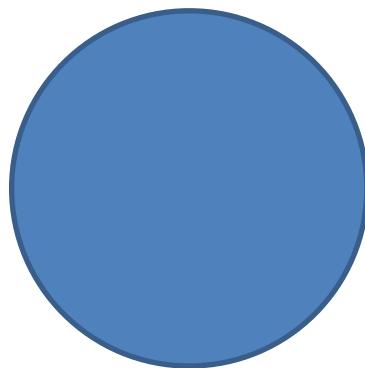


Machine Learning & Deep Learning

# Aprendizaje Supervisado y No Supervisado

Profesor: Carlos Moreno Morera





# Contenido

01

**Introducción al Aprendizaje Supervisado**

---

02

**Algoritmo general de optimización**

---

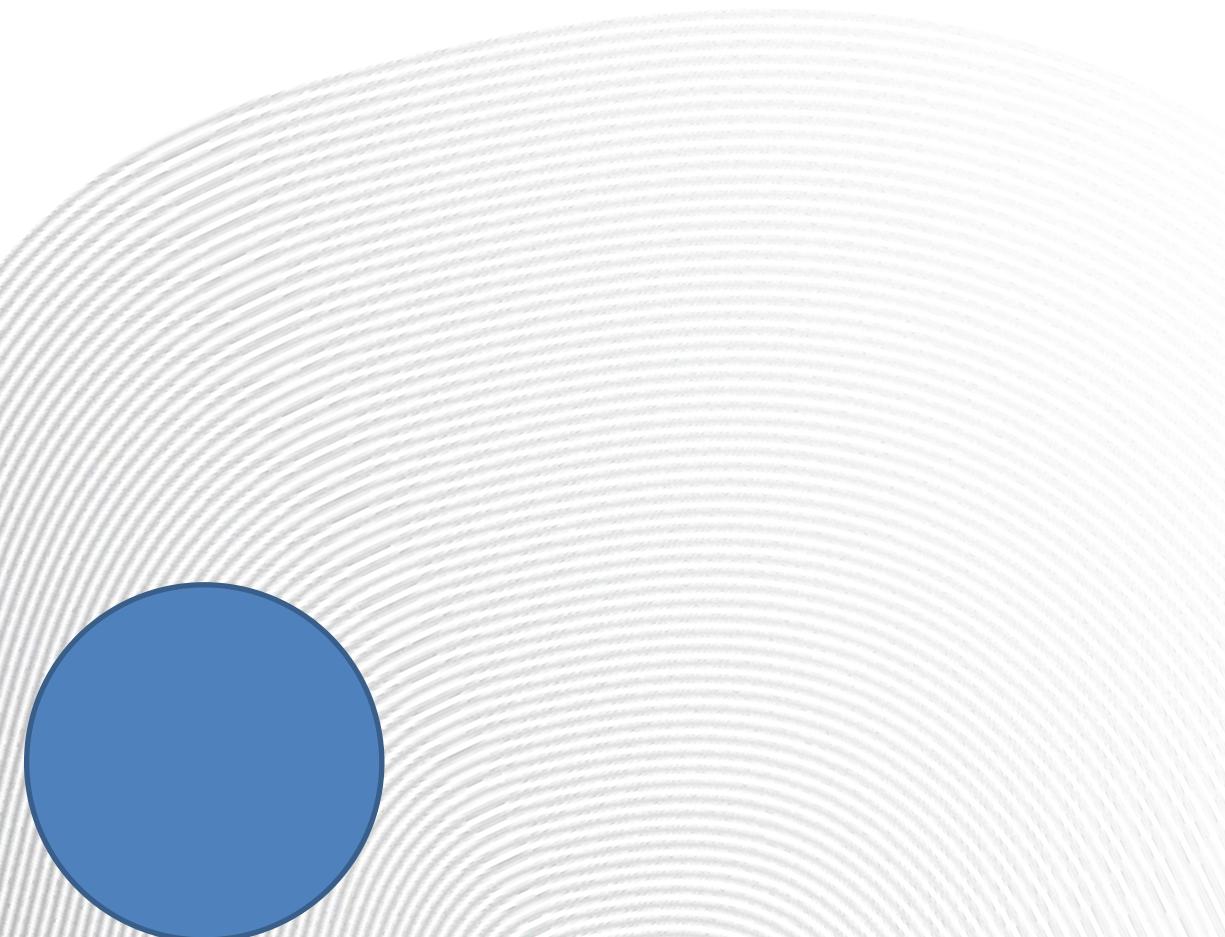
03

**La regresión lineal**

---

04

**Evaluación de modelos de clasificación**





# Contenido

05

Algoritmos de clasificación

06

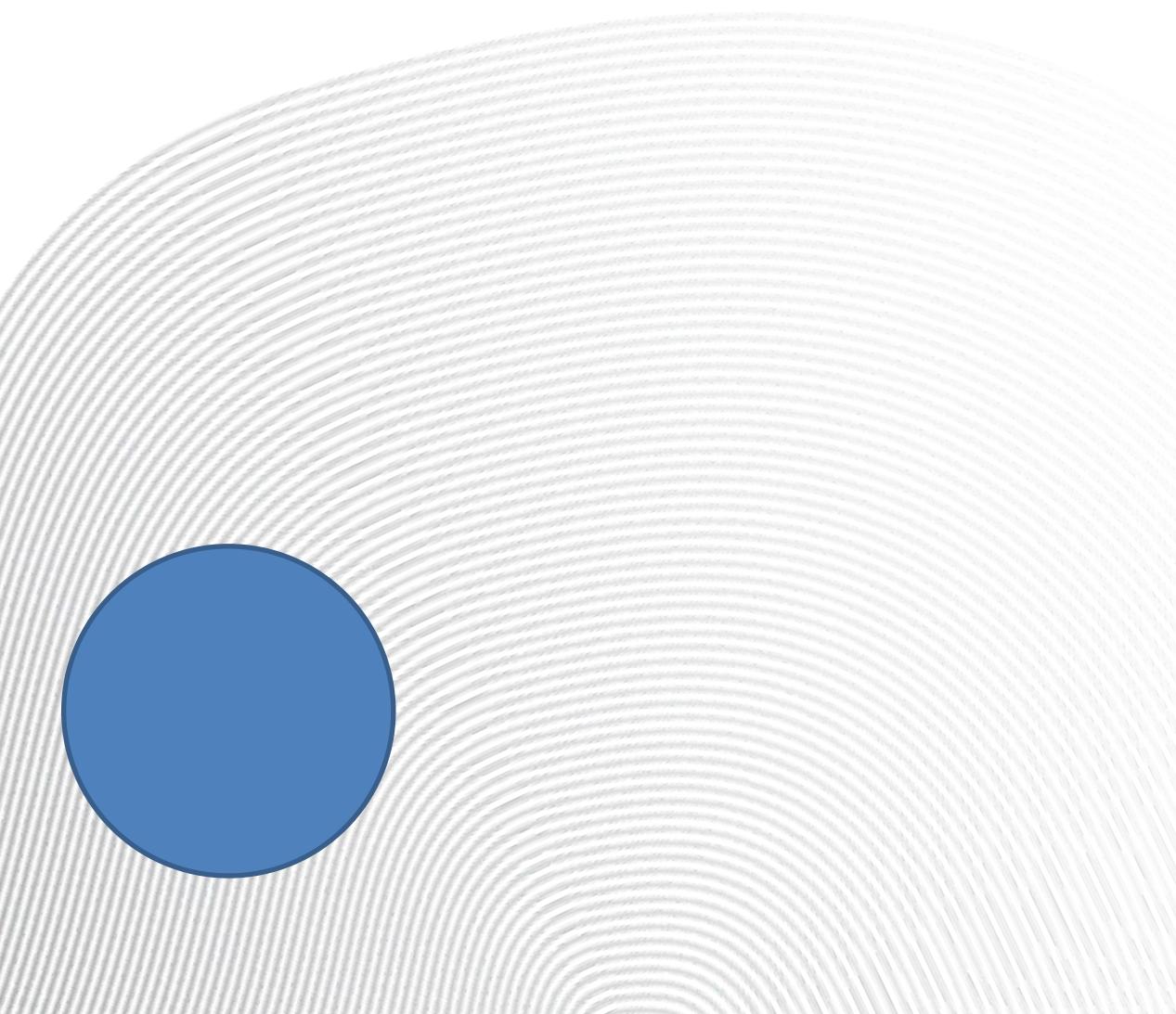
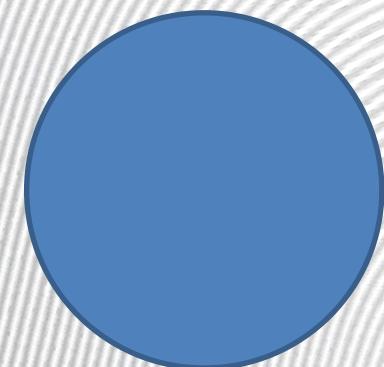
Evaluación de modelos de regresión

07

Algoritmos de regresión

08

Introducción al Aprendizaje No Supervisado





# Contenido

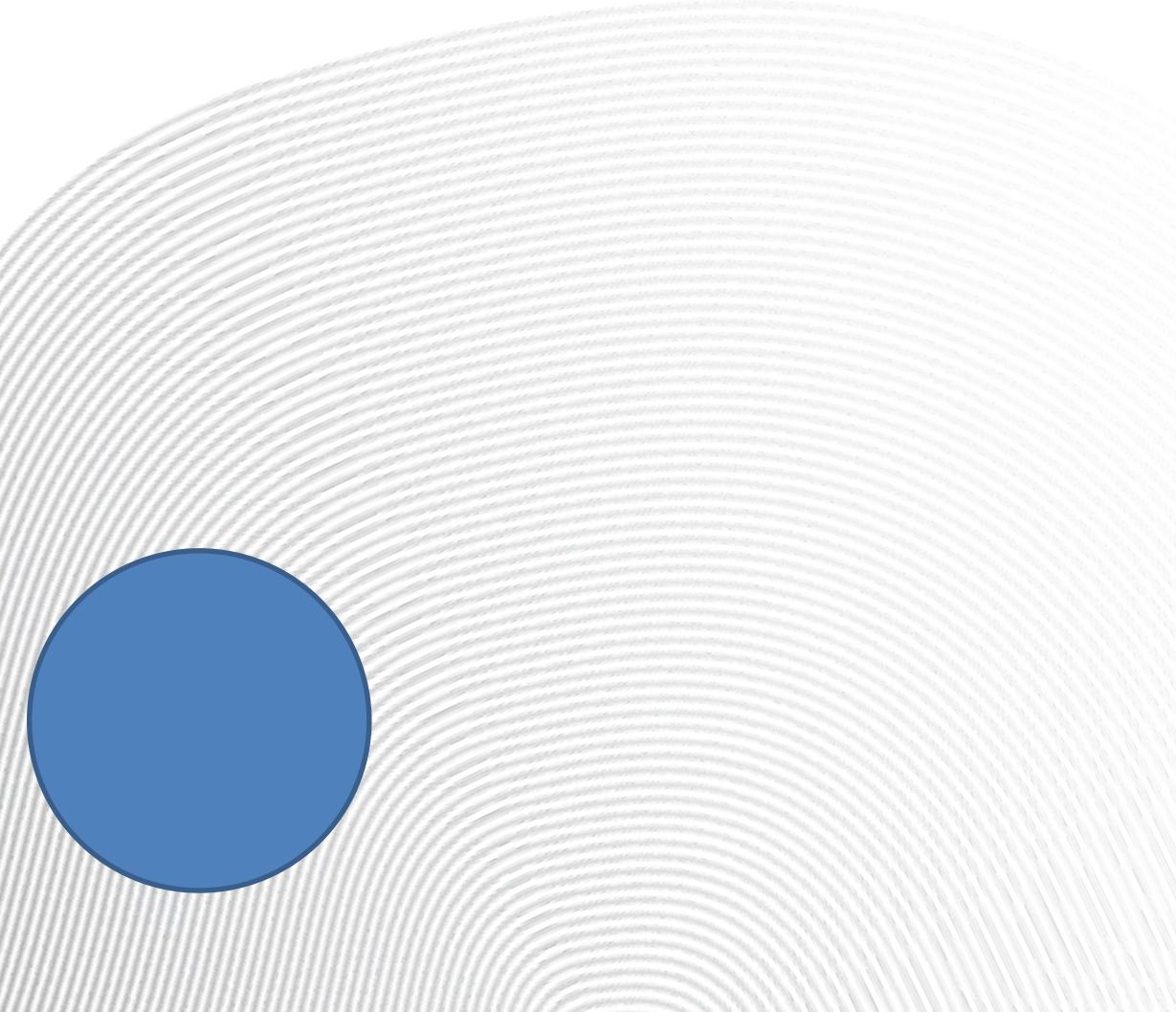
09

Algoritmos de Clustering

---

10

Algoritmos de reglas de asociación

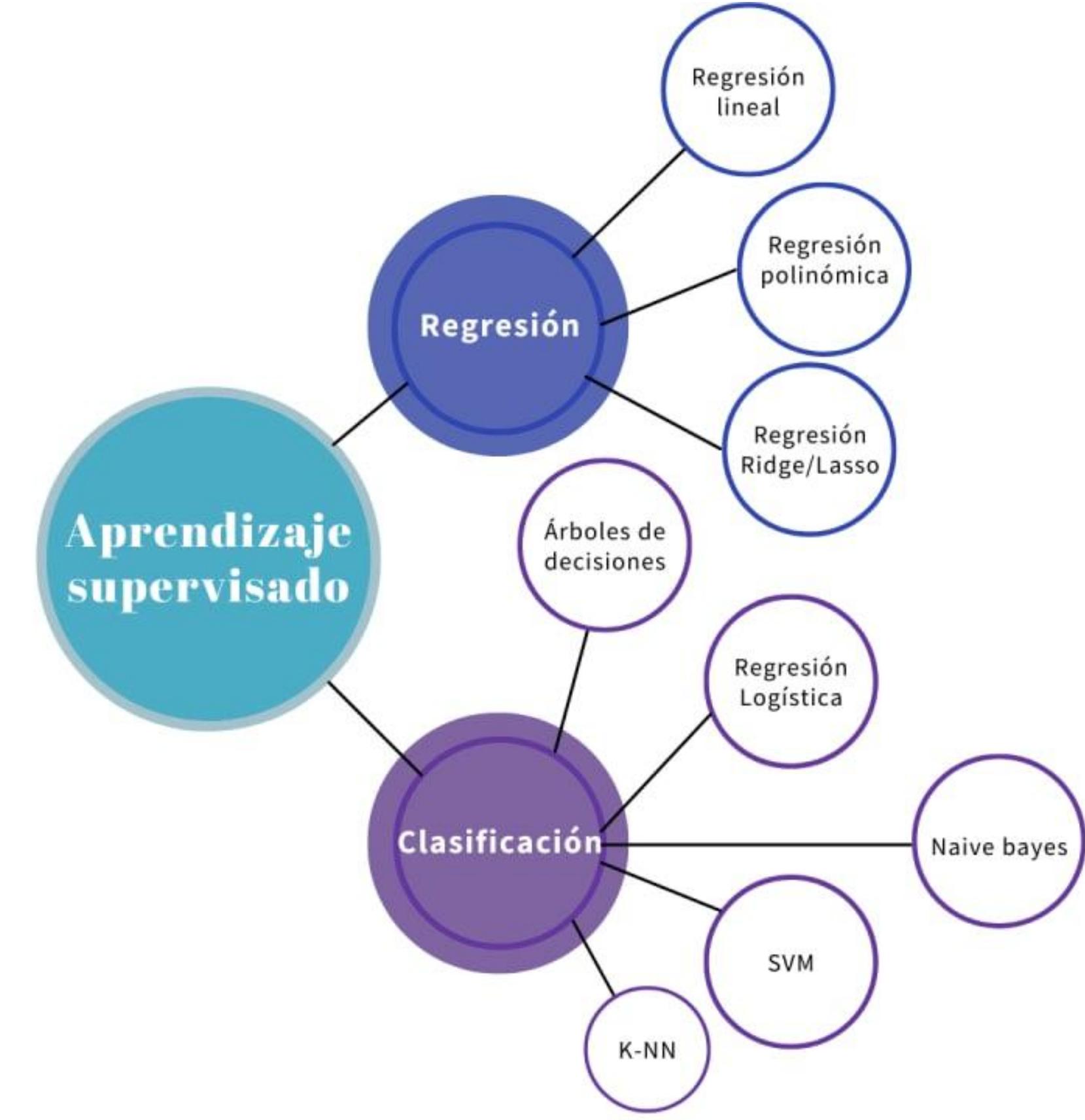
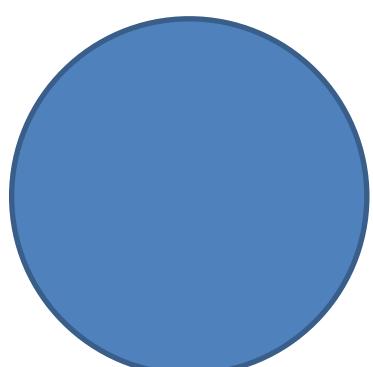




# 01

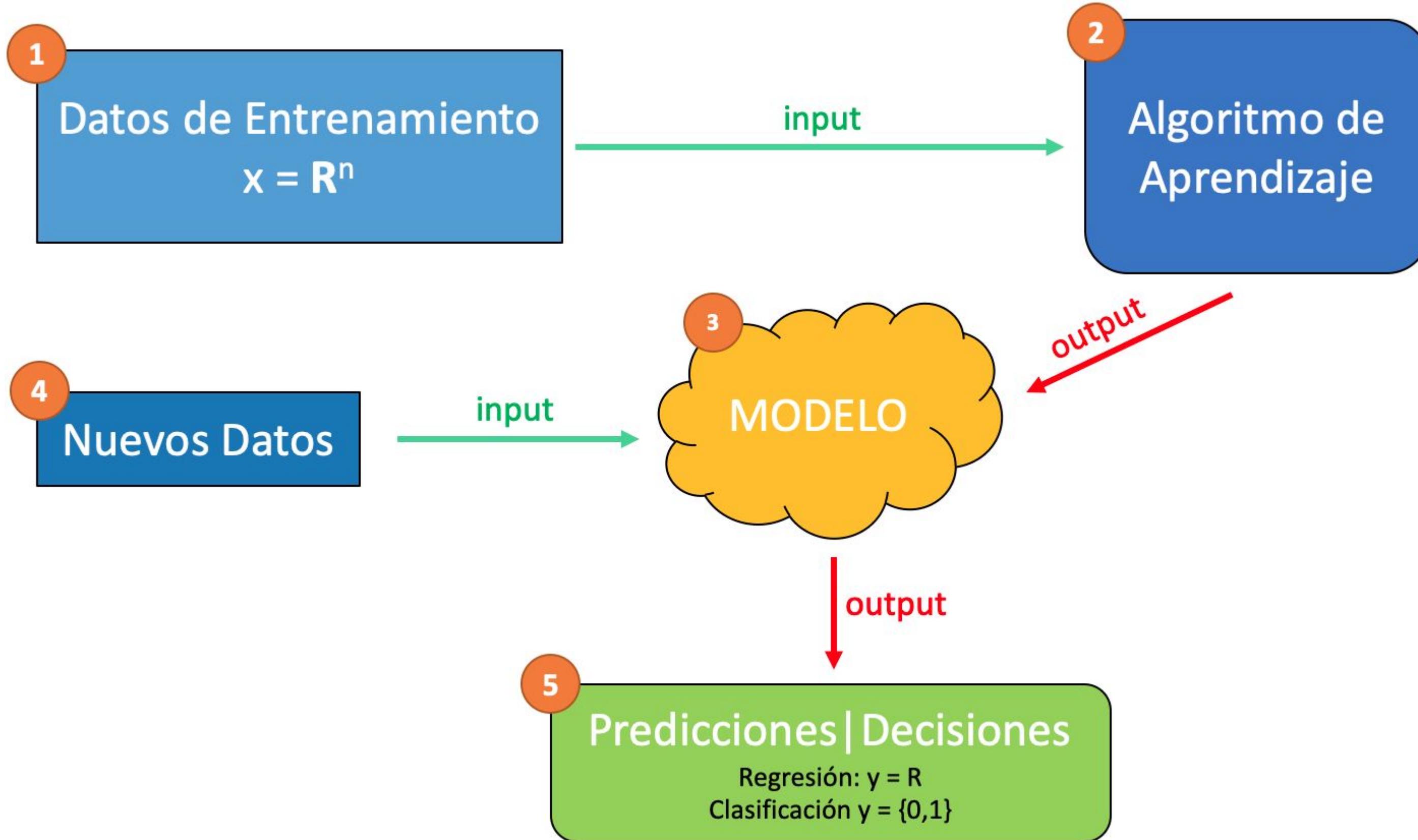
## Introducción al Aprendizaje Supervisado

Conceptos básicos del aprendizaje supervisado: datos etiquetados y su nomenclatura, hipótesis y coeficientes, función de pérdida, función de coste, regularización, early stopping, penalización y función objetivo.





# Esquema general





# Datos etiquetados: regresión

Diagram illustrating the regression model structure:

- y**: Price (price)
- X**: Features (year, manufacturer, condition, cylinders, fuel, odometer, title\_status, transmission, drive, type, paint\_color)
- X<sup>7</sup>**: Features (year, manufacturer, condition, cylinders, fuel, odometer, title\_status)
- X<sub>1,9</sub>**: Features (drive, type, paint\_color)
- y<sub>2</sub>**: Price (price)
- X<sub>4</sub>**: Features (year, manufacturer, condition, cylinders, fuel, odometer, title\_status)

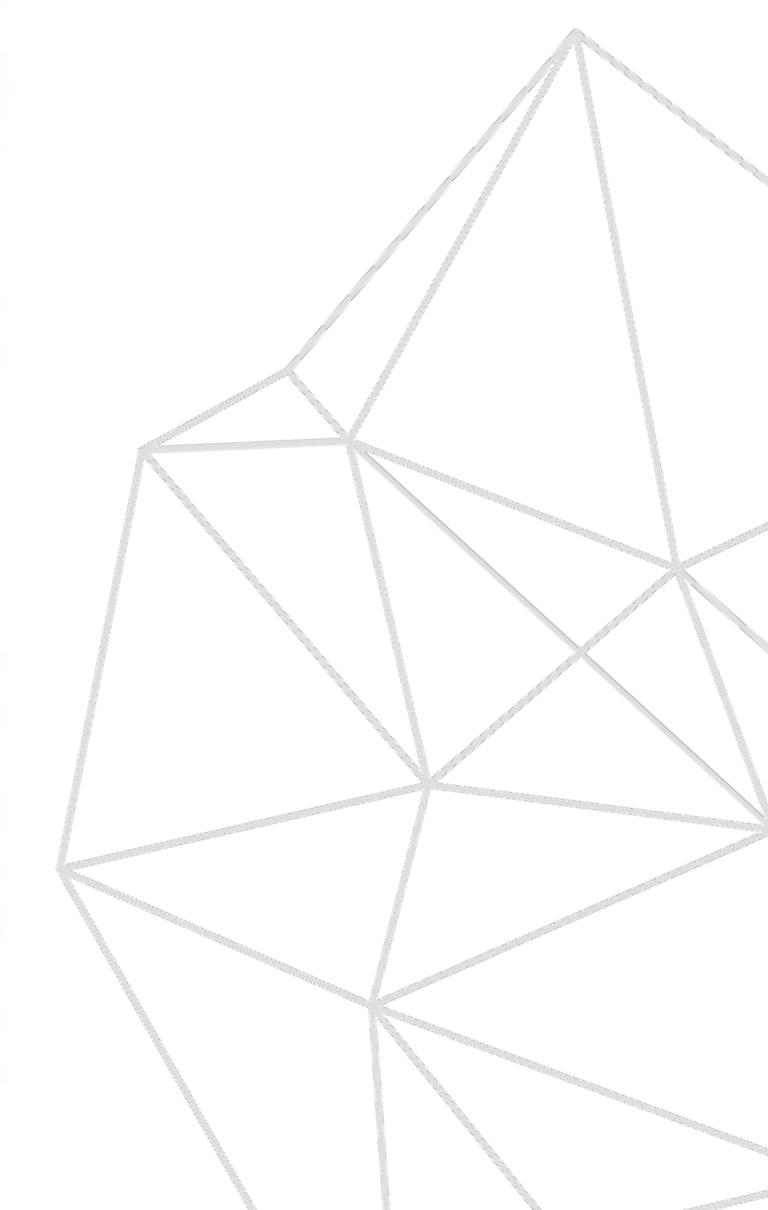
The diagram shows a regression tree structure where the final output  $y$  is derived from the features  $X$  through intermediate steps  $X^7$  and  $X_{1,9}$ . The features  $X_4$  are highlighted in red ovals.

Sample data table:

|        | price | year   | manufacturer | condition | cylinders   | fuel   | odometer | title_status | transmission | drive | type  | paint_color |
|--------|-------|--------|--------------|-----------|-------------|--------|----------|--------------|--------------|-------|-------|-------------|
| 0      | 6995  | 2000.0 | gmc          | excellent | 8 cylinders | gas    | 167783.0 | clean        | automatic    | 4wd   | Nan   | red         |
| 1      | 8750  | 2013.0 | hyundai      | excellent | 4 cylinders | gas    | 90821.0  | clean        | automatic    | fwd   | Nan   | grey        |
| 2      | 10900 | 2013.0 | toyota       | good      | 4 cylinders | hybrid | 92800.0  | clean        | automatic    | fwd   | Nan   | blue        |
| 3      | 12500 | 2003.0 | mitsubishi   | good      | 4 cylinders | gas    | Nan      | clean        | manual       | 4wd   | sedan | grey        |
| 4      | 16995 | 2007.0 | gmc          | good      | 8 cylinders | diesel | 254217.0 | clean        | automatic    | 4wd   | truck | white       |
| ...    | ...   | ...    | ...          | ...       | ...         | ...    | ...      | ...          | ...          | ...   | ...   | ...         |
| 423852 | 1600  | 2006.0 | hyundai      | fair      | 6 cylinders | gas    | 159980.0 | clean        | automatic    | fwd   | sedan | blue        |
| 423853 | 9000  | 2003.0 | toyota       | excellent | 8 cylinders | gas    | 160000.0 | clean        | automatic    | 4wd   | SUV   | green       |
| 423854 | 700   | 1994.0 | ford         | fair      | 6 cylinders | gas    | 212000.0 | clean        | manual       | rwd   | Nan   | green       |
| 423855 | 3800  | 1999.0 | lincoln      | excellent | 8 cylinders | gas    | 160000.0 | clean        | automatic    | rwd   | sedan | Nan         |
| 423856 | 8650  | 2015.0 | nissan       | Nan       | Nan         | gas    | 160526.0 | clean        | automatic    | fwd   | sedan | silver      |

# Datos etiquetados: clasificación

|     | X                 |                  |                   |                  | y         |
|-----|-------------------|------------------|-------------------|------------------|-----------|
|     | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target    |
| 0   | 5.1               | 3.5              | 1.4               | 0.2              | setosa    |
| 1   | 4.9               | 3.0              | 1.4               | 0.2              | setosa    |
| 2   | 4.7               | 3.2              | 1.3               | 0.2              | setosa    |
| 3   | 4.6               | 3.1              | 1.5               | 0.2              | setosa    |
| 4   | 5.0               | 3.6              | 1.4               | 0.2              | setosa    |
| ... | ...               | ...              | ...               | ...              | ...       |
| 145 | 6.7               | 3.0              | 5.2               | 2.3              | virginica |
| 146 | 6.3               | 2.5              | 5.0               | 1.9              | virginica |
| 147 | 6.5               | 3.0              | 5.2               | 2.0              | virginica |
| 148 | 6.2               | 3.4              | 5.4               | 2.3              | virginica |
| 149 | 5.9               | 3.0              | 5.1               | 1.8              | virginica |





# Algoritmo de aprendizaje: hipótesis y coeficientes

Los problemas de aprendizaje supervisado tienen como objetivo encontrar la relación entre  $X$  e  $y$ , es decir, obtener un modelo o hipótesis  $h$  que dado un  $X_i$  devuelva como resultado  $y_i$ . Esto se escribe como:

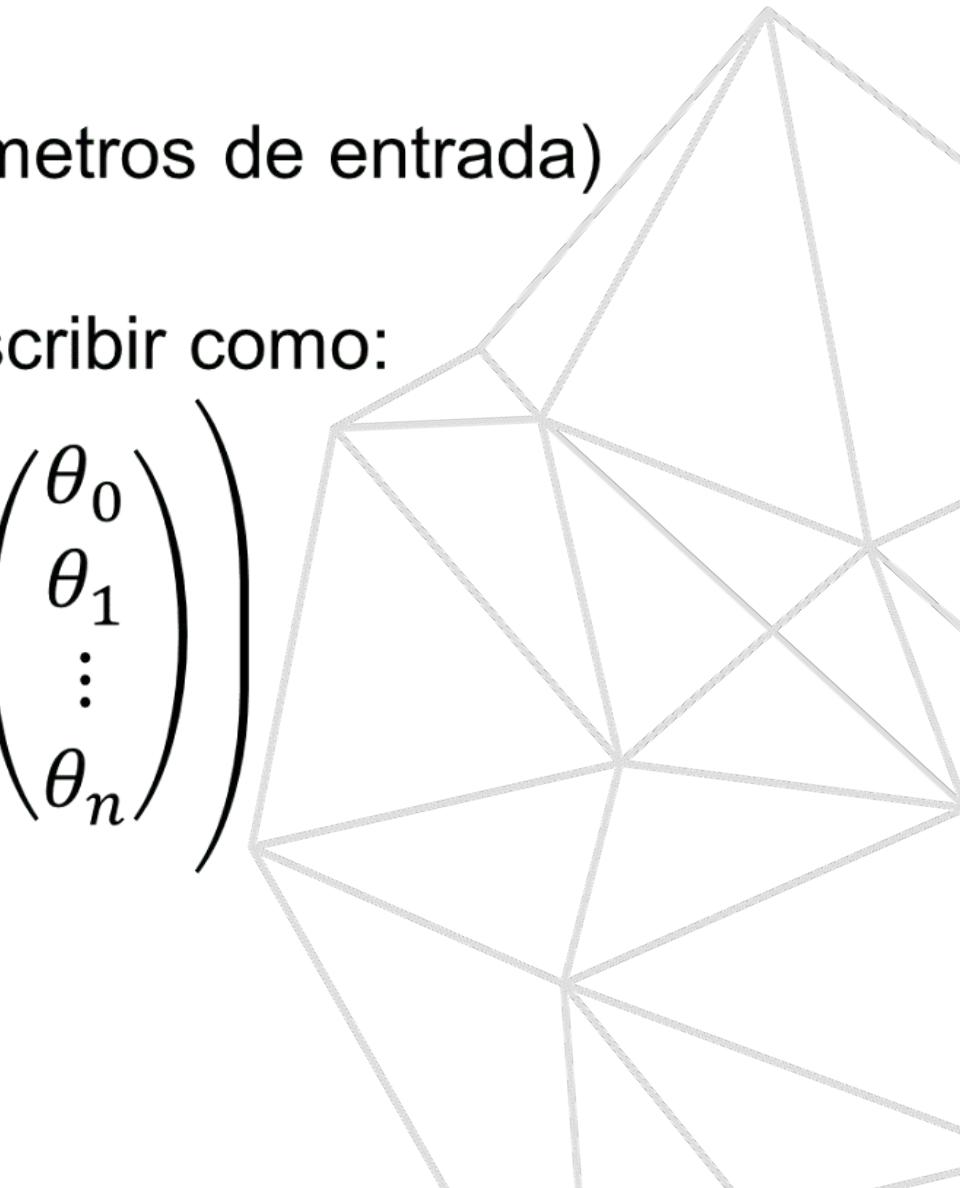
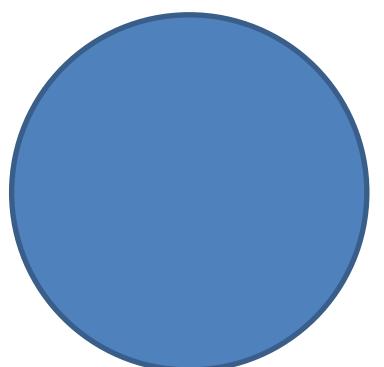
$$\hat{y}_i = h_{\theta}(X_i) \quad \hat{y} = h_{\theta}(X)$$

donde se tiene que:

- $X_i$  es una de las filas de los parámetros de entrada de nuestro dataset
- $h$  es el modelo o hipótesis
- $\theta$  son los coeficientes del modelo:  $\theta_0, \theta_1, \dots, \theta_n$  (suponiendo que hay  $n$  parámetros de entrada)
- $\hat{y}_i$  es la salida estimada del modelo dados los parámetros de entrada

Por lo general,  $\theta$  simplemente serán coeficientes de las variables pudiéndose escribir como:

$$\begin{aligned}\hat{y}_i &= h_{\theta}(X_i) = g(\bar{X}_i \theta) = g \left( \begin{pmatrix} 1 & x_{i,1} & x_{i,2} & \dots & x_{i,n} \end{pmatrix} \cdot \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \right) \\ &= g(\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2} + \dots + \theta_n x_{i,n})\end{aligned}$$





# Algoritmo de aprendizaje: función de pérdida y de coste

La función de pérdida (también denominada función de error) permite calcular el error cometido al realizar una predicción concreta con nuestro modelo. Algunos ejemplos son:

- $L_2(y_i, \hat{y}_i) = (\hat{y}_i - y_i)^2$
- $L_1(y_i, \hat{y}_i) = |\hat{y}_i - y_i|$
- $L_{CE}(y_i, \hat{y}_i) = -(y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i))$

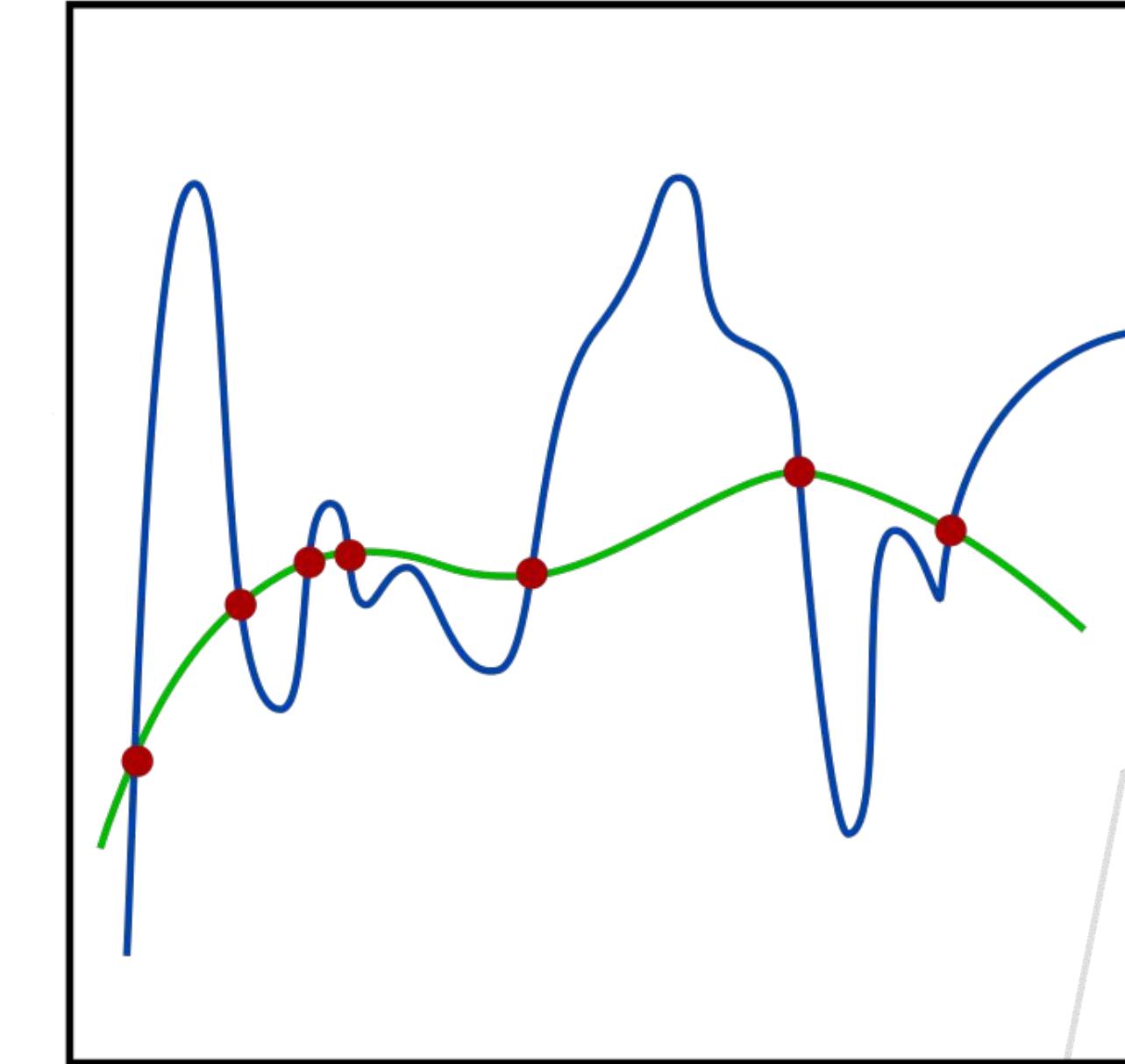
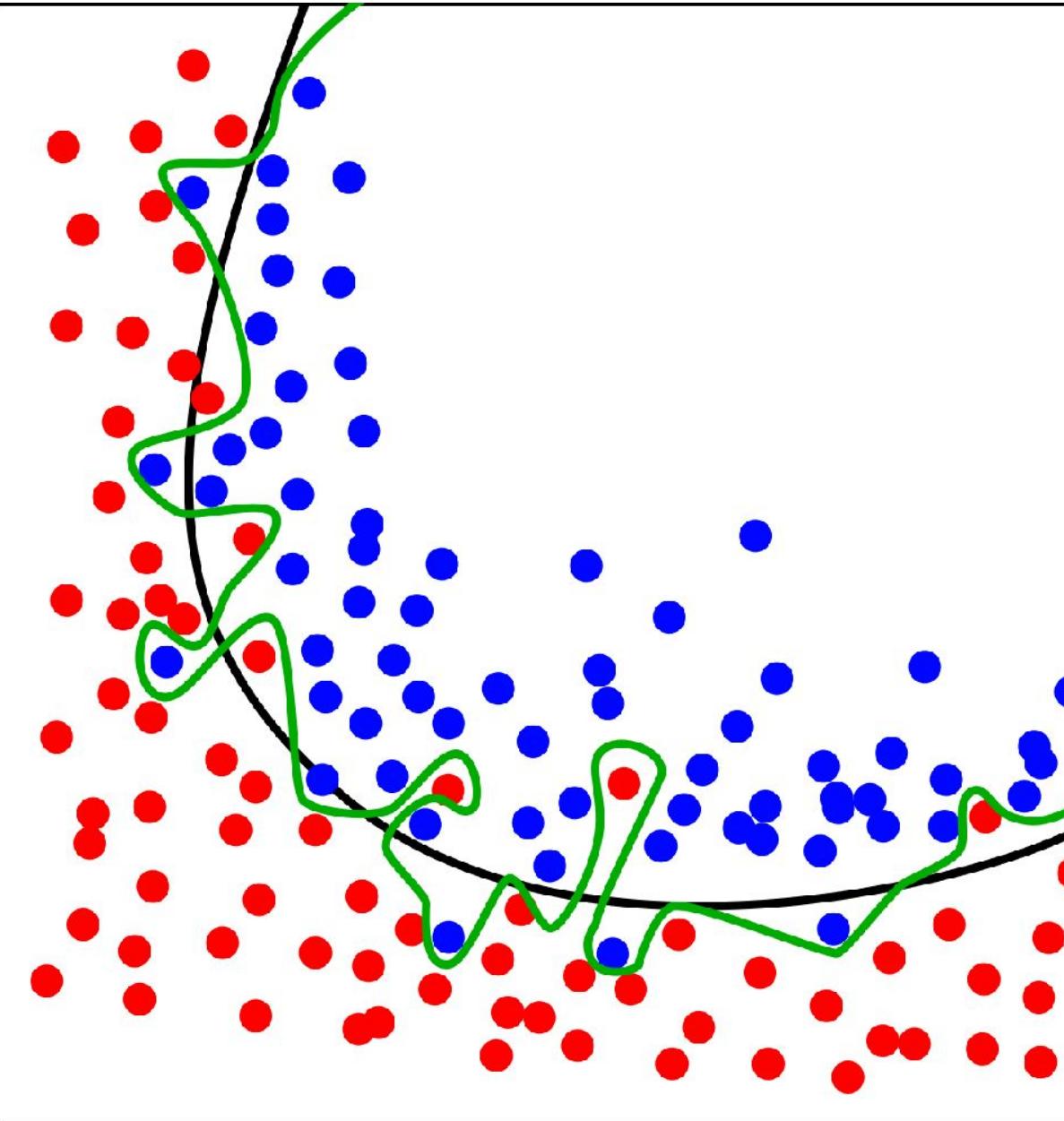
La función de coste calcula el error cometido por nuestro modelo sobre un conjunto de datos (tamaño  $m$ ). Utilizará por tanto la función de pérdida como ocurre en el siguiente ejemplo:

$$C(h_\theta, X, y) = \frac{1}{m} \sum_{i=1}^m L(y_i, h_\theta(X_i))$$

El objetivo del algoritmo de aprendizaje sería, por tanto, minimizar la función de coste. Pero esto puede producir sobreaprendizaje.



# Algoritmo de aprendizaje: sobreaprendizaje



Si se trata de minimizar completamente la función de coste **aumentará la varianza** (mayor número de errores en la clasificación de los datos de test o mayor diferencia con el valor de dichos datos).



# Algoritmo de aprendizaje: early stopping y regularización

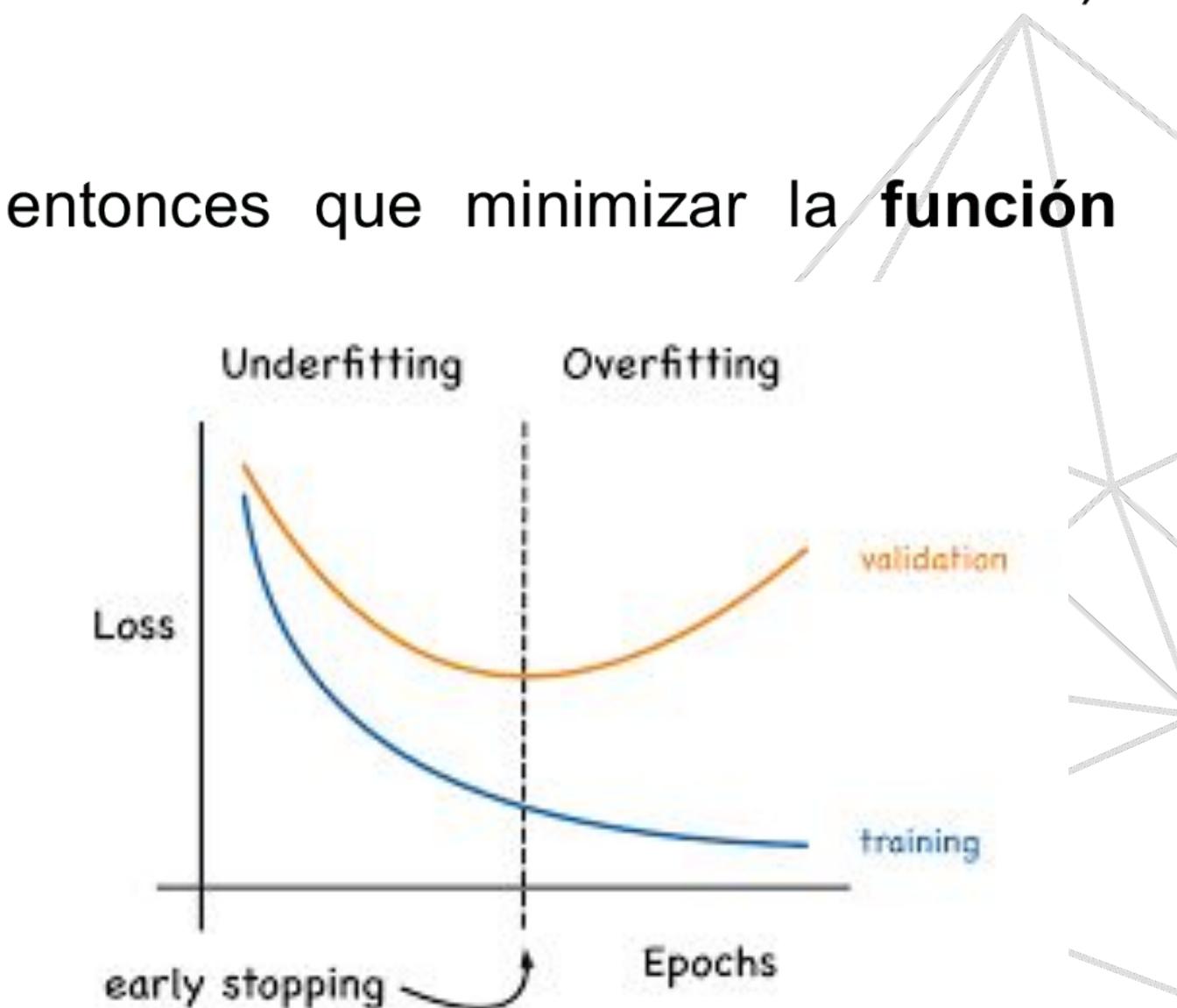
Ante el problema de sobreaprendizaje, se suelen utilizar dos técnicas:

- Parada temprana (*Early stopping*): parar el entrenamiento cuando se empieza a producir el sobreajuste evaluando con el conjunto de validación (se puede elegir con técnicas como *holdout* o *cross-validation*). Se considera una técnica de regularización implícita.
- Regularización: reduce la complejidad del modelo, teniendo entonces que minimizar la **función objetivo**

$$J(h_\theta, X, y) = C(h_\theta, X, y) + R_\lambda(h_\theta)$$

La función  $R_\lambda$  se caracteriza por el parámetro de regularización y la función de penalización:

$$R_\lambda(h_\theta) = \lambda p(h_\theta)$$





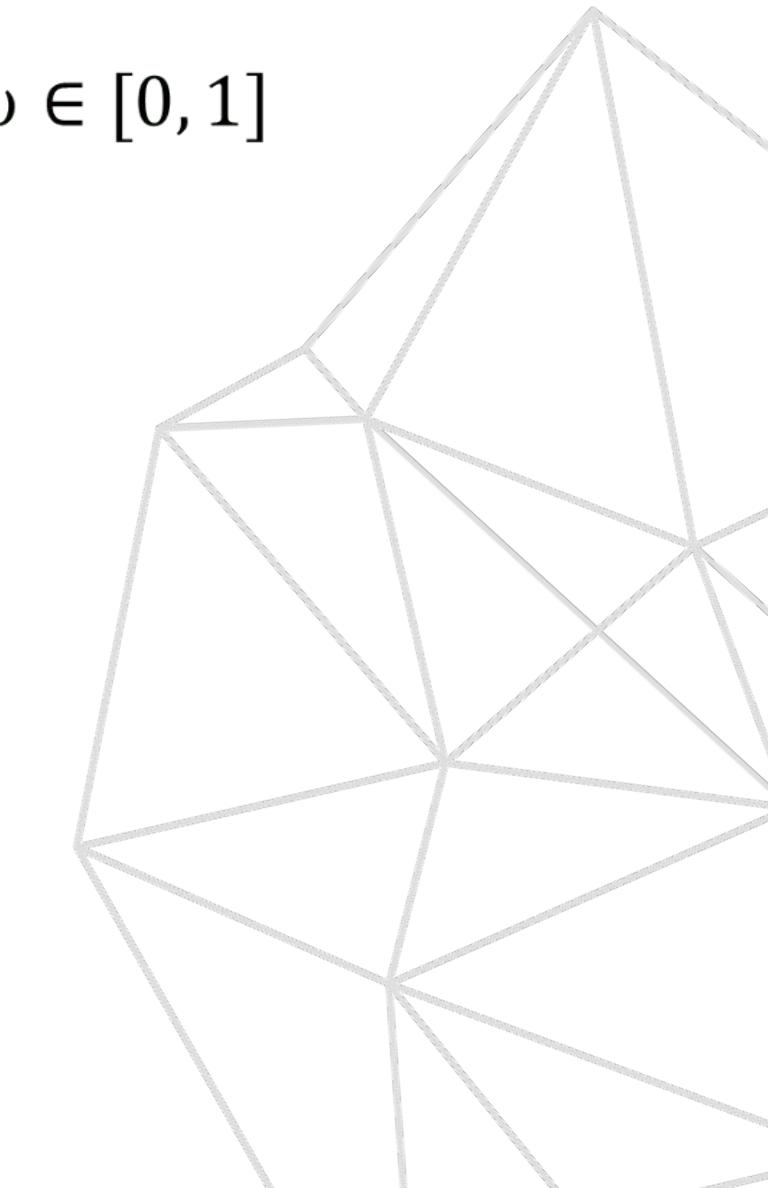
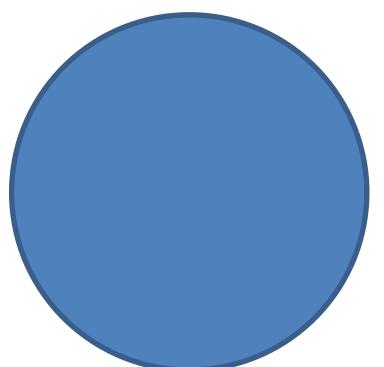
# Algoritmo de aprendizaje: penalización

