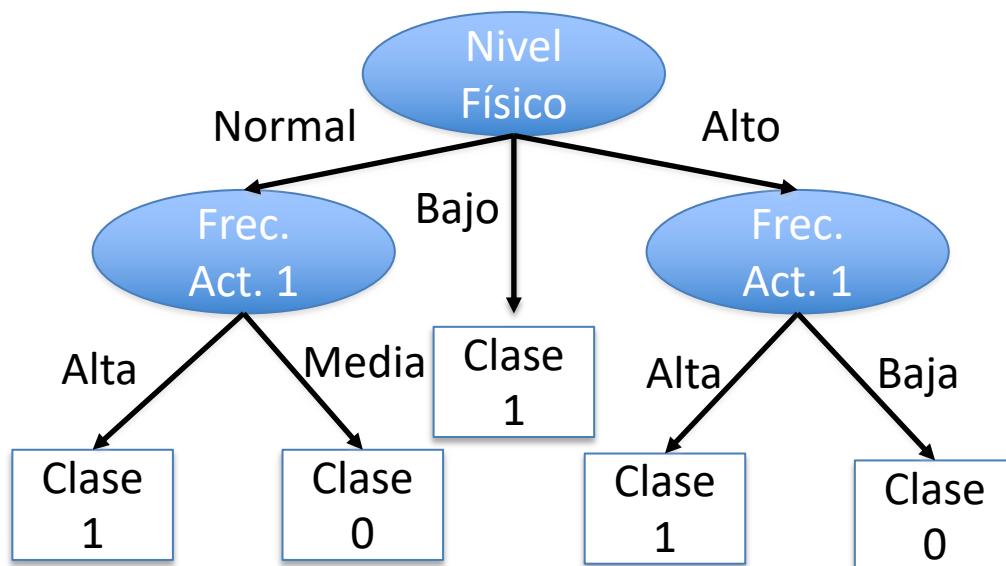
Cliente	Edad	Diagnóstico	Lágrima	Recomendaci ón
1	Joven	Miope	Normal	Duras
2	Joven	Hipermétrope	Normal	Duras
3	Prepresbicia	Miope	Normal	Duras
4	Presbicia	Miope	Normal	Duras
5	Joven	Miope	Reducida	Nada
6	Joven	Miope	Reducida	Nada
7	Joven	Hipermétrope	Reducida	Nada
8	Joven	Hipermétrope	Reducida	Nada

# Arboles de decisión

- Objetivo
- Definiciones del problema
- Funcionamiento de los arboles de decisión.
- Algoritmo ID3
- Algoritmo C4.5

# Definición del problema

- **Objetivo:** analizar decisiones secuenciales basadas en el uso de resultados y probabilidades asociadas. Facilita la toma de mejores decisiones, especialmente cuando existen riesgos, costes, beneficios y múltiples opciones. Se obtienen reglas.



```
If (Nivel_físico="Normal") AND  
(Frec_act_1="Media") Entonces  
    Clase 0
```

# Definiciones

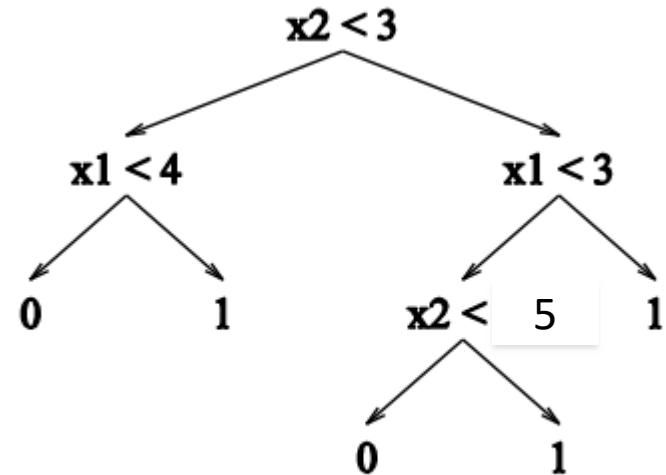
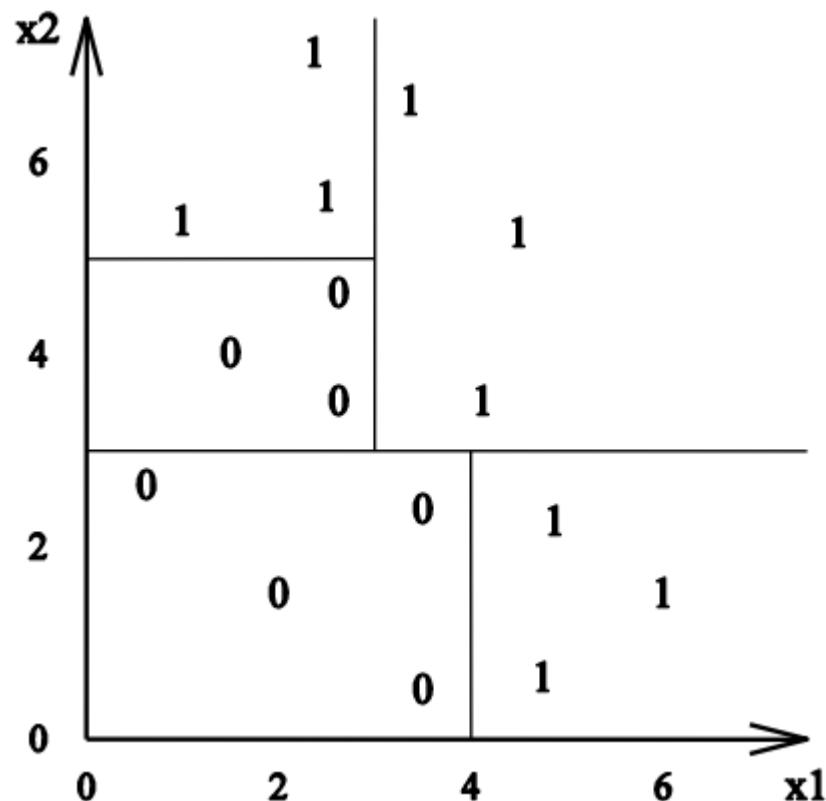
- Nodo de decisión (hoja): indica que decisión necesita tomarse en ese punto del proceso. Está representado por un cuadrado.
- Nodo de probabilidad: indica que en ese punto del proceso ocurre un evento aleatorio. Está representado por un círculo.
- Rama: muestra los distintos caminos que se pueden emprender cuando tomamos una decisión o bien ocurre algún evento aleatorio. Está representado por una flecha.

# Definiciones

- Objetivo: busca la mejor partición sobre el conjunto de datos, es decir que separe los distintos casos en grupos tan diferentes entre sí como sea posible.
- Los casos de la misma partición tendrán la máxima semejanza posible entre sí.
- Se consigue la máxima homogeneidad posible cuando todos los casos que aparecen en una partición pertenecen a la misma clase.

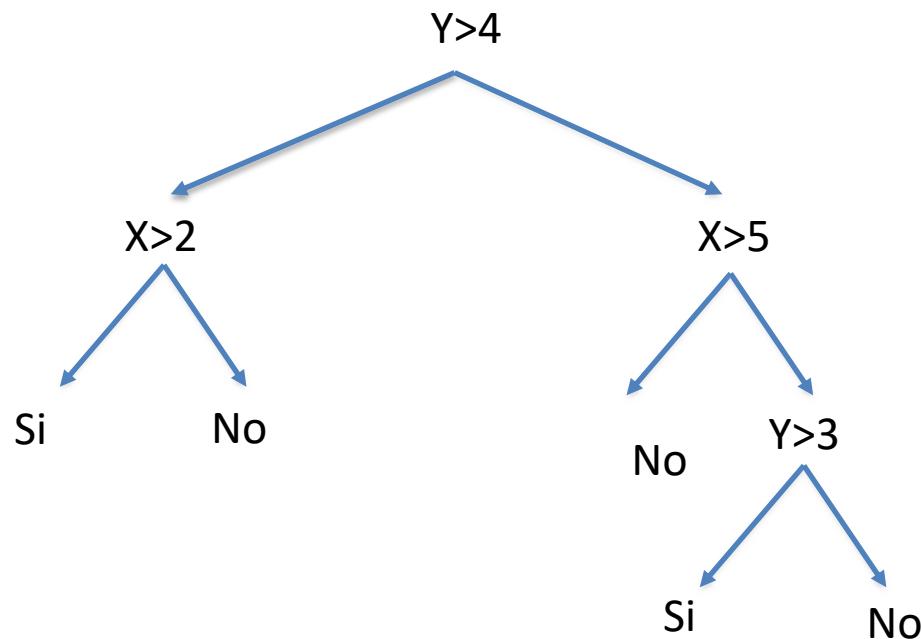
**La clave es elegir el atributo que mejor separa**

# Definiciones



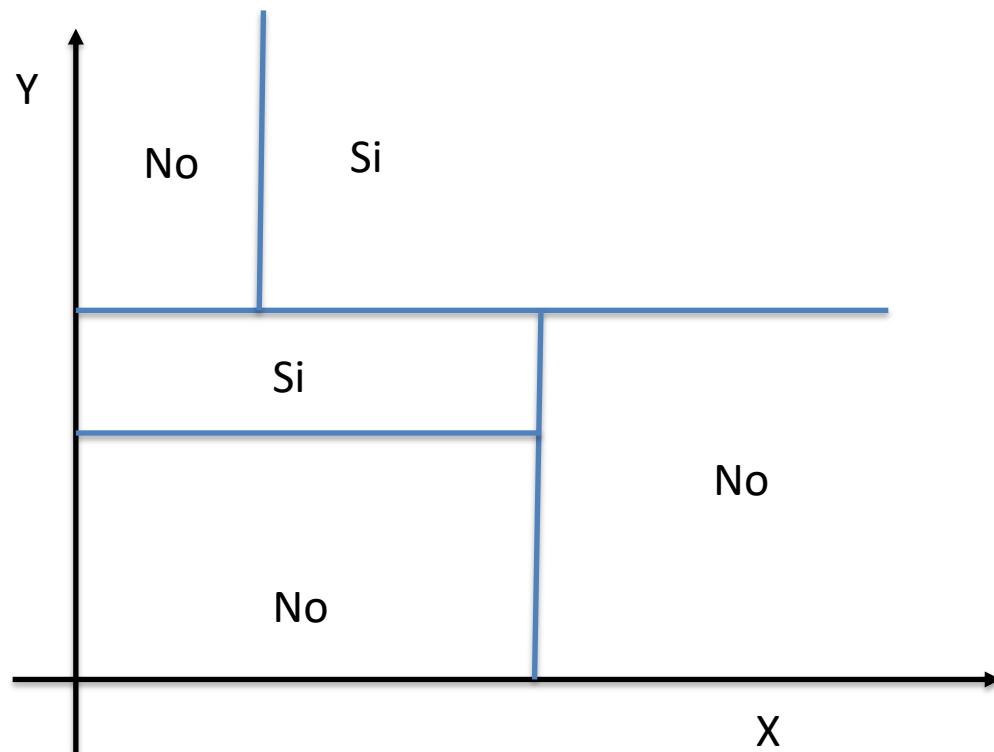
# Definiciones

Ejemplo, separación en espacio n-dimensional:



# Definiciones

## Solución



# Funcionamiento básico

Ejemplo (clientes gimnasio):

Cliente	Nivel físico	Frec. Act. 1	Frec. Act. 2	Clase
1	Normal	Media	Baja	0
2	Bajo	Baja	Baja	1
3	Normal	Alta	Alta	1
4	Alto	Alta	Alta	1
5	Alto	Baja	Baja	0
6	Bajo	Media	Media	1
7	Normal	Media	Media	0
8	Alto	Alta	Alta	1

# Funcionamiento básico

1) Se elige un atributo “Nivel\_físico”

Normal

Cliente	Clase
1	0
3	1
7	0

Bajo

Cliente	Clase
2	1
6	1



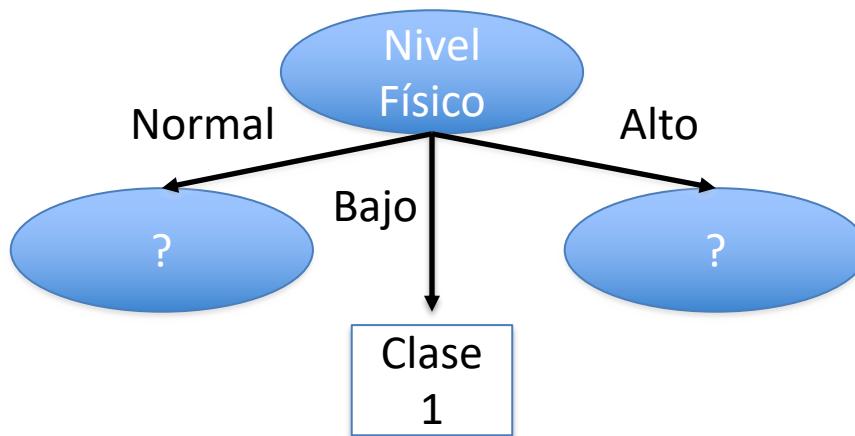
Son completamente  
homogéneos se puede  
agrupar en un nodo de  
decisión

Alto

Cliente	Clase
4	1
5	0
8	1

# Definición del problema

Para el atributo “Nivel\_físico”. Se ha generado un nodo de decisión con valor “Bajo” porque todas los individuos son de la misma clase. Para el resto de valores se generan nodos de probabilidad.



# Funcionamiento básico

2) Combinamos el atributo del paso anterior con uno nuevo, Frec\_act1.

Nivel\_físico=“Normal” AND Frec\_act1=“Medio”

Cliente	Clase
1	0
7	0

Nivel\_físico=“Normal” AND Frec\_act1=“Alta”

Cliente	Clase
3	1

Nivel\_físico=“Alto” AND Frec\_ac1=“Medio”

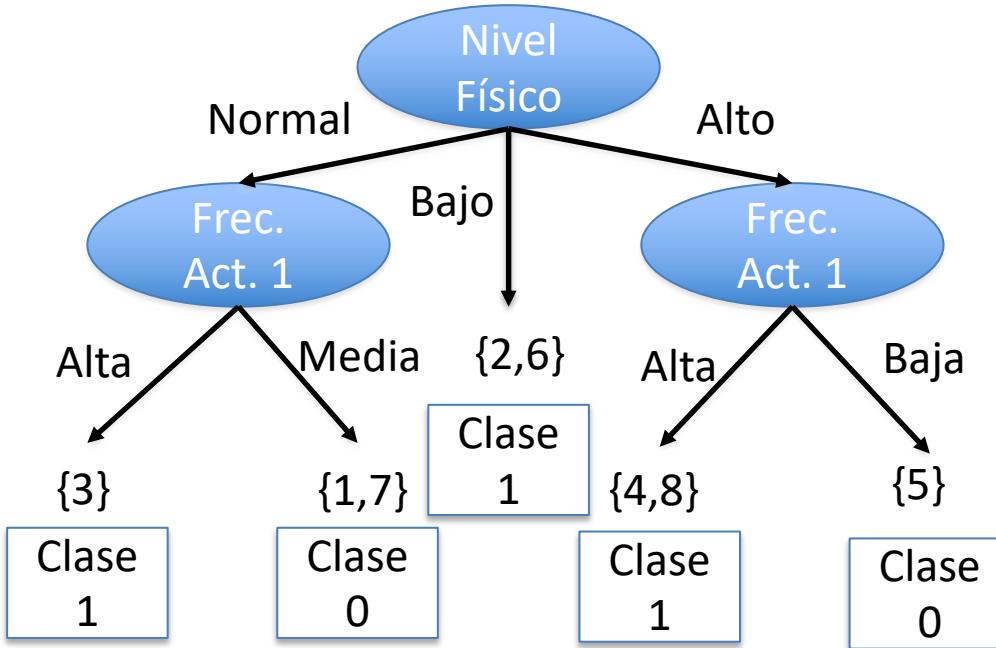
Cliente	Clase
4	1
8	1

Nivel\_físico=“Alto” AND Frec\_ac1=“Medio”

Cliente	Clase
5	0

# Funcionamiento básico

Para el atributo “Frec\_act1”. Como todos los individuos se han agrupado en clases, todos los nodos que resultan son de decisión



# Arboles de decisión

Algoritmo Hunt:

Sea  $D_t$  el conjunto de datos de entrenamiento sobre un nodo  $t$

Si  $D_t$  contiene registros que pertenecen todos a la misma clase  $y_t$  entonces  $t$  es un nodo hoja etiquetado como  $y_t$

Si  $D_t$  contiene registros que pertenecen más de una clase, usaremos un atributo para dividir los datos en subconjuntos más pequeños, aplicando de forma recursiva el algoritmo hasta tener sólo nodos hoja.

