



# Machine Learning con Python

Docente: M.Sc. Helmer Fellman Mendoza Jurado.



## Definición de Problema

Un problema es una circunstancia en la que se genera un obstáculo al curso normal de las cosas. Su etimología nos demuestra que un problema es aquel que requiere de solución. A nivel social, el concepto más genérico de problema puede ser vertido en cualquier campo, porque en teoría, problemas existen en todos lados. La falta de razonamiento no es un declive en la orientación del concepto, ejemplo, los animales de cualquier especie pueden afrontar circunstancias en las que se vean comprometidas su salud o incluso su vida y eso es un problema.

# **PROBLEMAS**

Vivimos inmersos de **problemas en la vida cotidiana**, tanto que desde un punto de vista más dogmático y filosófico se podría aseverar que son parte de nuestro ser y que nuestro propósito en la vida consiste en esencia en la **resolución de problemas y circunstancias que se presentan en el correcto curso de las cosas.**

# Características intrínsecas de los problemas

Que es propio o característico de un problema, que se expresa por sí misma y no depende de las circunstancias.

- ¿Puede el problema descomponerse en sub-problemas?, hablamos de Descomposición.
- ¿Puedo ignorar o deshacer lo ejecutado? (Ignorable, Recuperable, Irrecuperable), hablamos de Reversibilidad.
- ¿Se conocen las consecuencias exactas de la ejecución de cada acción? (consecuencia cierta, consecuencia incierta), hablamos de Universo.
- ¿La solución es absoluta o relativa? (alguna solución, mejor solución), hablamos de Bondad de la solución.
- ¿La solución es ruta o meta? (busco un camino, busco un estado), hablamos de Tipo de solución.

# Ejemplo de Descomposición

¿Puede el problema descomponerse en sub-problemas?

*Tengo que aprobar una asignatura de Inteligencia Artificial I (Macro problema)*

**Subproblema 1:** Aprobar Parcial 1

**Subproblema 2:** Aprobar Parcial 2

**Subproblema 3:** Aprobar el Proyecto

**Subproblema 4:** Cumplir y aprobar las actividades prácticas

**Subproblema 5:** Cumplir con % de asistencia

# Ejemplo de Descomposición

¿Puedo ignorar o deshacer lo ejecutado? (Ignorable, Recuperable, Irrecuperable)

## **PROBLEMA IGNORABLE**

Resolución de un teorema, arranco hoja y comienzo otra vez

## **PROBLEMA RECUPERABLE**

Error en código de programación, modifco líneas de código

## **PROBLEMA IRRECUPERABLE**

Cualquier juego con contrincante

# Ejemplo de Universo de problemas

¿Se conocen las consecuencias exactas de la ejecución de cada acción? (consecuencia cierta, consecuencia incierta)

## ***UNIVERSO CIERTO***

Juego de tipo solitario, sé exactamente como quedará la situación luego de una jugada.

## ***UNIVERSO INCIERTO***

Juego con contrincante, no sé como responderá el otro jugador por tanto, no sé exactamente como repercute mi jugada.

# Ejemplo de Bondad de la Solución

¿La solución es absoluta o relativa? (alguna solución, mejor solución)

## ***SOLUCIÓN ABSOLUTA***

Dado un viaje, descubrir la ruta que implique, el menor costo en combustible o en tiempo.

## ***SOLUCIÓN RELATIVA***

Dado un viaje, descubrir una ruta que implique, llegar al destino desde la ciudad de origen.

# Ejemplo de tipo de solución

¿La solución es ruta o meta? (busco un camino, busco un estado)

## ***SOLUCIÓN RUTA***

Dado un viaje, descubrir las ciudades por las que debo pasar para llegar a destino

## ***SOLUCIÓN META***

Encontrar un número telefónico.

# Inteligencia Artificial e Inteligencia Humana

«Somos seres más emocionales que racionales»



## Su arquitectura es distinta

Una máquina dotada de **inteligencia artificial** tiene una serie de puertos de entrada y salida de datos que podemos identificar fácilmente.



## La importancia del contexto

Nuestros cerebros orgánicos se adaptan como un guante a cada situación, a pesar de que cada una de las situaciones que vivimos sean únicas.



## Su funcionamiento es distinto

En cualquier estructura de inteligencia artificial se puede diferenciar el canal por el que viajan los datos (hardware) y la información propiamente dicha. En un cerebro, en cambio, la distinción entre información y el medio material por el que viaja no existe.



## La Inteligencia Artificial necesita regularidad

Los sistemas de inteligencia artificial necesitan estar montados de una manera muy concreta para poder ejecutar órdenes y hacer que la información pase de un lugar a otro de la manera correcta.



## Los datos con los que trabaja el cerebro no se pueden almacenar

Una consecuencia de que no distinguimos entre canal e información es que tampoco existen grandes depósitos de datos en nuestra cabeza. Es por eso que nunca recordamos algo de la misma forma, siempre hay pequeñas variaciones.



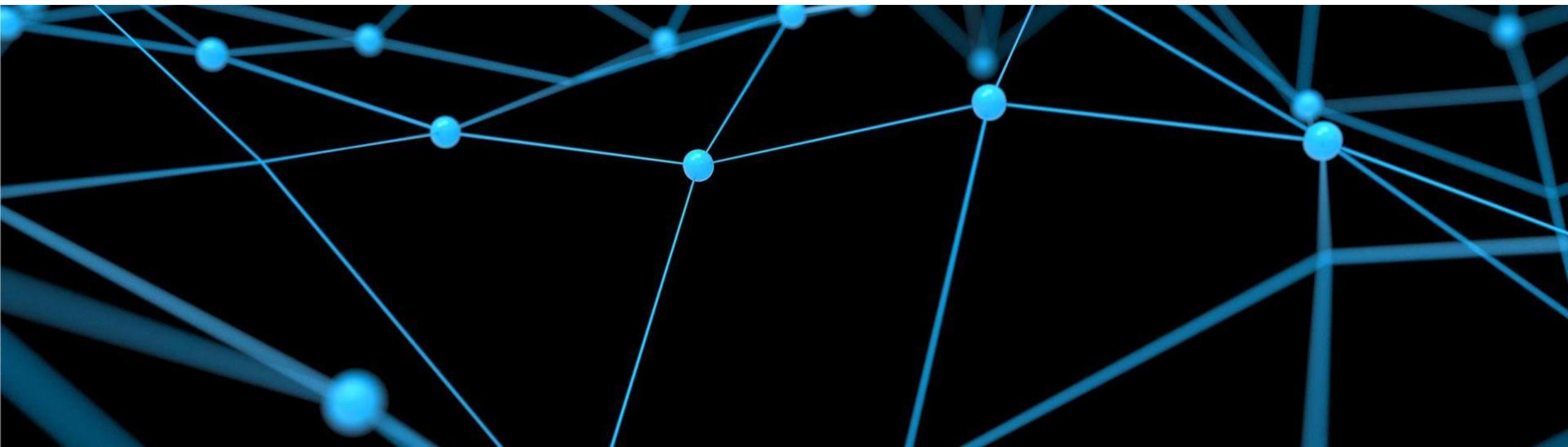
## Su origen es distinto

Cualquier sistema de inteligencia artificial ha sido construido por uno o más agentes intencionales: científicos, programadores, etc. Nuestros cerebros, sin embargo, han sido tallados por la evolución.

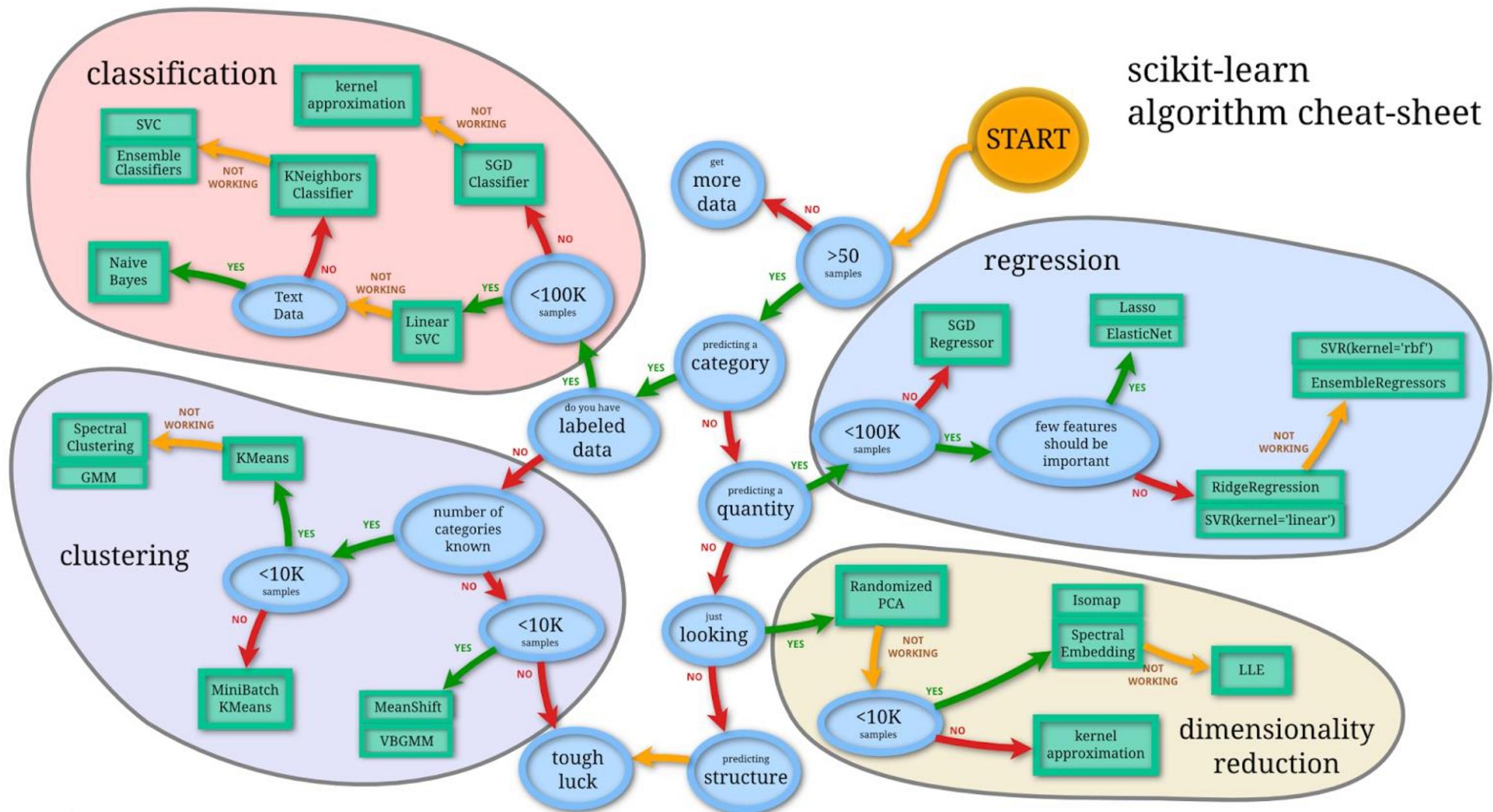
**Inteligencia Artificial robusta o Strong AI:** trata sobre una inteligencia real en el que las máquinas tienen similar capacidad cognitiva que los humanos, algo que, como los expertos se aventuran a predecir, aún quedan años para alcanzar. Digamos que esta es la Inteligencia de la que soñaban los pioneros del tema con sus vetustas válvulas.

**Inteligencia Artificial aplicada Weak AI (Narrow AI o Applied AI):** aquí es donde entran el uso que hacemos a través de algoritmos y aprendizaje guiado con el Machine Learning y el Deep Learning.

## Discutiendo entre el Machine Learning y el Deep Learning ¿A qué nos referimos con cada uno?



# scikit-learn algorithm cheat-sheet



# Algoritmos de Regresión

Linealidad para los amigos

Los modelos de machine learning para regresión lineal son algoritmos que utilizan técnicas de aprendizaje automático para predecir valores numéricos continuos en función de variables de entrada. La regresión lineal es un enfoque estadístico que establece una relación lineal entre la variable objetivo (o variable dependiente) y una o más variables predictoras (o variables independientes).

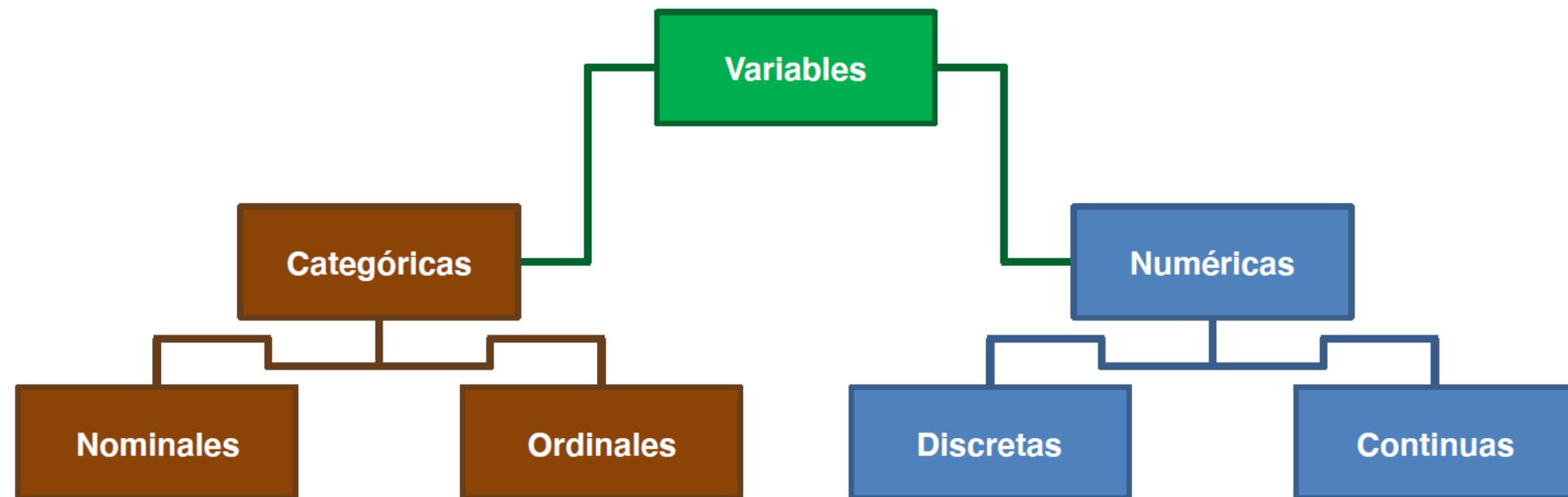
# Regresión Lineal Simple

El más simple de todos

En estadística la regresión lineal o ajuste lineal es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente  $Y$ , las variables independientes  $X_i$  y un término aleatorio  $\varepsilon$ .



# Variables



Hombre, Mujer  
Rojo, Verde, Azul

Pequeño, Mediano, Grande  
A, B, C (notas)

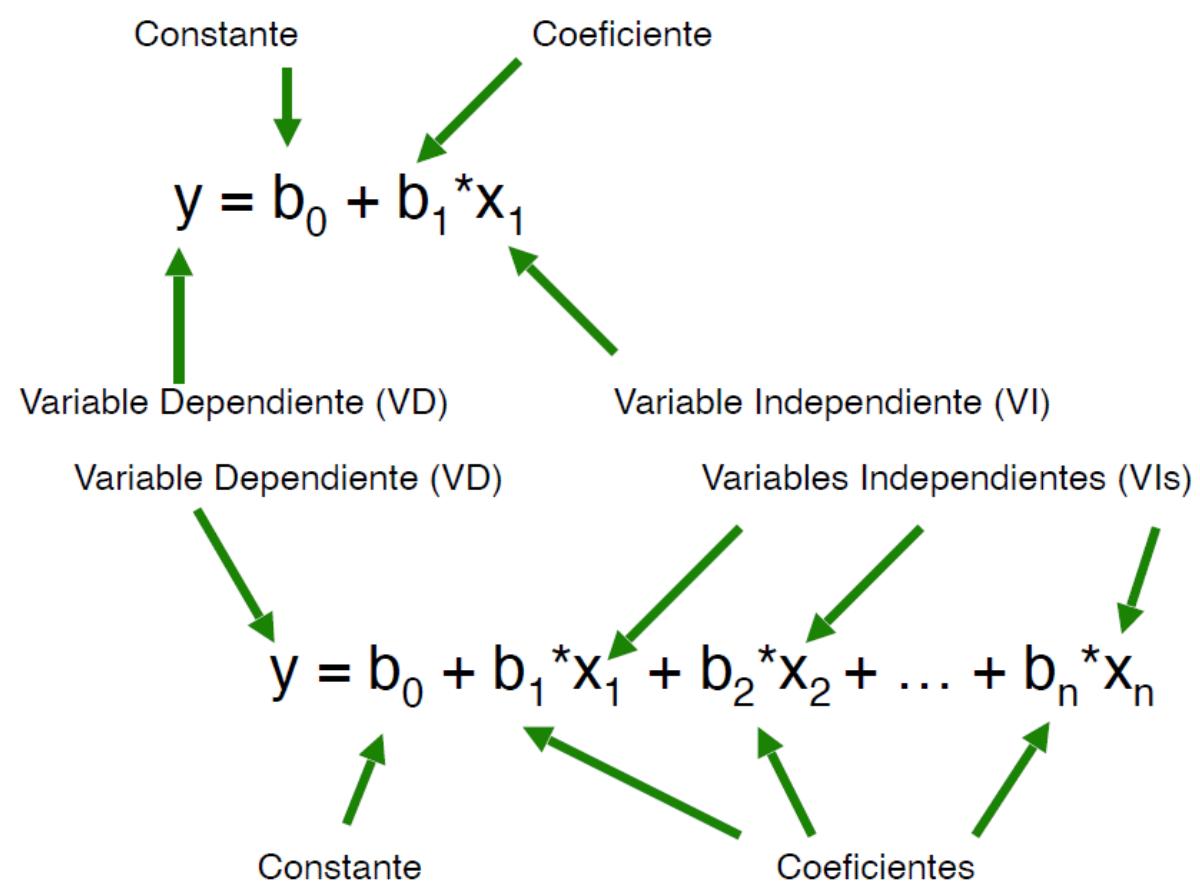
1, 2, 3 empleados  
568 personas

Edad  
Altura

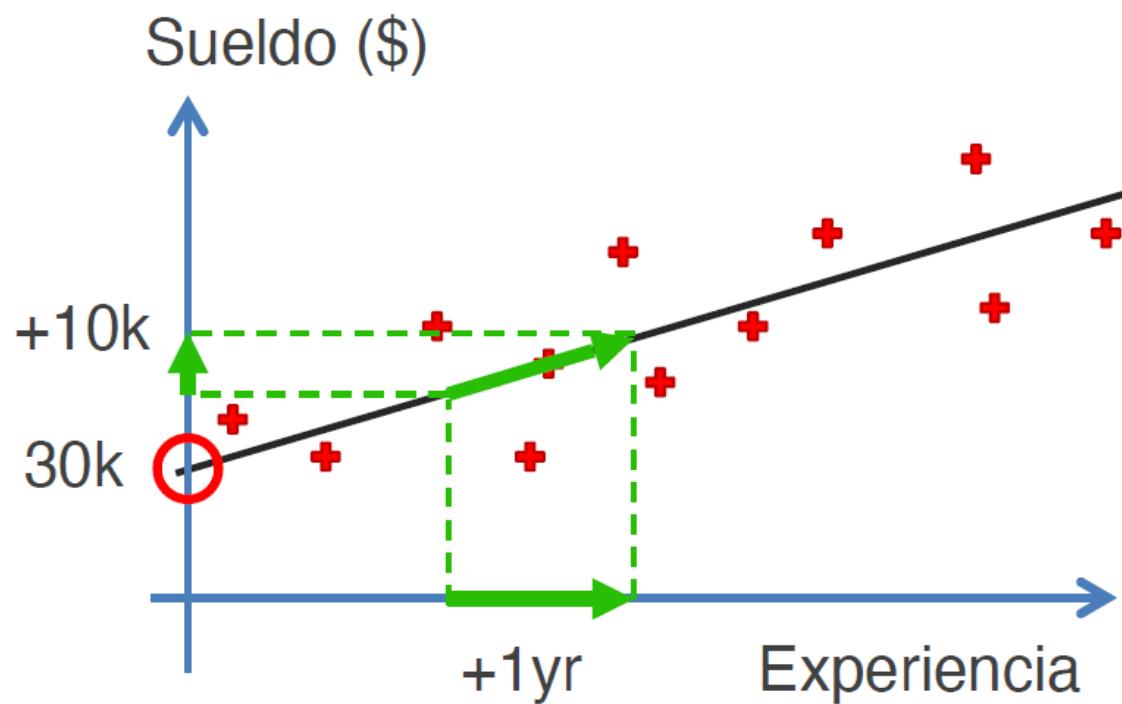
# Tipos de Regresiones Lineales

Regresión Lineal Simple

Regresión Lineal Múltiple



# Regresión Lineal Simple



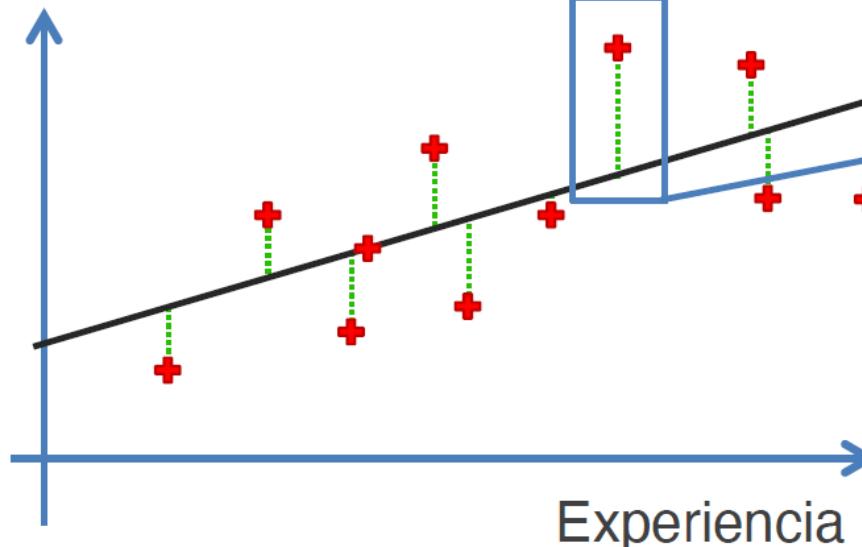
$$y = b_0 + b_1 * x$$

$$\text{Sueldo} = \textcolor{red}{b_0} + \textcolor{green}{b_1} * \text{Experiencia}$$