Aunque pueden utilizarse muchas funciones de penalizaciones las más comunes son:

- Lasso o  $L_1$ :  $p(h_\theta) = L_1(h_\theta) = \sum_{i=1}^n |\theta_i|$
- Ridge o  $L_2$ :  $p(h_\theta) = L_2(h_\theta) = \sum_{i=1}^n (\theta_i)^2$
- ElasticNet:  $p(h_\theta) = \omega L_1(h_\theta) + (1 - \omega)L_2(h_\theta) = \omega \sum_{i=1}^n |\theta_i| + (1 - \omega) \sum_{i=1}^n (\theta_i)^2$  con  $\omega \in [0, 1]$

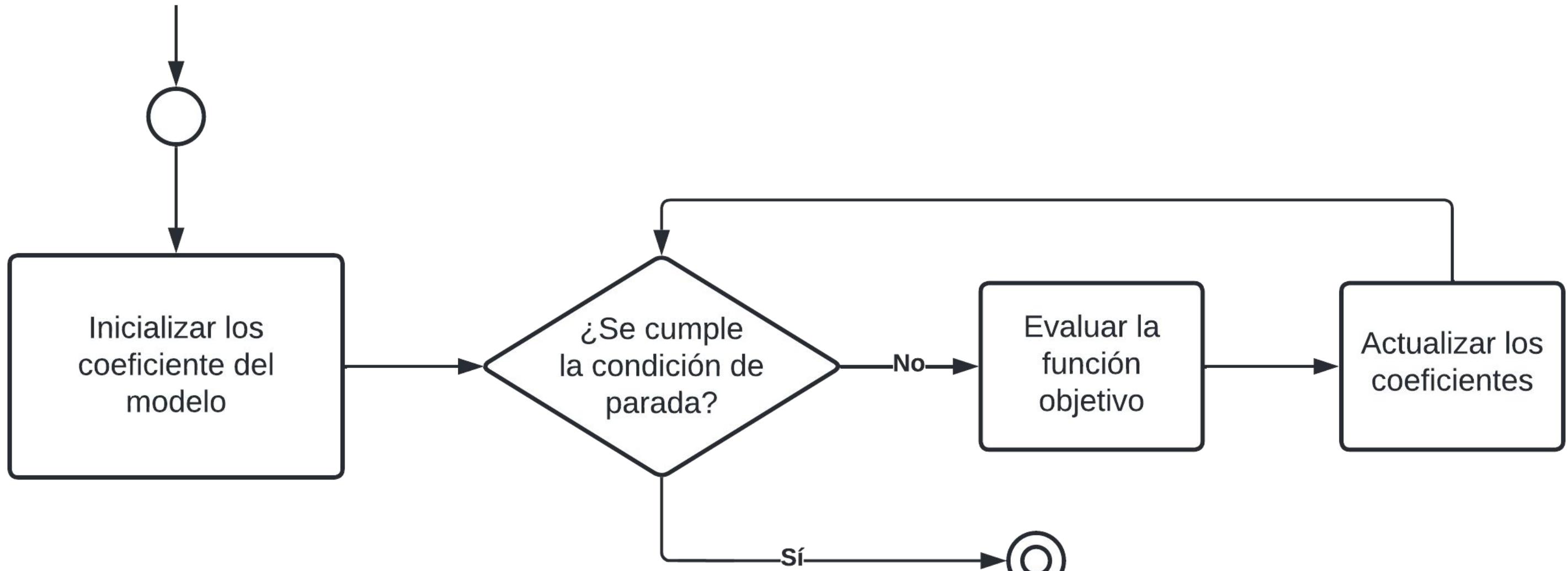
De esta forma, un ejemplo de función objetivo a optimizar sería:

$$J(h_\theta, X, y) = \underbrace{\frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2}_{C(h_\theta, X, y)} + \lambda \underbrace{\sum_{i=1}^n (\theta_i)^2}_{p(h_\theta)}$$



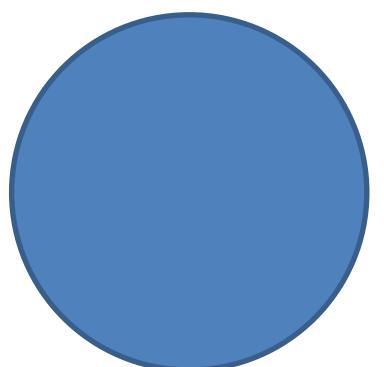
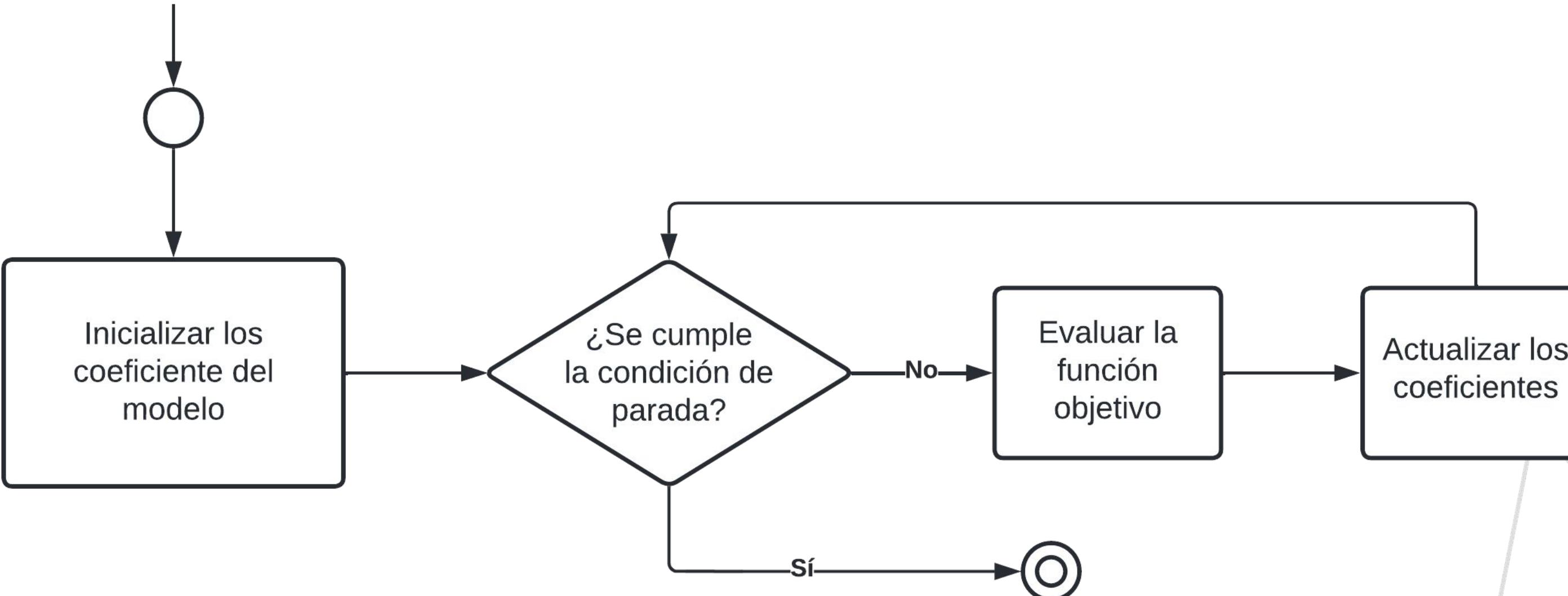
# 02

## Algoritmo general de optimización



# Algoritmo de aprendizaje: optimización

El algoritmo general de optimización de la función objetivo es el siguiente:





# Algoritmo de aprendizaje: actualización de coeficientes

Para actualizar los coeficientes  $\theta_0, \theta_1, \dots, \theta_n$  de nuestro modelo existen numerosas técnicas. Una de ellas es “El descenso de gradiente”, que se caracteriza por actualizar los coeficientes de la siguiente forma:

$$\theta'_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(h_\theta, X, y)$$

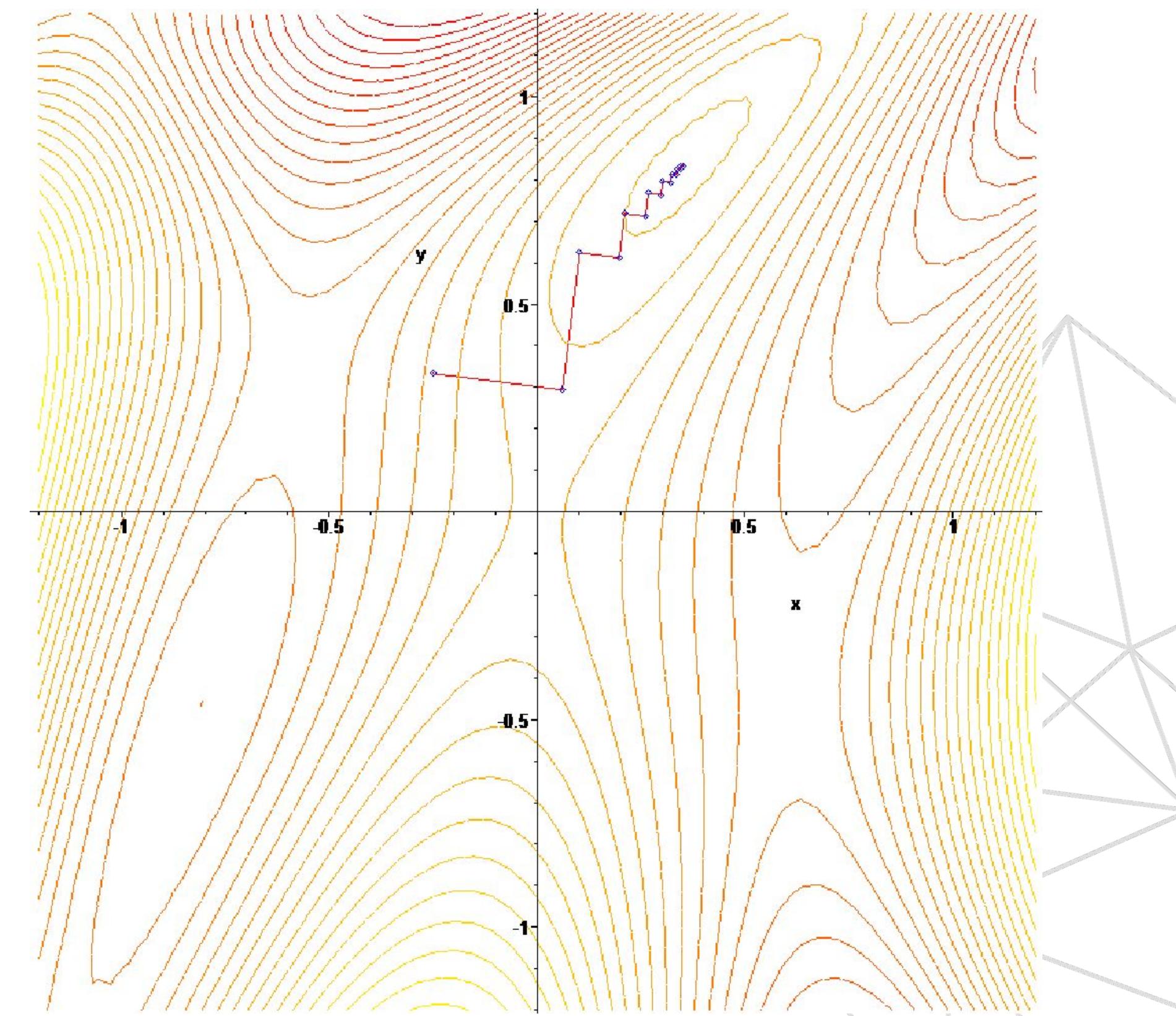
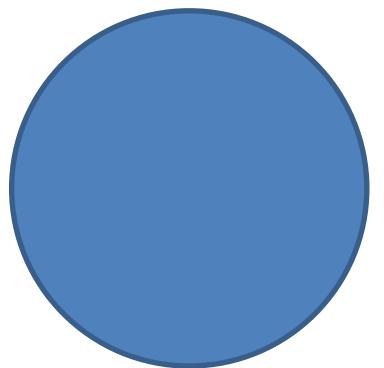
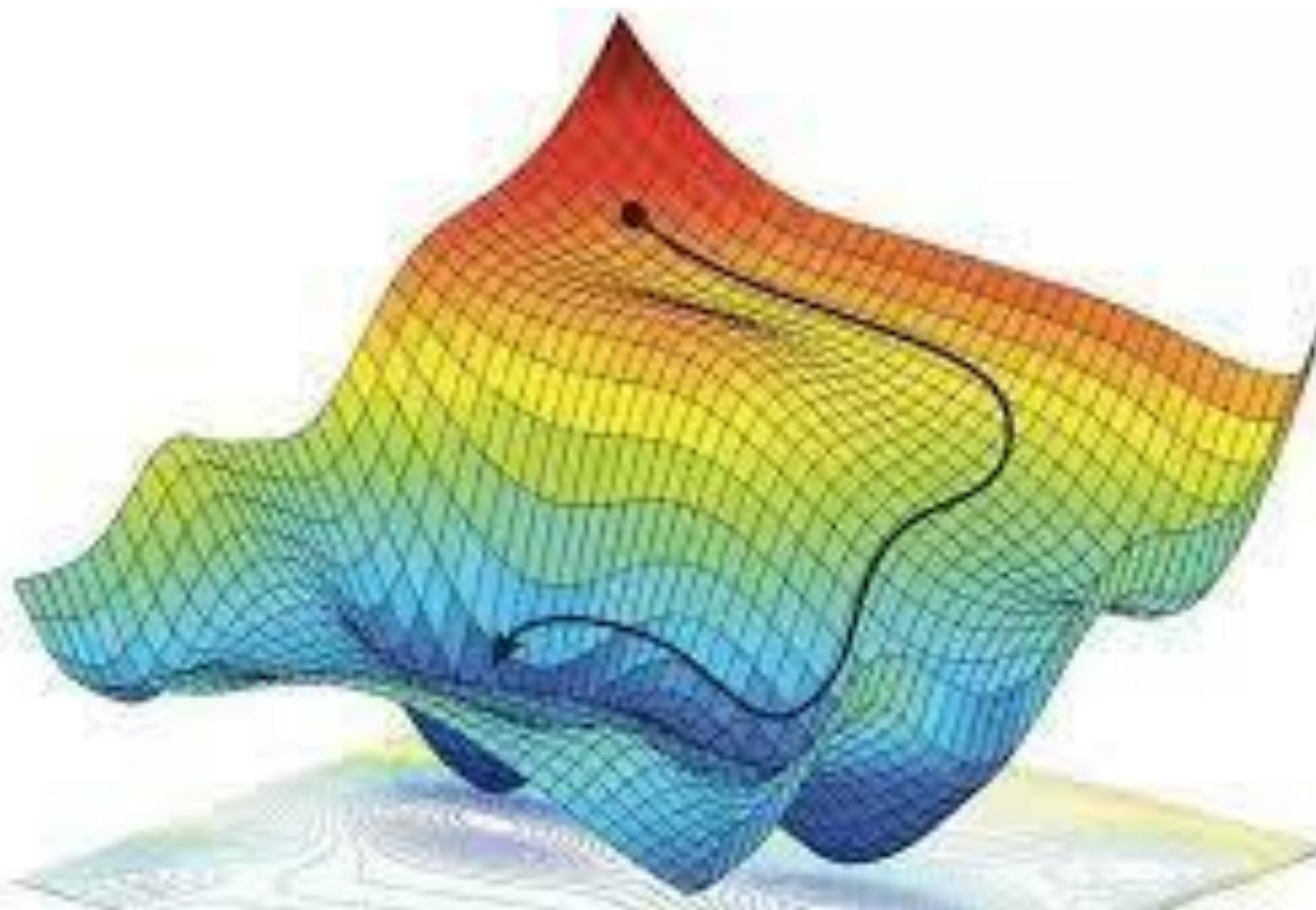
De manera que nuestra función objetivo sería  $J(h_{\theta'}, X, y)$  donde  $\theta'$  son los nuevos coeficientes  $\theta'_0, \theta'_1, \dots, \theta'_n$ . El parámetro  $\alpha$  es una constante de aprendizaje (o *Learning rate*) definida previamente, aunque existen técnicas (como el recocido simulado) que optimizan su valor para cada epoch.

Para elegir la constante de aprendizaje adecuadamente hay que tener en cuenta que:

- Para valores pequeños el avance hasta un máximo o mínimo puede ser muy lento
- Para valores grandes el máximo o mínimo puede no alcanzarse nunca por ser sobrepasado constantemente



# Algoritmo de aprendizaje: descenso del gradiente

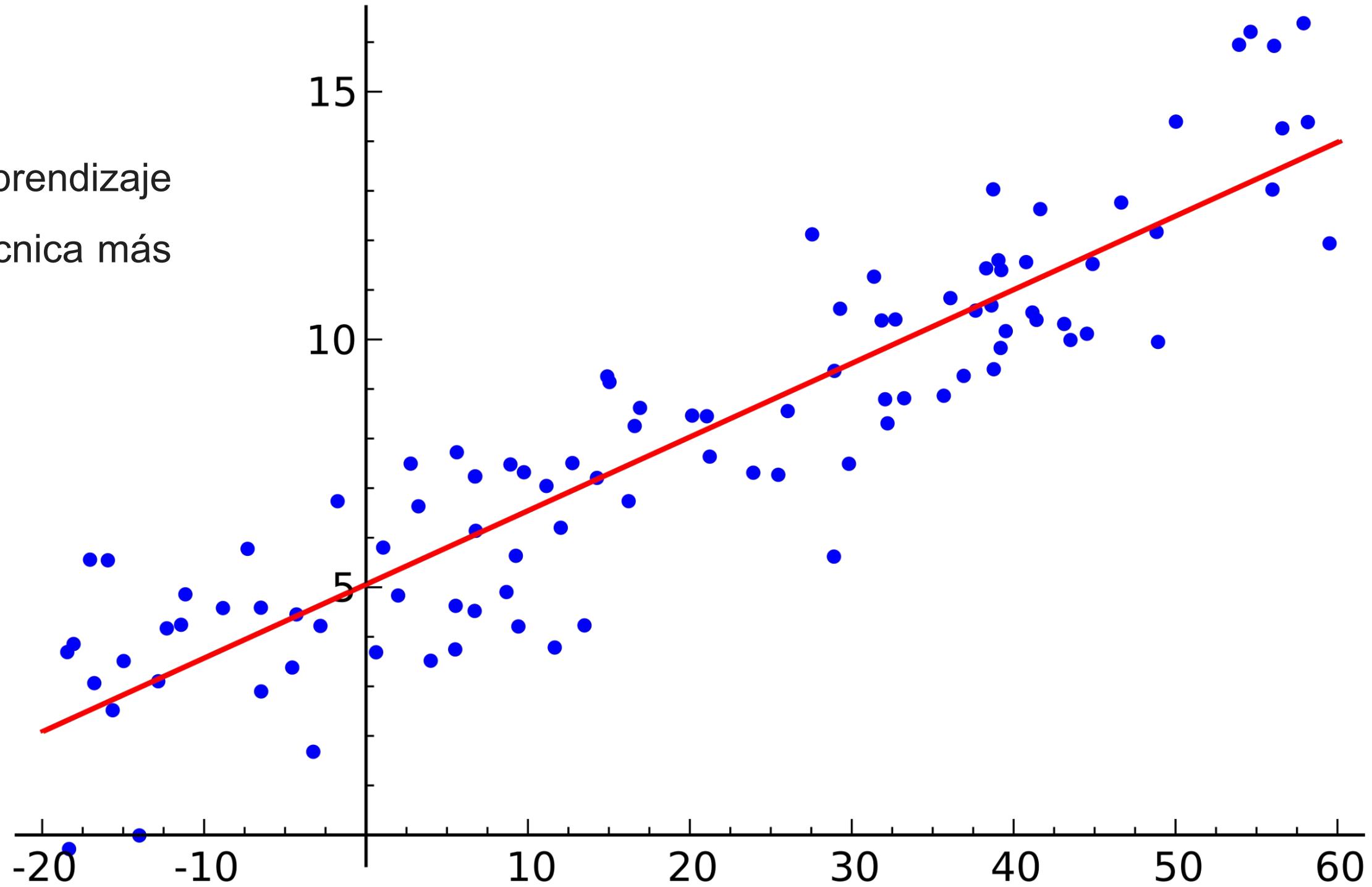
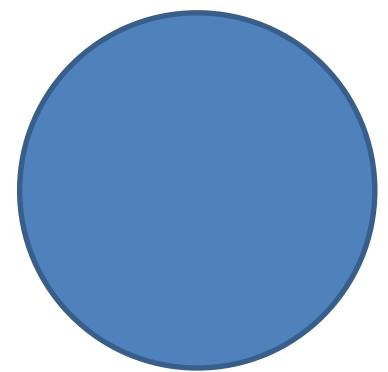




# 03

## La regresión lineal

Entenderemos todo lo estudiado sobre aprendizaje supervisado con este ejemplo utilizando la técnica más conocida para problemas de regresión.





# Modelo

Supongamos que contamos con **datos etiquetados** con  $n$  variables de entrada y la variable de salida y es continua (estamos ante un **problema de regresión**) entonces el modelo de regresión lineal vendrá dado por:

$$\hat{y}_i = h_{\theta}(X_i) = \theta_0 + \sum_{j=1}^n \theta_j x_{i,j} = \theta_0 + \theta_1 x_{i,1} + \dots + \theta_n x_{i,n}$$

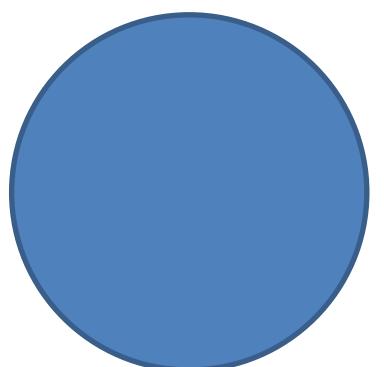
Por ejemplo, supongamos que queremos predecir el precio de un coche de segunda mano, a partir del kilometraje y su edad. Entonces el modelo será

$$\hat{y}_i = h_{\theta}(X_i) = h_{\theta}(x_{i,1}, x_{i,2}) = \theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}$$

Kilometraje del coche  $i$

Precio del coche  $i$

Edad del coche  $i$





# Función de pérdida, coste y objetivo

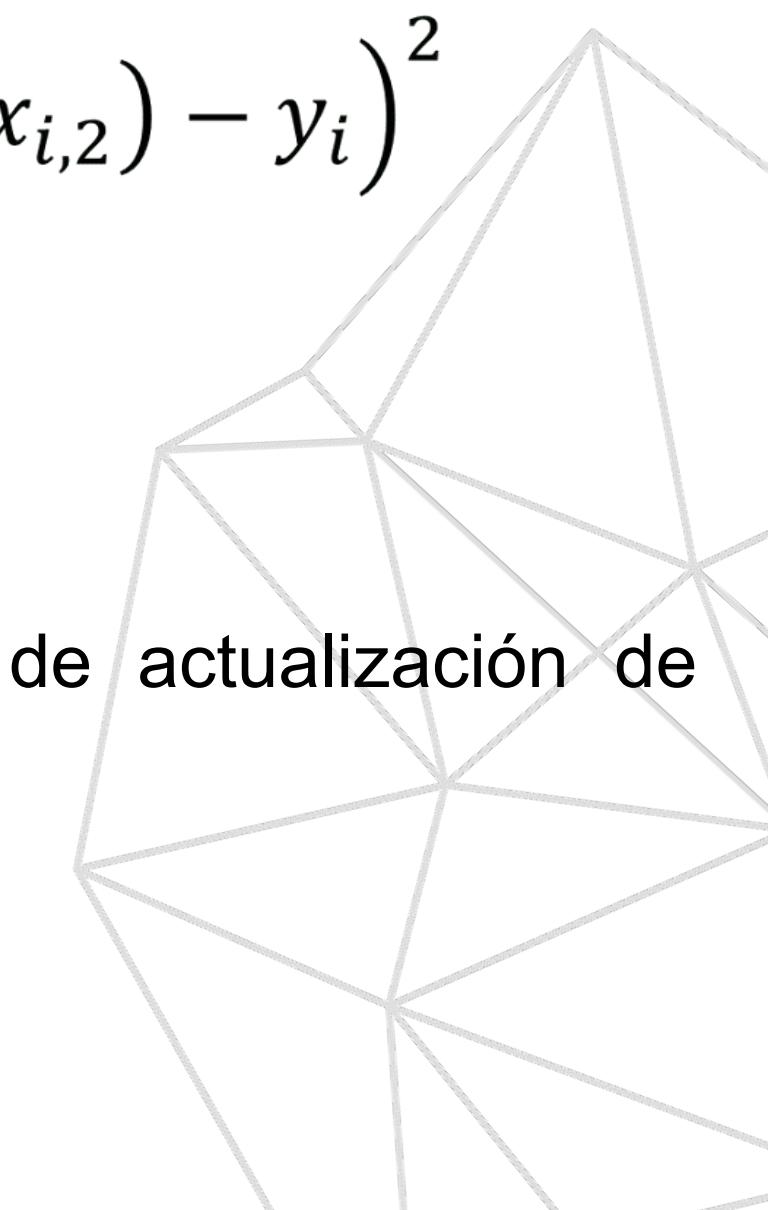
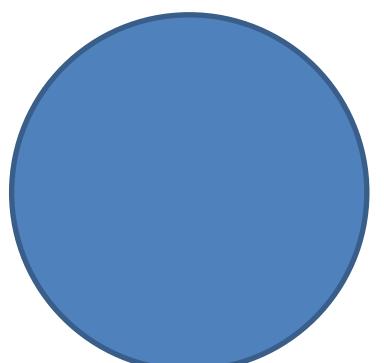
Siguiendo con nuestro ejemplo de predicción de precios, determinemos la función de pérdida, de coste y objetivo:

$$L(y_i, h_\theta(X_i)) = L_2(y_i, h_\theta(X_i)) = (h_\theta(X_i) - y_i)^2 = ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i)^2$$

$$C(h_\theta, X, y) = \frac{1}{2m} \sum_{i=1}^m L_2(y_i, h_\theta(X_i)) = \frac{1}{2m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i)^2$$

$$J(h_\theta, X, y) = C(h_\theta, X, y) + R_0(h_\theta) = C(h_\theta, X, y)$$

El objetivo será minimizar la función objetivo. Para ello, en el paso de actualización de parámetros utilizaremos el descenso del gradiente.





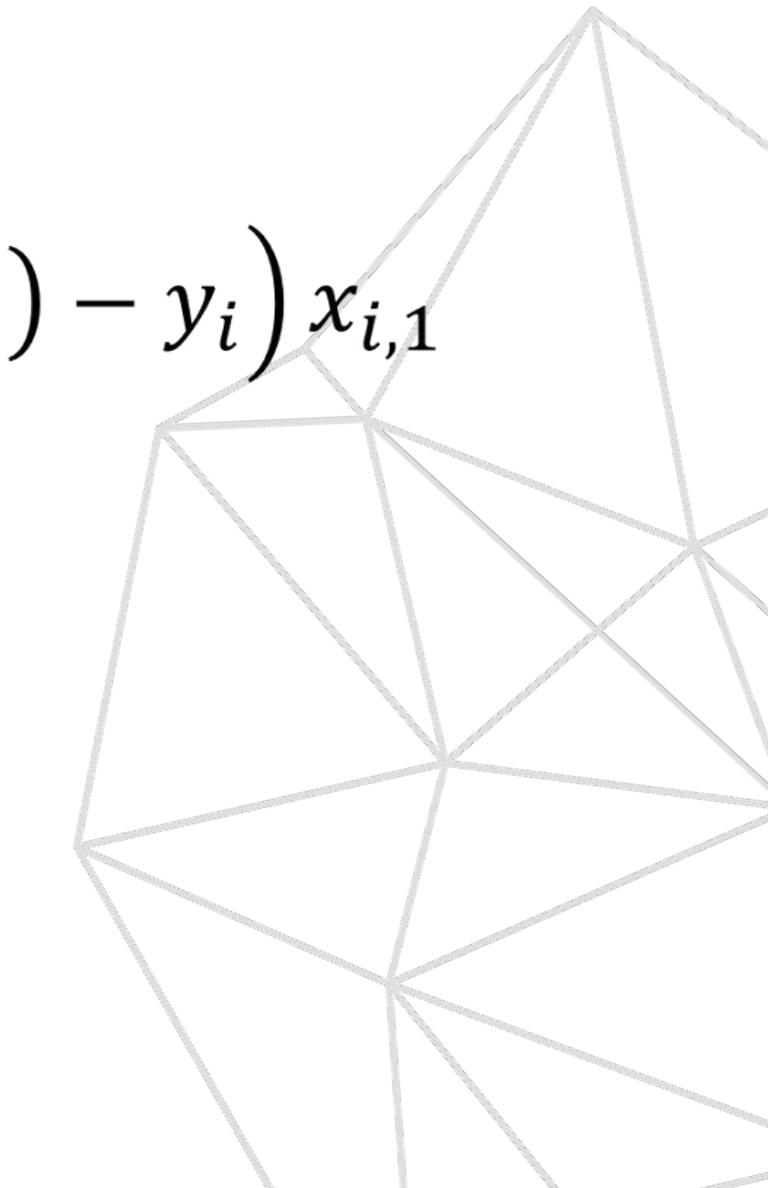
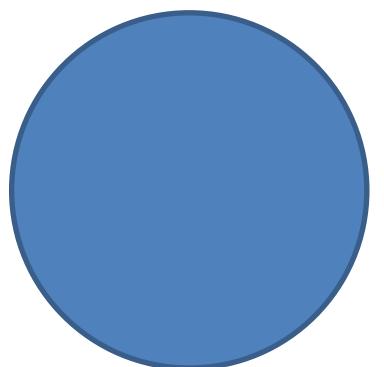
# Descenso del gradiente

Para actualizar los coeficientes por el método del descenso del gradiente, hay que calcular la derivada parcial de la función objetivo respecto de cada coeficiente, quedando entonces:

$$\theta'_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(h_\theta, X, y) = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i)$$

$$\theta'_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(h_\theta, X, y) = \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i) x_{i,1}$$

$$\theta'_2 = \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i) x_{i,2}$$





# Ejemplo de algoritmo: Inicialización y pérdida

|   | Precio (€) | Edad | Kilometraje (en decenas de miles) |
|---|------------|------|-----------------------------------|
| 0 | 22.050     | 0    | 0                                 |
| 1 | 18.205     | 2    | 1,7052                            |
| 2 | 13.840     | 3    | 5,3                               |
| 3 | 4.706      | 7    | 10,4258                           |
| 4 | 800        | 10   | 11                                |

Datos de entrenamiento

1. Utilizamos la función objetivo explicada anteriormente
2. Establecemos como condición de parada: la actualización de coeficientes es igual o inferior a 0,1.
3. Establecemos  $\alpha = 0,005$
4. Inicializamos los coeficientes  $\theta_0, \theta_1$  y  $\theta_2$  a 0 (elección arbitraria)

Comenzamos el primer epoch calculando la función de pérdida:

- $L_2(y_0, h_\theta(X_0)) = ((\theta_0 + \theta_1 x_{0,1} + \theta_2 x_{0,2}) - y_0)^2 = ((0 + 0 \cdot 0 + 0 \cdot 0) - 22050)^2 = 486202500$
- $L_2(y_1, \hat{y}_1) = ((\theta_0 + \theta_1 x_{1,1} + \theta_2 x_{1,2}) - y_1)^2 = ((0 + 0 \cdot 2 + 0 \cdot 1,7052) - 18205)^2 = 331422025$
- $L_2(y_2, \hat{y}_2) = ((\theta_0 + \theta_1 x_{2,1} + \theta_2 x_{2,2}) - y_2)^2 = ((0 + 0 \cdot 3 + 0 \cdot 5,3) - 13840)^2 = 191545600$
- $L_2(y_3, \hat{y}_3) = ((\theta_0 + \theta_1 x_{3,1} + \theta_2 x_{3,2}) - y_3)^2 = ((0 + 0 \cdot 7 + 0 \cdot 10,4258) - 4706)^2 = 22146436$
- $L_2(y_4, \hat{y}_4) = ((\theta_0 + \theta_1 x_{4,1} + \theta_2 x_{4,2}) - y_4)^2 = ((0 + 0 \cdot 10 + 0 \cdot 11) - 800)^2 = 640000$



# Ejemplo de algoritmo: función de coste y objetivo

|   | Precio (€) | Edad | Kilometraje |
|---|------------|------|-------------|
| 0 | 22.050     | 0    | 0           |
| 1 | 18.205     | 2    | 1.7052      |
| 2 | 13.840     | 3    | 5,3         |
| 3 | 4.706      | 7    | 10.4258     |
| 4 | 800        | 10   | 11          |

| Epoch | 0 | 0 | 0 | 0 | 103195656,1 |
|-------|---|---|---|---|-------------|
| 0     |   |   |   |   |             |
| 1     |   |   |   |   |             |
| 2     |   |   |   |   |             |
| 3     |   |   |   |   |             |
| 4     |   |   |   |   |             |

Con los errores calculamos la función de coste:

- $L_2(y_0, \hat{y}_0) = 486202500$
- $L_2(y_1, \hat{y}_1) = 331422025$
- $L_2(y_2, \hat{y}_2) = 191545600$
- $L_2(y_3, \hat{y}_3) = 22146436$
- $L_2(y_4, \hat{y}_4) = 640000$

$$C(h_\theta, X, y) = \frac{1}{2m} \sum_{i=1}^m L_2(y_i, h_\theta(X_i))$$



Como tenemos 5 datos ( $m = 5$ ) por lo que:

$$C(h_\theta, X, y) = \frac{1}{2 \cdot 5} \cdot 1031956561 = 103195656,1$$



# Ejemplo de algoritmo: nuevos coeficientes

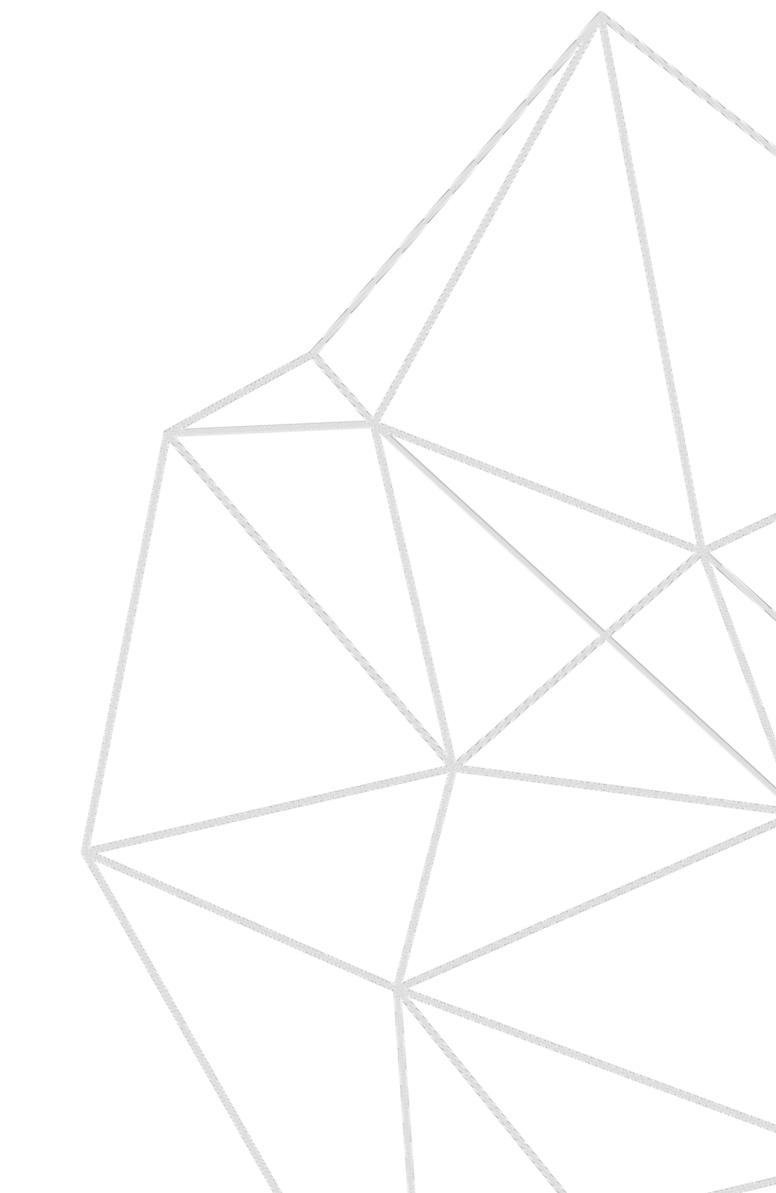
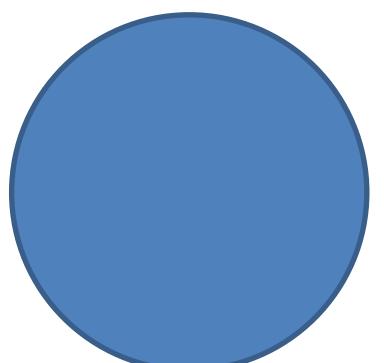
|   | Precio (€) | Edad | Kilometraje |
|---|------------|------|-------------|
| 0 | 22.050     | 0    | 0           |
| 1 | 18.205     | 2    | 1.7052      |
| 2 | 13.840     | 3    | 5,3         |
| 3 | 4.706      | 7    | 10.4258     |
| 4 | 800        | 10   | 11          |

| Epoch | 0 | 0      | 0 | 0 | 103195656,1 |
|-------|---|--------|---|---|-------------|
|       | 1 | 59,601 |   |   |             |

A continuación calculamos los nuevos valores de los coeficientes.

- $\theta'_0 = 0 - 0,005 \frac{1}{5} (-22050 - 18205 - 13840 - 4706 - 800) = 59,601$

$$\theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i)$$





# Ejemplo de algoritmo: nuevos coeficientes

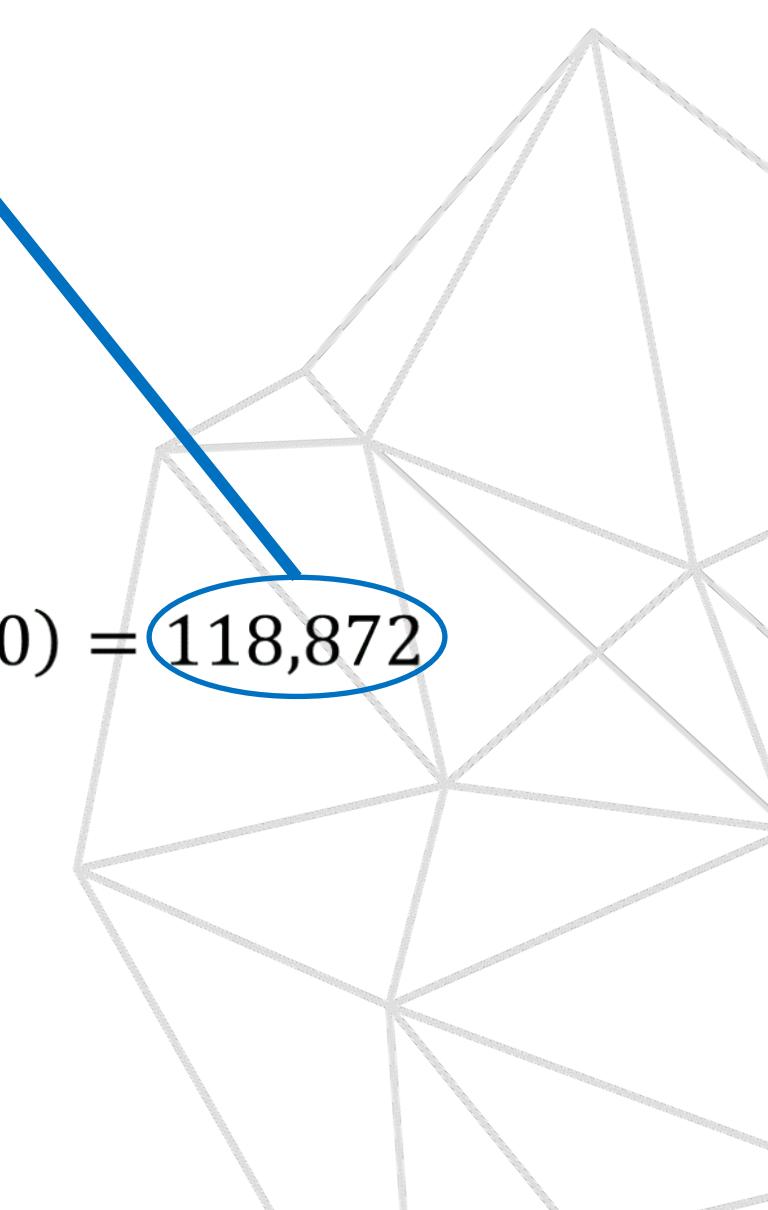
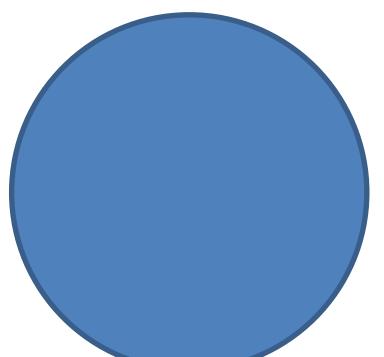
|   | Precio (€) | Edad | Kilometraje |
|---|------------|------|-------------|
| 0 | 22.050     | 0    | 0           |
| 1 | 18.205     | 2    | 1.7052      |
| 2 | 13.840     | 3    | 5,3         |
| 3 | 4.706      | 7    | 10.4258     |
| 4 | 800        | 10   | 11          |

| Epoch | 0 | 0      | 0       | 0 | 103195656,1 |
|-------|---|--------|---------|---|-------------|
|       | 0 | 59,601 | 118,872 |   |             |

A continuación calculamos los nuevos valores de los coeficientes:

- $\theta'_0 = 0 - 0,005 \frac{1}{5} (-22050 - 18205 - 13840 - 4706 - 800) = 59,601$
- $\theta'_1 = 0 - 0,005 \frac{1}{5} (-22050 \cdot 0 - 18205 \cdot 2 - 13840 \cdot 3 - 4706 \cdot 7 - 800 \cdot 10) = 118,872$

$$\theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i) x_{i,1}$$





# Ejemplo de algoritmo: nuevos coeficientes

|   | Precio (€) | Edad | Kilometraje |
|---|------------|------|-------------|
| 0 | 22.050     | 0    | 0           |
| 1 | 18.205     | 2    | 1.7052      |
| 2 | 13.840     | 3    | 5,3         |
| 3 | 4.706      | 7    | 10.4258     |
| 4 | 800        | 10   | 11          |

| Epoch | 0      | 0       | 0           | 0 | 103195656,1 |
|-------|--------|---------|-------------|---|-------------|
| 1     | 59,601 | 118,872 | 162,2589808 |   |             |

A continuación calculamos los nuevos valores de los coeficientes:

- $\theta'_0 = 0 - 0,005 \frac{1}{5} (-22050 - 18205 - 13840 - 4706 - 800) = 59,601$

- $\theta'_1 = 0 - 0,005 \frac{1}{5} (-22050 \cdot 0 - 18205 \cdot 2 - 13840 \cdot 3 - 4706 \cdot 7 - 800 \cdot 10) = 118,872$

- $\theta'_2 = 0 - 0,005 \frac{1}{5} (-22050 \cdot x_{0,2} - 18205 \cdot x_{1,2} - 13840 \cdot x_{2,2} - 4706 \cdot x_{3,2} - 800 \cdot x_{4,2})$

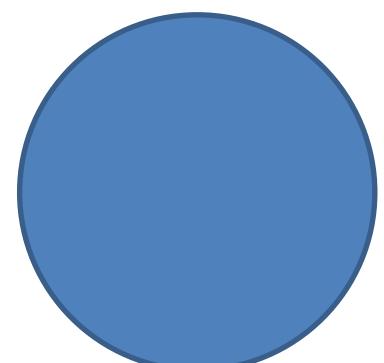
0

1,7052

5,3

10.4258

11



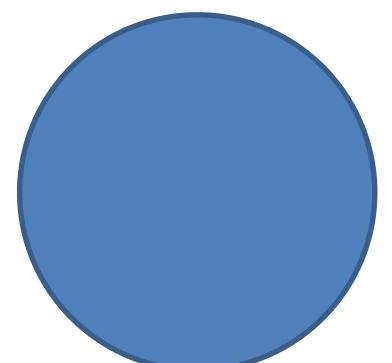


# Ejemplo de algoritmo: repetir

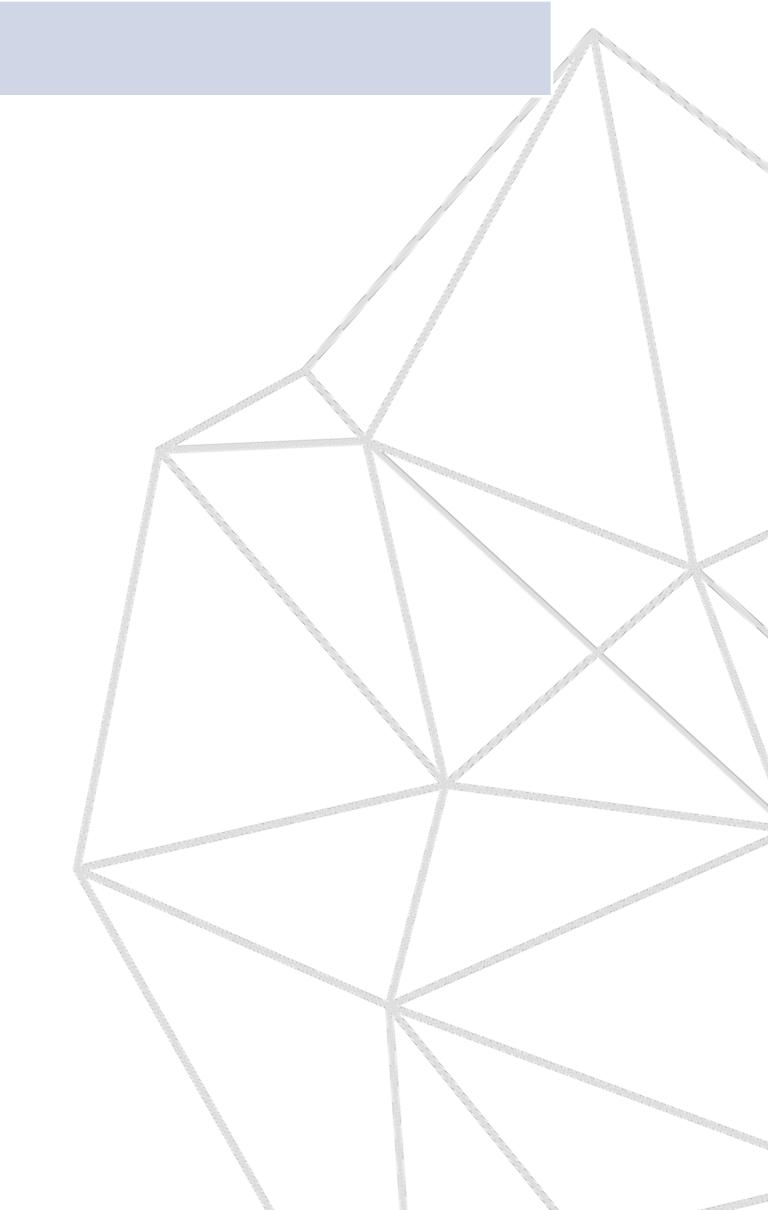
|   | Precio (€) | Edad | Kilometraje |  |  |  |
|---|------------|------|-------------|--|--|--|
| 0 | 22.050     | 0    | 0           |  |  |  |
| 1 | 18.205     | 2    | 1,7052      |  |  |  |
| 2 | 13.840     | 3    | 5,3         |  |  |  |
| 3 | 4.706      | 7    | 10,4258     |  |  |  |
| 4 | 800        | 10   | 11          |  |  |  |

En cada iteración repetimos los pasos que hemos llevado a cabo:

1. Calculamos la función de pérdida
2. Calculamos la función de coste
3. Calculamos los nuevos coeficientes



| Epoch | 0      | 0       | 0           | 0 | 103195656,1 |
|-------|--------|---------|-------------|---|-------------|
| 1     | 59,601 | 118,872 | 162,2589808 |   |             |
| 2     |        |         |             |   |             |





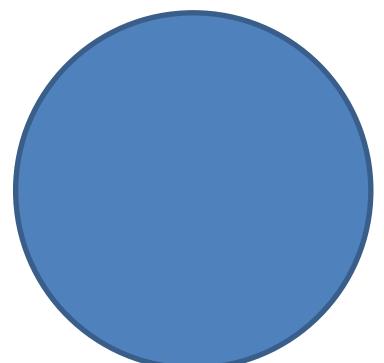
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |  |  |  |
|---|------------|------|-------------|--|--|--|
| 0 | 22.050     | 0    | 0           |  |  |  |
| 1 | 18.205     | 2    | 1.7052      |  |  |  |
| 2 | 13.840     | 3    | 5,3         |  |  |  |
| 3 | 4.706      | 7    | 10,4258     |  |  |  |
| 4 | 800        | 10   | 11          |  |  |  |