# Arboles de decisión

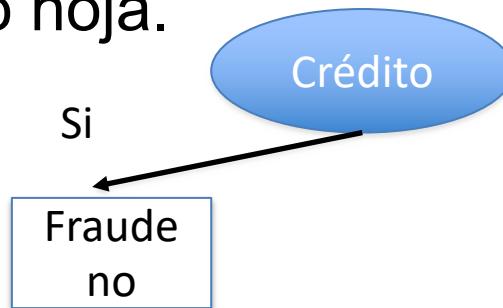
Ejemplo (fraude):

Crédito	Edad	Saldo	Fraude
Si	Joven	12,5k	No
No	Mediana	10k	No
No	Joven	7k	No
Si	Mediana	12k	No
No	Mayor	9,5k	Si
No	Mediana	6k	No
Si	Mayor	22k	No
No	Joven	8,5k	Si
No	Mediana	7,5k	No
No	Joven	9k	Si

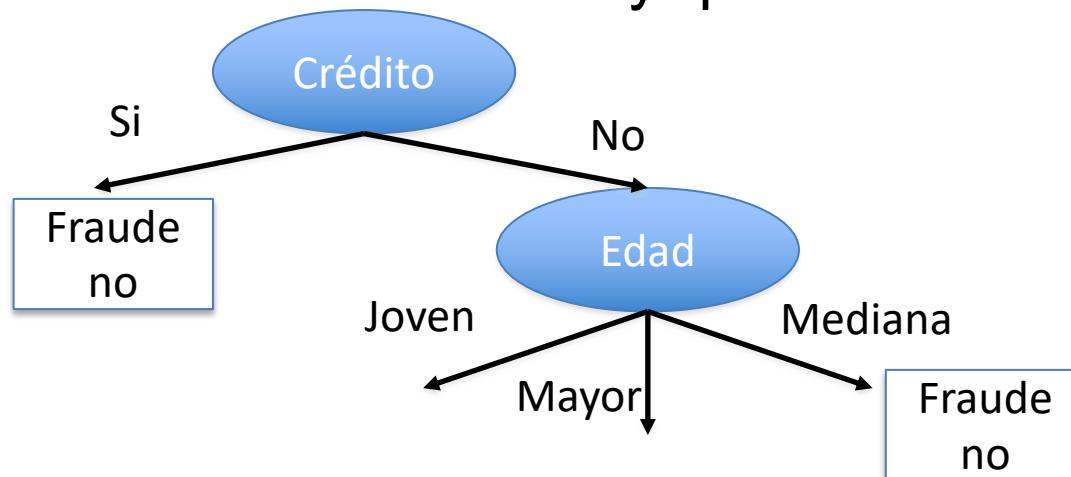
# Arboles de decisión

Ejemplo (fraude):

- 1) Los que tienen crédito pertenecen todos a la misma clase. Se le asigna un nodo hoja.

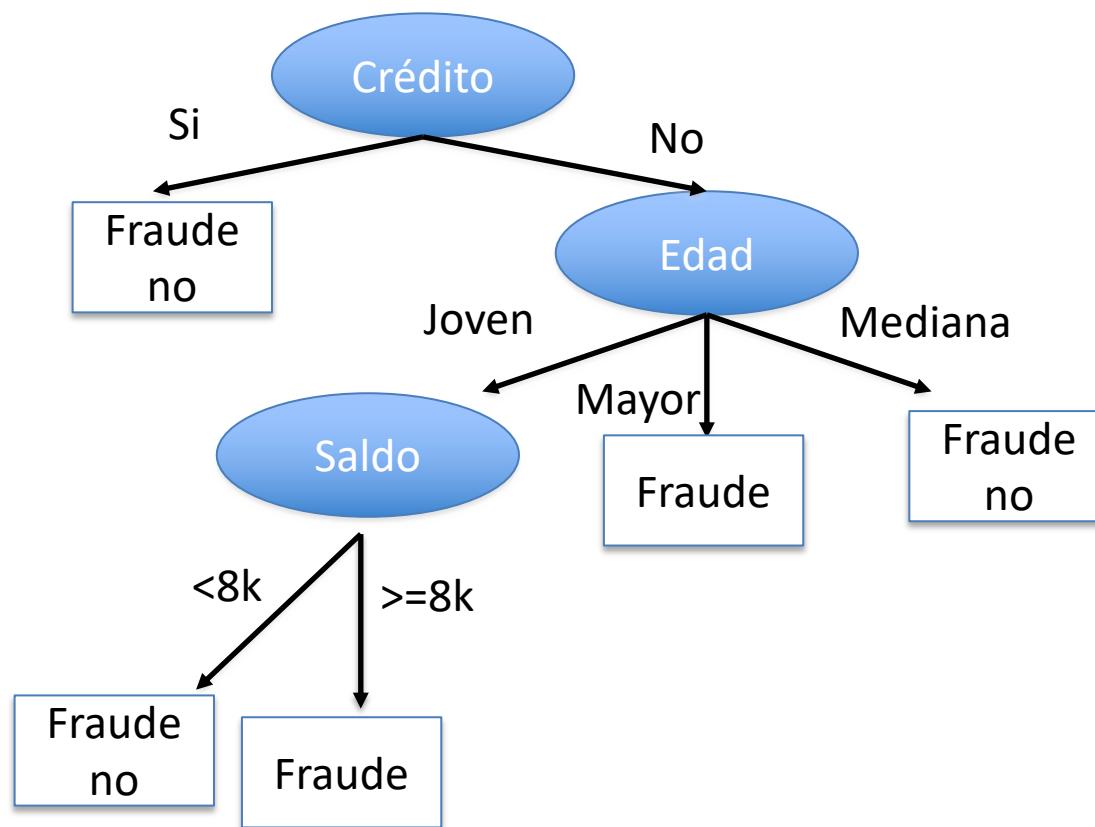


- 2) Para los que no tienen crédito hay que dividir en ramas.



# Arboles de decisión

3)



# Arboles de decisión – Criterios división

Decisiones a tomar:

¿Cómo dividir los registros?

¿Cómo especificar la condición del atributo?

¿Cómo encontrar la mejor división?

¿Cuándo parar la división?

Dependiendo del criterio para dividir el algoritmo se llama de una manera.

# Arboles de decisión – Criterios división

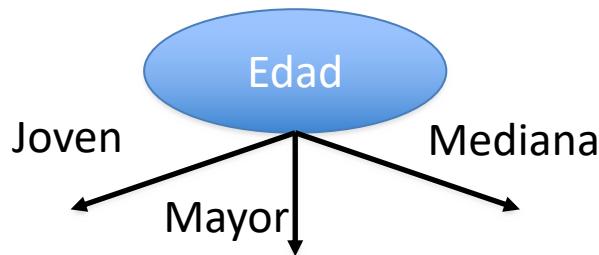
Depende de

- Tipo de dato:
  - Categórico: la condición tiene un valor concreto.
  - Ordinal: el orden de los hijos es relevante.
  - Continuo: la condición puede ser mayor o menor que.
- Tipo de árbol:
  - Binario: a veces obliga a agrupar valores por ramas.
  - Varios hijos: puede ocurrir que cada rama tenga un atributo.

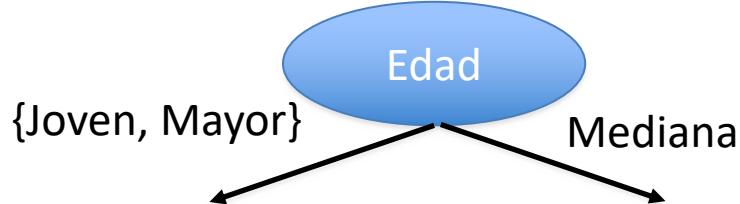
# Arboles de decisión – Criterios división

## División basada en atributos categóricos.

División múltiple: Usar varias particiones para valores distintos.



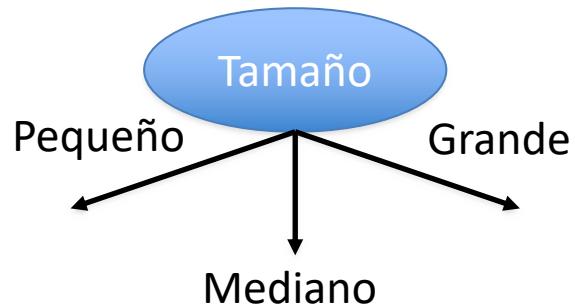
División binaria: Se emplean dos subconjuntos.



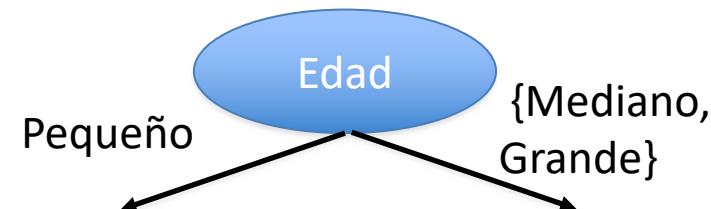
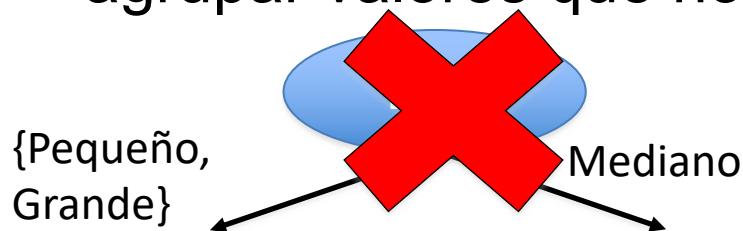
# Arboles de decisión – Criterios división

## División basada en atributos ordinales.

División múltiple: Usar tantas particiones como valores distintos.



División binaria: Se emplean dos subconjuntos, no se pueden agrupar valores que no sean seguidos.

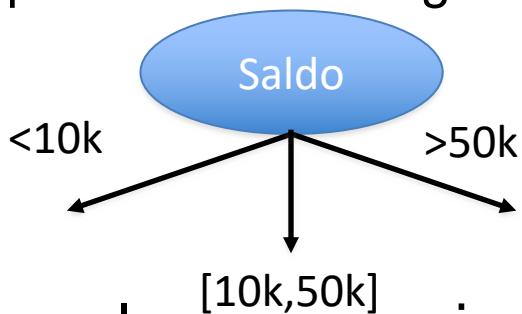


# Arboles de decisión – Criterios división

## División basada en atributos continuos.

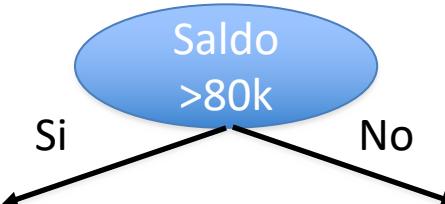
División múltiple: Discretización para conseguir un atributo ordinal.

- Estático: se discretiza al principio.
- Dinámico: se agrupan mediante rangos usando percentiles o clusters.



División binaria: Se emplean operaciones lógicas del tipo  $<$ ,  $\geq$  en el nodo.

- Muchas opciones dependiendo del valor de la condición.



# Arboles de decisión – Criterios división

Proporción: Número de casos de una clase entre el número de casos de la partición. Ej: Proporción>0,6

	Cliente	Clase
Normal	1	0
	3	1
	7	0
Bajo	Cliente	Clase
	2	1
	6	1
Alto	Cliente	Clase
	4	1
	5	0
	8	1

Para la clase 0

Partición	Nivel físico	Casos	Proporción
1	Normal	{1,3,7}	2/3
2	Bajo	{2,6}	0
3	Alto	{4,5,8}	1/3

# Arboles de decisión – Criterios división

- Medidas de desorden: Relacionada con la distribución de probabilidad que tiene un atributo. Se relaciona con la homogeneidad.
- Entropía (se calcula en el nodo actual): es la información esperada del atributo  $A_k$  en lo referente al conjunto de casos X. Es decir, el nivel de diversidad de este atributo en el conjunto de casos X. Cuando vale cero todos los registros pertenecen a la misma clase (totalmente homogénea).

$$E(t) = - \sum_{j=1}^k p(j)_t \log_2 p(j)_t$$

# Arboles de decisión – Criterios división

Ejemplo (6 individuos pertenecientes a 2 clases):

$$E(t) = - \sum_j p(j/t) \log_2 p(j/t)$$

C1	C2
0	6

$$\left. \begin{array}{l} p(C1/t)=0/6 \\ p(C2/t)=6/6 \end{array} \right\} E(t) = - 0/6 \log_2(0/6) - 6/6 \log_2(6/6) = 0$$

C1	C2
1	5

$$\left. \begin{array}{l} p(C1/t)=1/6 \\ p(C2/t)=5/6 \end{array} \right\} E(t) = - 1/6 \log_2(1/6) - 5/6 \log_2(5/6) = 0,65$$

C1	C2
2	4

$$\left. \begin{array}{l} p(C1/t)=2/6 \\ p(C2/t)=4/6 \end{array} \right\} E(t) = - 2/6 \log_2(2/6) - 4/6 \log_2(4/6) = 0,92$$

# Arboles de decisión – Criterios división

- Efectividad (se calcula para los nodos hijo): Cómo de efectivo es un atributo  $t$  al subdividir un conjunto de ejemplos en  $n$  subconjuntos (uno por cada posible valor de  $X$ ). Dicho de otra manera es el valor esperado de la entropía después de efectuar la partición.

$$Ef(t) = - \sum_{i=1}^n \frac{|E_i|}{|E|} * E(t)$$

- Ganancia de información: Propiedad estadística que mide cómo clasifica un atributo  $t$ . Mide en qué grado un atributo determinado hace aumentar o disminuir el desorden de las particiones que puede haber a partir de comparar el desorden que induce con respecto al existente en un momento determinado (se elige la mayor).

$$G(t) = E(t) - Ef(t, X)$$

# Arboles de decisión – Criterios división

- Error de cada regla: proporción de casos en la que la regla ha realizado una predicción erronea con respecto al total de casos que la cubría.

Regla 1, cubre 7 casos y predice mal 3 de ellos

$$\text{Error} = (3/7) = 0,42 = 42\%$$

- Error global (del modelo): la suma ponderada de los errores de todas las hojas del árbol. Es decir, el error de cada hoja multiplicado por la probabilidad de que un caso vaya a parar a la partición representada por la hoja.

$$\text{Error}_{\text{global}} = \sum_{i=1}^n W_i * \text{Error}_i$$

# Algoritmo ID3

## Algoritmo ID3 (Algoritmo Hunt + entropía)

Ejemplo (clientes gimnasio):

Cliente	Horario	Sexo	Nivel físico	Clase
1	Mañana	Hombre	Alto	1
2	Mañana	Mujer	Normal	1
3	Mediodía	Mujer	Normal	2
4	Tarde	Mujer	Normal	2
5	Tarde	Mujer	Alto	1
6	Mediodía	Mujer	Bajo	2
7	Tarde	Hombre	Bajo	2
8	Mañana	Mujer	Normal	1

# Algoritmo ID3

1) Calculamos la entropía para el conjunto inicial de datos:

$$E_{\text{inicial}} = \overbrace{-4/8 * \log_2(4/8)}^{\text{Clase 1}} + \overbrace{-4/8 * \log_2(4/8)}^{\text{Clase 2}} = 1$$

2) Medimos la efectividad para cada atributo:

$$\begin{aligned} E_f(X, \text{Sexo}) &= 2/8 * (-1/2 * \log_2(1/2) - 1/2 * \log_2(1/2)) \rightarrow \text{Hombres} \\ &+ 6/8 * (-3/6 * \log_2(3/6) - 3/6 * \log_2(3/6)) \rightarrow \text{Mujeres} \end{aligned}$$

Valores  
del  
atributo

De 8 individuos, 6 son mujeres

De esas 6 mujeres, 3 pertenecen a la clase 1 y 3 a la clase 2

# Algoritmo ID3

3) Para el resto de atributos:

$$Ef(X, \text{Horario}) = 3/8 * (-1 * \log_2 1 - 0 * \log_2 0) + 2/8 * (-0 * \log_2 0 - 1 * \log_2 1) + 3/8 * (-1/3 * \log_2 1/3 - 2/3 * \log_2 2/3) = 0,344$$

$$Ef(X, \text{Nivel_físico}) = 2/8 * (-1 * \log_2 1 - 0 * \log_2 0) + 4/8 * (-1/2 * \log_2 1/2 - 1/2 * \log_2 1/2) + 2/8 * (-0 * \log_2 0 - 1 * \log_2 1) = 0,5$$

4) Calculamos la ganancia y elegimos la mayor:

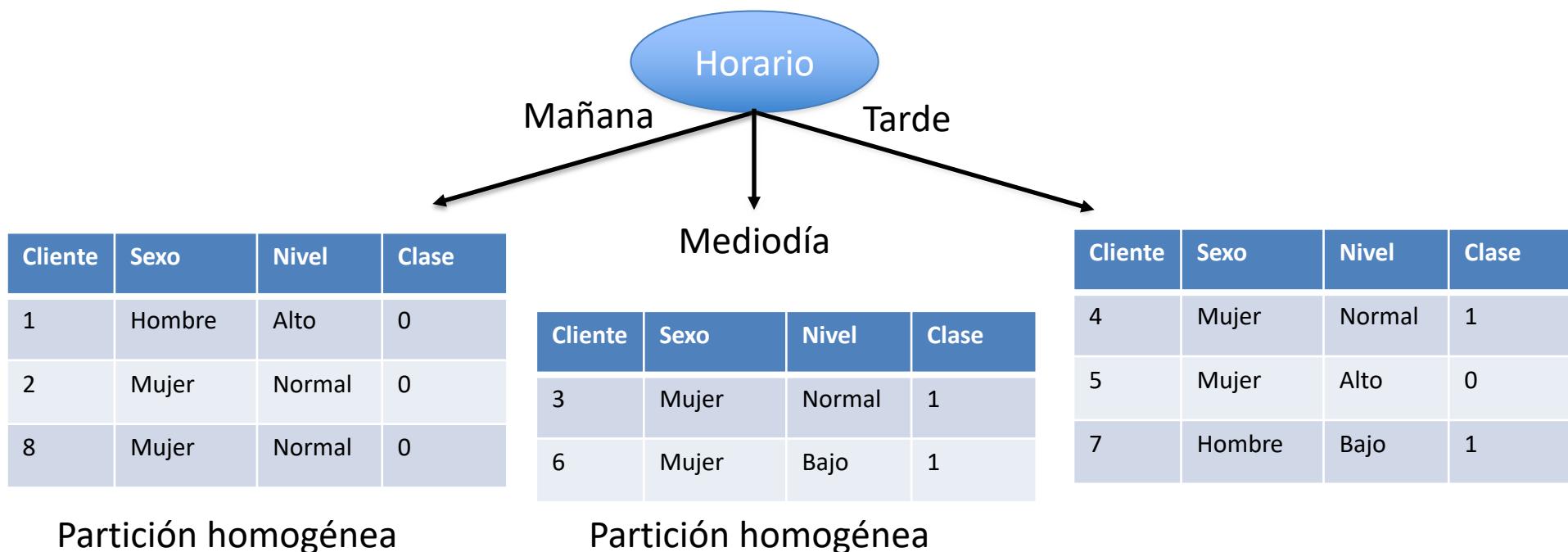
$$\text{Sexo} = 1 - 1 = 0$$

$$\text{Horario} = 1 - 0,344 = 0,656$$

$$\text{Nivel_físico} = 1 - 0,5 = 0,5$$

# Algoritmo ID3

5) Dividir el nodo elegido (Horario) y generar subtablas



# Algoritmo ID3

5) Calculamos la entropía para los nodos no homogéneos:

$$E_{\text{nivel}} = -1/3 \log_2(1/3) - 2/3 \log_2(2/3) = 0,918$$

6) Medimos la efectividad para cada atributo posible:

$$\begin{aligned} E_f(X, \text{Sexo}) &= 1/3 * (-0 * \log_2(0) - 1 * \log_2(1)) + \rightarrow \text{Hombres} \\ &\quad + 2/3 * (-1/2 * \log_2(1/2) - 1/2 * \log_2(1/2)) \rightarrow \text{Mujeres} \\ &= 0,666 \end{aligned}$$