# Método de los Mínimos Cuadrados

Regresión Lineal Simple

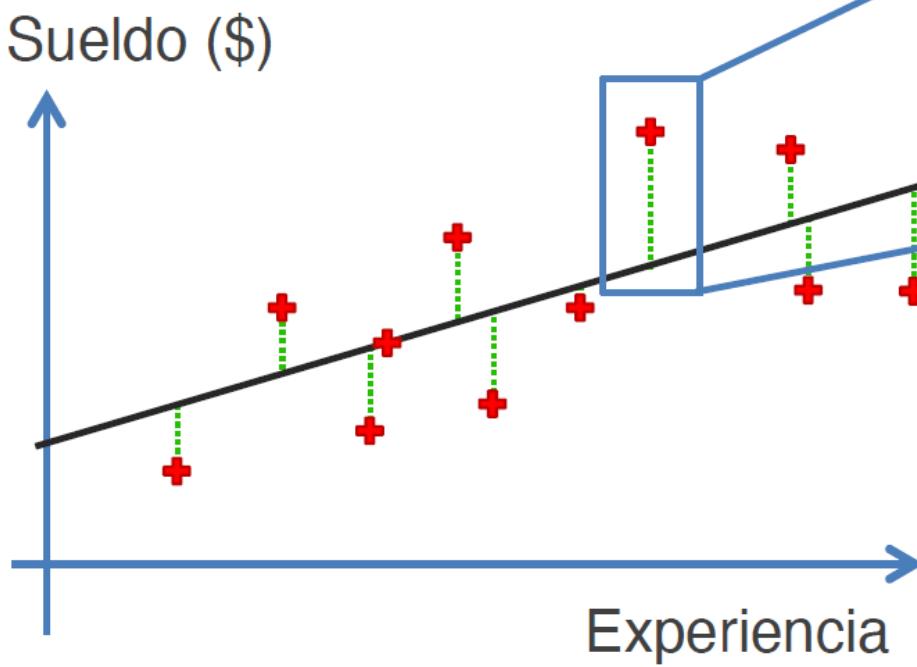
Sueldo (\$)



$$\min \sum_i (y_i - \hat{y}_i)^2$$

# Método de los Cuadrados

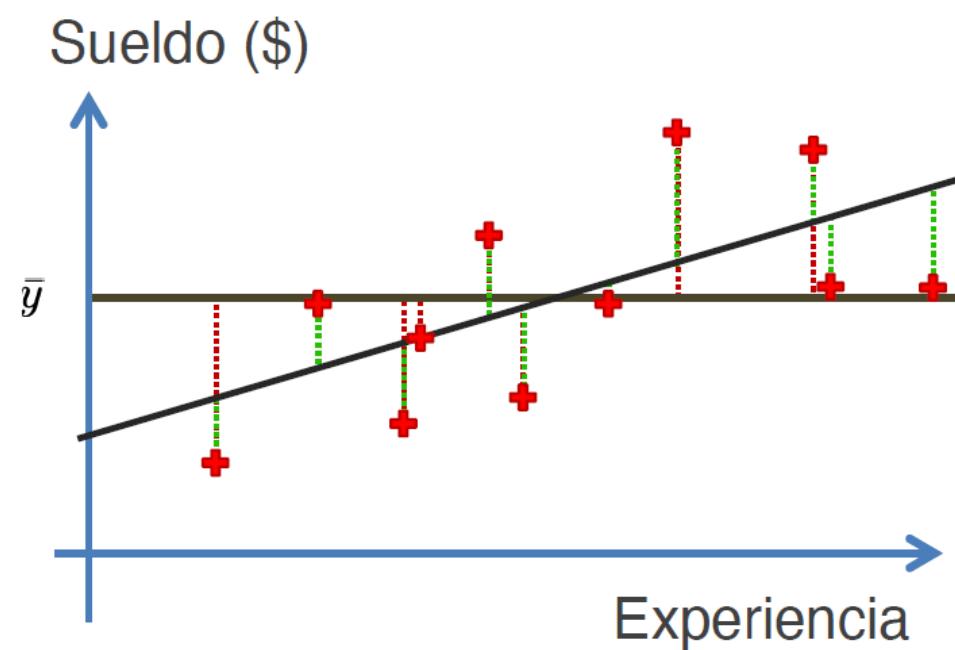
Regresión Lineal Simple:



$$\sum_i (y_i - \hat{y}_i)^2$$

## Método de los Cuadrados

Regresión Lineal Simple:



$$SS_{\text{res}} = \sum_i (y_i - \hat{y}_i)^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

## Método de los Cuadrados

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$R^2$  – Bondad de Ajuste  
(cuanto más grande mejor)

$$y = b_0 + b_1 * x_1$$

$$y = b_0 + b_1 * x_1 + b_2 * x_2$$

$$SS_{res} \rightarrow \text{Min}$$

**Problema:**

$$+ b_3 * x_3$$

$R^2$  nunca va a decrecer!

## Método de los Cuadrados Ajustados

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$\text{Adj } R^2 = 1 - (1 - R^2) \cdot \frac{n - 1}{n - p - 1}$$

p - número de variables regresoras  
n – tamaño de la muestra

# Una Advertencia

Restricciones de la Regresión Lineal

## *Linealidad...*

Se entiende por linealidad la capacidad de un método analítico de obtener resultados proporcionales a la concentración de analito en la muestra dentro de un intervalo determinado.

La **linealidad** indica si el sistema de medición tiene la misma exactitud para todos los valores de referencia.

01

### **Homocedasticidad**

En estadística se dice que un modelo predictivo presenta homocedasticidad cuando la varianza del error condicional a las variables explicativas es constante a lo largo de las observaciones.

02

### **Normalidad multivariable**

En probabilidad y estadística, una distribución normal multivariante, también llamada distribución gaussiana multivariante, es una generalización de la distribución normal unidimensional a dimensiones superiores.

03

### **Independencia de los errores**

Para datos transversales en las que se supone que las observaciones son independientes entre sí, probablemente no sucederá que éstas se encuentren relacionadas entre sí.

04

### **Ausencia de multicolinealidad**

Se presenta cuando no existe una fuerte correlación entre variables explicativas del modelo. La correlación ha de ser fuerte, ya que siempre existirá correlación entre dos variables explicativas en un modelo, es decir, la no correlación de dos variables es un proceso idílico, que sólo se podría encontrar en condiciones de laboratorio.

# Variables Dummy

No todo son  
números

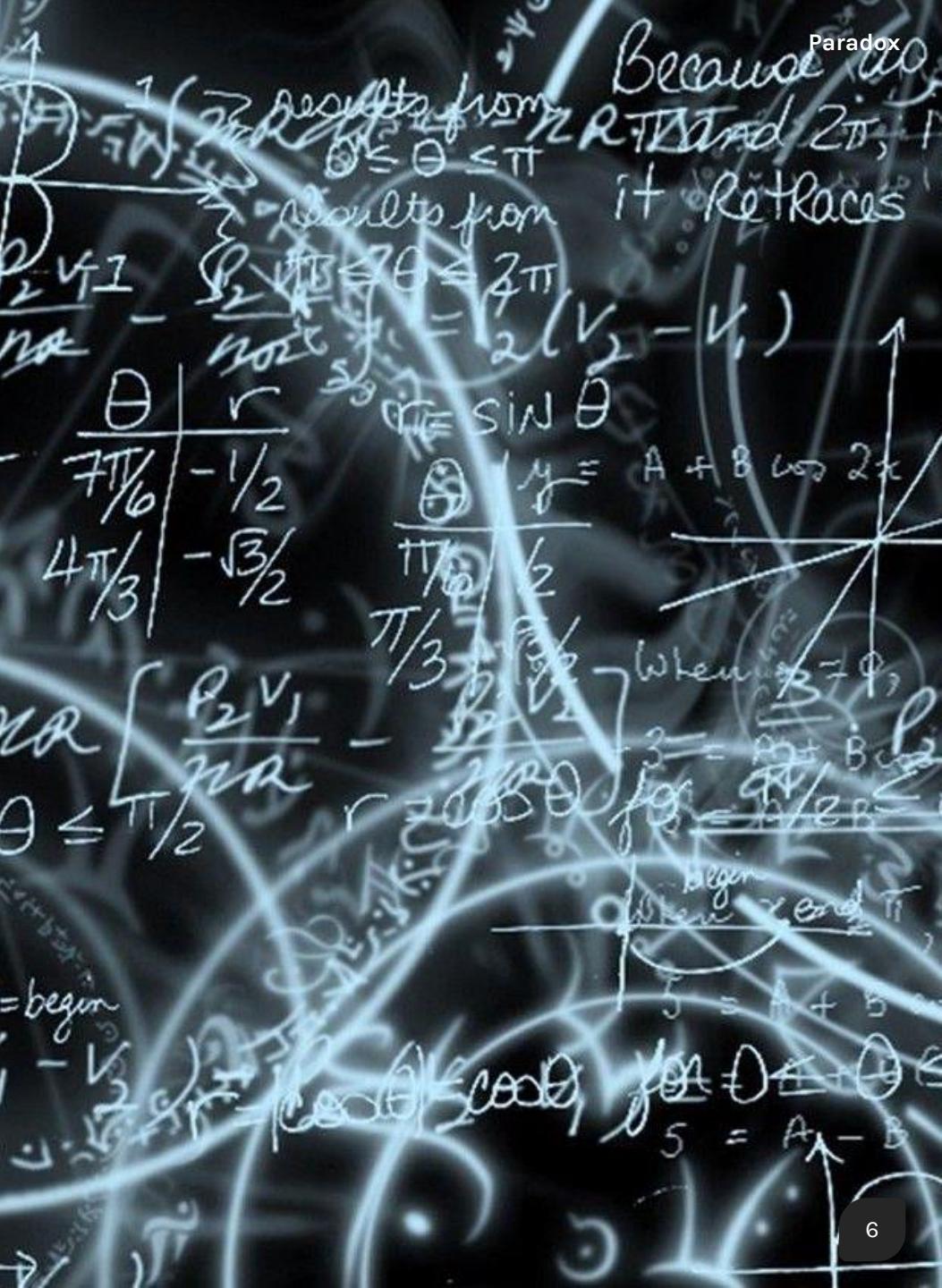
*Algunas de las variables son por su naturaleza propia Cualitativas...*

Estas variables reciben el denominativo de variables **dummy**, artificiales o indicadoras.

En cualquiera de los casos, cuando la variable solo presenta dos categorías, se trata de una variable **dicotómica**.

No obstante, una variable cualitativa puede presentar más de dos categorías, es decir, puede ser **multicategórica**, por ejemplo:

1. Soltero
2. Casado (civil, iglesia o ambos)
3. Unión
4. Separado o divorciado
5. Viudo



## Variables Dummy

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

New York	California
1	0
0	1
0	1
1	0
0	1

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + ???$$

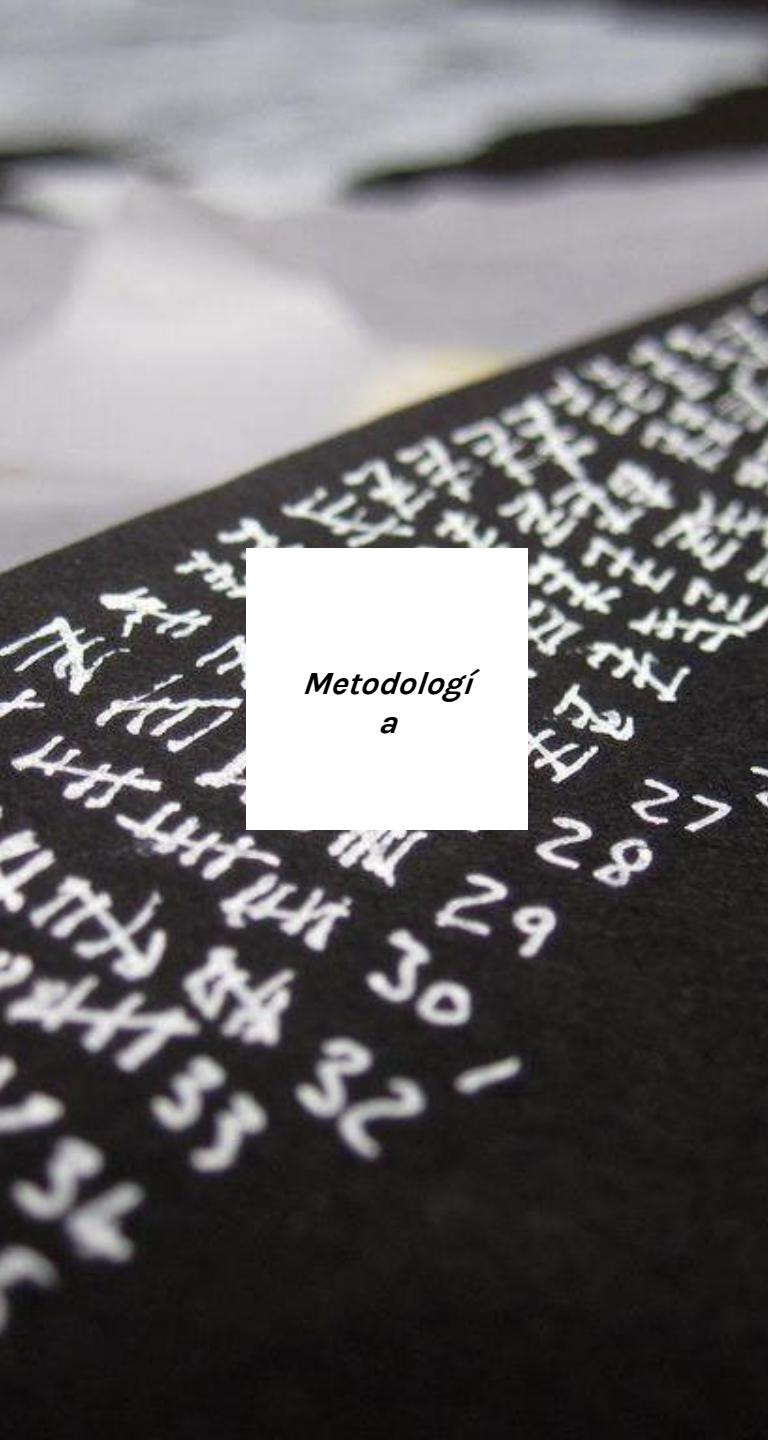
Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

## Variables Dummy

New York	California
1	0
0	1
0	1
1	0
0	1

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + b_4 * D_1$$





# Construir un Modelo “Eliminación hacia Atras”

Una receta de Cocina

## Paso 1



Seleccionar el nivel de significación para permanecer en el modelo (p.e. SL = 0.05)

## Paso 2



Se calcula el modelo con todas las posibles variables predictoras

## Paso 3



Considera la variable predictora con el p-valor más grande. Si  $P > SL$ , entonces vamos al PASO 4, si no vamos a FIN

## Paso 4



Se elimina la variable predictora

## Paso 5

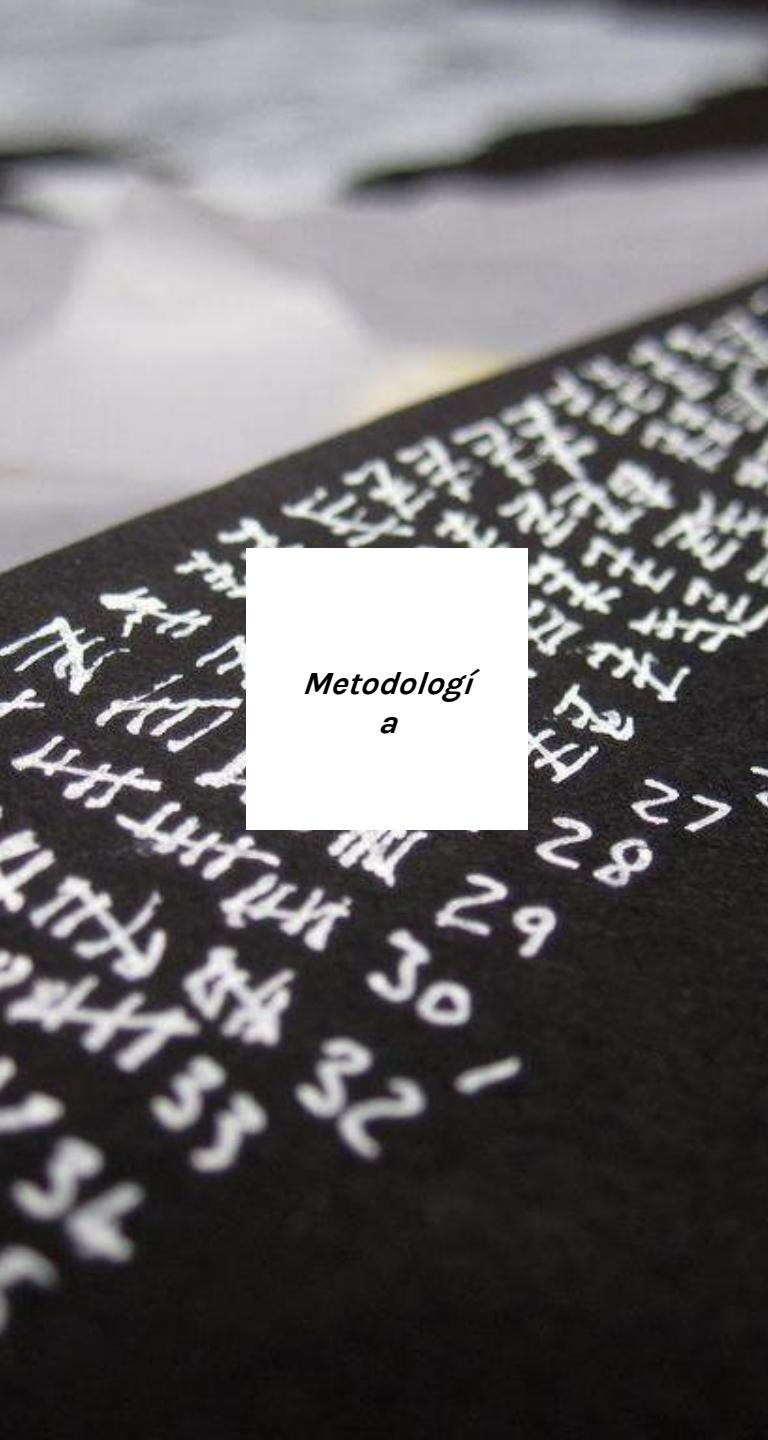


Ajustar el modelo sin dicha variable (si aún existe un valor muy grande para la variable predictora, volvemos al paso 3, generamos un bucle)

## Paso 6



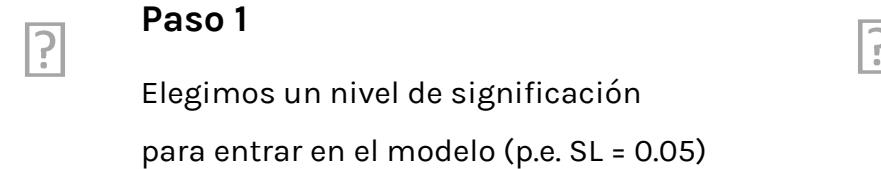
Finaliza el modelo



# Construir un Modelo “Eliminación hacia Adelante”

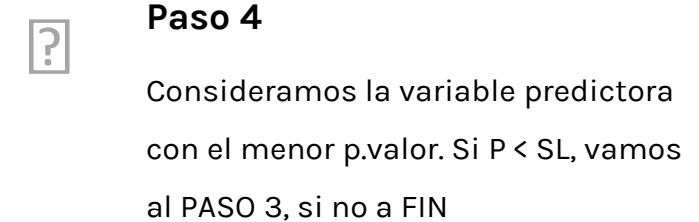
Una receta de Cocina

## Paso 1



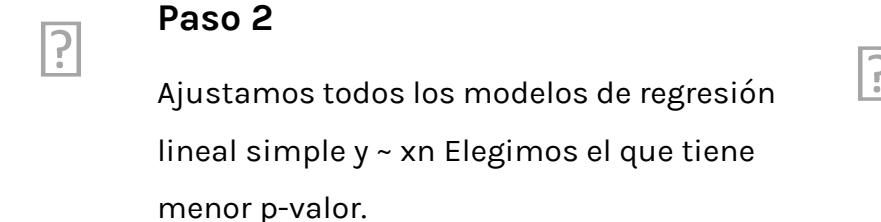
Elegimos un nivel de significación para entrar en el modelo (p.e. SL = 0.05)

## Paso 4



Consideramos la variable predictora con el menor p.valor. Si  $P < SL$ , vamos al PASO 3, si no a FIN

## Paso 2



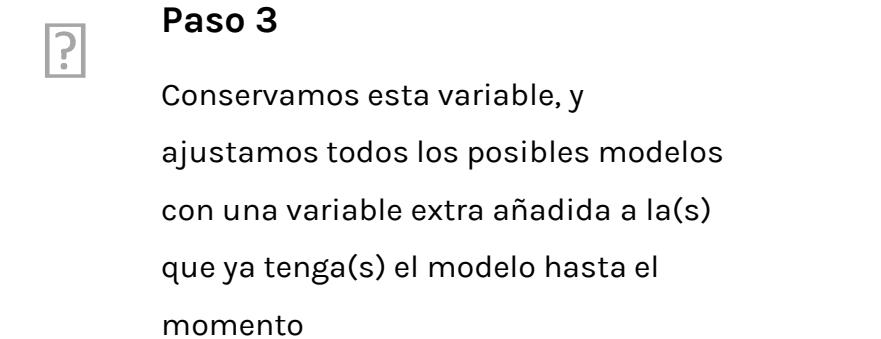
Ajustamos todos los modelos de regresión lineal simple  $y \sim xn$  Elegimos el que tiene menor p-valor.