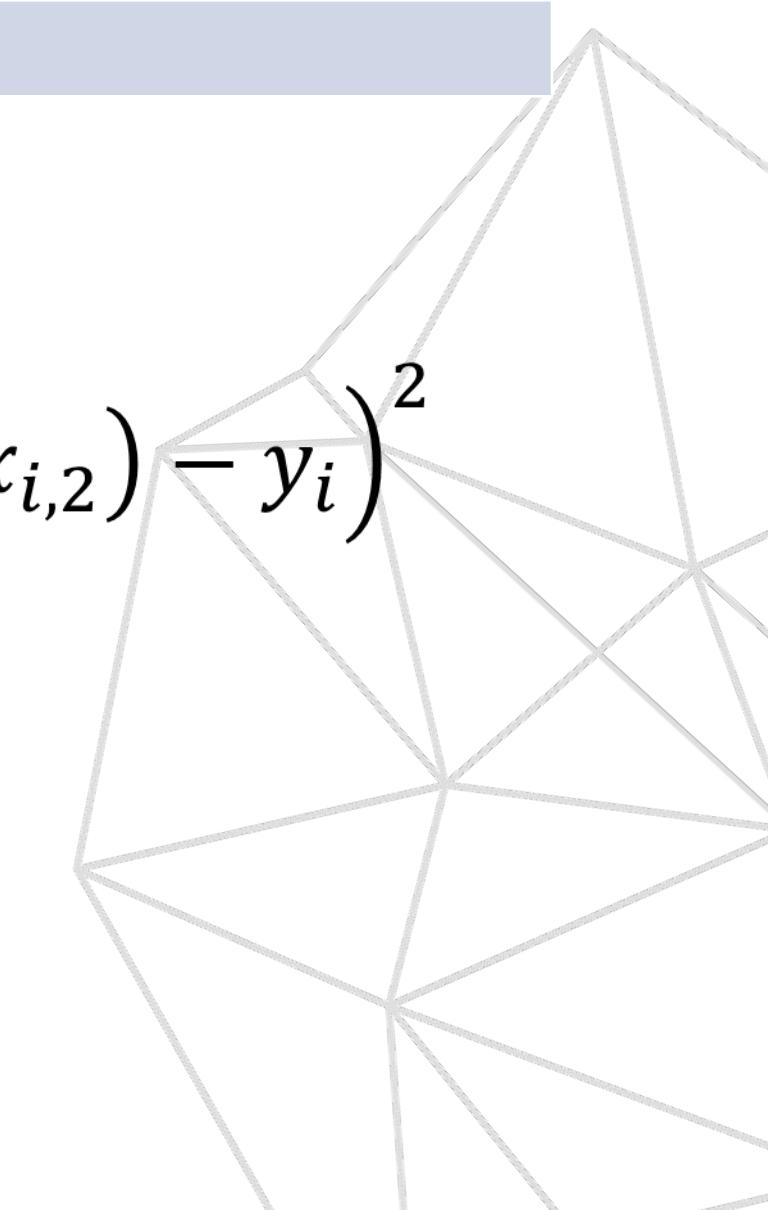
En cada iteración repetimos los pasos que hemos llevado a cabo:

- 1. Calculamos la función de pérdida**
2. Calculamos la función de coste
3. Calculamos los nuevos coeficientes

$$L_2(y_i, h_\theta(X_i)) = \left( (\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i \right)^2$$



| Epoch | 0      | 0       | 0           | 0 | 103195656,1 |
|-------|--------|---------|-------------|---|-------------|
| 1     | 59,601 | 118,872 | 162,2589808 |   |             |
| 2     |        |         |             |   |             |





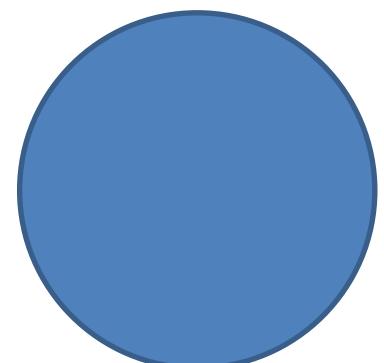
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |  |  |
|---|------------|------|-------------|-------------|--|--|
| 0 | 22.050     | 0    | 0           | 59,601      |  |  |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 |  |  |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 |  |  |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 |  |  |
| 4 | 800        | 10   | 11          | 3033,169789 |  |  |

En cada iteración repetimos los pasos que hemos llevado a cabo:

- 1. Calculamos la función de pérdida**
2. Calculamos la función de coste
3. Calculamos los nuevos coeficientes

$$L_2(y_i, h_\theta(X_i)) = \left( (\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i \right)^2$$



| Epoch |        |         |             |             |
|-------|--------|---------|-------------|-------------|
| 0     | 0      | 0       | 0           | 103195656,1 |
| 1     | 59,601 | 118,872 | 162,2589808 |             |
| 2     |        |         |             |             |



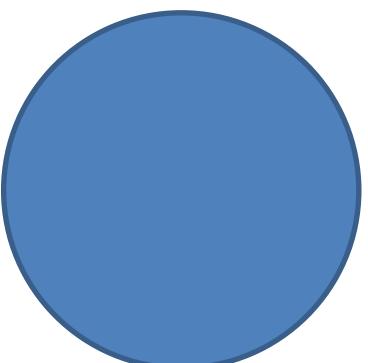
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |  |
|---|------------|------|-------------|-------------|--------------|--|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   |  |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 |  |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  |  |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 |  |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  |  |

En cada iteración repetimos los pasos que hemos llevado a cabo:

- 1. Calculamos la función de pérdida**
2. Calculamos la función de coste
3. Calculamos los nuevos coeficientes

$$L_2(y_i, h_\theta(X_i)) = \left( (\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i \right)^2$$



| Epoch |        |         |             |             |
|-------|--------|---------|-------------|-------------|
| 0     | 0      | 0       | 0           | 103195656,1 |
| 1     | 59,601 | 118,872 | 162,2589808 |             |
| 2     |        |         |             |             |



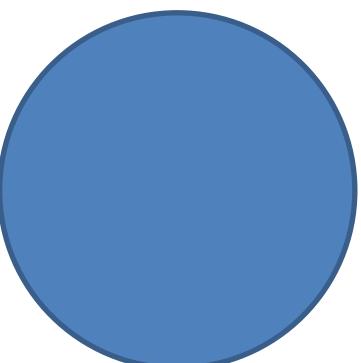
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   | 483577648,2 |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 | 310851137,9 |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  | 157849331,8 |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 | 4505495,788 |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  | 4987047,306 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

- 1. Calculamos la función de pérdida**
2. Calculamos la función de coste
3. Calculamos los nuevos coeficientes

$$L_2(y_i, h_{\theta}(X_i)) = \left( (\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i \right)^2$$



| Epoch |        |         |             |             |
|-------|--------|---------|-------------|-------------|
| 0     | 0      | 0       | 0           | 103195656,1 |
| 1     | 59,601 | 118,872 | 162,2589808 |             |
| 2     |        |         |             |             |



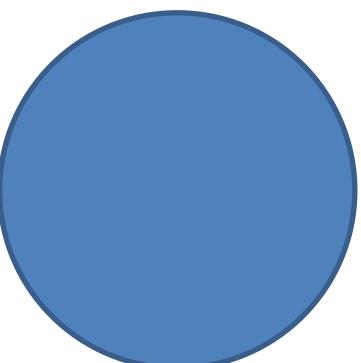
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   | 483577648,2 |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 | 310851137,9 |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  | 157849331,8 |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 | 4505495,788 |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  | 4987047,306 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

1. Calculamos la función de pérdida
2. **Calculamos la función de coste**
3. Calculamos los nuevos coeficientes

$$C(h_\theta, X, y) = \frac{1}{2m} \sum_{i=1}^m L_2(y_i, h_\theta(X_i))$$



| Epoch |        |         |             |             |
|-------|--------|---------|-------------|-------------|
| 0     | 0      | 0       | 0           | 103195656,1 |
| 1     | 59,601 | 118,872 | 162,2589808 |             |
| 2     |        |         |             |             |





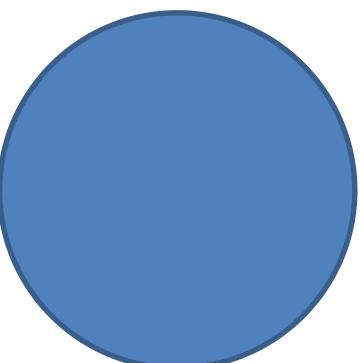
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   | 483577648,2 |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 | 310851137,9 |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  | 157849331,8 |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 | 4505495,788 |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  | 4987047,306 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

1. Calculamos la función de pérdida
2. **Calculamos la función de coste**
3. Calculamos los nuevos coeficientes

$$C(h_\theta, X, y) = \frac{1}{2m} \sum_{i=1}^m L_2(y_i, h_\theta(X_i))$$



| Epoch |        |         |             |             |
|-------|--------|---------|-------------|-------------|
| 0     | 0      | 0       | 0           | 103195656,1 |
| 1     | 59,601 | 118,872 | 162,2589808 | 96177066,1  |
| 2     |        |         |             |             |





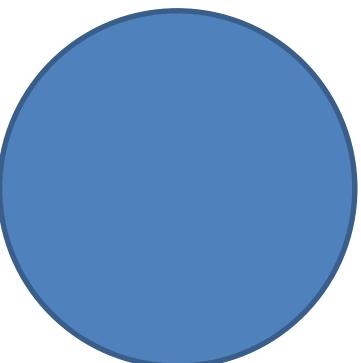
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   | 483577648,2 |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 | 310851137,9 |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  | 157849331,8 |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 | 4505495,788 |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  | 4987047,306 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

1. Calculamos la función de pérdida
2. Calculamos la función de coste
3. **Calculamos los nuevos coeficientes**

$$\theta'_0 = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i)$$



| Epoch |        |         |             |             |
|-------|--------|---------|-------------|-------------|
| 0     | 0      | 0       | 0           | 103195656,1 |
| 1     | 59,601 | 118,872 | 162,2589808 | 96177066,1  |
| 2     |        |         |             |             |



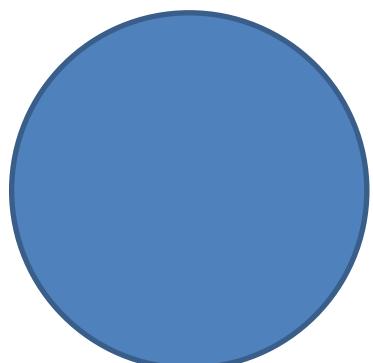
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   | 483577648,2 |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 | 310851137,9 |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  | 157849331,8 |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 | 4505495,788 |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  | 4987047,306 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

1. Calculamos la función de pérdida
2. Calculamos la función de coste
3. **Calculamos los nuevos coeficientes**

$$\theta'_0 = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i)$$



| Epoch | 0           | 0       | 0           | 0          | 103195656,1 |
|-------|-------------|---------|-------------|------------|-------------|
| 1     | 59,601      | 118,872 | 162,2589808 | 96177066,1 |             |
| 2     | 111,6756259 |         |             |            |             |



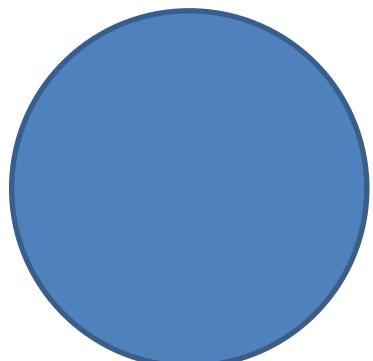
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   | 483577648,2 |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 | 310851137,9 |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  | 157849331,8 |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 | 4505495,788 |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  | 4987047,306 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

1. Calculamos la función de pérdida
2. Calculamos la función de coste
3. **Calculamos los nuevos coeficientes**

$$\theta'_1 = \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i) x_{i,1}$$



| Epoch | 0           | 0       | 0           | 0          | 103195656,1 |
|-------|-------------|---------|-------------|------------|-------------|
| 1     | 59,601      | 118,872 | 162,2589808 | 96177066,1 |             |
| 2     | 111,6756259 |         |             |            |             |



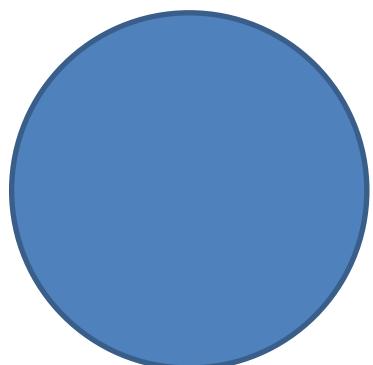
# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 59,601      | -21990,399   | 483577648,2 |
| 1 | 18.205     | 2    | 1,7052      | 574,0290141 | -17630,97099 | 310851137,9 |
| 2 | 13.840     | 3    | 5,3         | 1276,189598 | -12563,8104  | 157849331,8 |
| 3 | 4.706      | 7    | 10,4258     | 2583,384682 | -2122,615318 | 4505495,788 |
| 4 | 800        | 10   | 11          | 3033,169789 | 2233,169789  | 4987047,306 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

1. Calculamos la función de pérdida
2. Calculamos la función de coste
3. **Calculamos los nuevos coeficientes**

$$\theta'_2 = \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m ((\theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}) - y_i) x_{i,2}$$



| Epoch |             |             |             |             |
|-------|-------------|-------------|-------------|-------------|
| 0     | 0           | 0           | 0           | 103195656,1 |
| 1     | 59,601      | 118,872     | 162,2589808 | 96177066,1  |
| 2     | 111,6756259 | 184,3519825 | 256,4766028 |             |

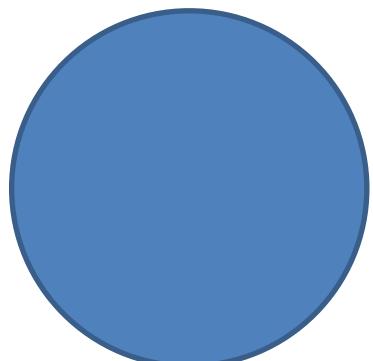


# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |  |  |  |
|---|------------|------|-------------|--|--|--|
| 0 | 22.050     | 0    | 0           |  |  |  |
| 1 | 18.205     | 2    | 1,7052      |  |  |  |
| 2 | 13.840     | 3    | 5,3         |  |  |  |
| 3 | 4.706      | 7    | 10,4258     |  |  |  |
| 4 | 800        | 10   | 11          |  |  |  |

En cada iteración repetimos los pasos que hemos llevado a cabo:

- 1. Calculamos la función de pérdida**
2. Calculamos la función de coste
3. Calculamos los nuevos coeficientes



| Epoch | 0           | 0           | 0           | 0          | 103195656,1 |
|-------|-------------|-------------|-------------|------------|-------------|
| 1     | 59,601      | 118,872     | 162,2589808 | 96177066,1 |             |
| 2     | 111,6756259 | 184,3519825 | 256,4766028 |            |             |



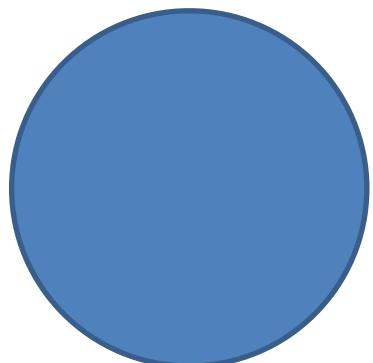


# Ejemplo de algoritmo: repetir

|   | Precio (€) | Edad | Kilometraje |             |              |             |
|---|------------|------|-------------|-------------|--------------|-------------|
| 0 | 22.050     | 0    | 0           | 111,6756259 | -21938,32437 | 481290076,3 |
| 1 | 18.205     | 2    | 1,7052      | 917,723494  | -17287,27651 | 298849929   |
| 2 | 13.840     | 3    | 5,3         | 2024,057568 | -11815,94243 | 139616495,6 |
| 3 | 4.706      | 7    | 10,4258     | 4076,113269 | -629,8867311 | 396757,2941 |
| 4 | 800        | 10   | 11          | 4776,438082 | 3976,438082  | 15812059,82 |

En cada iteración repetimos los pasos que hemos llevado a cabo:

- 1. Calculamos la función de pérdida**
- 2. Calculamos la función de coste**
- 3. Calculamos los nuevos coeficientes**



| Epoch | 0           | 0           | 0           | 0          | 103195656,1 |
|-------|-------------|-------------|-------------|------------|-------------|
| 1     | 59,601      | 118,872     | 162,2589808 | 96177066,1 |             |
| 2     | 111,6756259 | 184,3519825 | 256,4766028 | 93596531,8 |             |

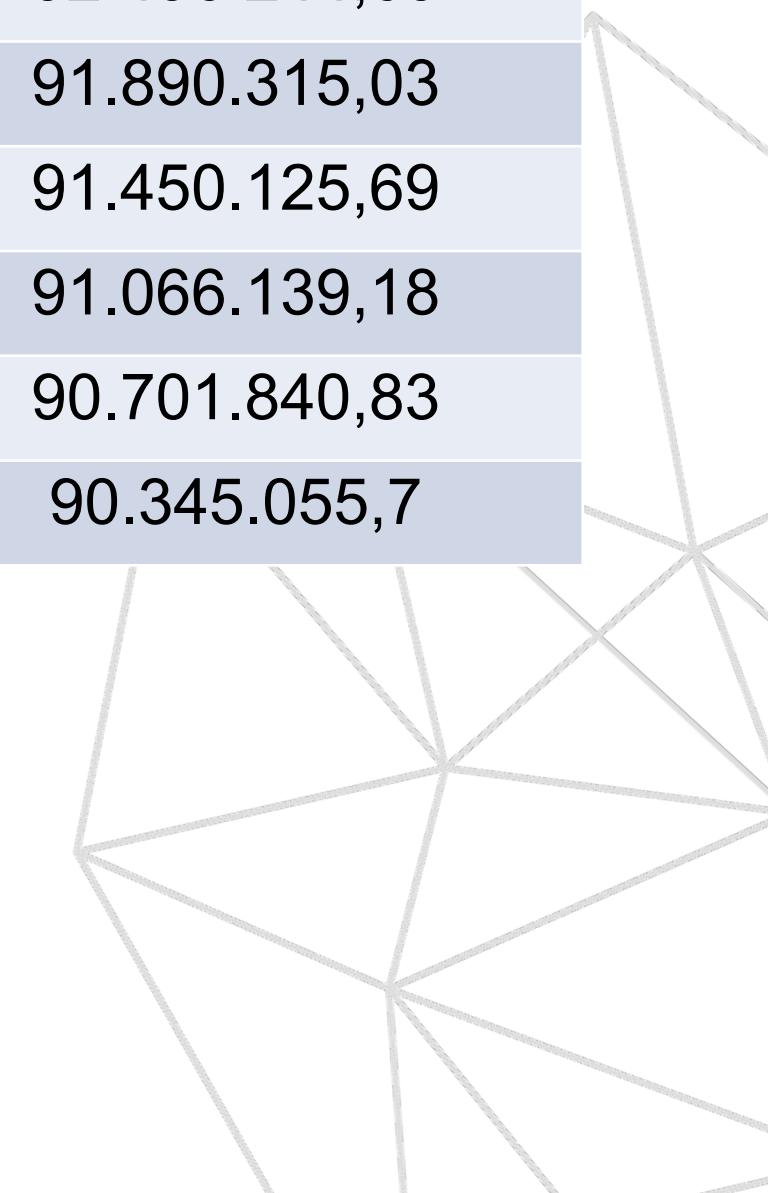
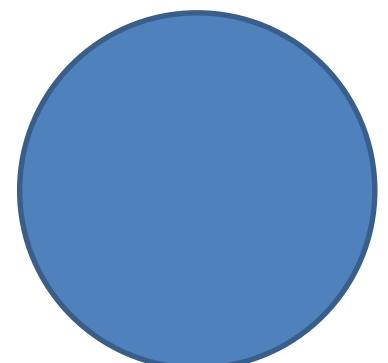




# Ejemplo de algoritmo: repetir

| Epoch |             |             |             |               |               |
|-------|-------------|-------------|-------------|---------------|---------------|
| 0     | 0           | 0           | 0           | 0             | 103.195.656,1 |
| 1     | 59,601      | 118,872     | 162,2589808 | 96.177.066,1  |               |
| 2     | 111,6756259 | 184,3519825 | 256,4766028 | 93.596.531,8  |               |
| 3     | 159,3706179 | 219,0191891 | 311,4056158 | 92.496.244,69 |               |
| 4     | 204,5027696 | 235,9093734 | 343,6460288 | 91.890.315,03 |               |
| 5     | 248,1210493 | 242,5484952 | 362,7816425 | 91.450.125,69 |               |
| 6     | 290,8311323 | 243,2815147 | 374,3455512 | 91.066.139,18 |               |
| 7     | 332,9827649 | 240,6168885 | 381,532233  | 90.701.840,83 |               |
| 8     | 374,7779366 | 236,0027948 | 386,1859995 | 90.345.055,7  |               |

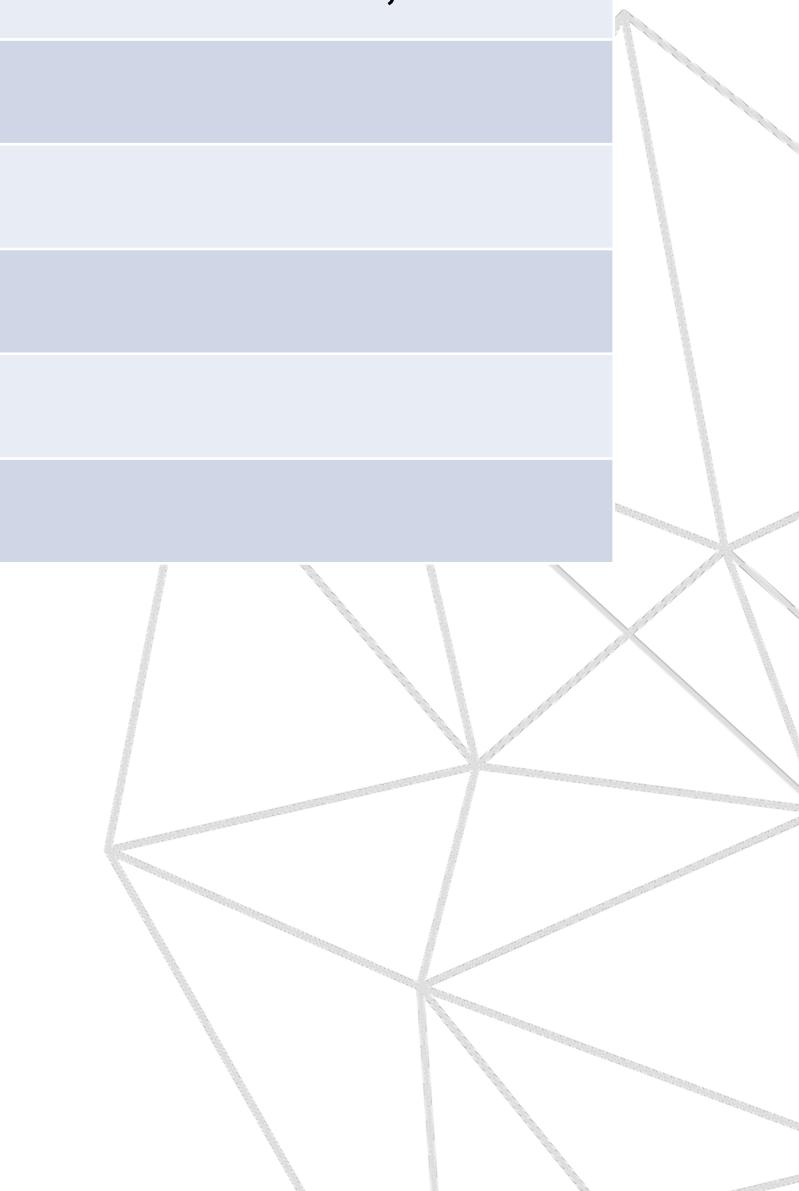
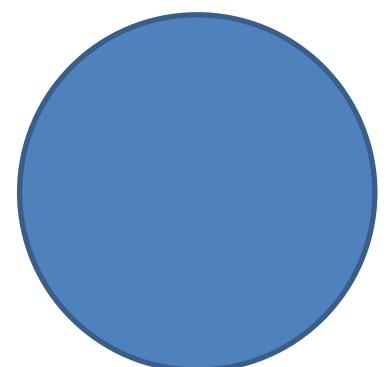
¿Y si repetimos este proceso muchas veces?





# Ejemplo de algoritmo: repetir

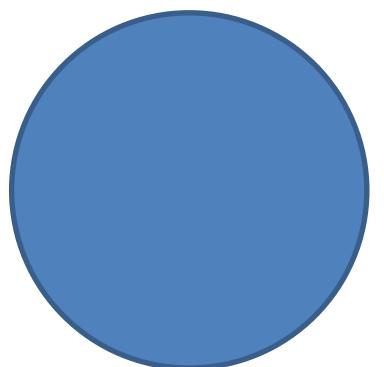
| Epoch |             |              |             |               |               |
|-------|-------------|--------------|-------------|---------------|---------------|
| 0     | 0           | 0            | 0           | 0             | 103.195.656,1 |
| 1     | 59,601      | 118,872      | 162,2589808 | 96.177.066,1  |               |
| ...   | ...         | ...          | ...         | ...           | ...           |
| 50    | 2044,032109 | -42,88488877 | 430,1738742 | 76.700.756,76 |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |





# Ejemplo de algoritmo: repetir

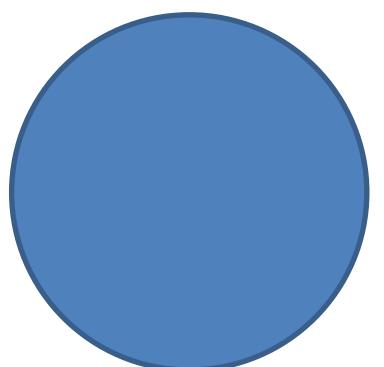
| Epoch |             |              |             |               |               |
|-------|-------------|--------------|-------------|---------------|---------------|
| 0     | 0           | 0            | 0           | 0             | 103.195.656,1 |
| 1     | 59,601      | 118,872      | 162,2589808 | 96.177.066,1  |               |
| ...   | ...         | ...          | ...         | ...           | ...           |
| 50    | 2044,032109 | -42,88488877 | 430,1738742 | 76.700.756,76 |               |
| ...   | ...         | ...          | ...         | ...           | ...           |
| 100   | 3860,957427 | -322,8381216 | 451,1829339 | 63.182.458,72 |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |





# Ejemplo de algoritmo: repetir

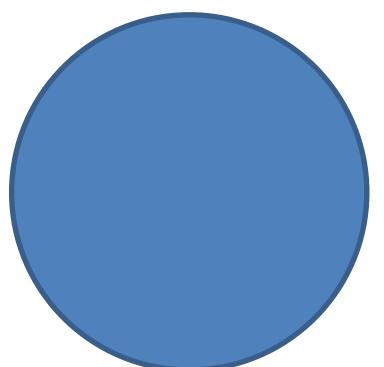
| Epoch |             |              |             |               |               |
|-------|-------------|--------------|-------------|---------------|---------------|
| 0     | 0           | 0            | 0           | 0             | 103.195.656,1 |
| 1     | 59,601      | 118,872      | 162,2589808 | 96.177.066,1  |               |
| ...   | ...         | ...          | ...         | ...           | ...           |
| 50    | 2044,032109 | -42,88488877 | 430,1738742 | 76.700.756,76 |               |
| ...   | ...         | ...          | ...         | ...           | ...           |
| 100   | 3860,957427 | -322,8381216 | 451,1829339 | 63.182.458,72 |               |
| ...   | ...         | ...          | ...         | ...           | ...           |
| 250   | 8366,740646 | -884,6992658 | 399,315326  | 35.510.344,59 |               |
|       |             |              |             |               |               |





# Ejemplo de algoritmo: repetir

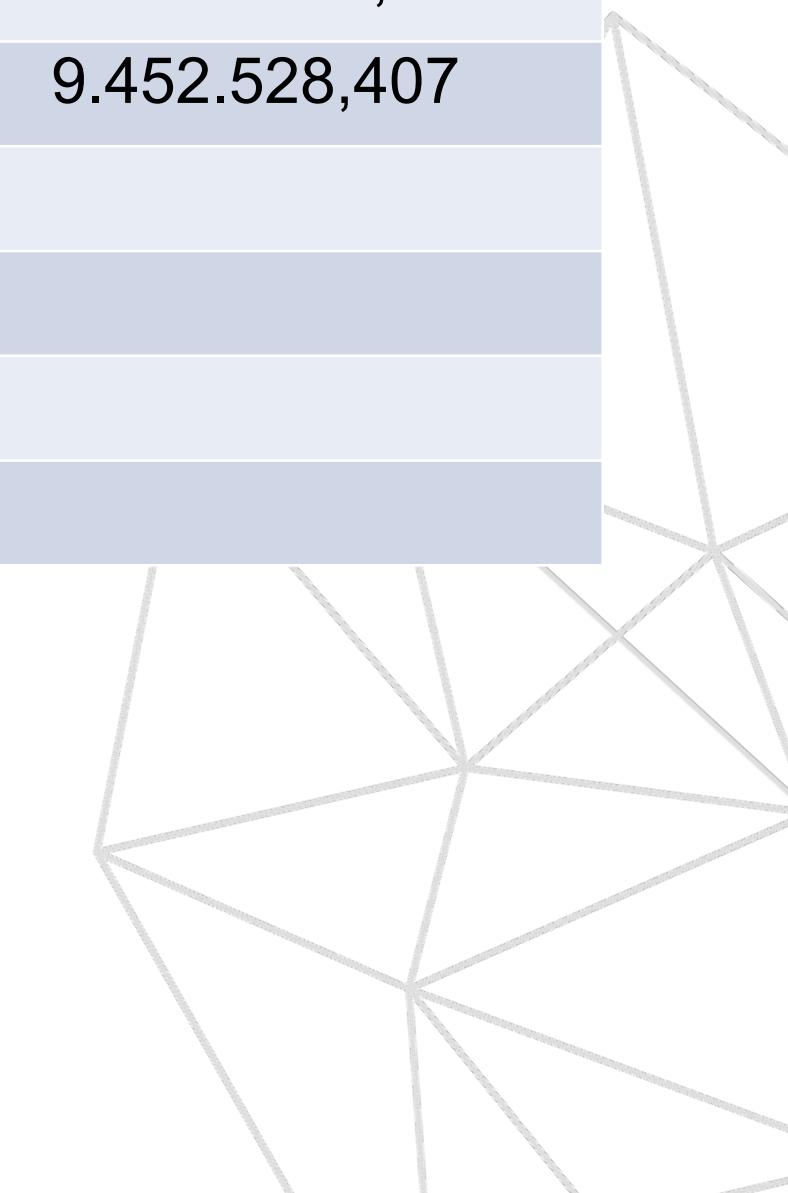
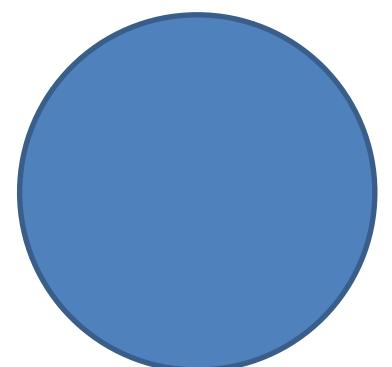
| Epoch |             |              |             |               |               |
|-------|-------------|--------------|-------------|---------------|---------------|
| 0     | 0           | 0            | 0           | 0             | 103.195.656,1 |
| 100   | 3860,957427 | -322,8381216 | 451,1829339 | 63.182.458,72 |               |
| 250   | 8366,740646 | -884,6992658 | 399,315326  | 35.510.344,59 |               |
| 400   | 11742,69196 | -1171,200296 | 254,8529261 | 20.082.241,19 |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |





# Ejemplo de algoritmo: repetir

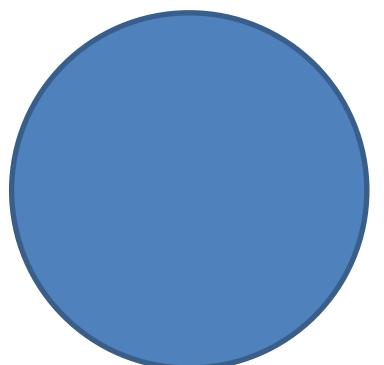
| Epoch |             |              |             |               |               |
|-------|-------------|--------------|-------------|---------------|---------------|
| 0     | 0           | 0            | 0           | 0             | 103.195.656,1 |
| 100   | 3860,957427 | -322,8381216 | 451,1829339 | 63.182.458,72 |               |
| 250   | 8366,740646 | -884,6992658 | 399,315326  | 35.510.344,59 |               |
| 400   | 11742,69196 | -1171,200296 | 254,8529261 | 20.082.241,19 |               |
| 600   | 14972,72676 | -1324,088257 | 21,43333607 | 9.452.528,407 |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |
|       |             |              |             |               |               |





# Ejemplo de algoritmo: repetir

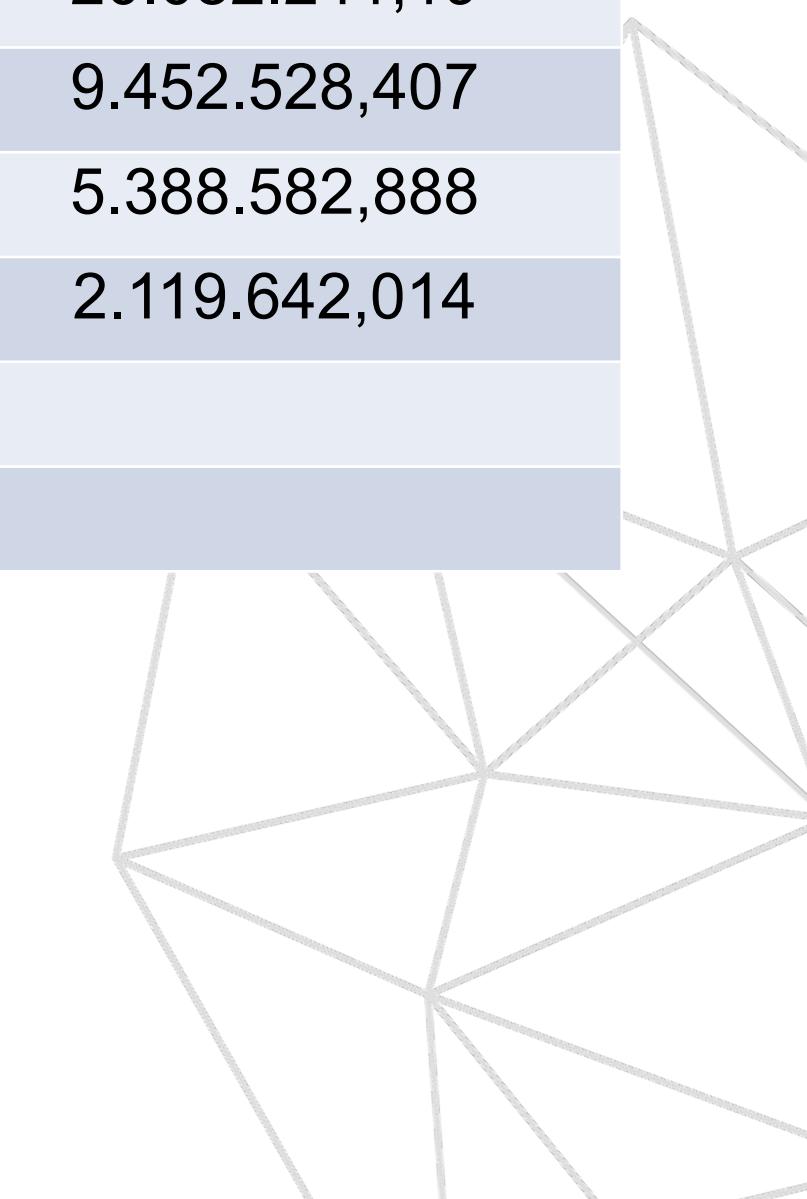
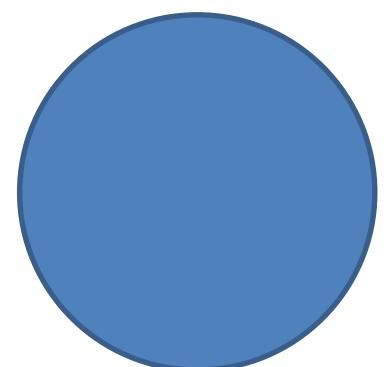
| Epoch |             |              |              |               |               |
|-------|-------------|--------------|--------------|---------------|---------------|
| 0     | 0           | 0            | 0            | 0             | 103.195.656,1 |
| 100   | 3860,957427 | -322,8381216 | 451,1829339  | 63.182.458,72 |               |
| 250   | 8366,740646 | -884,6992658 | 399,315326   | 35.510.344,59 |               |
| 400   | 11742,69196 | -1171,200296 | 254,8529261  | 20.082.241,19 |               |
| 600   | 14972,72676 | -1324,088257 | 21,43333607  | 9.452.528,407 |               |
| 750   | 16705,18155 | -1351,371027 | -146,7343859 | 5.388.582,888 |               |
|       |             |              |              |               |               |
|       |             |              |              |               |               |
|       |             |              |              |               |               |





# Ejemplo de algoritmo: repetir

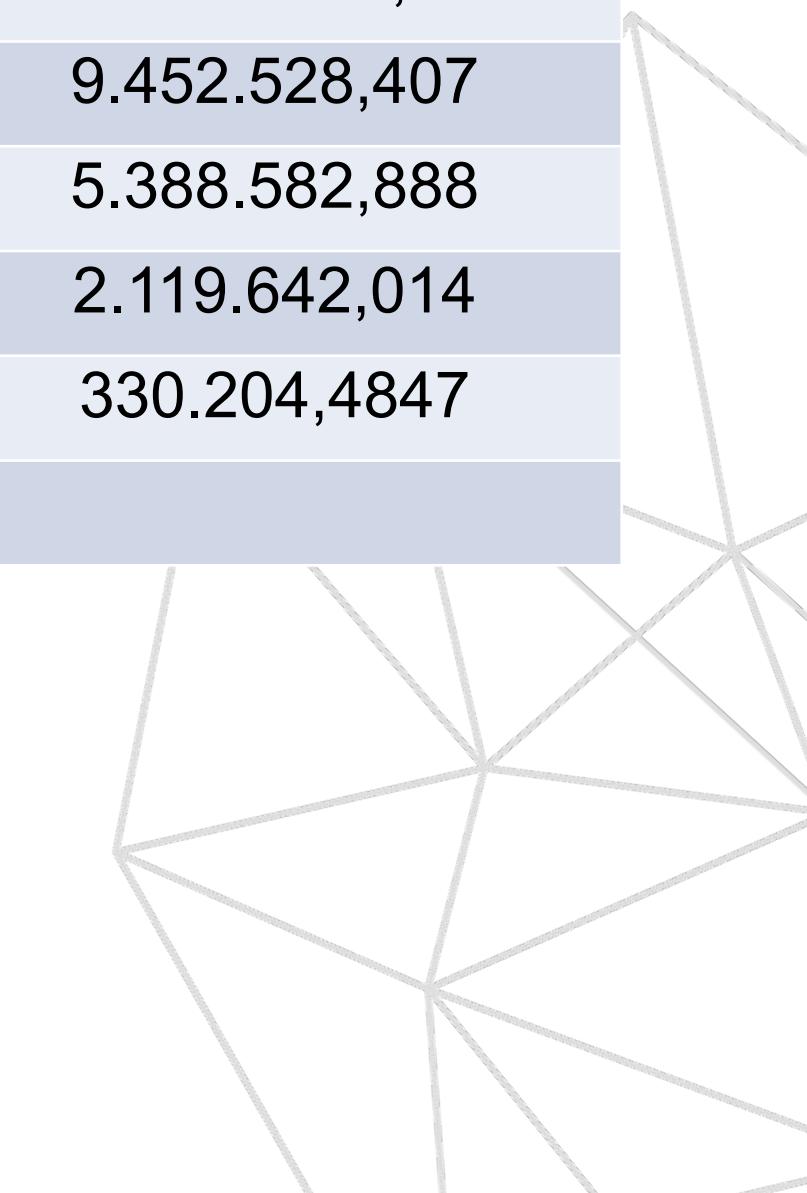
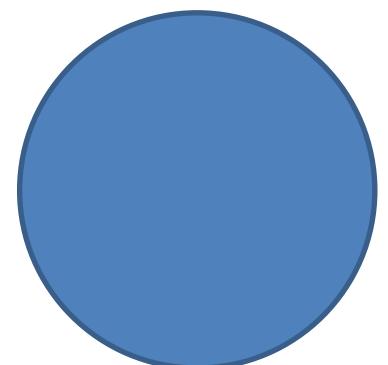
| Epoch |             |              |              |               |               |
|-------|-------------|--------------|--------------|---------------|---------------|
| 0     | 0           | 0            | 0            | 0             | 103.195.656,1 |
| 100   | 3860,957427 | -322,8381216 | 451,1829339  | 63.182.458,72 |               |
| 250   | 8366,740646 | -884,6992658 | 399,315326   | 35.510.344,59 |               |
| 400   | 11742,69196 | -1171,200296 | 254,8529261  | 20.082.241,19 |               |
| 600   | 14972,72676 | -1324,088257 | 21,43333607  | 9.452.528,407 |               |
| 750   | 16705,18155 | -1351,371027 | -146,7343859 | 5.388.582,888 |               |
| 1000  | 18695,27742 | -1329,143668 | -381,9773559 | 2.119.642,014 |               |
|       |             |              |              |               |               |
|       |             |              |              |               |               |





# Ejemplo de algoritmo: repetir

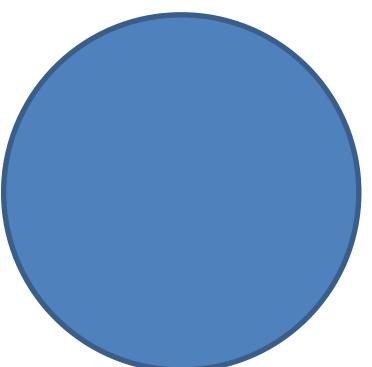
| Epoch |             |              |              |               |               |
|-------|-------------|--------------|--------------|---------------|---------------|
| 0     | 0           | 0            | 0            | 0             | 103.195.656,1 |
| 100   | 3860,957427 | -322,8381216 | 451,1829339  | 63.182.458,72 |               |
| 250   | 8366,740646 | -884,6992658 | 399,315326   | 35.510.344,59 |               |
| 400   | 11742,69196 | -1171,200296 | 254,8529261  | 20.082.241,19 |               |
| 600   | 14972,72676 | -1324,088257 | 21,43333607  | 9.452.528,407 |               |
| 750   | 16705,18155 | -1351,371027 | -146,7343859 | 5.388.582,888 |               |
| 1000  | 18695,27742 | -1329,143668 | -381,9773559 | 2.119.642,014 |               |
| 1500  | 20716,39551 | -1237,786979 | -674,9021989 | 330.204,4847  |               |





# Ejemplo de algoritmo: repetir

| Epoch |             |              |              |               |
|-------|-------------|--------------|--------------|---------------|
| 0     | 0           | 0            | 0            | 103.195.656,1 |
| 100   | 3860,957427 | -322,8381216 | 451,1829339  | 63.182.458,72 |
| 250   | 8366,740646 | -884,6992658 | 399,315326   | 35.510.344,59 |
| 400   | 11742,69196 | -1171,200296 | 254,8529261  | 20.082.241,19 |
| 600   | 14972,72676 | -1324,088257 | 21,43333607  | 9.452.528,407 |
| 750   | 16705,18155 | -1351,371027 | -146,7343859 | 5.388.582,888 |
| 1000  | 18695,27742 | -1329,143668 | -381,9773559 | 2.119.642,014 |
| 1500  | 20716,39551 | -1237,786979 | -674,9021989 | 330.204,4847  |
| 2000  | 21509,76424 | -1178,325688 | -808,4198832 | 51684,71789   |



$$\hat{y}_i = h_{\theta}(X_i) = h_{\theta}(x_{i,1}, x_{i,2}) = \theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2}$$

Kilometraje del coche  $i$      
 Precio del coche  $i$      
 Edad del coche  $i$

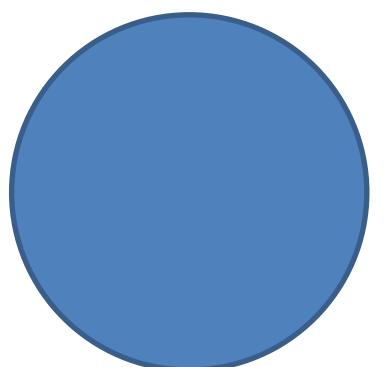




# 04

## Evaluación de modelos de clasificación

Métricas de evaluación de los modelos de clasificación: matriz de confusión, accuracy, error rate, precisión, recall, F1-score, especificidad, coeficiente de correlación de matthews, ROC y AUC.



|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |



# Matriz de confusión

Una vez contamos con un modelos entrenado, debemos ser capaces de evaluarlos. La evaluación de estos modelos depende del tipo de problema. En clasificación, la **matriz de confusión** es un concepto clave para evaluar los modelos.

