



# RECONOCIMIENTO DE PATRONES EN IMÁGENES TICS 585

FACULTAD DE INGENIERÍA Y CIENCIAS  
UNIVERSIDAD ADOLFO IBÁÑEZ

SEGUNDO SEMESTRE 2021

PROFESOR: MIGUEL CARRASCO

EVALUACIÓN DE MODELOS

## ■ Técnicas de selección

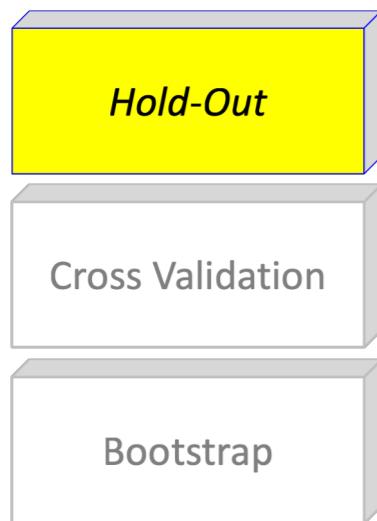
- Hold-Out Estimate
- Validación Cruzada (Cross-Validation)
- Bootstrap

*OBJETIVO:*

Seleccionar un conjunto de datos  
para estimar medidas estadísticas  
de rendimiento o error



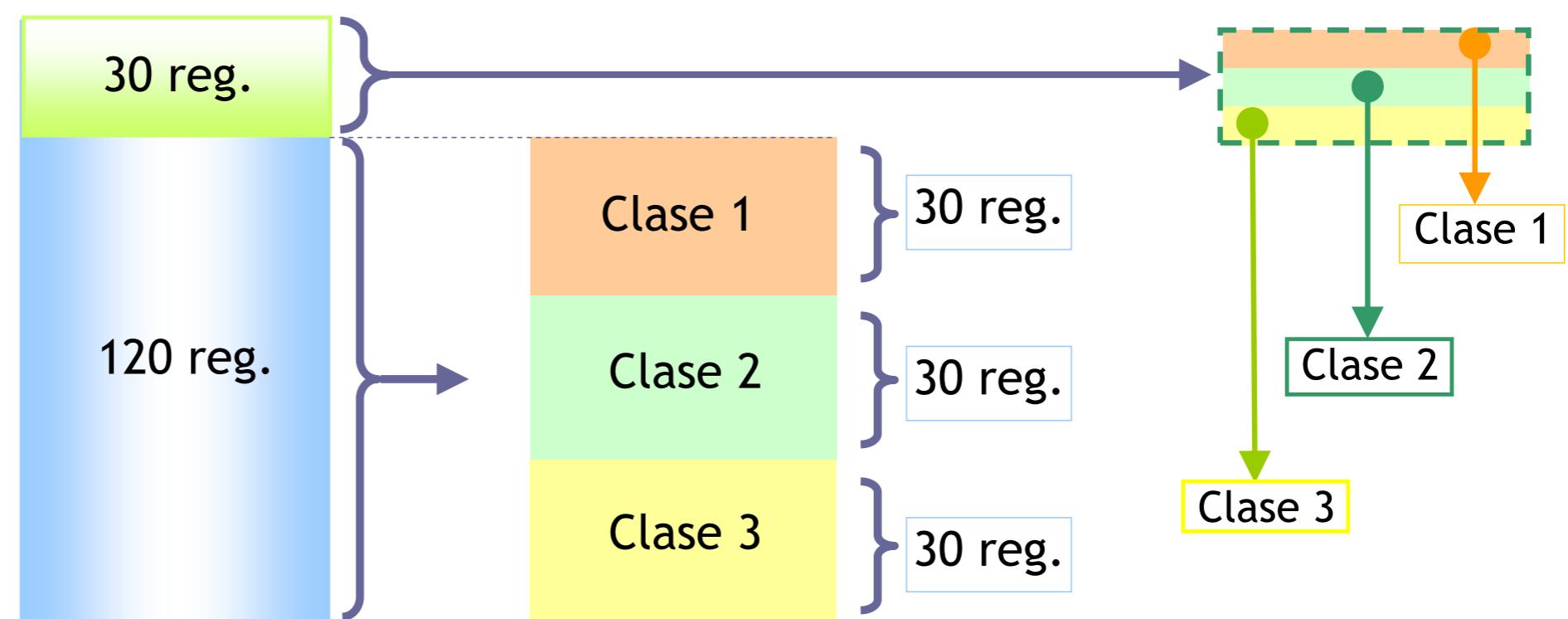
## ■ Técnicas de Selección



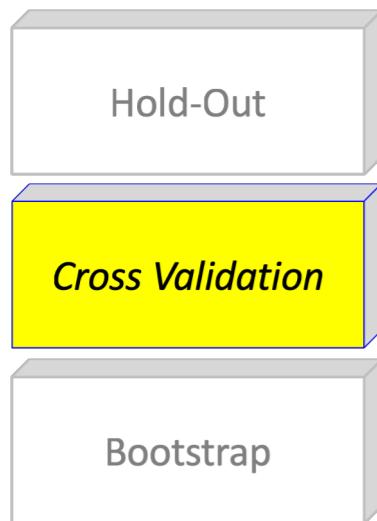
### ■ Hold-Out estimate

Es una técnica de selección de datos para entrenar y evaluar nuestro modelo. Usualmente utiliza un porcentaje fijo de los datos para entrenar y otro para modelar.

- Recordemos que cualquiera sea la técnica a emplear para clasificar, nuestro objetivo es construir un clasificador genérico, de tal forma que sirva para registros no conocidos

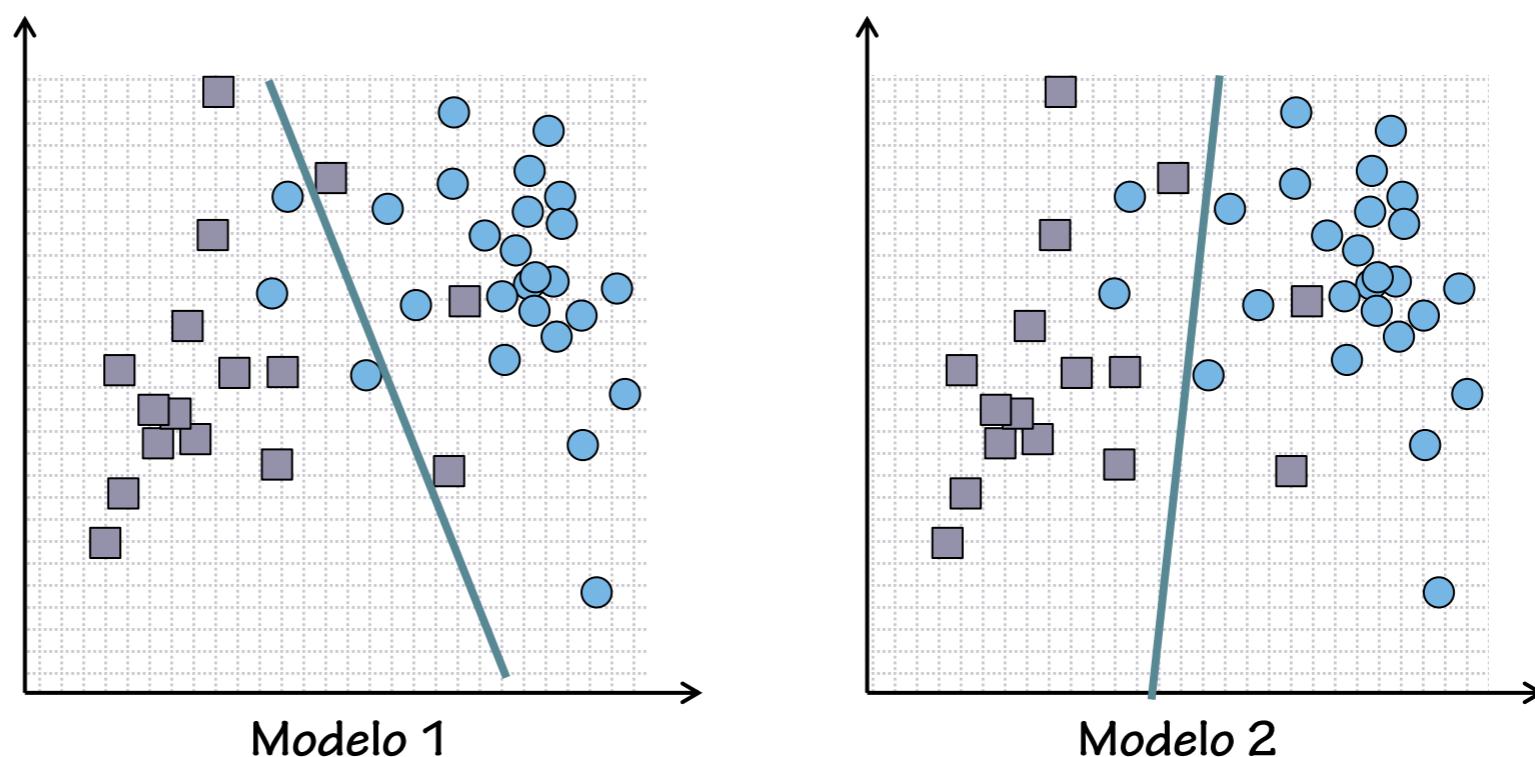


## Técnicas de Selección

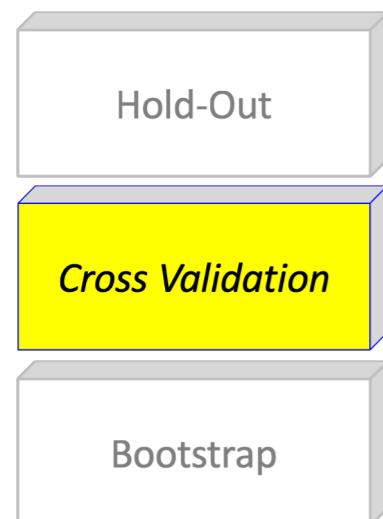


### ■ Validación cruzada (Cross-Validation)

- ¿Qué modelo es mejor?
- ¿Si en ambos casos empleamos el 30% de los datos de testing en forma aleatoria, porqué entonces obtenemos resultados distintos?
- ¿Cuál es el rendimiento final de nuestro clasificador?
- ¿Cuál modelo predice mejor frente a datos no conocidos?

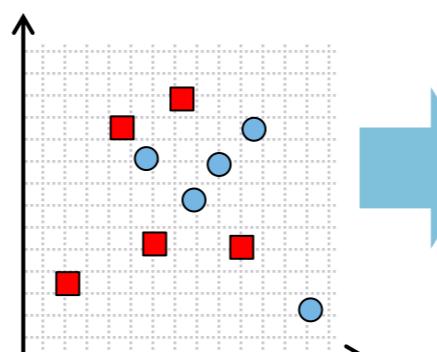


## Técnicas de Selección



### LOOCV (Leave-One-Out Cross Validation)

Este técnica consiste en entrenar con  $n-1$  datos. De esta forma siempre dejamos afuera a un dato y evaluamos con el dato sobrante.



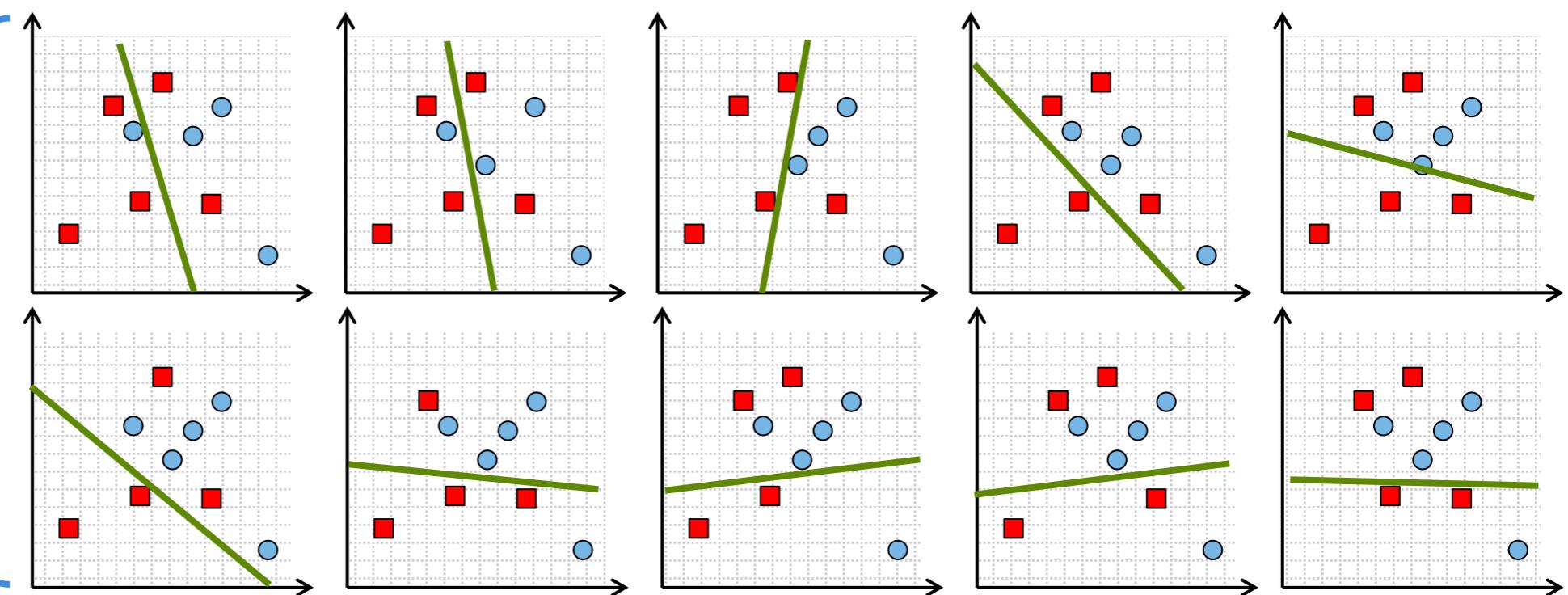
Datos originales:

- 5 puntos azules
- 5 puntos rojos

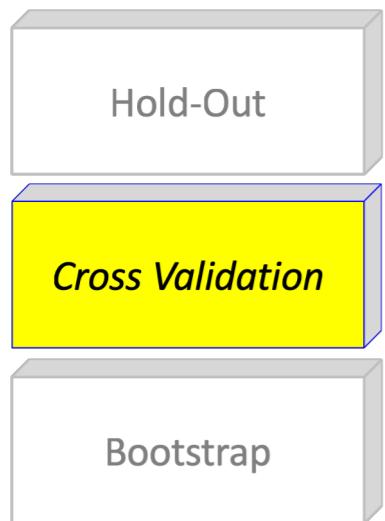
Entrenamiento  
con 9 puntos



Esta técnica es muy  
demandante computa-  
cionalmente

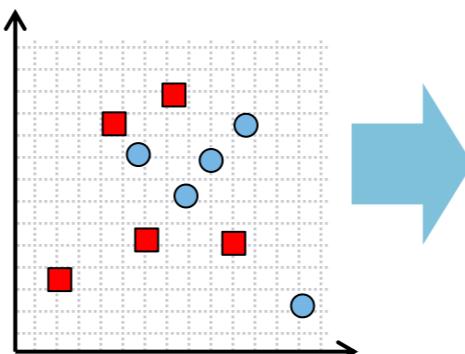


## Técnicas de Selección



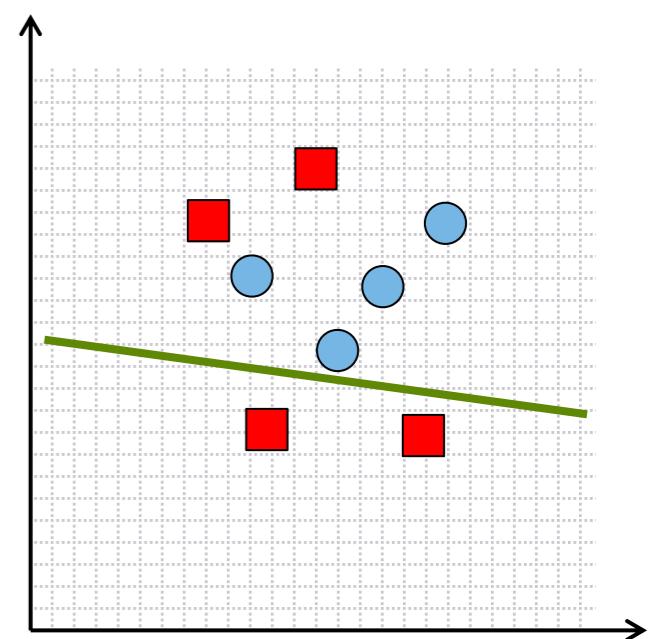
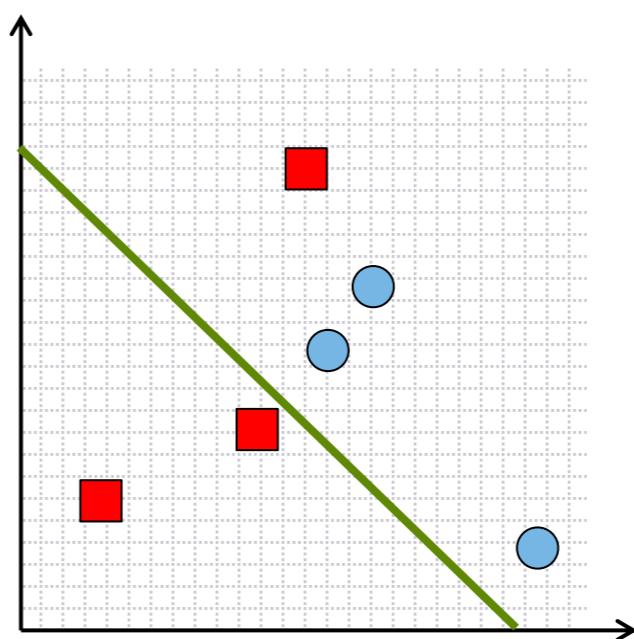
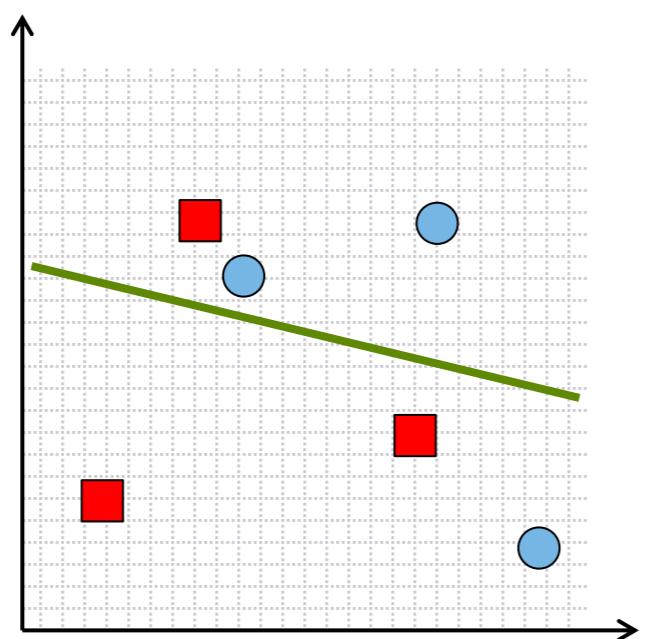
### ■ *k-CV (Cross Validation)*