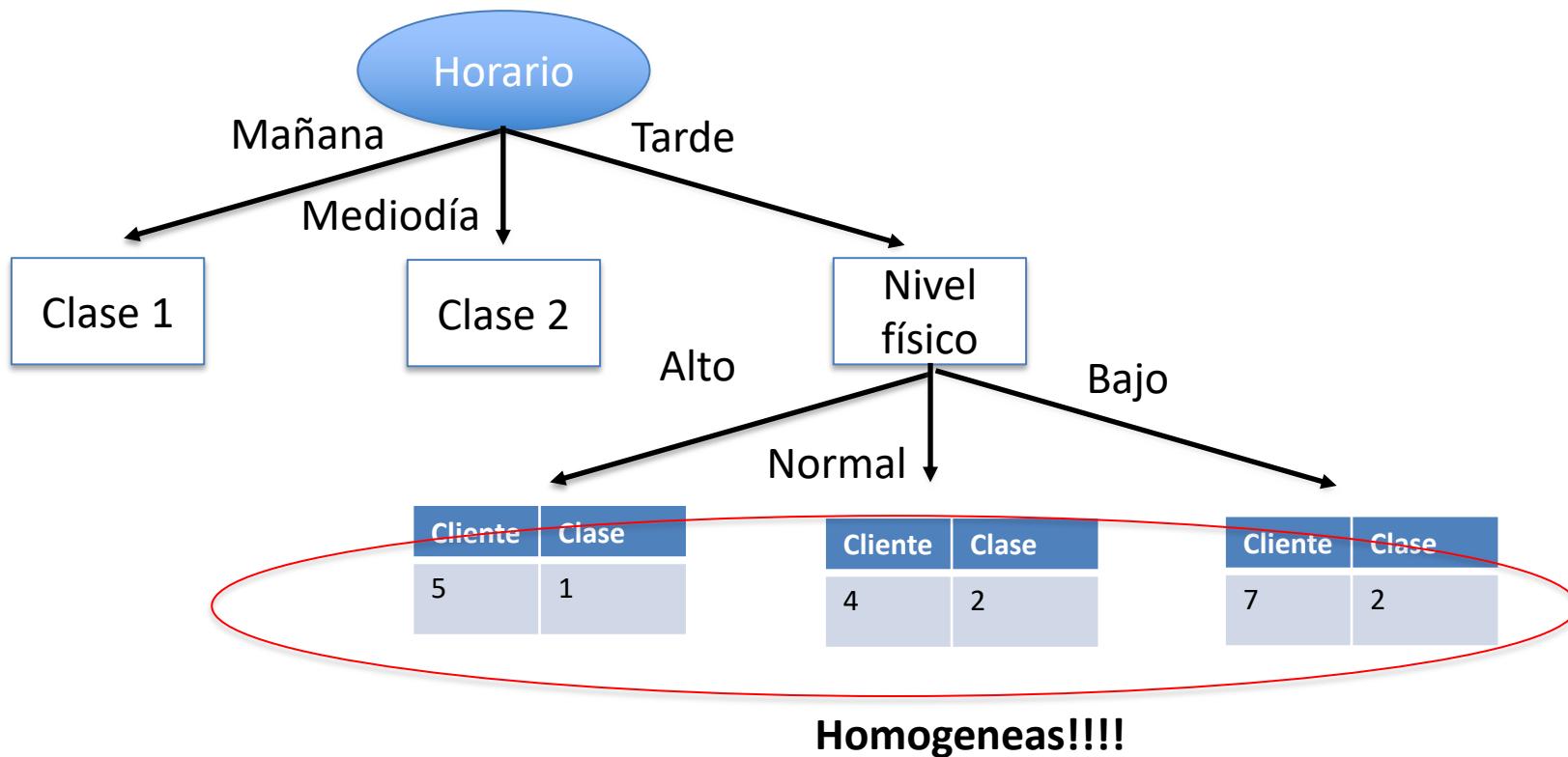
$$\begin{aligned} E_f(X, \text{Nivel\_fisico}) &= 1/3 * (-0 * \log_2(0) - 1 * \log_2(1)) + 1/3 * (-1 * \log_2(1) - \\ &0 * \log_2(0)) + 1/3 * (-0 * \log_2(0) - 1 * \log_2(1)) = 0 \end{aligned}$$

# Algoritmo ID3

7) Se calcula la ganancia y elegimos el más alto:

$$\text{Sexo} = 0,918 - 0,666 = 0,252$$

$$\text{Nivel_físico} = 0,918 - 0 = 0,918$$



# Algoritmo ID3

8) Obtener las reglas:

R1 Horario=“Mañana” → Clase 1

R2 Horario=“Mediodía” → Clase 2

R3 Horario=“Tarde” AND Nivel\_físico=“Alto” → Clase 1

R4 Horario=“Tarde” AND Nivel\_físico=“Normal” → Clase 2

R5 Horario=“Tarde” AND Nivel\_físico=“Bajo” → Clase 2

# Algoritmo ID3

9) Una vez obtenido el modelo. Se pasa a la fase de evaluación.  
Hay que contraponer el valor predicho con el valor real:

Cliente	Horario	Sexo	Nivel físico	Clase	Clase predicha	Regla
101	Mañana	Hombre	Alto	1	1	R1
324	Mañana	Hombre	Bajo	2	1	R1
5344	Mediodía	Hombre	Bajo	2	2	R2
23	Mañana	Mujer	Bajo	1	1	R1
28	Mañana	Hombre	Normal	2	1	R1
29	Mañana	Mujer	Bajo	1	1	R1
333	Mediodía	Mujer	Bajo	2	2	R2
442	Mañana	Mujer	Normal	2	1	R1
32	Mediodía	Hombre	Bajo	2	2	R2
112	Mediodía	Hombre	Normal	1	2	R2

# Algoritmo ID3

Cliente	Horario	Sexo	Nivel físico	Clase	Clase predicha	Regla
187	Mediodía	Hombre	Normal	1	2	R2
54	Tarde	Hombre	Bajo	2	2	R5
588	Tarde	Mujer	Alto	1	1	R3
6536	Mediodía	Hombre	Bajo	1	2	R2
72	Mañana	Hombre	Normal	1	1	R1
811	Tarde	Hombre	Normal	1	2	R4

# Algoritmo ID3

9) Para cada regla se calcula el error:

Regla	Casos cubiertos	Casos incorrectos	Error
1	7	3	0,42=42%
2	6	3	0,5=50%
3	1	0	0%
4	1	1	1=100%
5	1	0	0%

10) Por último se calcula el error global

$$\text{Error}_{\text{global}} = (7/16)*0,42 + (6/16)*0,5 + (1/16)*0 + (1/16)*1 + (1/16)*0 = 0,43$$

# Otros criterios de división

- Índice de GINI: cuando se divide un nodo p en k particiones y se quiere calcular la calidad de la partición.

$$GINI(t) = 1 - \sum_{j=1}^k \binom{j}{t}^2$$

Al aplicar el índice de GINI pasamos a usar el algoritmo CART. Aunque sólo se pueden generar arboles binarios.

- Error de clasificación:

$$Error(t) = 1 - max \binom{j}{t}$$

# Otros criterios de división

Ejemplo (GINI):

$$GINI(t) = 1 - \sum_{j=1}^k \binom{j}{t}^2$$

C1	C2
0	6

$$\left. \begin{array}{l} p(C1/t)=0/6 \\ p(C2/t)=6/6 \end{array} \right\} GINI(t) = 1 - (0/6)^2 -(6/6)^2 = 0$$

C1	C2
1	5

$$\left. \begin{array}{l} p(C1/t)=1/6 \\ p(C2/t)=5/6 \end{array} \right\} GINI(t) = 1 - (1/6)^2 -(5/6)^2 = 0,278$$

C1	C2
2	4

$$\left. \begin{array}{l} p(C1/t)=2/6 \\ p(C2/t)=4/6 \end{array} \right\} GINI(t) = 1 - (2/6)^2 -(4/6)^2 = 0,444$$

# Otros criterios de división

Ejemplo (Error de clasificación):

$$\text{Error}(t) = 1 - \max\left(\frac{j}{t}\right)$$

C1	C2
0	6

$$\left. \begin{array}{l} p(C1/t)=0/6 \\ p(C2/t)=6/6 \end{array} \right\} \text{Error}(t) = 1 - \max(0,1) = 0$$

C1	C2
1	5

$$\left. \begin{array}{l} p(C1/t)=1/6 \\ p(C2/t)=5/6 \end{array} \right\} \text{Error}(t) = 1 - \max(1/6,5/6) = 1 - 5/6 = 1/6$$

C1	C2
2	4

$$\left. \begin{array}{l} p(C1/t)=2/6 \\ p(C2/t)=4/6 \end{array} \right\} \text{Error}(t) = 1 - \max(2/6,4/6) = 1 - 4/6 = 2/6$$

# Árboles de decisión con error de clasif.

Ejemplo (clientes gimnasio):

Cliente	Nivel físico	Frec. Act. 1	Frec. Act. 2	Clase
1	Normal	Media	Baja	0
2	Bajo	Baja	Baja	1
3	Normal	Alta	Alta	1
4	Alto	Alta	Alta	1
5	Alto	Baja	Baja	0
6	Bajo	Media	Media	1
7	Normal	Media	Media	0
8	Alto	Alta	Alta	1

# Métodos de poda

Intentan conseguir un árbol que no tenga más niveles ni particiones de los que son necesarios para alcanzar un buen nivel de predicción.

- Método de prepoda: ahorra el proceso de construcción de los árboles poco prometedores y pregunta a cada nivel cuál es el valor de predicción que asegura el árbol parcialmente construido.
- Método de postpoda: consiste en construir primero el árbol, y después analizar la tasa de predicción que obtenemos cuándo eliminamos parte del árbol.

# Métodos de poda

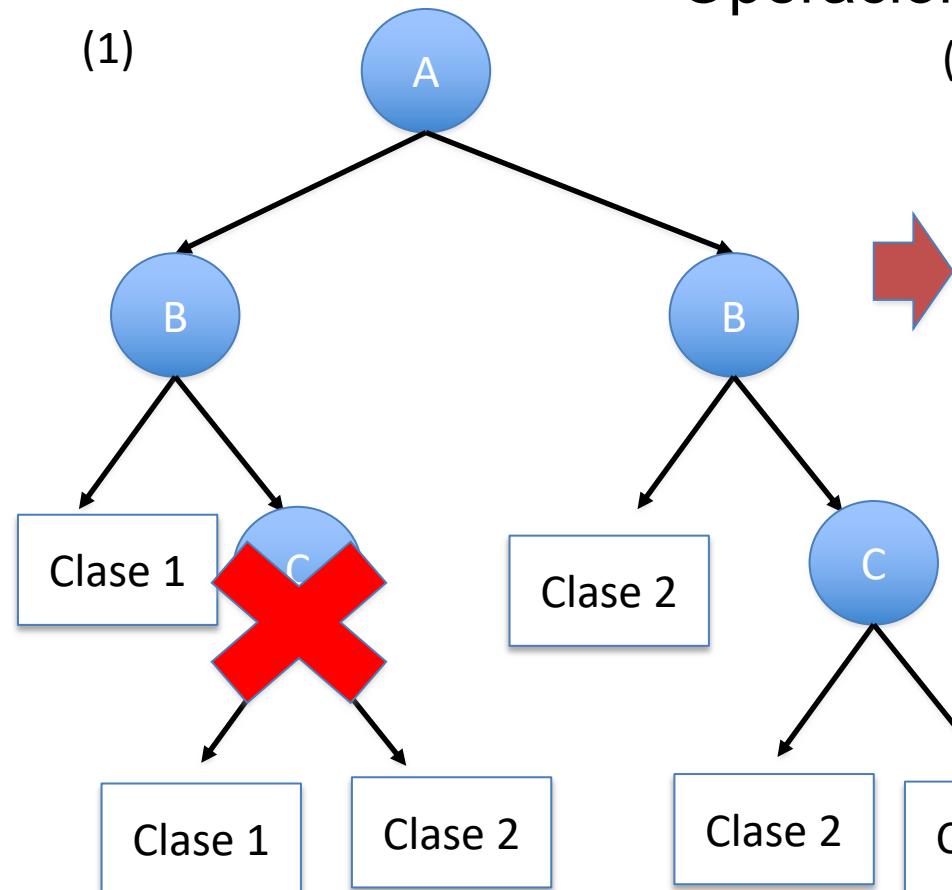
## Métodos de postpoda:

- La operación de sustitución. Consiste en analizar el subárbol a partir del nodo en el que el método efectúa la construcción y sustituirlo por un conjunto de hojas finales. Eso supone establecer las categorías correctas para etiquetar las ramas.
- La operación de promoción. Actúa en sentido contrario al de sustitución, en forma ascendente (de los nodos hoja hacia la raíz). Se detecta un subárbol que no es útil o interesante (predicción baja). Implica una reclasificación de ejemplos. Todos los ejemplos que se encontraban bajo el nodo original, han de ser ubicados bajo el nuevo.

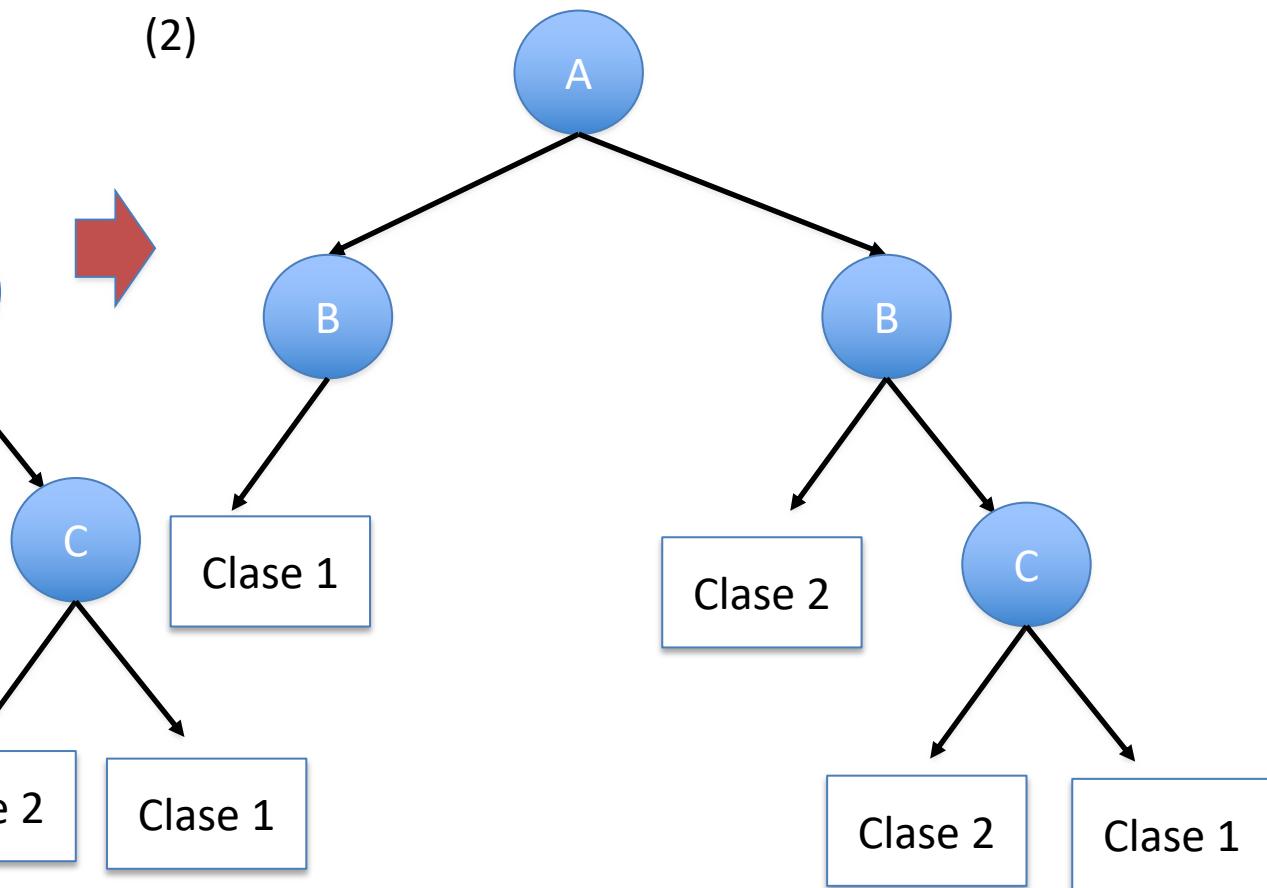
# Métodos de poda

Operación de sustitución.