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **TP:** True Positive
- **TN:** True Negative
- **FP:** False Positive (Error tipo I)
- **FN:** False Negative (Error tipo II)



# Accuracy

Partiendo de la matriz de confusión se definen las siguientes métricas:

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **Accuracy:** indica el porcentaje de acierto

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Por tanto su valor máximo es 1 y mínimo 0



# Error rate

Partiendo de la matriz de confusión se definen las siguientes métricas:

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **Error rate:** indica la tasa de error

$$\frac{FP + FN}{TP + TN + FP + FN}$$

Por tanto su valor máximo es 1 y mínimo 0



# Precisión

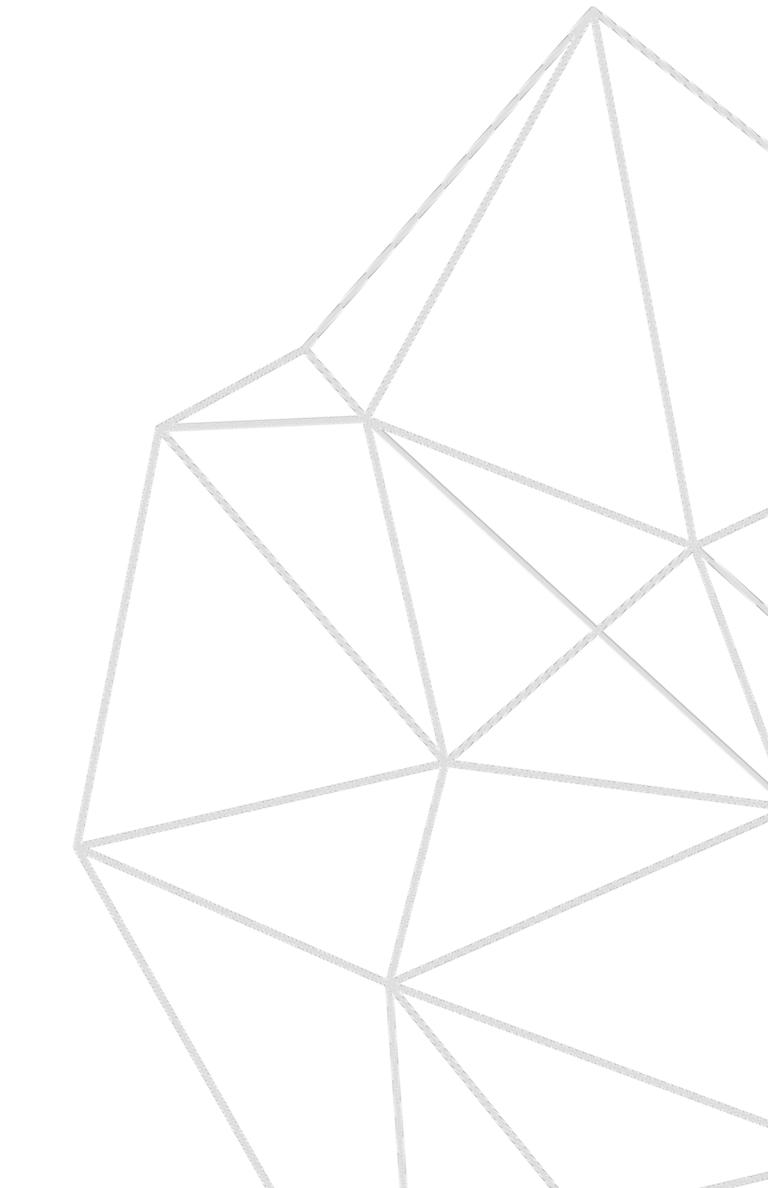
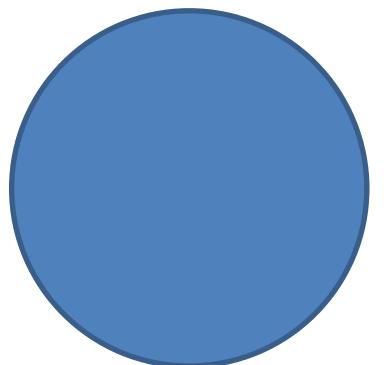
Partiendo de la matriz de confusión se definen las siguientes métricas:

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **Precisión o PPV: Positive Predictive Value**

$$\frac{TP}{TP + FP}$$

Por tanto su valor máximo es 1 y mínimo 0





# Recall

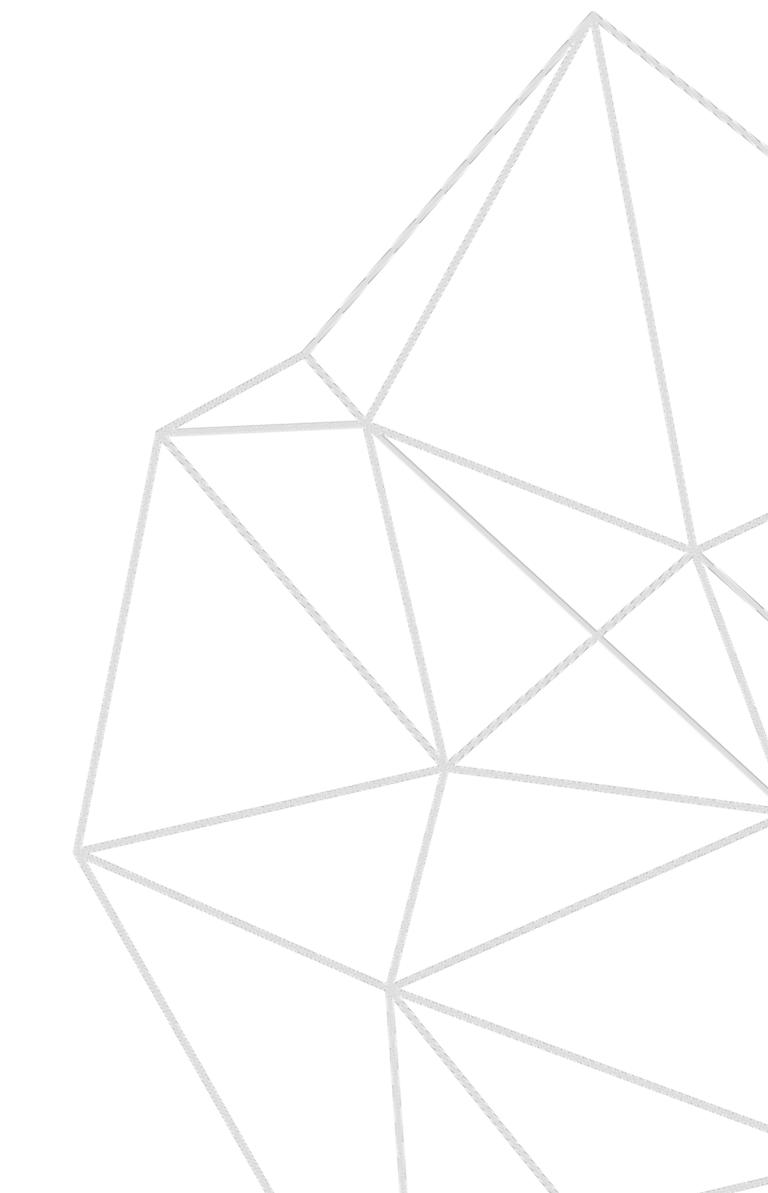
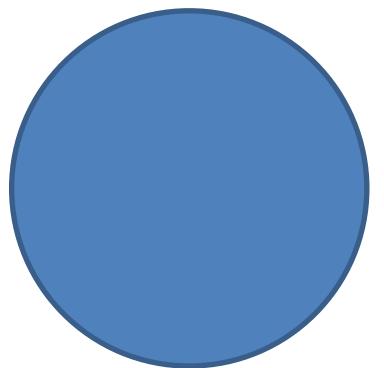
Partiendo de la matriz de confusión se definen las siguientes métricas:

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **Recall o TPR: True Positive Rate**

$$\frac{TP}{TP + FN}$$

Por tanto su valor máximo es 1 y mínimo 0





# F1-score

Partiendo de la matriz de confusión se definen las siguientes métricas:

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **F1-score**: combina la precisión y el recall:

$$2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} = 2 \cdot \frac{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}{\frac{TP}{TP + FP} + \frac{TP}{TP + FN}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Por tanto su valor máximo es 1 y mínimo 0



# Especificidad

Partiendo de la matriz de confusión se definen las siguientes métricas:

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **Especificidad o TNR: True Negative Rate**

$$\frac{TN}{FP + TN}$$

Por tanto su valor máximo es 1 y mínimo 0



# Coeficiente de correlación de Matthews

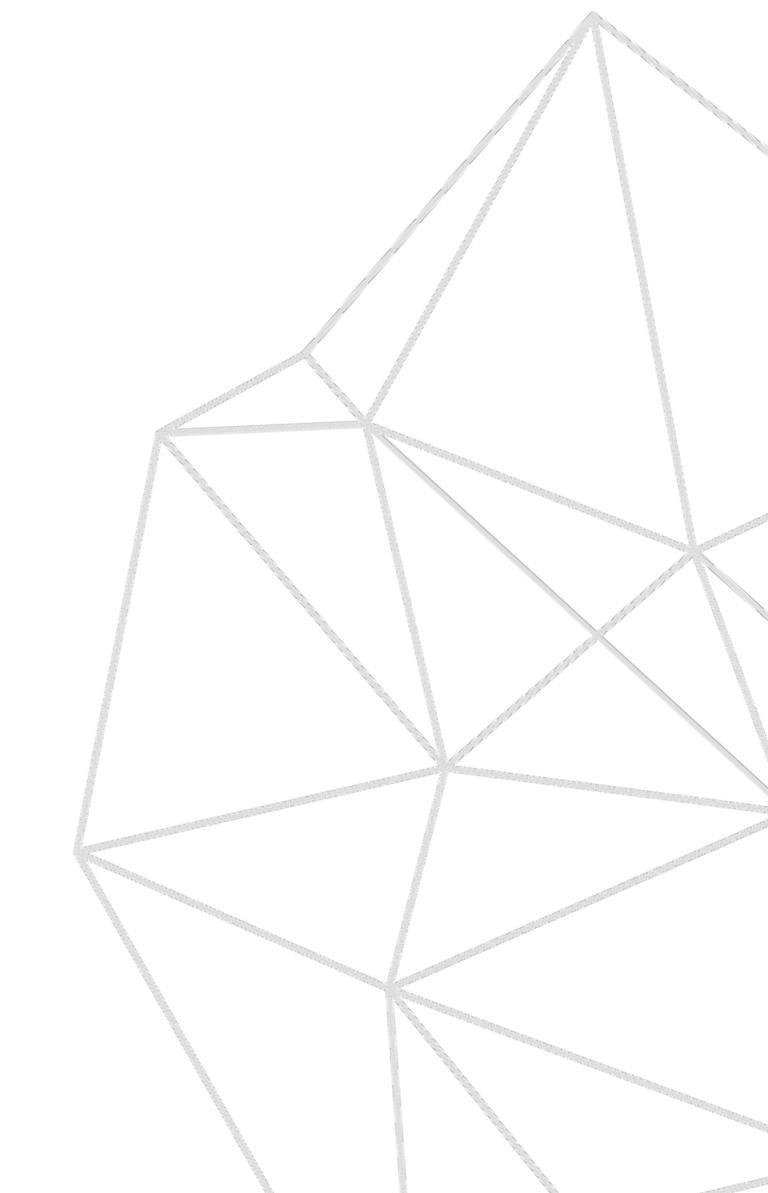
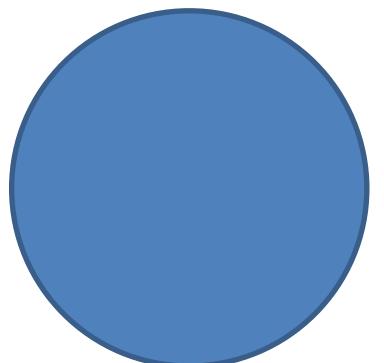
Partiendo de la matriz de confusión se definen las siguientes métricas:

|          |    | Predicción |    |
|----------|----|------------|----|
|          |    | Sí         | No |
| Realidad | Sí | TP         | FN |
|          | No | FP         | TN |

- **Coeficiente de correlación de Matthews o MCC:**

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Por tanto su valor máximo es 1 y mínimo -1

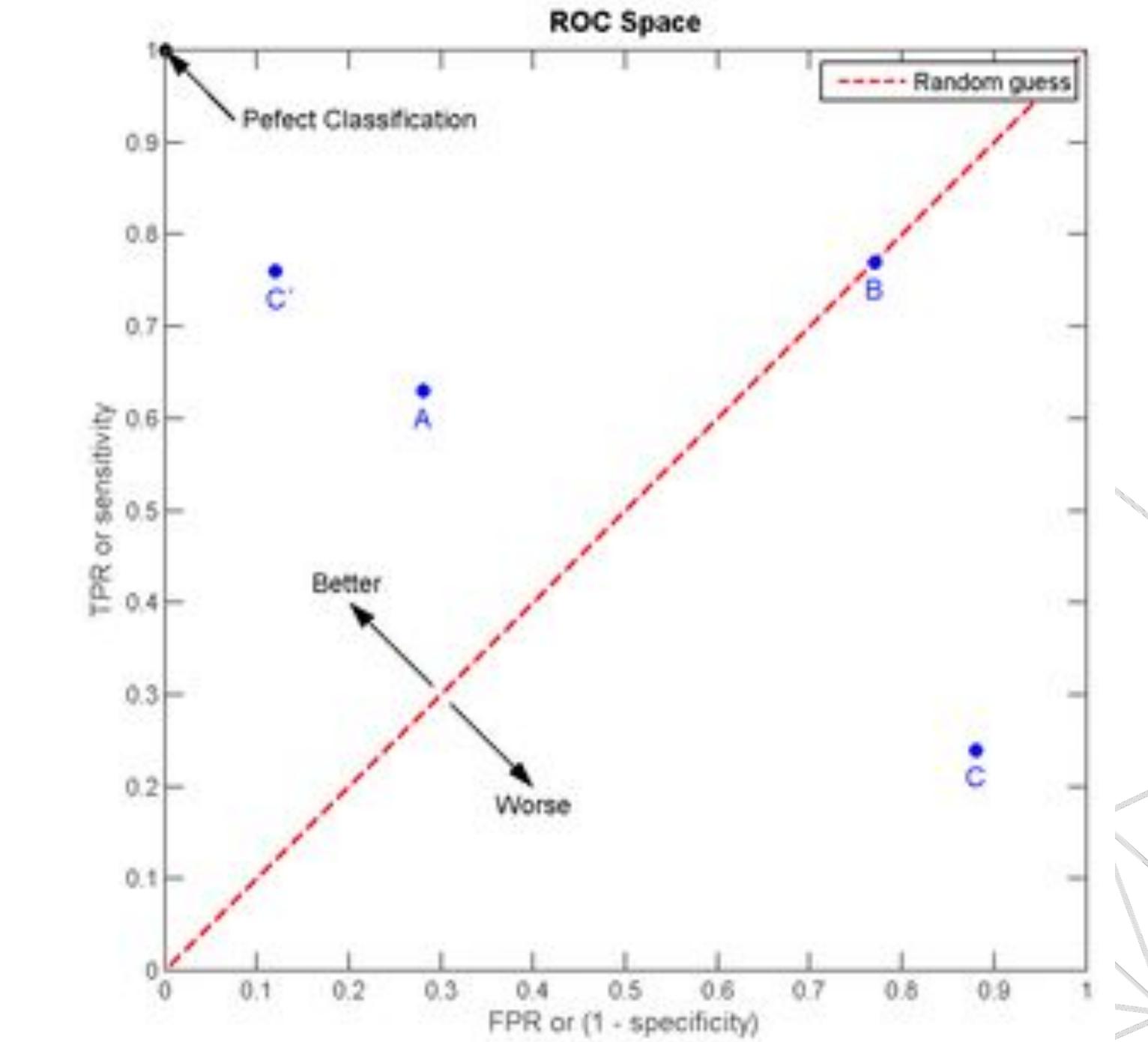




# Espacio ROC

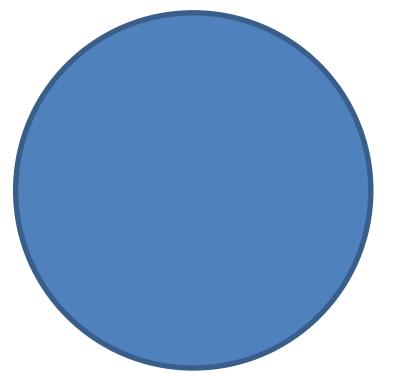
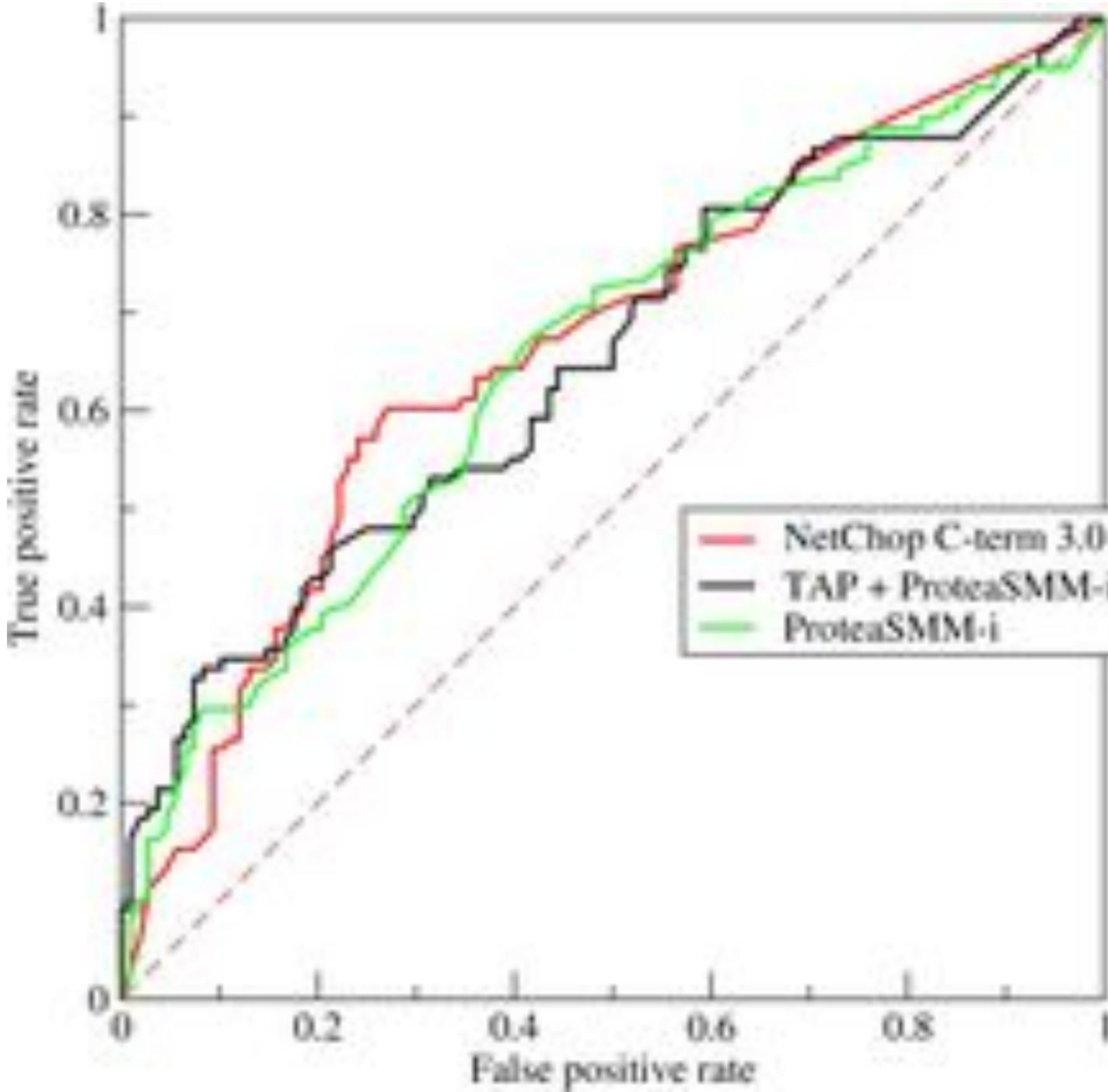
|          |    | Predicción |    |    |
|----------|----|------------|----|----|
|          |    | A          | Sí | No |
| Realidad | A  | 63         | 37 |    |
|          | Sí | 28         | 72 |    |
|          |    | C          | Sí | No |
| Realidad | C  | 24         | 76 |    |
|          | No | 88         | 12 |    |

|          |    | Predicción |    |    |
|----------|----|------------|----|----|
|          |    | B          | Sí | No |
| Realidad | B  | 77         | 23 |    |
|          | No | 77         | 23 |    |
|          |    | C'         | Sí | No |
| Realidad | C' | 76         | 24 |    |
|          | No | 12         | 88 |    |





# AUC



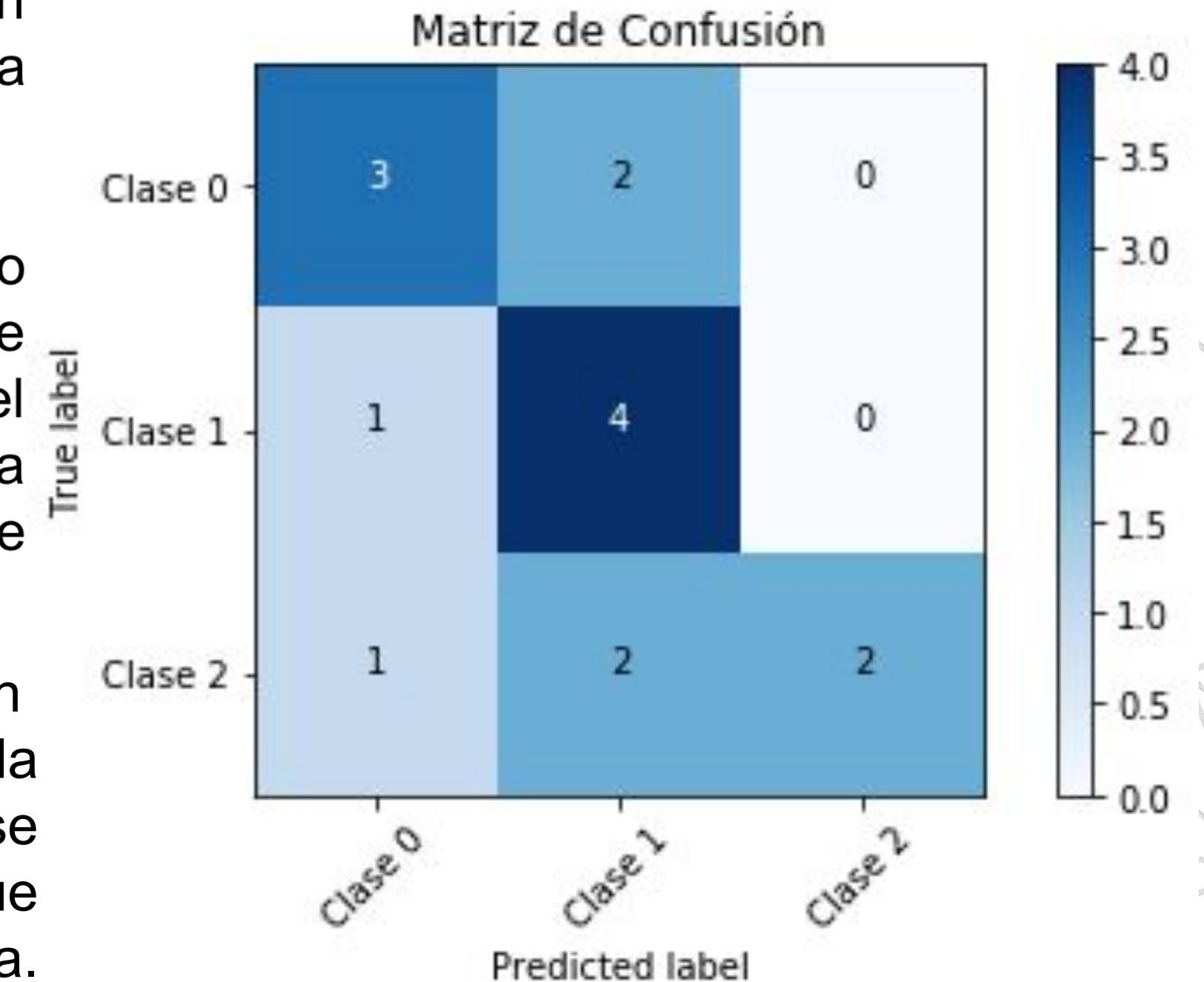


# Modelos de clasificación multiclas

Cuando nos enfrentamos a problemas multiclas, también se puede extraer una matriz de confusión.

Para el caso de algoritmos de aprendizaje de clasificación binaria existen dos estrategias para hacer frente a problemas multiclas:

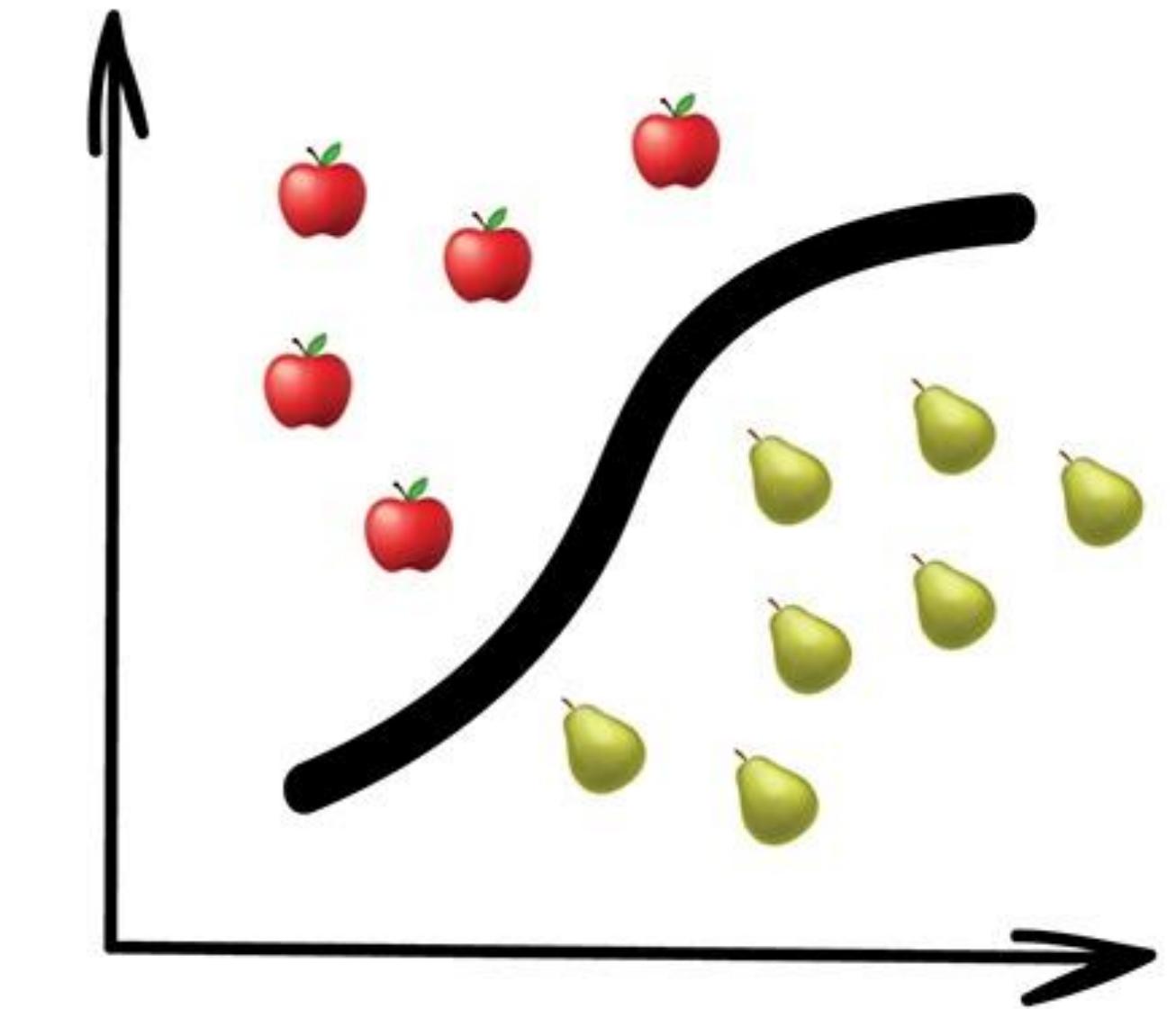
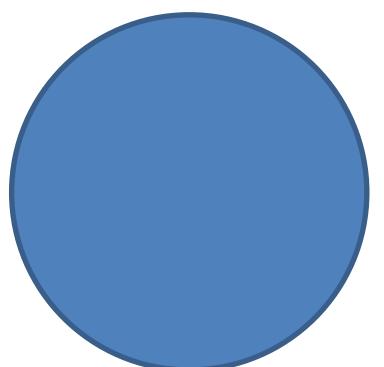
- **One vs. One (OVO)**: se entrena un clasificador binario para cada par de clases (si hubiera n clases habría que entrenar  $n(n - 1)/2$  clasificadores). Se da un voto (o el valor de confianza) a la clase indicada por cada clasificador y la que obtiene más votos es la que se asigna. Es muy costoso computacionalmente.
  - **One vs. Rest (OVR)**: se entrena un clasificador binario para cada clase. Se da un voto (o el valor de confianza) a la clase indicada por cada clasificador y la que obtiene más votos es la que se asigna. Problemas con el balanceo de los datos.



# 05

## Algoritmos de clasificación

Se presentan los algoritmos de aprendizaje más famosos para el problema de clasificación: regresión logística, K-NN, Naive Bayes, Árboles de decisión y SVM.

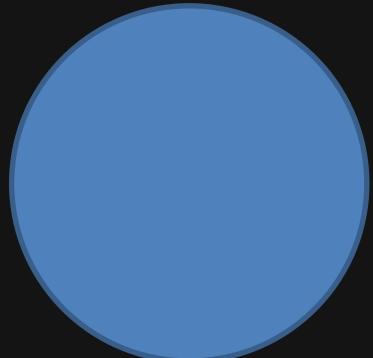


Classification



“ Todos los modelos son incorrectos, pero  
algunos son útiles ”

George Edward Pelham Box  
Estadístico

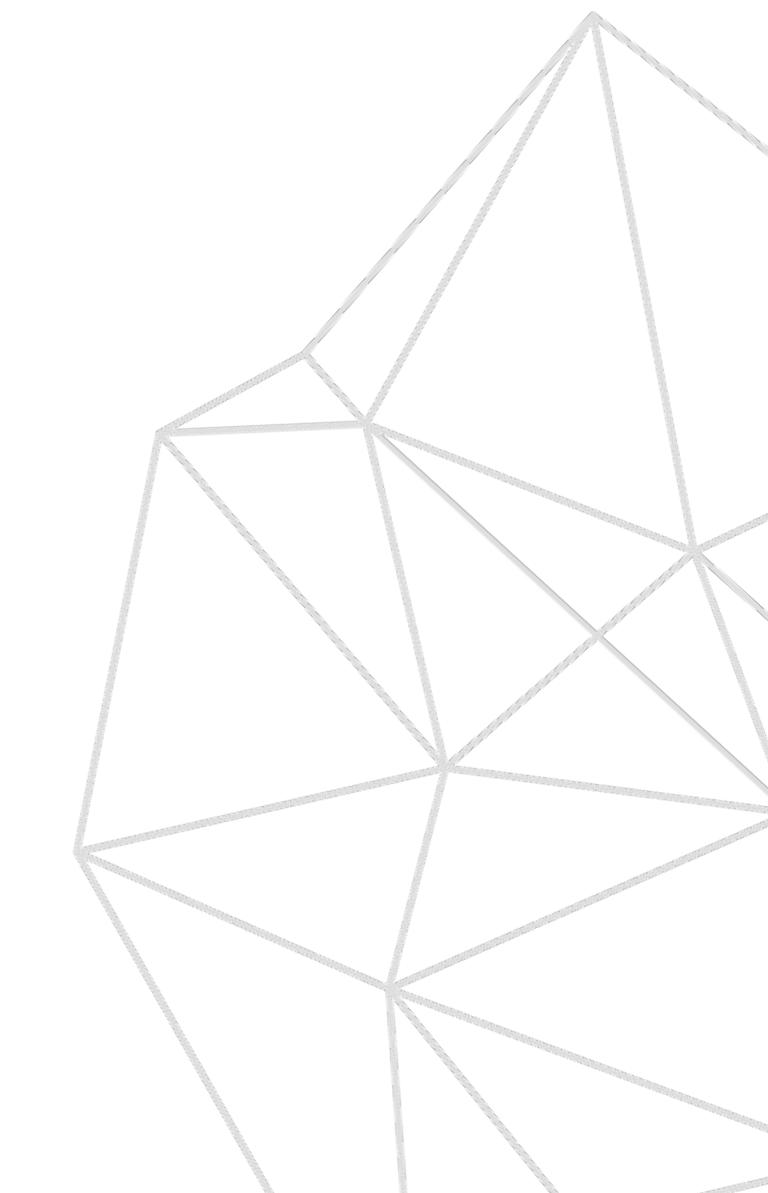
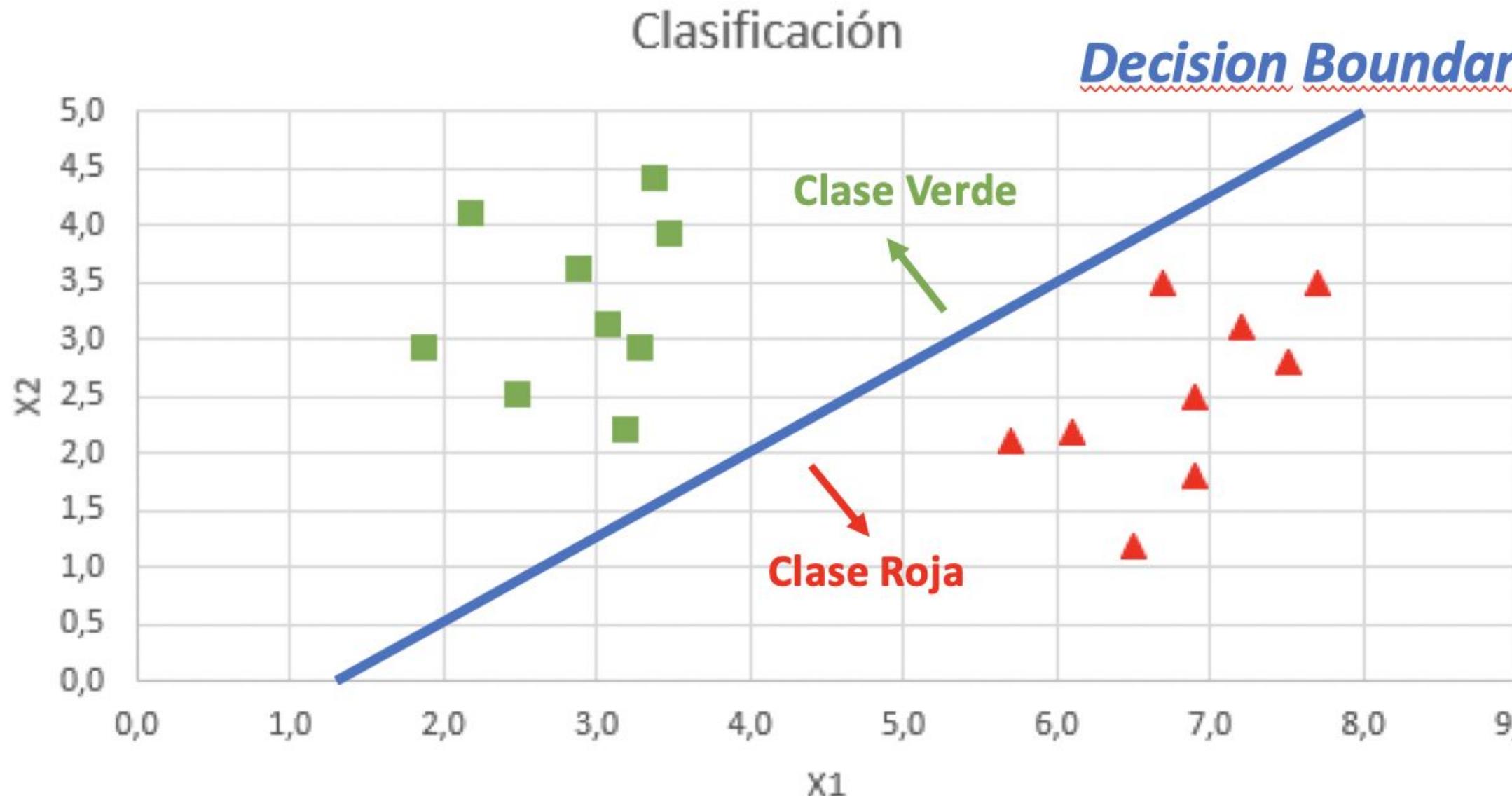




# Regresión logística: Decision Boundary

El objetivo es encontrar una línea (llamada *Decision Boundary*) que separe las dos clases a clasificar. Esta línea vendrá dada por:

$$Z = \theta_0 + \sum_{j=1}^n \theta_j x_{i,j} = \theta_0 + \theta_1 x_{i,1} + \dots + \theta_n x_{i,n}$$

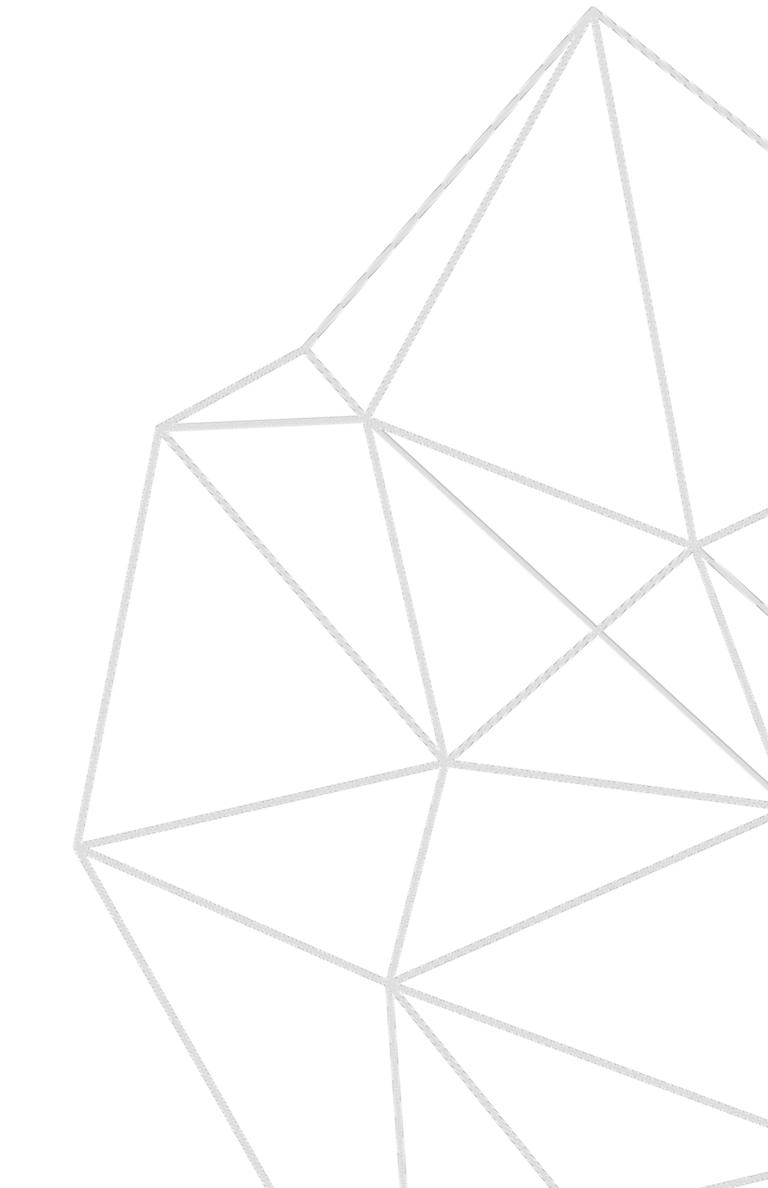
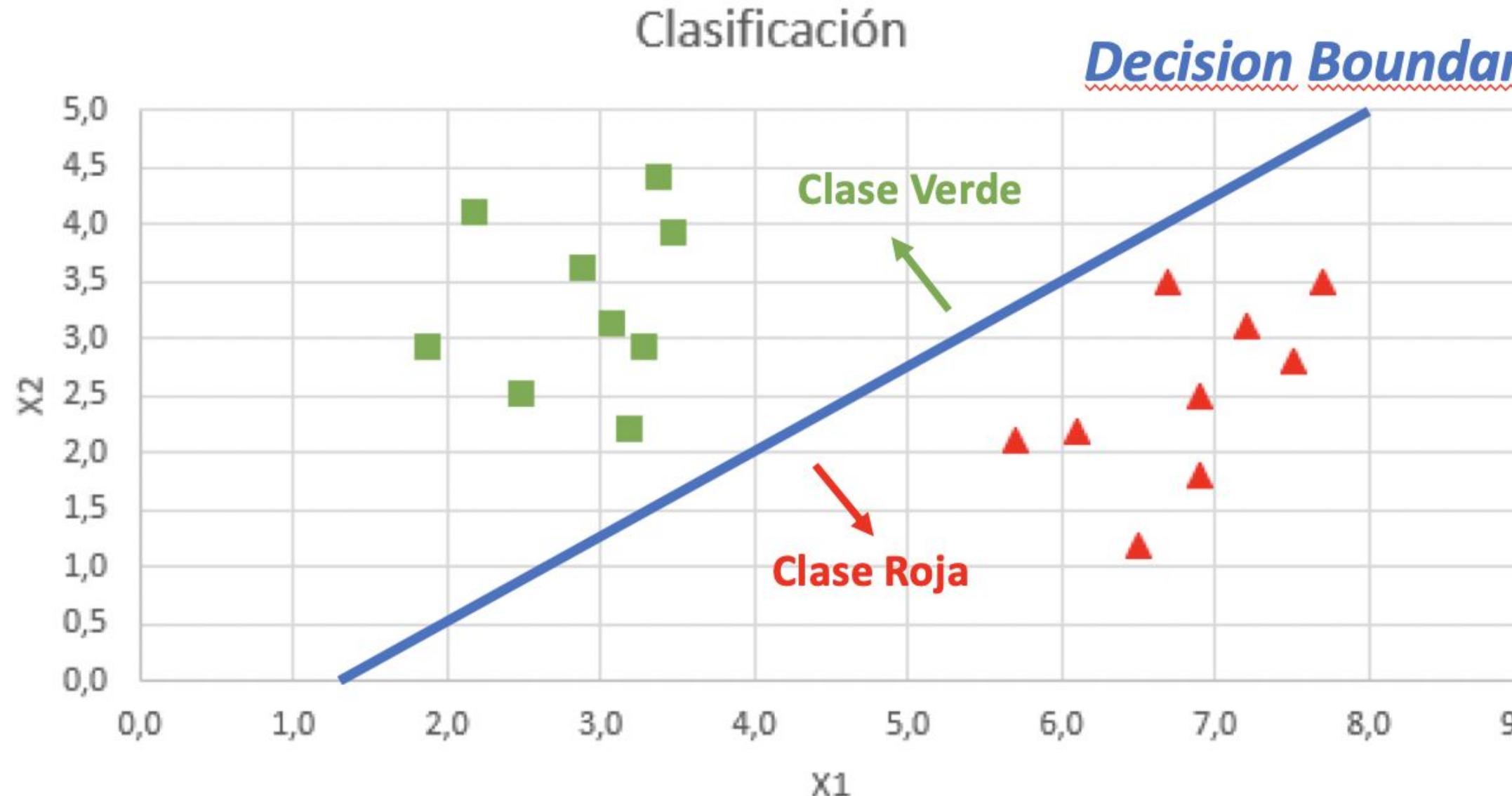




# Regresión logística: Decision Boundary

$$Z = \theta_0 + \theta_1 x_{i,1} + \dots + \theta_n x_{i,n} < 0$$

$$Z = \theta_0 + \theta_1 x_{i,1} + \dots + \theta_n x_{i,n} \geq 0$$

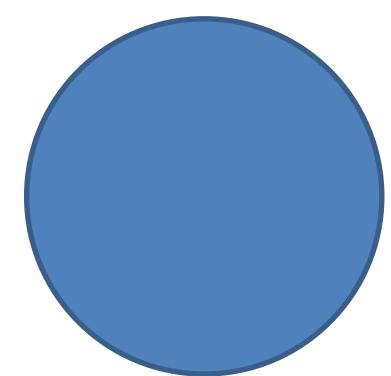
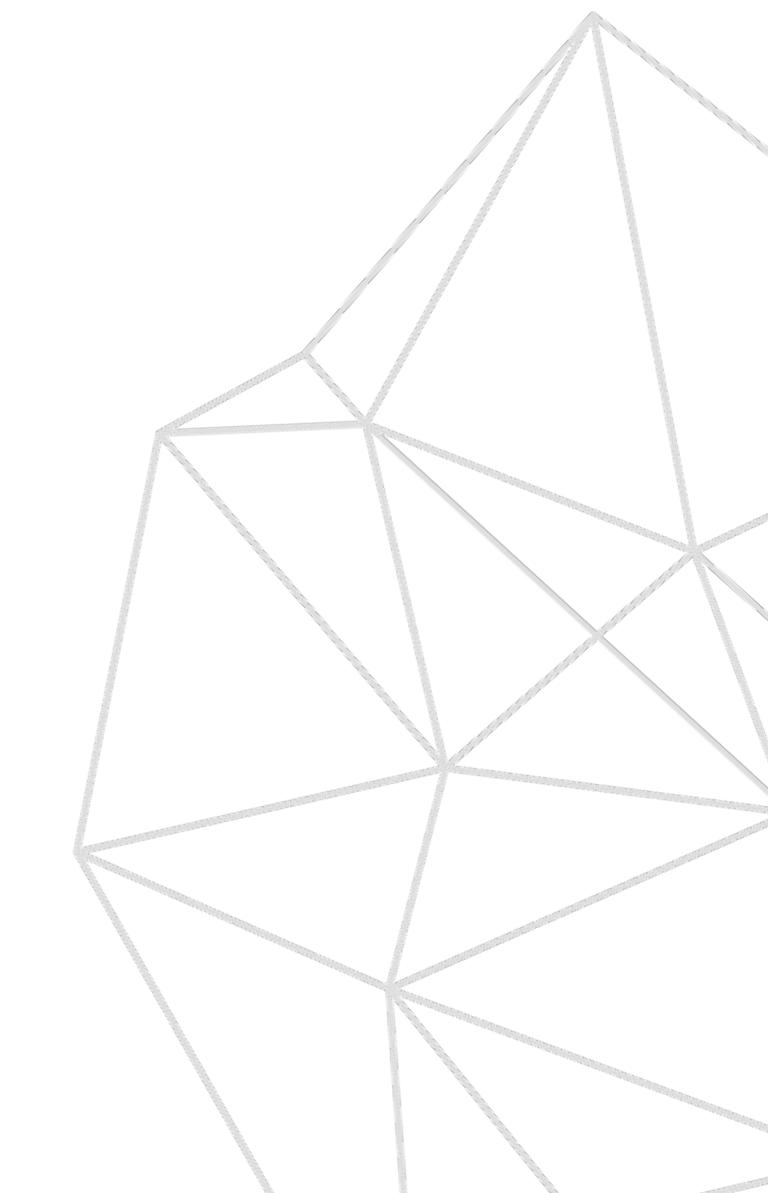
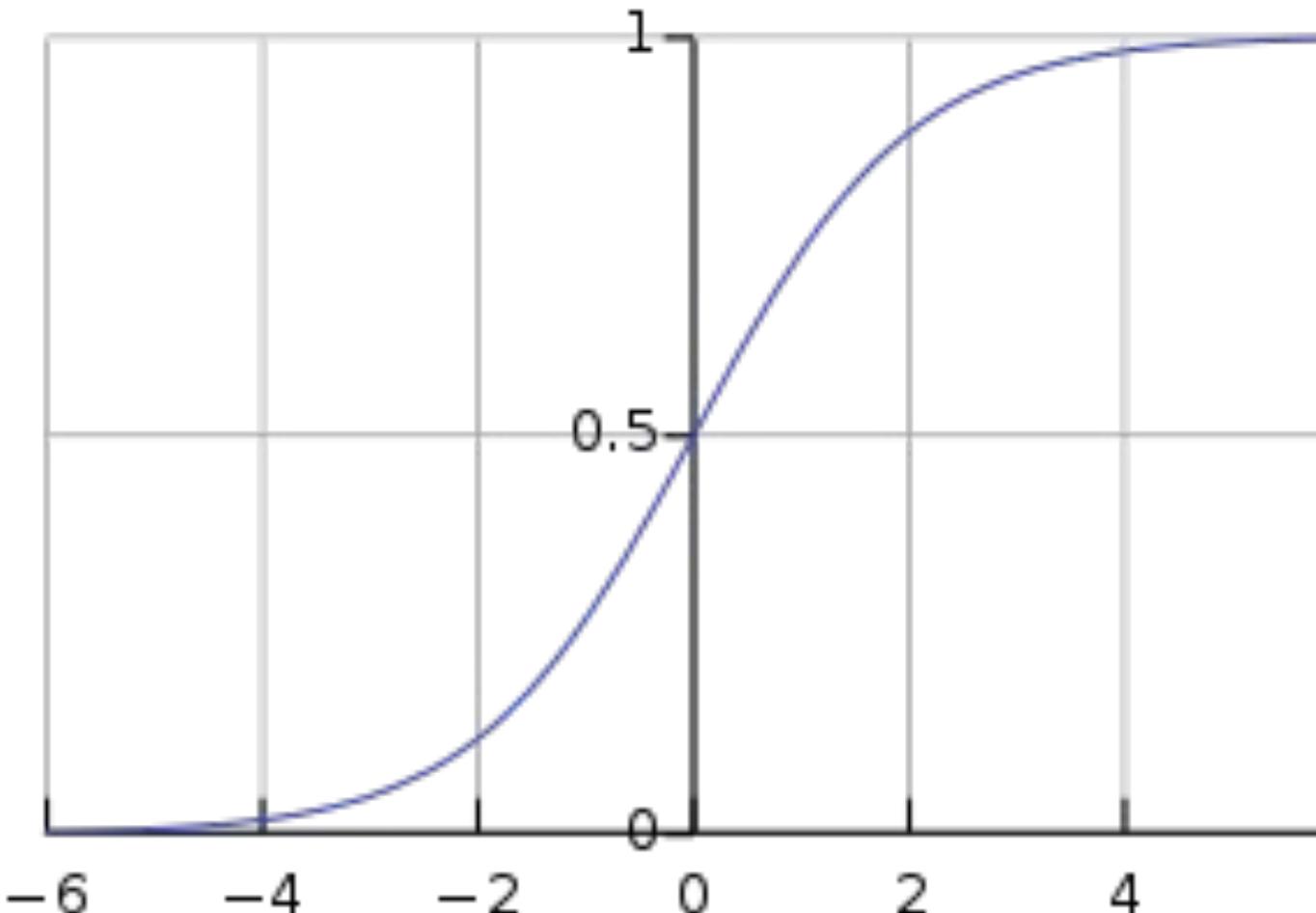




# Regresión logística: modelo

Pero en la clasificación no solo nos interesa saber a qué clase pertenece, sino **la probabilidad de pertenecer a una clase**. Para ello aplicamos la función sigmoide (o función logística):

$$h_{\theta}(X_i) = g(Z) = \frac{1}{1 + e^{-Z}} = \frac{1}{1 + e^{-(\theta_0 + \sum_{j=1}^n \theta_j x_{i,j})}} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_{i,1} + \dots + \theta_n x_{i,n})}}$$

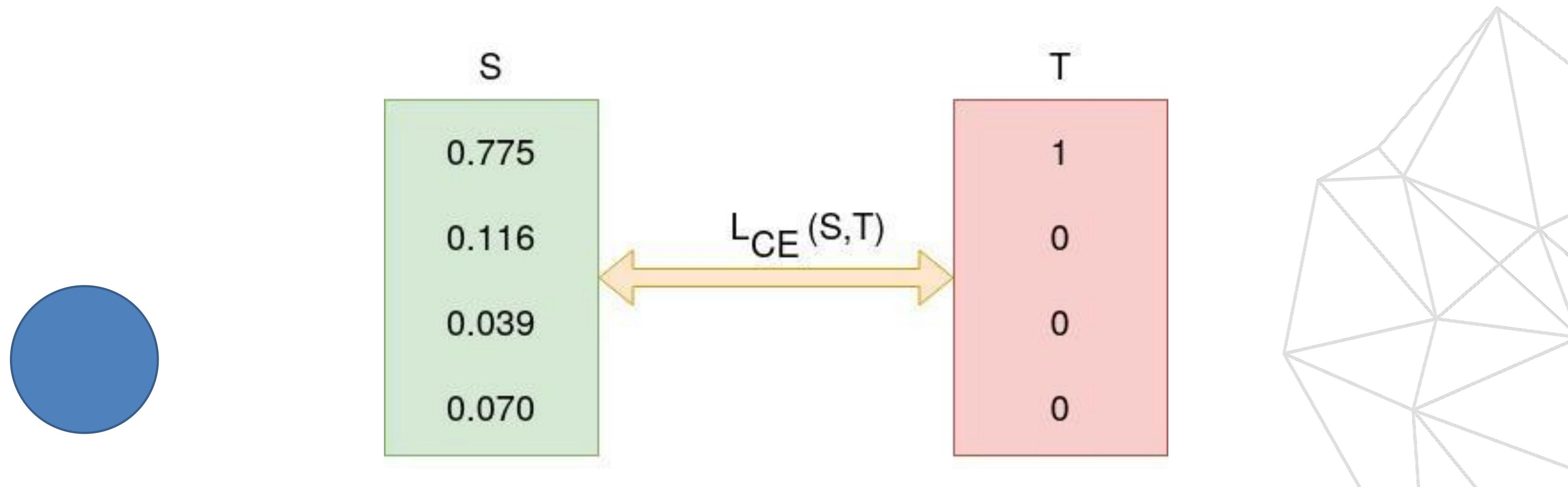




# Regresión logística: función de pérdida

La función de pérdida que debemos usar debe ser diferente a las utilizadas en regresión. Podríamos utilizar la siguiente que tiene en cuenta el error de clasificación en función de la probabilidad obtenida por nuestro modelo:

$$L_{CE}(y_i, \hat{y}_i) = -(y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i))$$





# K-NN

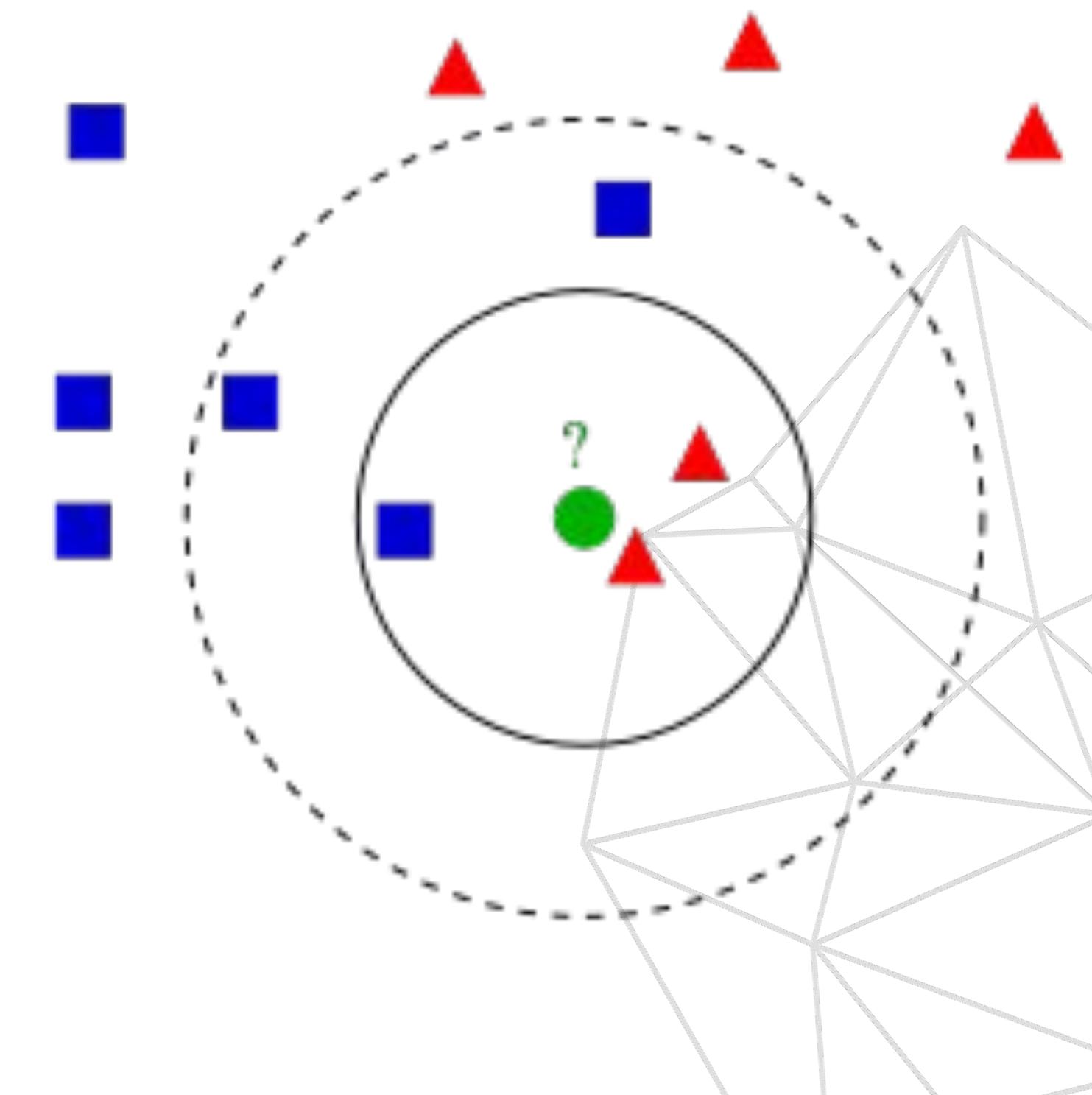
El k-NN o el método de los k vecinos más cercanos (*k nearest neighbors* en inglés) es un método de aprendizaje que asume que los elementos similares se encuentran próximo en el espacio (utilizando la función de distancia adecuada).