Este técnica consiste dividir el conjunto de datos en  $k$ -grupos disjuntos. Cada conjunto es seleccionado al azar. Si existen varias clases, estas deben ser divididas

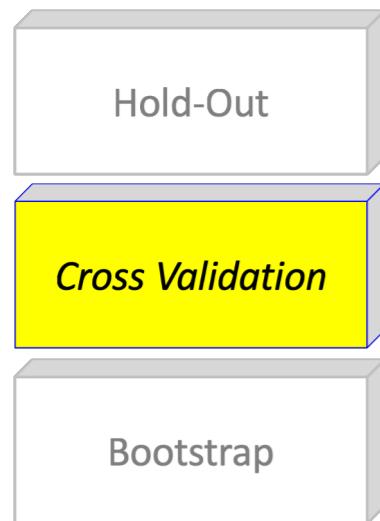


Datos originales:

- 5 puntos azules
- 5 puntos rojos

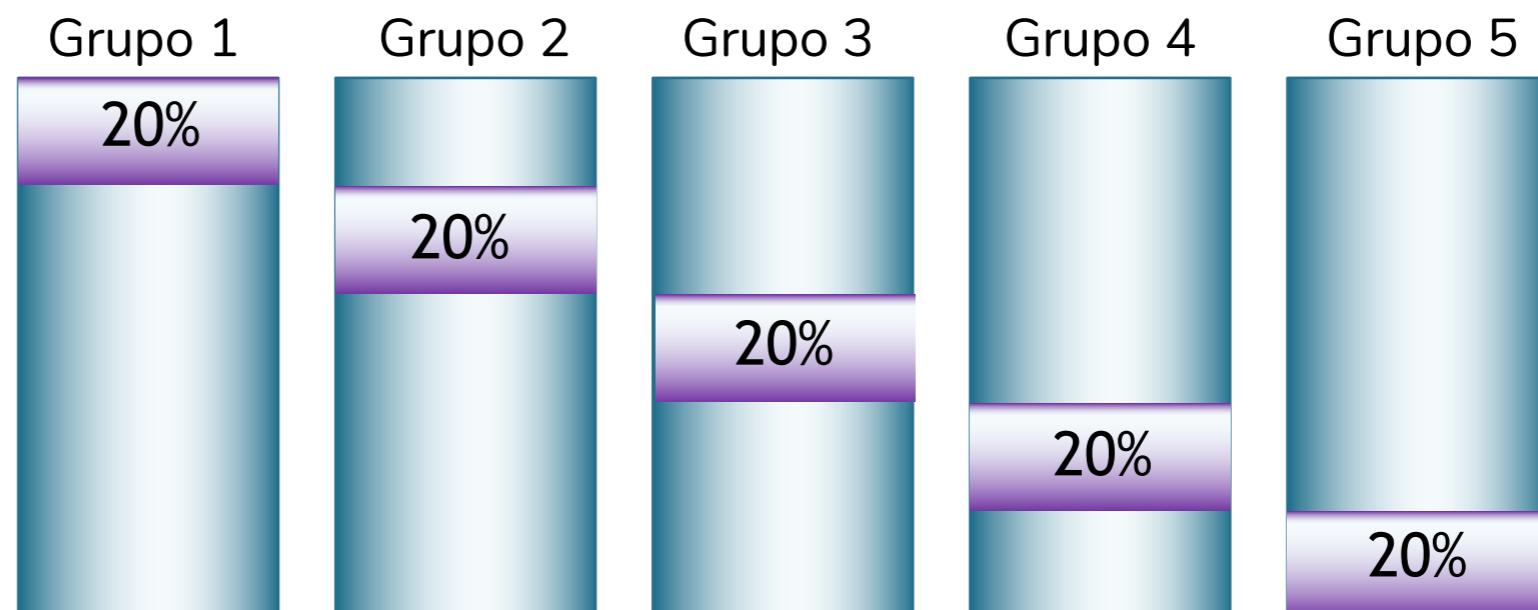


## ■ Técnicas de Selección



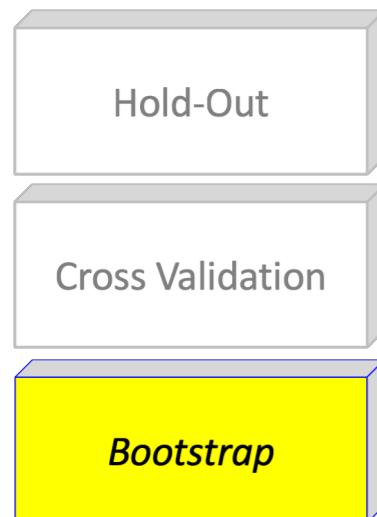
### ■ *k-CV (Cross Validation)*

Este técnica consiste dividir el conjunto de datos en  $k$ -grupos disjuntos. Cada conjunto es seleccionado al azar. Si existen varias clases, estas deben ser divididas



Podemos ver el problema desde otro ángulo. En este caso seleccionamos  $k=5$  y por lo tanto tenemos cinco rendimientos. Promedie todos los resultados intermedios para obtener el rendimiento del modelo.

## ■ Técnicas de Selección



### ■ *Bootstrap*

Este técnica consiste seleccionar aleatoriamente un subgrupo de datos. Es muy empleada cuando tenemos muy pocos datos. De esta forma podemos testear nuevas combinaciones empleando subgrupos elegidos al azar, no importando si existe repetición

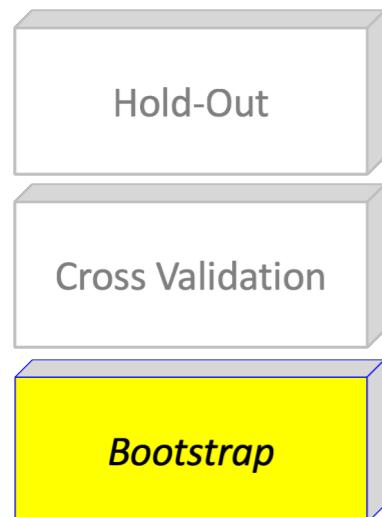
14	3	16	1	19
11	6	8	4	9

Supongamos que solo tenemos 10 muestras y deseamos construir un clasificador. **Con tan pocos datos es impracticable**, por ello seleccionamos datos al azar y los grupos que necesitemos

A continuación seleccionaremos cinco datos aleatorios para testear el modelo. Podemos generar el número de combinaciones que requiramos

4	1	14	19	3
6	14	9	1	16
:				
14	3	4	11	19

## Técnicas de Selección



### ■ *Bootstrap*

**Ejemplo de Bootstrap**

\*Obligatorio

¿Cuál es tu estatura? \*

Tu respuesta

¿Cuál es tu peso aproximado? \*

Tu respuesta

¿Qué nota final crees que obtendrás en el examen? \*

Tu respuesta

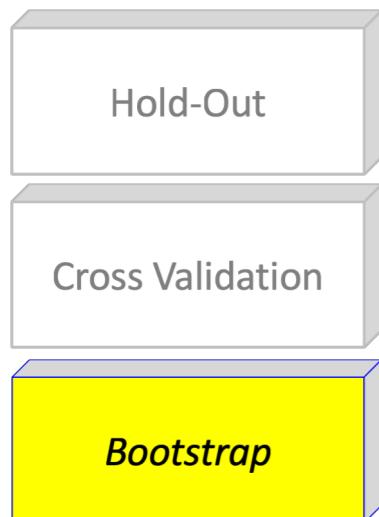
**ENVIAR**

Nunca envíes contraseñas a través de Formularios de Google.

Veamos como podemos aplicar esta técnica con datos reales en Python.

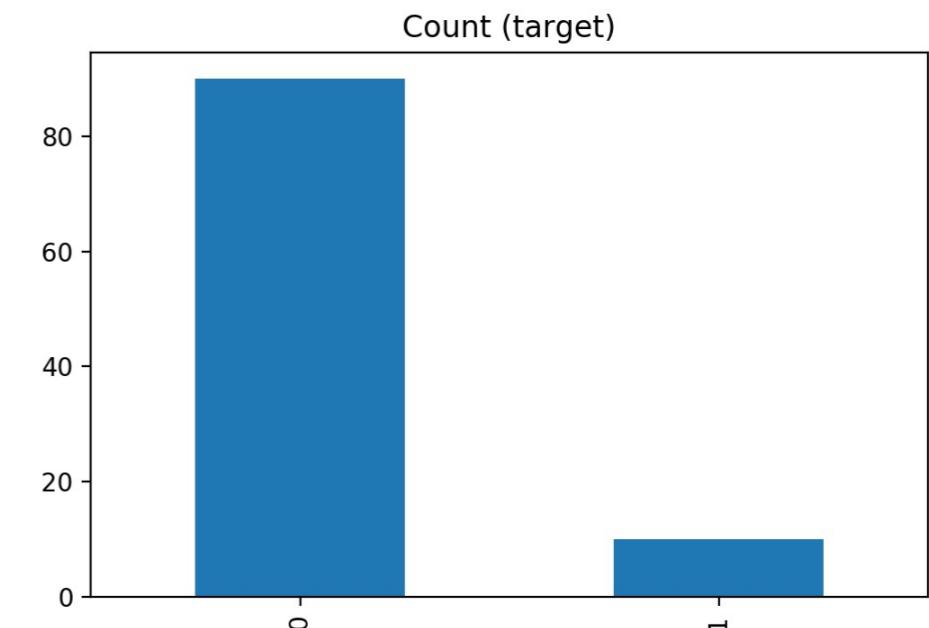
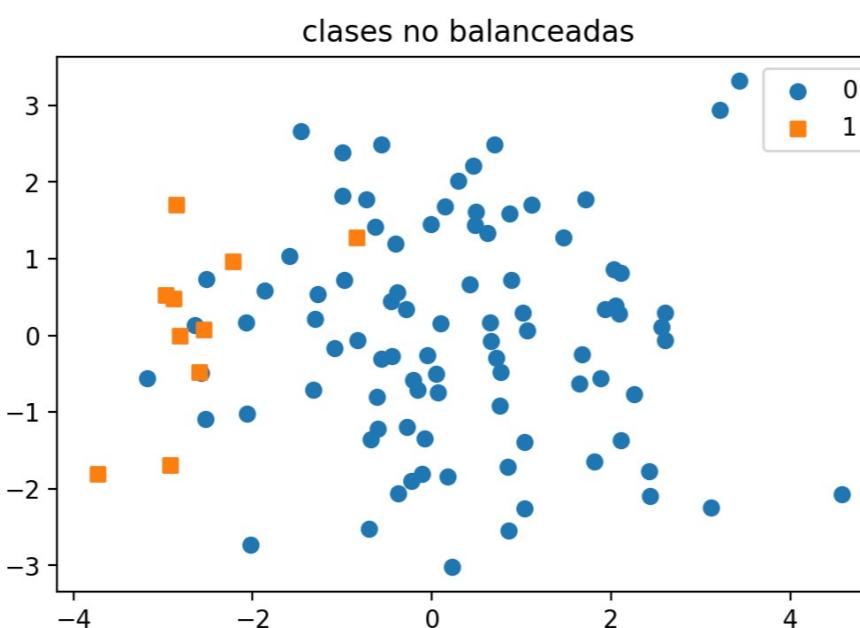


## Técnicas de Selección



### *Clases desbalanceadas*

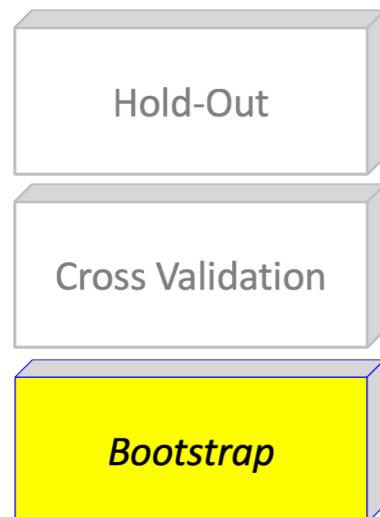
Cuando tenemos conjuntos de datos desbalanceados, existen técnicas que nos permiten remuestrear el espacio de valores. De esta forma, mejoramos el rendimiento del modelo.



En este ejemplo, vemos como el 90% de los datos corresponde a la clase azul



## ■ Técnicas de Selección



### ■ *Clases desbalanceadas*

Cuando tenemos conjuntos de datos desbalanceados, existen técnicas que nos permiten remuestrear el espacio de valores. De esta forma, mejoramos el rendimiento del modelo.

Por ejemplo en el grupo **donde hay más muestras de una clase, podemos realizar un sub sampleo (undersampling)**, y en cambio en las **clases donde hay menos datos, realizamos un sobre sampleo (oversampling)**

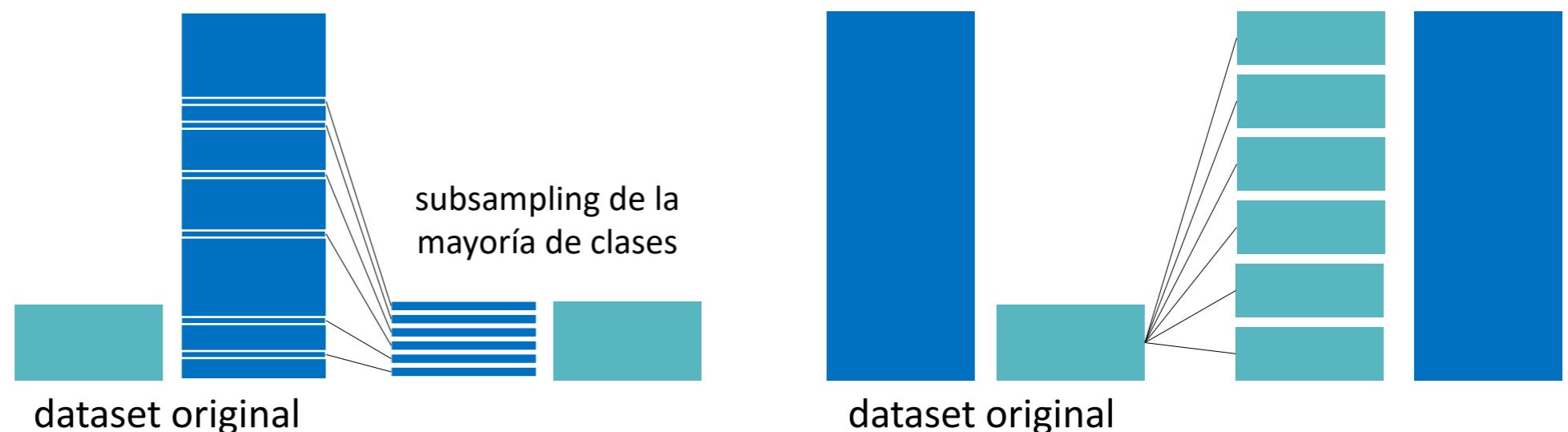
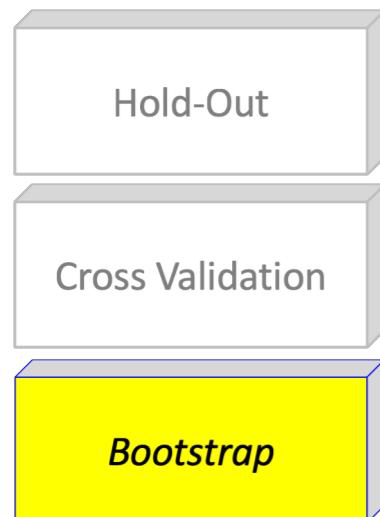


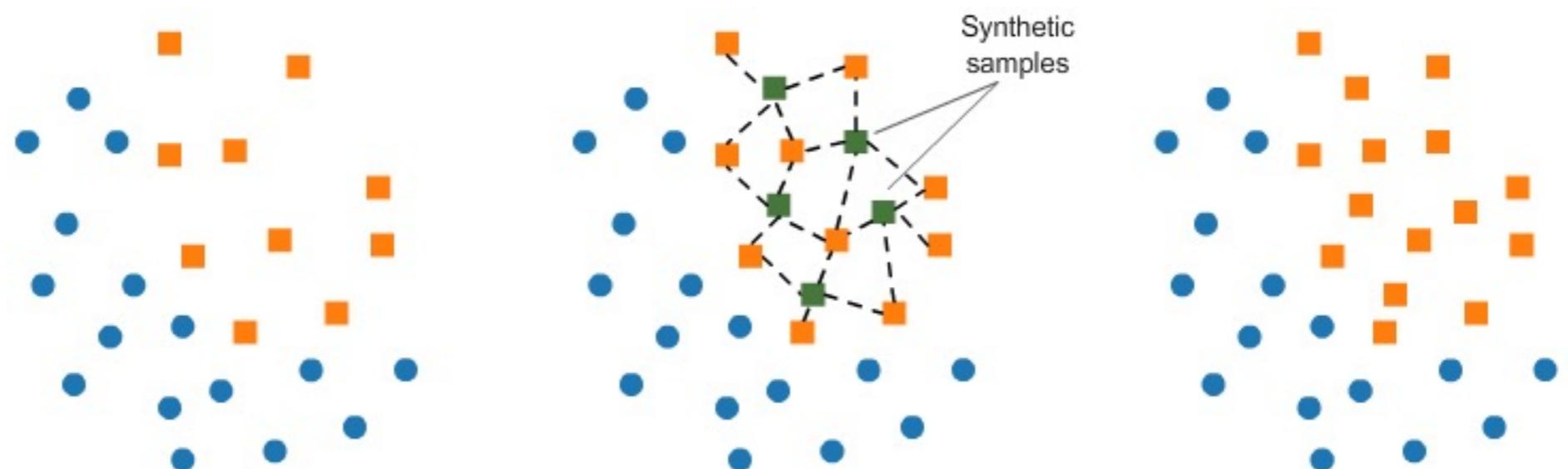
figura basada en: <https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>

## Técnicas de Selección



### ■ *SMOTE (Synthetic Minority Oversampling TTechnique)*

Este técnica consiste en crear nuevos puntos para las clases donde haya un menor número. Esto ocurre normalmente en caso de clases desbalanceadas