(1)



(2)

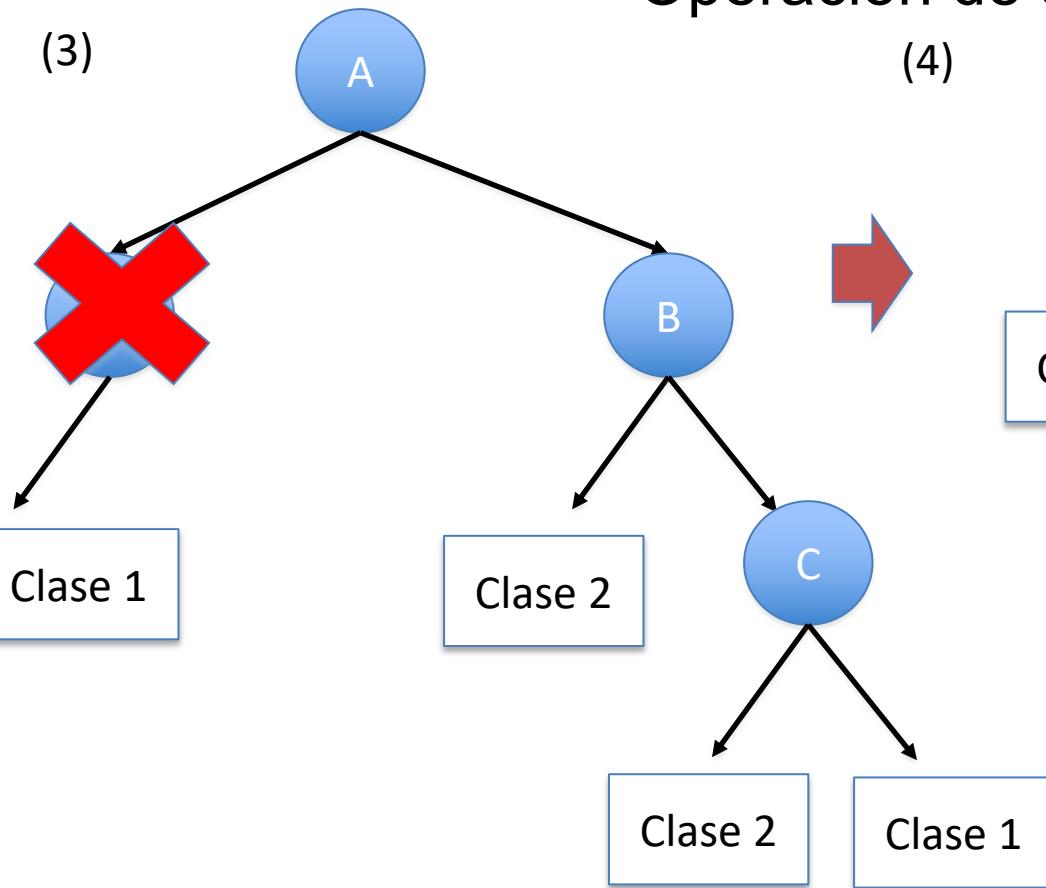


Mismas condiciones en B y C.

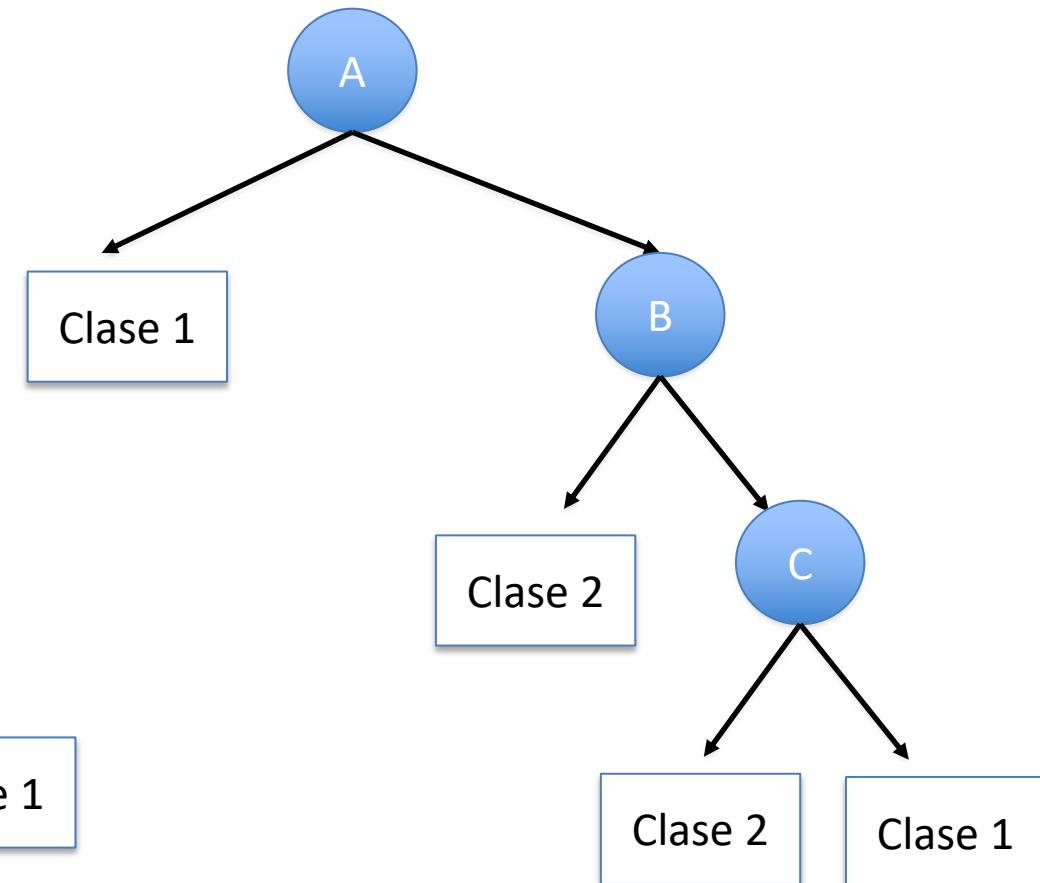
# Métodos de poda

Operación de sustitución.

(3)



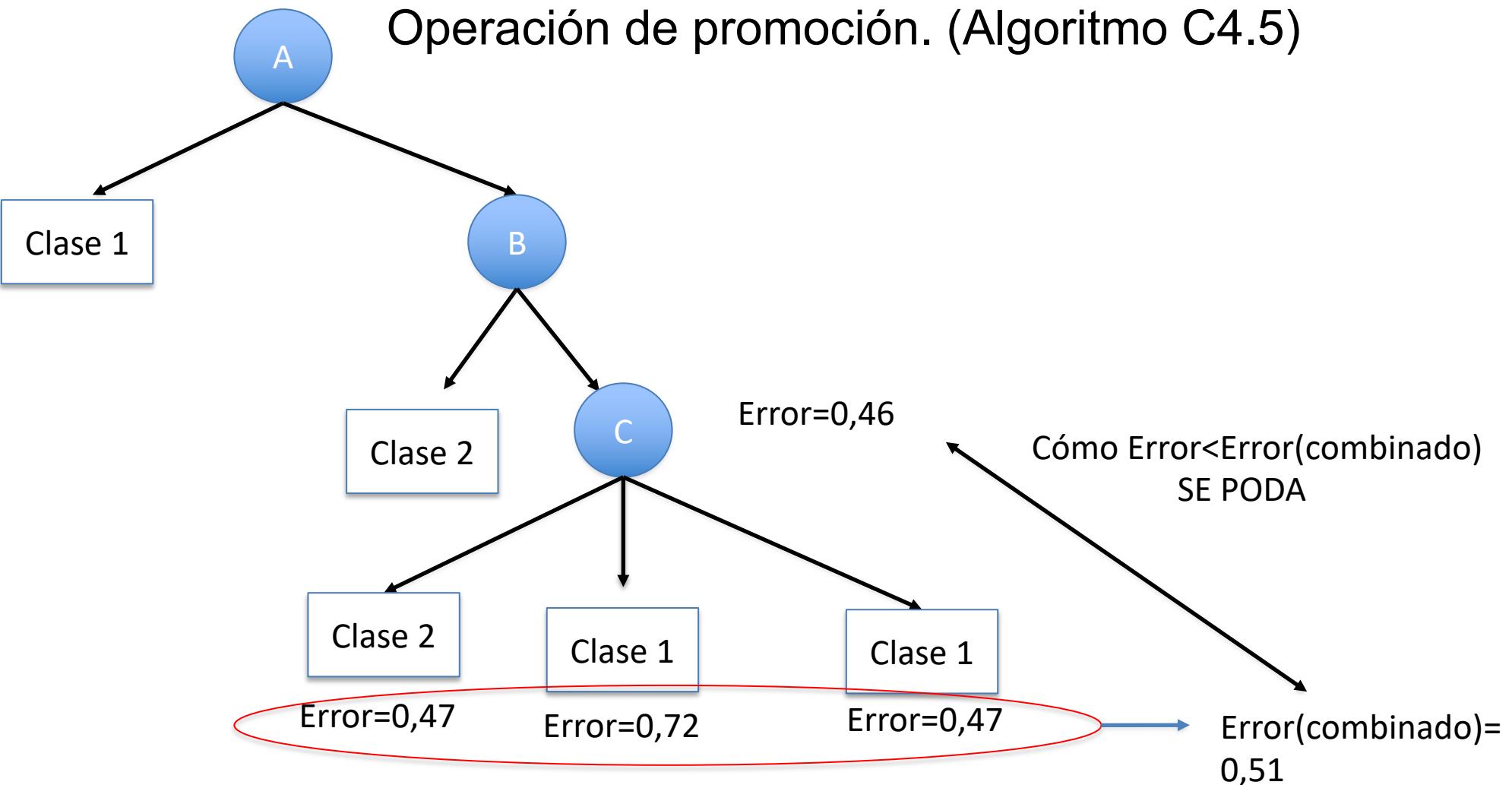
(4)



Operación de sustitución.

# Métodos de poda

Operación de promoción. (Algoritmo C4.5)

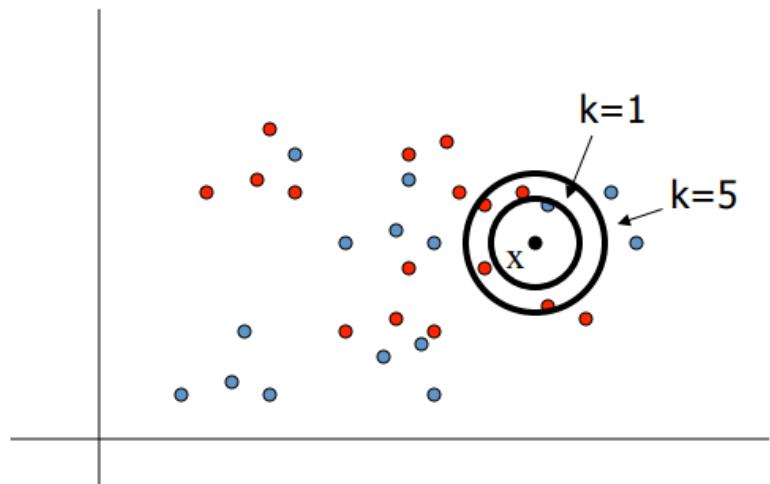


# K-Nearest-neighbors

- Objetivo
- Definiciones del problema
- Funcionamiento del algoritmo

# Definición del problema

- **Objetivo:** clasificar nuevos individuos con respecto a su distancia en un espacio en el que se ha distribuido un conjunto de entrenamiento teniendo en cuenta los atributos.



Dependiendo de la distancia de X al resto de elementos K, estos serán más parecidos o no (mayor o menor distancia). Se clasifica la clase en relación a los k vecinos más cercanos.

# Definiciones

Idea general:

1. Para un nuevo individuo. Se incluye en el espacio donde esté el conjunto de entrenamiento.
2. Se calcula la distancia del individuo nuevo con el resto de conjunto de entrenamiento.
3. Identifica los  $k$  más próximos.
4. Usa las clases de esos  $k$  vecinos para determinar la clase (por ejemplo la que más ocurra)

Hay que definir la métrica para medir la distancia, el  $k$  que se va a usar y el criterio para determinar la clase que se le asigna.

# Definiciones

Métricas para la distancia (para valores continuos):

- Distancia Euclídea: la distancia “ordinaria” entre dos puntos de un espacio euclídeo.

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

- Distancia de Manhattan: la suma de las diferencias absolutas de sus coordenadas.

$$\sum_{i=1}^k |x_i - y_i|$$

Métricas para la distancia (para valores categóricos):

- Distancia de Hamming:

$$D_H = \sum_{i=1}^k |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

# Definiciones

Ejemplo distancia de Hamming:

- Diferencia entre dos números binarios

X1	X2	X3	X4	X5	X6	X7	X8
0	1	0	0	0	1	0	1
0	1	1	0	0	1	0	0

Distancia 2

- Diferencia entre dos instancias:

Sexo	Profesión	Nacionalidad	Edad	Estudios
Mujer	Química	Española	Joven	Grado
Mujer	Informática	Sueca	Joven	Doctorado

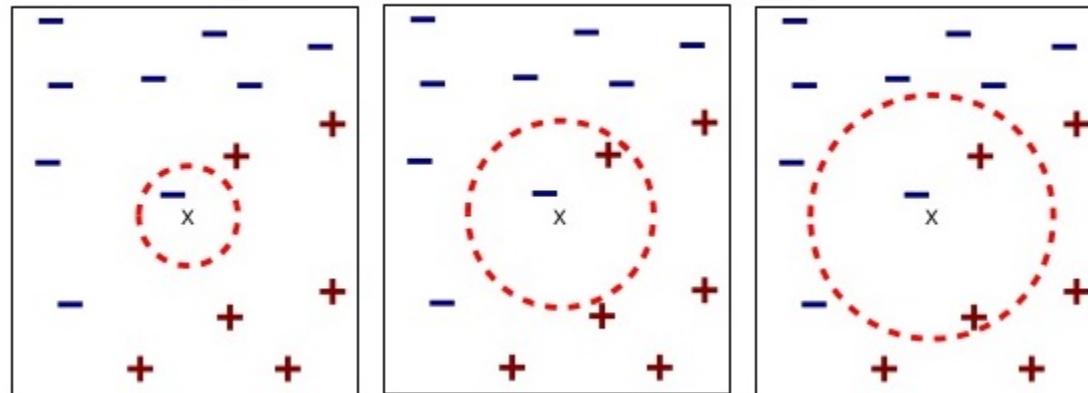
Distancia 3

# Definiciones

¿Métrica para  $k=1$ ?

¿Métrica para  $k=2$ ?

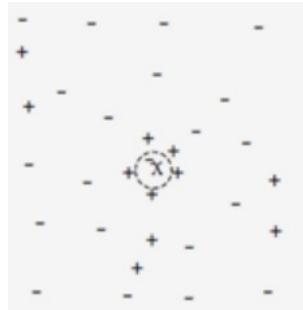
¿Métrica para  $k=3$ ?



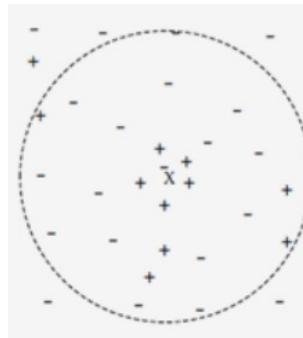
¿Qué pasa cuando  $k=\infty$ ?

# Definiciones

Si  $k$  es muy pequeño el modelo categoriza con pocos individuos. Será muy sensible a puntos que son atípicos o que están corruptos.