El k-NN consiste en los siguientes pasos:

- Elegimos el k adecuado
- Tomamos los k elementos de entrenamiento más cercanos al que queremos predecir
- La clase será la moda entre dichos elementos

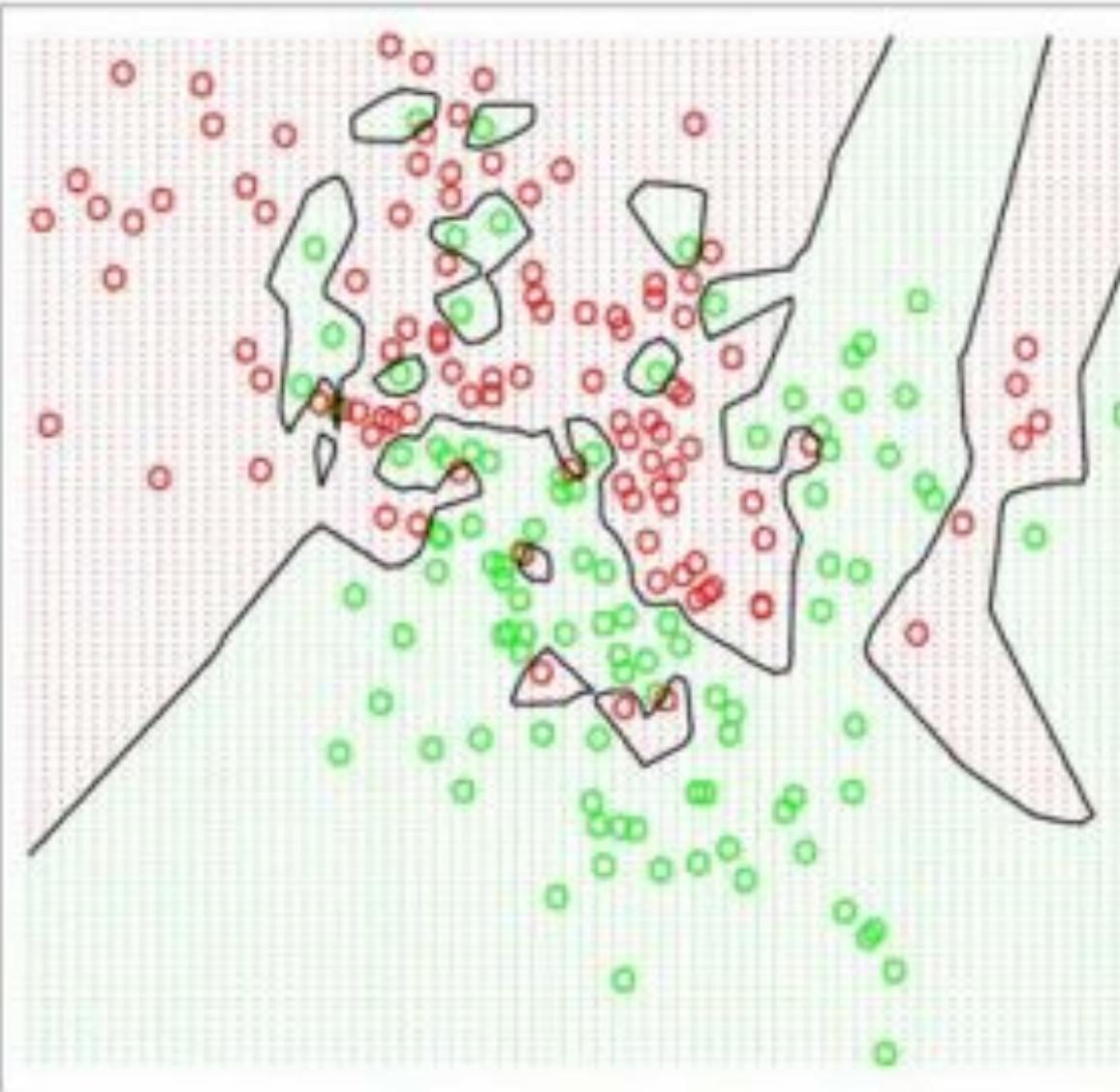
La desventaja principal del k-NN es su alto coste computacional. Sin embargo es simple y muy explicable

$$d(X_1, X_2) = \sqrt{\sum_{j=1}^n (x_{1,j} - x_{2,j})^2}$$

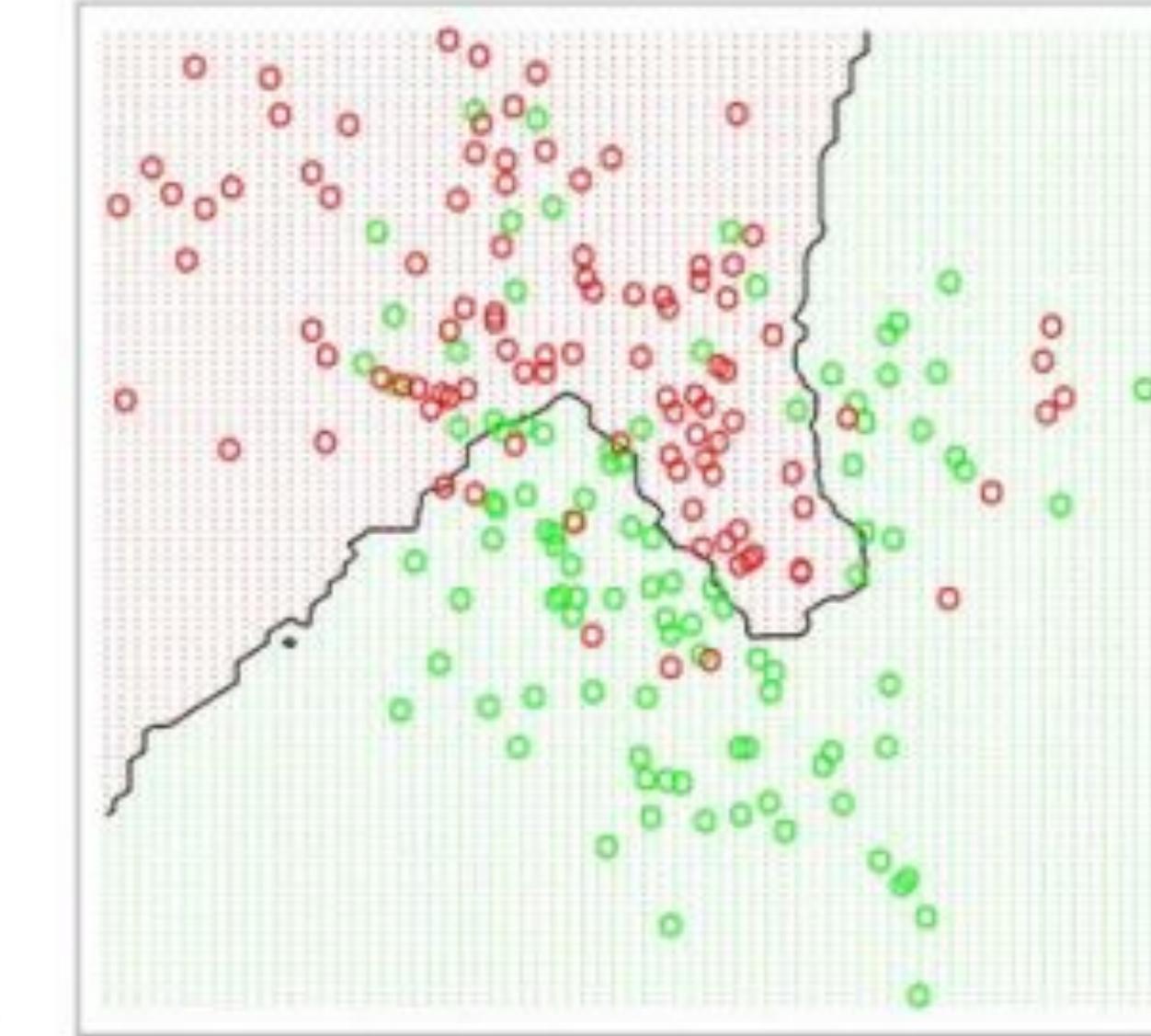




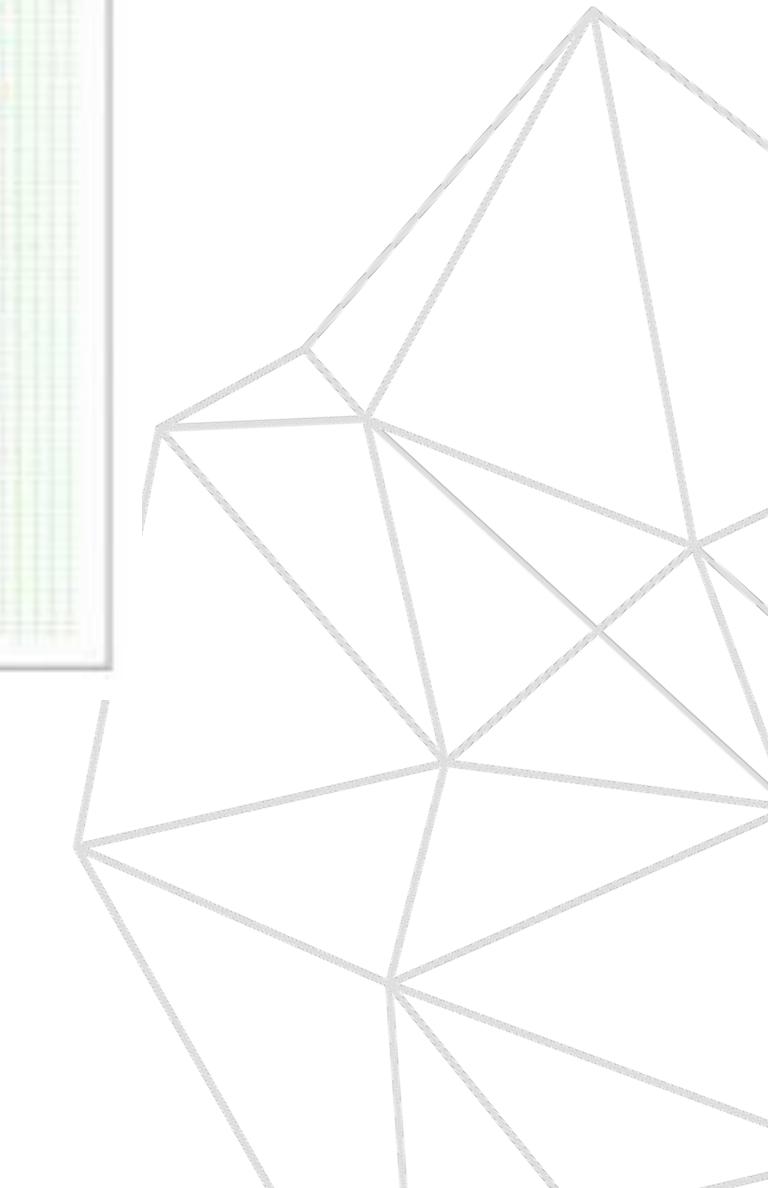
# K-NN: Elección de la k



$k = 1$



$k = 15$





# Naive Bayes: Teorema de Bayes

Supongamos que nos encontramos con el problema de clasificar un elemento con valores de entrada  $X_i$  entre las clases  $C_1, C_2, \dots, C_m$ . El teorema de Bayes establece la siguiente relación:

$$P(C_k|X_i) = \frac{P(X_i|C_k)P(C_k)}{P(X_i)}$$

Existen entonces dos estrategias para decidir la clase en la que se clasifica el elemento:

- MAP (*maximum a posteriori probability*): busca la hipótesis (la clase) que maximiza el numerador.
- ML (*maximum likelihood*): asumimos que las hipótesis tienen la misma probabilidad a priori (la misma  $P(C_k)$ ) por lo que busca la clase que maximiza  $P(X_i|C_k)$ .

Para calcular  $P(X_i|C_k)$  (que puede llegar a ser muy costoso) **se presupone que no existe relación entre las variables** y por tanto:

$$P(X_i|C_k) = P(x_{i,1}, x_{i,2}, \dots, x_{i,n}|C_k) = \prod_{j=1}^n P(x_{i,j} | C_k)$$



# Naive Bayes: modelo

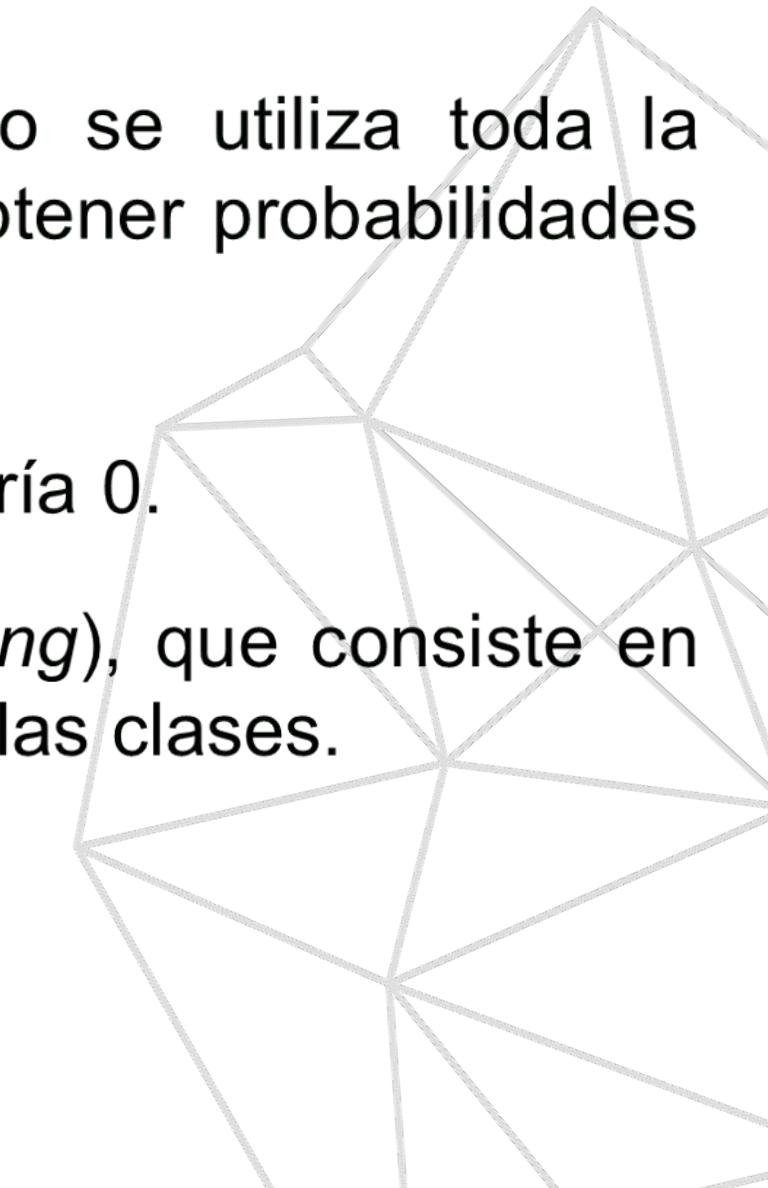
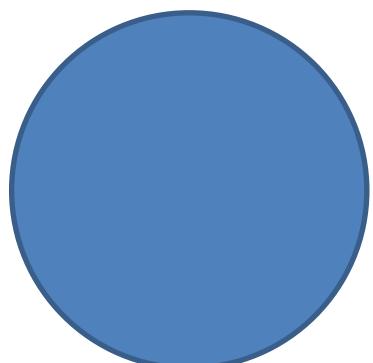
De esta forma nos queda un clasificador simplificado que será nuestro modelo Naive Bayes:

$$h_{\theta}(X_i) = \arg \max_{C_k \in C} \left( P(C_k) \prod_{j=1}^n P(x_{i,j} | C_k) \right)$$

Al obviar la información relacionada con la dependencia entre las variables, no se utiliza toda la información necesaria para los cálculos de probabilidades, pero el objetivo no es obtener probabilidades precisas, sino clasificaciones precisas.

¿Qué pasaría si para algún caso se tiene que  $P(x_{i,j} | C_k) = 0$ ? Toda la probabilidad sería 0.

La solución más común es la corrección laplaciana (*Laplacian smoothing*), que consiste en añadir una aparición a la frecuencia absoluta de todos los  $x_{i,j}$  para todas las clases.

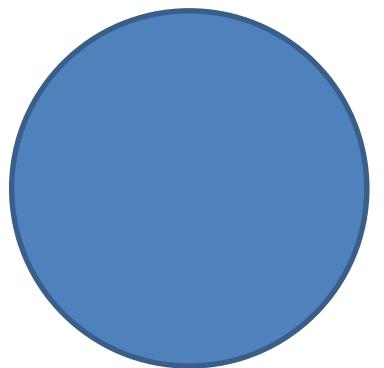




# Naive Bayes: ejemplo

| Contrato   | Solvente | Nómina  | Deudor | Concedido |
|------------|----------|---------|--------|-----------|
| Indefinido | Sí       | Elevada | No     | Sí        |
| Indefinido | Sí       | Baja    | Sí     | No        |
| Temporal   | Sí       | Baja    | Sí     | No        |
| Indefinido | No       | Media   | No     | No        |
| Interino   | Sí       | Baja    | No     | Sí        |
| Temporal   | No       | Media   | No     | No        |
| Temporal   | No       | Media   | Sí     | No        |
| Indefinido | No       | Elevada | No     | Sí        |
| Interino   | No       | Media   | Sí     | No        |
| Indefinido | Sí       | Media   | Sí     | Sí        |
| Indefinido | No       | Baja    | No     | No        |
| Temporal   | Sí       | Baja    | Sí     | No        |
| Indefinido | Sí       | Media   | No     | Sí        |
|            |          |         |        |           |
|            |          |         |        |           |
| Temporal   | Sí       | Media   | No     |           |

Queremos calcular las probabilidades:  $P(\text{Sí}|\text{Temporal}, \text{Solvente}, \text{Media}, \text{No deudor})$  y  $P(\text{No}|\text{Temporal}, \text{Solvente}, \text{Media}, \text{No deudor})$





# Naive Bayes: ejemplo

$$P(Sí|Temporal, Solvente, Media, No deudor)$$

$$P(C_k) \prod_{j=1}^n P(x_{i,j} | C_k)$$

$$P(C_k) = P(Sí) = \frac{5}{13}$$

$$P(Temporal|Sí) = \frac{0 + 1}{5 + 1} = \frac{1}{6}$$

$$P(Solvente|Sí) = \frac{4 + 1}{5 + 1} = \frac{5}{6}$$

$$P(Media|Sí) = \frac{2 + 1}{5 + 1} = \frac{3}{6} = \frac{1}{2}$$

$$P(No\ deudor|Sí) = \frac{4 + 1}{5 + 1} = \frac{5}{6}$$

| Contrato   | Solvente | Nómina  | Deudor | Concedido |
|------------|----------|---------|--------|-----------|
| Indefinido | Sí       | Elevada | No     | Sí        |
| Indefinido | Sí       | Baja    | Sí     | No        |
| Temporal   | Sí       | Baja    | Sí     | No        |
| Indefinido | No       | Media   | No     | No        |
| Interino   | Sí       | Baja    | No     | Sí        |
| Temporal   | No       | Media   | No     | No        |
| Temporal   | No       | Media   | Sí     | No        |
| Indefinido | No       | Elevada | No     | Sí        |
| Interino   | No       | Media   | Sí     | No        |
| Indefinido | Sí       | Media   | Sí     | Sí        |
| Indefinido | No       | Baja    | No     | No        |
| Temporal   | Sí       | Baja    | Sí     | No        |
| Indefinido | Sí       | Media   | No     | Sí        |

$$P(C_k) \prod_{j=1}^n P(x_{i,j} | C_k) = \frac{5}{13} \cdot \frac{1}{6} \cdot \frac{5}{6} \cdot \frac{1}{2} \cdot \frac{5}{6} = \frac{125}{5616} \approx 0,0223$$



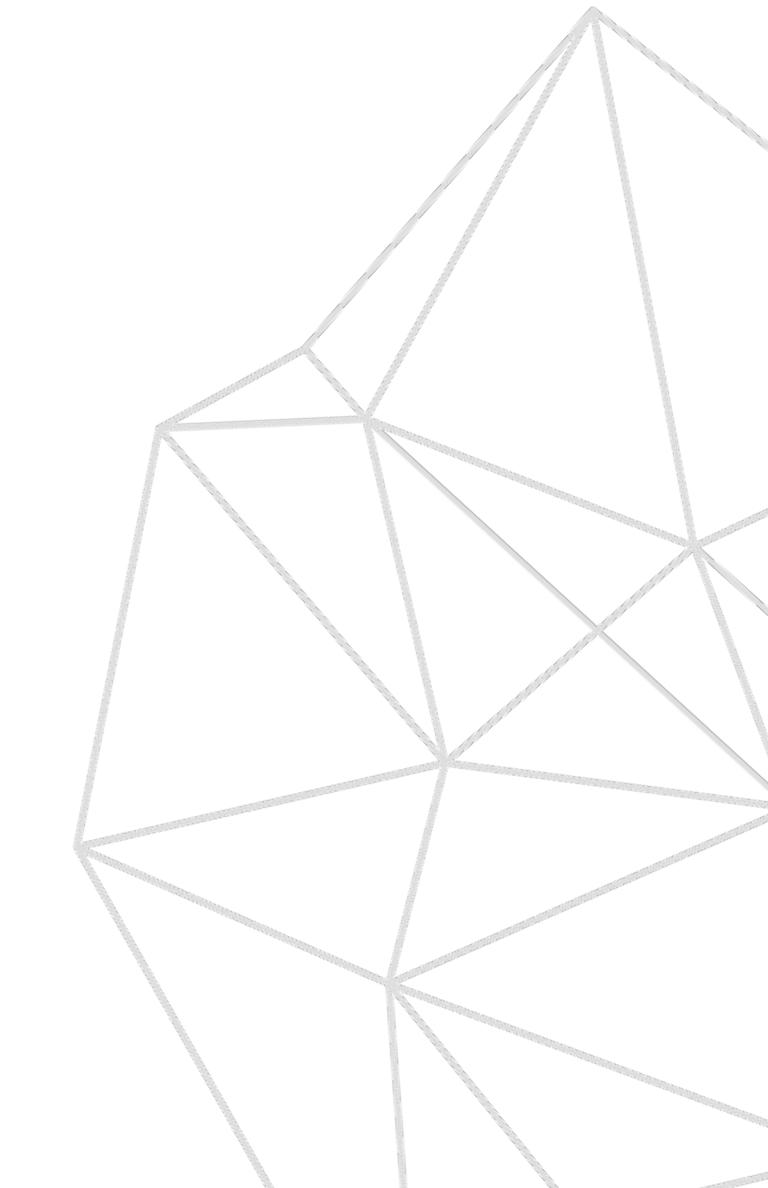
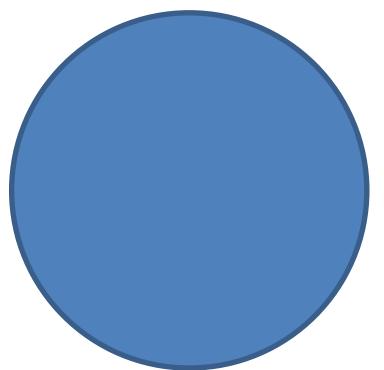
# Naive Bayes: ejemplo

$$P(\text{Sí} | \text{Temporal, Solvente, Media, No deudor}) \approx 0,0223$$

$$P(\text{No} | \text{Temporal, Solvente, Media, No deudor}) \approx 0,0216$$

$$h_{\theta}(X_i) = \arg \max_{C_k \in C} \left( P(C_k) \prod_{j=1}^n P(x_{i,j} | C_k) \right)$$

| Contrato   | Solvente | Nómina  | Deudor | Concedido |
|------------|----------|---------|--------|-----------|
| Indefinido | Sí       | Elevada | No     | Sí        |
| Indefinido | Sí       | Baja    | Sí     | No        |
| Temporal   | Sí       | Baja    | Sí     | No        |
| Indefinido | No       | Media   | No     | No        |
| Interino   | Sí       | Baja    | No     | Sí        |
| Temporal   | No       | Media   | No     | No        |
| Temporal   | No       | Media   | Sí     | No        |
| Indefinido | No       | Elevada | No     | Sí        |
| Interino   | No       | Media   | Sí     | No        |
| Indefinido | Sí       | Media   | Sí     | Sí        |
| Indefinido | No       | Baja    | No     | No        |
| Temporal   | Sí       | Baja    | Sí     | No        |
| Indefinido | Sí       | Media   | No     | Sí        |





# Naive Bayes: cálculo de la probabilidad condicionada

Como hemos visto en el ejemplo los resultados del numerador del Teorema de Bayes son valores bajos. Conforme aumente el número de variables los valores serán más pequeños. Una solución es modificar ligeramente el clasificador (recordemos que no nos interesa obtener probabilidades precisas, sino clasificar correctamente):

$$h_{\theta}(X_i) = \arg \max_{C_k \in C} - \left| P(C_k) \prod_{j=1}^n \ln(P(x_{i,j} | C_k)) \right|$$

La pregunta ahora es: ¿Cómo se calcula  $P(x_{i,j} | C_k)$  cuando la **variable es continua** (como por ejemplo que tienes el dato de la nómina exacta) y no categórica como ocurre en el ejemplo? Para cada tipo de variable existen formas de calcular la probabilidad:

- **Naive Bayes Bernoulli:** se utiliza para las variables booleanas.
- **Naive Bayes Categórico:** se utiliza para las variables categóricas múltiples (no binarias).
- **Naive Bayes Multinomial:** se utiliza para las variables que siguen una distribución multinomial.
- **Naive Bayes Gaussiano:** se utiliza para las variables continuas asumiendo que siguen una distribución normal.



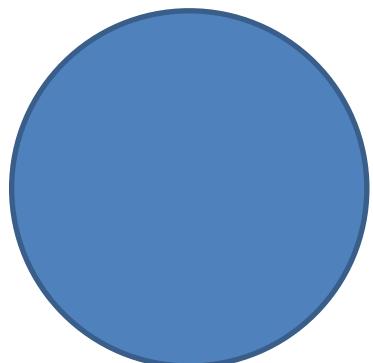
# Naive Bayes: probabilidad para variables continuas

El Naive Bayes Gaussiano asume que siguen una distribución normal. Para hallar la probabilidad dado un valor se utiliza la ecuación de la distribución normal:

$$P(x_{i,j} | C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x_{i,j}-\mu_k)^2}{2\sigma_k^2}}$$

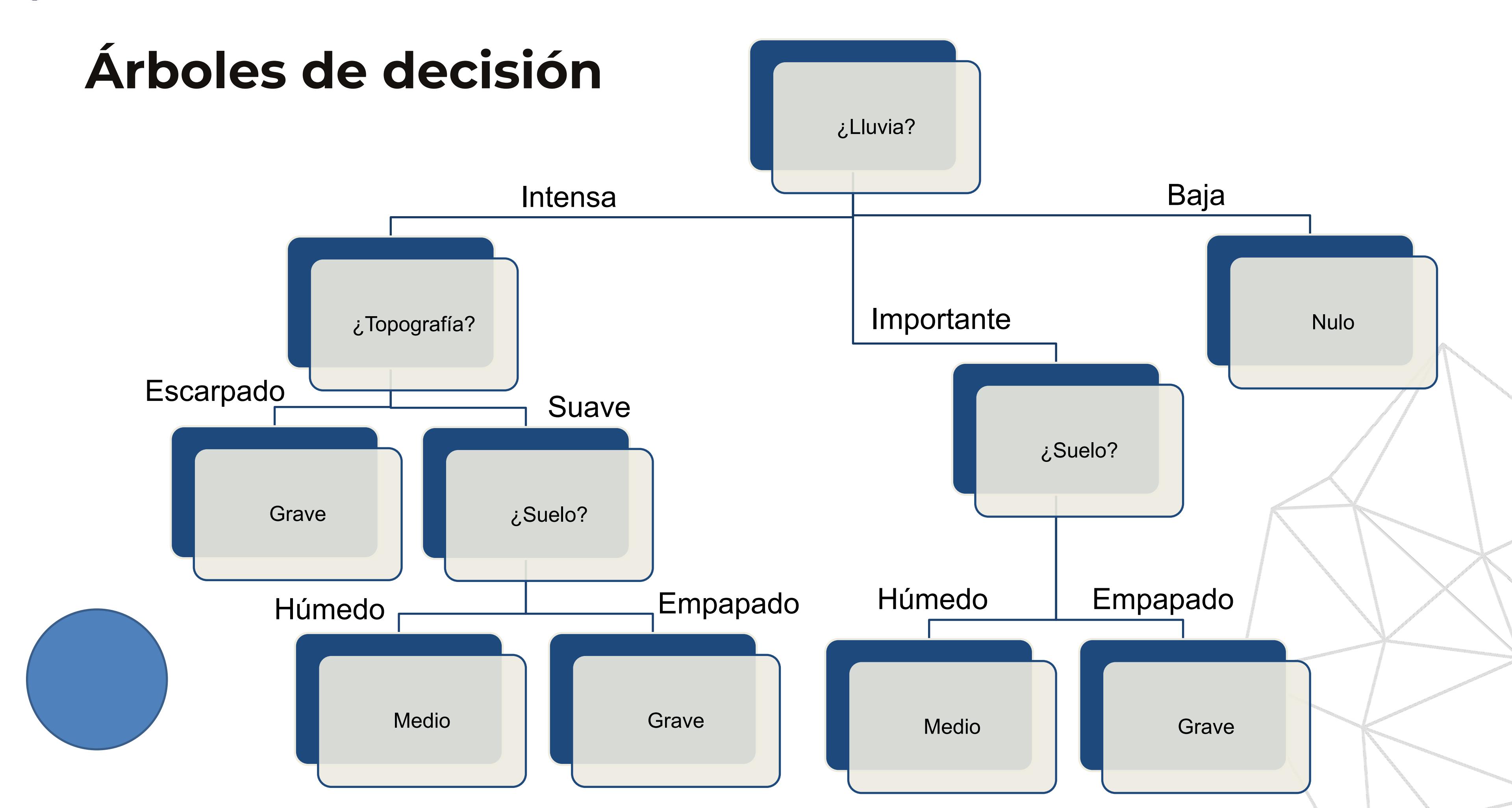
Donde  $\mu_k$  es la media de la columna j asociada a la clase k y  $\sigma_k^2$  es la cuasivarianza de la columna j asociada a la clase k.

Otra técnica común cuando nos encontramos con variables continuas consiste en discretizarlas con técnicas como el agrupamiento (o *binning*) obteniendo un nuevo conjunto de variables de distribución de Bernoulli. Naive Bayes Gaussiano suele ser una mejor opción con pocos datos de entrenamiento, mientras que la discretización cuando hay una gran cantidad de datos.





# Árboles de decisión





# Árboles de decisión: ID3

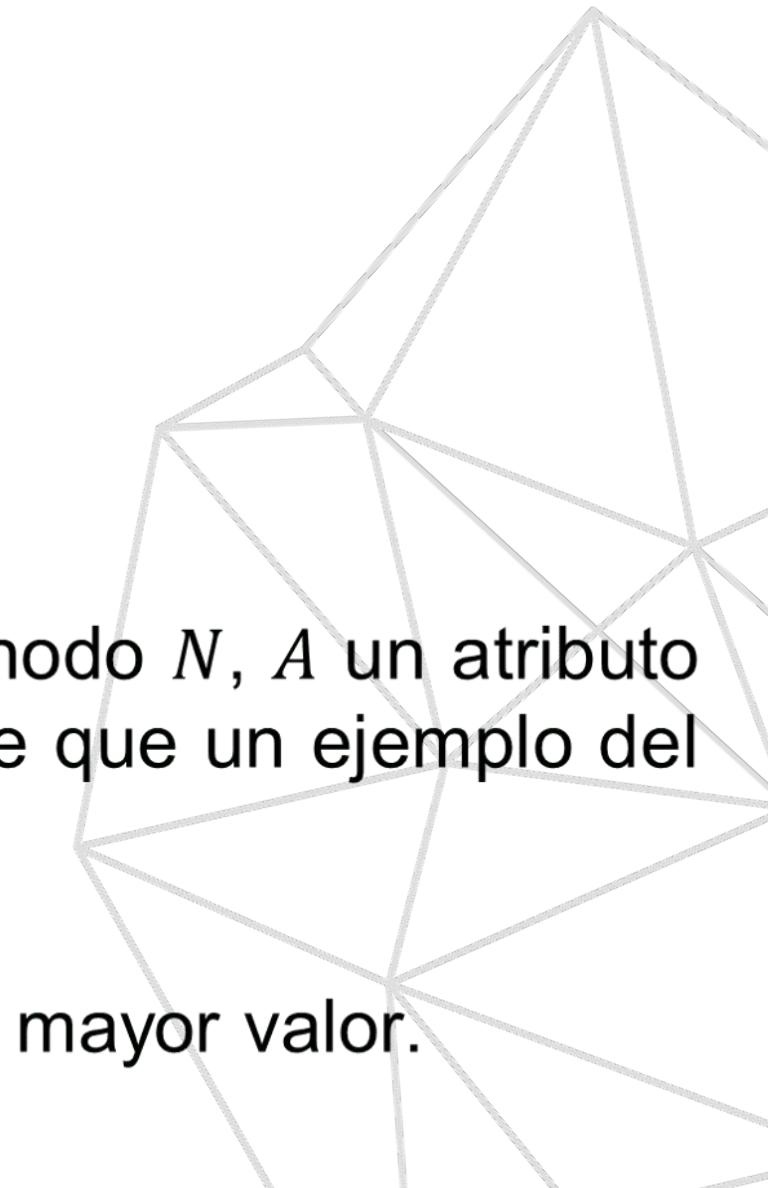
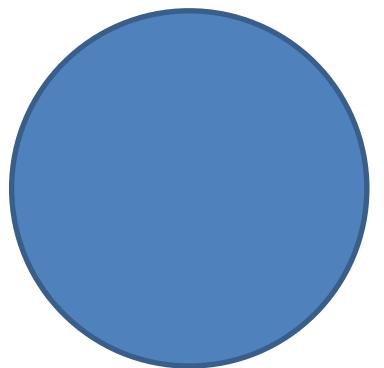
Método de selección de variables que se basa en la disminución de entropía (o ganancia de información):

$$E(N) = - \sum_{k=1}^m P(C_k) \log_2(P(C_k))$$

$$E(N|A) = \sum_{j=1}^p P(a_j) \left( - \sum_{k=1}^m P(C_k|a_j) \log_2(P(C_k|a_j)) \right)$$

Siendo  $P(C_k)$  la probabilidad de que un ejemplo tenga la clase  $C_k$  en el nodo  $N$ ,  $A$  un atributo (o variable) que posee  $p$  valores distintos  $a_j$  y  $P(C_k|a_j)$  la probabilidad de que un ejemplo del nodo hijo con  $A = a_j$  tenga la clase  $C_k$ .

Se calcula  $E(N) - E(N|A)$  para todos los atributos y se elige el que tenga mayor valor.





# Árboles de decisión: ID3

Calculemos la raíz:

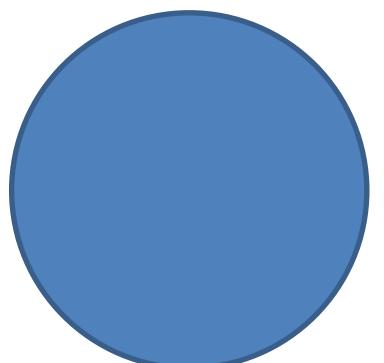
$$E(N) = - \sum_{k=1}^m P(C_k) \log_2(P(C_k))$$

$$P(\text{grave}) = 0,4 \quad P(\text{medio}) = 0,3 \quad P(\text{nulo}) = 0,3$$

$$E(\text{raiz}) = -0,4 \log_2(0,4) - 0,3 \log_2(0,3) - 0,3 \log_2(0,3)$$

$$E(\text{raiz}) = 1,571$$

| Lluvia     | Suelo    | Topografía | Riesgo |
|------------|----------|------------|--------|
| intensa    | empapado | escarpada  | grave  |
| intensa    | empapado | suave      | grave  |
| intensa    | húmedo   | escarpada  | grave  |
| intensa    | húmedo   | suave      | medio  |
| importante | empapado | escarpada  | grave  |
| importante | húmedo   | escarpada  | medio  |
| importante | húmedo   | suave      | medio  |
| baja       | empapado | escarpada  | nulo   |
| baja       | húmedo   | escarpada  | nulo   |
| baja       | húmedo   | suave      | nulo   |





# Árboles de decisión: ID3

$$E(\text{raiz}) = 1,571$$

$$E(N|A) = \sum_{j=1}^p P(a_j) \left( - \sum_{k=1}^m P(C_k|a_j) \log_2(P(C_k|a_j)) \right)$$

Empezamos clasificando según lluvia (A):

$$P(\text{grave|intensa}) = 0,75 \quad P(\text{medio|intensa}) = 0,25$$

$$P(\text{grave|importante}) = 1/3 \quad P(\text{medio|importante}) = 2/3$$

$$P(\text{nulo|baja}) = 1$$

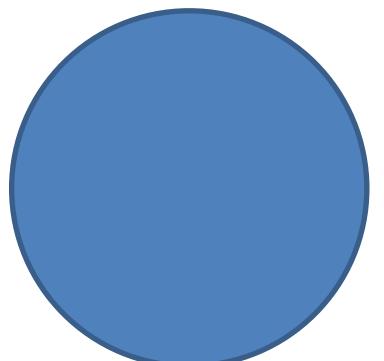
$$E(\text{raiz|intensa}) = -0,75 \log_2(0,75) - 0,25 \log_2(0,25) = 0,811$$

$$E(\text{raiz|importante}) = 0,918$$

$$E(\text{raiz|baja}) = 0$$

$$E(\text{raiz|lluvia}) = 0,4 \cdot 0,811 + 0,3 \cdot 0,918 + 0,3 \cdot 0 = 0,6$$

| Lluvia     | Suelo    | Topografía | Riesgo |
|------------|----------|------------|--------|
| intensa    | empapado | escarpada  | grave  |
| intensa    | empapado | suave      | grave  |
| intensa    | húmedo   | escarpada  | grave  |
| intensa    | húmedo   | suave      | medio  |
| importante | empapado | escarpada  | grave  |
| importante | húmedo   | escarpada  | medio  |
| importante | húmedo   | suave      | medio  |
| baja       | empapado | escarpada  | nulo   |
| baja       | húmedo   | escarpada  | nulo   |
| baja       | húmedo   | suave      | nulo   |





# Árboles de decisión: ID3

$$E(\text{raiz}) = 1,571$$

$$E(\text{raiz}|\text{lluvia}) = 0,6$$

$$DE(\text{raiz}|\text{lluvia}) = 1,571 - 0,6 = 0,971$$

Si hacemos el mismo cálculo para el resto de variables (suelo y topografía) se tendrá:

$$DE(\text{raiz}|\text{suelo}) = 1,571 - 1,2 = 0,371$$

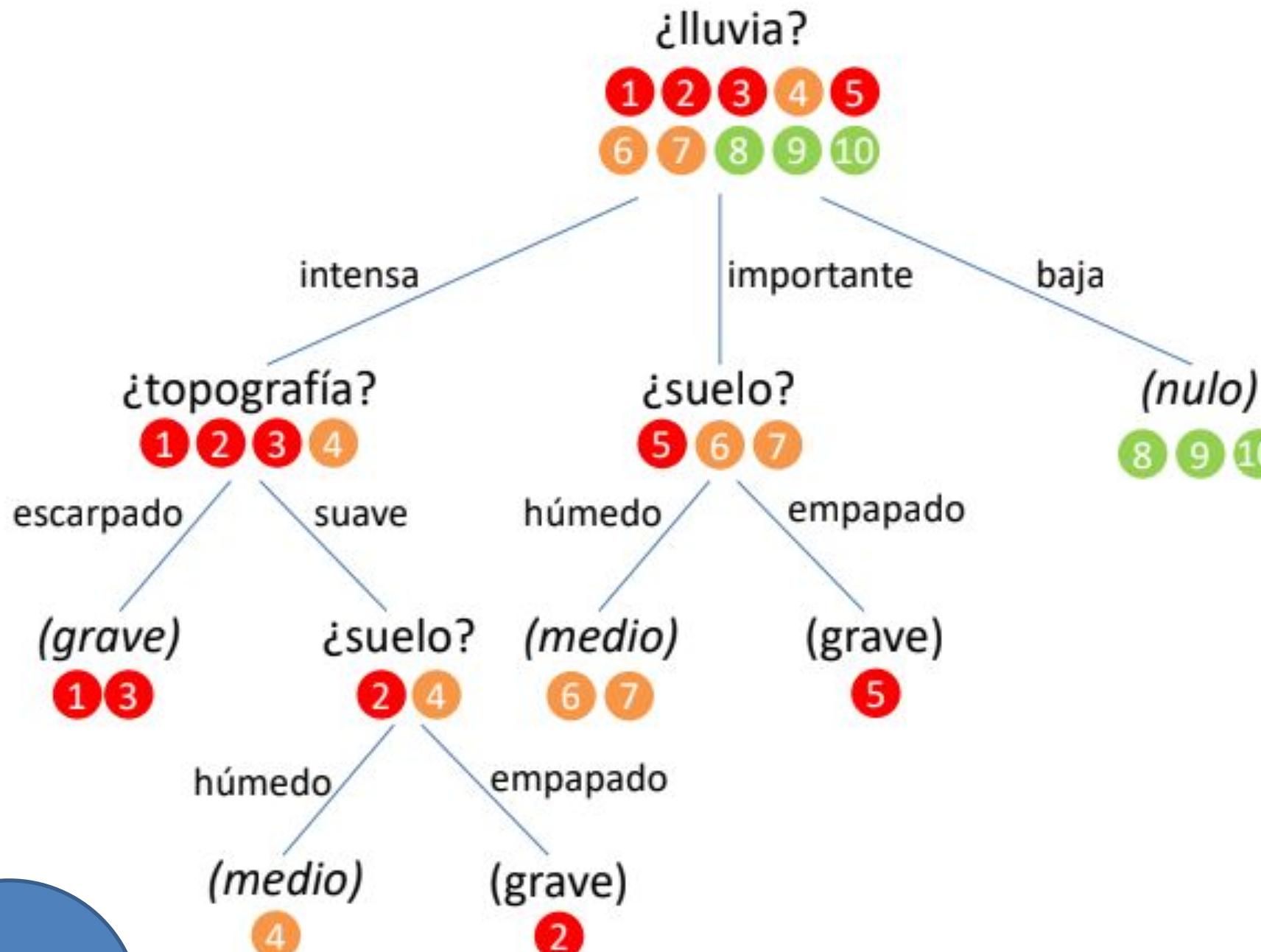
$$DE(\text{raiz}|\text{topografía}) = 1,571 - 1,475 = 0,096$$

| Lluvia     | Suelo    | Topografía | Riesgo |
|------------|----------|------------|--------|
| intensa    | empapado | escarpada  | grave  |
| intensa    | empapado | suave      | grave  |
| intensa    | húmedo   | escarpada  | grave  |
| intensa    | húmedo   | suave      | medio  |
| importante | empapado | escarpada  | grave  |
| importante | húmedo   | escarpada  | medio  |
| importante | húmedo   | suave      | medio  |
| baja       | empapado | escarpada  | nulo   |
| baja       | húmedo   | escarpada  | nulo   |
| baja       | húmedo   | suave      | nulo   |

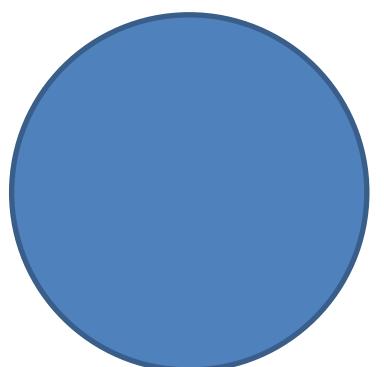
En los siguientes nodos que se formen, se vuelve a aplicar el cálculo considerando en cada uno el subconjunto de ejemplos obtenido y eliminando lluvia del conjunto de atributos.