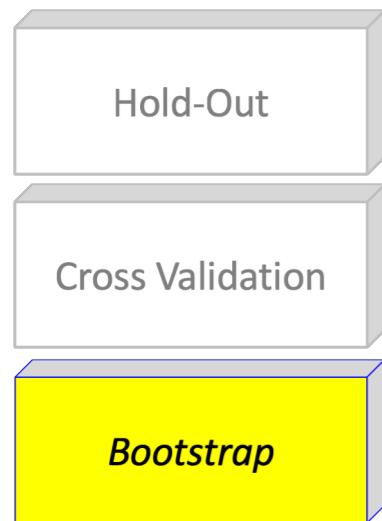
más info en:

<https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>

images from

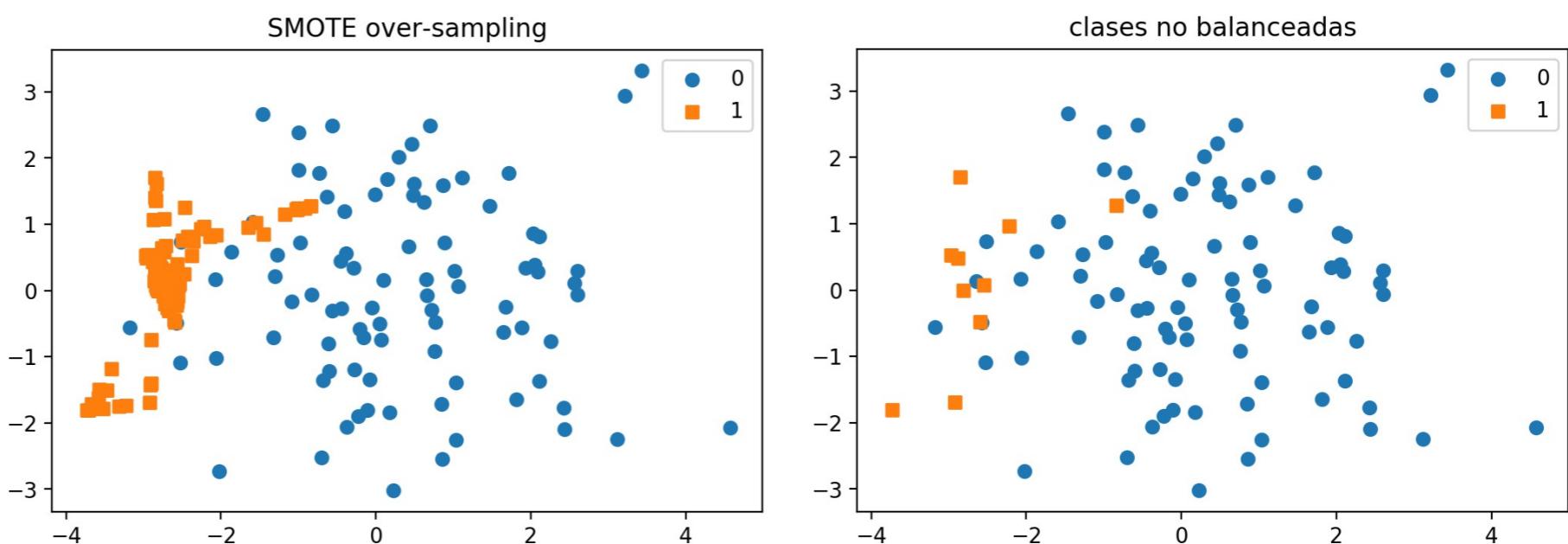
[https://raw.githubusercontent.com/rafjaa/machine\\_learning\\_fecib/master/src/static/img/smote.png](https://raw.githubusercontent.com/rafjaa/machine_learning_fecib/master/src/static/img/smote.png)

## Técnicas de Selección



### ■ *SMOTE (Synthetic Minority Oversampling TTechnique)*

Este técnica consiste en crear nuevos puntos para las clases donde haya un menor número. Esto ocurre normalmente en caso de clases desbalanceadas



más info en:

<https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>

images from

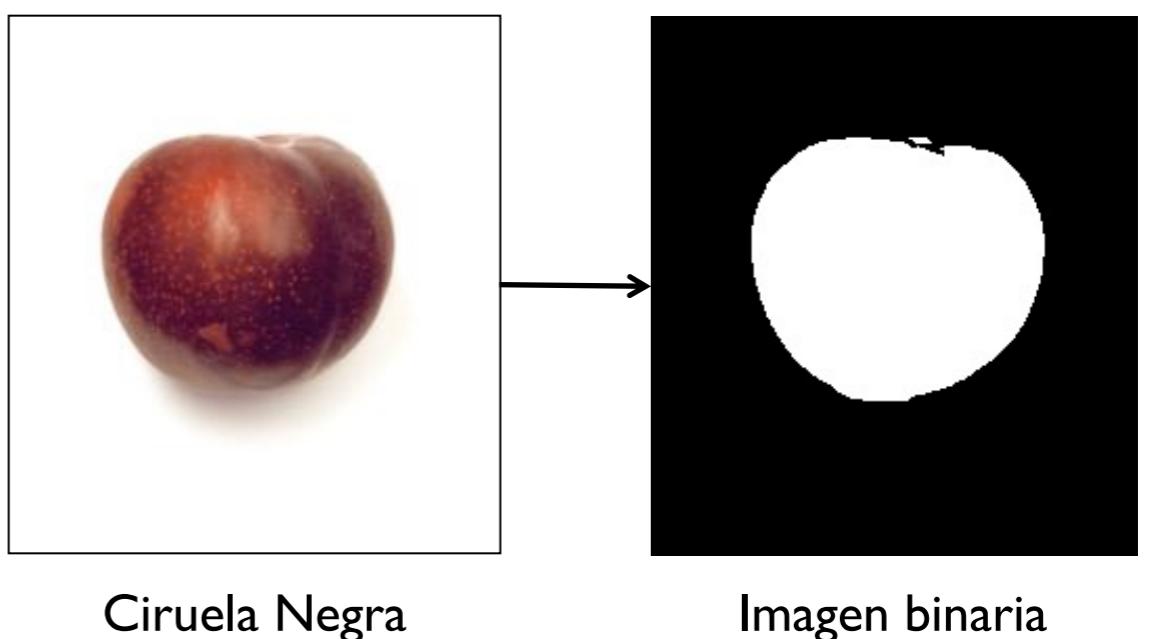
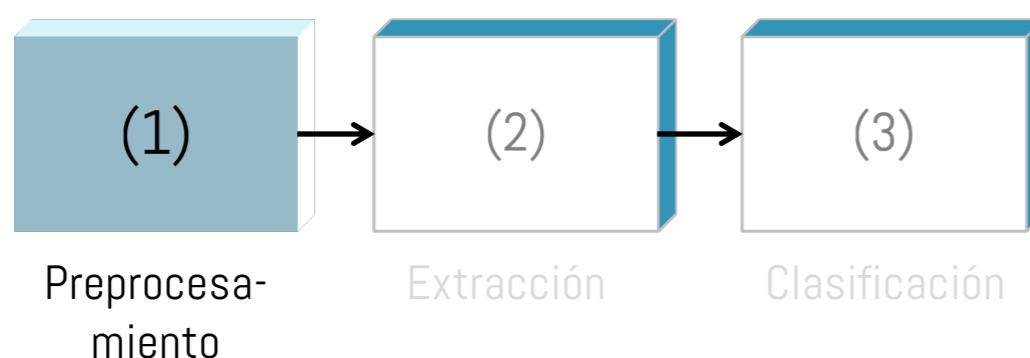
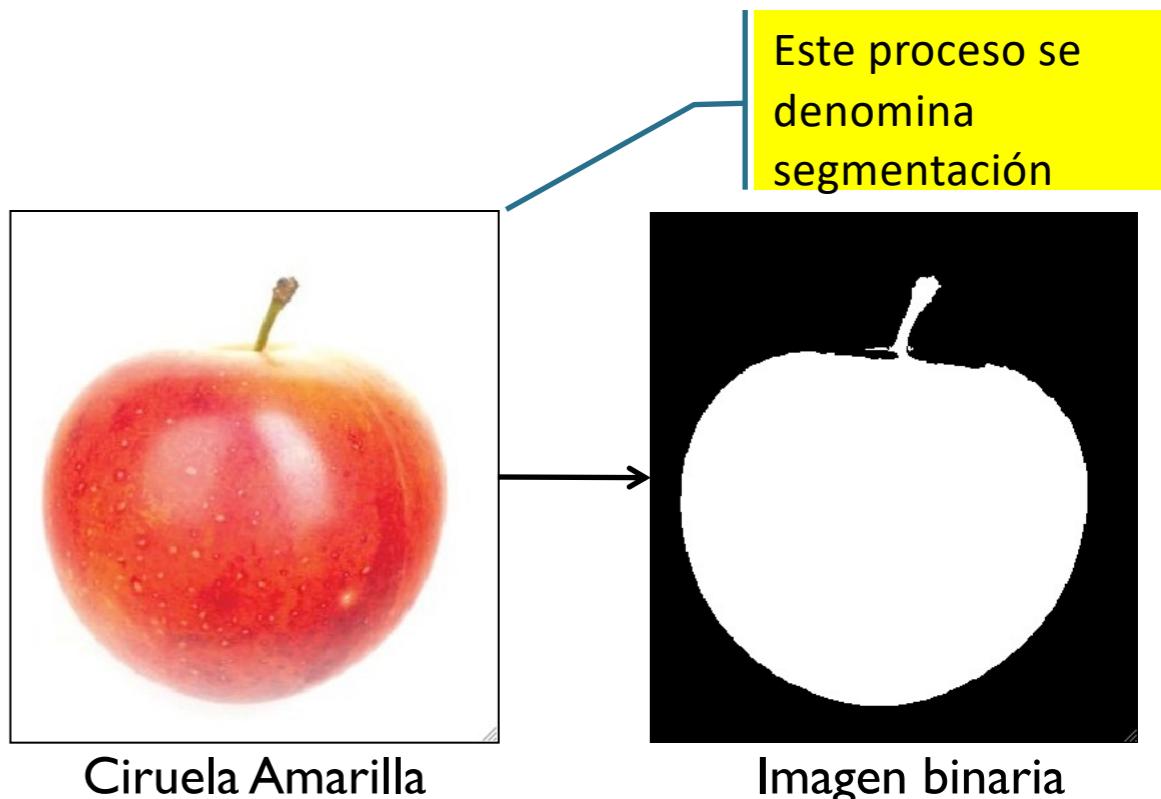
[https://raw.githubusercontent.com/rafjaa/machine\\_learning\\_fecib/master/src/static/img/smote.png](https://raw.githubusercontent.com/rafjaa/machine_learning_fecib/master/src/static/img/smote.png)

- Modelación predictiva
- Métricas de evaluación de rendimiento

## ■ Preprocesamiento

En esta etapa la información se encuentra en forma bruta, para ello debe ser extraída y adecuada de tal forma que se pueda procesar.

Ejemplo. Quitar ruido, limpiar la imagen, mejorar el contraste, completar datos incompletos, etc.

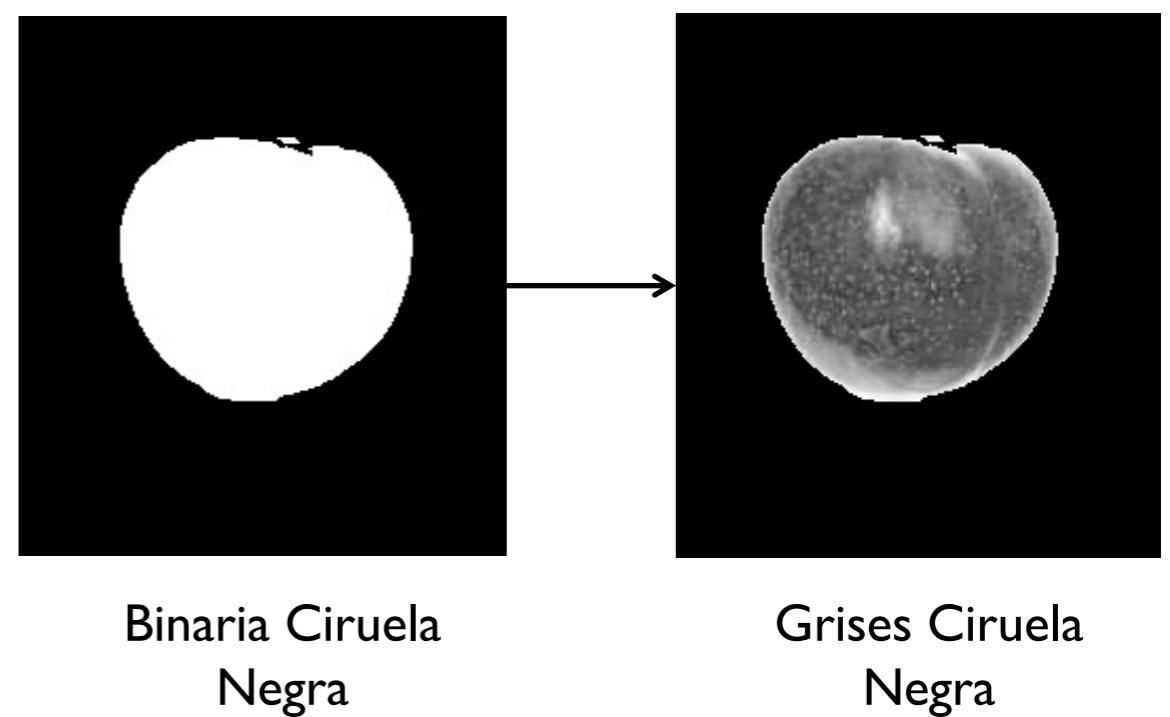
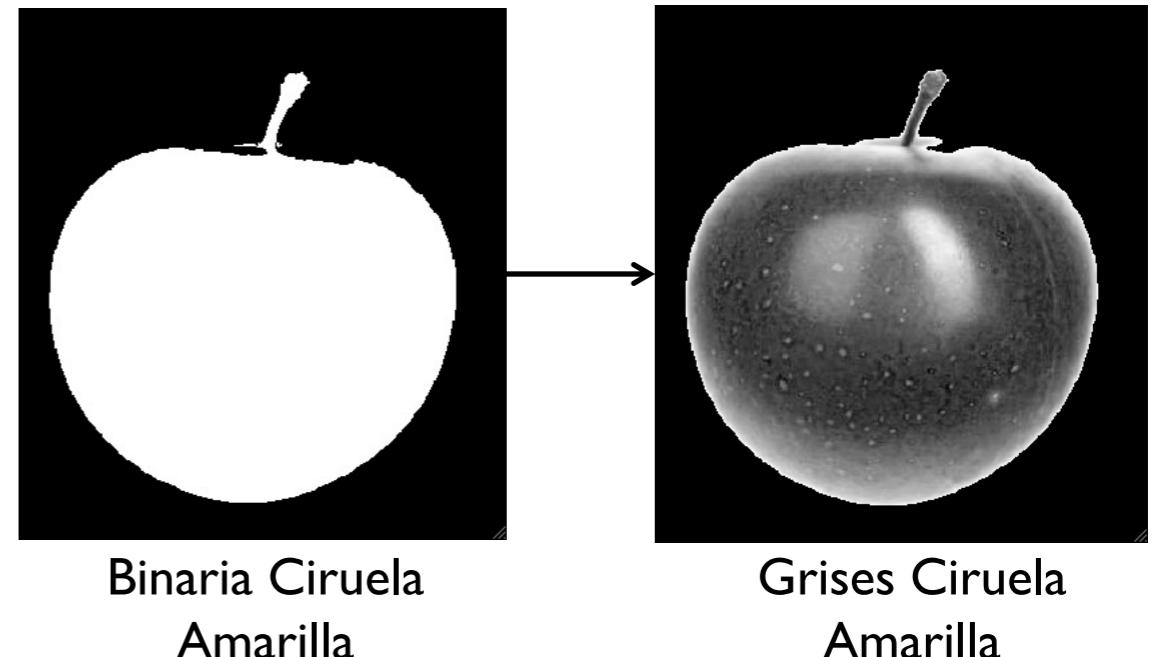
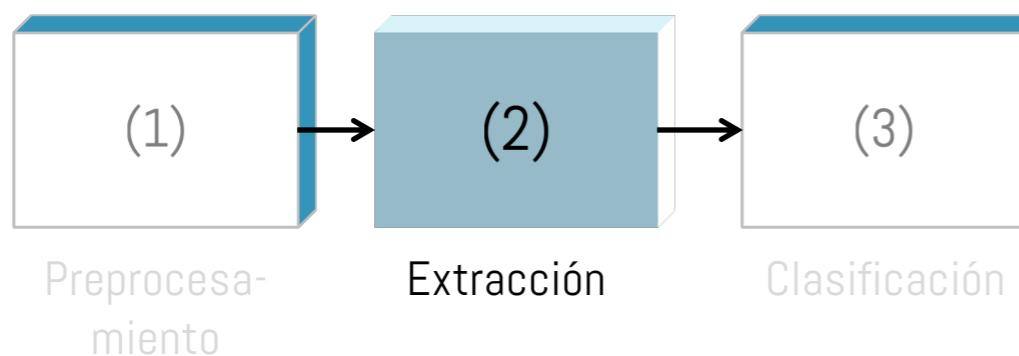


## ■ Extracción de características

En esta etapa se extraen y miden las características (*features*) de los objetos segmentados.

Podemos extraer cualquier característica, independiente si esta es útil o no. Más adelante estudiaremos algoritmos para definir cuál o cuales de estas características son útiles.

*Ejemplo:* tamaño, diámetro, peso, color, forma, momentos.