## Paso 5

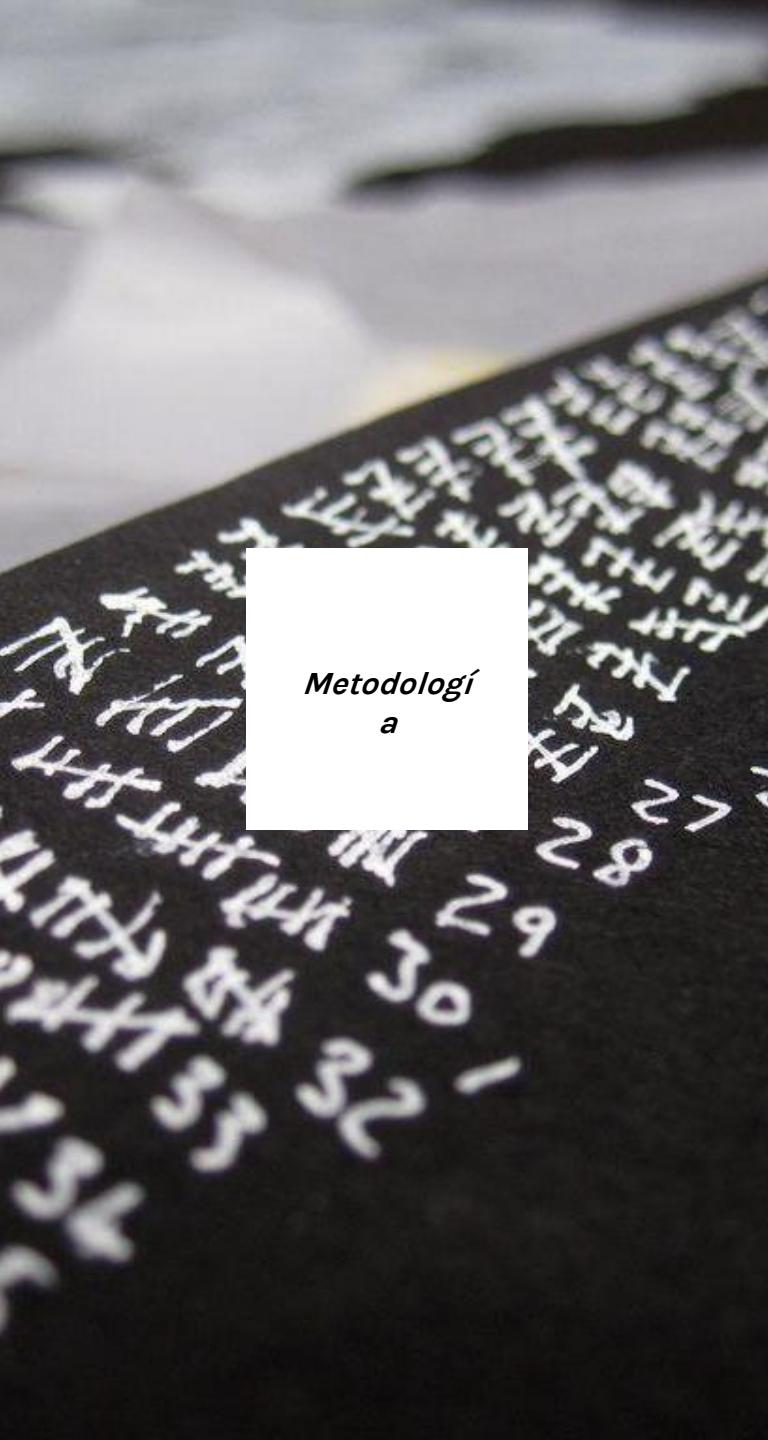


Finaliza el modelo

## Paso 3



Conservamos esta variable, y ajustamos todos los posibles modelos con una variable extra añadida a la(s) que ya tenga(s) el modelo hasta el momento



# Construir un Modelo “Todos los modelos posibles”

Una receta de Cocina



## Paso 1

Seleccionar un criterio de bondad de ajuste (p.e. criterio de Akaike)



## Paso 3

Seleccionar el modelo con el mejor criterio elegido



## Paso 2

Construir todos los posibles modelos de regresión:  $2N-1$  combinaciones en total



## Paso 4

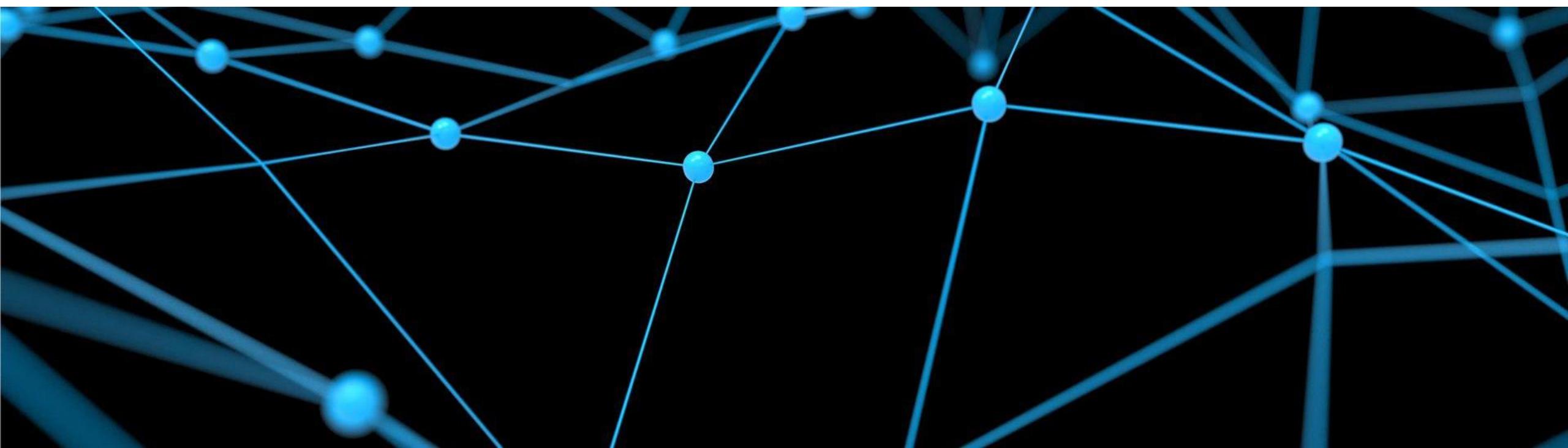
Finaliza el modelo

**Por ejemplo:**  
**10 columnas significan**  
**1,023 modelos**

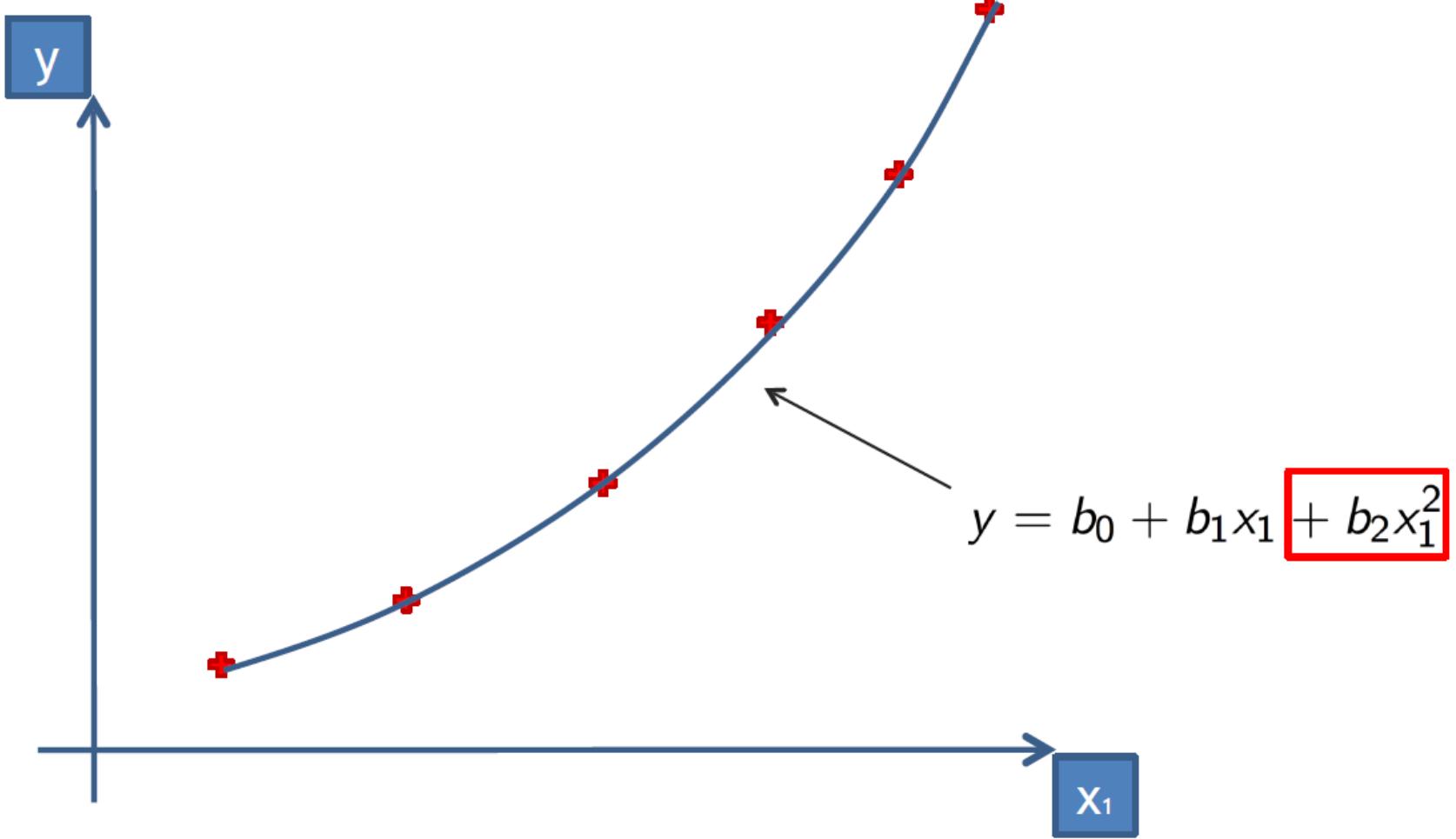
La regresión polinómica es una extensión de la regresión lineal en la que se permite ajustar una relación no lineal entre las variables predictoras y la variable objetivo. En lugar de utilizar una función lineal, la regresión polinómica utiliza una función polinómica para modelar la relación entre las variables.

## Regresión Polinómica

Una línea no tan lineal



# Regresión Lineal Polinómica



# SVM para Regresión

Máquinas de Soporte  
Vectorial

Las Máquinas de Soporte Vectorial (Support Vector Machines, SVM) son un tipo de algoritmo de aprendizaje supervisado utilizado tanto para problemas de clasificación como para regresión. SVM se basa en el concepto de encontrar un hiperplano óptimo que separe o clasifique de manera óptima los puntos de datos en diferentes clases.

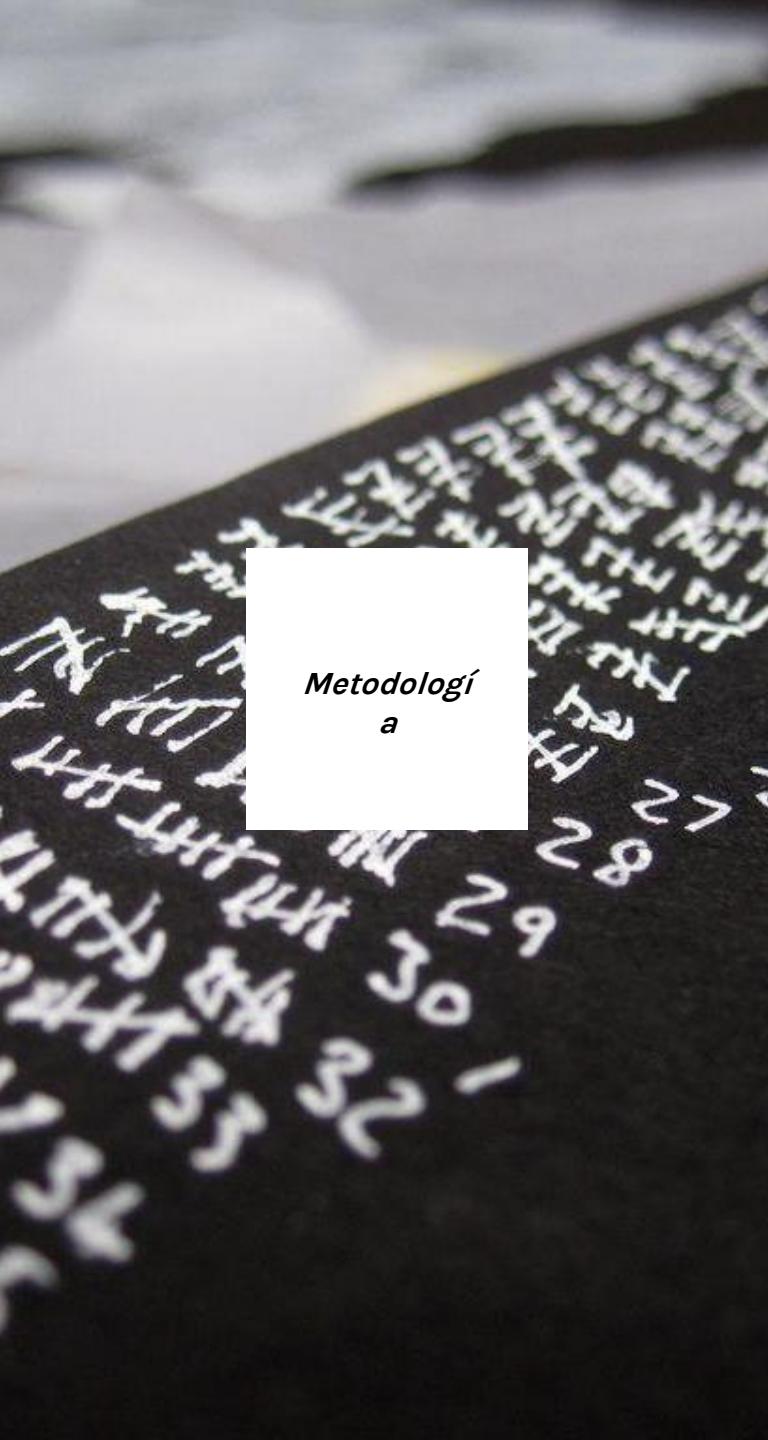
# SVM para regresión

Un poco de matemática compleja

## *¿Cómo funciona un SVR?*

En un problema de clasificación, los vectores X se utilizan para definir un hiperplano que separe las dos categorías en nuestra solución.

Estos vectores se utilizan para llevar a cabo la regresión lineal. Los vectores más cercanos al punto de test se llaman vectores de soporte. Podemos evaluar nuestra función en cualquier lugar, por lo que cualquier vector podría estar más cerca de nuestra ubicación de evaluación de prueba.



# Construir un modelo de SVR

Una receta de

Cocina

**Paso 1**

Tener un conjunto de entrenamiento

$$\mathcal{T} = \{\vec{X}, \vec{Y}\}$$

**Paso 2**

Elegir un núcleo y sus parámetros así como llevar a cabo cualquier regularización que sea necesaria.

**Paso 3**

Crear la matriz de correlaciones

$$K$$

**Paso 4**

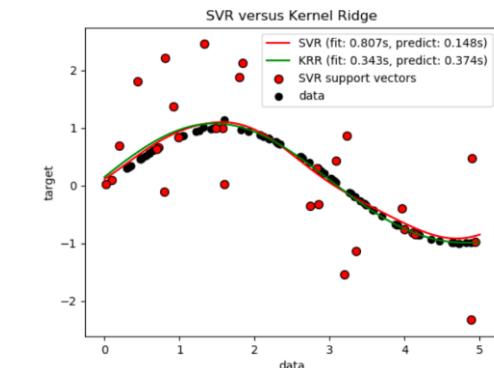
Entrenar el modelo, de forma exacta o aproximada para obtener los coeficientes de con

$$\vec{\alpha} = \{\alpha_i\}$$

**Paso 5**

Utilizar estos coeficientes para crear un estimador

$$f(\vec{X}, \vec{\alpha}, x^*) = y^*$$



## Matriz de Correlaciones

$$K_{i,j} = \exp \left( \sum_k \theta_k |x_k^i - x_k^j|^2 \right) + \varepsilon \delta_{i,j}$$

## El núcleo de SVM

*El corazón de todo...*

Lo siguiente es elegir un núcleo:

- Lineal  $\langle x, y \rangle$
- No Lineal  $\langle \varphi(x), \varphi(y) \rangle = K(x, y)$
- Gaussiano

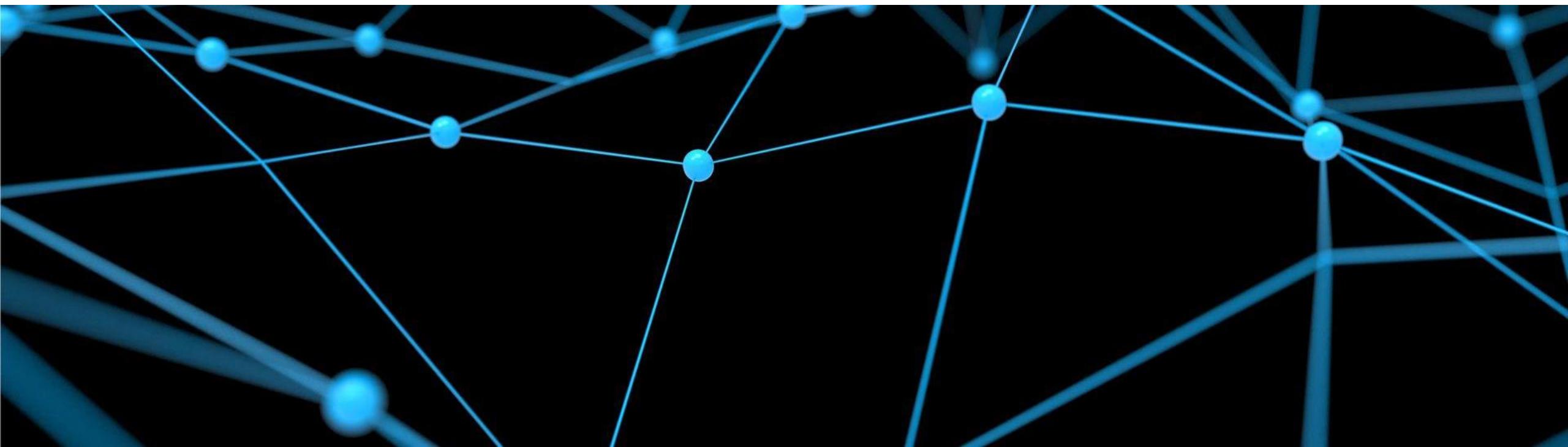
Regularización

- Ruido

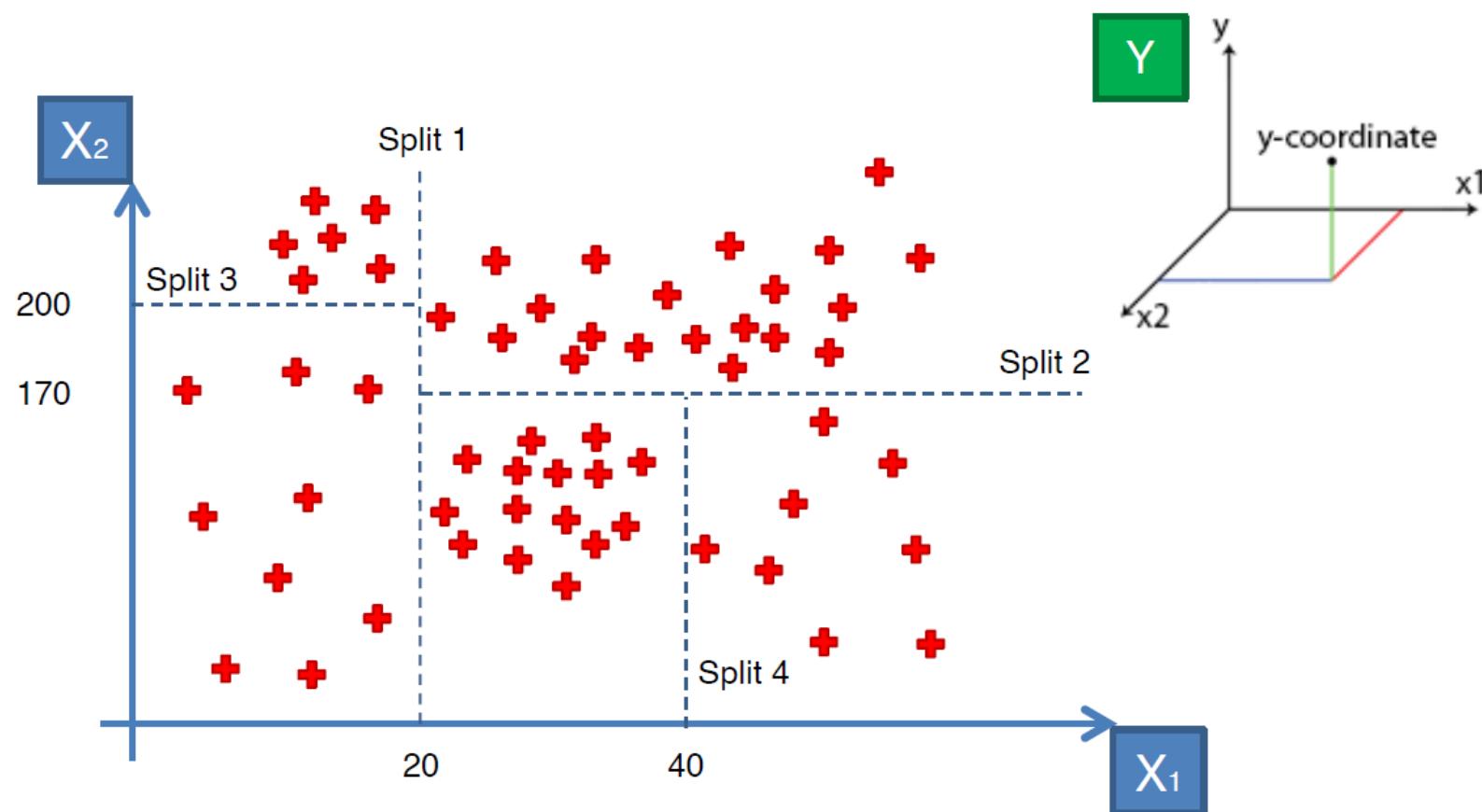
Aprendizaje basado en **árboles de decisión** utiliza un árbol de decisión como un modelo predictivo que mapea observaciones sobre un artículo a conclusiones sobre el valor objetivo del artículo.