# Árboles de decisión: ID3



| Lluvia     | Suelo    | Topografía | Riesgo |
|------------|----------|------------|--------|
| intensa    | empapado | escarpada  | grave  |
| intensa    | empapado | suave      | grave  |
| intensa    | húmedo   | escarpada  | grave  |
| intensa    | húmedo   | suave      | medio  |
| importante | empapado | escarpada  | grave  |
| importante | húmedo   | escarpada  | medio  |
| importante | húmedo   | suave      | medio  |
| baja       | empapado | escarpada  | nulo   |
| baja       | húmedo   | escarpada  | nulo   |
| baja       | húmedo   | suave      | nulo   |





# Árboles de decisión: C4.5

Es otro método de selección de variables para construir el árbol que se basa en el ratio de ganancia de información:

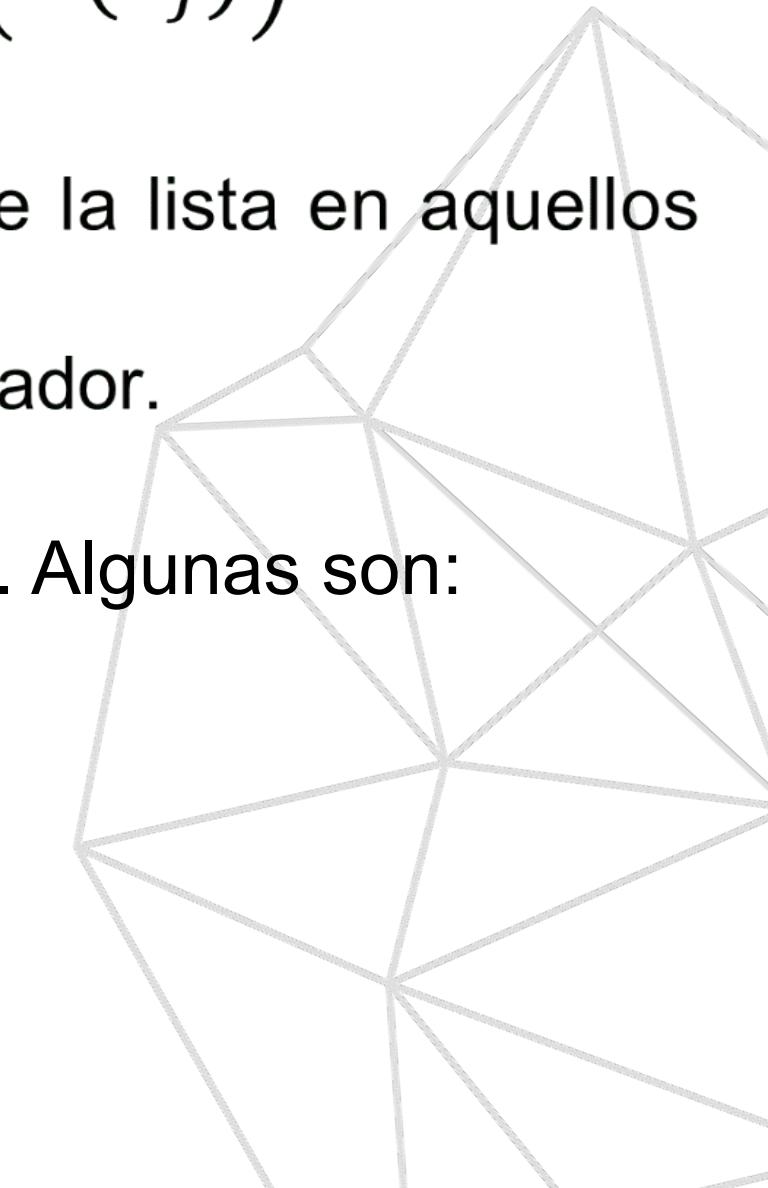
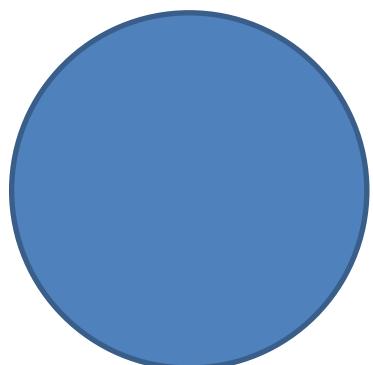
$$RG(N|A) = \frac{E(N) - E(N|A)}{-\sum_{j=1}^p P(a_j) \log_2(P(a_j))} = \frac{DE(N|A)}{-\sum_{j=1}^p P(a_j) \log_2(P(a_j))}$$

Para tratar con variables continuas, C4.5 crea un umbral (o varios) y luego se divide la lista en aquellos cuyo valor del atributo es superior e inferior o igual a dicho umbral.

Además C4.5 puede trabajar con *missing values* sin usarlos en los cálculos del numerador.

Existe una mejora a C4.5 que es el C5.0 que ofrece una serie de mejoras. Algunas son:

- Mayor velocidad
- Uso más eficiente de memoria
- Árboles de decisión más pequeños
- Soporte para boosting





# Árboles de decisión: CART

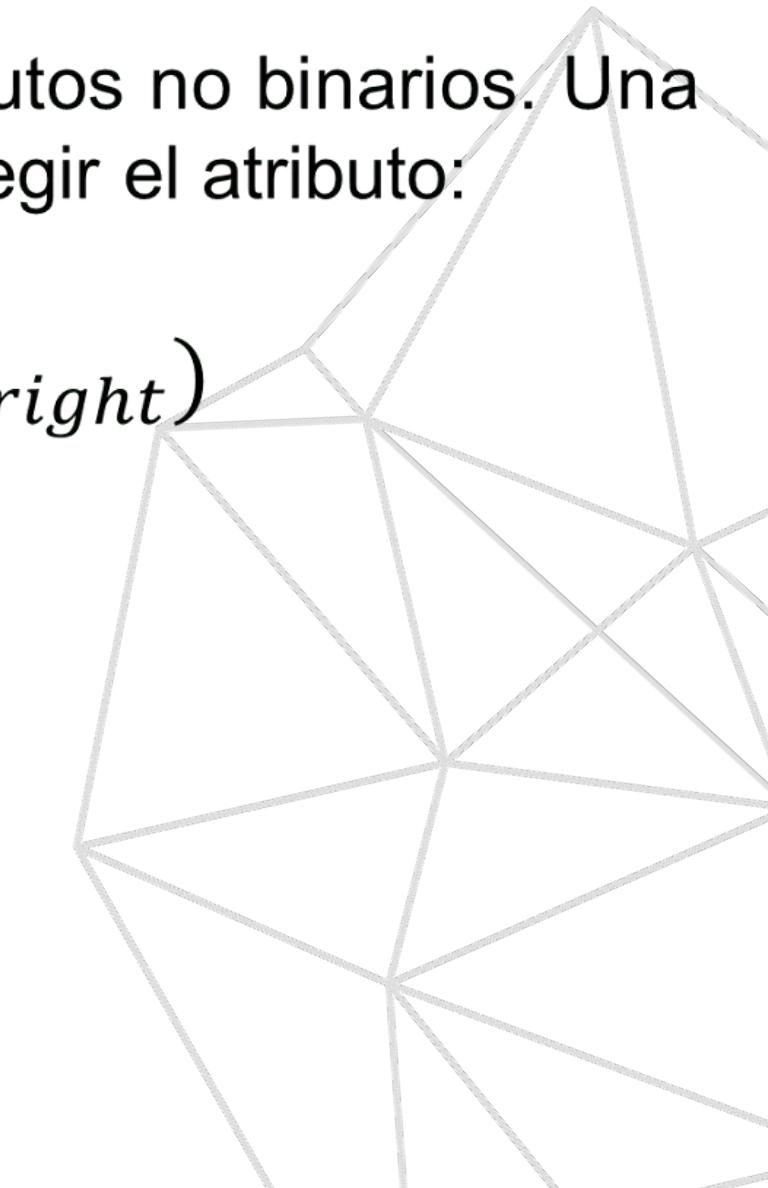
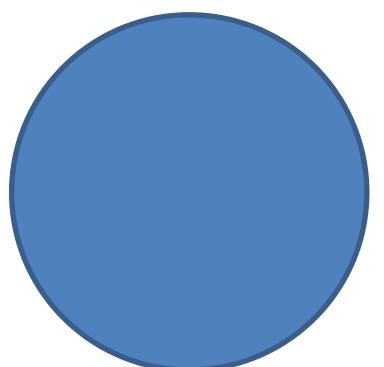
Es otro método de selección de variables para construir el árbol que se basa en el índice de Gini:

$$GI(N) = 1 - \sum_{k=1}^m P(C_k)^2$$

La particularidad de CART es que solo crea árboles binarios, transformando los atributos no binarios. Una vez son todos binarios, se calcula la reducción impureza de cada uno de ellos para elegir el atributo:

$$\Delta GI(N|A) = GI(N) - P(N_{left})GI(N_{left}) + P(N_{right})GI(N_{right})$$

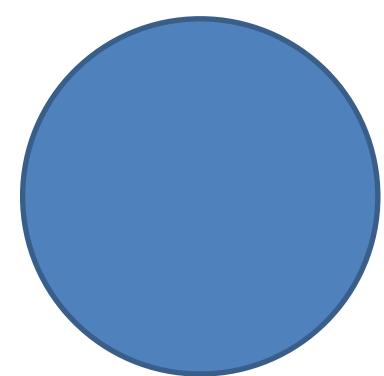
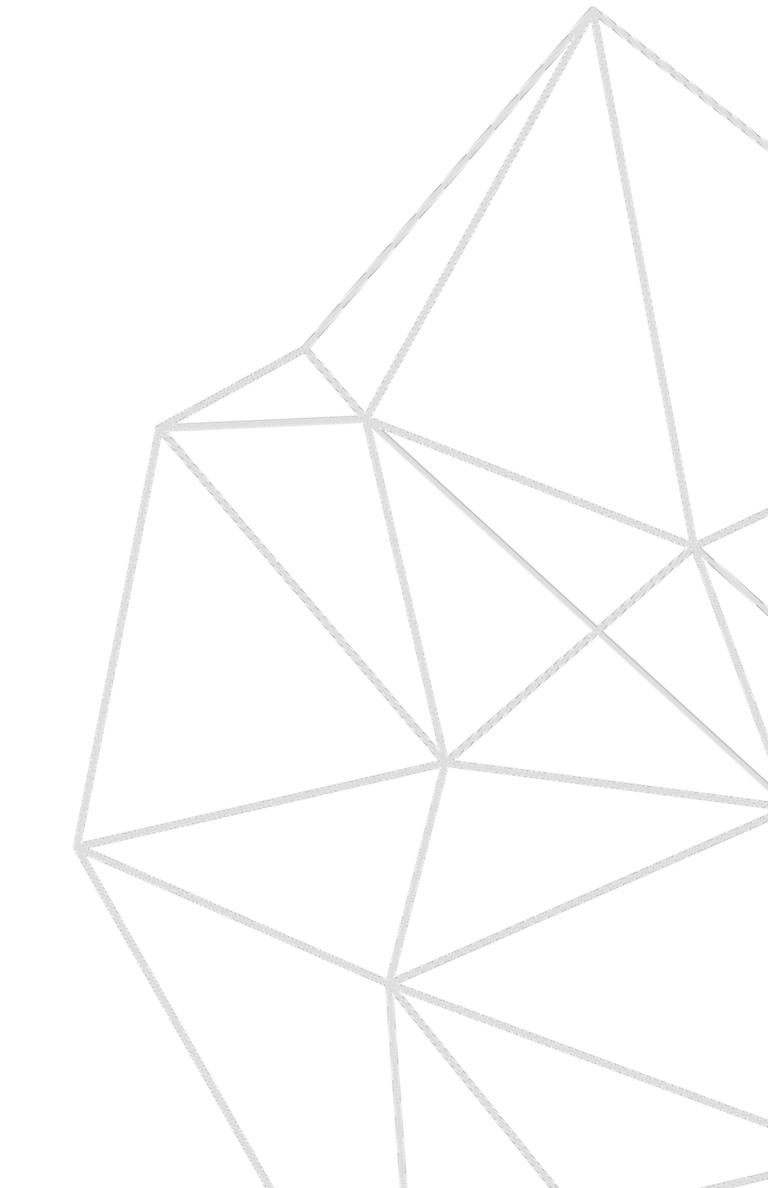
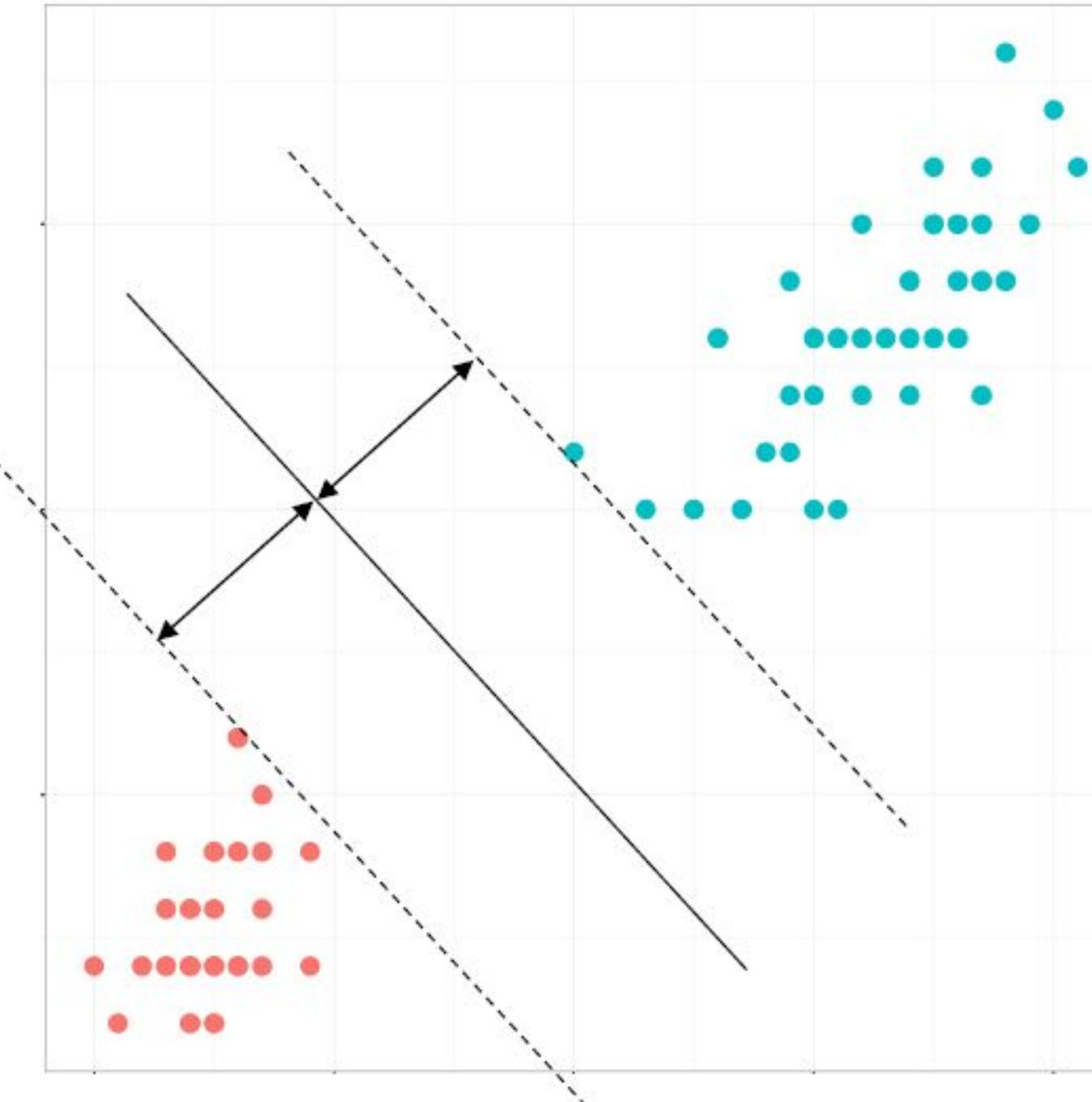
Se escoge el atributo que maximiza la reducción de impureza.





# Support Vector Machine: Hiperplano Separador

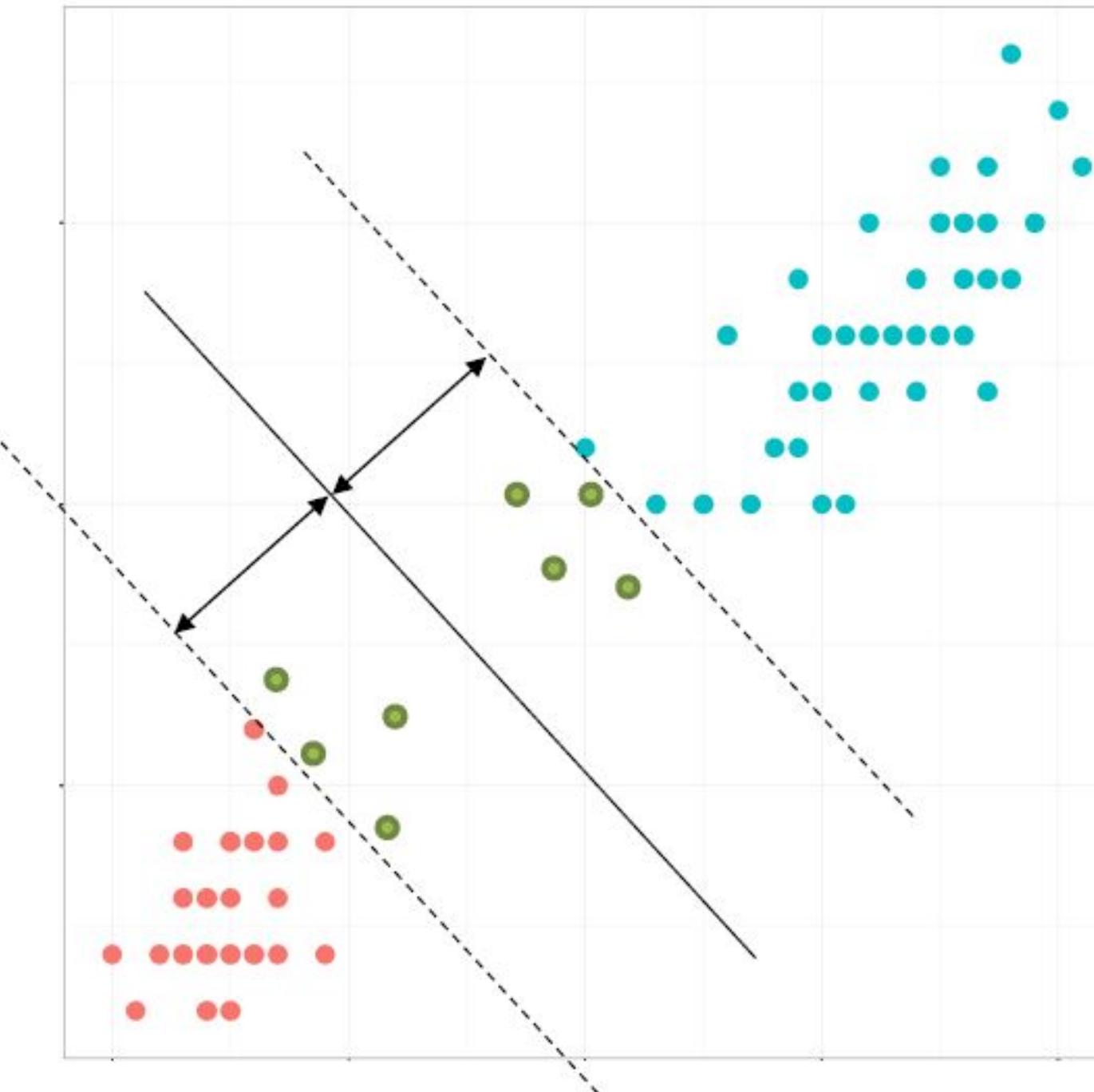
Support Vector Machine (SVM) es un clasificador diseñado para obtener un **hiperplano de máximo margen** (*maximum margin hyperplane, MMH*) que separe las instancias de dos clases diferentes maximizando los márgenes de separación a las instancias más cercanas de cada clase.





# SVM: Hiperplano de Máximo Margen

Al maximizar los márgenes, reducimos la posibilidad de clasificar erróneamente las nuevas instancias. En principio, las SVM solamente pueden clasificar problemas linealmente separables y con dos clases, pero existen técnicas para permitir resolver problemas más complejos.





# SVM: Hiperplano de Máximo Margen

El MMH viene definido por la ecuación:

$$\theta_0 + \sum_{j=1}^n \theta_j x_{i,j} = \theta_0 + \theta_1 x_{i,1} + \dots + \theta_n x_{i,n} = 0$$

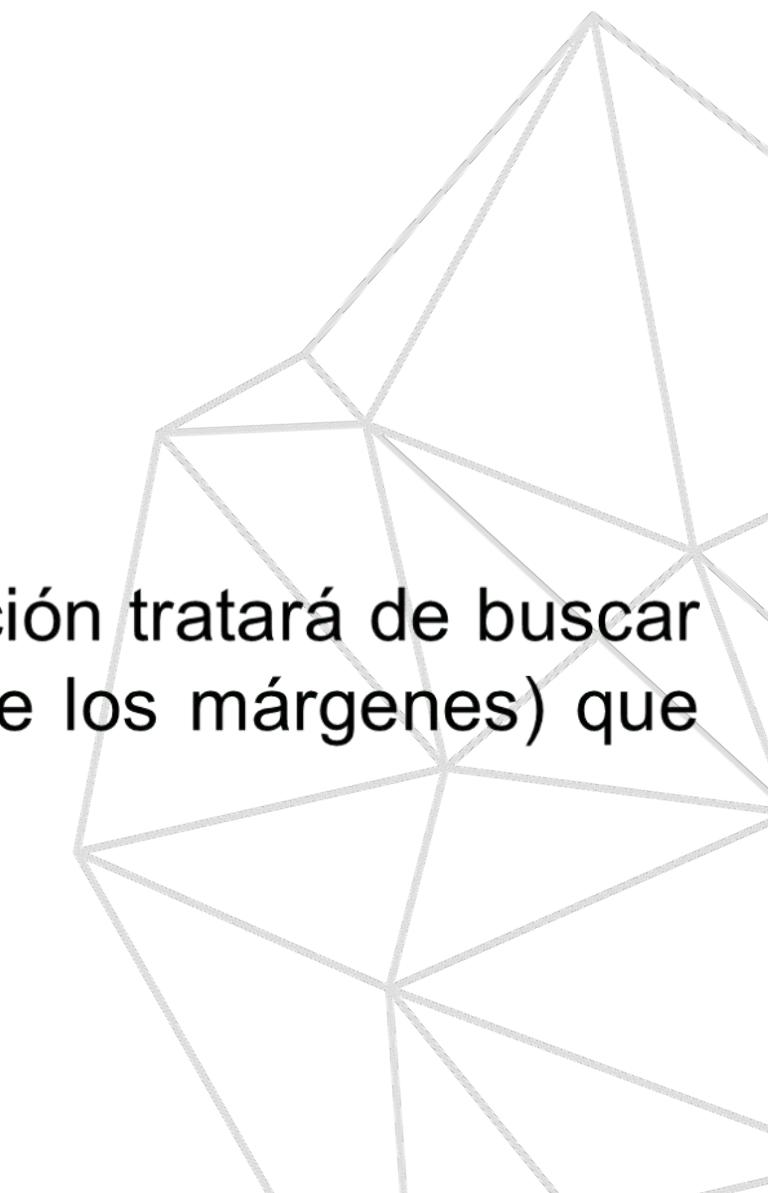
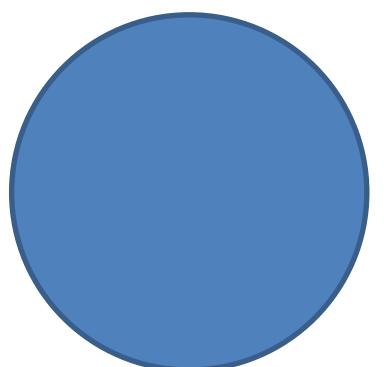
Los coeficientes del MMH pueden ser ajustados para obtener las ecuación de los márgenes  $H_1$  y  $H_2$ :

$$H_1: \theta_0 + \sum_{j=1}^n \theta_j x_{i,j} \geq 1 \text{ para } y_i = 1$$

$$H_2: \theta_0 + \sum_{j=1}^n \theta_j x_{i,j} \leq -1 \text{ para } y_i = -1$$

La distancia de los márgenes al MMH es  $1/\|\theta\|$ . El proceso de optimización tratará de buscar los vectores de soporte (elementos del conjunto de entrenamiento sobre los márgenes) que minimicen  $\|\theta\|$  sujeto a:

$$y_i \left( \theta_0 + \sum_{j=1}^n \theta_j x_{i,j} \right) \geq 1$$





# SVM: Optimización

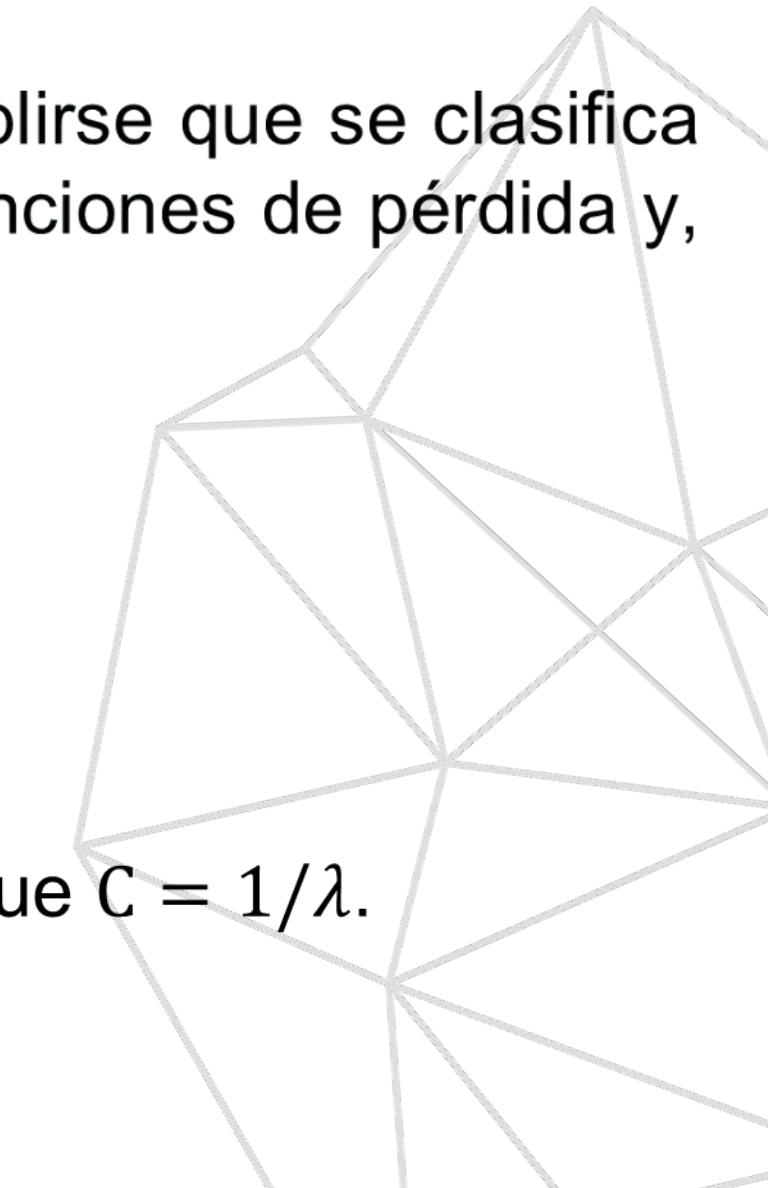
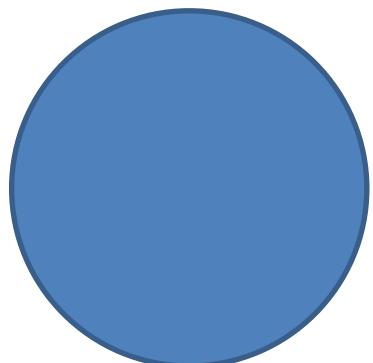
Para lograr optimizar el modelo, se hará uso de la función de pérdida *hinge*:

$$\max \left( 0,1 - y_i \left( \theta_0 + \sum_{j=1}^n \theta_j x_{i,j} \right) \right)$$

La función será 0 cuando se cumpla la restricción. Para lograr que, además de cumplirse que se clasifica correctamente, se minimiza  $\|\theta\|$ , se elegirá como función de coste la media de las funciones de pérdida y, como penalización puede tomarse  $\|\theta\|^2$  quedando la función objetivo:

$$J(h_\theta, X, y) = \lambda \|\theta\|^2 + \frac{1}{m} \sum_{i=1}^m \max \left( 0,1 - y_i \left( \theta_0 + \sum_{j=1}^n \theta_j x_{i,j} \right) \right)$$

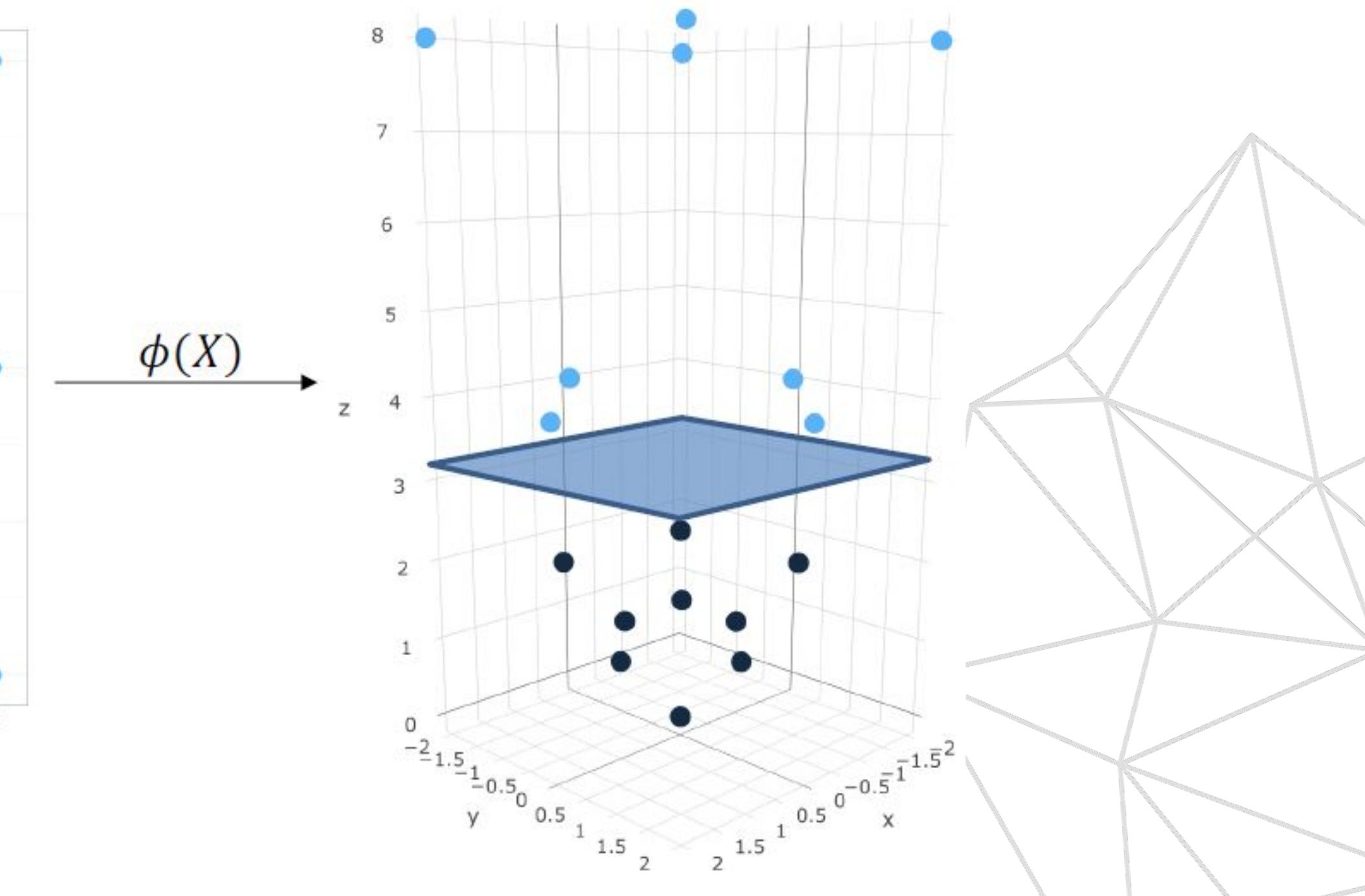
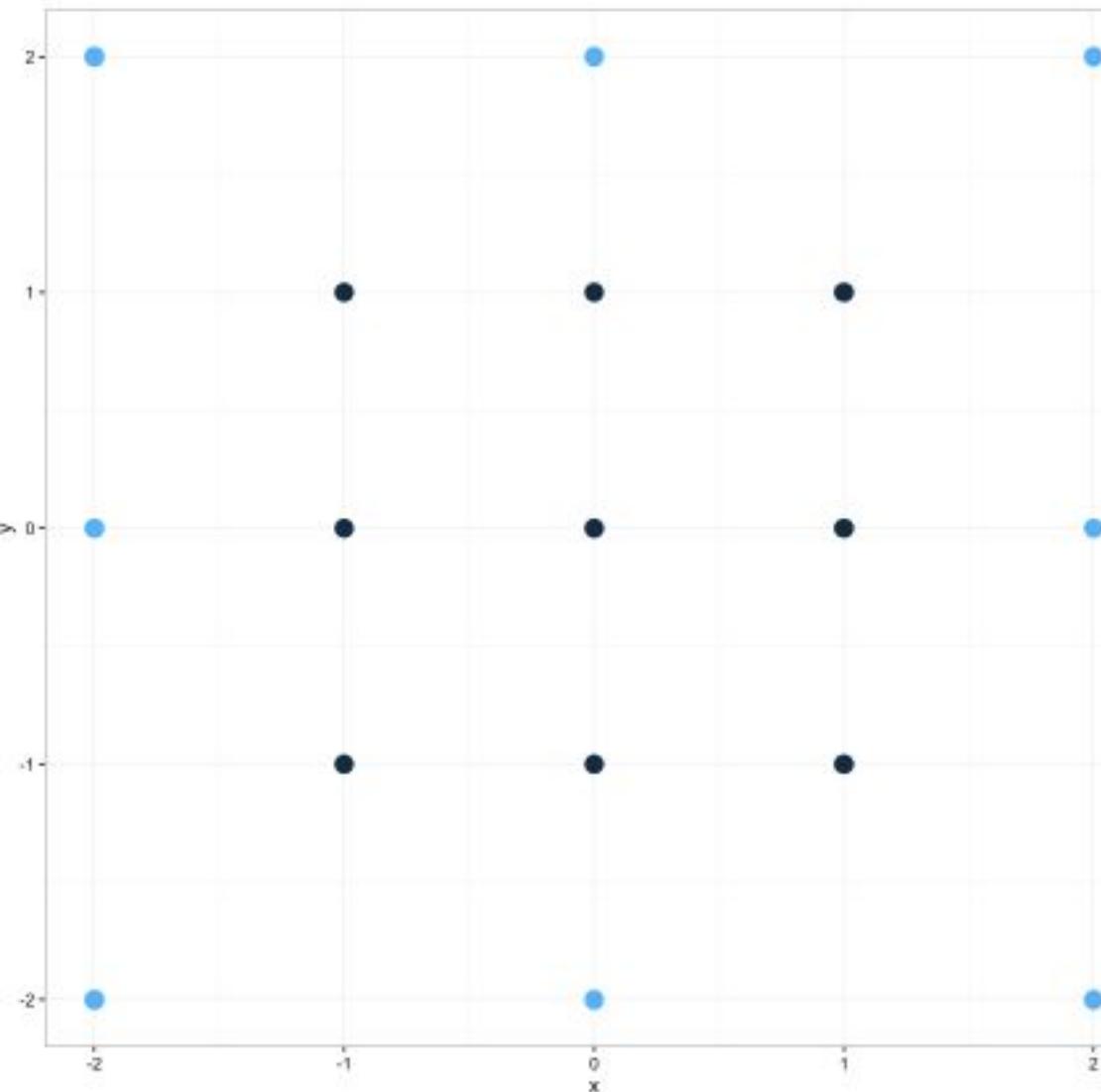
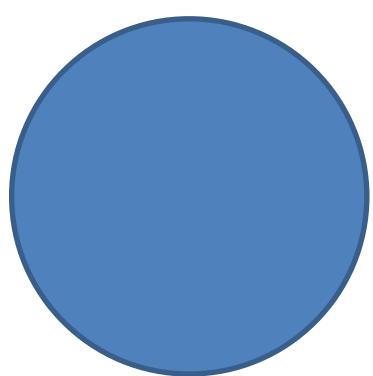
En ocasiones puede encontrarse el parámetro  $\lambda$  como  $C$  donde se tendrá que  $C = 1/\lambda$ .





# SVM: Problemas linealmente no separables

El conjunto de datos de un problema linealmente no separable puede ser transformado en otro conjunto de datos equivalente de dimensionalidad mayor que permita utilizar técnicas de clasificación para problemas linealmente separables.



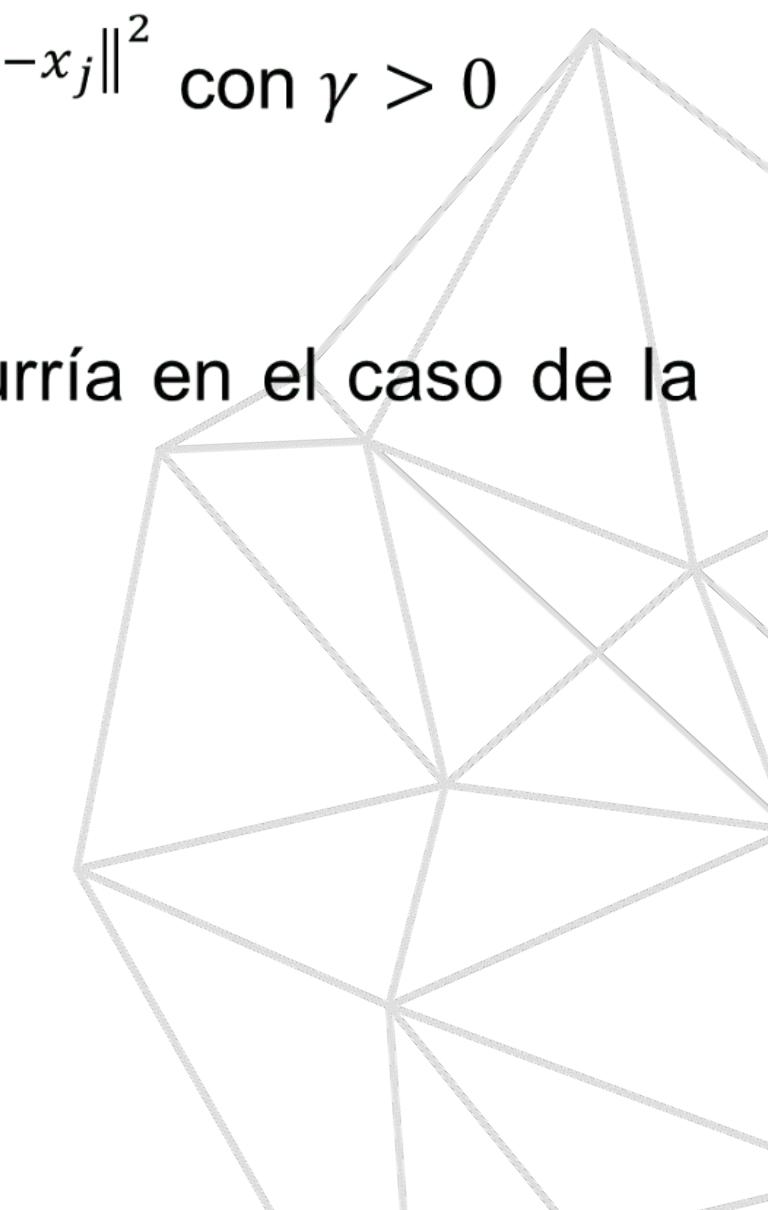
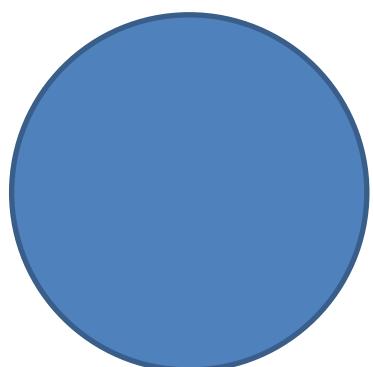


# SVM: Problemas linealmente no separables

Para conseguirlo, en lugar de aplicar una función a cada punto del espacio, reescribimos el producto escalar como una función que se llama *kernel*:

- Polinómico de grado d homogéneo:  $k(x_i, x_j) = (x_i \cdot x_j)^d$
- Polinómico de grado d no homogéneo:  $k(x_i, x_j) = (x_i \cdot x_j + r)^d$
- RBF (*radial basis function* o de función de base radial) gaussiano:  $k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$  con  $\gamma > 0$
- Sigmoidal:  $k(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c)$

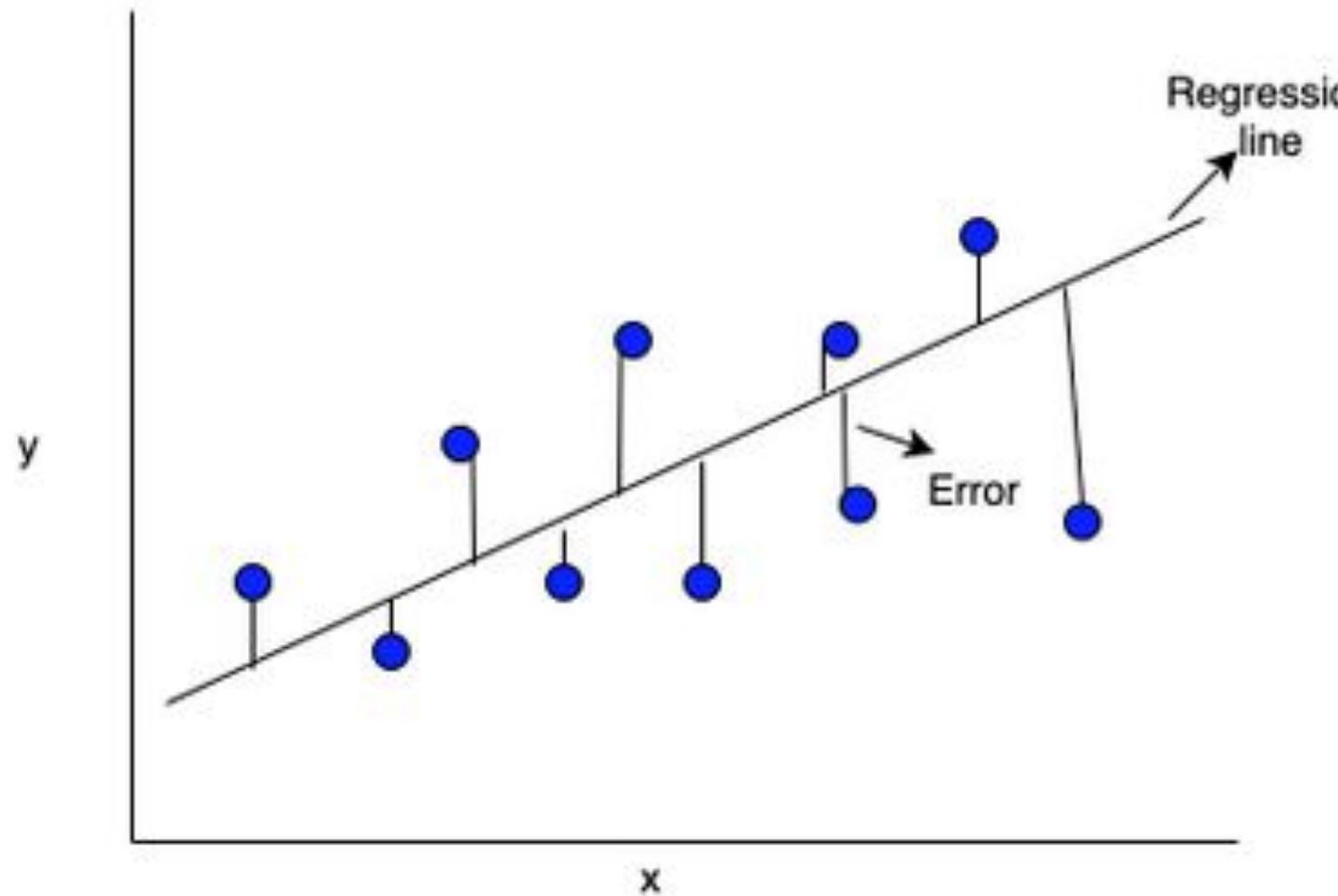
Para los problemas multiclase se pueden utilizar las técnicas OVR o OVO como ocurría en el caso de la regresión logística.



# 06

## Evaluación de modelos de regresión

Se presentan las métricas más comunes para evaluar los modelos de regresión: MSE, RMSE, MAE, Median Absolute Error y el coeficiente de determinación.





# Funciones de coste

Para evaluar nuestro modelo de regresión, existen diversas funciones de coste. Algunas de las más utilizadas son:

- **MSE (Mean Squared Error)**: el error cuadrático medio viene definida por la función de coste:

$$C(h_\theta, X, y) = \frac{1}{m} \sum_{i=1}^m (h_\theta(X_i) - y_i)^2$$

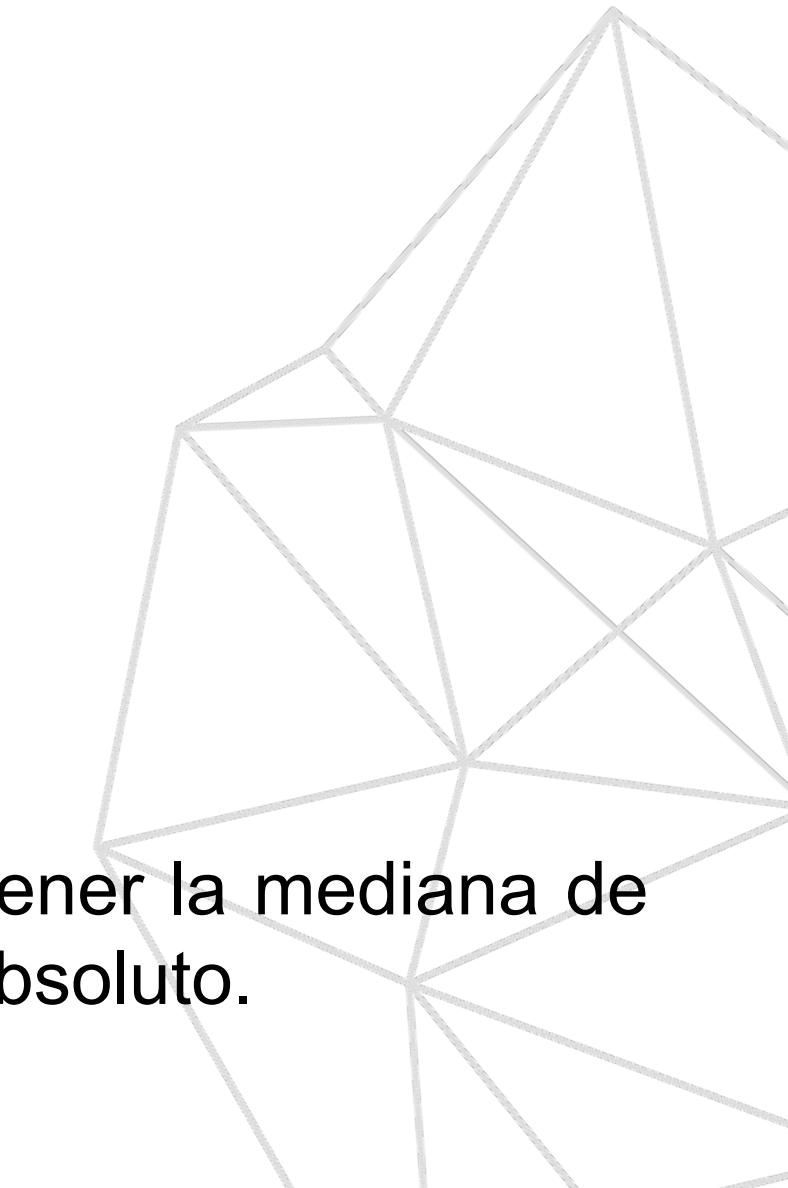
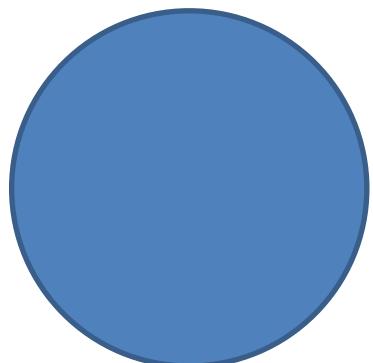
- **RMSE (Root Mean Squared Error)**:

$$C(h_\theta, X, y) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h_\theta(X_i) - y_i)^2}$$

- **MAE (Mean Absolute Error)**: error absoluto medio:

$$C(h_\theta, X, y) = \frac{1}{m} \sum_{i=1}^m |h_\theta(X_i) - y_i|$$

- **Median Absolute Error**: la mediana del error absoluto consiste en obtener la mediana de entre todas las funciones de pérdida que se obtienen hallando el error absoluto.