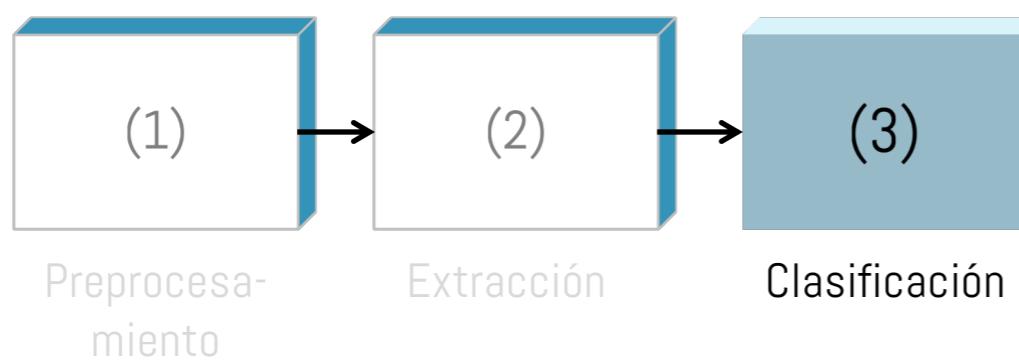


## Clasificación

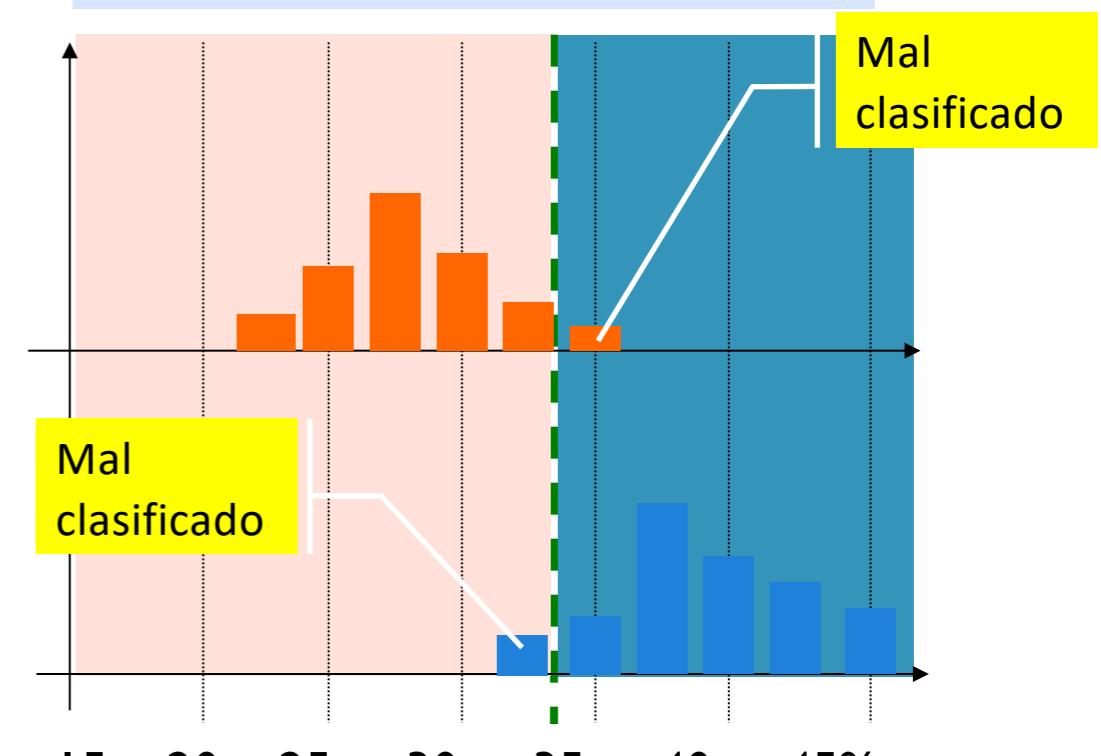
El objetivo del clasificador consiste en asignar clases a los objetos según sus características anteriormente extraídas.

Para diseñar el clasificador es necesario realizar un entrenamiento “off-line”. El entrenamiento permite que el sistema funcione para datos no conocidos por el sistema, de esta forma podemos emplear el clasificador para nuevos datos.

Sin embargo, no siempre obtendremos una buena separación entre las clases, y quedarán mal clasificados algunos objetos.



## Histograma

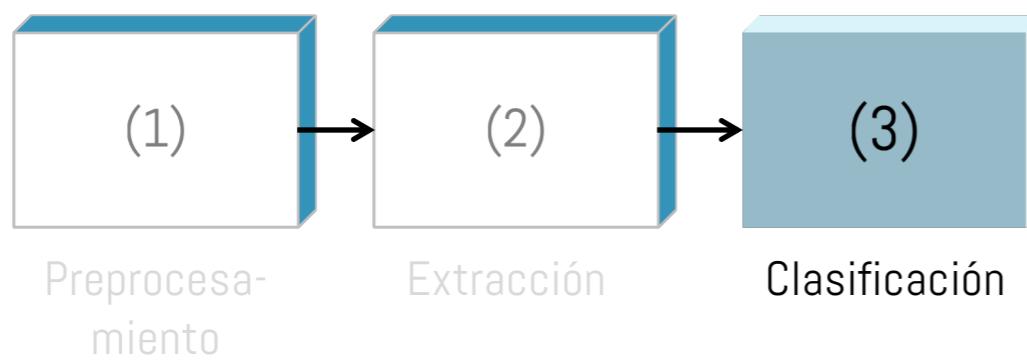
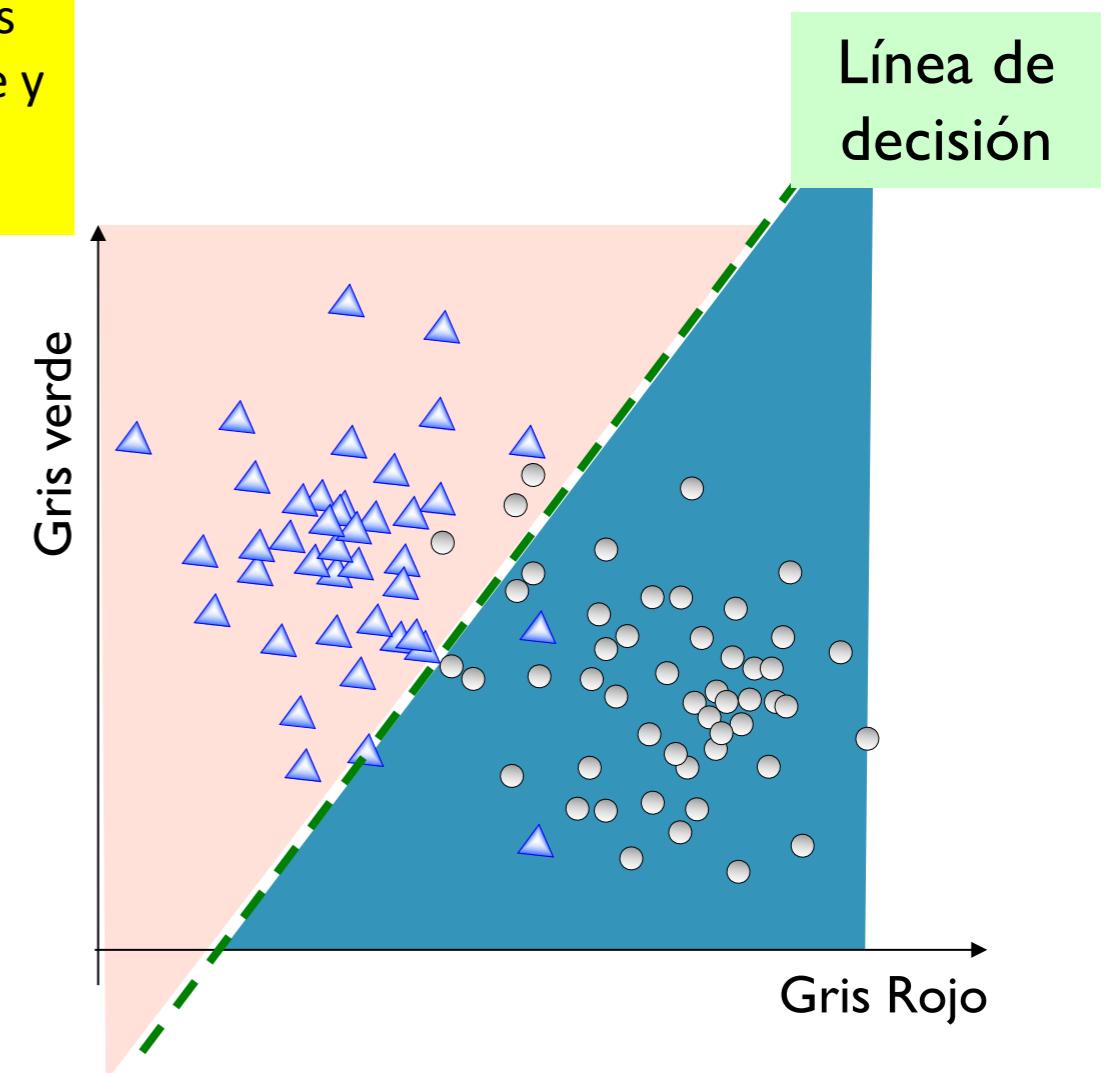
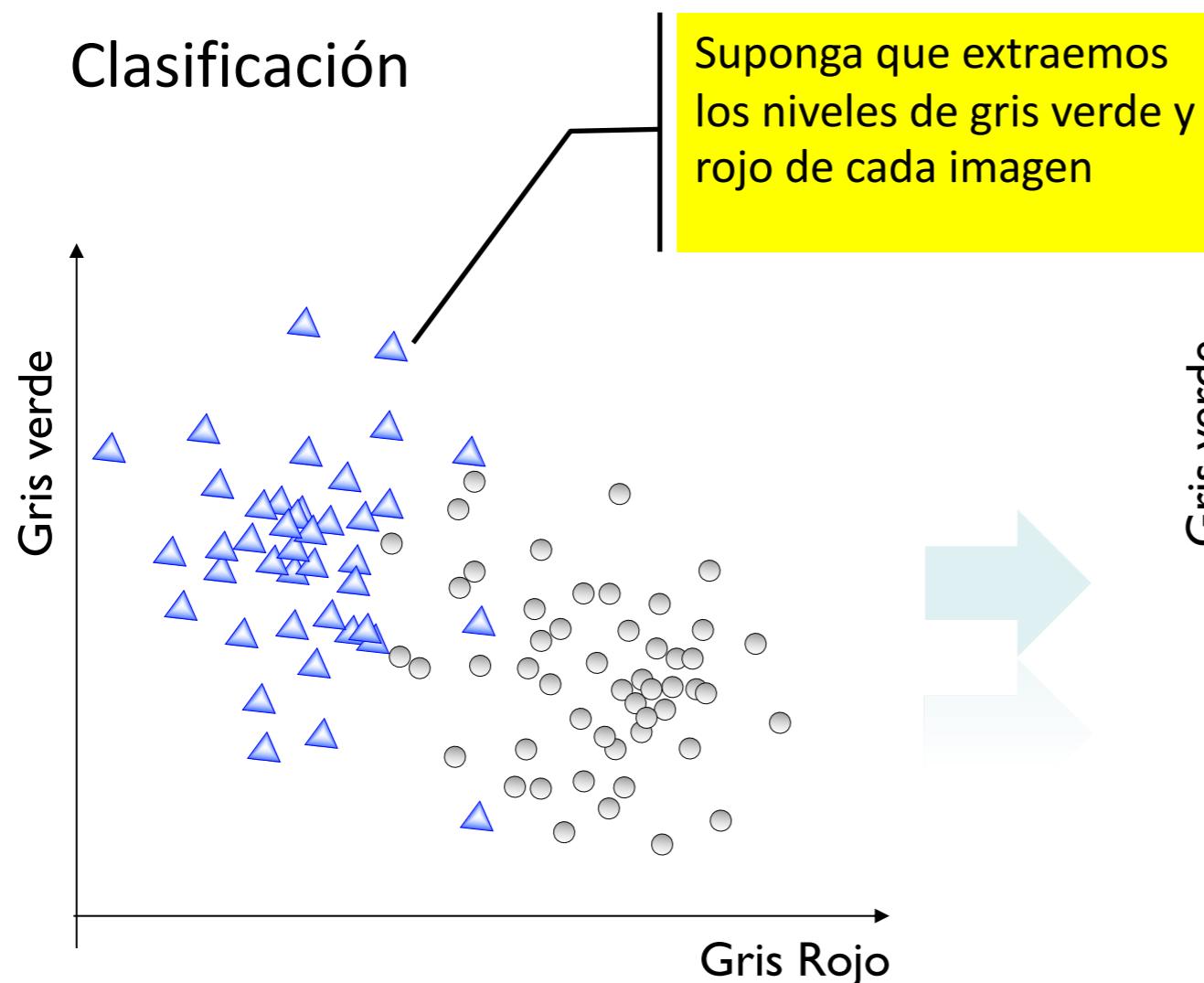


15 20 25 30 35 40 45% ← →

Ciruelas Negras ← → Ciruelas Amarillas

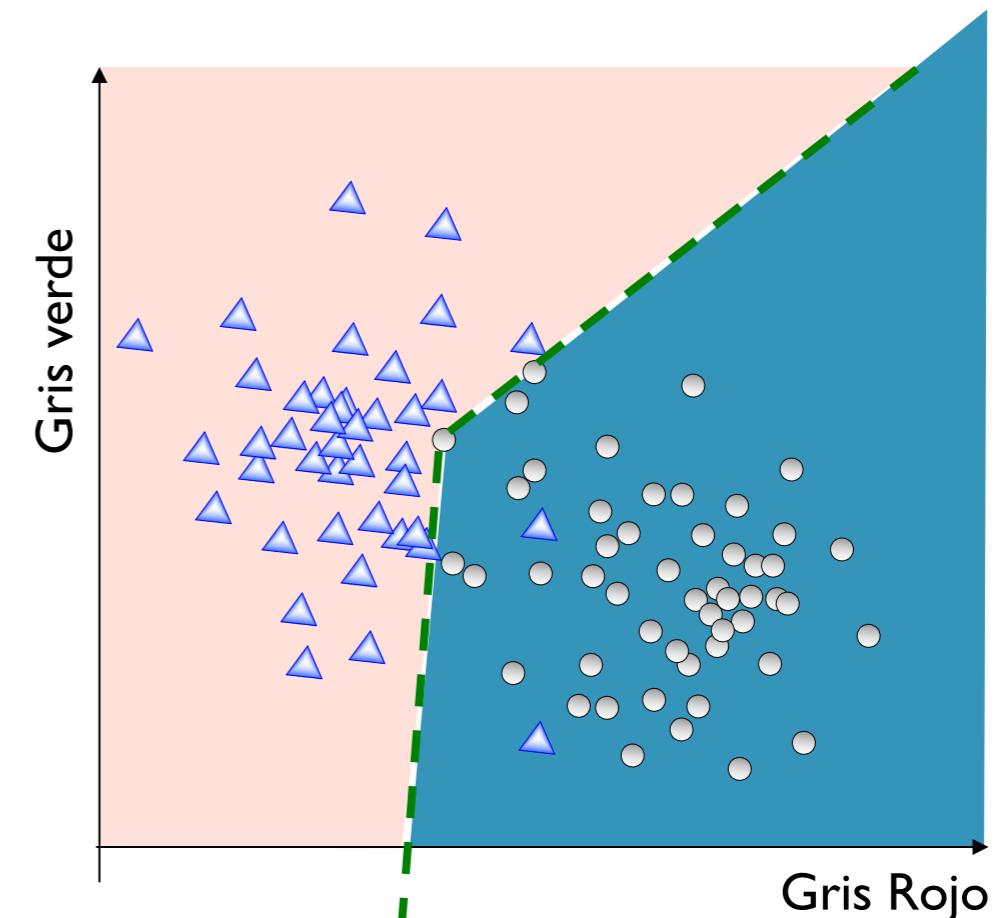
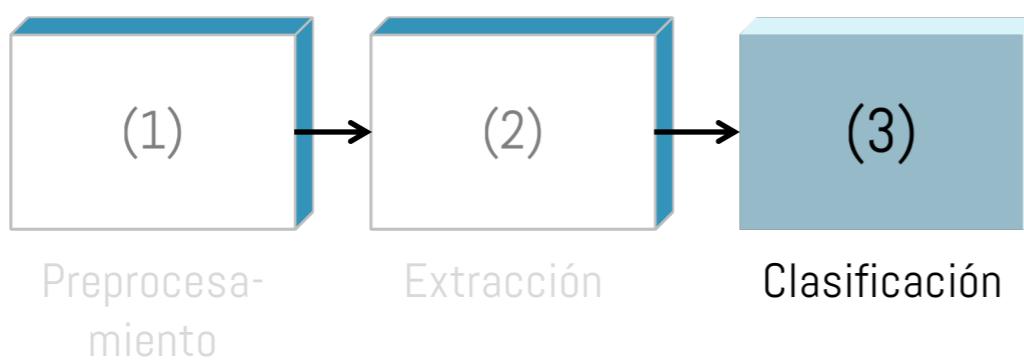
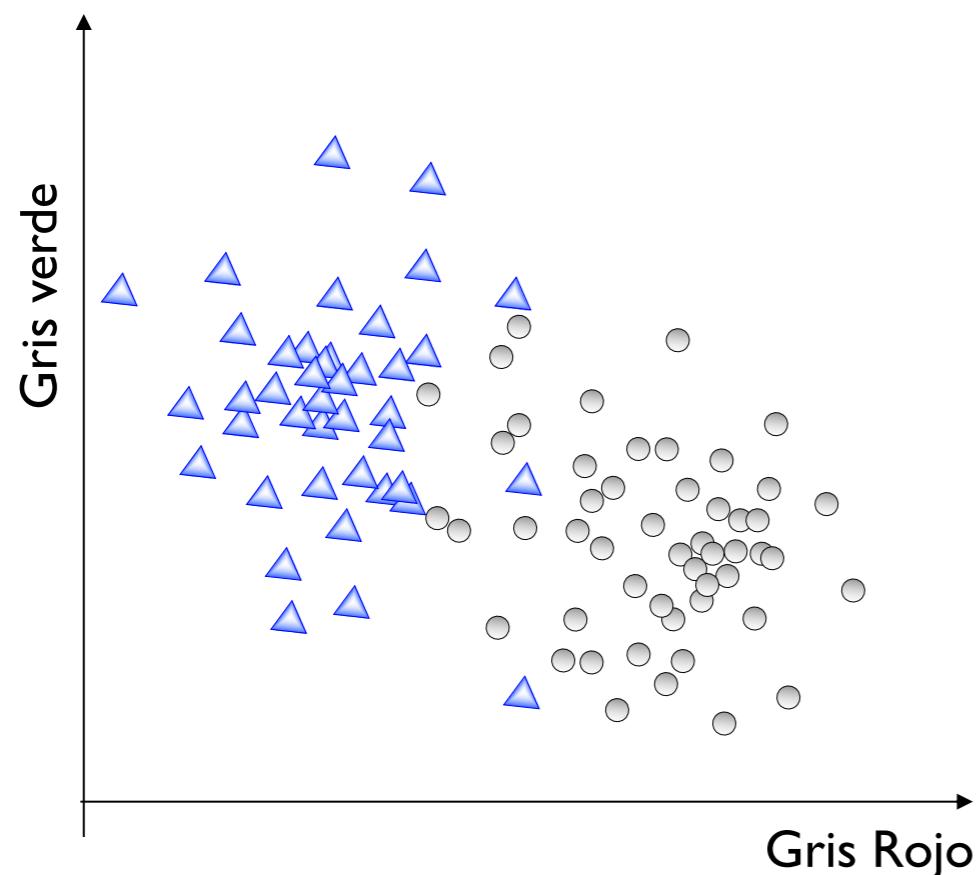
Línea de decisión