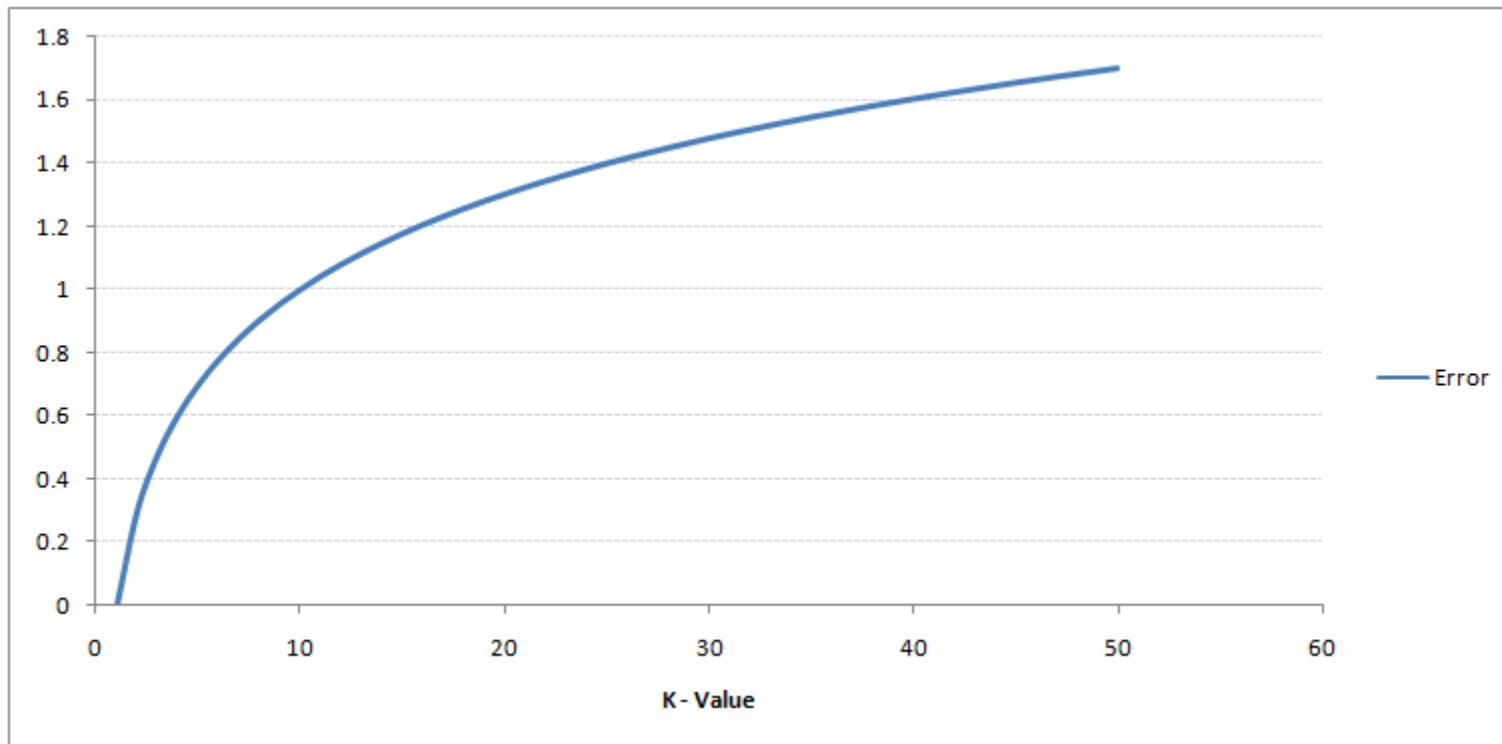


Si  $k$  es muy grande el modelo tiende a asignar siempre a la clase mas grande.



# Definiciones

Probar con  $k=1, k=2 \dots$  hasta encontrar un  $k$  bueno.  
Tiene un coste computacional alto.



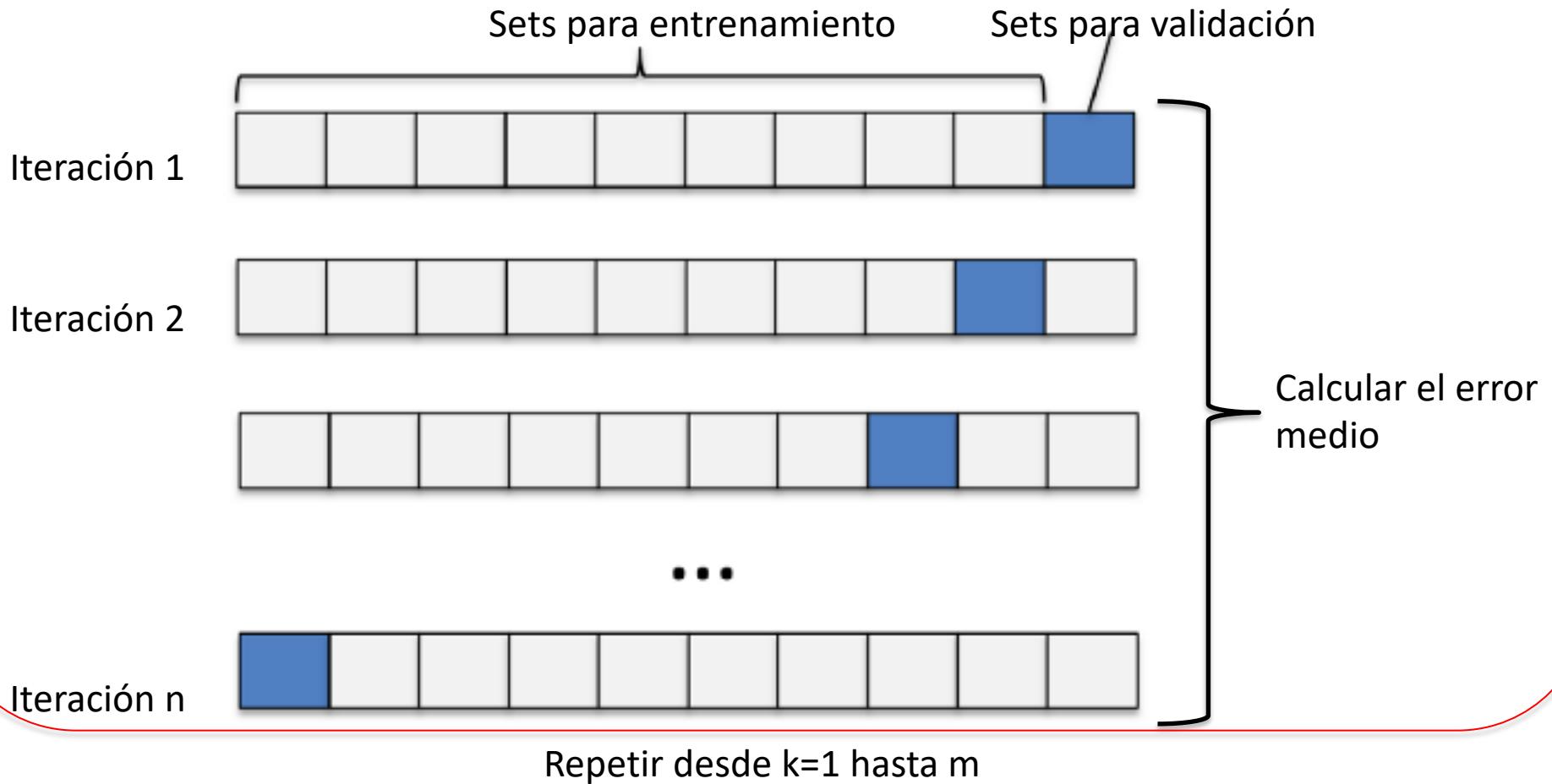
# Definiciones

Usar cross-validation para elegir k:

- 1) Dividir los datos en n subconjuntos. Por ejemplo 5
- 2) Utilizar n-1 para entrenar y el que queda para evaluar.
- 3) Calcular el error cometido al clasificar.
- 4) Iterar este proceso cambiando el subconjunto usado para evaluar. Hasta que todos hayan evaluado el modelo.
- 5) Calcular el error medio de cada una de las iteraciones anteriores.
- 6) Repetir esta estrategia para los k que se quieran medir.
- 7) Escoger el k que tenga el mejor error. Si hay empate, elegir aquel que agrupe más individuos.

# Definiciones

## Usar cross-validation para elegir k



# Definiciones

Con Euclídea entonces  $k=4$ . Con Manhattan  $k=5$  por agrupar más individuos.

Error=(Número de registros mal clasificados/Número de registros evaluados)

K	Error con d Euclídea	Error con d Manhattan
1	10,2	10,7
2	10,2	10,7
3	10,1	10
4	10	10
5	10,5	10
6	10,8	10,1
7	11,1	10,3
8	11,3	10,4
9	11,6	10,6
10	11,8	10,7

# Definiciones

¿Cómo elegir la clase dentro de k?

- Escoger el que más ocurrencias tenga. Si hay empate:
  - 1) Seleccionar la clase con el vecino más cercano.
  - 2) Seleccionar la clase con distancia media menor.
  - 3) Crear métricas propias.
- Elegir k impar para que nunca haya empate.

# K-nearest-neighbors

Algoritmo k-nearest-neighbors:

Sea  $X_n = (A_1 \dots A_n)$  un nuevo individuo a clasificar

Para cada individuo del set de entrenamiento  $(X_1, c_1) \dots (X_n, c_n)$

Calcular la distancia  $d_i(X_i, X_n)$

Ordenar  $d_i$  en orden ascendente

Obtener los  $k$  casos más cercanos a  $X_n$

Asignar a  $X_n$  la clase correspondiente al aplicar la métrica elegida (por ejemplo la más frecuente)

# K-nearest-neighbors

Ejemplo (flores):

1) Dado un conjunto de entrenamiento:

Largo sépalo	Ancho sépalo	Largo pétalo	Ancho pétalo	Clase
5,1	3,5	1,4	0,2	Setosa
4,0	3,0	1,4	0,2	Setosa
4,7	3,0	1,3	0,2	Setosa
7,0	3,2	4,7	1,4	Versicolor
3,4	3,2	4,5	1,5	Versicolor
6,9	3,1	4,9	1,5	Versicolor

# K-nearest-neighbors

2) Clasificar un nuevo individuo (5,0 3,3 1,4 0,2), se calcula la distancia Euclídea a cada individuo del set de entrenamiento:

$$D(X_1, X) = \sqrt{(5,0 - 5,0)^2 + (3,3 - 3,3)^2 + (1,4 - 1,4)^2 + (0,2 - 0,2)^2} = 0,22$$

$$D(X_2, X) = \sqrt{(4,0 - 5,0)^2 + (3,0 - 3,3)^2 + (1,4 - 1,4)^2 + (0,2 - 0,2)^2} = 1,04$$

$$D(X_3, X) = \sqrt{(4,7 - 5,0)^2 + (3,0 - 3,3)^2 + (1,3 - 1,4)^2 + (0,2 - 0,2)^2} = 0,43$$

$$D(X_4, X) = \sqrt{(7,0 - 5,0)^2 + (3,2 - 3,3)^2 + (4,7 - 1,4)^2 + (1,4 - 0,2)^2} = 4,04$$

$$D(X_5, X) = \sqrt{(3,4 - 5,0)^2 + (3,2 - 3,3)^2 + (4,5 - 1,4)^2 + (1,5 - 0,2)^2} = 3,72$$

$$D(X_6, X) = \sqrt{(6,9 - 5,0)^2 + (3,1 - 3,3)^2 + (4,9 - 1,4)^2 + (1,5 - 0,2)^2} = 4,19$$

# K-nearest-neighbors

3) Para k=1

$D(X_1, X) = 0,22 \rightarrow \text{Setosa}$

4) Para otro individuo

$D_1 = 3,4 \ D_2 = 3,7 \ D_3 = 0,2 \ D_4 = 0,4 \ D_5 = 0,3 \ D_6 = 1,4$

Para k=1 → Setosa

Para k=3 → Versicolor

# K-nearest-neighbors

Ejercicio ¿Cuál es el mejor k=1, 2 o 3 para (0,5 y 0,7)

Clase	x1	x2
1	0.3	0.6
1	0.2	0.8
2	0.5	0.5
1	0.2	0.6
2	0.7	0.8
1	0.3	0.9
2	0.7	0.5

# K-nearest-neighbors

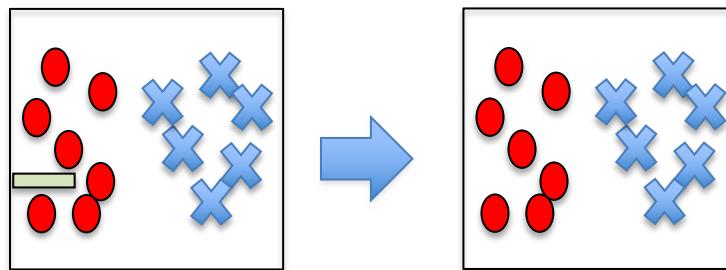
Algoritmo k-weighted-nearest-neighbors:

Es una modificación de k-nn donde los vecinos más cercanos tienen mas peso. Para ello podemos usar de métrica la inversa de la distancia ( $w=1/d$ ):

$$\hat{f}(x_q) = \sum_{i=1,k} w_i f(x_i) / \sum_{i=1,k} w_i$$

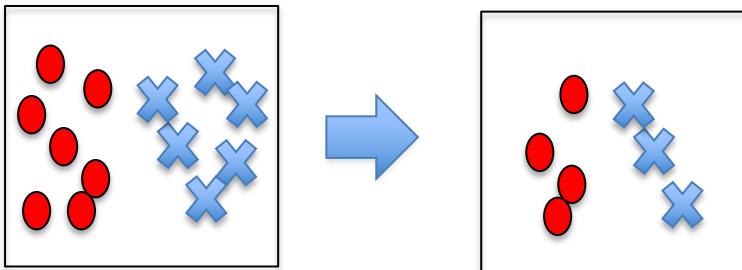
# Selección de instancias

Hay instancias que generan ruido (o solapan clases): confunden al clasificador. Al eliminarlas, mejorará el porcentaje de aciertos.



Editing: elimina unas pocas.  
Las que están rodeadas por  
otra clase

Hay instancias superfluas. Aquellas que no son necesarias para clasificar. Al eliminarlas, el tiempo de clasificación será menor.

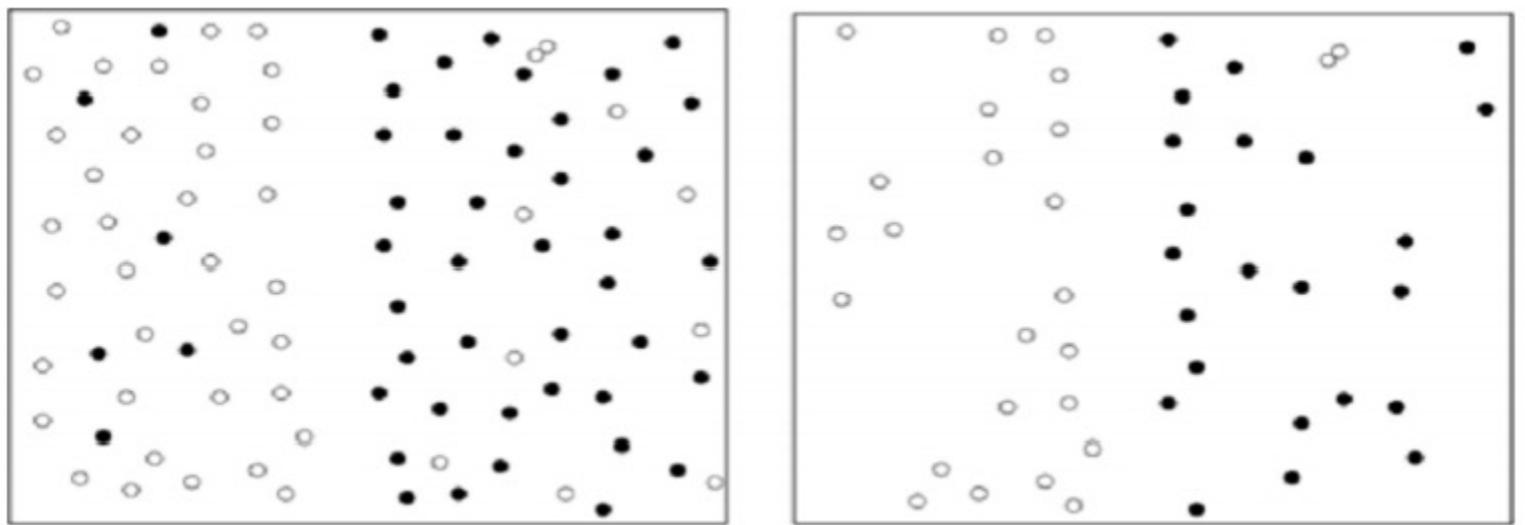


Condensación: se pueden eliminar  
muchas. Las que están en el interior  
dejando las de las fronteras.

# Selección de instancias

Wilson editing: elimina la instancia  $X_i$  si es clasificada incorrectamente por sus vecinos. Estas serán las excepciones en el interior de una clase. Algunos puntos de la frontera.

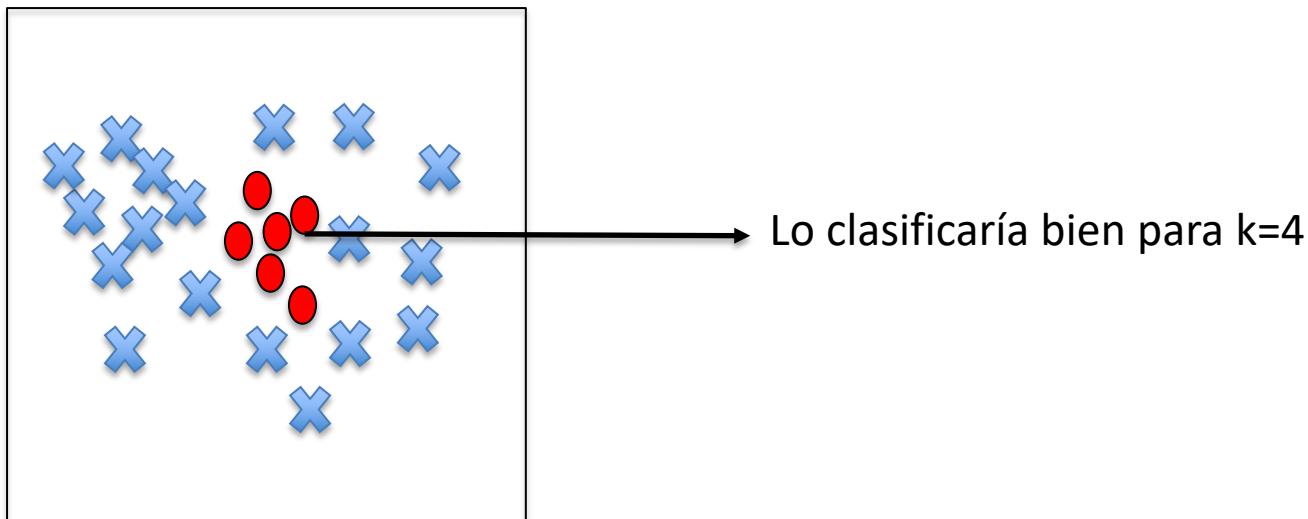
Se cogerá el set de entrenamiento. Se clasificarán todas las instancias y se borrarán aquellas que no se hayan clasificado bien.