## Arbóles de Decisión

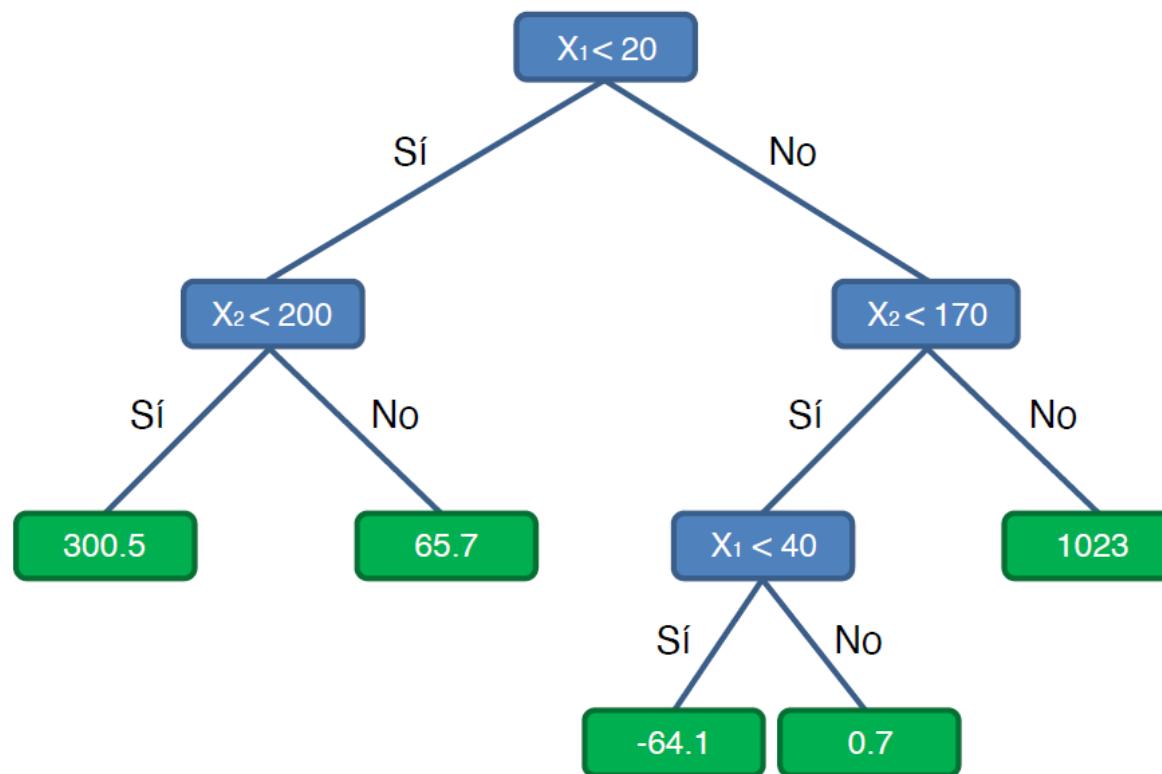
Arboles de Regresión



# Idea de los Árboles de Regresión



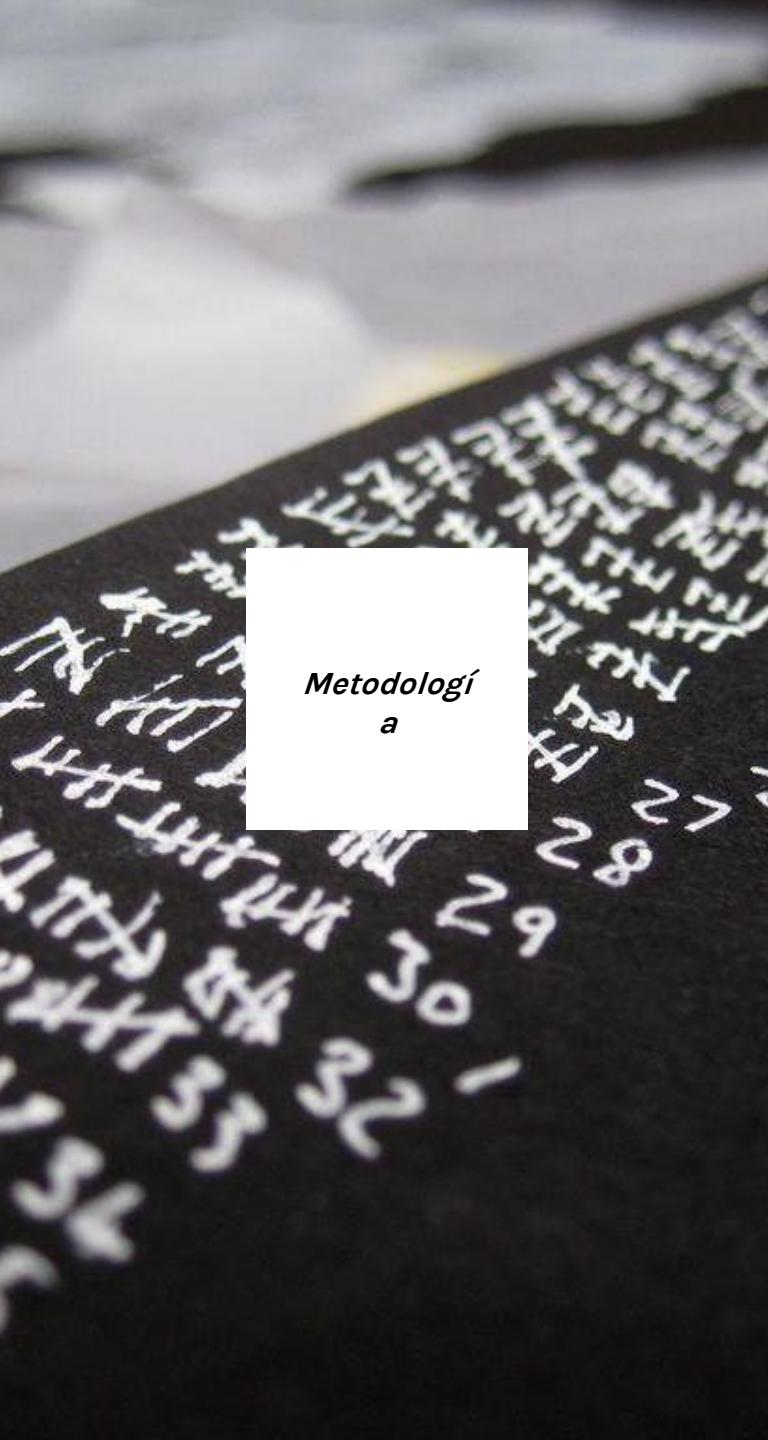
# Idea de los Árboles de Regresión



# Bosques Aleatorios

Arboles en cantidad

Es la combinación de **árboles predictores** tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.



# Idea de los Bosques Aleatorios

Una receta de

Cocina

## Paso 1



Elegir un número aleatorio K de puntos de datos del Conjunto de Entrenamiento.

## Paso 2



Construir el Árbol de Decisión asociado a esos K Puntos de Datos.



## Paso 3

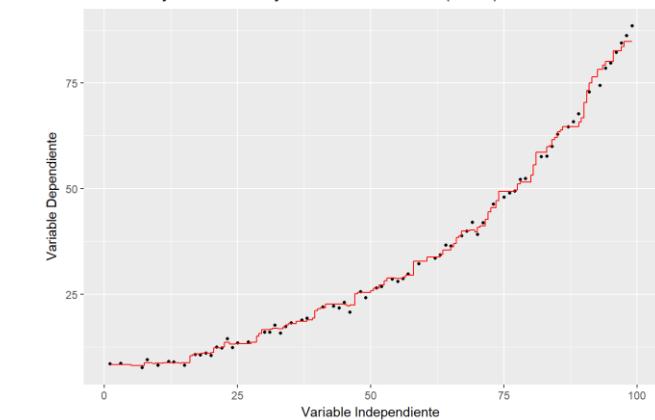
Elegir el número Ntree de árboles que queremos construir y repetimos los PASOS 1 y 2.

## Paso 4



Para un nuevo punto de datos, hacer que cada uno de los Ntree árboles haga una predicción del valor de Y para el punto en cuestión, y asigne al nuevo punto la predicción final basada en el promedio de todas las

Curva de Ajuste sobre Conjunto de Entrenamiento (ntrain)



# Evaluar el Rendimiento en Modelos de Regresión

```
Call:
lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend +
    State, data = dataset)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-33504	-4736	90	6672	17338

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	5.008e+04	6.953e+03	7.204	5.76e-09 ***
R.D.Spend	8.060e-01	4.641e-02	17.369	< 2e-16 ***
Administration	-2.700e-02	5.223e-02	-0.517	0.608
Marketing.Spend	2.698e-02	1.714e-02	1.574	0.123
State2	4.189e+01	3.256e+03	0.013	0.990
State3	2.407e+02	3.339e+03	0.072	0.943

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9439 on 44 degrees of freedom  
 Multiple R-squared: 0.9508, Adjusted R-squared: 0.9452  
 F-statistic: 169.9 on 5 and 44 DF, p-value: < 2.2e-16

Call:

```
lm(formula = Profit ~ R.D.Spend + Marketing.Spend, data = dataset)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-33645	-4632	-414	6484	17097

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4.698e+04	2.690e+03	17.464	<2e-16 ***
R.D.Spend	7.966e-01	4.135e-02	19.266	<2e-16 ***
Marketing.Spend	2.991e-02	1.552e-02	1.927	0.06 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9161 on 47 degrees of freedom  
 Multiple R-squared: 0.9505, Adjusted R-squared: 0.9483  
 F-statistic: 450.8 on 2 and 47 DF, p-value: < 2.2e-16

```
Call:
lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend,
    data = dataset)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-33534	-4795	63	6606	17275

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	5.012e+04	6.572e+03	7.626	1.06e-09 ***
R.D.Spend	8.057e-01	4.515e-02	17.846	< 2e-16 ***
Administration	-2.682e-02	5.103e-02	-0.526	0.602
Marketing.Spend	2.723e-02	1.645e-02	1.655	0.105

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9232 on 46 degrees of freedom  
 Multiple R-squared: 0.9507, Adjusted R-squared: 0.9475  
 F-statistic: 296 on 3 and 46 DF, p-value: < 2.2e-16

Call:

```
lm(formula = Profit ~ R.D.Spend, data = dataset)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-34351	-4626	-375	6249	17188

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4.903e+04	2.538e+03	19.32	<2e-16 ***
R.D.Spend	8.543e-01	2.931e-02	29.15	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9416 on 48 degrees of freedom  
 Multiple R-squared: 0.9465, Adjusted R-squared: 0.9454  
 F-statistic: 849.8 on 1 and 48 DF, p-value: < 2.2e-16



## Clasificación

La diferencia está en el tipo de resultado que queremos que la técnica de machine learning produzca.



## Clasificación

Cuando usamos clasificación, el resultado es una clase, entre un número limitado de clases. Con clases nos referimos a categorías arbitrarias según el tipo de problema.

Por ejemplo, si queremos detectar si un correo es spam o no, sólo hay 2 clases.



## Regresión

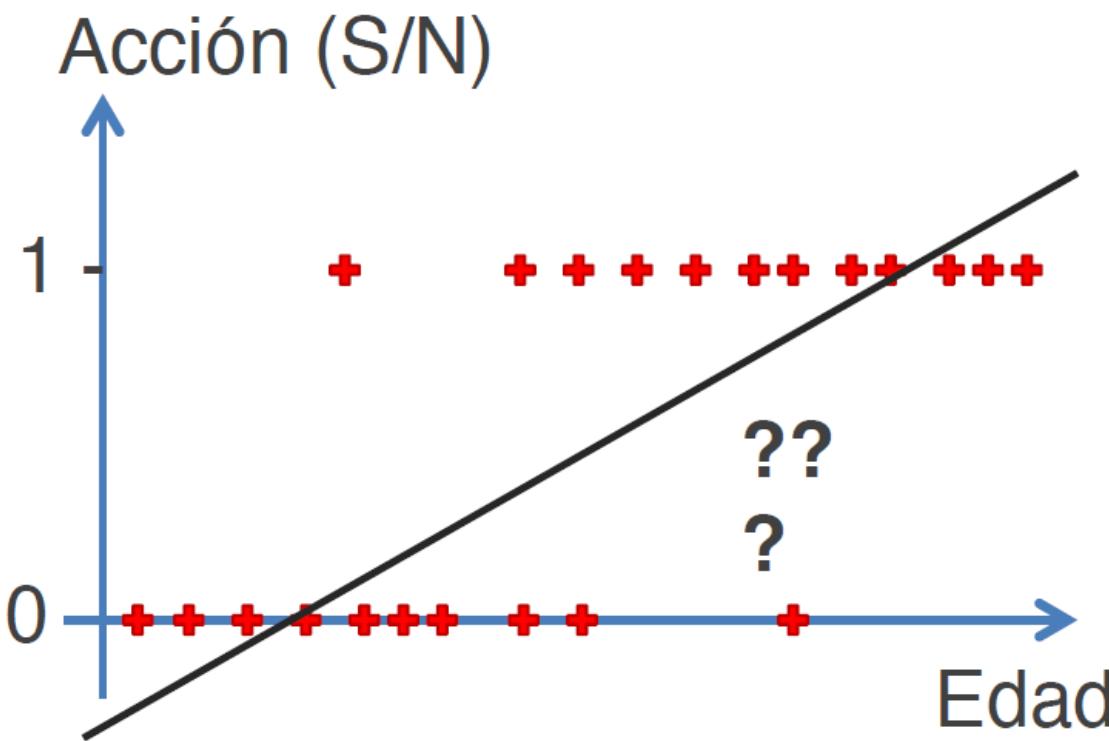
Cuando usamos regresión, el resultado es un número. Es decir, el resultado de la técnica de machine learning que estemos usando será un valor numérico, dentro de un conjunto infinito de posibles resultados.

# Regresión Logística

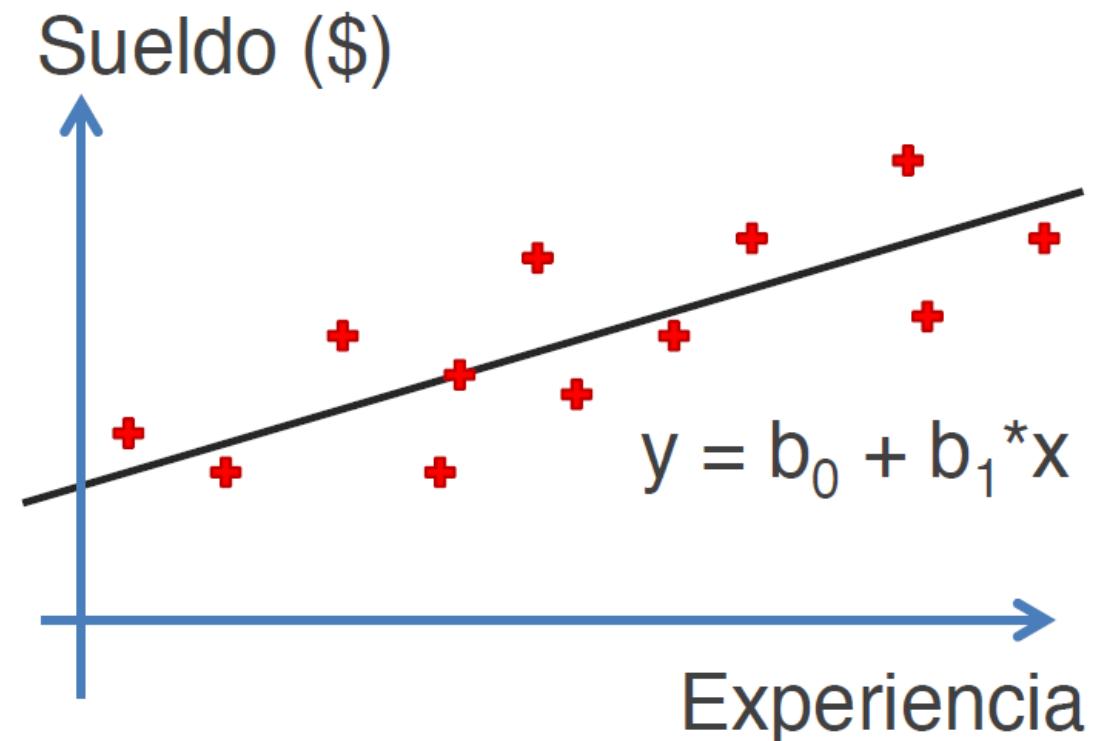
probabilidad de una variable cualitativa  
binaria

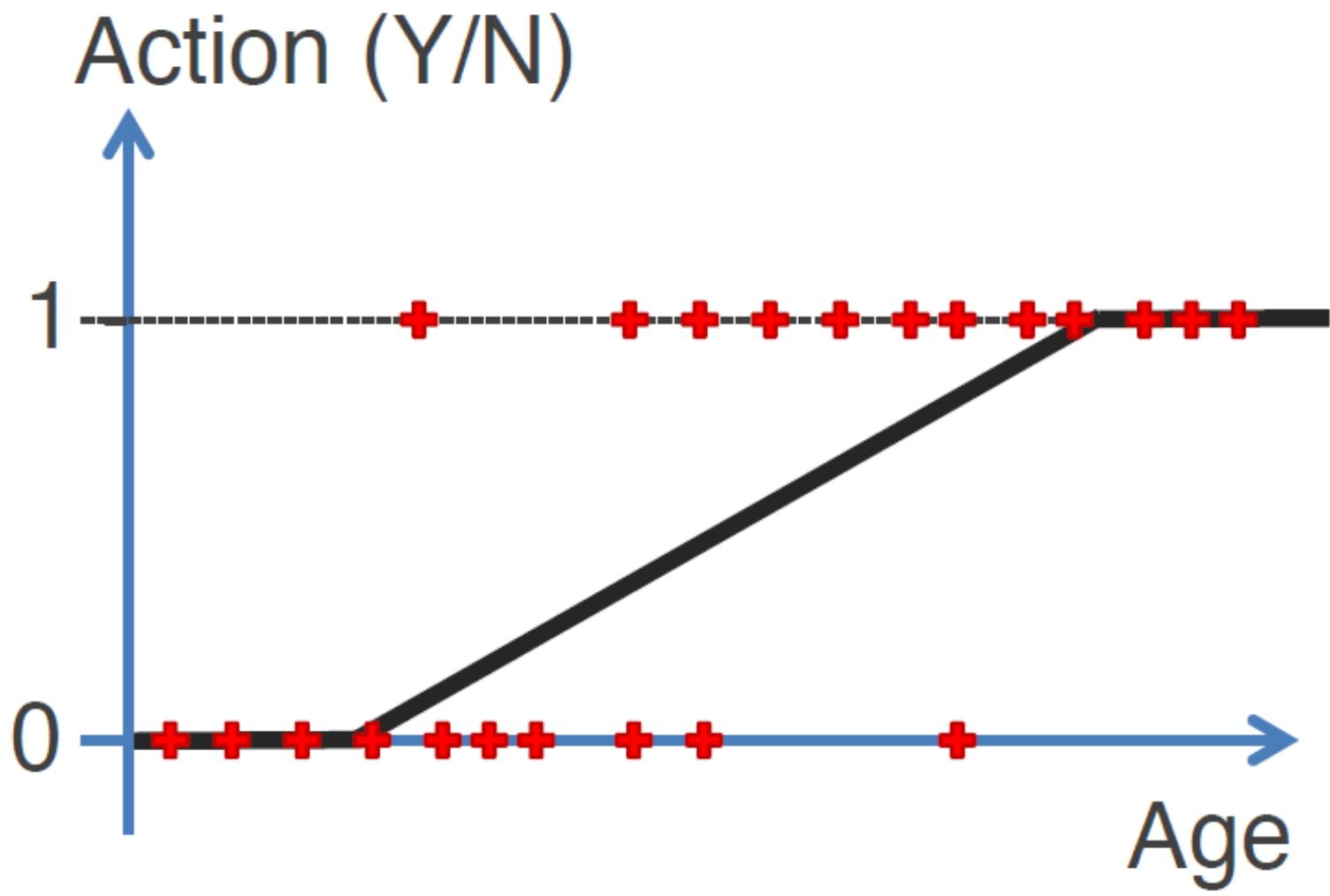
La **Regresión Logística** es un método estadístico para predecir clases binarias. El resultado o variable objetivo es de naturaleza dicotómica. Dicotómica significa que solo hay dos clases posibles. Por ejemplo, se puede utilizar para problemas de detección de cáncer o calcular la probabilidad de que ocurra un evento.