## Clasificación



Supongamos que tenemos dos clases en vez de una

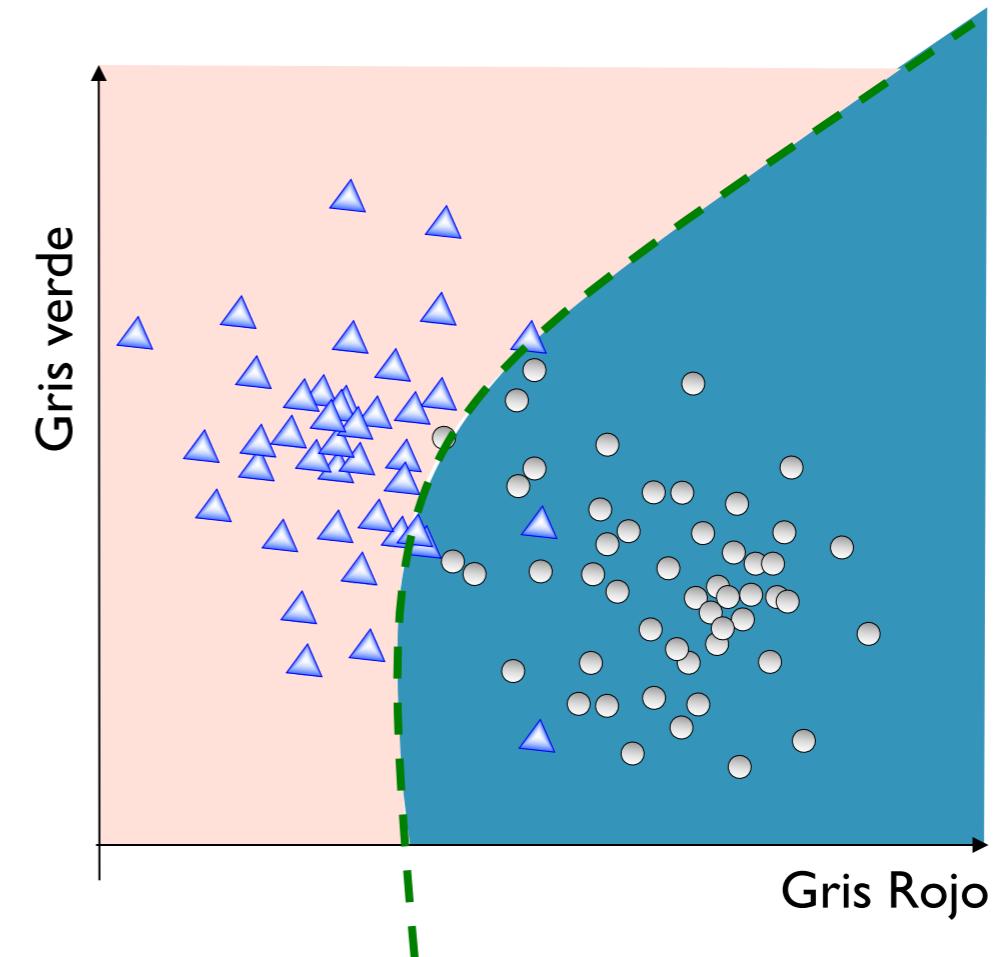
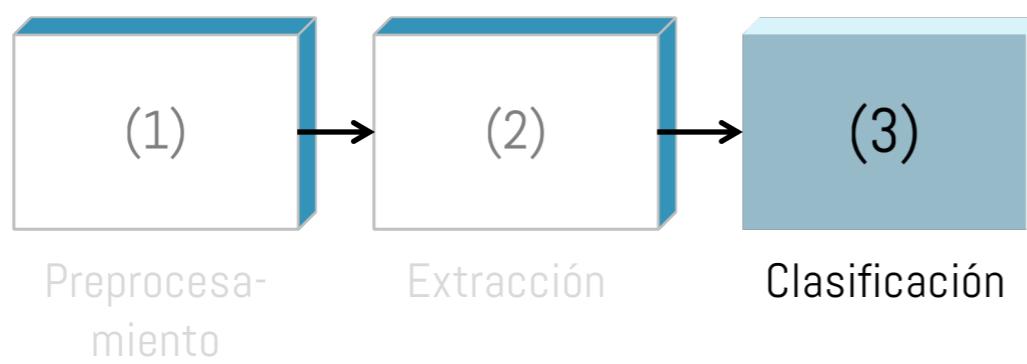
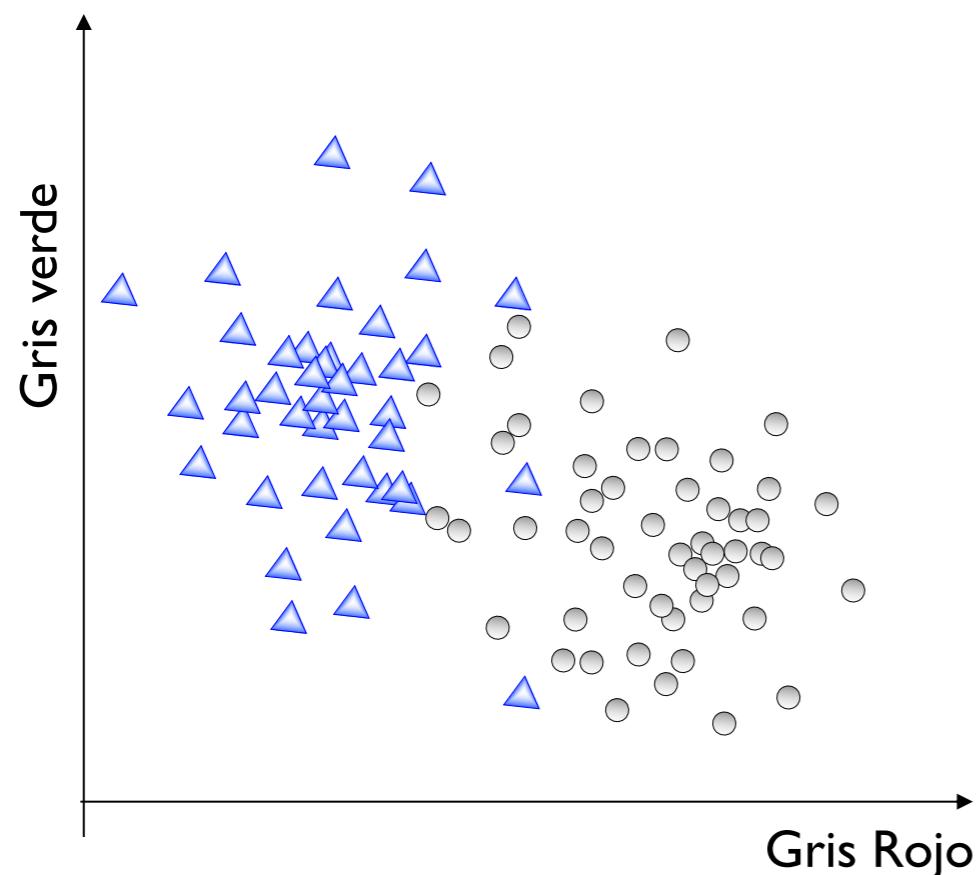
## Clasificación



Línea de  
decisión

Esta línea es menos  
general que la anterior

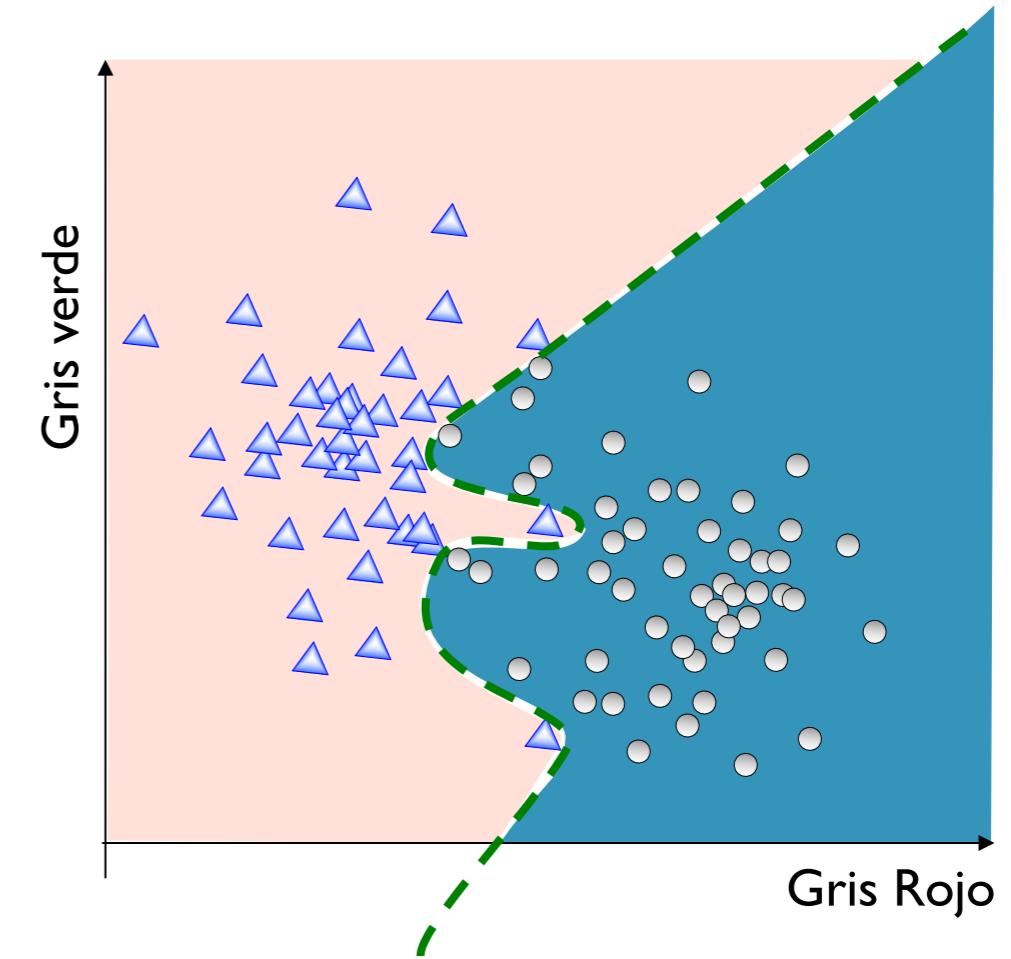
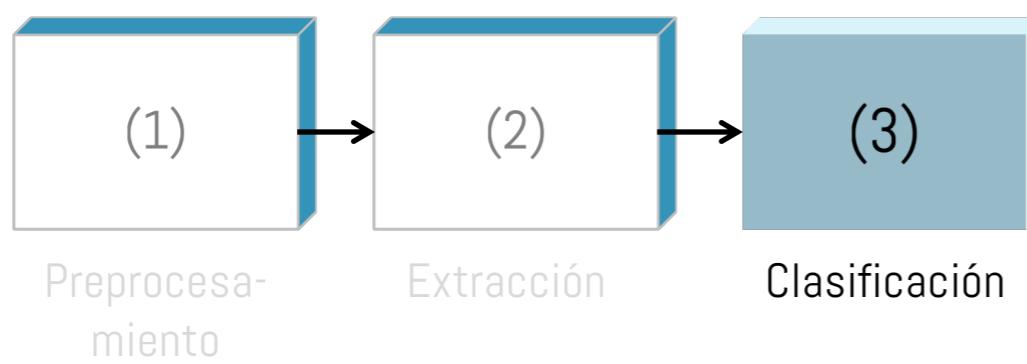
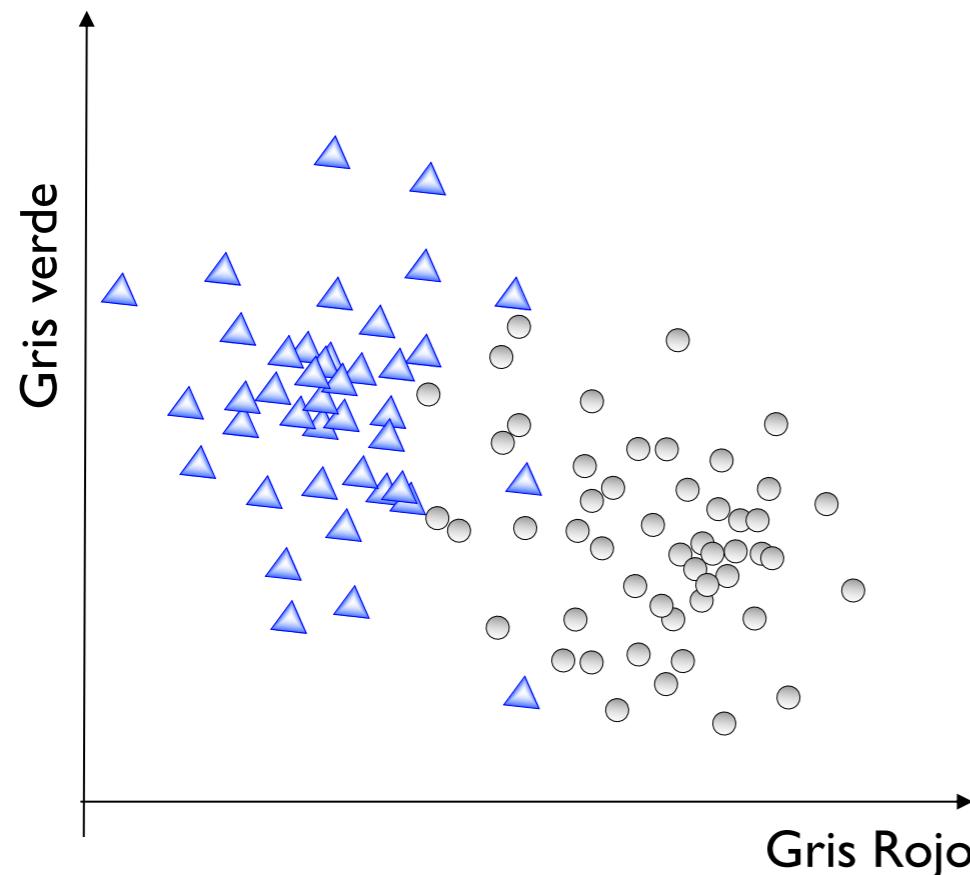
## Clasificación



Línea de  
decisión

Esta línea responde  
a una función

## Clasificación



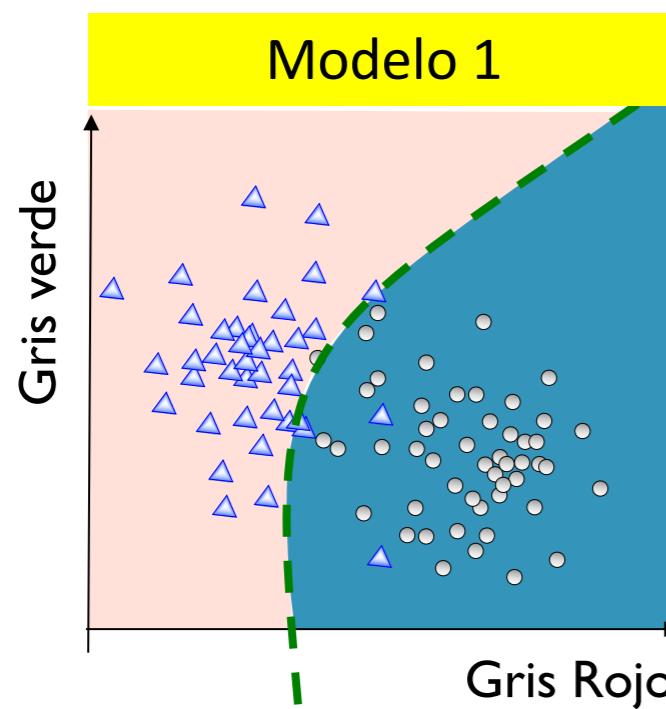
Línea de  
decisión

**SOBRE  
ENTRENAMIENTO!!  
muy muy malo**

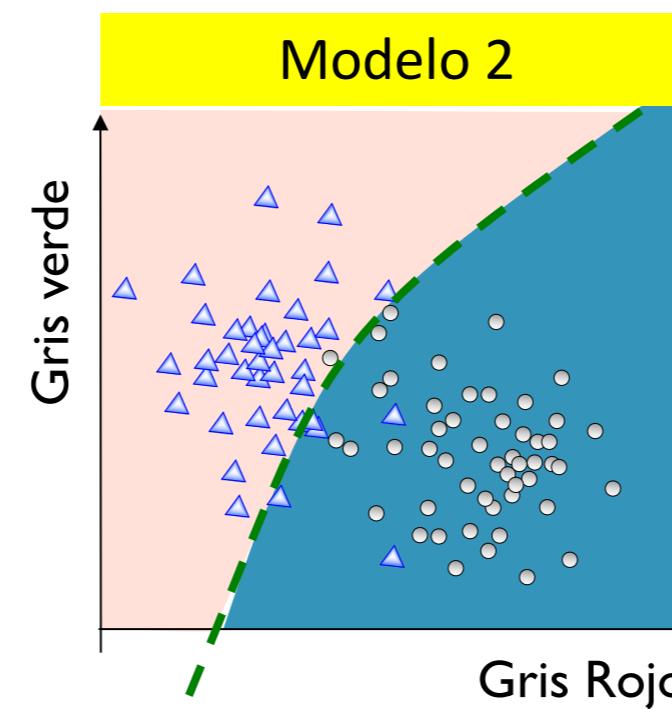


Podemos evaluar un número infinito de modelos, sin embargo, existe un que se ajuste mejor al conjunto de datos del problema. Para evaluar cada modelo debemos:

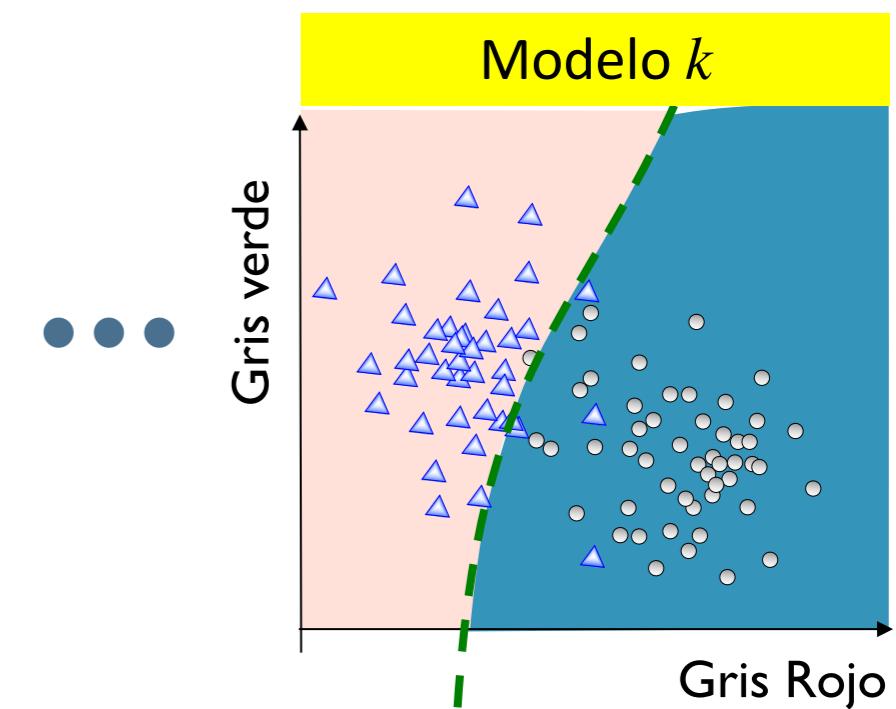
1. Definir un espacio de posibles modelos:  $M = \{M_1, M_2, \dots, M_k\}$
2. Poblar el espacio de modelos (con diferentes estructuras y parámetros)
3. Evaluar cada modelo con una función de puntaje para determinar cuál modelo ajusta mejor los datos



$$Score(M_1) = 1.2$$



$$Score(M_2) = 1.8$$



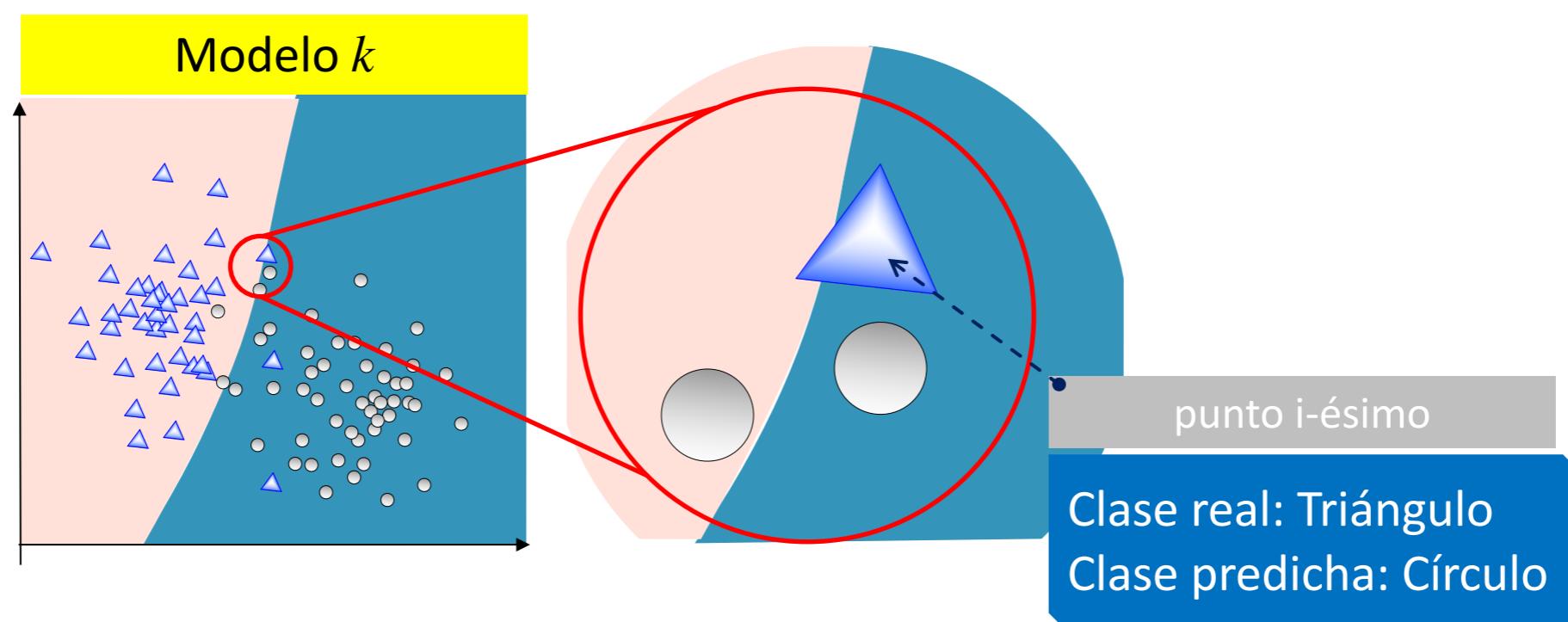
$$Score(M_k) = 2.1$$



Para medir el rendimiento del clasificador debemos medir la calidad de las predicciones en un conjunto de datos.

Esto significa que para cada punto i-ésimo, debemos medir la diferencia entre la clase real y la clase predicha.

**El objetivo** es rankear todos los modelos en términos de su utilidad con el fin de buscar el "mejor" modelo que se ajuste al dataset.





Una formulación general del rendimiento del modelo es:

$$S(M) = \sum_{i=1}^{N_{test}} d[f(x(i); M), y(i)]$$

Suma de errores para cada punto

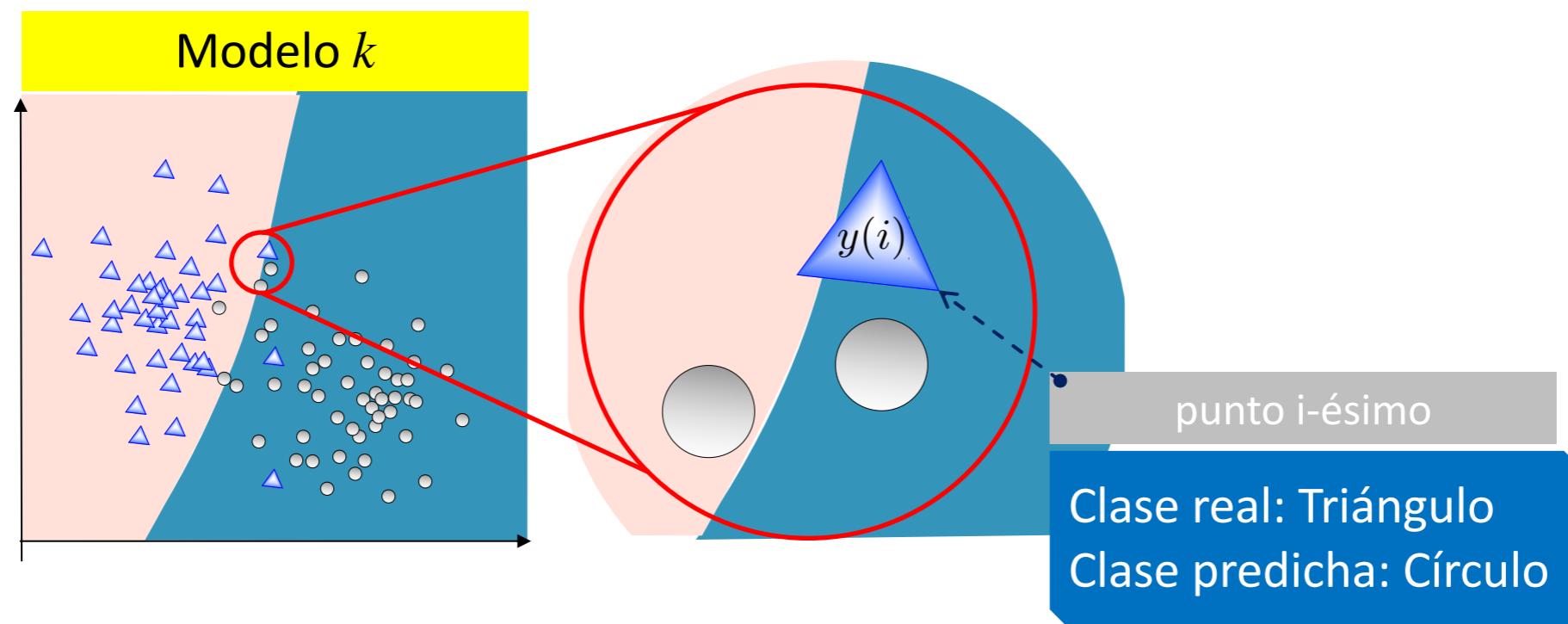
distancia

punto

clase predicha para el punto i

clase real

modelo





## Funciones de evaluación

Zero  
one-loss

$$S_{0/1}(M) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} I[f(x(i); M), y(i)]$$

where  $I(a, b) = \begin{cases} 1 & a \neq b \\ 0 & \text{otherwise} \end{cases}$

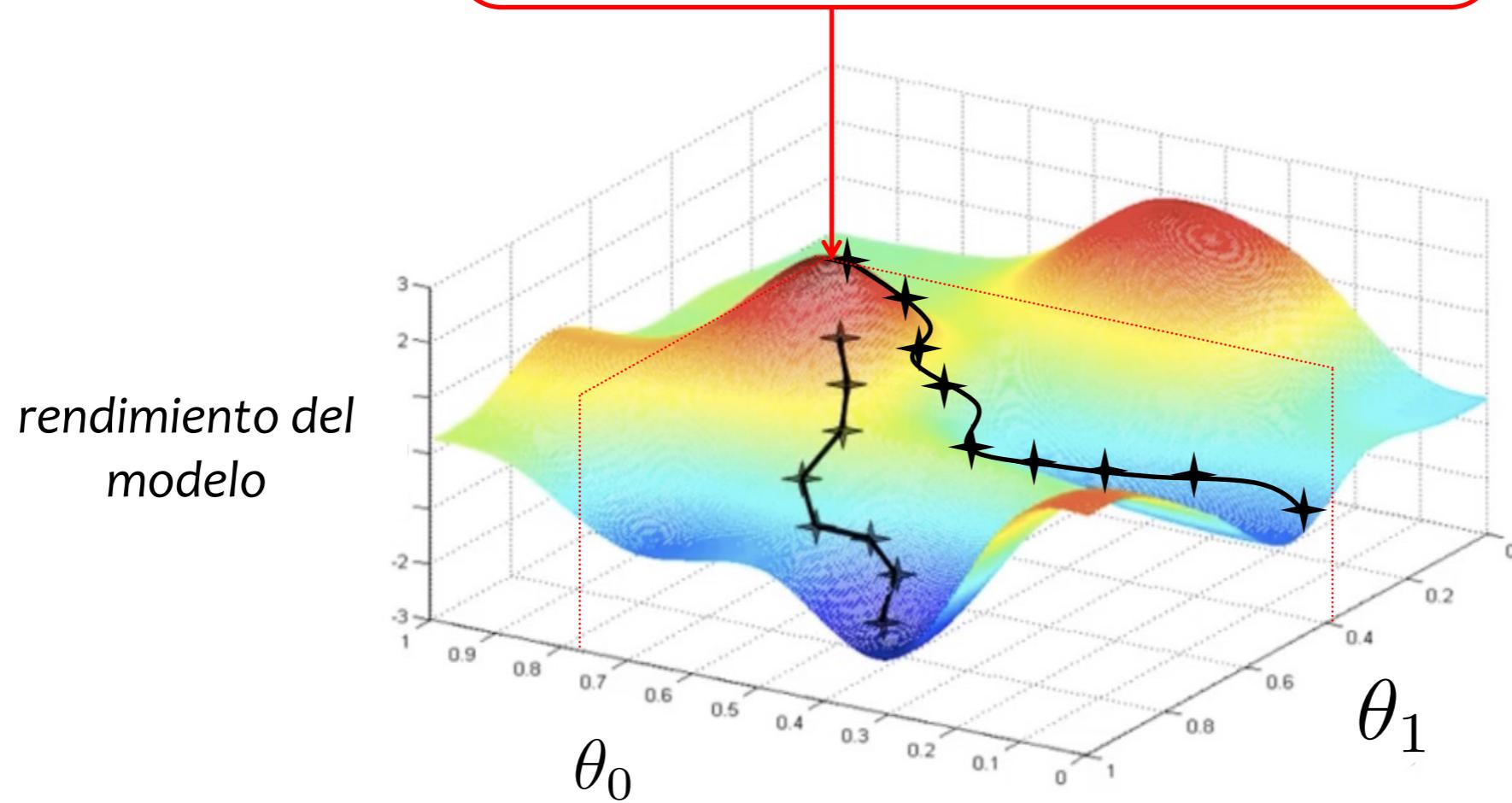
Squared  
one-loss

$$S_{sq}(M) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} [f(x(i); M) - y(i)]^2$$



**Búsqueda de parámetros:** A medida que modificamos los parámetros del modelo, generamos diferentes rendimientos los cuales generan una superficie en el espacio paramétrico  $\theta$ .

Mejor modelo  $\approx (\theta_0 = 0.77, \theta_1 = 0.4)$



- Modelación predictiva
- Métricas de evaluación de rendimiento

## ■ Rendimiento de clasificación binaria

- **EJEMPLO:** Suponga que usted diseña un software para determinar automáticamente si usted enfermo o sano a través de un algoritmo de procesamiento de imágenes.
- Luego de una posterior clasificación, estos fueron los resultados. De **100 personas sanas**, el sistema predijo que 95 estaban sanas y 5 enfermas. De **80 personas enfermas**, el sistema predijo que 10 estaban sanas y 70 enfermas

		predicho	
		Negative	Positive
es	Negative	TN	FP
	Positive	FN	TP

		predicho		$\Sigma$
		Sano	Enfermo	
es	Sano	95	5	100
	Enfermo	10	70	80

True Positive Rate o Recall	True Negative Rate
$TPR = \frac{70}{80}$	$TNR = \frac{95}{100}$
False Positive Rate (Error T.I)	False Negative Rate (Error T.2)
$FPR = \frac{5}{100}$	$FNR = \frac{10}{80}$

Sensibilidad	Precisión
$S_n = \frac{70}{70+10}$	$P = \frac{70}{5+70} = 0.933$
Especificidad	F-Score
$S_p = \frac{95}{5+95}$	$F = 2 \times \frac{0.875 \times 0.933}{0.875 + 0.933}$

Más medidas en  
[http://en.wikipedia.org/wiki/Sensitivity\\_and\\_specificity](http://en.wikipedia.org/wiki/Sensitivity_and_specificity)

## ■ Rendimiento de clasificación binaria

### Resultados

- 1. La **sensibilidad** es 87.5%.
  - 2. La **especificidad** es 95%.
  - 3. La **precisión** es 93%
  - 4. El **F-Score** es 90.32%
- } **El óptimo es 100%**
- 
- 5. El **Error Tipo 1 o FPR** es de un 5%
  - 6. El **Error tipo 2 o FNR** es de un 12.5%
- } **El óptimo es 0%**

Existen muchas técnicas de clasificación de optimización que buscan lograr el óptimo. Sin embargo, no siempre podrá ser determinado. En dicho caso deberemos determinar nuestro propio objetivo.

### Rendimiento

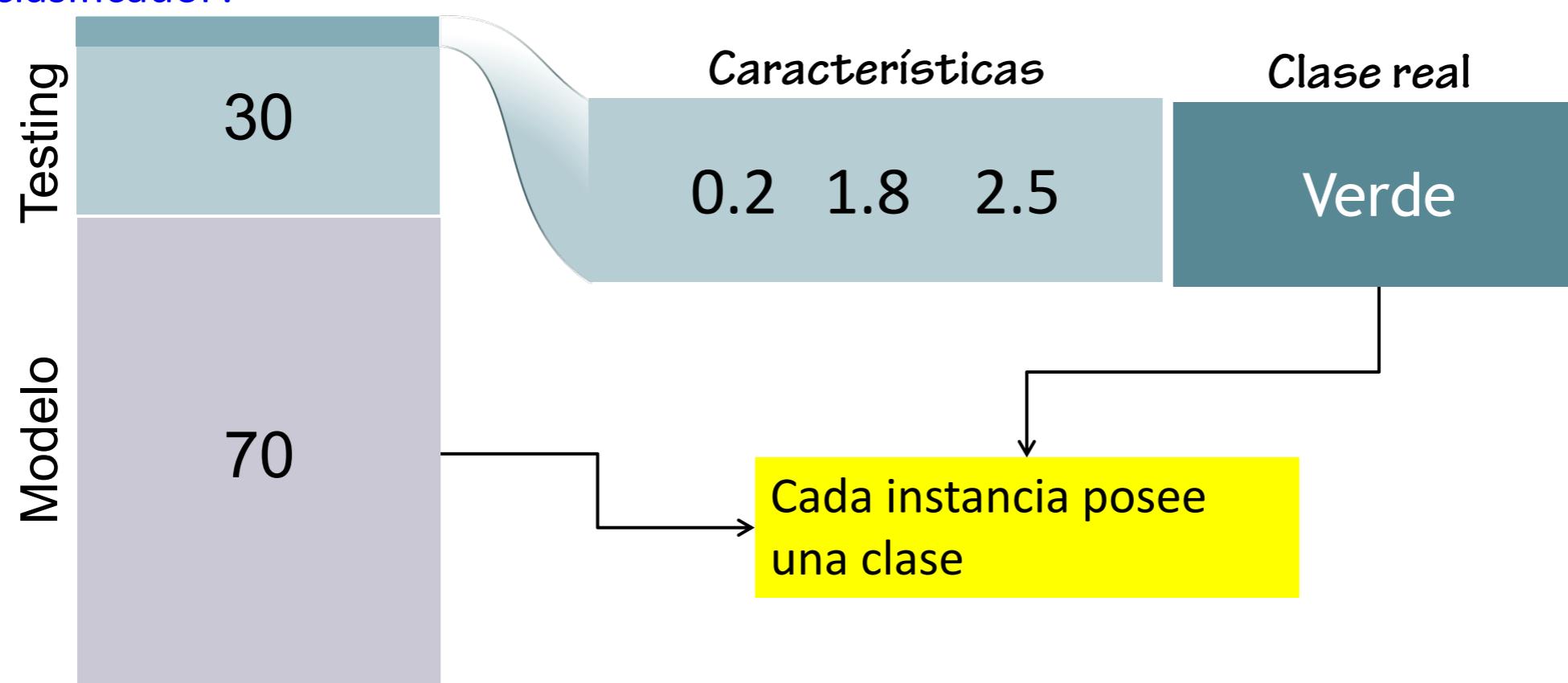
True Positive Rate o Recall	True Negative Rate
$TPR = \frac{70}{80}$	$TNR = \frac{95}{100}$
False Positive Rate (Error T.I)	False Negative Rate (Error T.2)
$FPR = \frac{5}{100}$	$FNR = \frac{10}{80}$