# Selección de instancias

No elimina demasiados datos. Por lo tanto la mejora no es muy grande.

Funciona bien cuando no hay mucho ruido. En caso contrario, las instancias con ruido pueden clasificarse bien teniendo alrededor instancias de ruido de la misma clase.



# Selección de instancias

Condensación de Hart: intenta reducir el número de instancias, eliminando las superfluas. Va recorriendo las instancias en el orden en que se encuentran en el dataset, y se clasifica (usando las anteriores a la actual), si está bien clasificada no la guarda. Sólo guarda aquellas que no se clasifican bien con las ya existentes.

Para entrenar se usará el conjunto de datos generados por aquellas que no se clasifican bien.

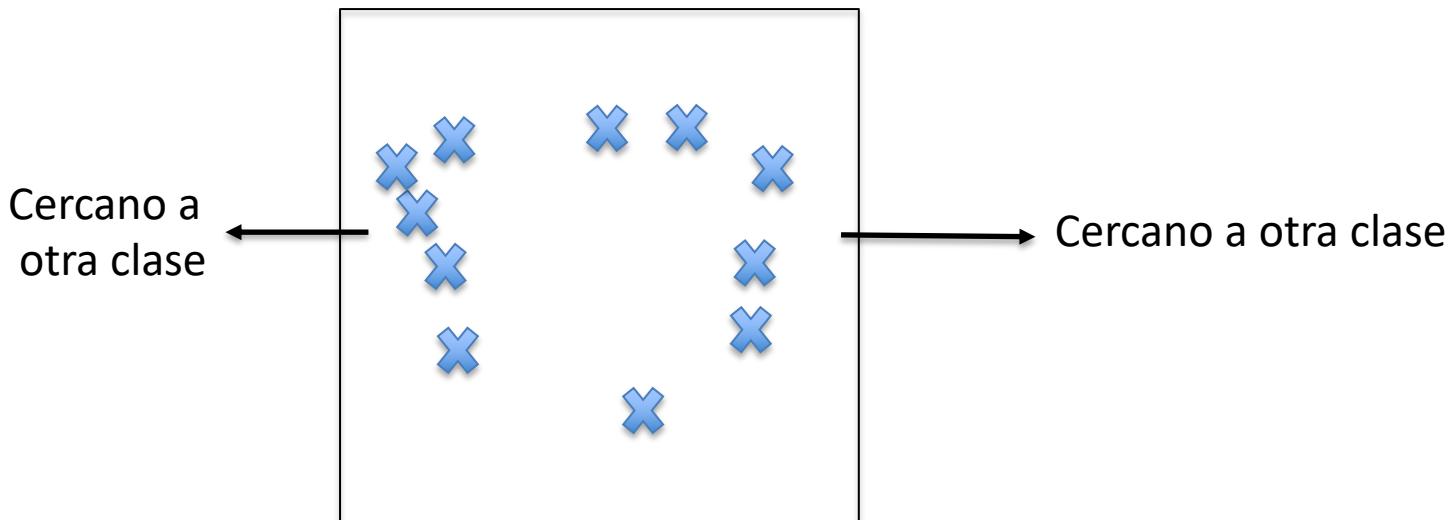
# Selección de instancias

- 1) Inicializa el nuevo conjunto con  $X_1$  (el primer elemento del set de entrenamiento)
- 2) Usa K-nn para clasificar  $X_2$ . Si se clasifica mal, moverlo al nuevo conjunto.
- 3) Entrenar el clasificador con el nuevo set de entrenamiento.
- 4) Volver a 2 para  $X_3$  y resto de instancias hasta que no se muevan mas instancias al nuevo conjunto.
- 5) Para clasificar con K-nn, usar el nuevo conjunto de datos.

# Selección de instancias

Elimina todas las instancias no críticas, reduciendo la necesidad de almacenamiento. Conserva aquellas instancias que tienen más ruido. Son las instancias que no han sido bien clasificadas y estarán en la frontera.

El orden de las instancias depende del resultado final.



# Clasificadores Bayesianos

- Objetivo
- Definiciones del problema
- Teorema de Bayes
- Clasificadores Bayesianos

# Definiciones

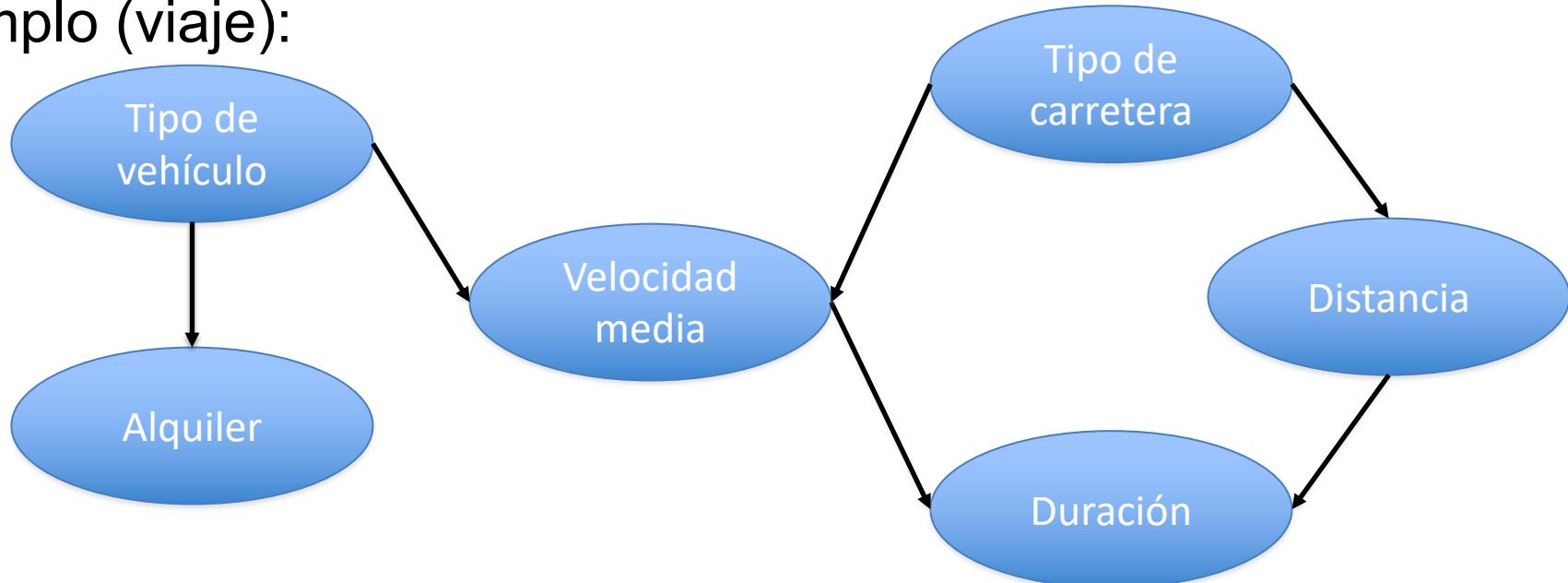
Grafo dirigido acíclico: grafo en el que la dirección de los enlaces es relevantes en la que nunca puede suceder que en un camino entre dos nodos el inicial o el final, estén repetidos.

Red bayesiana: grafo dirigido acíclico cuyos nodos representan las variables  $X_i$  del dominio (atributos) donde cada variable es independiente del resto de las variables  $X_p \dots X_j$  del dominio dados sus predecesores directos.

Un enlace entre dos variables  $X_i$  e  $X_j$  del dominio representa una asociación directa entre los dos; es decir,  $X_i$  influye sobre  $X_j$ .

# Definiciones

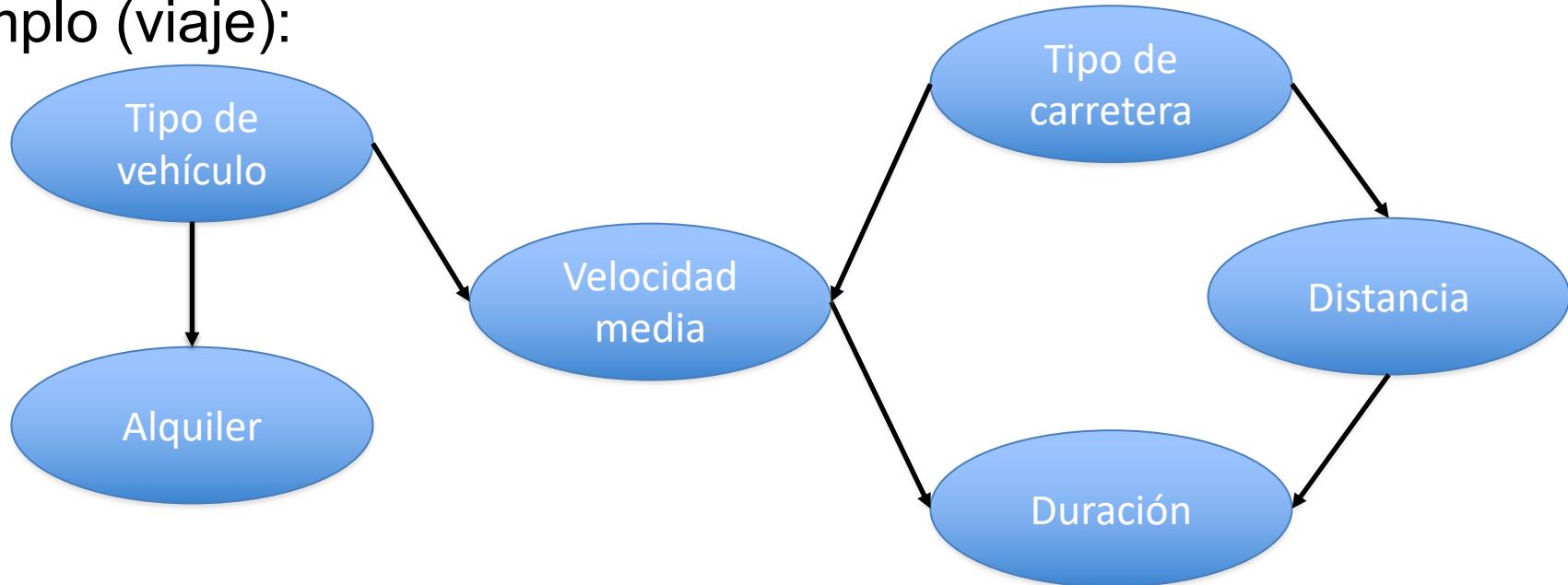
Ejemplo (viaje):



Interpretación: el alquiler depende del tipo de vehículo. La velocidad media a la que se puede hacer el viaje dependen del tipo de vehículo y carretera. La distancia también depende del tipo de carretera. Finalmente la duración del viaje depende de esta última y de la velocidad media.

# Definiciones

Ejemplo (viaje):



Es un grafo dirigido y acíclico. Los arcos entre nodos tienen dirección. No existe un arco que salga de un nodo y termine en el mismo nodo.

Enlace: Tipo\_de\_vehiculo → Alquiler

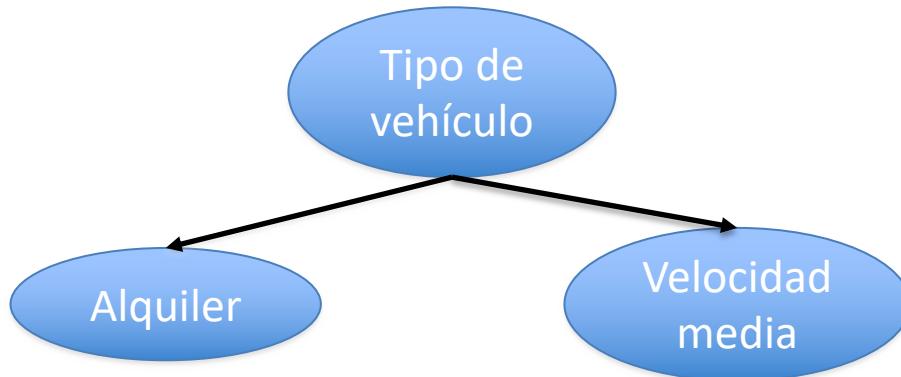
# Definiciones

Criterio de d-separación: permite leer las relaciones de independencia condicional entre las diferentes variables de un grafo.



- Conexión secuencial: el primer atributo tiene influencia sobre el segundo que la puede propagar al tercero. El primer atributo y el tercero son independientes gracias al segundo. Están d-separados por “Velocidad\_media”

# Definiciones



- Conexión divergente: la influencia puede pasar por todos los descendientes de “Tipo\_de\_vehiculo”. Están d-separados por él.

# Definiciones

Probabilidad condicionada: Sean A y B eventos tales que la  $P(B)>0$ , la probabilidad condicionada es la probabilidad de que ocurre A dado que sucede B.

- A priori: probabilidad de A antes de realizar el experimento.  $P(A)$
- A posteriori: probabilidad de A cuando se realiza el experimento.  $P(A|B)$

Ejemplo (fabrica): la probabilidad de que una pieza la fabrique la maquina A (a priori). La probabilidad de que una pieza defectuosa la fabrique la maquina A (a posteriori).

# Definiciones

Teorema de Bayes: dada una hipótesis ( $H$ ) y una evidencia ( $E$ ), se verifica la siguiente relación:

$$P(H|E) = P(E|H)P(H)/P(E)$$

La probabilidad de que dada una hipótesis haya una evidencia. Por la probabilidad de la hipótesis. Todo ello dividido entre la probabilidad de la evidencia.

# Definiciones

Ejemplo (pacientes de meningitis):

Probabilidad condicionada: La meningitis causa rigidez de cuello en el 50% de los casos. ( $R|M$ )

Hipótesis: La probabilidad de tener meningitis es 1/50.000. ( $M$ )

Evidencia: La probabilidad de sentir rigidez en el cuello de 1/20. ( $R$ )

¿Cuál es la probabilidad de que si un paciente siente rigidez en el cuello tenga meningitis?

$$P(M|R) = P(R|M) * P(M) / P(R) = 0,5 * (1/50.000) / (1/20) = 0,0002$$

# Clasificación basada en redes bayesianas

1) Calcular la probabilidad de  $P(C|A_1, A_2 \dots A_{n-1}, A_n)$  para todos los posibles valores de C empleando el teorema de Bayes:

$$P(C|A_1, A_2 \dots A_{n-1}, A_n) = P(A_1, A_2 \dots A_{n-1}, A_n|C) * P(C) / P(A_1, A_2 \dots A_{n-1}, A_n)$$

2) Escoger los valores de C que maximicen  $P(C|A_1, A_2 \dots A_{n-1}, A_n)$

# Clasificación basada en reglas

Ejemplo (Decidir tipo de lentes):

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended Lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

# Clasificación basada en redes bayesianas

Para los atributos siguientes. Está la siguiente distribución de clases.

## Edad

	Ninguna	Blandas	Duras
Joven	4	2	2
Prepresbicia	5	2	1
Presbicia	6	1	1

# Clasificación basada en redes bayesianas

## Diagnóstico

	Ninguna	Blandas	Duras
Miope	7	2	3
Hipermétrope	8	3	1

## Astigmatismo

	Ninguna	Blandas	Duras
Sí	8	0	4
No	7	5	0

## Lágrimas

	Ninguna	Blandas	Duras
Normal	3	5	4
Reducida	12	0	0

# Clasificación basada en redes bayesianas

## Edad

	Ninguna	Blandas	Duras
Joven	4/15	2/5	2/4
Prepresbicia	5/15	2/5	1/4
Presbicia	6/15	1/5	1/4

## Diagnóstico

	Ninguna	Blandas	Duras
Miope	7/15	2/5	3/4
Hipermétrope	8/15	3/5	1/4

## Astigmatismo

	Ninguna	Blandas	Duras
Sí	8/15	0/5	4/4
No	7/15	5/5	0/4

## Lágrima

	Ninguna	Blandas	Duras
Normal	3/15	5/5	4/4
Reducida	12/15	0/5	0/4

# Clasificación basada en redes bayesianas

Para un cliente con los siguientes valores en los atributos:

{“Joven”, “Hipermétrope”, “No astigmatismo”, “Lágrima normal”}

OJO!!!  $P(E)$  no es relevante porque todos se dividen por el mismo valor.

$$P(\text{Blandas}|E) = \frac{P(\text{Joven}|\text{Blandas}) * P(\text{Hipermétrope}|\text{Blandas}) * P(\text{No}|\text{Blandas}) * P(\text{Normal}|\text{Blandas}) * P(\text{Blandas})}{P(E)}$$

$$P(\text{Blandas}|E) = (2/5) * (3/5) * (5/5) * (5/5) * (5/24)$$

$$P(\text{Blandas}|E) = 0,05$$

$$P(\text{Ninguna}|E) = \frac{P(\text{Joven}|\text{Ninguna}) * P(\text{Hipermétrope}|\text{Ninguna}) * P(\text{No}|\text{Ninguna}) * P(\text{Normal}|\text{Ninguna}) * P(\text{Ninguna})}{P(E)}$$

$$P(\text{Ninguno}|E) = (4/15) * (8/15) * (7/15) * (3/15) * (15/24)$$

$$P(\text{Ninguno}|E) = 0,00829$$

$$P(\text{Duras}|E) = 0$$

# Clasificación basada en redes bayesianas

Como  $P(E)$  es una constante y costosa de calcular:

$$P(\text{Blandas}) * P(\text{Joven}|\text{Blandas}) * P(\text{Hiperm\u00e9trope}|\text{Blandas}) * P(\text{No}|\text{Blandas}) * P(\text{Normal}|\text{Blandas}) + \\ P(\text{Ninguna}) * P(\text{Joven}|\text{Ninguna}) * P(\text{Hiperm\u00e9trope}|\text{Ninguna}) * P(\text{No}|\text{Ninguna}) * P(\text{Normal}|\text{Ninguna}) + \\ P(\text{Duras}) * P(\text{Joven}|\text{Duras}) * P(\text{Hiperm\u00e9trope}|\text{Duras}) * P(\text{No}|\text{Duras}) * P(\text{Normal}|\text{Duras})$$

Los valores están desnormalizados, habría que normalizarlos.