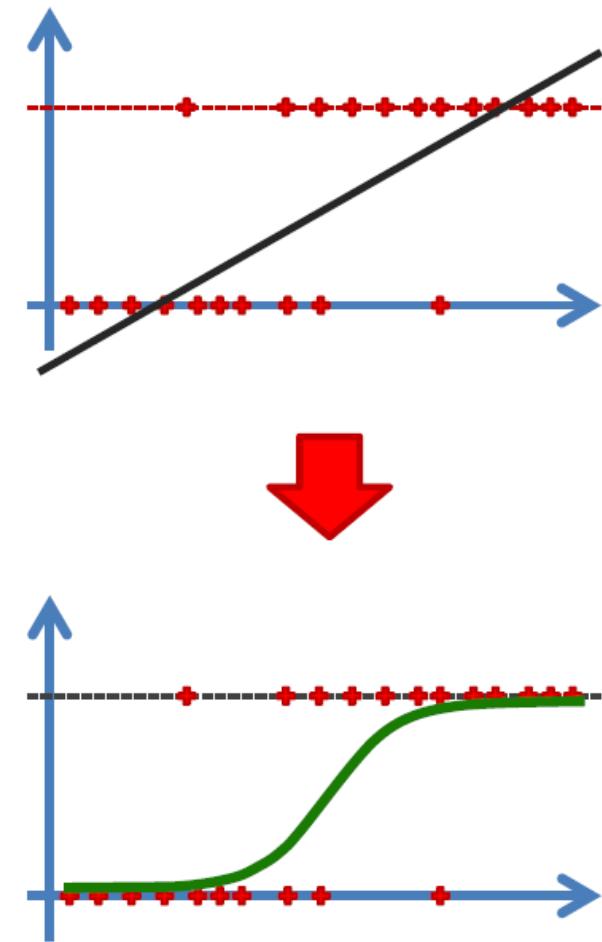
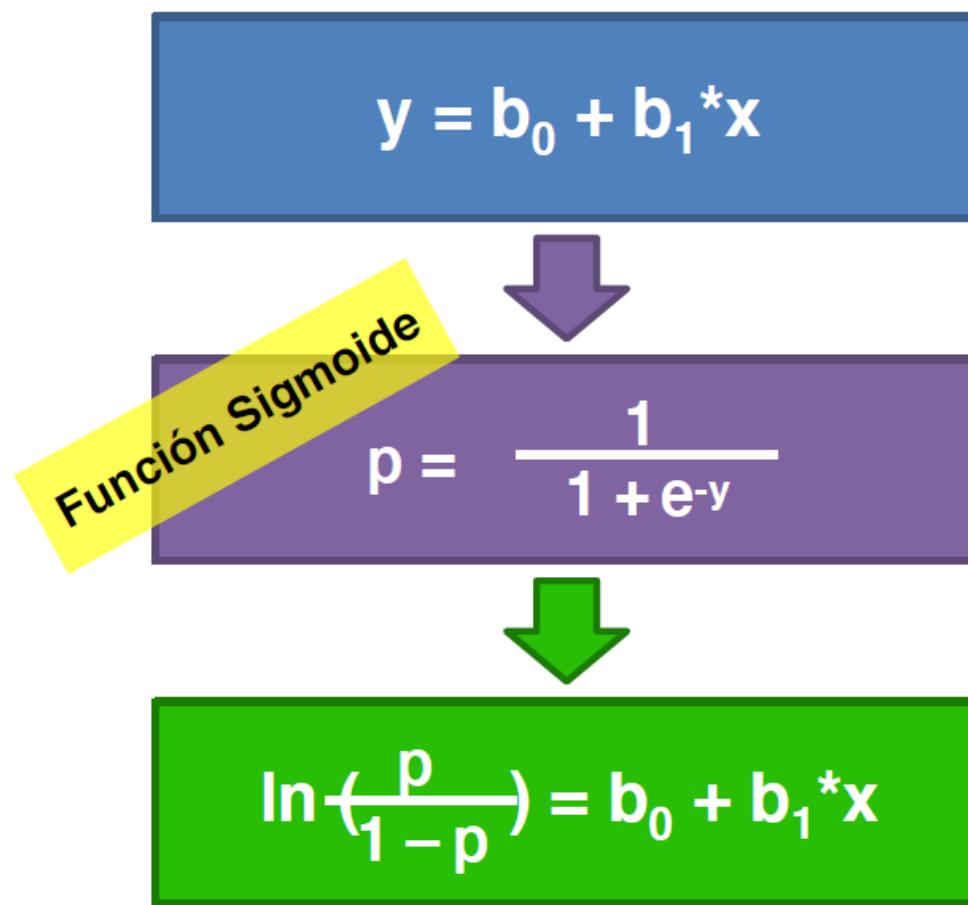
Lo nuevo es:

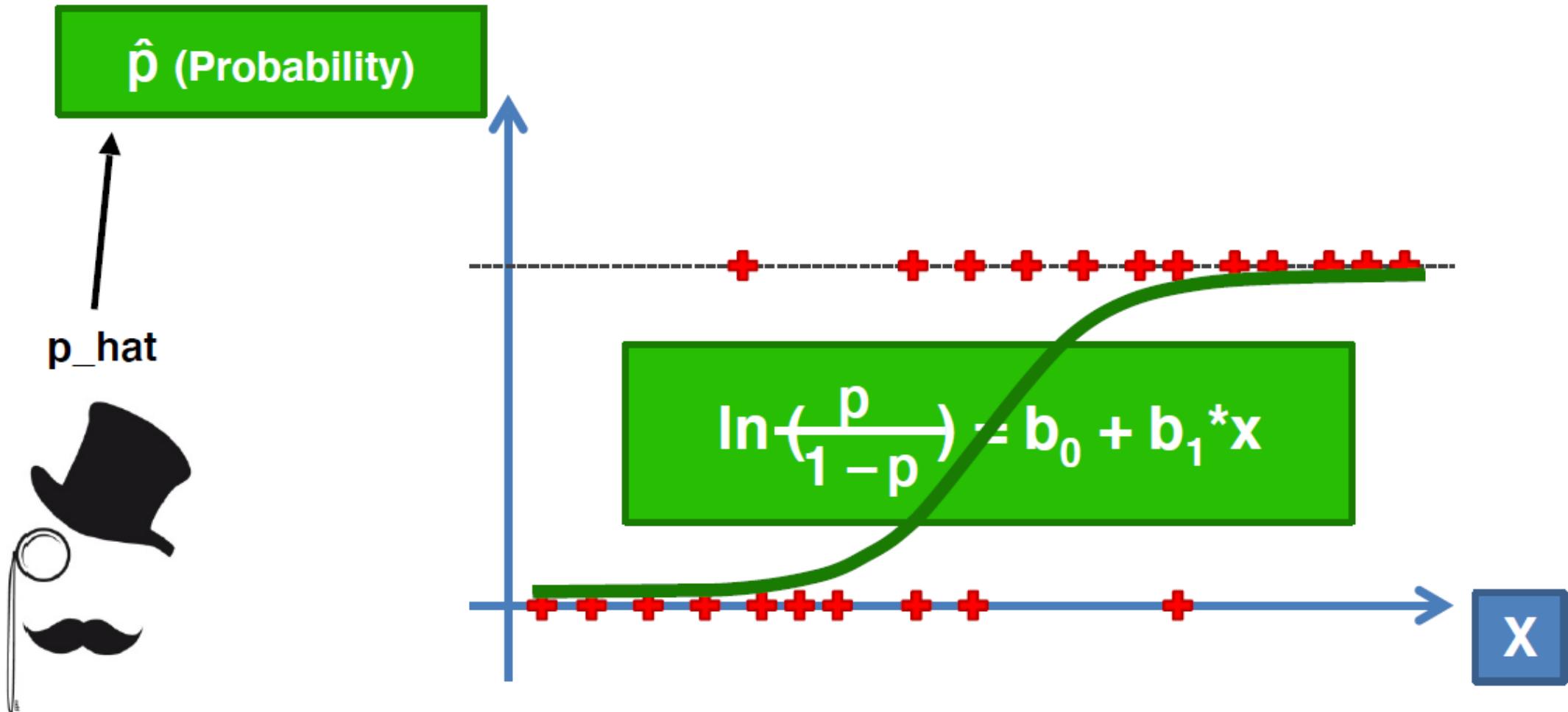


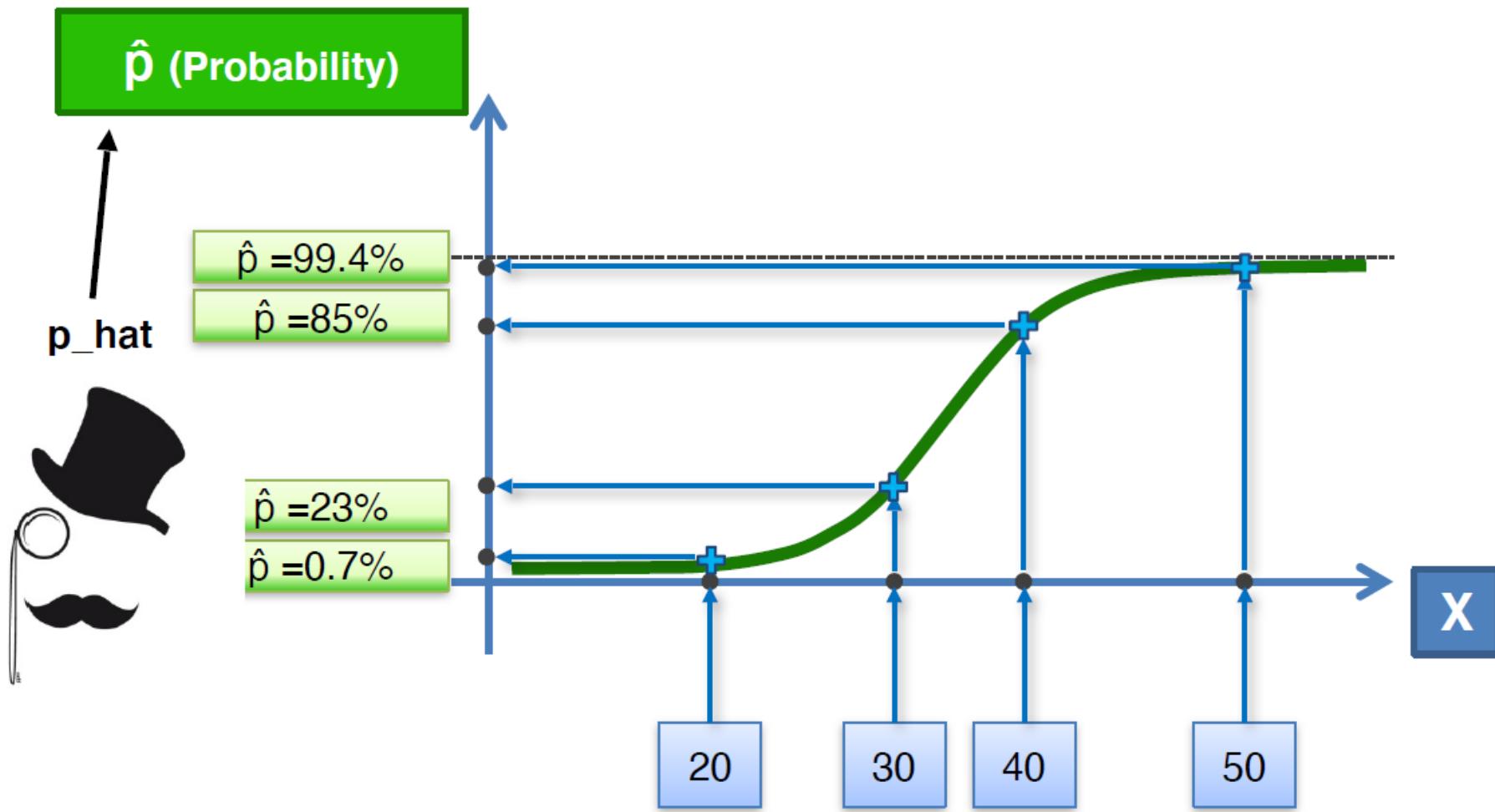
Sabemos que:











# ¿DÓNDE Y CUÁNDO APLICARLA?

Cuando queremos investigar si una o varias variables explican una variable dependiente que toma un carácter cualitativo.

Este hecho es muy frecuente en **medicina** ya que constantemente intentamos dar respuesta a preguntas formuladas en base a la presencia o ausencia de una determinada **característica que no es cuantificable** sino que representa la existencia o no de un **efecto de interés**, como por ejemplo el desarrollo de un **«evento cardiovascular»**, **«un paciente hospitalizado muere o no antes del alta»**, **«se produce o no un reingreso»**, **«un paciente desarrolla o no nefropatía diabética»**.



## Ejemplos de Uso

Vamos a clasificar problemas con dos posibles estados «SI/NO»: binario o un número finito de «etiquetas» o «clases»: múltiple.

Algunos Ejemplos de Regresión

Logística son:

- Clasificar si el correo que llega es Spam o No es Spam
- Dados unos resultados clínicos de un tumor clasificar en «Benigno» o «Maligno».
- El texto de un artículo a analizar es: Entretenimiento, Deportes, Política ó Ciencia
- A partir de historial bancario conceder un crédito o no

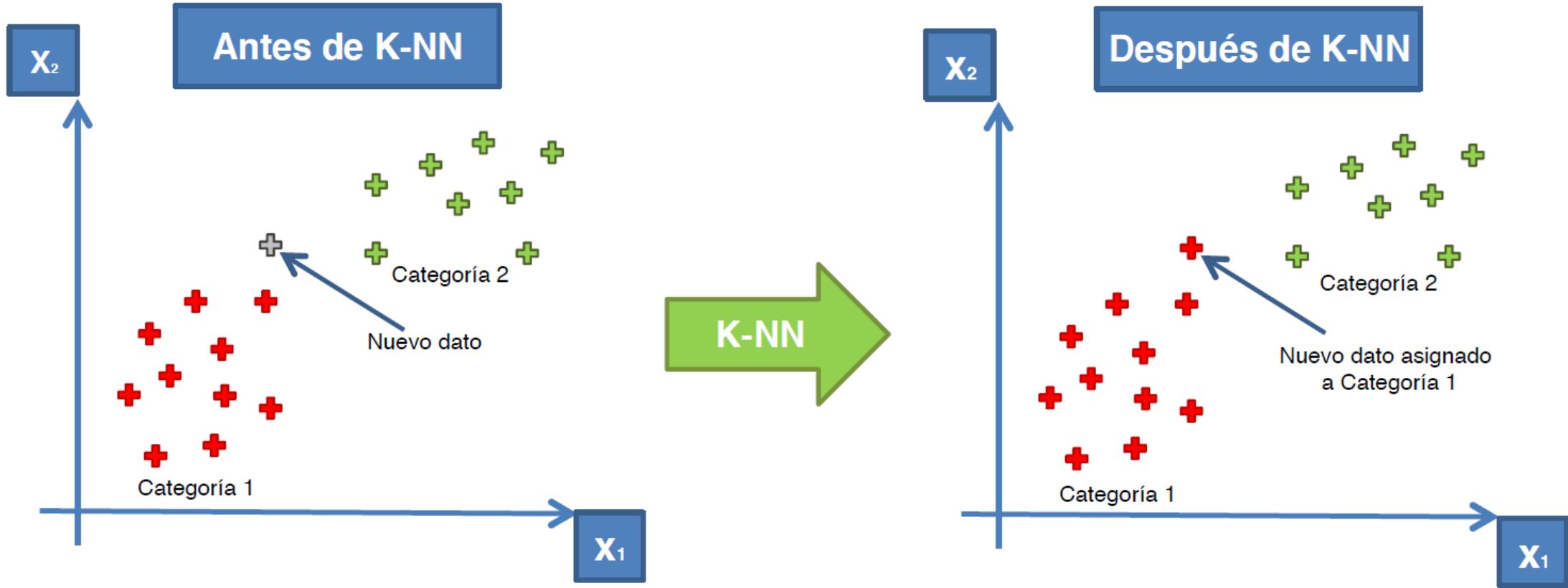
Es un método que simplemente busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de datos que le rodean.

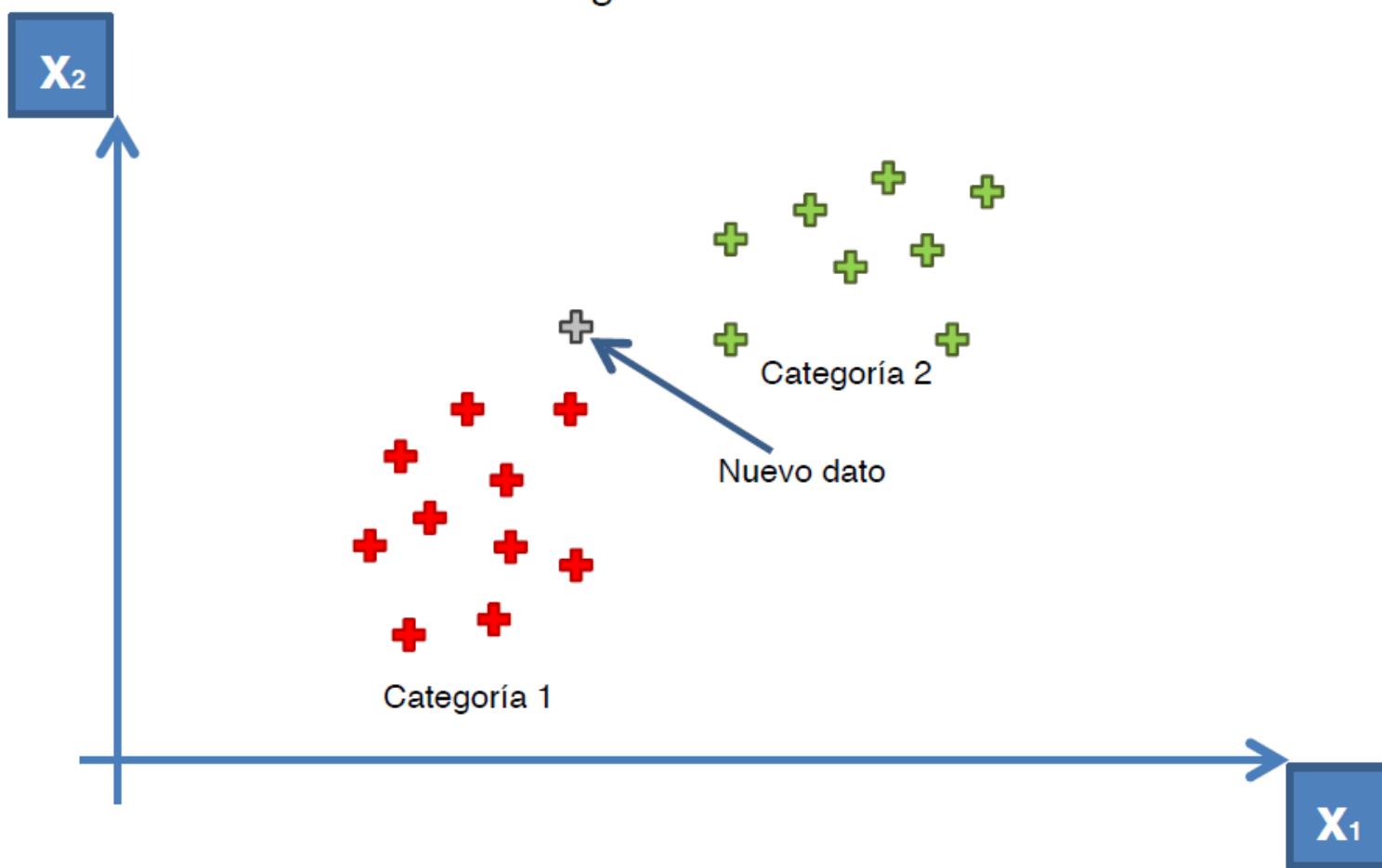
## K-Nearest-Neighbor (KNN)