Sensibilidad	Precisión
$S_n = \frac{70}{70+10}$	$P = \frac{70}{5+70} = 0.933$
Especificidad	F-Score
$S_p = \frac{95}{5+95}$	$F = 2 \times \frac{0.875 \times 0.933}{0.875 + 0.933}$



## Ejemplo 1

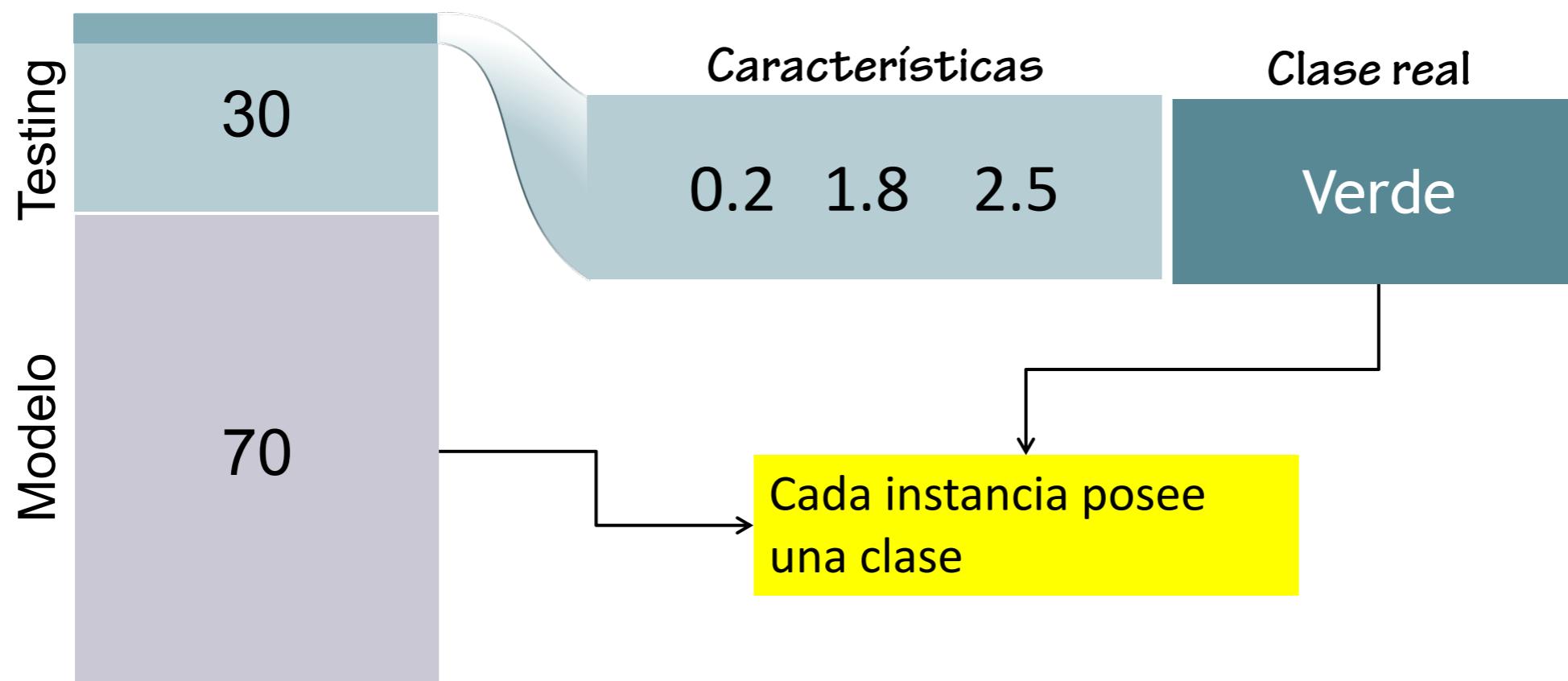
- Supongamos que usted ha diseñado un sistema de visión industrial para clasificar frutas en una de las tres categorías existentes [*verde*, *roja*, *madura*]. En la etapa de diseño usted clasificó manualmente 100 frutas ([clasificación supervisada](#)).
- Usted desea emplear un clasificador no-paramétrico, en particular el algoritmo kNN. Para verificar el número óptimo de vecinos del algoritmo, usted decide emplear 70% de los datos para entrenar su modelo y 30% para testear. [¿Cómo determinamos el rendimiento del clasificador?](#)





## Ejemplo 1

Vamos a suponer que usted no conoce la clase de la instancia de test y desea conocer su clase. Al ingresar la instancia al algoritmo kNN usted debe determinar la distancia de dicho punto **respecto al 70% de los datos**. De esta forma, usted debe emplear el clasificador kNN sobre cada una de las instancias del conjunto de prueba o testing.

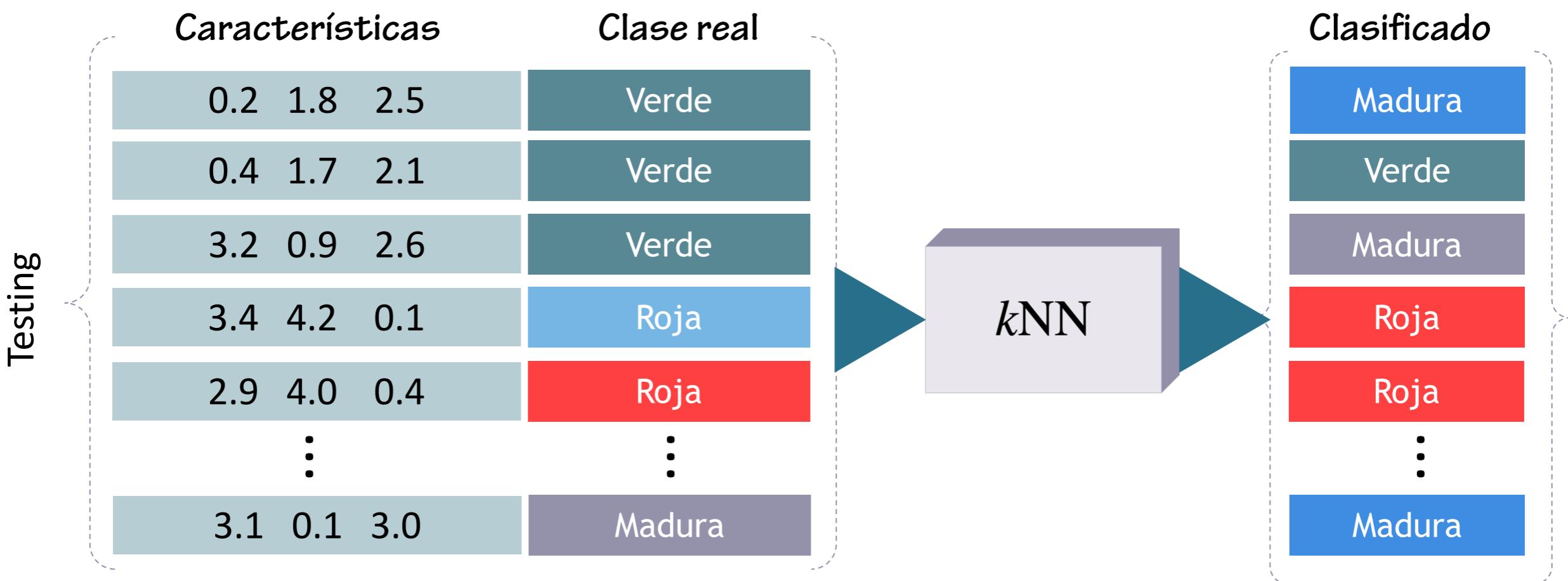




## ■ Ejemplo 1

De los datos de prueba o testing, usted sabe que hay **8 frutas** que son efectivamente **clase verde**, hay unas **10 frutas** de **clase roja** y finalmente **12 frutas** de **clase madura**. En el conjunto de prueba sólo hay 30 instancias

- Por cada dato o instancia de prueba usted debe determinar la clasificación del algoritmo kNN. Esto genera un resultado de clasificación.





## Ejemplo 1

Luego de clasificar con el algoritmo kNN sobre 30 datos de prueba o testing, obtenemos los resultados:

- De las 8 frutas verdes, kNN predijo que **cinco son verdes**, **dos** son maduras y solo **una** roja.
- De las 10 frutas rojas, kNN predijo que **siete** son rojas, **dos** son verdes y **una** es madura
- De las 12 frutas maduras, kNN predijo que **once** son maduras, y **una** es verde.

### Matriz de confusión

Empleando los datos anteriores, procedemos a determinar la matriz de confusión

		predicho o clasificado			<p>Lo predicho o clasificado es el resultado del claseficador</p>
		Verde	Roja	Madura	
Datos reales	Verde	5	1	2	8
	Roja	2	7	1	10
	Madura	1	0	11	12



## Ejemplo 2

Supongamos que usted ha diseñado un sistema de clasificación de animales empleando únicamente un sistema de visión artificial. Este sistema fue **evaluado** en **60** animales, de los cuales sabemos que 10 son perros, 20 son gatos y 30 son conejos. El resultado del clasificador es el siguiente.

- De los 10 perros, el clasificador predijo que **seis** son perros, **tres** son gatos y solo **un** conejo.
- De los 20 gatos, el clasificador predijo que **trece** son gatos, **cuatro** son perros y **tres** son conejos
- De los 30 conejos, el clasificador predijo que **veintidós** son conejos, y **ocho** son gatos.

### Matriz de confusión

Empleando los datos anteriores,  
procedemos a determinar la  
matriz de confusión

		→ predicho o clasificado			
		Perro	Gato	Conejo	$\Sigma$
Datos reales	Perro	6	3	1	10
	Gato	4	13	3	20
	Conejo	0	8	22	30

Lo predicho o  
clasificado es el  
resultado del  
clasificador

- A partir de la matriz de confusión determinamos el rendimiento de la clasificación

		→ predicho o clasificado		
		Perro	Gato	Conejo
Datos reales	Perro	6	3	1
	Gato	4	13	3
	Conejo	0	8	22

True Positive Rate (TPR)

Recall o Sensibilidad

Perro	$TPR = \frac{6}{6+3+1} = 0.6$
Gato	$TPR = \frac{13}{4+13+3} = 0.65$
Conejo	$TPR = \frac{22}{22+8} = 0.73$

- True Positive Rate o bien también conocido como Recall o también bien Sensibilidad es la proporción de datos correctamente clasificados según su clase versus el total de datos de dicha clase.

$$TPR(Z) = \frac{\# \text{ instancias bien clasificadas de la clase } Z}{\text{Total de instancias de la clase } Z}$$

- En el ejemplo, la clase Perro fueron clasificados correctamente 6 perros. Dicho número se divide por el total de instancias reales de la clase perro, que en este caso sabemos de antemano que son 10.

- A partir de la matriz de confusión determinamos el rendimiento de la clasificación

		→ predicho o clasificado		
		Perro	Gato	Conejo
Datos reales	Perro	6	3	1
	Gato	4	13	3
	Conejo	0	8	22

## False Positive Rate (FPR)

Perro

$$FPR = \frac{4}{60 - 10} = 0.08$$

Gato

$$FPR = \frac{8 + 3}{60 - 20} = 0.275$$

Conejo

$$FPR = \frac{1 + 3}{60 - 30} = 0.13$$

- False Positive Rate* es la proporción de falsas alarmas clasificadas como Z versus la diferencia entre el total de instancias del problema y las instancias de la clase Z. (esto corresponde al conjunto complemento)

$$FPR(Z) = \frac{\text{# falsas alarmas clasificadas de la clase } Z}{\text{Total de instancias} - \text{Total de instancias de la clase } Z}$$

- En el ejemplo, la clase *Gato* fueron clasificados incorrectamente (3+8) gatos. Dicho número se divide por la diferencia entre el total de instancias del problema (60) y el total de instancias de la clase gato (20), es decir 40 instancias no gato.

- A partir de la matriz de confusión determinamos el rendimiento de la clasificación

		predicho o clasificado		
		Perro	Gato	Conejo
Datos reales	Perro	6	3	1
	Gato	4	13	3
	Conejo	0	8	22

## Especificidad o Specificity o TNR

Perro

$$E = 1 - FPR = 0.92$$

Gato

$$E = 1 - FPR = 0.725$$

Conejo

$$E = 1 - FPR = 0.87$$

- Especificidad* es la proporción entre verdaderos negativos versus la diferencia entre el total de instancias del problema y las instancias de la clase Z. O bien es
- $$E(Z) = 1 - FPR(Z)$$
- En el ejemplo, las instancias que no corresponden a la clase *Perro* o bien True Negative son  $(13+3+8+22)$ . Dicho número se divide por la diferencia entre el total de instancias del problema (60) y el total de instancias de la clase perro (10).

$$E(\text{Perro}) = \frac{13+3+8+22}{60-10} = 0.92 = 1 - FPR(\text{Perro})$$

- A partir de la matriz de confusión determinamos el rendimiento de la clasificación

		→ predicho o clasificado		
		Perro	Gato	Conejo
Datos reales	Perro	6	3	1
	Gato	4	13	3
	Conejo	0	8	22

## Precision o PPV

Perro

$$P = \frac{6}{6+4} = 0.6$$

Gato

$$P = \frac{13}{3+13+8} = 0.54$$

Conejo

$$P = \frac{22}{1+3+22} = 0.85$$

- Precision* es el número de instancias bien clasificadas de la clase Z versus el total de instancias de la clase Z

$$P(Z) = \frac{\# \text{ instancias bien clasificadas de la clase Z}}{\text{Total de instancias clasificadas como clase Z}}$$

- En el ejemplo, la clase *Conejo* fueron clasificados correctamente 22 conejos. Dicho número se divide por el total de instancias clasificadas como conejos (1+3+22).

# Matriz de Confusión

## Resumiendo

		predicho o clasificado			F-Score = $2 \times \frac{R \times P}{R + P}$
		Perro	Gato	Conejo	
Datos reales	Perro	6	3	1	
	Gato	4	13	3	
	Conejo	0	8	22	

	TPR - Recall	FPR	Specificity	Precision	F-score
Perro	0.6	0.08	0.92	0.6	$2 \times \frac{0.6 \times 0.6}{0.6 + 0.6} = 0.6$
Gato	0.65	0.275	0.725	0.54	$2 \times \frac{0.65 \times 0.54}{0.65 + 0.54} = 0.589$
Conejo	0.73	0.13	0.87	0.85	$2 \times \frac{0.73 \times 0.85}{0.73 + 0.85} = 0.785$

- **Accuracy** es un índice para toda una matriz de Confusión. En síntesis corresponde a la suma de los TP versus el total de instancias.

$$Acc = \frac{TP}{TOTAL} = \frac{6+13+22}{10+20+30} = 0.68$$

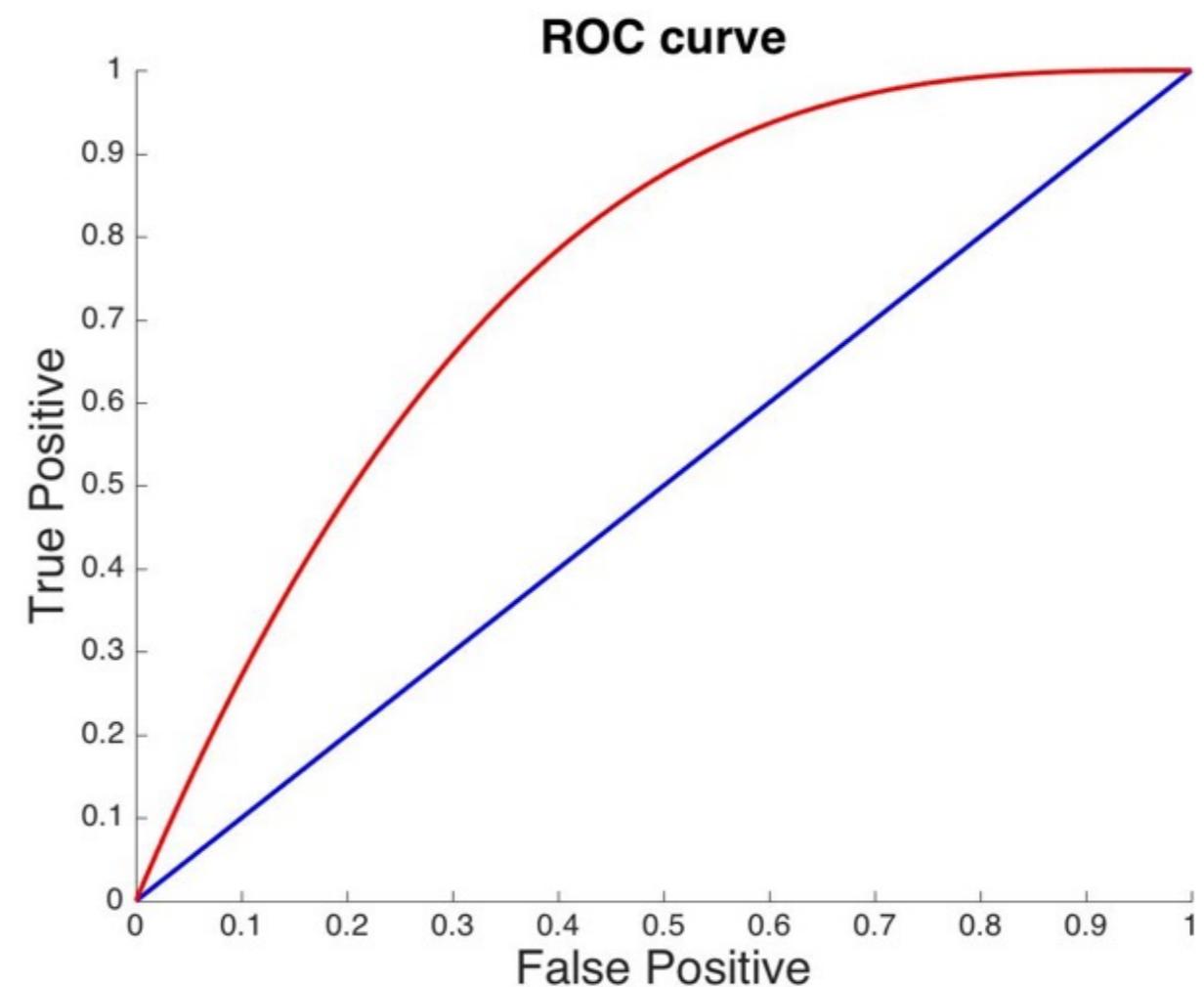


El rendimiento del clasificador permite evaluar **el porcentaje de error que genera la clasificación**. Para evaluar dicho rendimiento se construye **la matriz de confusión**. A través de ella Podemos determinar

- **True Positive Rate** o bien también conocido como **Recall** o también bien **Sensibilidad** es la proporción de datos correctamente clasificados según su clase versus el total de datos de dicha clase.
  - **False Positive Rate** es la proporción de falsas alarmas clasificadas como Z versus la diferencia entre el total de instancias del problema y las instancias de la clase Z. (esto corresponde al conjunto complemento)
  - **Especificidad (o TNR)** es la proporción entre verdaderos negativos versus la diferencia entre el total de instancias del problema y las instancias de la clase
  - **Precision o PPV** es el número de instancias bien clasificadas de la clase Z versus el total de instancias de la clase Z
- 
  - **Accuracy** es un índice para toda una matriz de Confusión. En síntesis corresponde a la suma de los TP versus el total de instancias.