Aplicamos una regla de 3.

# Clasificación basada en redes bayesianas

La suma de las P anteriores es  $0,05+0,00829+0=0,05829$

Si  $0,05829$  es 1 (100%) y  $P(\text{Blandas}|E)=0,05$ . Entonces  $P(\text{Blandas}|E)=0,8578$

En el caso de  $P(\text{Ninguno}|E)=0,1422$

En el caso de  $P(\text{Duras}|E)=0$

Si sumamos todo  $0,84+0,15+0=0,99\approx 1$

Por lo tanto corresponde a los porcentajes de probabilidad: Blandas un 85,78% Ninguna un 14,22% y Duras un 0%.

Se recomendaría usar Blandas a este cliente.

# Clasificación basada en redes bayesianas

Ejemplo (dar préstamo):

Para {Hombre, Baja, Joven}

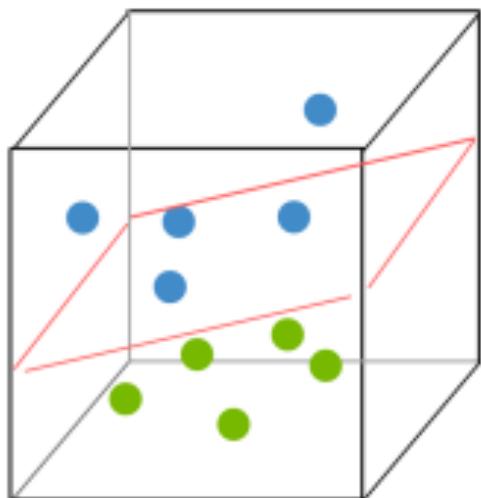
Sexo	Renta	Edad	Préstamo
Mujer	Alta	Media	No
Hombre	Media	Media	No
Mujer	Baja	Joven	No
Hombre	Media	Joven	No
Hombre	Alta	Mayor	Sí
Hombre	Baja	Mayor	No
Mujer	Baja	Media	Sí
Mujer	Media	Media	Sí

# Support Vector Machines

- Objetivo
- Definiciones del problema
- Suport Vector Machines

# Definición del problema

- **Objetivo:** buscar un hiperplano que separe un conjunto de datos en distintas clases.



Dependiendo de la inclinación del  
plano ( $w$ ) y la posición en el eje y(  $b$ )

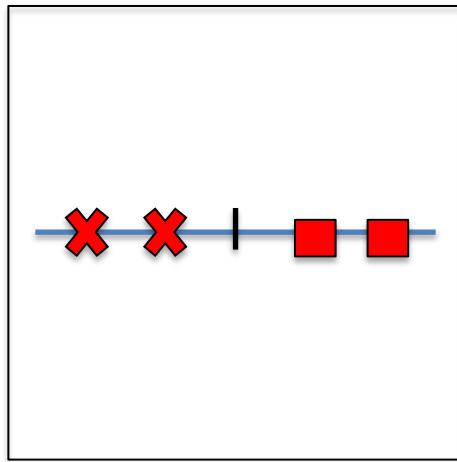
# Definiciones

- Dado un conjunto de individuos  $S=\{X_1, Y_1 \dots X_n, Y_n\}$ 
  - $X_i=\{i_1 \dots i_n\}$  es un vector formado por los atributos que definen a un individuo.
  - $Y_n$  la clase a la que pertenece el individuo  $\{-1, 1\}$  en el caso binario.
- Se representarán  $S$  en un conjunto de  $n$  dimensiones, una por cada atributo y se buscará el plano de dimensión  $n-1$ .

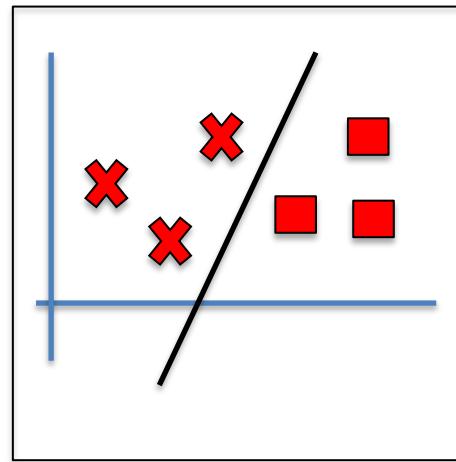
# Definiciones

El número de características del espacio fijará el número de dimensiones.

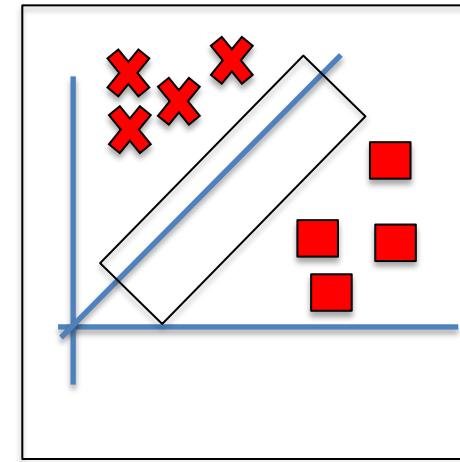
1D



2D

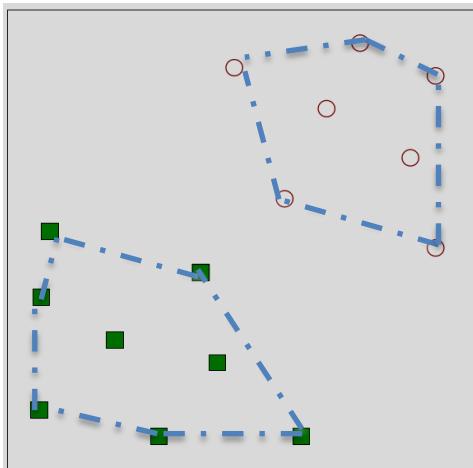


3D



# Definiciones

- Conjuntos linealmente separables: si existe una función lineal que los separa.
- Envolvente convexa: Dado un conjunto de puntos, su envolvente convexa es el conjunto convexo minimal que contiene al conjunto. La intersección de las envolventes de dos clases linealmente separables es el conjunto vacío.



# Definiciones

Dependiendo de la naturaleza del problema habrá separadores lineales o no lineales.

- Lineales: los hiperplanos creados, forman dos subgrupos que clasifican los datos de entrada con respecto a dos etiquetas.

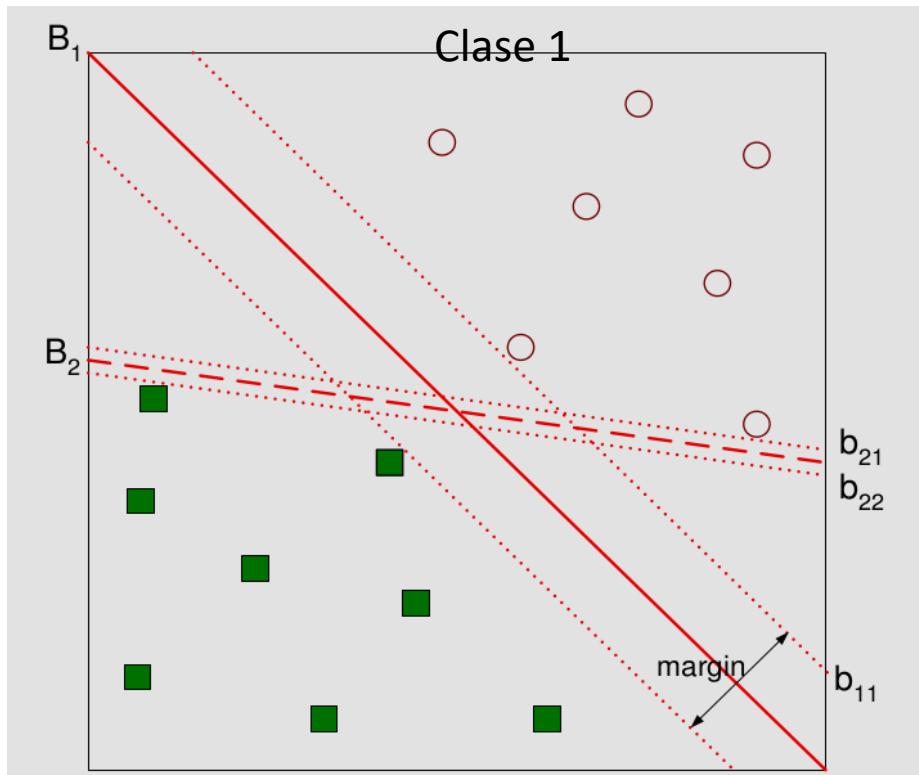
Si los datos no son separables ni teniendo en cuenta los datos de entrada, ni transformándolos en un nuevo espacio de características. Se aplicará el denominado “soft margin”.

- No lineales:

Los datos son separables pero tienen un número de características grande y necesitan un espacio multidimensional para su representación. Se aplicará el denominado “kernel”.

# Definiciones

En un espacio determinado puede haber muchos hiperplanos que separen bien las clases. El problema es que algunos de ellos no lo harán de forma óptima



# Definiciones

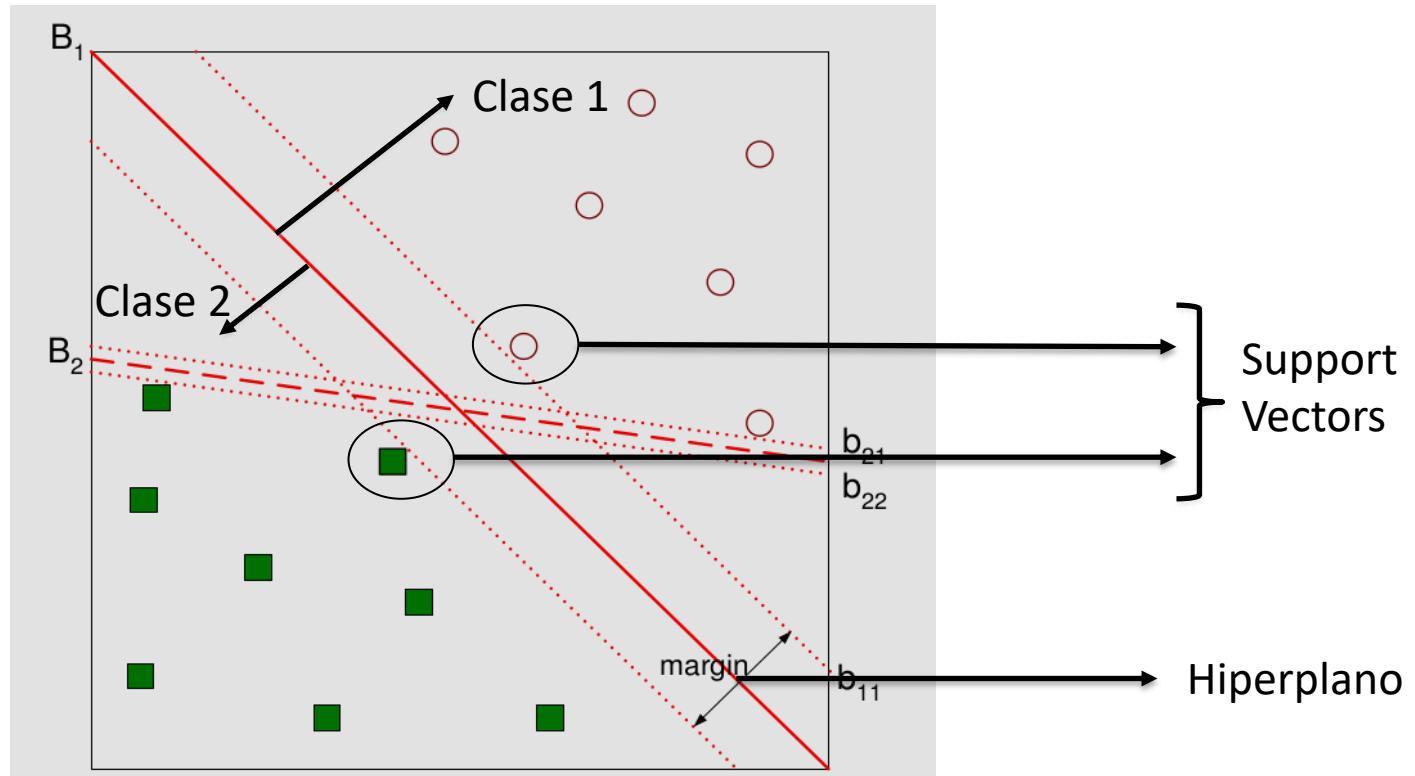
- Support Vectors: son los puntos que están cerca del hiperplano. Son aquellos que son más difíciles de clasificar. Tienen relación directa con la posición óptima del hiperplano.
- Margen la distancia entre los de puntos distintas clases que quedan más cerca del hiperplano. Hay que maximizarlo. La distancia mínima entre dos envolventes convexas.
- El hiperplano más óptimo será que el tenga el margen máximo. Perpendicular al segmento anterior y cortándolo por la mitad.
- El hiperplano separará individuos de distintas clases. Unos estarán a un lado del hiperplano y otros al otro.

# Definiciones

- Los Support Vectors son al menos uno por clase. Aunque posiblemente sean más.
- El conjunto de Support Vectors es el que permite definir de forma única el hiperplano que tiene el margen máximo. Por lo tanto el resto de los individuos no son relevantes.
- Si hubiese casos anómalos en el plano. Aumentar el margen acosta de incluir casos mal clasificados. “Soft-margin”.

# Definiciones

El mejor hiperplano es B1 porque tiene un margen mayor.



# Definiciones

- El hiperplano viene representado por  $w^*x+b$  siendo w perpendicular a este (índica el angulo) y b la posición que ocupa el plano en el eje Y.

Para un espacio de 2 dimensiones (una recta):

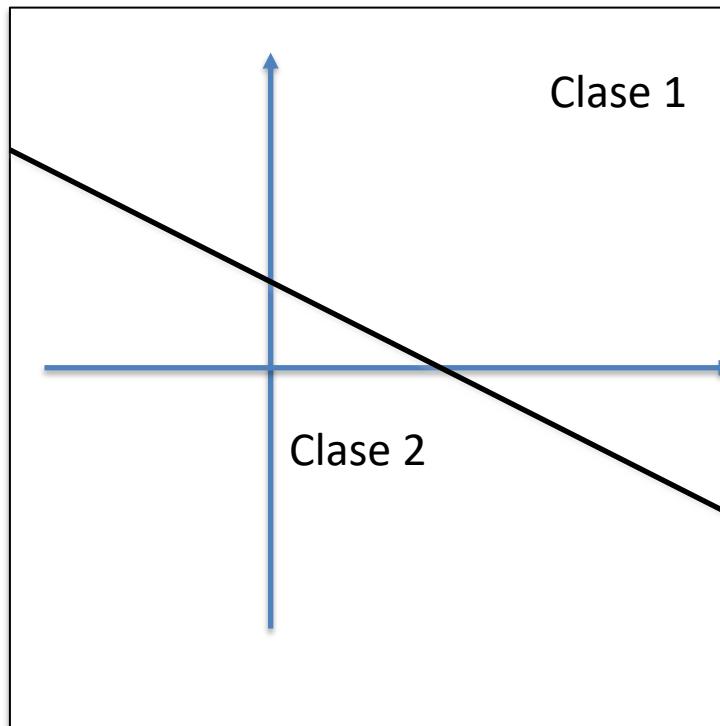
$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$

Los valores que cumplen esta condición pertenecen al hiperplano.

Los que no:  $>0$  o  $<0$  pertenecen a una clase u otra.

# Definición del problema

- **Ejemplo:** Hiperplano  $1 + 2X_1 + 3X_2 = 0$



Lo que quede a la izquierda del hiperplano es una clase  
y lo que quede a la derecha es otro.

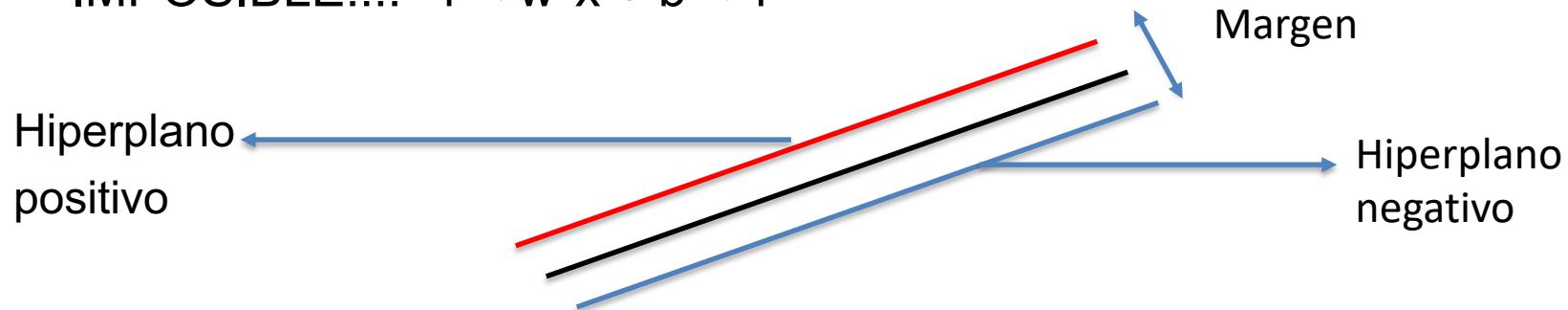
# Support Vector Machine

## Ejemplo binario.

El objetivo es maximizar el valor del margen M con respecto a w y b.