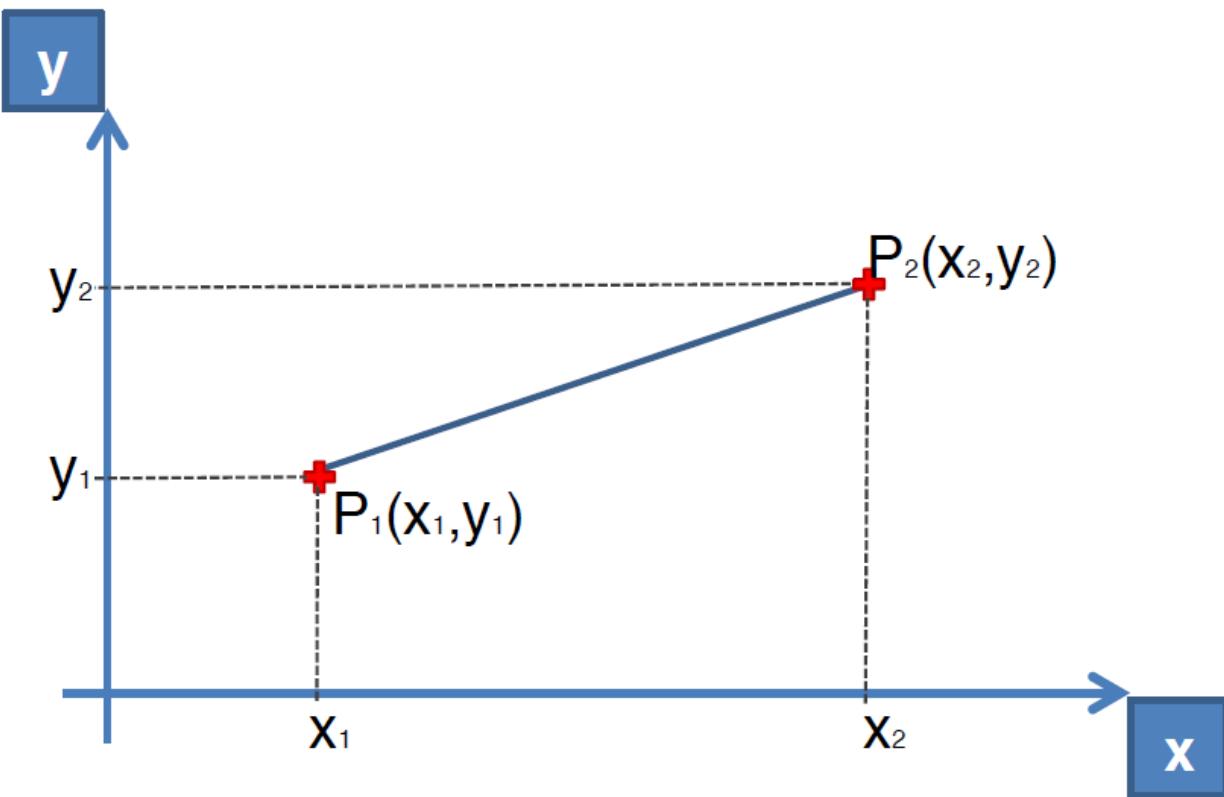
Supervisado - Basado en

Instancia



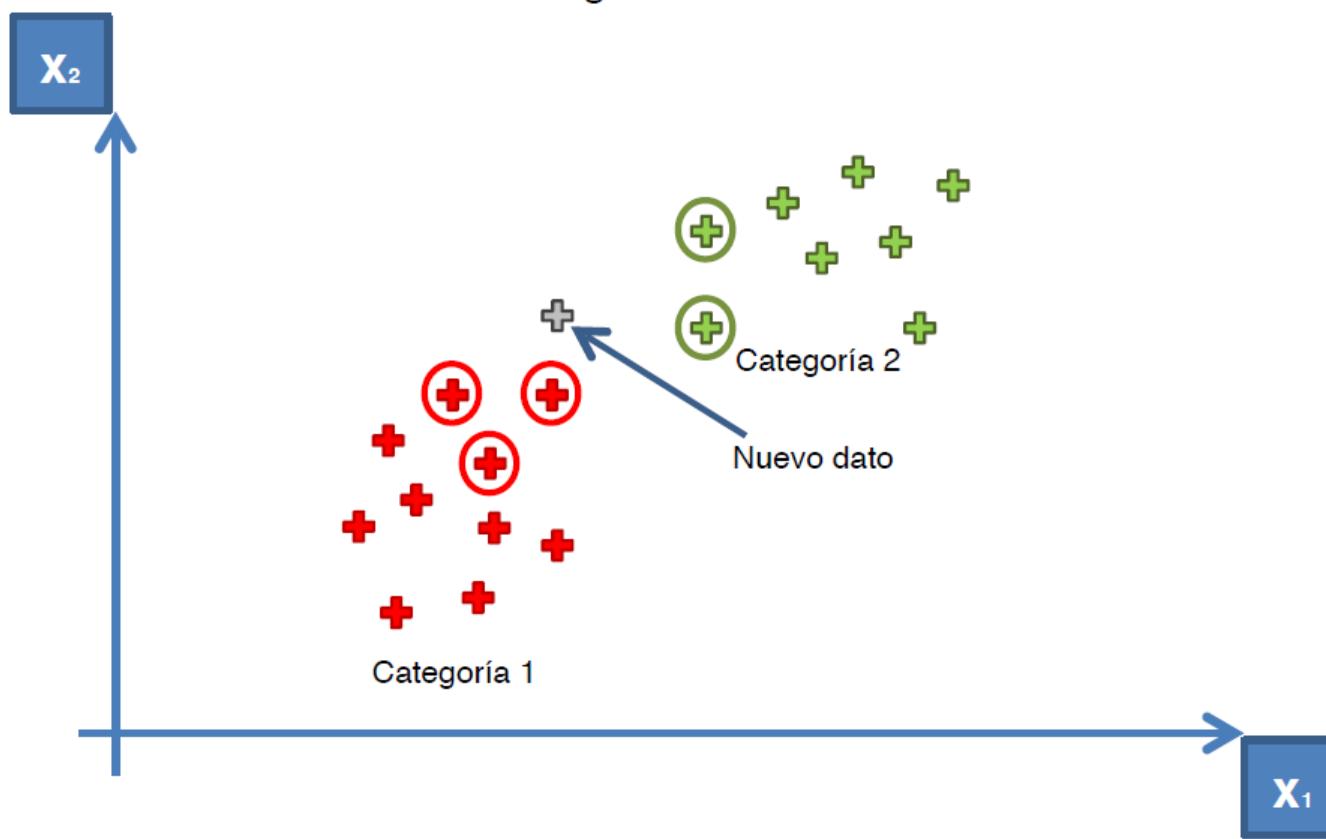


**PASO 1:** Elegir el número K de vecinos:  $K = 5$ 

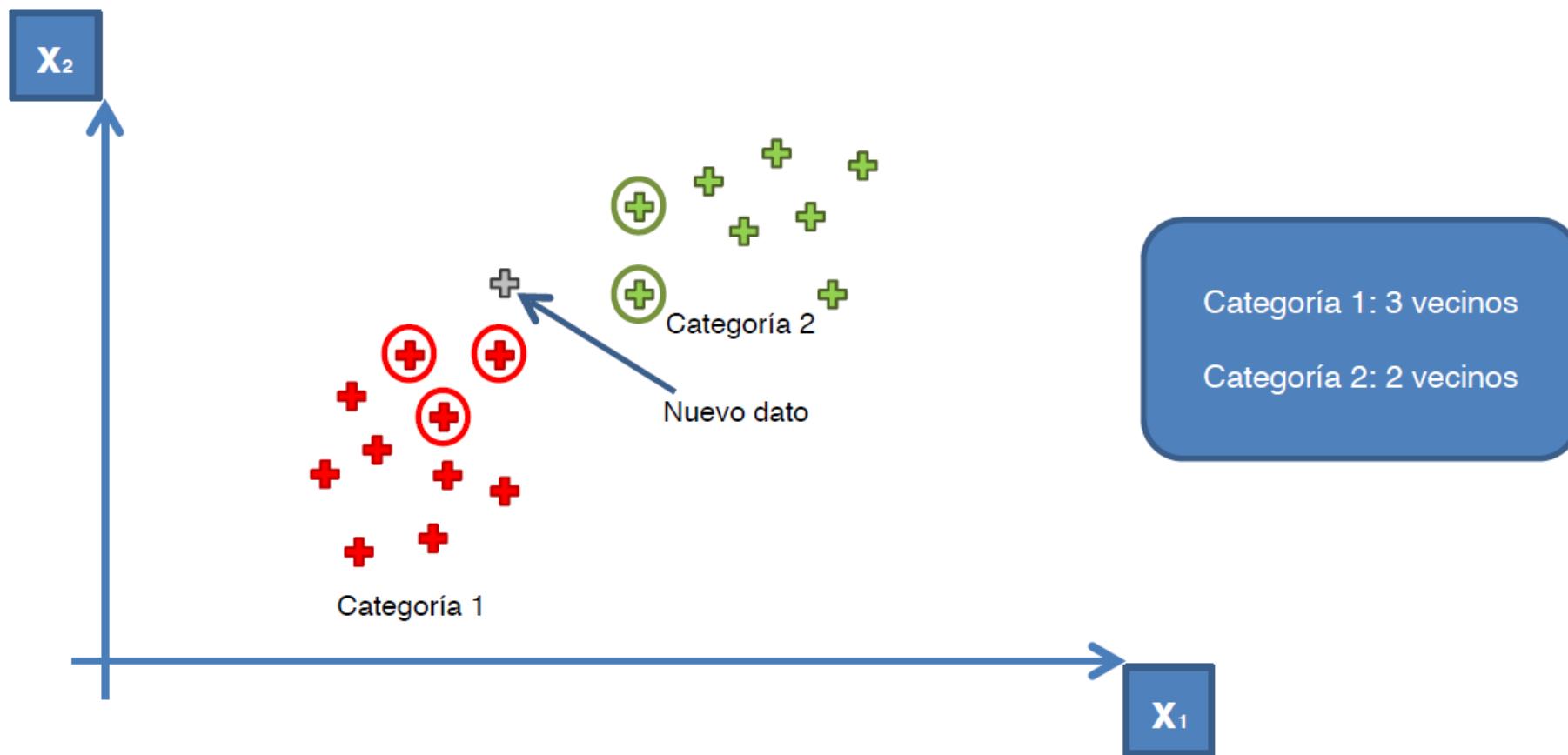


$$\text{Distancia Euclíadiana entre } P_1 \text{ y } P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

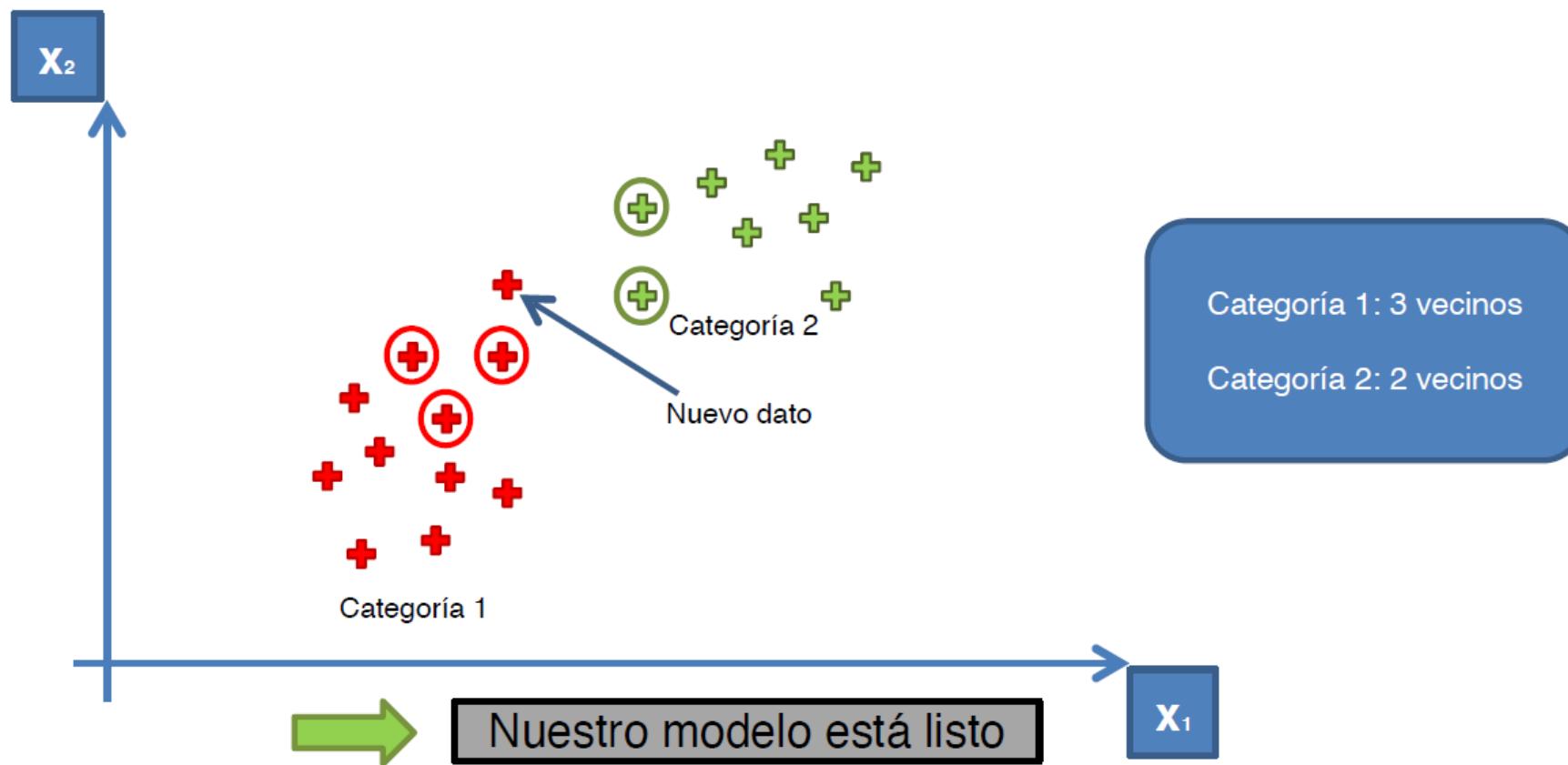
**PASO 2:** Tomar los  $K = 5$  vecinos más cercanos del nuevo dato,  
según la distancia Euclidiana



**PASO 3:** De entre esos K vecinos, contar el número de puntos de cada categoría



#### PASO 4: Asignar el nuevo dato a la categoría con más vecinos



# ¿DÓNDE Y CUÁNDO APLICARLA?

Aunque sencillo, se utiliza en la resolución de multitud de problemas, como en sistemas de recomendación, búsqueda semántica y detección de anomalías.

Como pros tiene sobre todo que es sencillo de aprender e implementar. Tiene como contras que utiliza todo el dataset para entrenar «cada punto» y por eso requiere de uso de mucha memoria y recursos de procesamiento (CPU). Por estas razones kNN tiende a funcionar mejor en datasets pequeños y sin una cantidad enorme de features (las columnas).



## Ejemplos de Uso

Data mining (minería de datos), por lo que su uso en aprendizaje automático (Machine Learning) es una de sus aplicaciones clave para, por ejemplo, sistemas de recomendación (plataformas de contenido digital, procesos de venta cruzada en eCommerce, etc.).

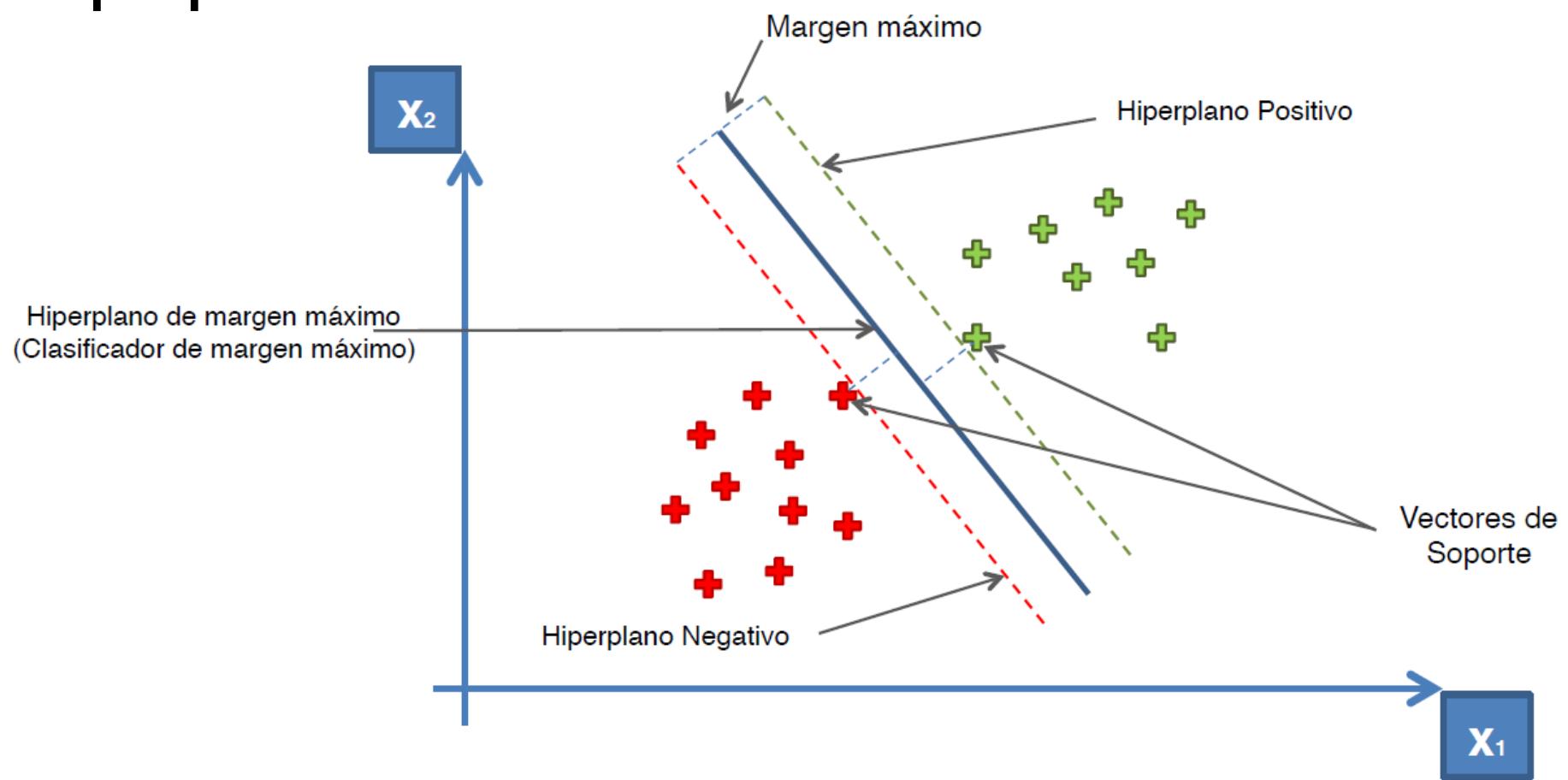
Por último y, a modo de comentario, el concepto de los vecinos más cercanos es un importante factor en la física de estado sólido, ya que las propiedades de la materia se modifican en función de la estructura de las redes y las distancias entre los átomos. De hecho, es habitual hacer aproximaciones en los modelos físicos para despreciar aquellas partículas que no forman parte de los K vecinos más próximos.



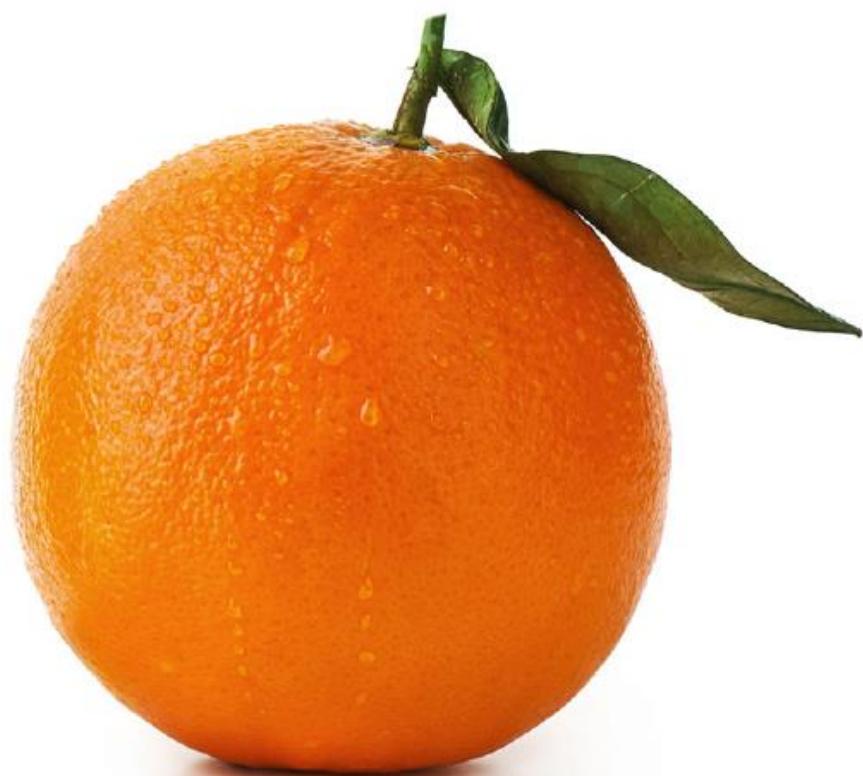
# Support Vector Machina SVM

Un modelo de clasificación de Máquinas de Soporte Vectorial (Support Vector Machine, SVM) es un algoritmo de aprendizaje supervisado utilizado para resolver problemas de clasificación binaria y, mediante extensiones, también problemas de clasificación multiclas.