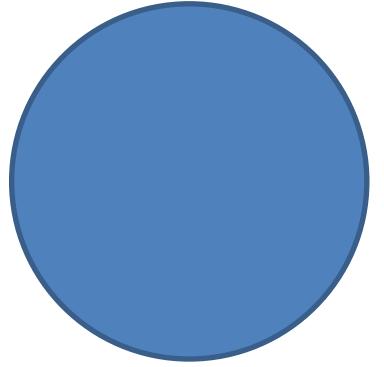
# Coeficiente de determinación

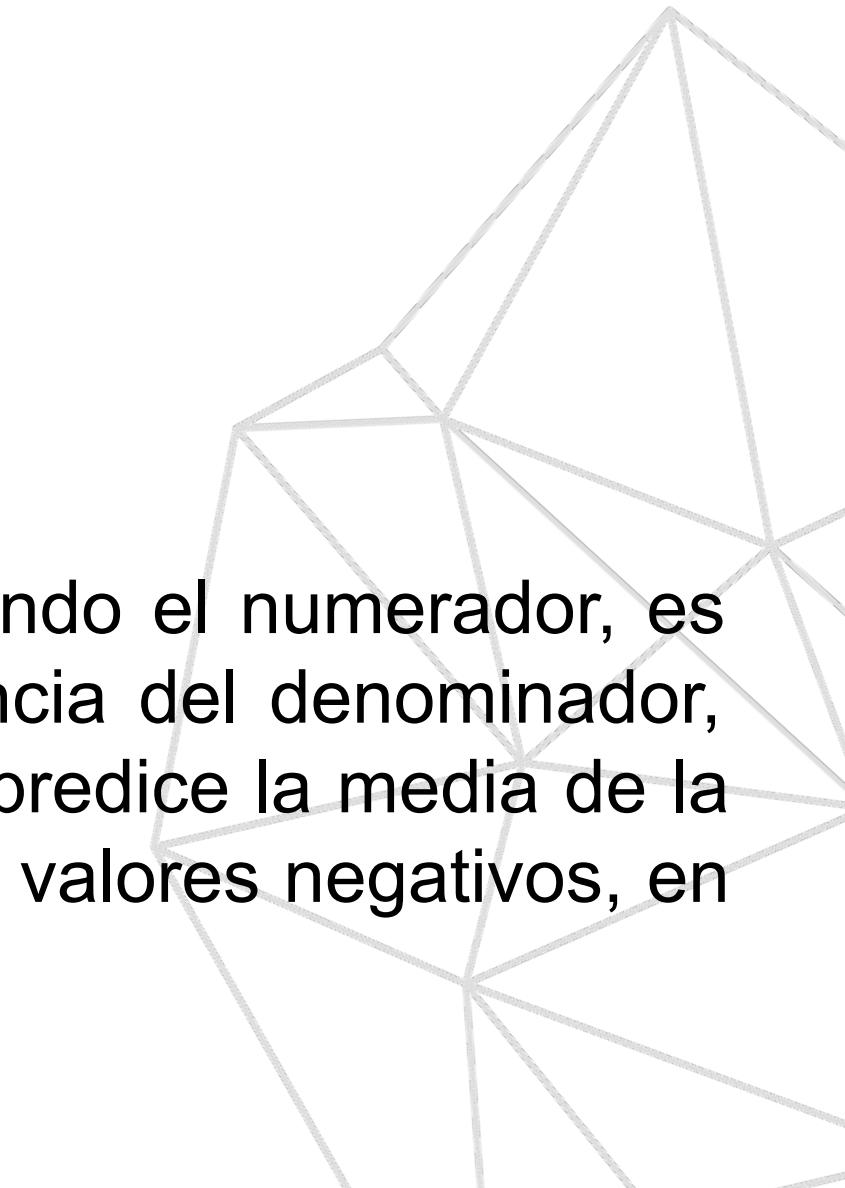
El coeficiente de determinación denotado por  $R^2$  es una métrica de evaluación de un modelo de regresión que puede ser (intuitivamente) más informativo que las vistas anteriormente. Se define como:

$$R^2 = 1 - \frac{\sum_{i=1}^m (h_\theta(X_i) - y_i)^2}{\sum_{i=1}^m (\bar{y} - y_i)^2}$$

Donde tenemos que  $\bar{y}$  es la media de la variable objetivo, es decir:

$$\bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$$

El mejor valor que puede obtenerse es 1 (ya que solo se alcanza cuando el numerador, es decir, la suma de los cuadrados de los errores, es 0). Dada la presencia del denominador, cuando se obtiene un valor 0 es equivalente a un modelo que siempre predice la media de la variable objetivo. Sin embargo, este coeficiente también puede devolver valores negativos, en cuyo caso sería un modelo peor que el que siempre predice la media.

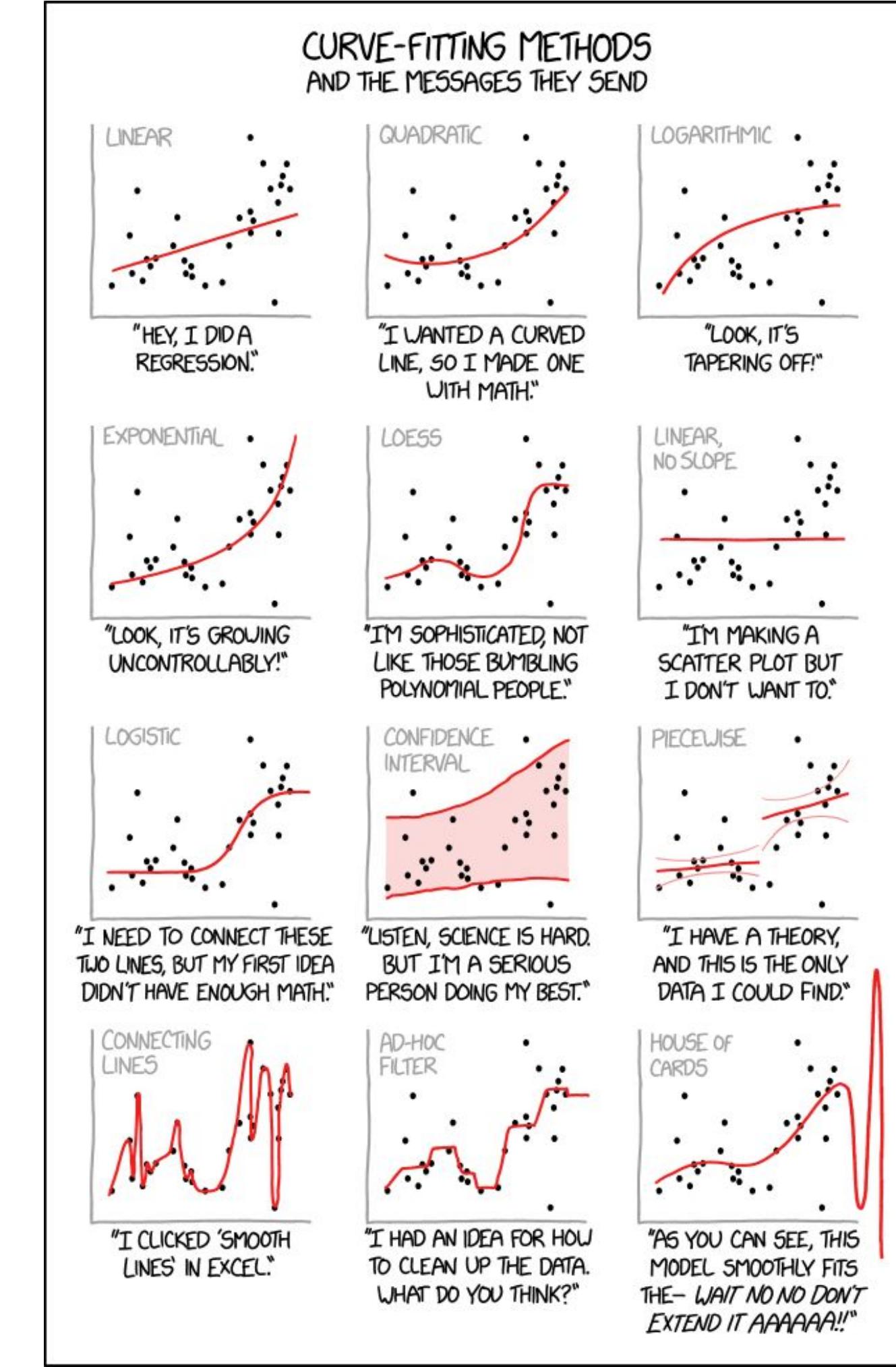
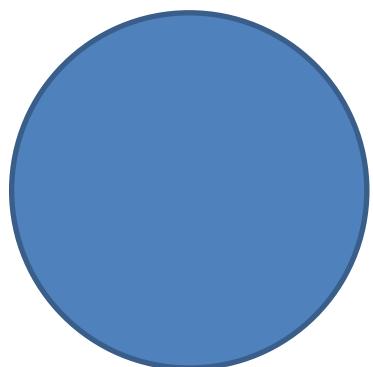




# 07

## Algoritmos de regresión

Se presentan los algoritmos de regresión como la regresión polinómica, K-NN, Árboles de decisión, SVM y Naive Bayes





# Regresión polinómica

Este tipo de regresión puede considerarse un caso concreto de la regresión lineal. Su modelo viene dado por una función polinómica de la que queremos ajustar sus coeficientes. Estos pueden calcularse de la misma forma que la regresión lineal y los términos de grado mayor que 1 pueden calcularse a partir del conjunto de datos. Por ejemplo, el modelo con tres variables de grado 2 sería:

$$h_{\theta}(X_i) = \theta_0 + \theta_1 x_{i,1}^2 + \theta_2 x_{i,2}^2 + \theta_3 x_{i,3}^2 + \theta_4 x_{i,1} x_{i,2} + \theta_5 x_{i,1} x_{i,3} + \theta_6 x_{i,2} x_{i,3} + \theta_7 x_{i,1} + \theta_8 x_{i,2} + \theta_9 x_{i,3}$$

Por ejemplo, supongamos que queremos predecir el precio de un coche de segunda mano, a partir del kilometraje y su edad. Entonces el modelo de grado 2 será:

$$h_{\theta}(X_i) = h_{\theta}(x_{i,1}, x_{i,2}) = \theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2} + \theta_3 x_{i,1} x_{i,2} + \theta_4 x_{i,1}^2 + \theta_5 x_{i,2}^2$$

Kilometraje del coche  $i$

Precio del coche  $i$

Edad del coche  $i$

Es equivalente a escribir:

$$h_{\theta}(X_i) = h_{\theta}(x_{i,1}, x_{i,2}) = \theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2} + \theta_3 x_{i,3} + \theta_4 x_{i,4} + \theta_5 x_{i,5}$$



# Otros algoritmos de regresión

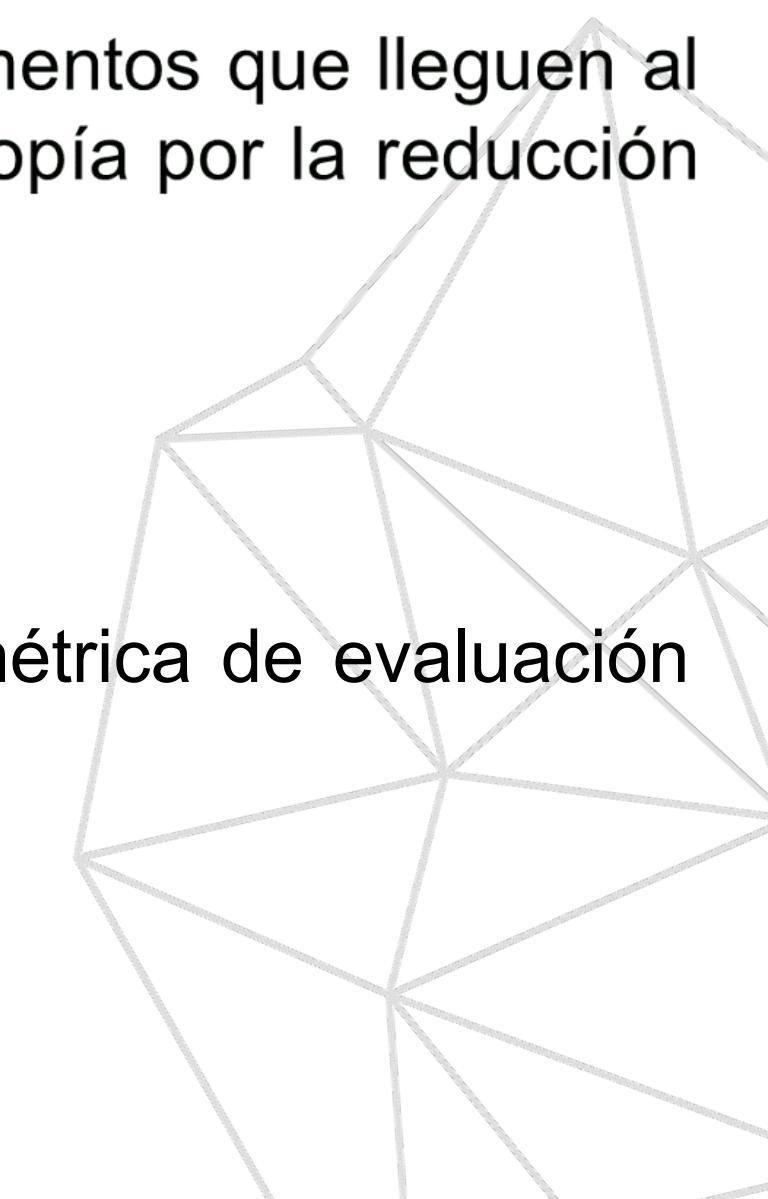
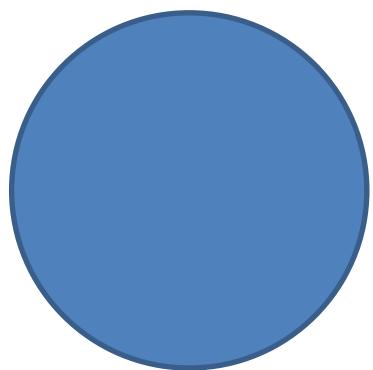
Muchos de los algoritmos de clasificación vistos pueden adaptarse para ser usados como algoritmos de regresión. Estos son:

- **K-NN**: en lugar de devolver la clase más frecuente entre los k elementos más cercanos, se devuelve la media de la variable objetivo de dichos elementos.
- **Árboles de decisión**: para devolver una predicción calculará la media de los elementos que lleguen al nodo hoja. De cara a construir el árbol, con ID3 se sustituirá la diferencia de entropía por la reducción de desviación estándar:

$$SDR(N|A) = \sigma(N) - \sigma(N|A) = \sigma(N) - \sum_{j=1}^p P(a_j)\sigma(N|a_j)$$

Mientras que en CART, en lugar de el índice de Gini se utilizará una métrica de evaluación como puede ser el MSE.

- **SVM**
- **Naive Bayes**

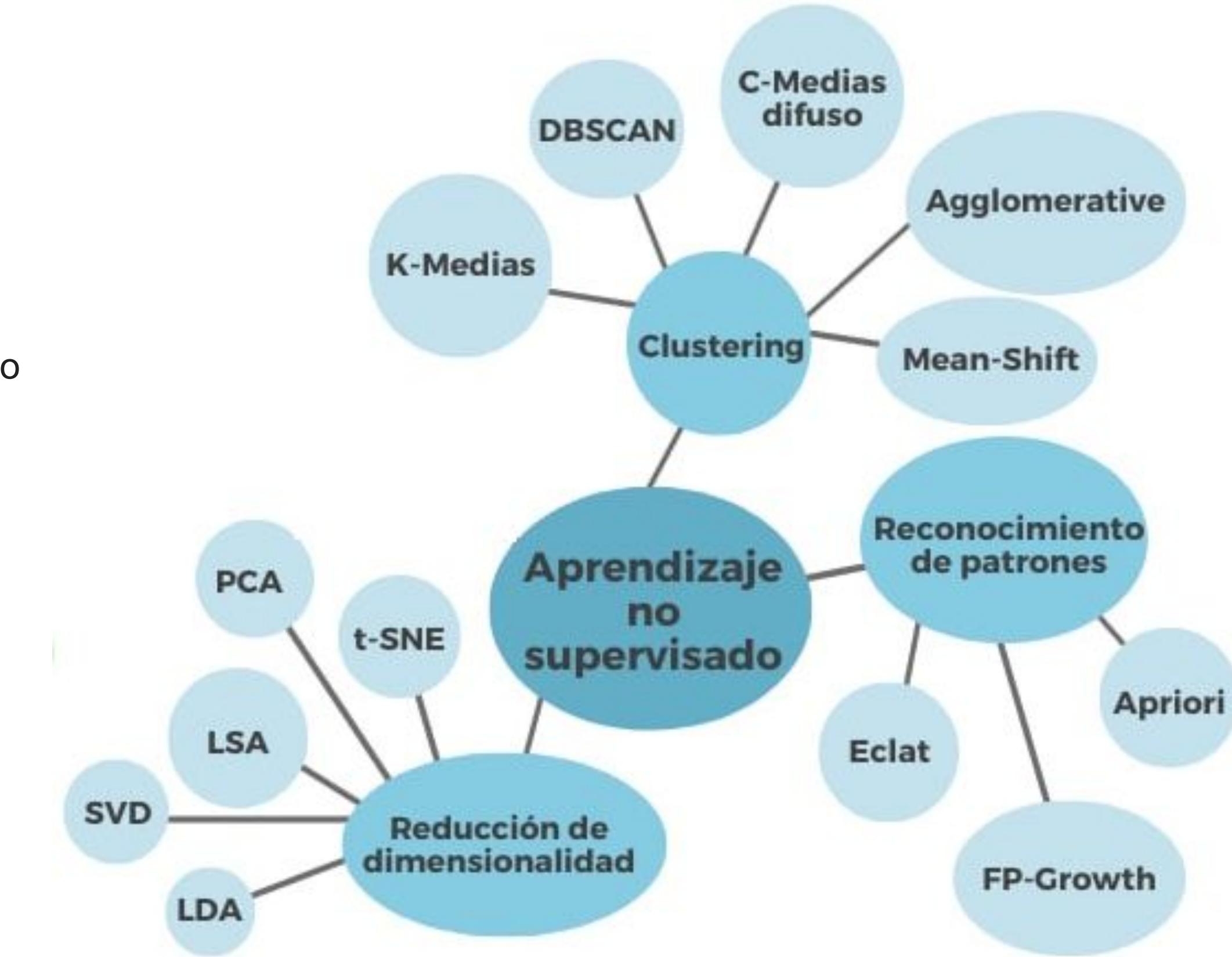




# 08

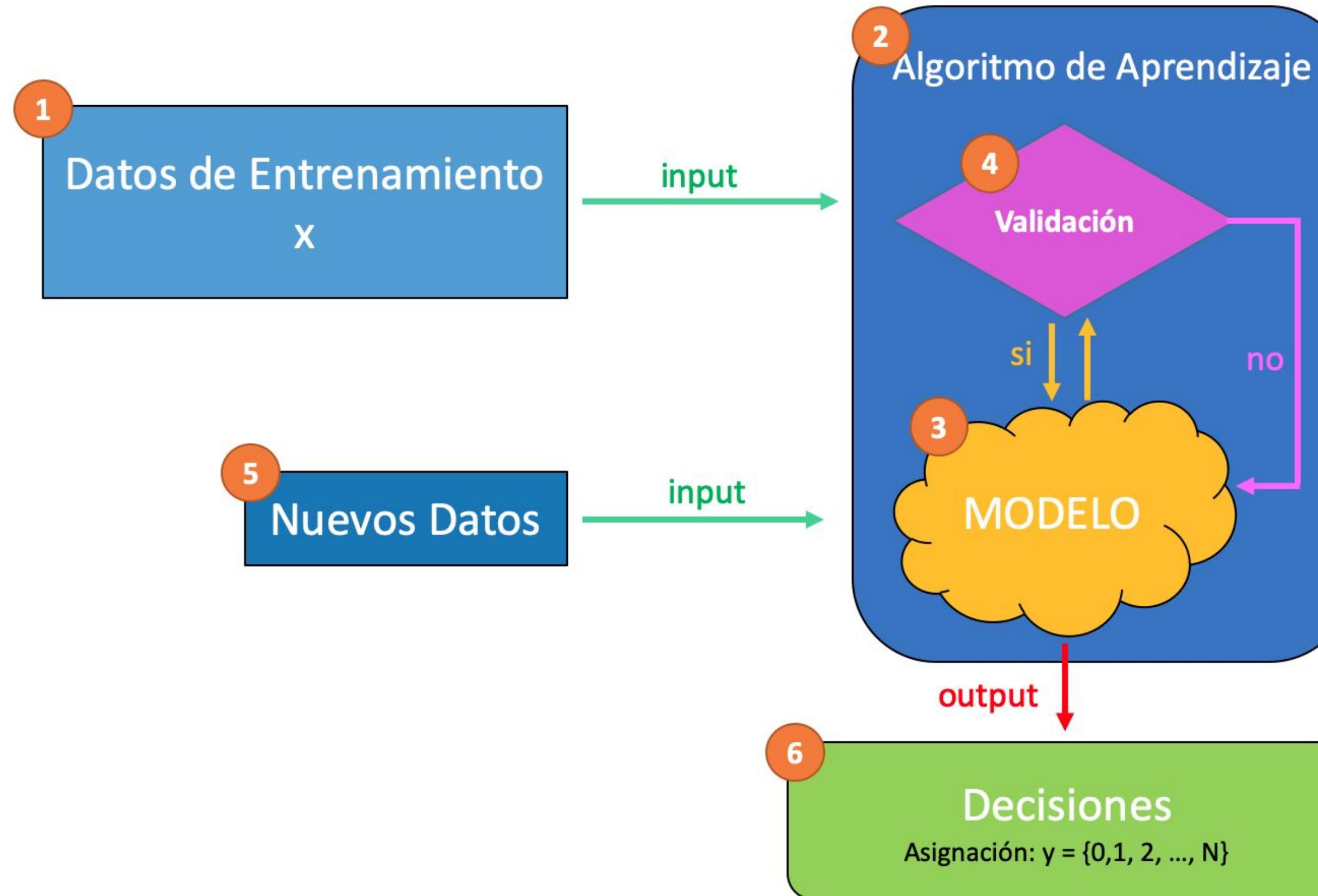
## Introducción al Aprendizaje No Supervisado

Introducción y esquema general del aprendizaje no supervisado.





# Esquema general

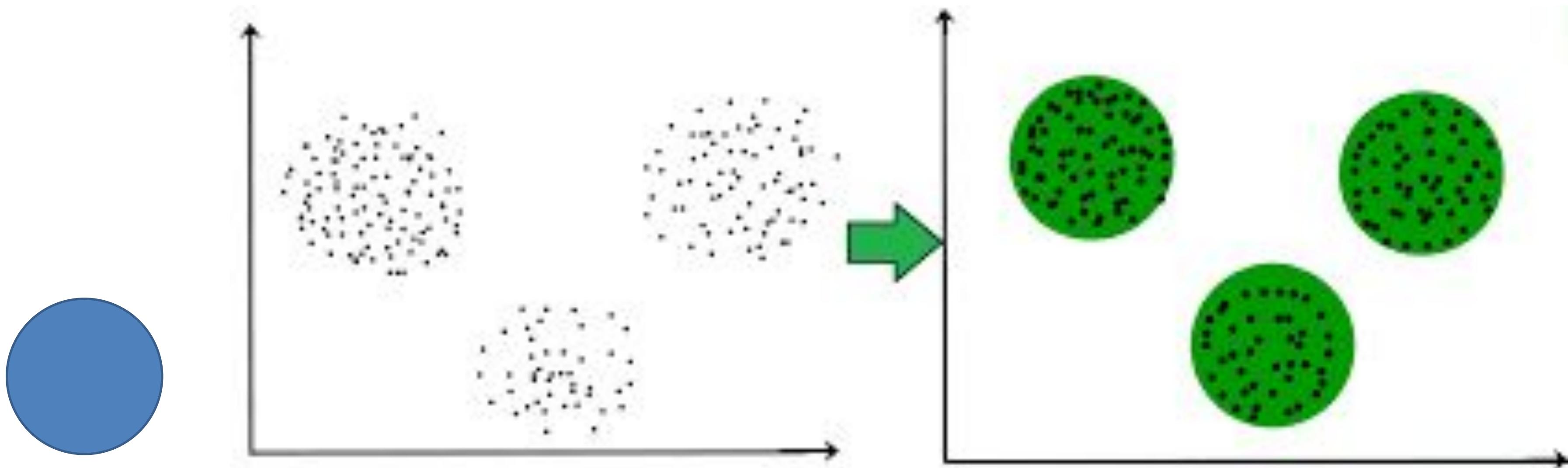




# 09

## Algoritmos de Clustering

Se presentan los algoritmos de clustering más utilizados: K-Medias y DBSCAN.



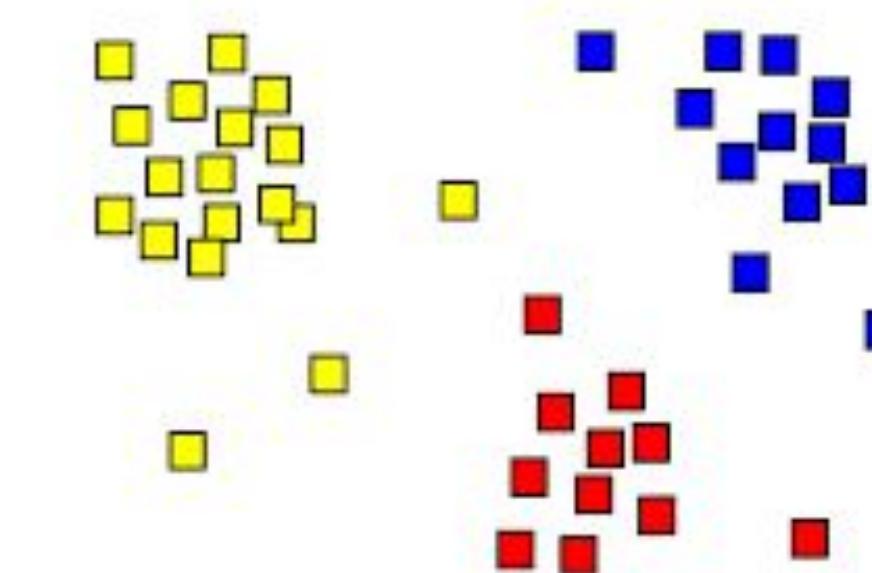
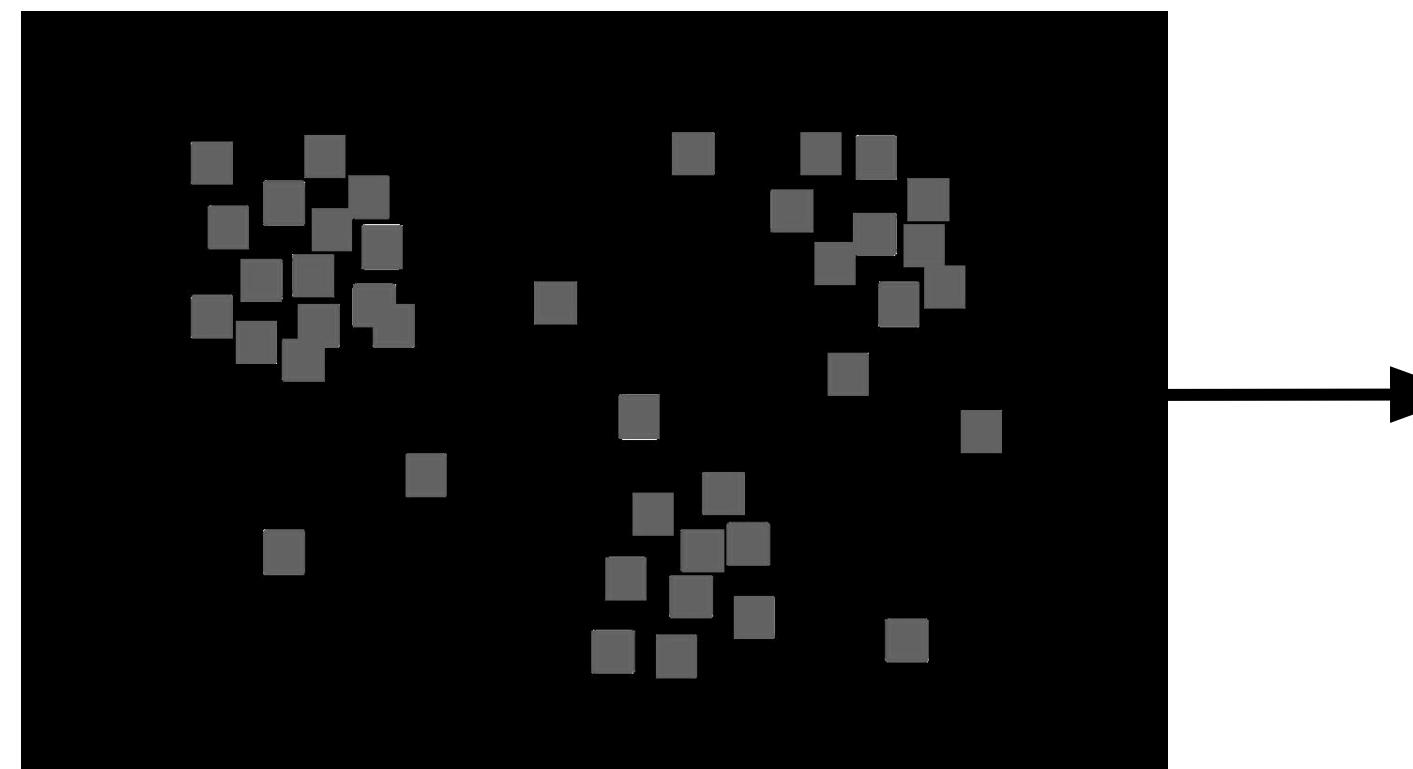
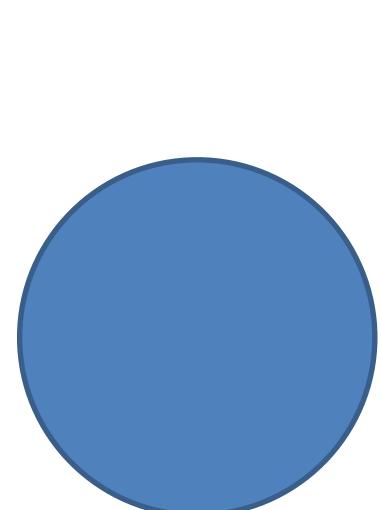


# Introducción

El análisis de grupos (más conocido como *clustering*) es una metodología de análisis de datos que nos permite identificar posibles agrupaciones de datos en función de las similitudes y disimilitudes entre ellas.

**Cluster** es un conjunto de instancias que son similares entre sí y diferentes a las instancias de otros clústeres.

Algunas aplicaciones del clustering son: segmentación de mercado (tipos de clientes, fidelidad, ...), agrupar puntos cercanos en un mapa (datos geográficos), analizar y etiquetar nuevos datos y detección de ruido o comportamientos fuera de lo normal.

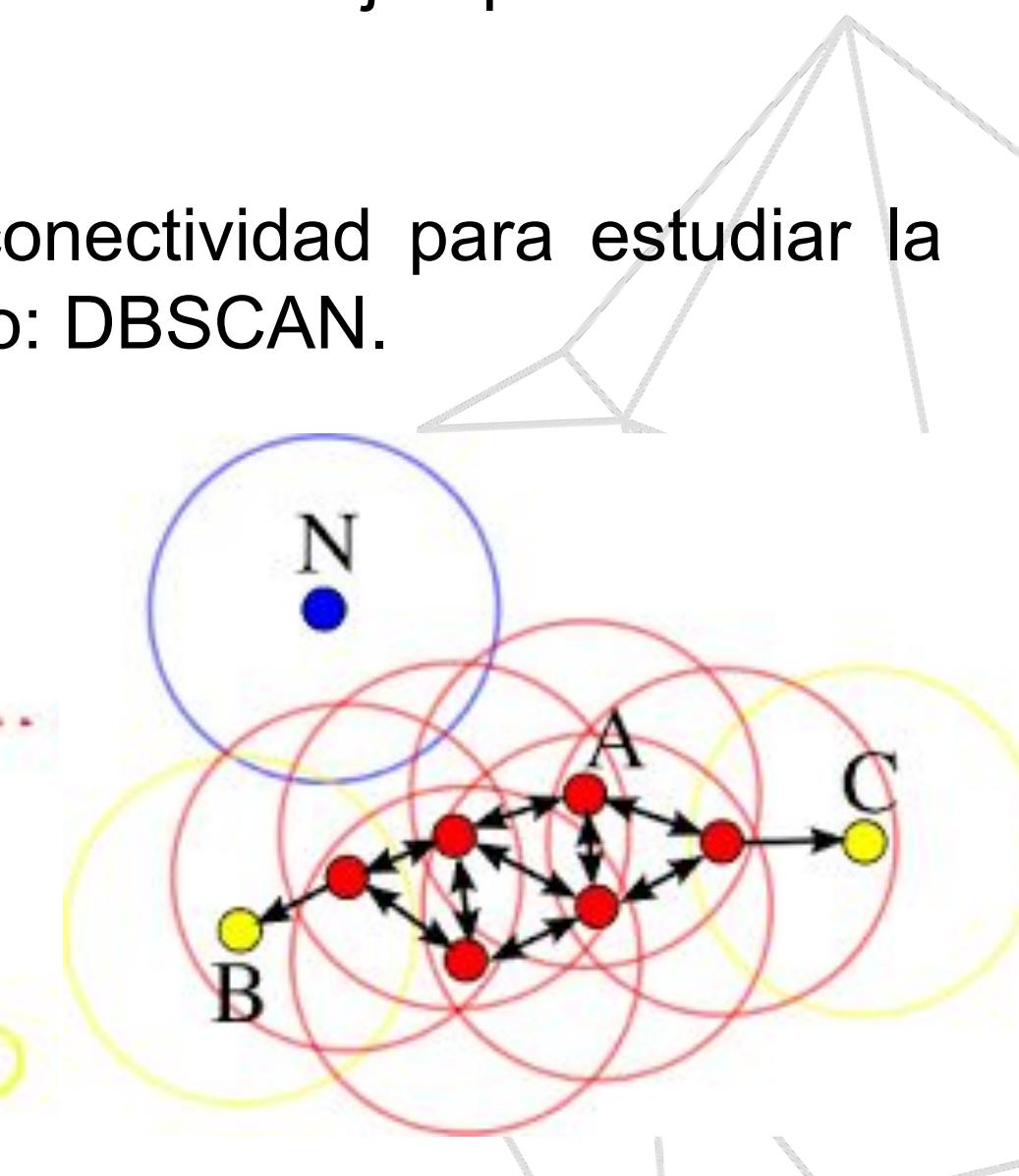
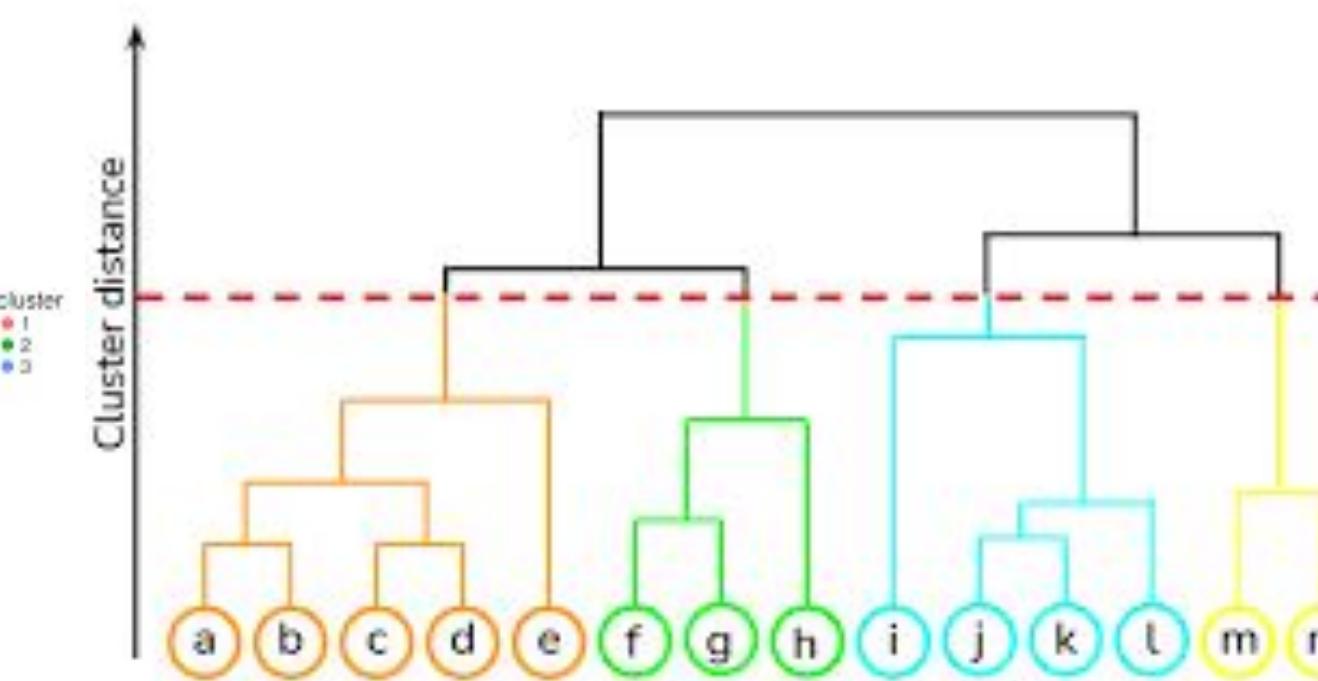
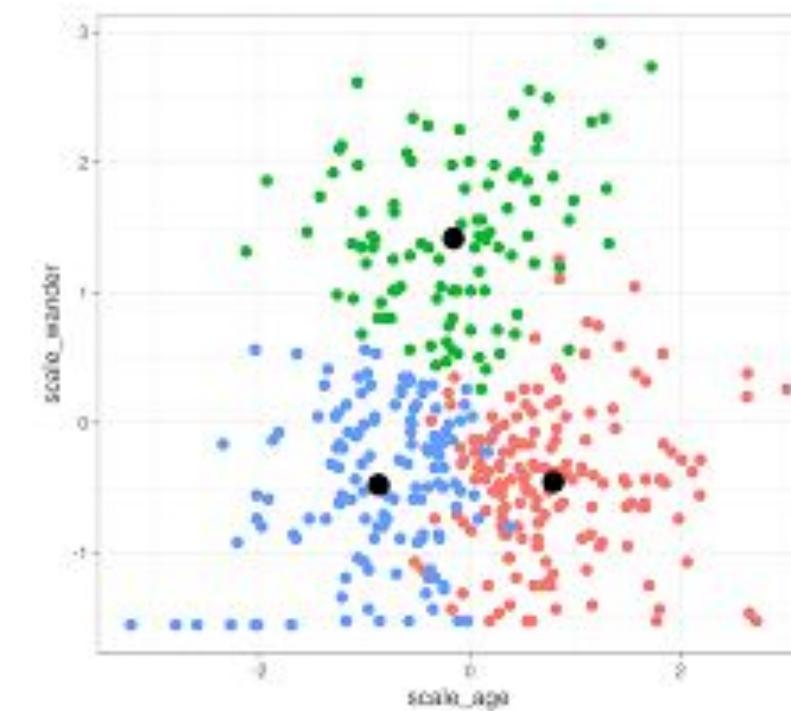
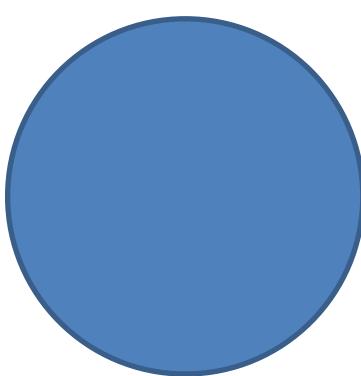




# Introducción

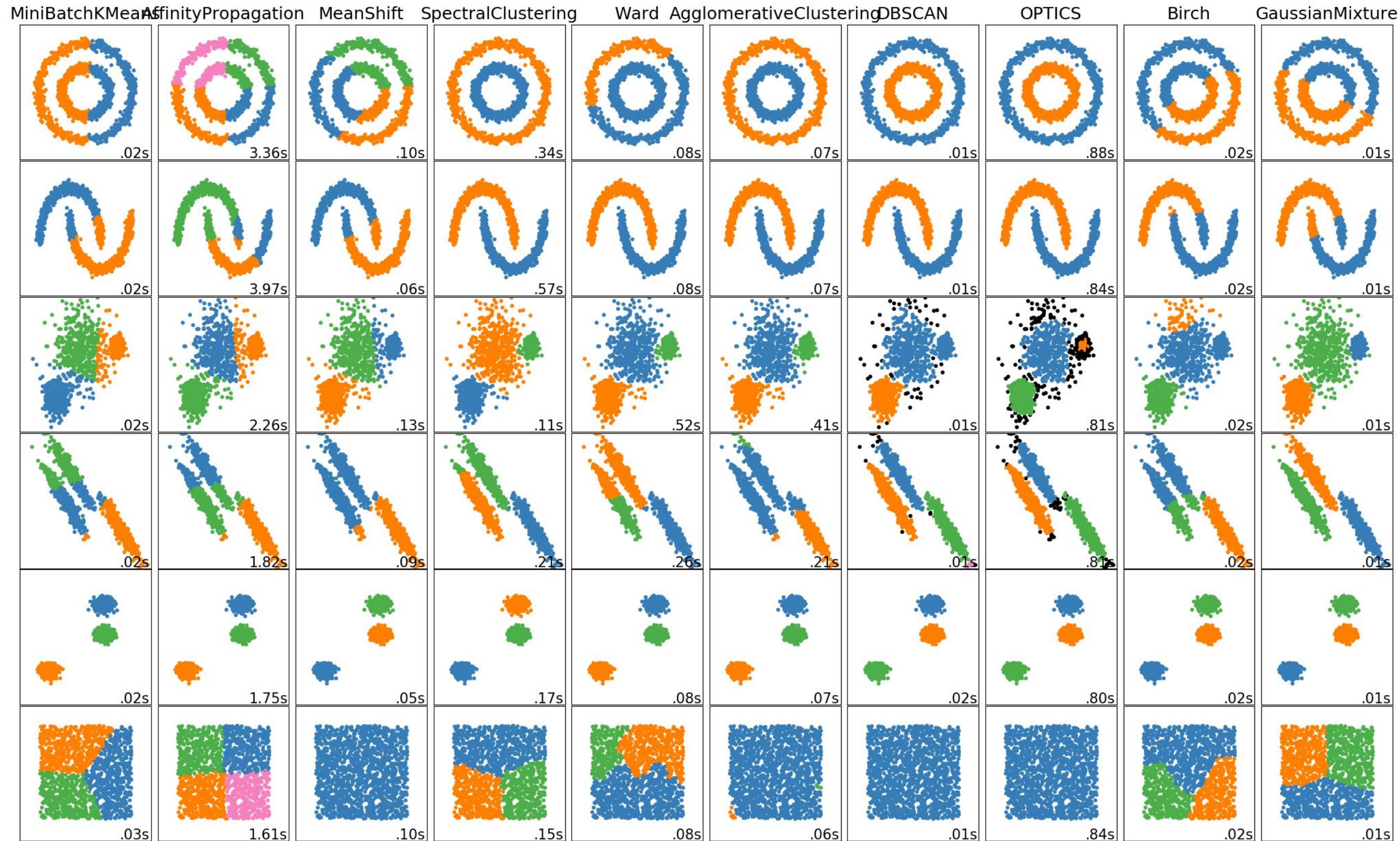
Existen distintos tipos de clustering de acuerdo al método utilizada. Los métodos más comunes son:

- **Método de partición:** se crean particiones sucesivas del conjunto de datos de entrenamiento. Ejemplos: K-Medias o K-Medianas.
- **Métodos jerárquicos:** se descompone jerárquicamente el conjunto de datos. Ejemplos: AGNES (aglomerativo o *bottom-up*) o DIANA (divisorio o *top down*).
- **Métodos basados en densidad:** se utilizan funciones de densidad y conectividad para estudiar la interrelación. Pueden detectar ruido (datos en zonas poco densas). Ejemplo: DBSCAN.





# Dependencia del algoritmo

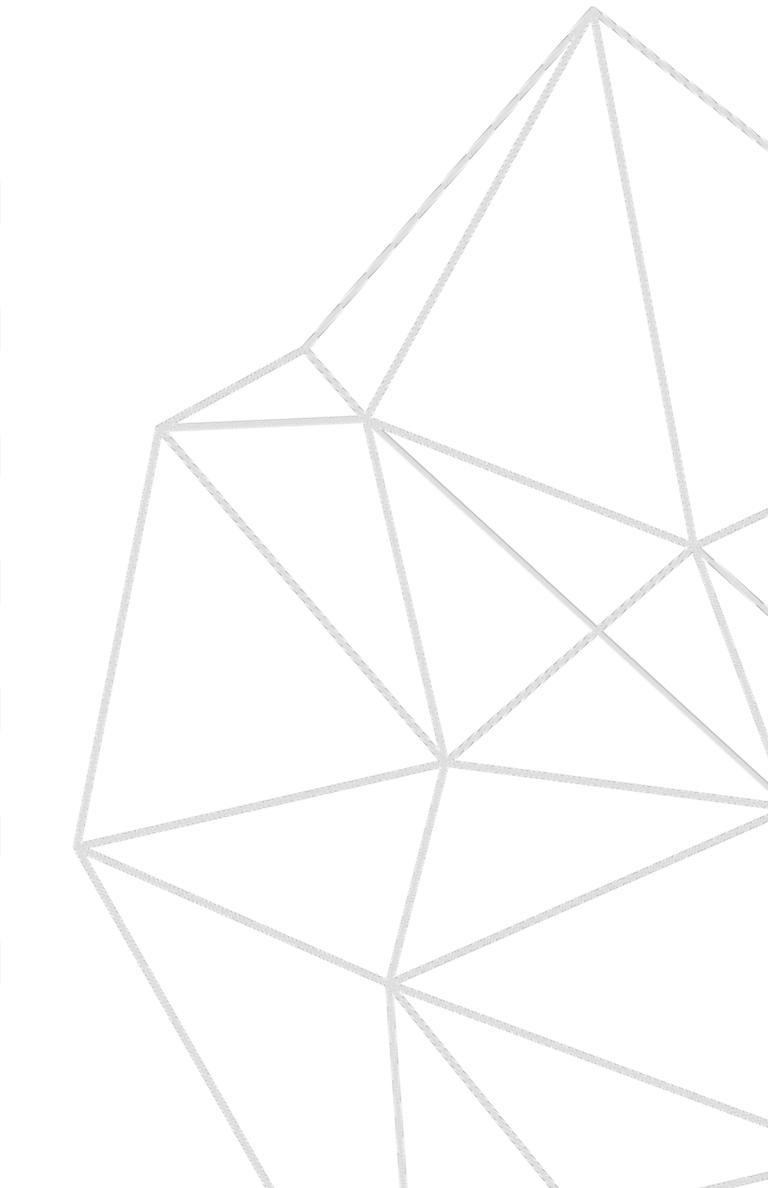
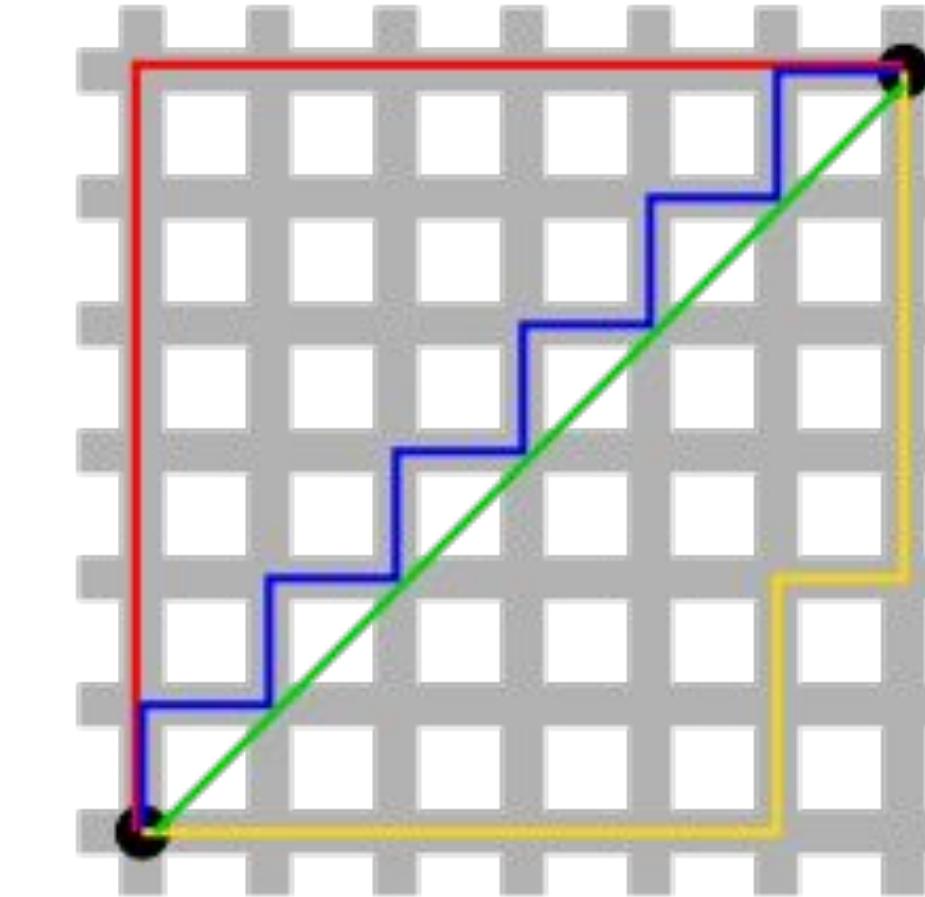
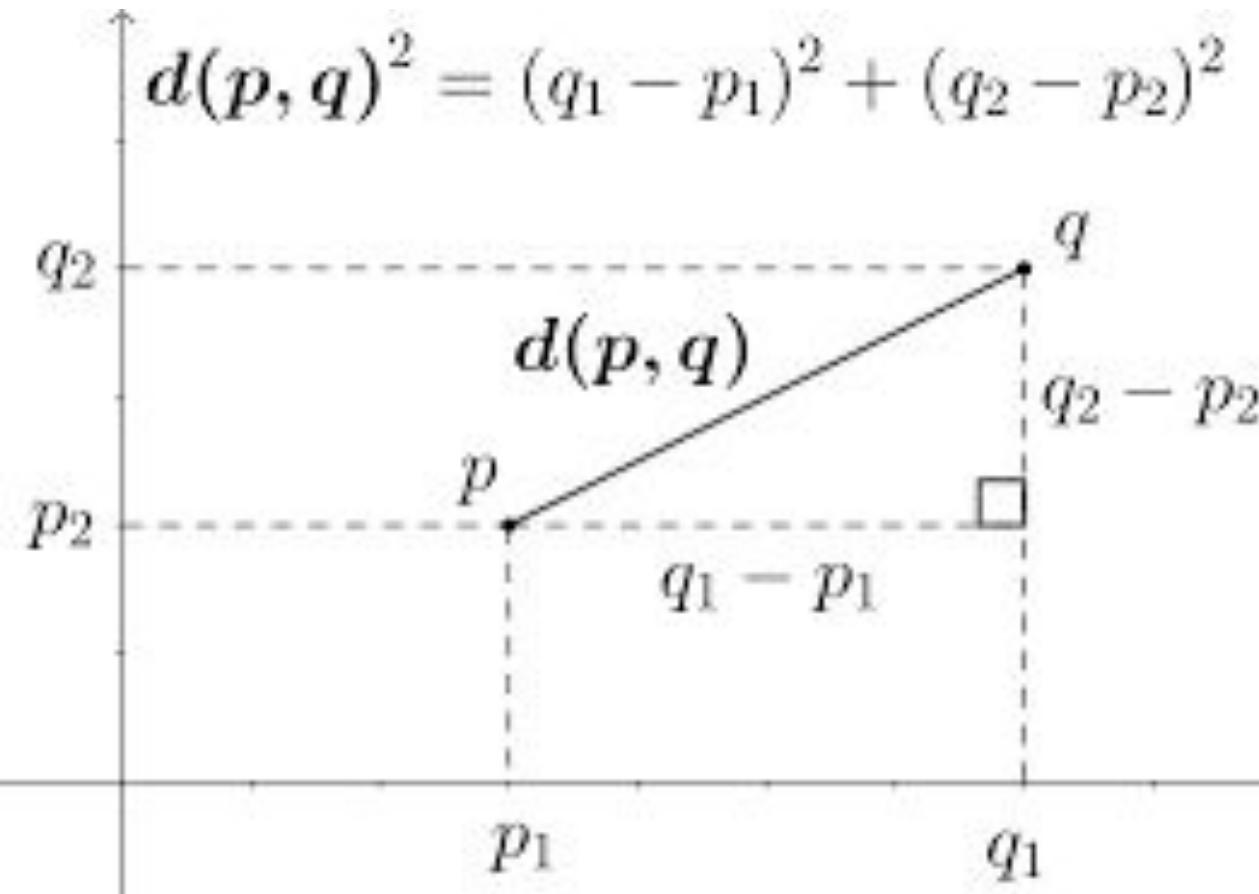
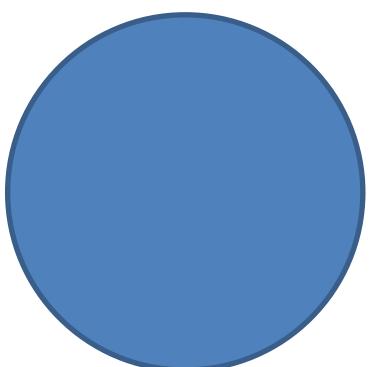




# K-Medias: distancias

El objetivo es dividir el conjunto de datos de tamaño  $m$  en  $k$  particiones (valor predefinido). Utiliza **medidas de distancia** para analizar similitudes.