Se cumple que para cada clase:

- Para la clase -1.  $w^*x + b \leq -1$  (Hiperplano negativo)
- Para la clase 1.  $w^*x + b \geq 1$  (Hiperplano positivo)
- IMPOSIBLE!!!!  $-1 < w^*x + b < 1$



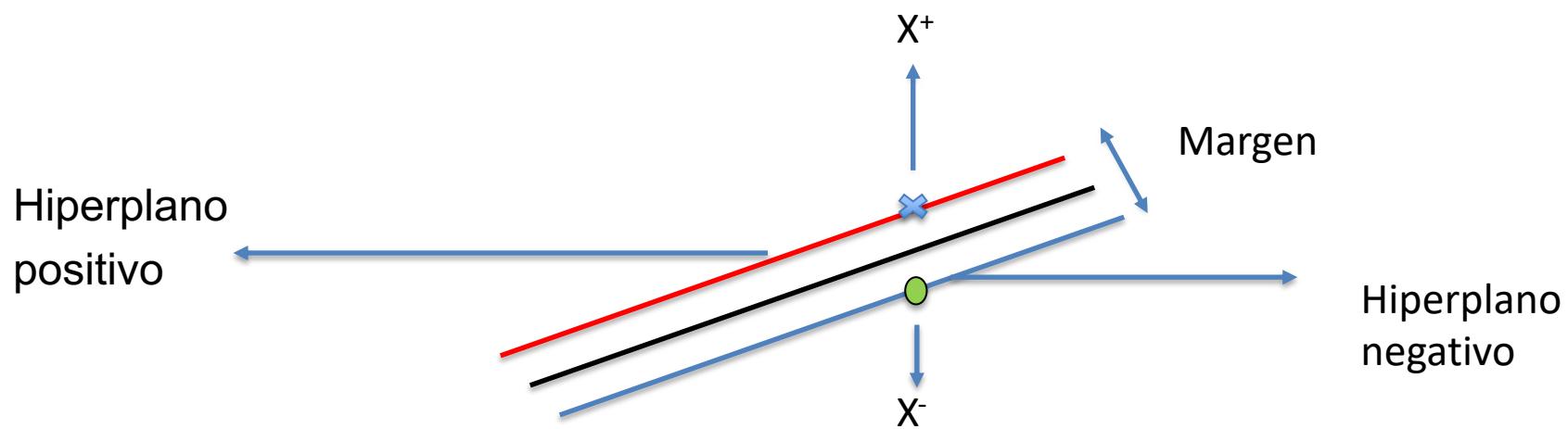
# Support Vector Machine

$X^+$  es un punto en el hiperplano positivo (el de la clase +1)

$X^-$  es el punto más cercano a  $X^+$  en el hiperplano negativo (el de la clase -1)

$$w^*x^+ + b = 1$$

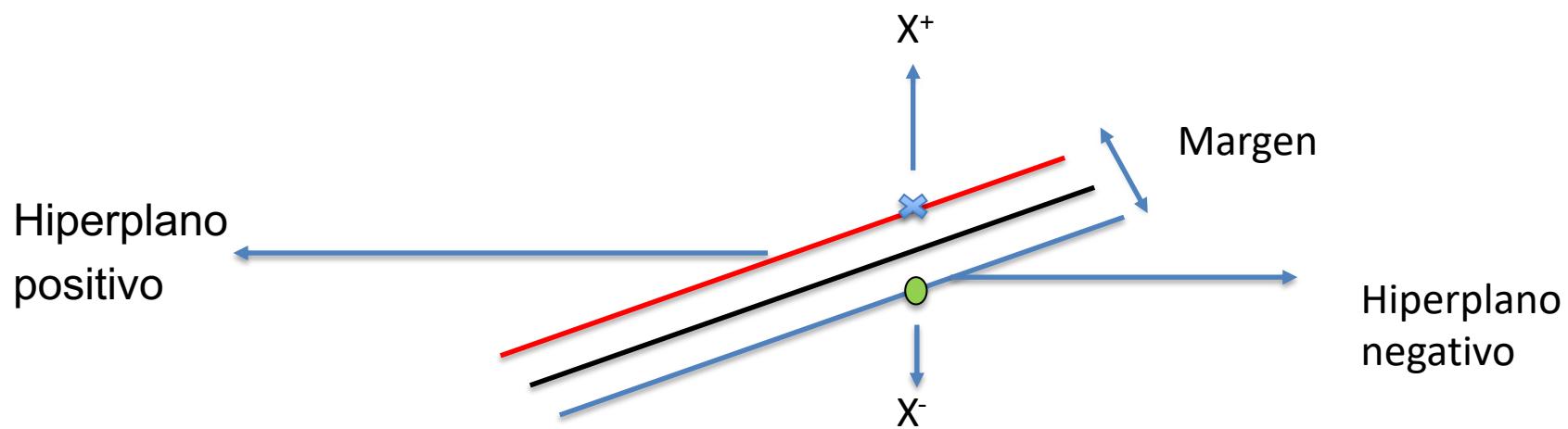
$$w^*x^- + b = -1$$



# Support Vector Machine

Entonces  $X^+ = X^- + \lambda^* w$  siendo  $\lambda$  un valor determinado. Es decir la distancia que hay que recorrer en la dirección de  $w$ .

Por lo tanto  $|X^+ - X^-| = M$



# Support Vector Machine

Teniendo:

$$1. \underline{w^*x^+ + b = 1}$$

$$2. \underline{w^*x^- + b = -1}$$

$$3. \underline{x^+ = x^- + \lambda^*w}$$

$$4. |x^+ - x^-| = M$$

Con 1, 2 y 3:

$$w^*(x^- + \lambda^*w) + b = 1$$

$$w^*x^- + b + \lambda^*w^2 = 1$$

$$w^*x^- + (-1 - w^*x^-) + \lambda^*w^2 = 1$$

$$\lambda = 2/(w^2)$$



# Support Vector Machine

Teniendo:

$$1. \mathbf{w}^* \mathbf{x}^+ + b = 1$$

$$2. \mathbf{w}^* \mathbf{x}^- + b = -1$$

$$3. \mathbf{x}^+ = \mathbf{x}^- + \lambda^* \mathbf{w}$$

$$4. |\mathbf{x}^+ - \mathbf{x}^-| = M$$

$$5. \lambda = 2/(\mathbf{w}^2)$$

Con 3, 4 y 5:

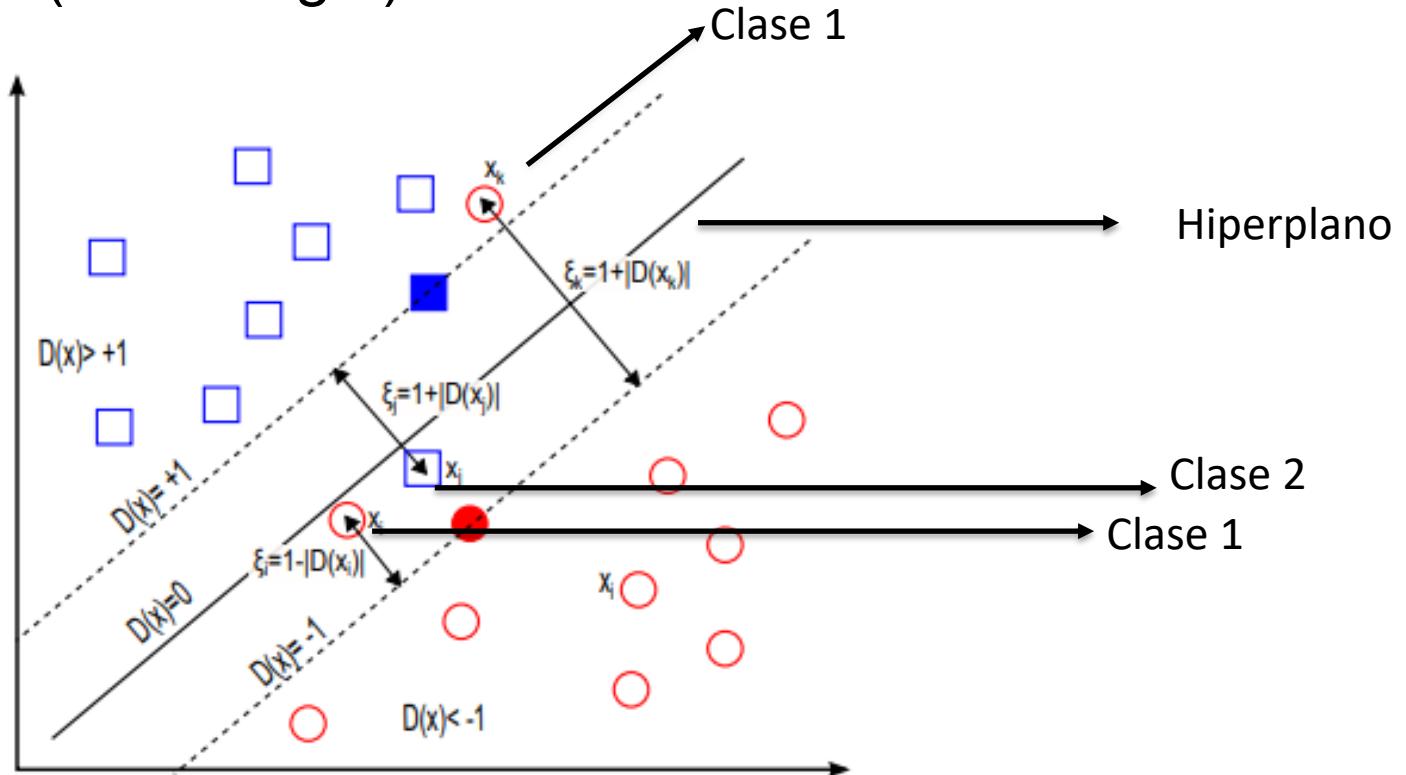
$$M = |\mathbf{x}^+ - \mathbf{x}^-| = |\lambda^* \mathbf{w}| = \lambda^* |\mathbf{w}| = 2 * |\mathbf{w}| / (\mathbf{w}^2) == 2 * |\mathbf{w}| / |\mathbf{w}^2| = 2 / |\mathbf{w}|$$

Maximizar

$$M = \frac{2}{|\mathbf{w}|}$$

# Support Vector Machine

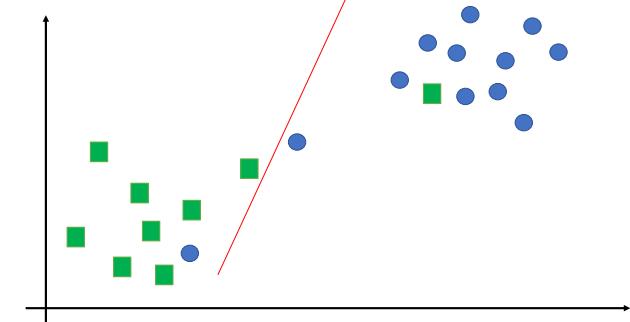
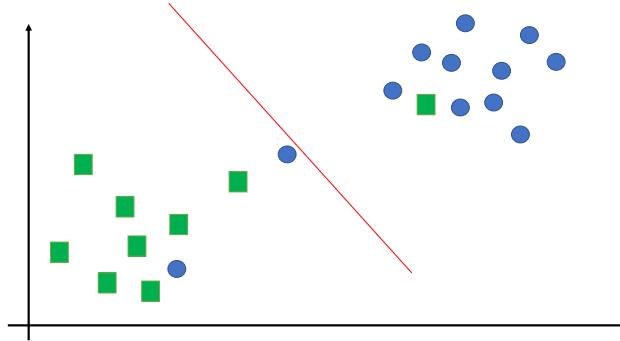
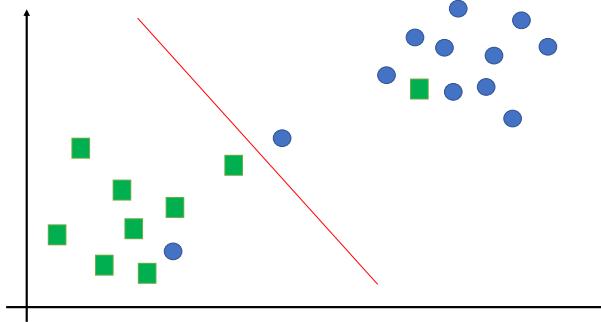
Caso no separable (soft margin):



Dejar en el hiperplano instancias mal clasificadas.

# Support Vector Machine

Ejercicio (soft margin):  
Elige el mejor hiperplano.



# Support Vector Machine

- Una instancia  $(x_i, y_i)$ , su variable de holgura,  $\xi_i$ , representa la desviación del caso separable, medida desde el borde del margen que corresponde a la clase  $y_i$ .
  - Variables de holgura de valor cero corresponden a ejemplos separables.
  - Mayores que cero corresponden a ejemplos no separables.
  - Mayores que uno corresponden a ejemplos no separables y, además, mal clasificados.
- La suma de todas las variables de holgura, permite, de alguna manera, medir el coste asociado al número de ejemplos no-separables

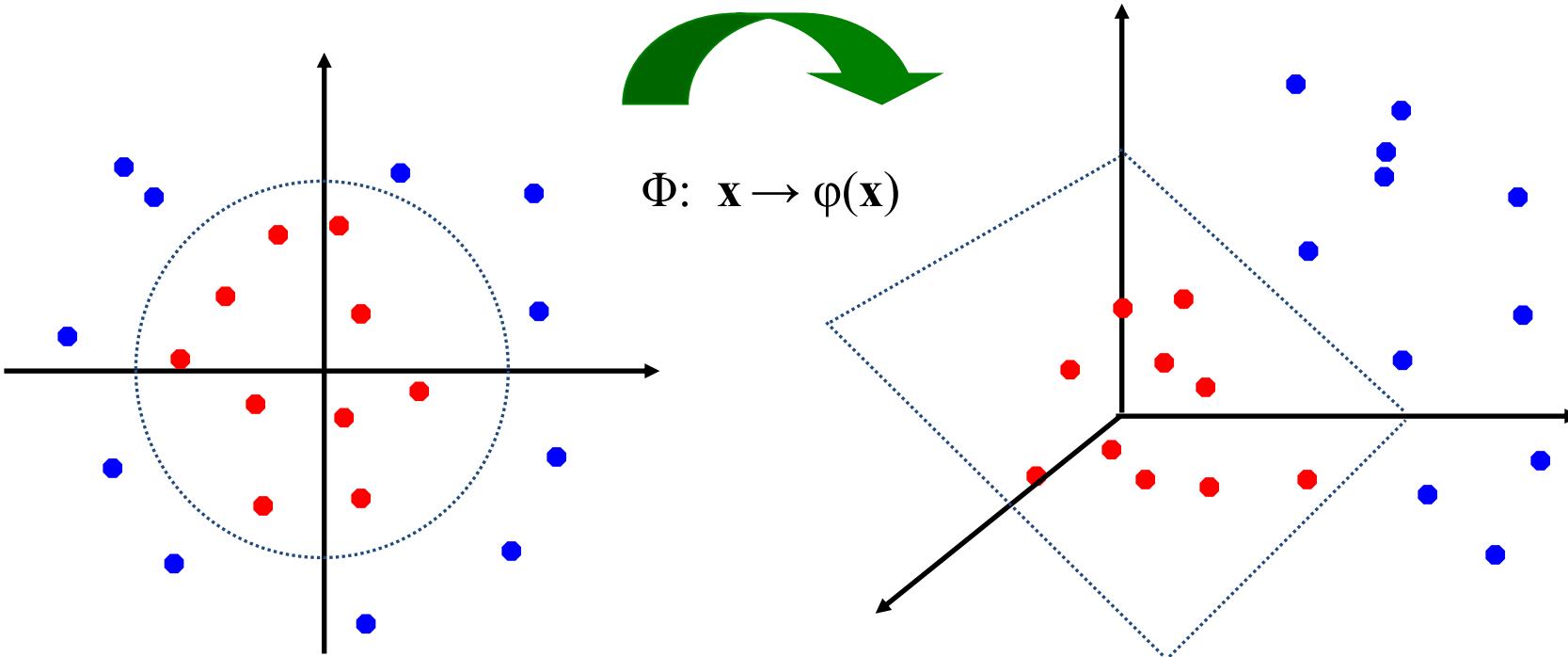
# Support Vector Machine

Minimizar  $f(w, \xi) = 1/(2^*|w|^2) + C^* \sum_{i=1}^n \xi_i$

C es una constante, suficientemente grande, elegida por el usuario, que permite controlar en qué grado influye el término del coste de ejemplos no-separables.

# Support Vector Machine

Caso no separable linealmente (kernel):



Mapear el set de datos transformándolo a un espacio de más dimensiones donde sea separable.

# Support Vector Machine

Lineal:  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$

Polinómico  $p$ :  $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$

Gaussiana (radial-basis function network o RBF):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

Sigmoid:  $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$