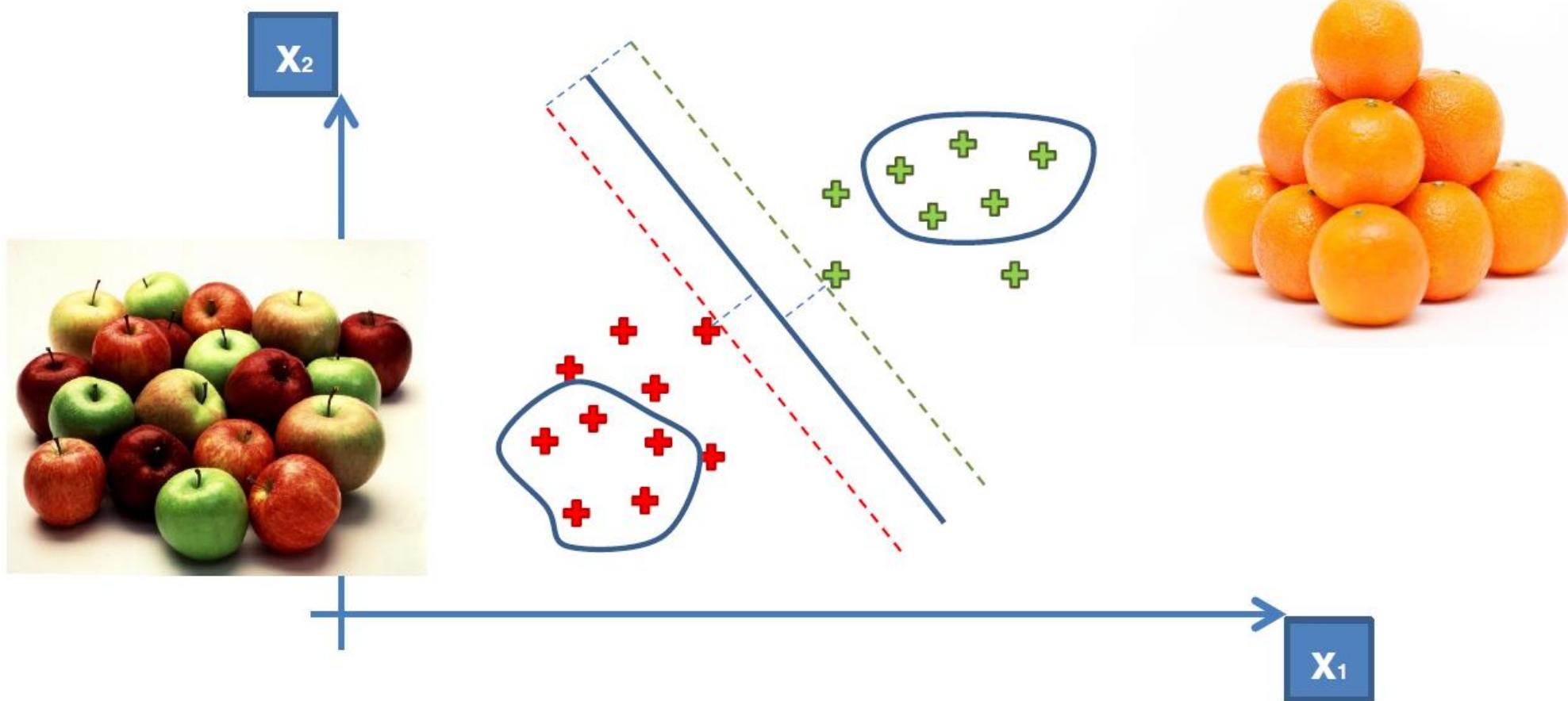
# Hiperplanos



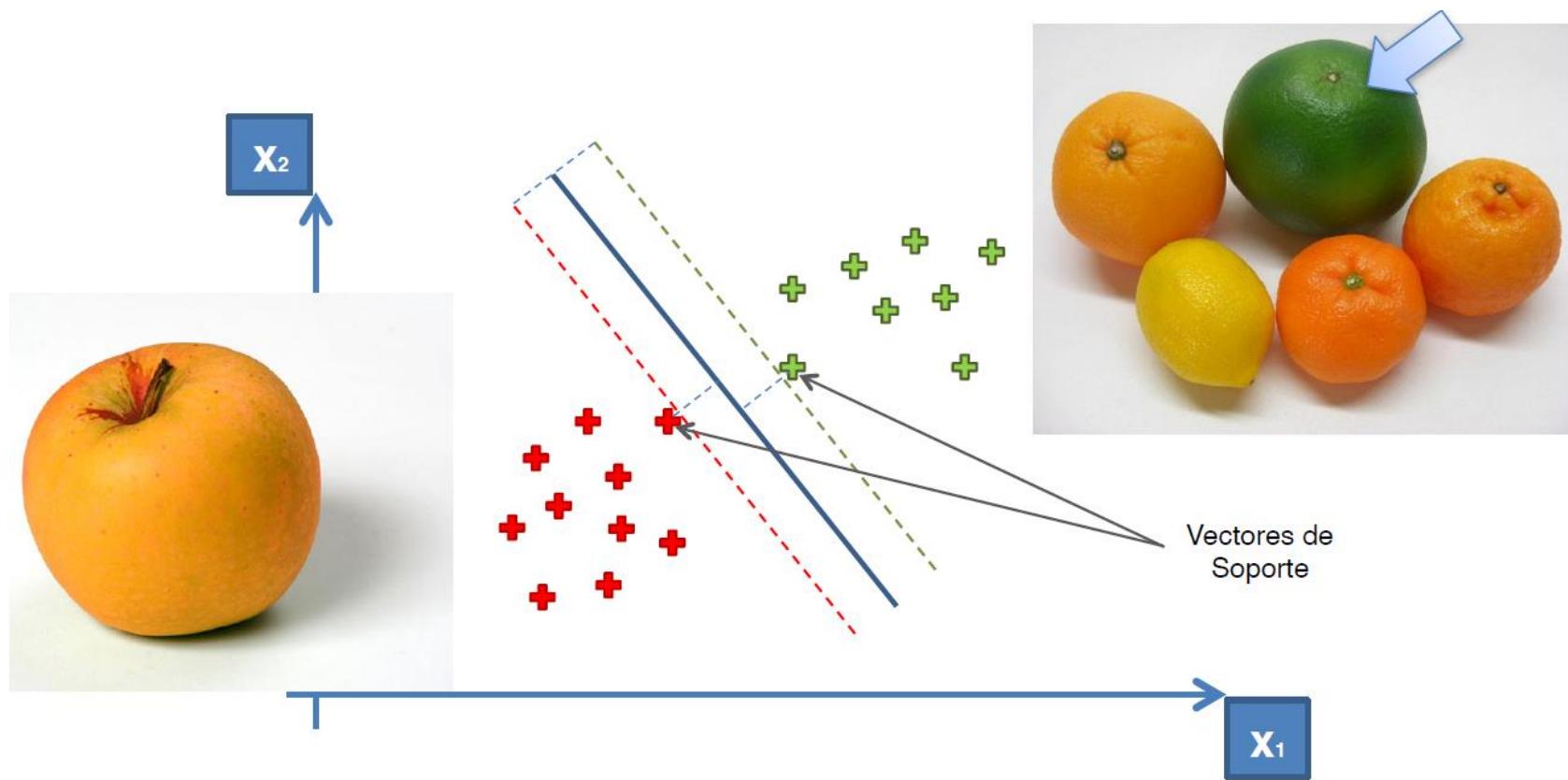
**¿Qué tienen de especial las SVM?**



## ¿Qué tienen de especial las SVM?

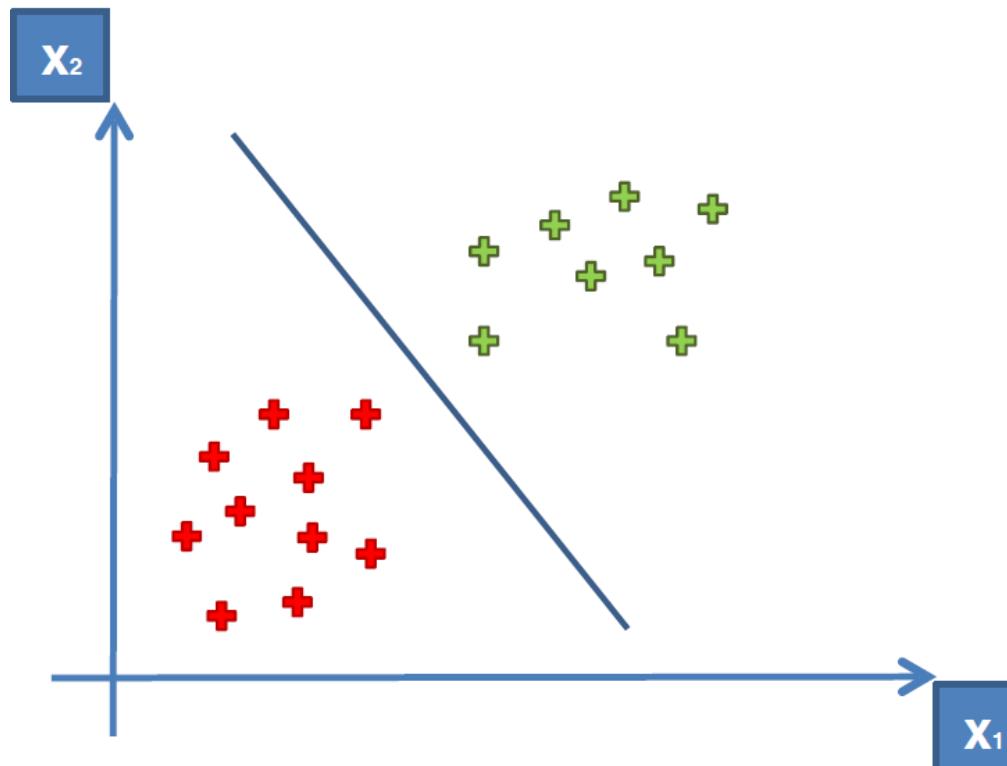


## ¿Qué tienen de especial las SVM?

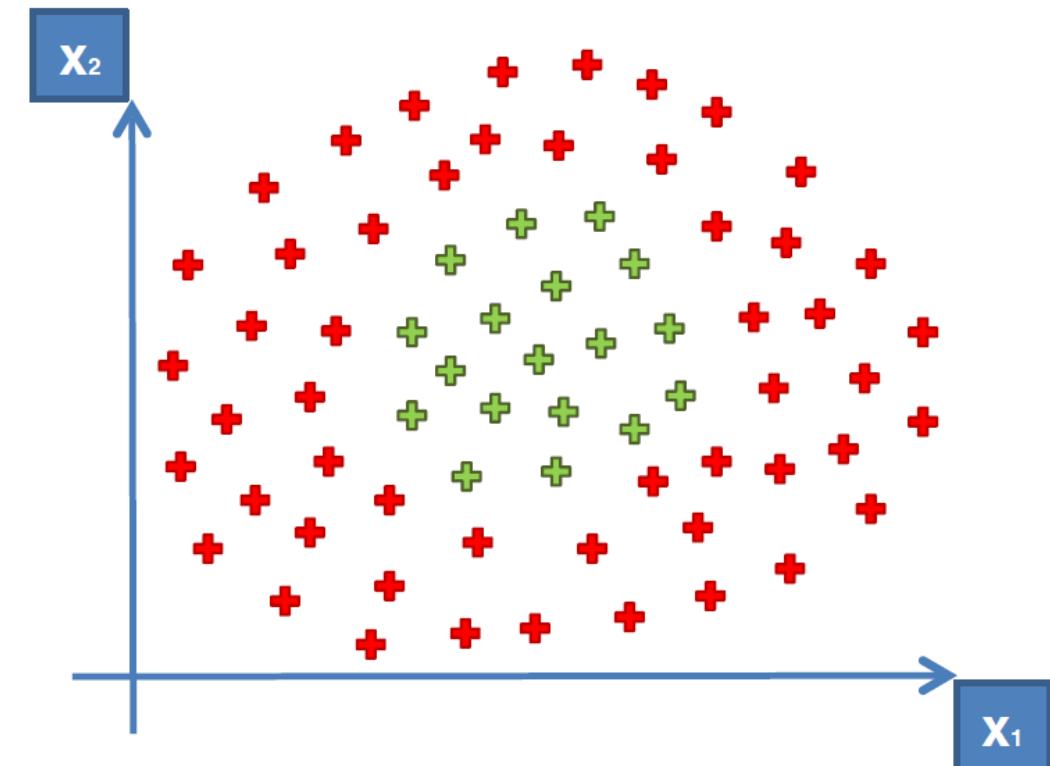


# Kernel SVM

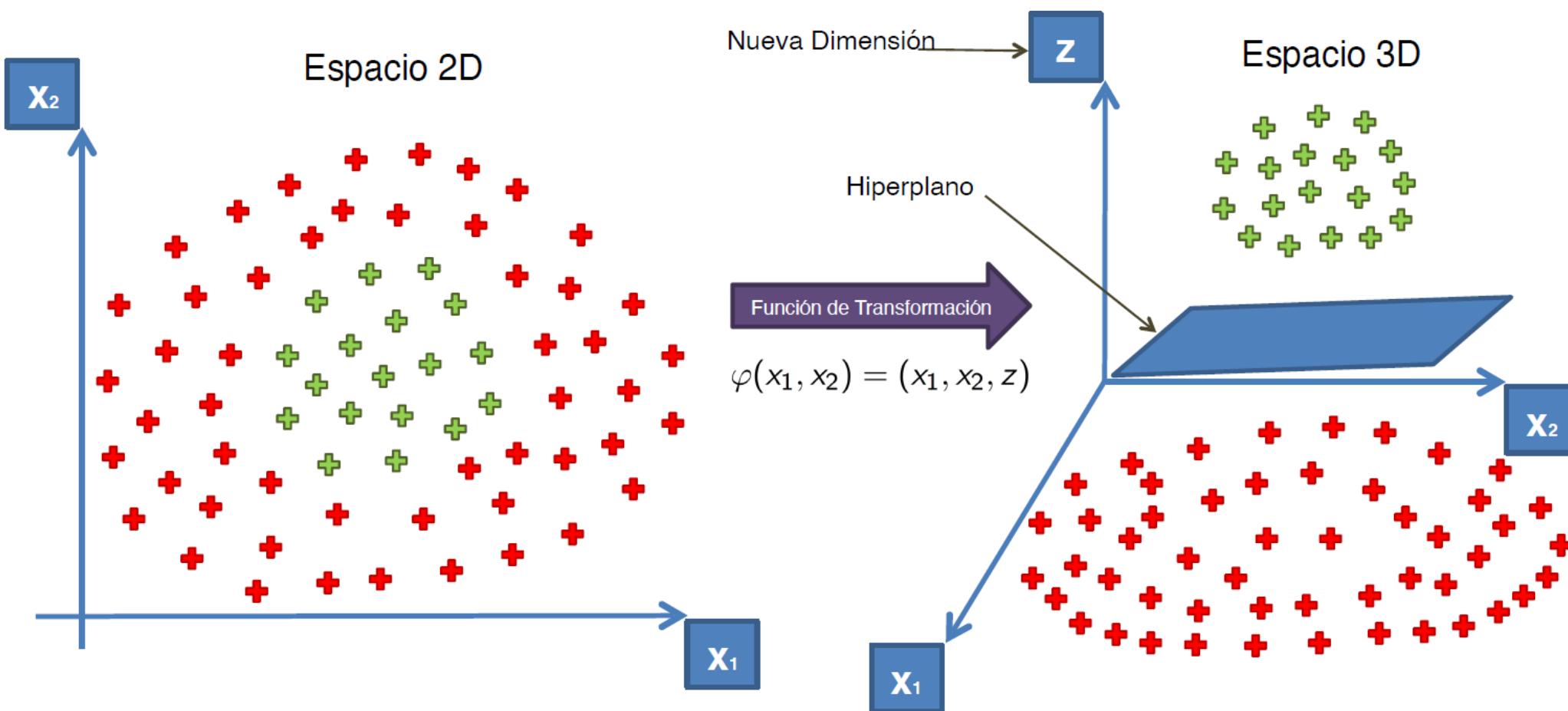
Linealmente Separable



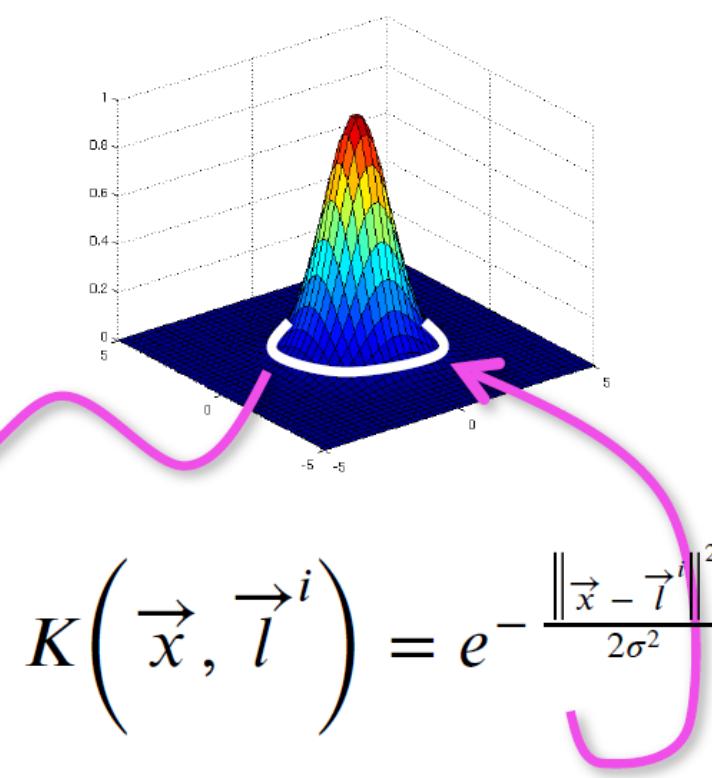
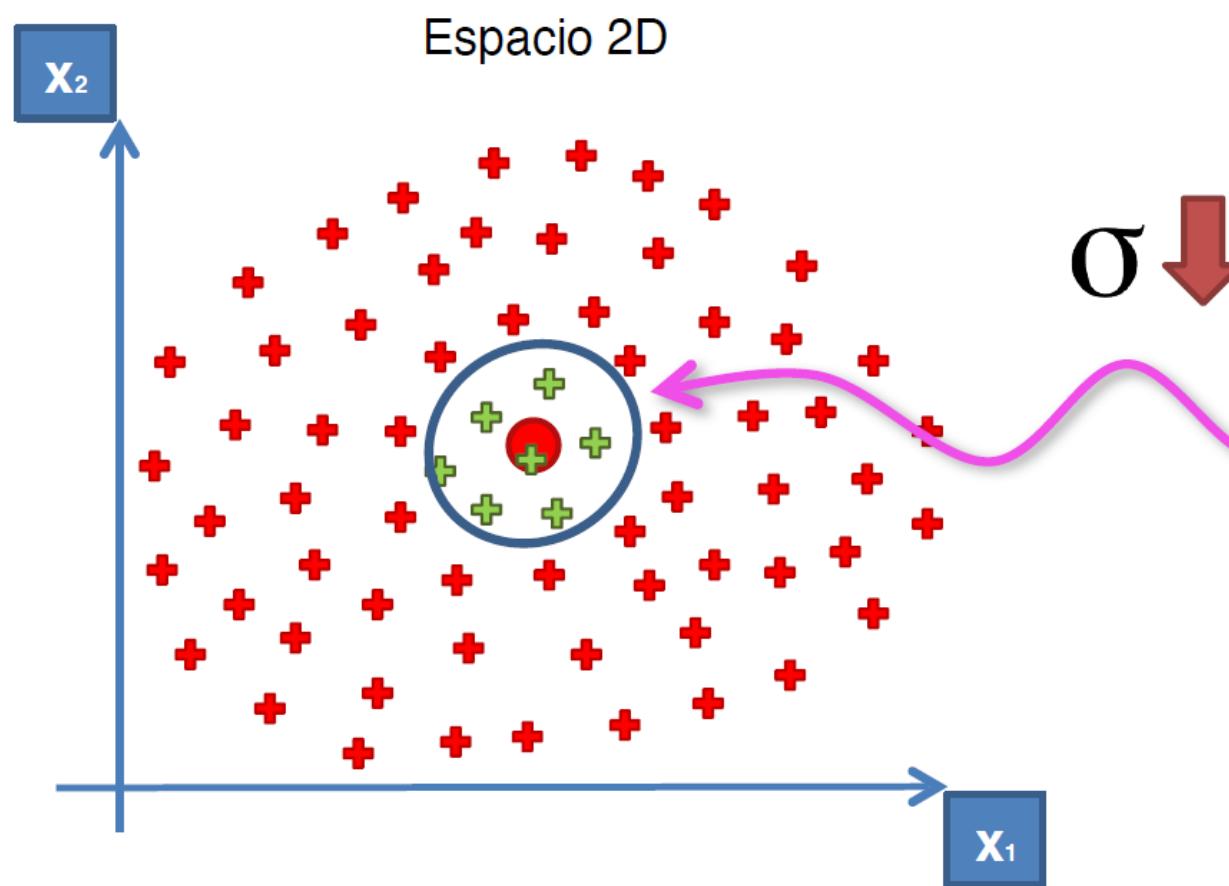
No Linealmente Separable



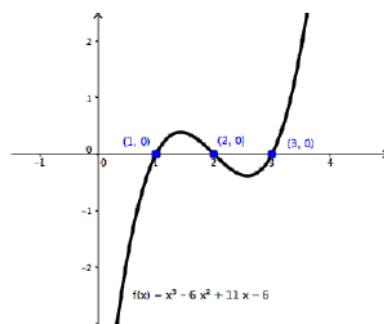
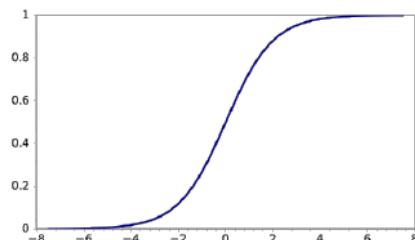
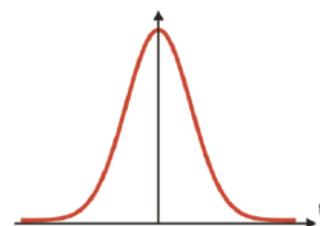
# Kernel SVM



# Kernel SVM



# Kernel SVM



**Kernel Gaussiano RBF**

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x} - \vec{l}^i\|^2}{2\sigma^2}}$$

**Kernel Sigmoide**

$$K(X, Y) = \tanh(\gamma \cdot X^T Y + r)$$

**Kernel Polinómico**

$$K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

# ¿DÓNDE Y CUÁNDO APLICARLA?

Imaginemos que buscamos encontrar qué tipo de usuario tiene más probabilidad de hacer clic en un determinado banner. Está claro que esta decisión implica varias variables a tener en cuenta: no solo de las características del propio usuario, sino también podremos considerar su región geográfica, la tecnología empleada, día/hora en que se encuentra con el banner, etc.

- **¿Qué queremos medir?** En función de ello, sabremos si debemos resolver un problema de clasificación o regresión.

- **¿Qué variables tenemos que considerar?** Según esto, la dimensionalidad del problema varía. Por ejemplo, podríamos tener en cuenta: día, hora, fecha, web/app, localización del usuario (ciudad, país, etc.), tipo de dispositivo, navegador, etc., a más variables, mayor complejidad del algoritmo.

- **¿El problema es lineal? ¿necesito kernel?** Dependiendo de las variables consideradas, estaremos en uno u otro caso que debemos analizar. La respuesta a estas preguntas necesitaría un proceso de prueba y error hasta

## Ejemplos de Uso

Clasificación de células cervicales con Máquinas de Soporte Vectorial empleando rasgos del núcleo.

### Algunas Ventajas

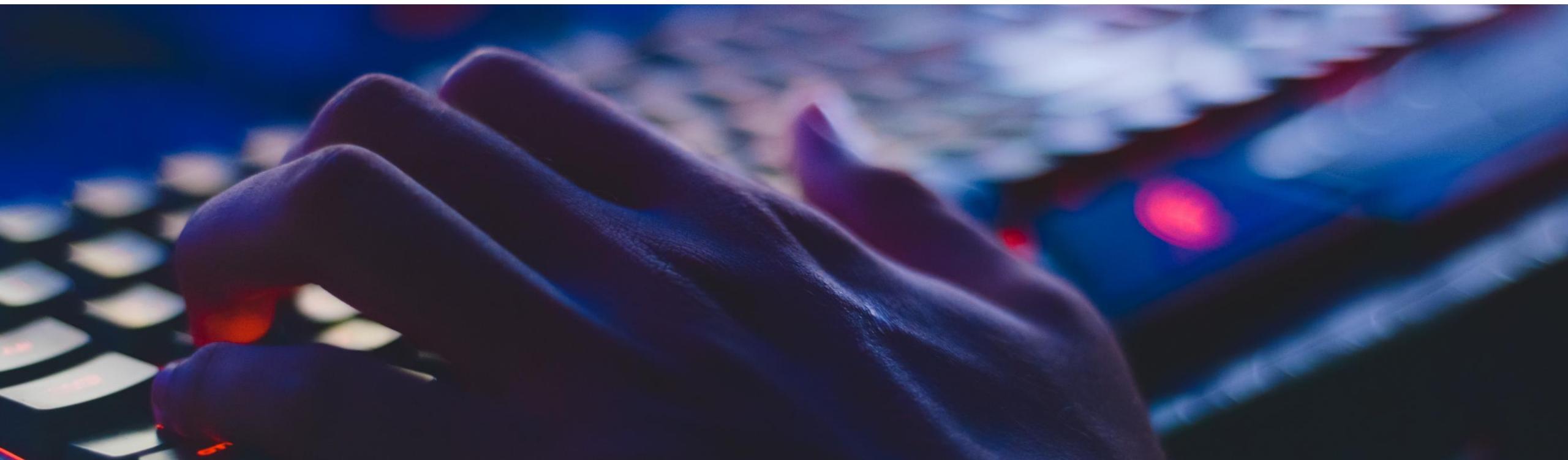
- El entrenamiento es relativamente fácil. No hay óptimo local, como en las redes neuronales.
- Se escalan relativamente bien para datos en espacios dimensionales altos.
- El compromiso entre la complejidad del clasificador y el error puede ser controlado explícitamente.
- Datos no tradicionales como cadenas de caracteres y árboles pueden ser usados como entrada a la SVM, en vez de vectores de características.