Es un buen algoritmo analizando conjuntos de datos con agrupaciones con formas simples (esférica para la distancia euclídea, cuadrada para distancia Manhattan y para distancia infinito).





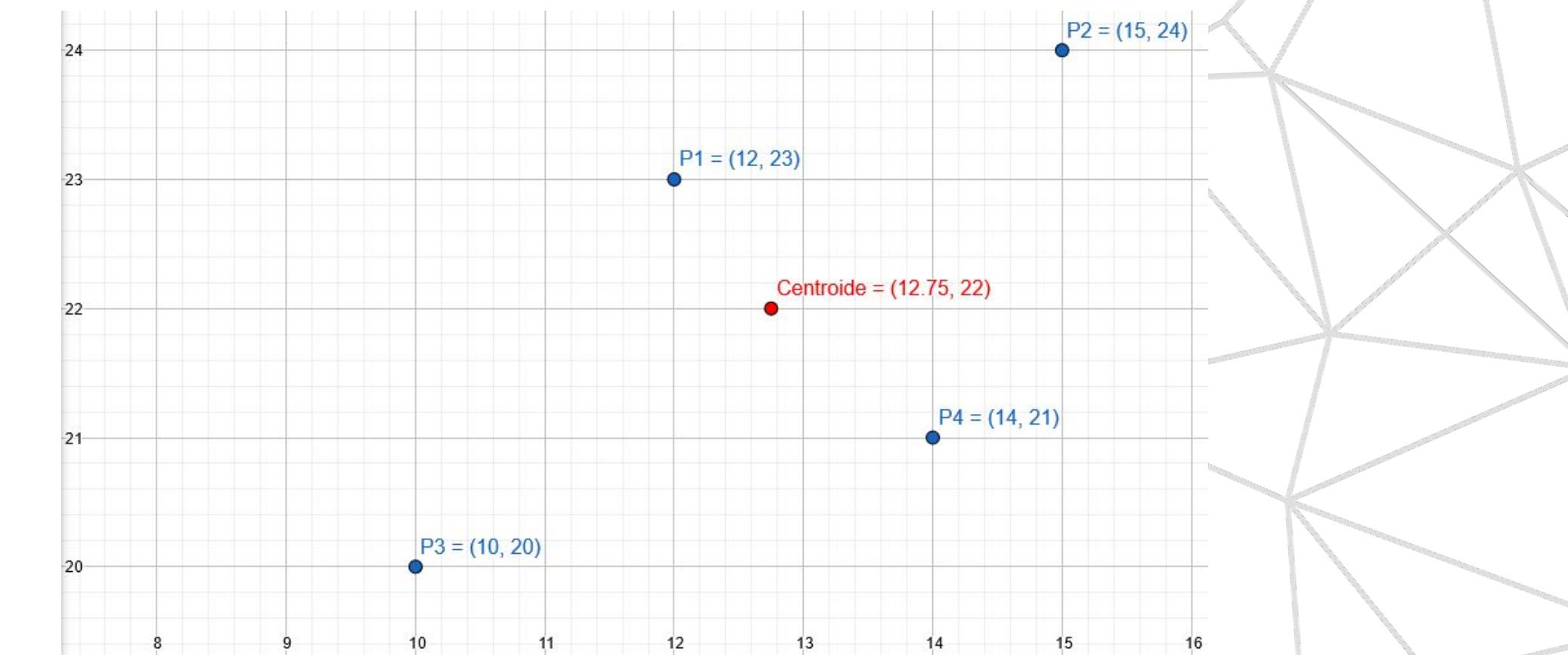
# K-Medias: centroides

Dado un cluster con  $m$  elementos podemos calcular la coordenada  $C^{(j)}$  del centroide del cluster como la media de la misma coordenada de sus elementos:

$$C^{(j)} = \frac{1}{m} \sum_{i=1}^m x_{i,j}$$

Quedando entonces el centroide como:  $C = (C^{(1)}, C^{(2)}, \dots, C^{(n)})$ . Veamos un ejemplo:

| Artículos comprados | Dinero gastado |
|---------------------|----------------|
| 12                  | 23             |
| 15                  | 24             |
| 10                  | 20             |
| 14                  | 21             |





# K-Medias: algoritmo

1. Seleccionar aleatoriamente  $k$  instancias como centroides iniciales de las particiones. También puede utilizarse  $k$  puntos aleatorios del espacio de búsqueda.
2. Asignar las instancias a la partición (o cluster) con el centroide más próximo (los centroides definen los clusters).
3. Calcular los centroides de cada partición (utilizando la media de cada coordenada) en función de las nuevas instancias asignadas.
4. Si las particiones han cambiado las instancias que les pertenecen, ir al paso 2, si no, acabar el algoritmo.

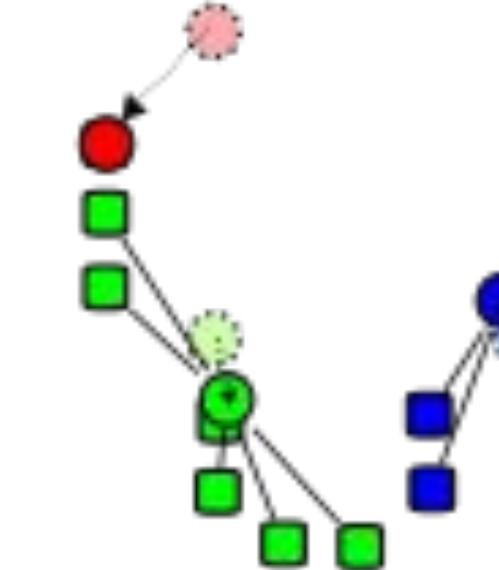
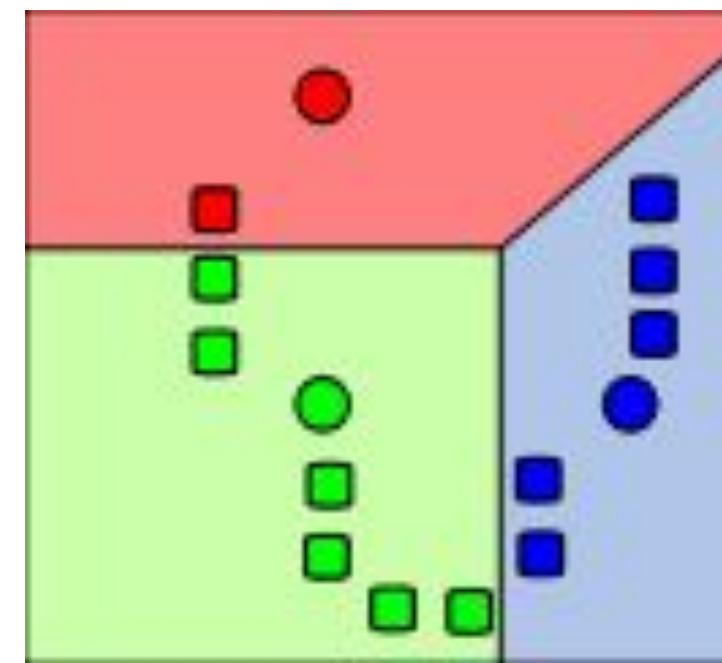
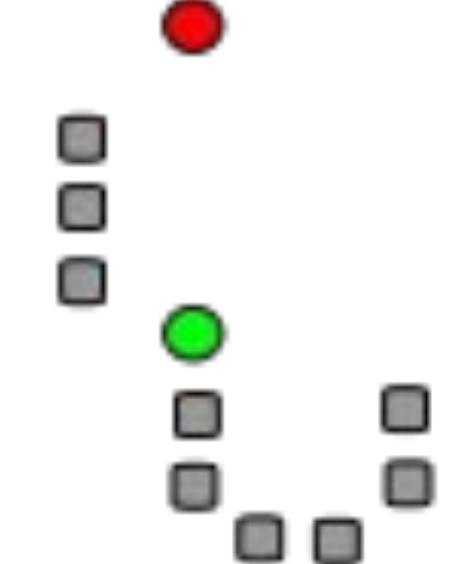
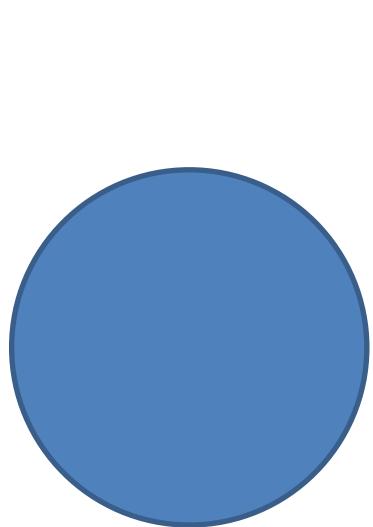


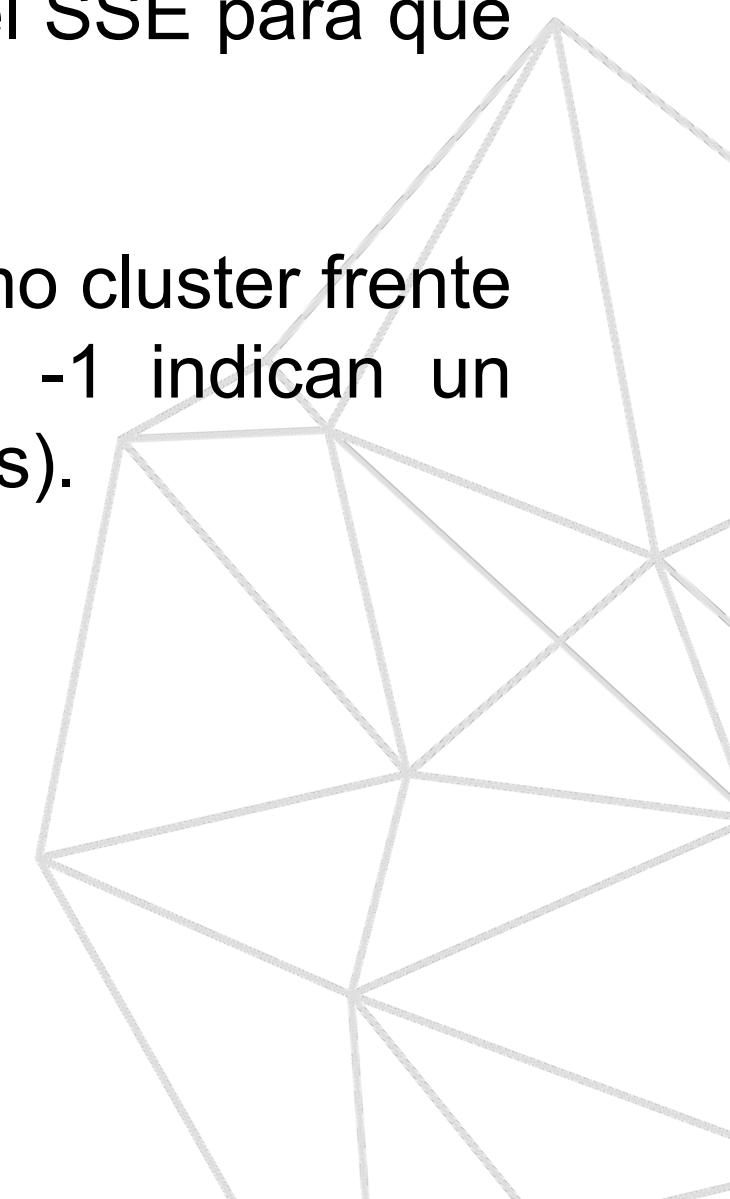
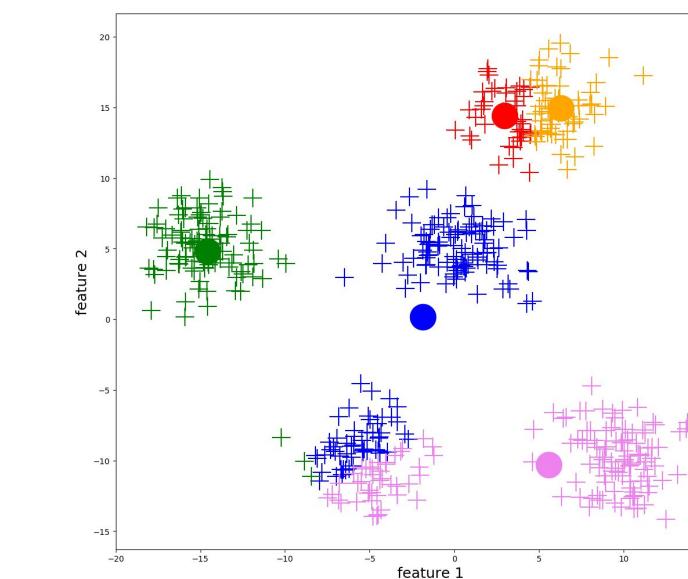
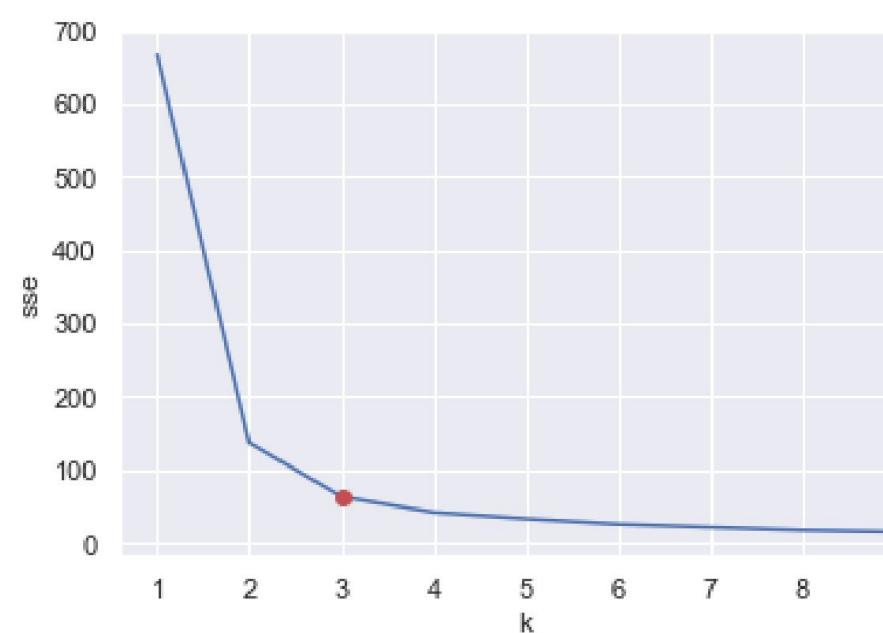
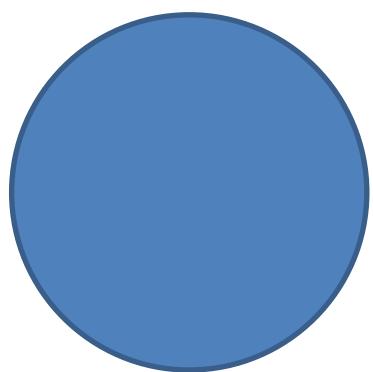
Imagen extraída de: <https://es.wikipedia.org/wiki/K-medias>



# K-Medias: hallar el número de particiones

Es recomendable utilizar K-Medias cuando se tiene conocimiento previo del número de particiones. No obstante existen varios métodos que permiten hacer una aproximación al número óptimo de particiones (en aprendizaje no supervisado no se puede utilizar un conjunto de validación):

- **Método Elbow** (método visual): para cada número de particiones se obtiene su error SSE (*Sum of Squared Errors*) y se muestra en una gráfica. Hay que buscar el balance entre  $k$  y el SSE para que ambos tomen valores reducidos.
- **Método de coeficiente de Silhouette**: mide la similitud entre elementos de un mismo cluster frente a la disimilitud frente a elementos de clústeres diferentes (valores cercanos a -1 indican un clustering erróneo, a 0 indica clústeres solapados y a 1 clústeres densos y separados).





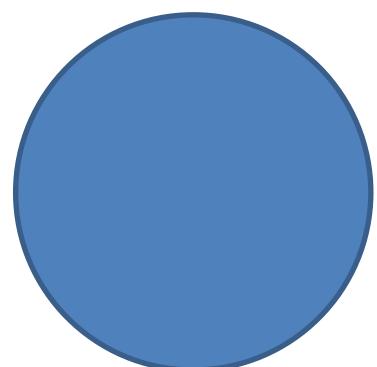
# DBSCAN: Introducción

DBSCAN es un algoritmo de clustering basado en densidad. Requiere dos parámetros iniciales:

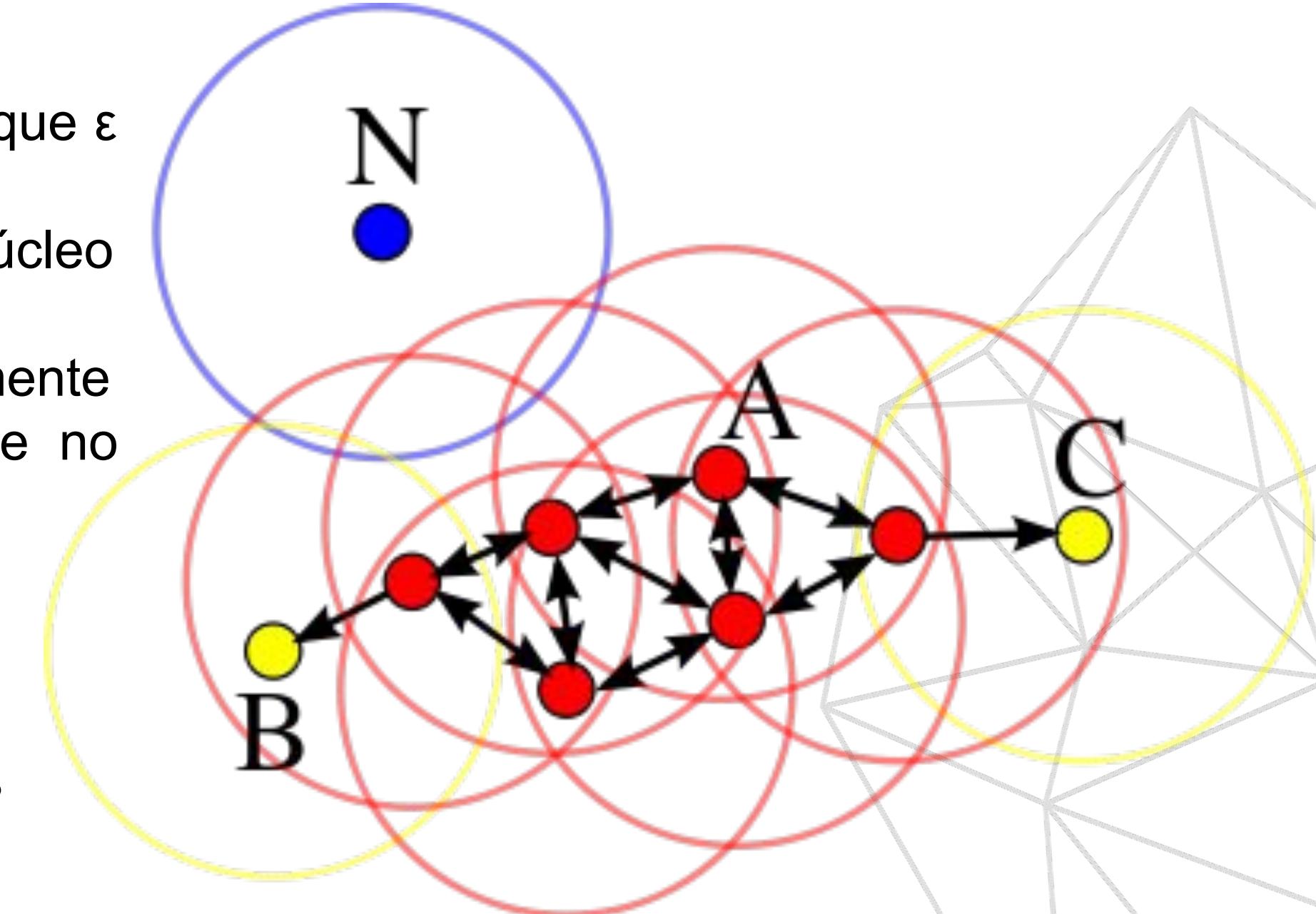
- $\epsilon$ : distancia máxima para considerar dos puntos como vecinos.
- $d$ : llamado factor de densidad.

Un punto puede ser:

- **Núcleo**: si posee  $d$  puntos a una distancia menor que  $\epsilon$  de él (los rojos).
- **Directamente alcanzable**: si es vecino de un núcleo (los amarillos).
- **Alcanzable**: si es vecino de un punto directamente alcanzable (si los amarillos tuvieran vecinos que no fueran vecinos de los rojos)
  - **Ruido**: si no son ninguno de los anteriores (el azul).



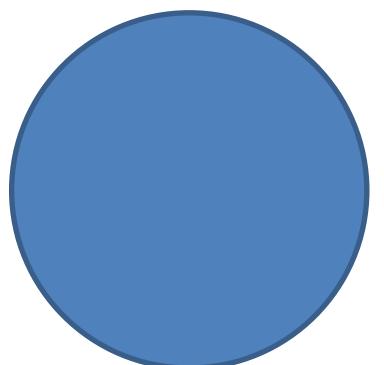
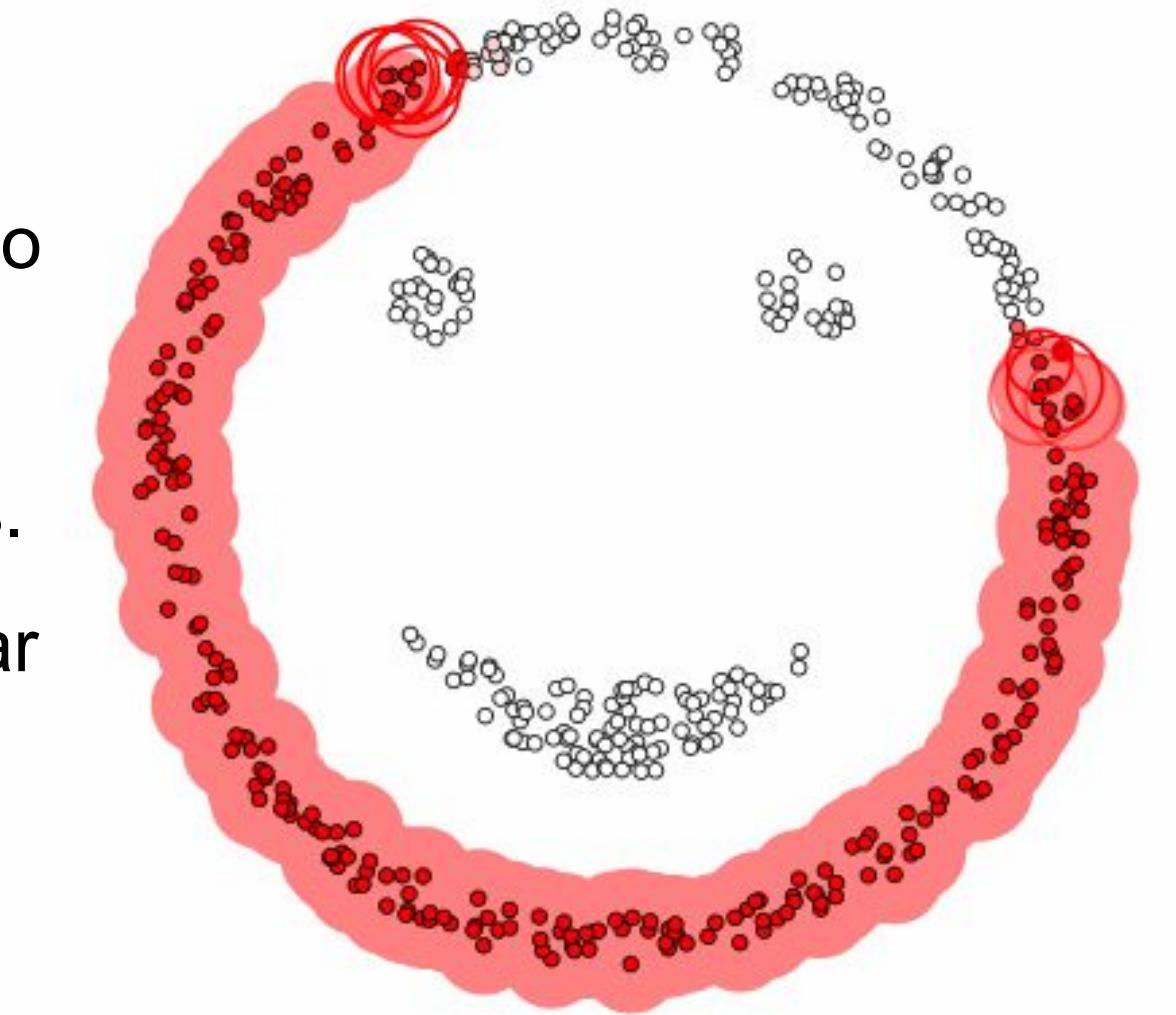
Los clústeres los conforman los tres primeros tipos de puntos.





# DBSCAN: Algoritmo

1. Creamos una lista  $L$  con todos los puntos
2. Tomamos el primer punto  $p$  de la lista  $L$
3. Si  $p$  no ha sido etiquetado ir al paso 4, si no, al paso 7
4. Si  $p$  tiene menos de  $d$  vecinos etiquetar como ruido e ir al paso 7, si no ir al paso 5.
5. Etiquetar  $p$  como núcleo y sus vecinos como directamente alcanzables.
6. Si alguno de sus vecinos estaba etiquetado a otro cluster, unificar ambos
7. Quitamos el punto  $p$  de la lista  $L$  y, si  $L$  no está vacía volver a 2



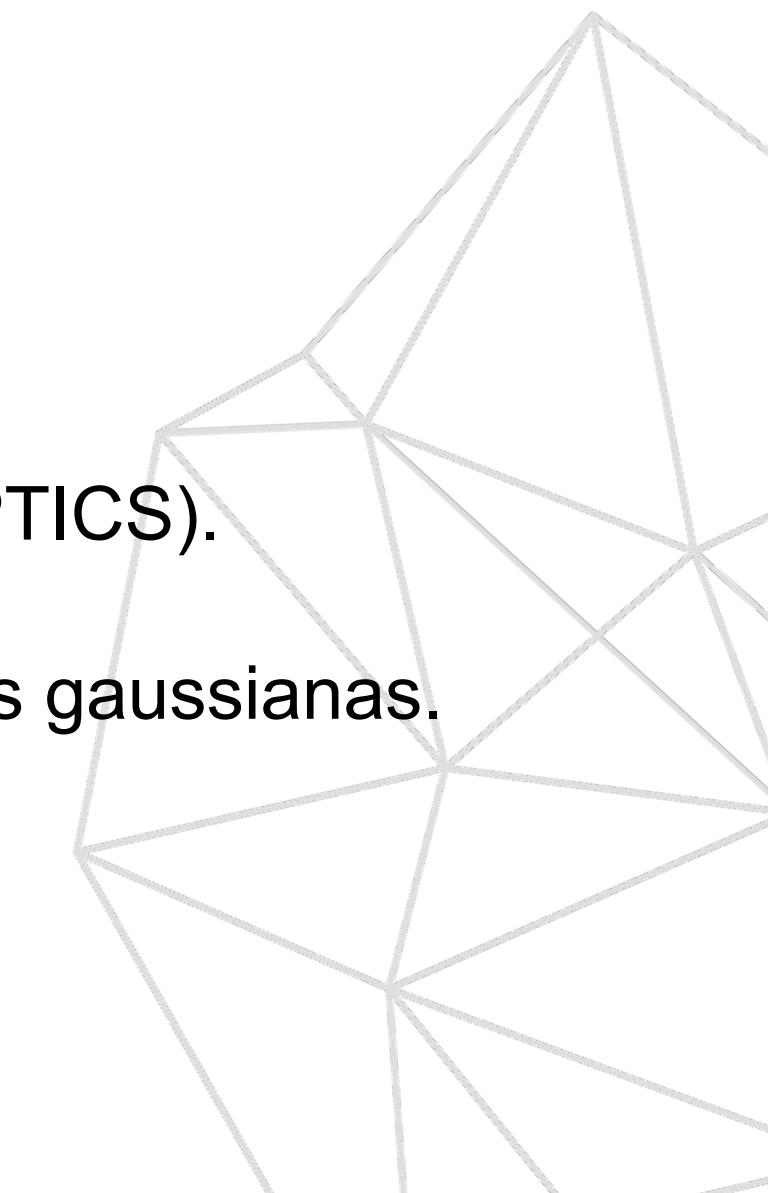
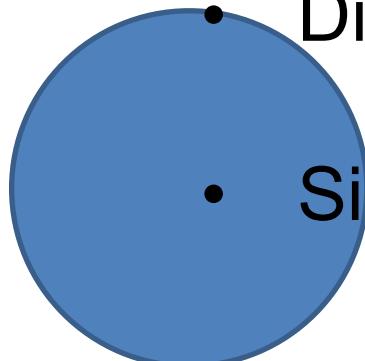


# DBSCAN: Ventajas y desventajas

## Ventajas:

## Desventajas:

- No tiene número de clústeres predefinido.
- No le afecta la forma de los clústeres.
- Es robusto ante ruido y outliers.
- Tiene dificultades para trabajar con clústeres de diferente densidad (alternativa: OPTICS).
- Dificultad para definir las fronteras de clústeres solapados que siguen distribuciones gaussianas.
- Si no se conoce el problema bien, es difícil definir  $\epsilon$

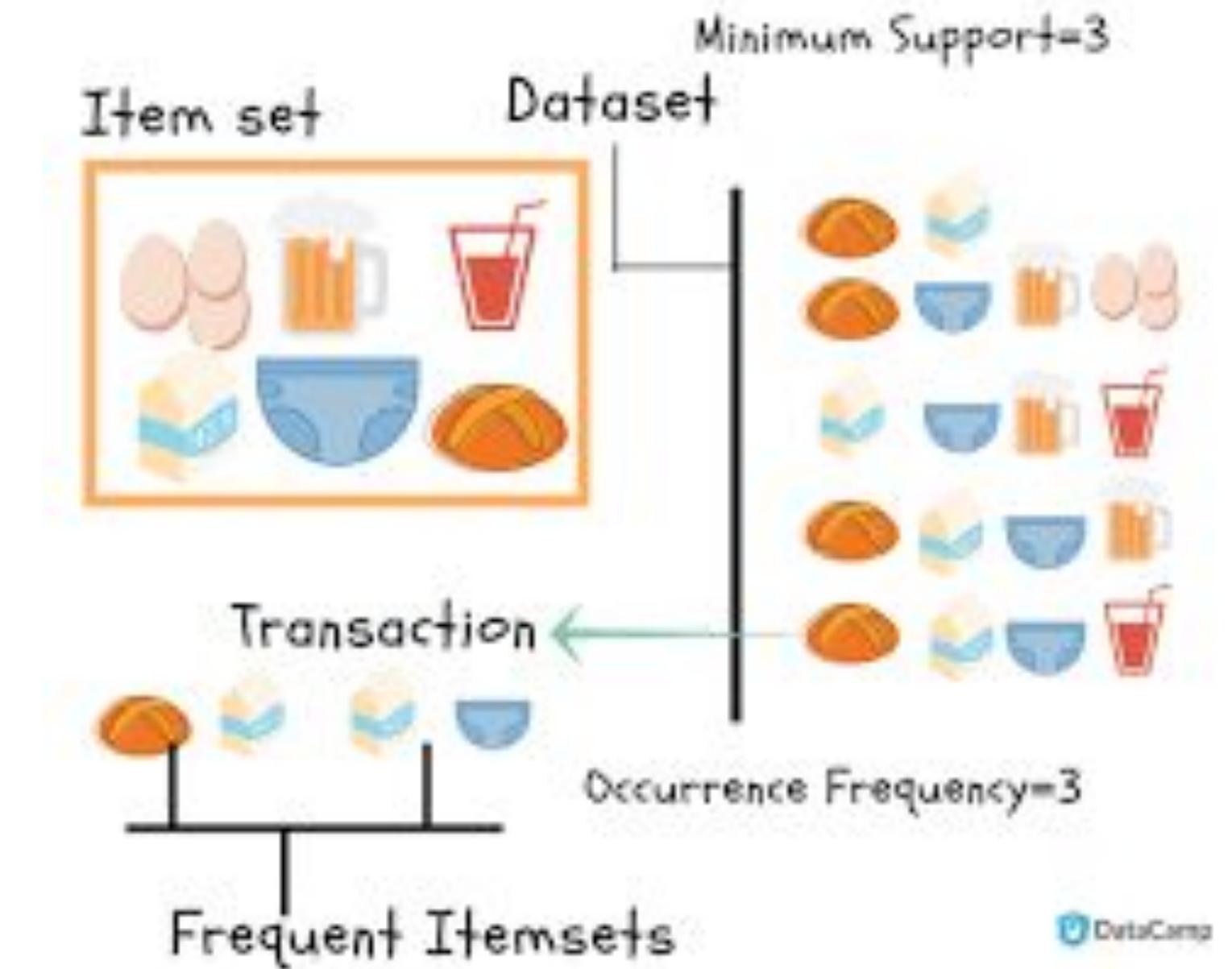
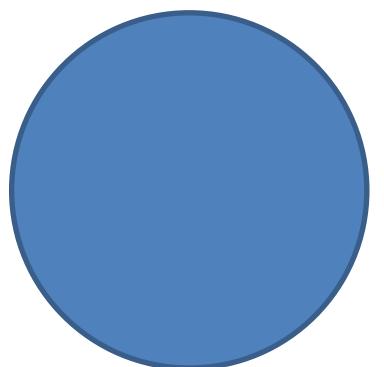




# 10

## Algoritmos de reglas de asociación

Explicación del algoritmo apriori.

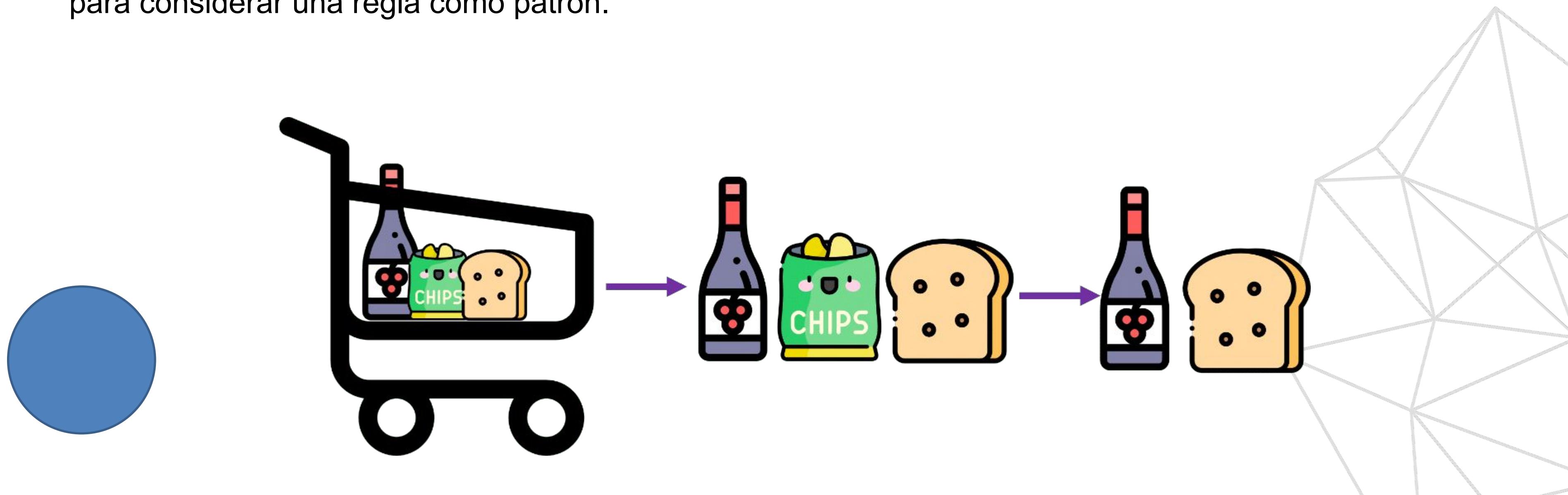




# Apriori: Introducción

Está diseñado para operar sobre bases de datos que contienen transacciones (por ejemplo, colecciones de artículos comprados por consumidores o detalles sobre la frecuentación a un sitio web). Cada transacción es vista como un conjunto de ítems.

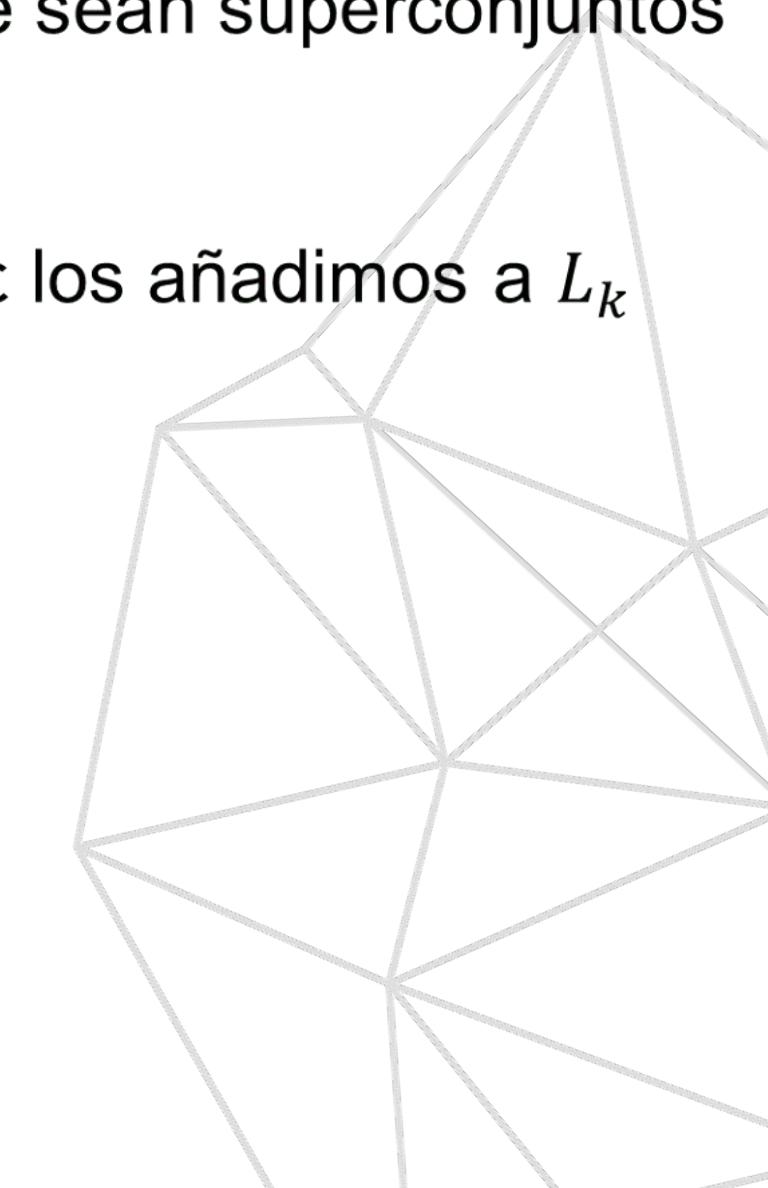
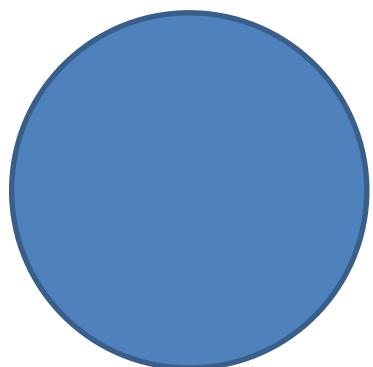
Requiere establecer un parámetro  $\epsilon$  que es la frecuencia mínima (mínimo número de repeticiones) para considerar una regla como patrón.





# Apriori: Algoritmo

1. Introducimos en el diccionario  $L_1$  todos los elementos de las transacciones y su frecuencia en la BBDD.
2. Eliminamos de  $L_1$  los elementos que tengan frecuencia inferior a  $\epsilon$
3. Inicializamos  $k = 2$  y  $R = L_1$  e  $I = \{e \mid \{e\} \notin L_1\}$
4. Sea  $C$  el resultado de añadir un elemento a los conjuntos de  $L_{k-1}$  sin que sean superconjuntos de los elementos de  $I$ .
5. Contamos la frecuencia de los elementos de  $C$  y si es superior o igual a  $\epsilon$  los añadimos a  $L_k$
6. Si la frecuencia de un elemento de  $C$  es inferior a  $\epsilon$  lo añadimos a  $I$ .
7. Añadimos los elementos de  $L_k$  a  $R$  e incrementamos  $k$  en 1
8. Si  $L_{k-1}$  no es vacío, ve al paso 4, de lo contrario finaliza.



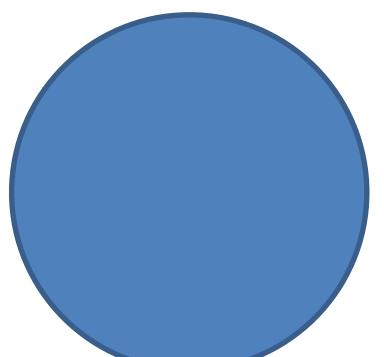


# Apriori: Ejemplo

| L1  |   |
|-----|---|
| {a} | 3 |
| {b} | 7 |
| {c} | 4 |
| {d} | 6 |

| I   |
|-----|
| {e} |

$k = 2$



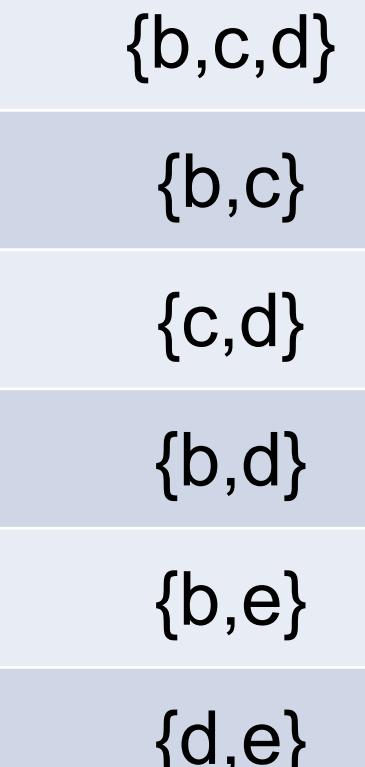
| C     |   |
|-------|---|
| {a,b} | 3 |
| {a,c} | 1 |
| {a,d} | 2 |
| {b,c} | 3 |
| {b,d} | 4 |
| {c,d} | 3 |

| L2    |   |
|-------|---|
| {a,b} | 3 |
| {b,c} | 3 |
| {b,d} | 4 |
| {c,d} | 3 |

| I     |
|-------|
| {e}   |
| {a,c} |
| {a,d} |

$k = 3$

| Transacciones |
|---------------|
| {a,b,c,d}     |
| {a,b,d}       |
| {a,b}         |
| {b,c,d}       |
| {b,c}         |
| {c,d}         |
| {b,d}         |
| {b,e}         |
| {d,e}         |





# Apriori: Ejemplo

| L1  |   |
|-----|---|
| {a} | 3 |
| {b} | 7 |
| {c} | 4 |
| {d} | 6 |

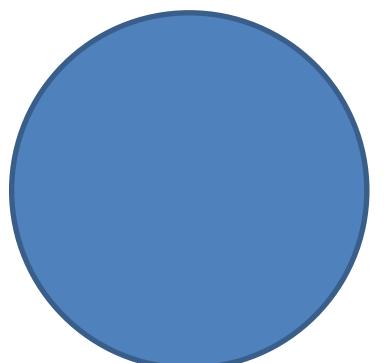
| L2    |   |
|-------|---|
| {a,b} | 3 |
| {b,c} | 3 |
| {b,d} | 4 |
| {c,d} | 3 |

| C       |   |
|---------|---|
| {b,c,d} | 2 |

| Transacciones |
|---------------|
| {a,b,c,d}     |
| {a,b,d}       |
| {a,b}         |
| {b,c,d}       |
| {b,c}         |
| {c,d}         |
| {b,d}         |
| {b,e}         |
| {d,e}         |

| I     |
|-------|
| {e}   |
| {a,c} |
| {a,d} |

$$k = 3$$



**Muchas gracias  
por vuestra  
atención.**

**Carlos Moreno Morera**  
Consultor de IA en IBM  
[carmor06@ucm.es](mailto:carmor06@ucm.es)