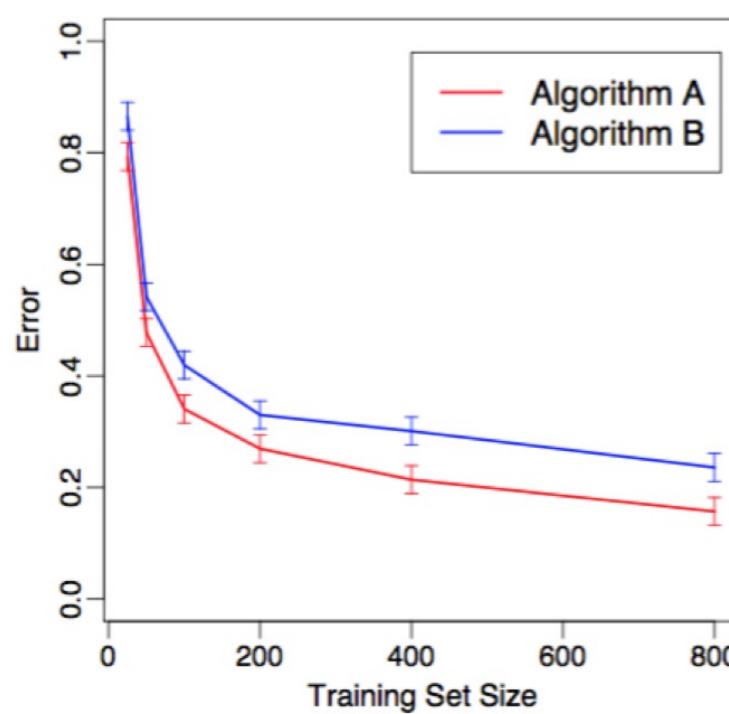
- Modelación predictiva
- Técnicas de selección
  - Hold-out estimate
  - Cross-validation
  - Bootstrap
- Evaluación de modelos
  - Curvas ROC

- La curva ROC fue desarrollada en 1950 como una teoría de señales para analizar señales ruidosas. La curva ROC permite al operador contrapesar la tasa de verdaderos positivos (eje-Y) versus los falsos positivos (eje-X)
- El rendimiento de cada clasificador representa un punto en la curva ROC. Para ello podemos algún parámetro del algoritmo, el umbral, la distribución, o la matriz de costos.
- El área bajo la curva (AUC) resumen el rendimiento del modelo

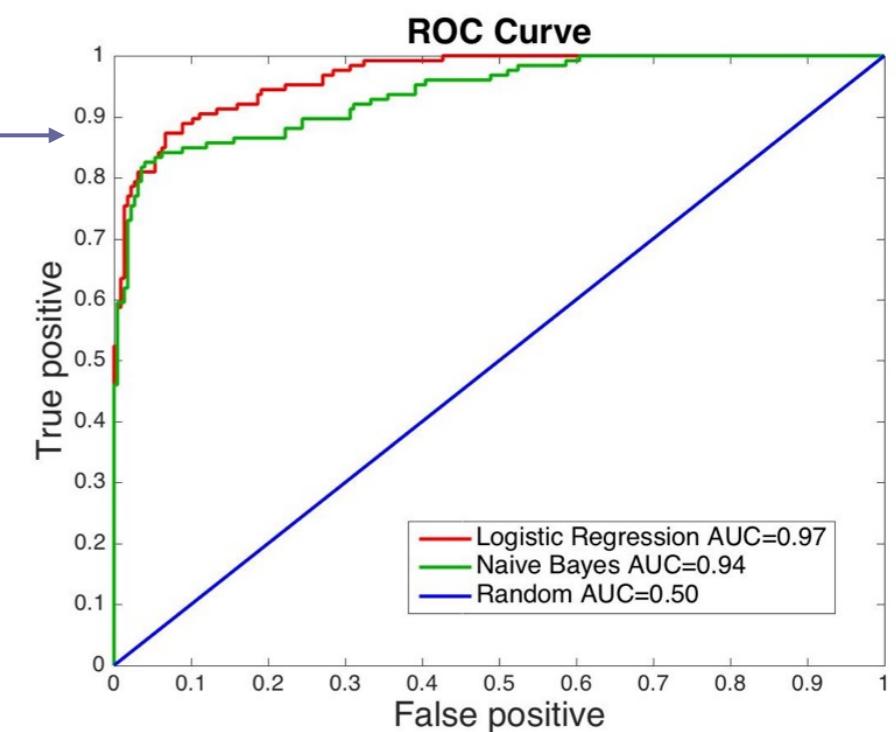


## k-cross validation

```
para j= 1 a m modelos:  
  para i= 1 a k pruebas:  
    aprende modelo j en el i-ésimo set de entrenamiento  
    evalúa modelo j en el i-ésimo set de prueba  
  promedia resultados del modelo j de todos las k pruebas  
  plotea el error con la desviación estándar
```

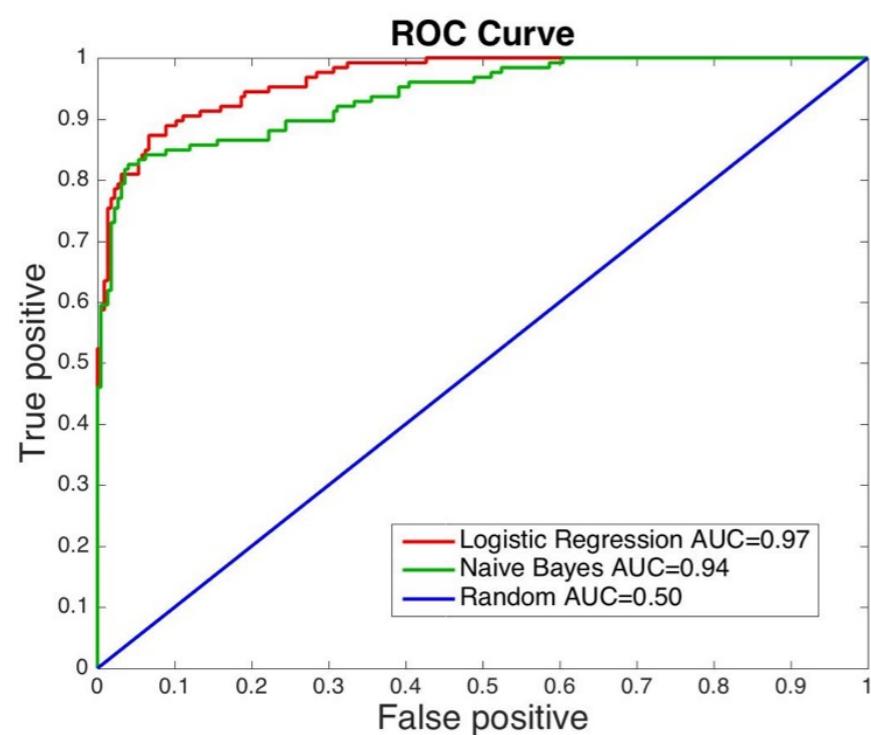


Cada punto de evaluación del modelo genera un punto en la curva ROC



## k-cross validation

```
para j= 1 a m modelos:  
  para i= 1 a k pruebas:  
    aprende modelo j en el i-ésimo set de entrenamiento  
    evalúa modelo j en el i-ésimo set de prueba  
  
  promedia resultados del modelo j de todos las k pruebas  
  plotea el error con la desviación estándar
```



El área bajo la curva se denomina AUC  
(area under the curve)

AUC resumen el rendimiento del modelo y de esta forma se pueden comparar





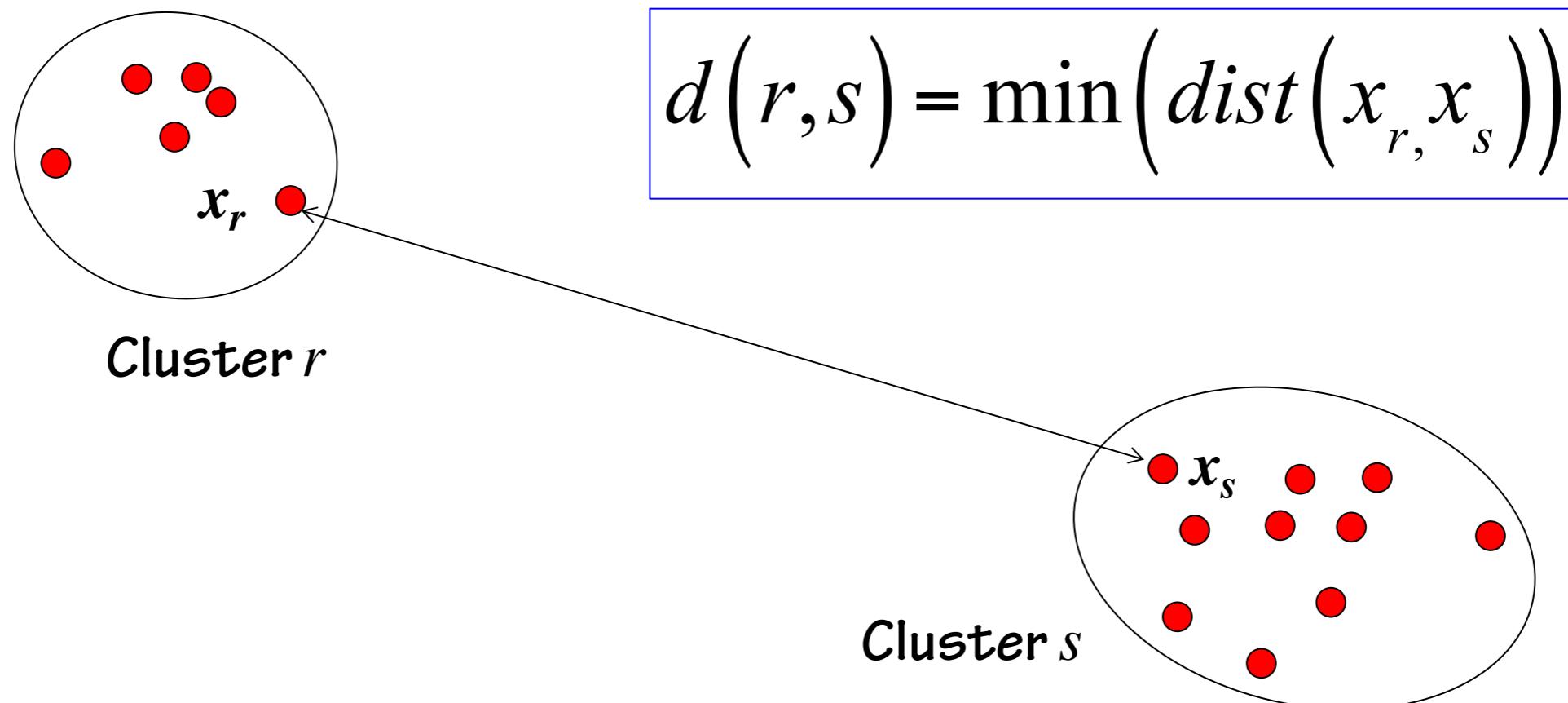
# Anexo

Métricas de agrupamiento



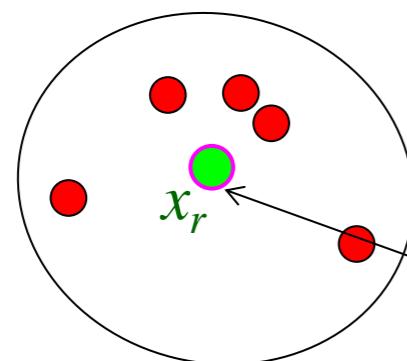
## ■ Métricas de agrupamiento

- Existe diversos tipos de técnicas de enlace para cada cluster, dentro de las más conocidas, están el enlace simple, enlace completo y enlace medio
- Enlace Simple: es la menor distancia entre entre todos los puntos de dos clusters



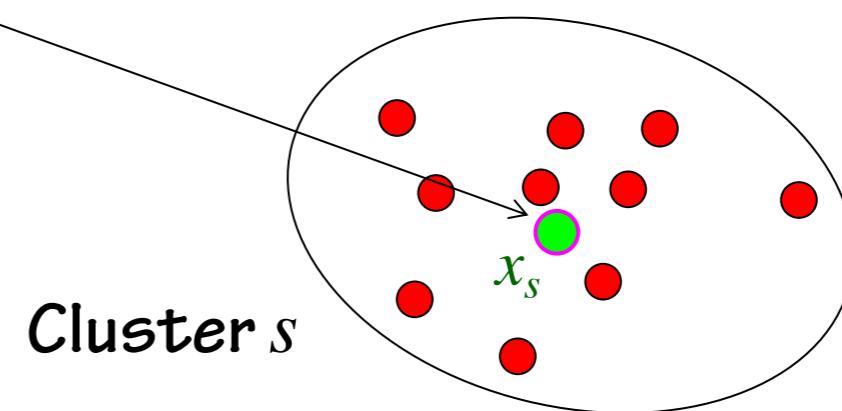
## ■ Métricas de agrupamiento

- Existe diversos tipos de técnicas de enlace para cada cluster, dentro de las más conocidas, están el enlace simple, enlace completo y enlace medio
- Enlace medio: es la distancia entre los promedios o centroides de dos clusters



Cluster  $r$

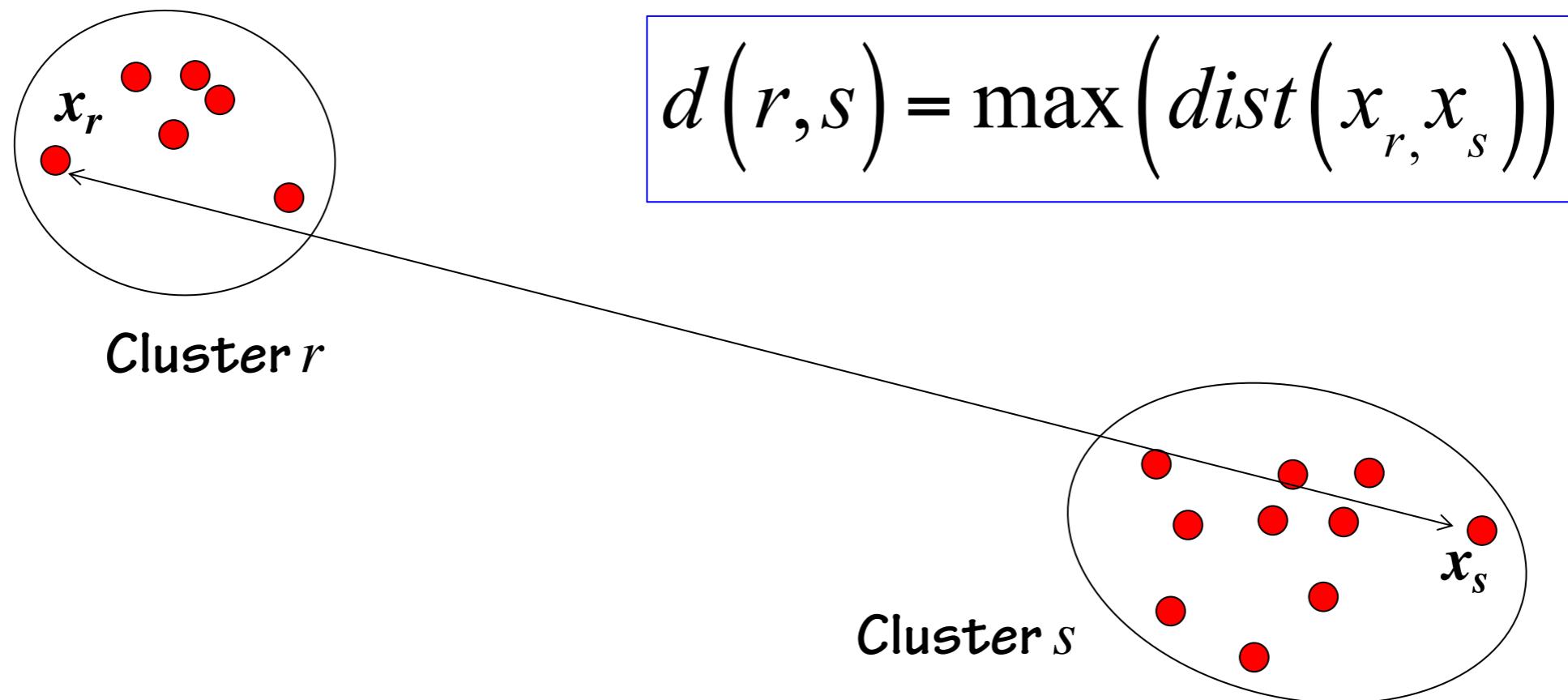
$$d(r, s) = \frac{1}{n_r \cdot n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})$$



Cluster  $s$

## ■ Métricas de agrupamiento

- Existe diversos tipos de técnicas de enlace para cada cluster, dentro de las más conocidas, están el enlace simple, enlace completo y enlace medio
- Enlace completo: es la máxima distancia entre todos los puntos de dos clusters.



## ■ Métricas de agrupamiento

- Existen diversas formas para medir la distancia entre dos vectores multidimensionales. Cada una depende del problema a resolver, de acuerdo a los criterios de optimización del problema o minimización del error. Uno de los más comunes es la distancia Euclídea.

Distancia Euclídea

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

El propósito de estas medidas, en conocer un valor numérico para la disimilaridad entre dos vectores.

Distancia CityBlock

$$d_{cb} = \sqrt{\sum_{i=1}^N |x_i - y_i|}$$

Distancia Chebyshev

$$d_{ch} = \max_i |x_i - y_i|$$

Distancia de Minkowski  
de orden  $m$

$$d_m = \left\{ \sum_{i=1}^N (x_i - y_i)^m \right\}^{\frac{1}{m}}$$

Coeficiente de  
correlación de Pearson

$$d_{pcc} = \frac{\sum_{i=1}^N (\mathbf{X}_i - \bar{\mathbf{X}}_i)^2 \cdot (\mathbf{Y}_i - \bar{\mathbf{Y}}_i)}{\sum_{i=1}^N (\mathbf{X}_i - \bar{\mathbf{X}}_i)^2 \cdot \sum_{i=1}^N (\mathbf{Y}_i - \bar{\mathbf{Y}}_i)^2}$$