### Debilidades

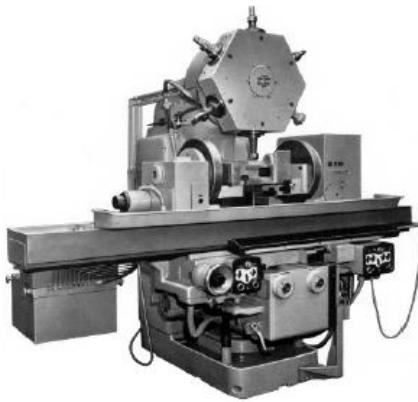
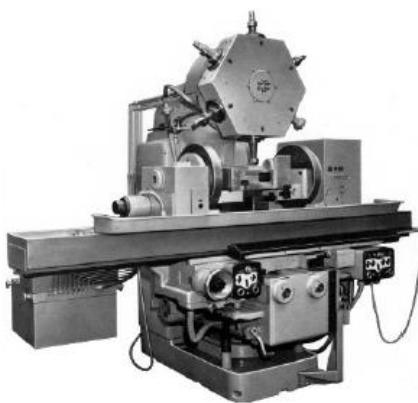
- Se necesita una “buena” función kernel, es decir, se necesitan metodologías eficientes para

Naïve Bayes o el Ingenuo Bayes es uno de los algoritmos más simples y poderosos para la clasificación basado en el Teorema de Bayes con una suposición de independencia entre los predictores. Naive Bayes es fácil de construir y particularmente útil para conjuntos de datos muy grandes.

## Naïve Bayes



# Teorema



$$P(A | B) = \frac{P(B | A) * P(A)}{P(B)}$$

## Teorema

#4

Probabilidad a Posteriori

#3

Probabilidad Condicionada

#1

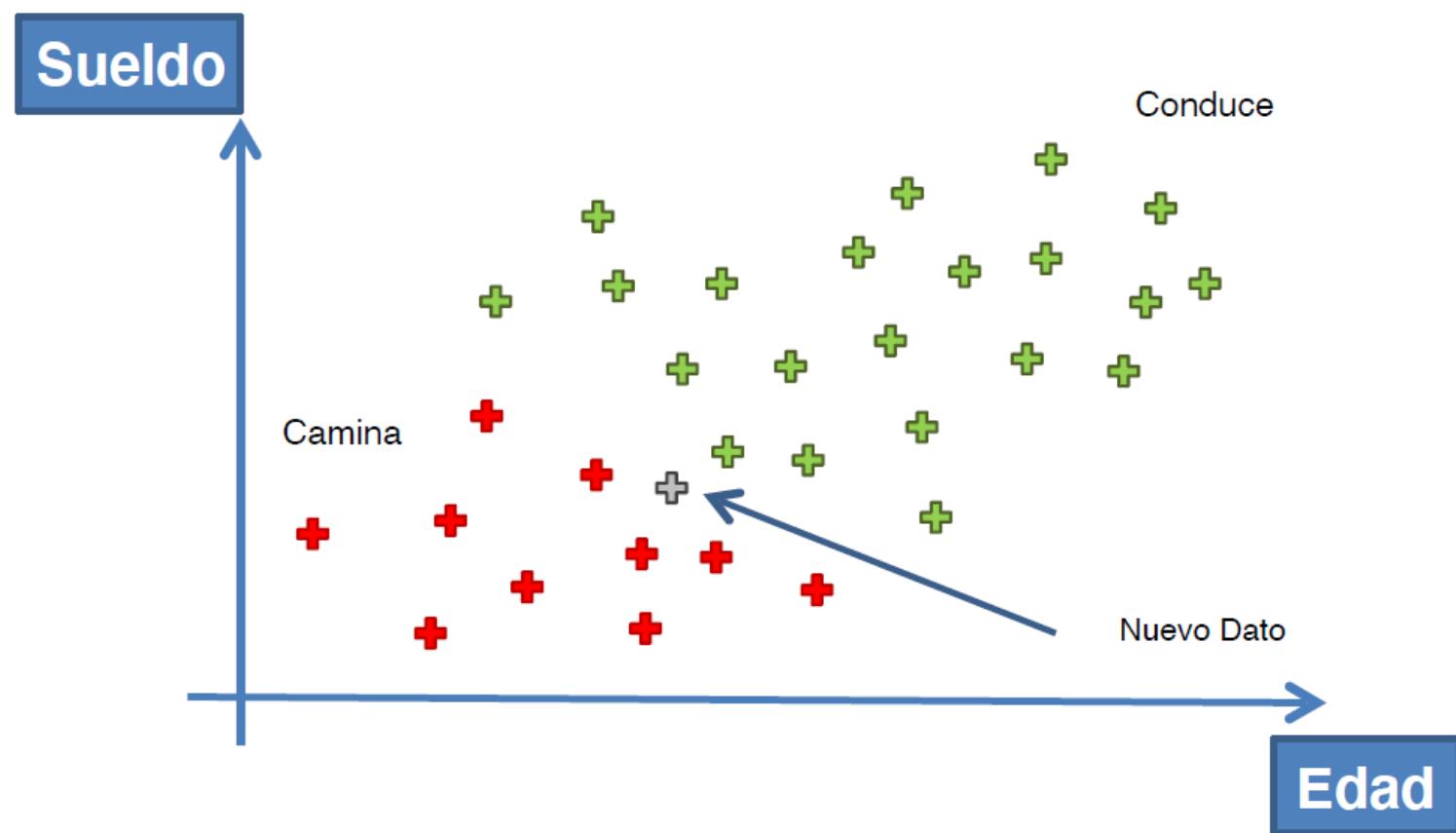
Probabilidad a Priori

$$P(Drives \mid X) = \frac{P(X \mid Drives) * P(Drives)}{P(X)}$$

#2

Probabilidad Marginal

# Teorema



# ¿DÓNDE Y CUÁNDO APLICARLA?

Donde:

- $P(h)$ : es la probabilidad de que la hipótesis  $h$  sea cierta (independientemente de los datos). Esto se conoce como la probabilidad previa de  $h$ .
- $P(D)$ : probabilidad de los datos (independientemente de la hipótesis). Esto se conoce como probabilidad previa.
- $P(h|D)$ : es la probabilidad de la hipótesis  $h$  dada los datos  $D$ . Esto se conoce como la probabilidad posterior.
- $P(D|h)$ : es la probabilidad de los datos  $d$  dado que la hipótesis  $h$  era cierta. Esto se conoce como probabilidad posterior.

En caso de que se tenga una sola característica, el clasificador Naive Bayes calcula la probabilidad de un evento en los siguientes pasos:

**Paso 1:** calcular la probabilidad previa para las etiquetas de clase dadas.

**Paso 2:** determinar la probabilidad de probabilidad con cada atributo para cada clase.

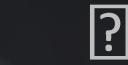
**Paso 3:** poner estos valores en el teorema de Bayes y calcular la probabilidad posterior.

## Ejemplos de Uso

**Algunas Ventajas:** Es rápido, simple de implementar, funciona bien con conjunto de datos pequeños, va bien con muchas dimensiones (features) y llega a dar buenos resultados aún siendo «ingenuo» sin que se cumplan todas las condiciones de distribución necesarias en los datos.

**Contras:** Requiere quitar las dimensiones con correlación y para buenos resultados las entradas deberían cumplir las 2 suposiciones de distribución normal e independencia entre sí (muy difícil que sea así ó deberíamos hacer transformaciones en los datos de entrada).

**Ejemplos de Clasificador de documentos, Toma de Decisiones.**

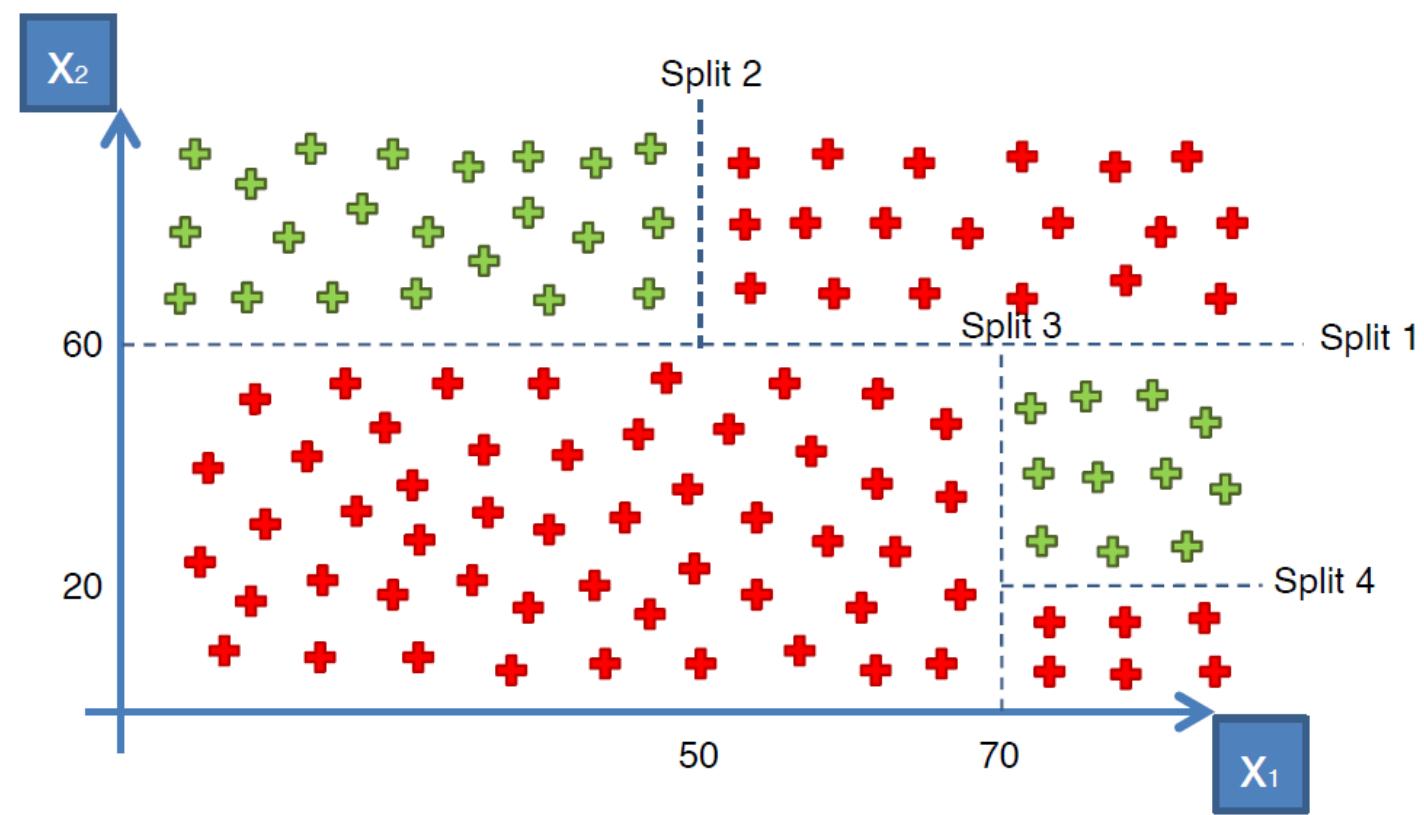


# Árboles de Decisión

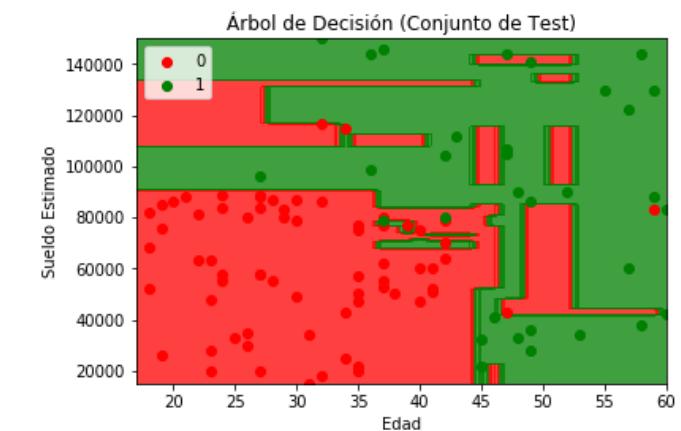
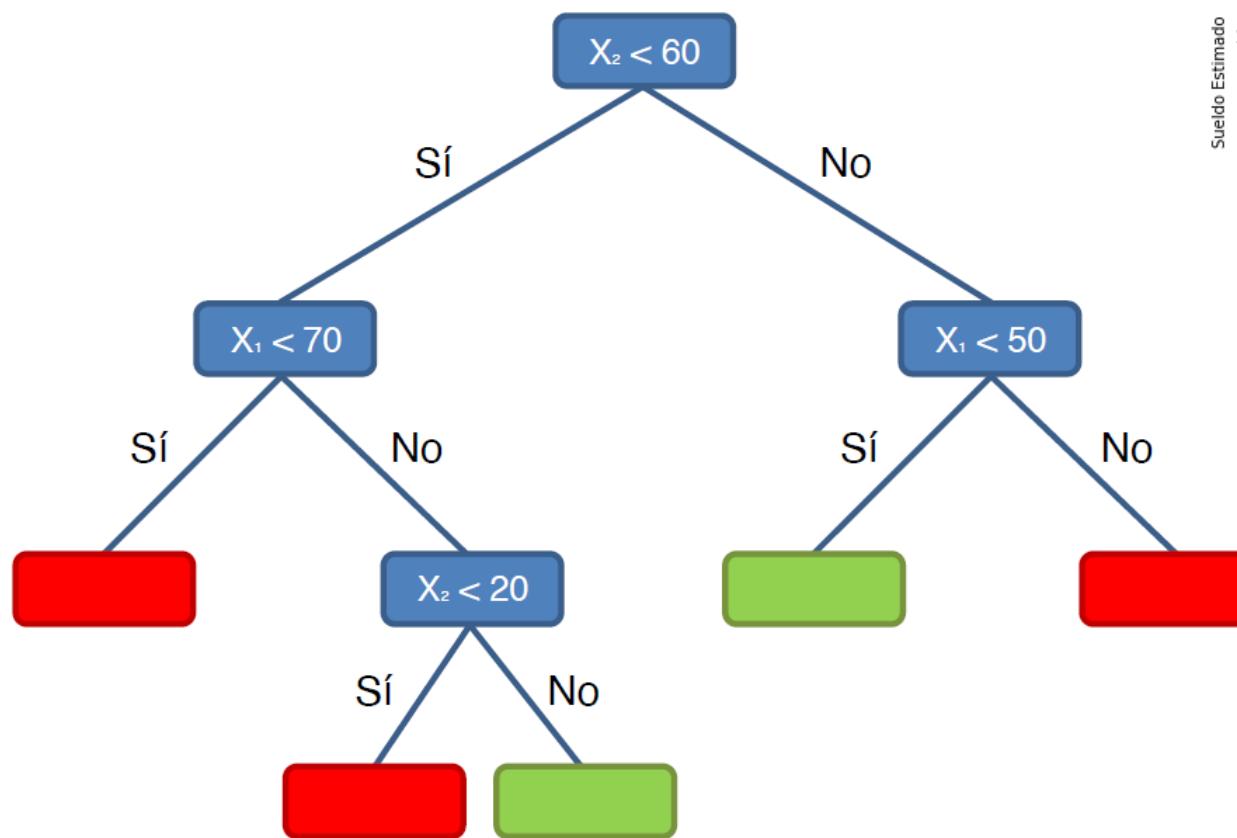
Arboles en cantidad

Es la combinación de **árboles predictores** tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

# Árboles de Decisión



# Árboles de Decisión



# ¿DÓNDE Y CUÁNDO APLICARLA?

1. Regresión con una variable dependiente continua.
2. Regresión binaria.
3. Problemas de clasificación con categorías múltiples ordinales.
4. Problemas de clasificación con categorías múltiples nominales.



## Ejemplos de Uso

**Algunas Ventajas:** Es rápido, simple de implementar, funciona bien con conjunto de datos pequeños, va bien con muchas dimensiones (features) y llega a dar buenos resultados aún siendo «ingenuo» sin que se cumplan todas las condiciones de distribución necesarias en los datos.

**Contras:** Requiere quitar las dimensiones con correlación y para buenos resultados las entradas deberían cumplir las 2 suposiciones de distribución normal e independencia entre sí (muy difícil que sea así ó deberíamos hacer transformaciones en los datos de entrada).

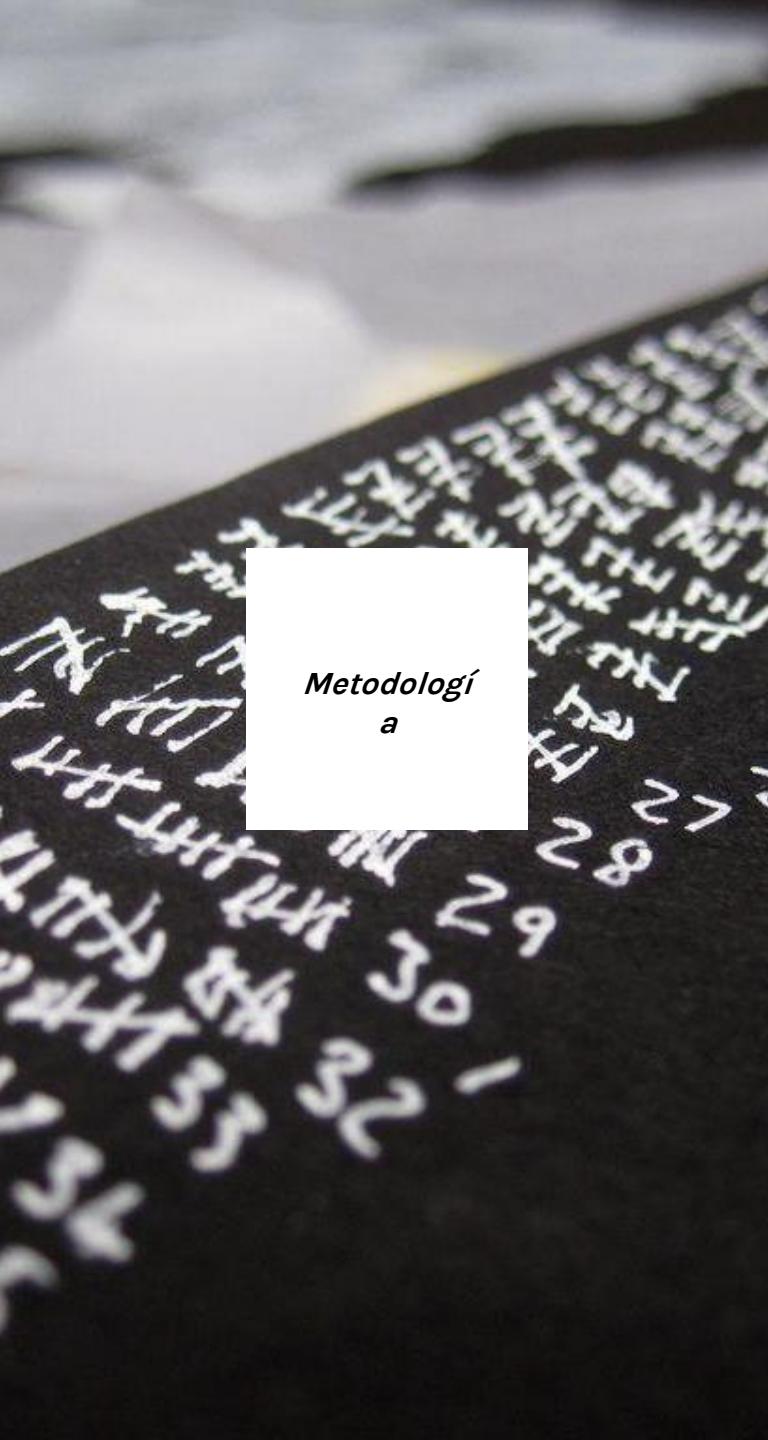
**Ejemplos de Clasificador de documentos, Toma de Decisiones.**



Naïve Bayes o el Ingenuo Bayes es uno de los algoritmos más simples y poderosos para la clasificación basado en el Teorema de Bayes con una suposición de independencia entre los predictores. Naive Bayes es fácil de construir y particularmente útil para conjuntos de datos muy grandes.

## Bosques Aleatorios





# Idea de los Bosques Aleatorios

Una receta de

Cocina

## Paso 1

Elegir un número aleatorio K de puntos de datos del Conjunto de Entrenamiento.

## Paso 2

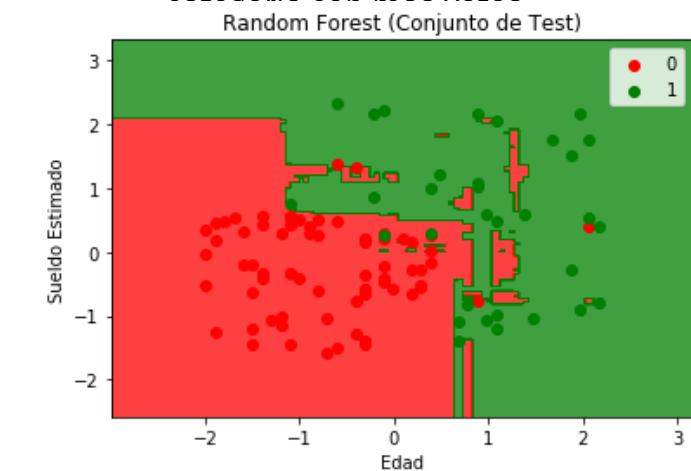
Construir el Árbol de Decisión asociado a esos K Puntos de Datos.

## Paso 3

Elegir el número Ntree de árboles que queremos construir y repetimos los PASOS 1 y 2.

## Paso 4

Para clasificar un nuevo punto, hacer que cada uno de los Ntree árboles elabore la predicción de a qué categoría pertenece y asignar el nuevo punto a la



# ¿DÓNDE Y CUÁNDO APLICARLA?

1. Regresión con una variable dependiente continua.
2. Regresión binaria.
3. Problemas de clasificación con categorías múltiples ordinales.
4. Problemas de clasificación con categorías múltiples nominales.



## Ventajas y Desventajas

### Algunas Ventajas:

- Existen muy pocas suposiciones y por lo tanto la preparación de los datos es mínima.
- Puede manejar hasta miles de variables de entrada e identificar las más significativas. Método de reducción de dimensionalidad.
- Una de las salidas del modelo es la importancia de variables.
- Incorpora métodos efectivos para estimar valores faltantes.
- Es posible usarlo como método no supervisado (clustering) y detección de outliers.

### Contras:

- Pérdida de interpretación
- Bueno para clasificación, no tanto para regresión. Las predicciones no son de naturaleza continua.
- En regresión, no puede predecir más allá del rango de valores del conjunto de entrenamiento.
- Poco control en lo que hace el modelo (modelo caja negra para modeladores)



# Clustering

Agrupaciones



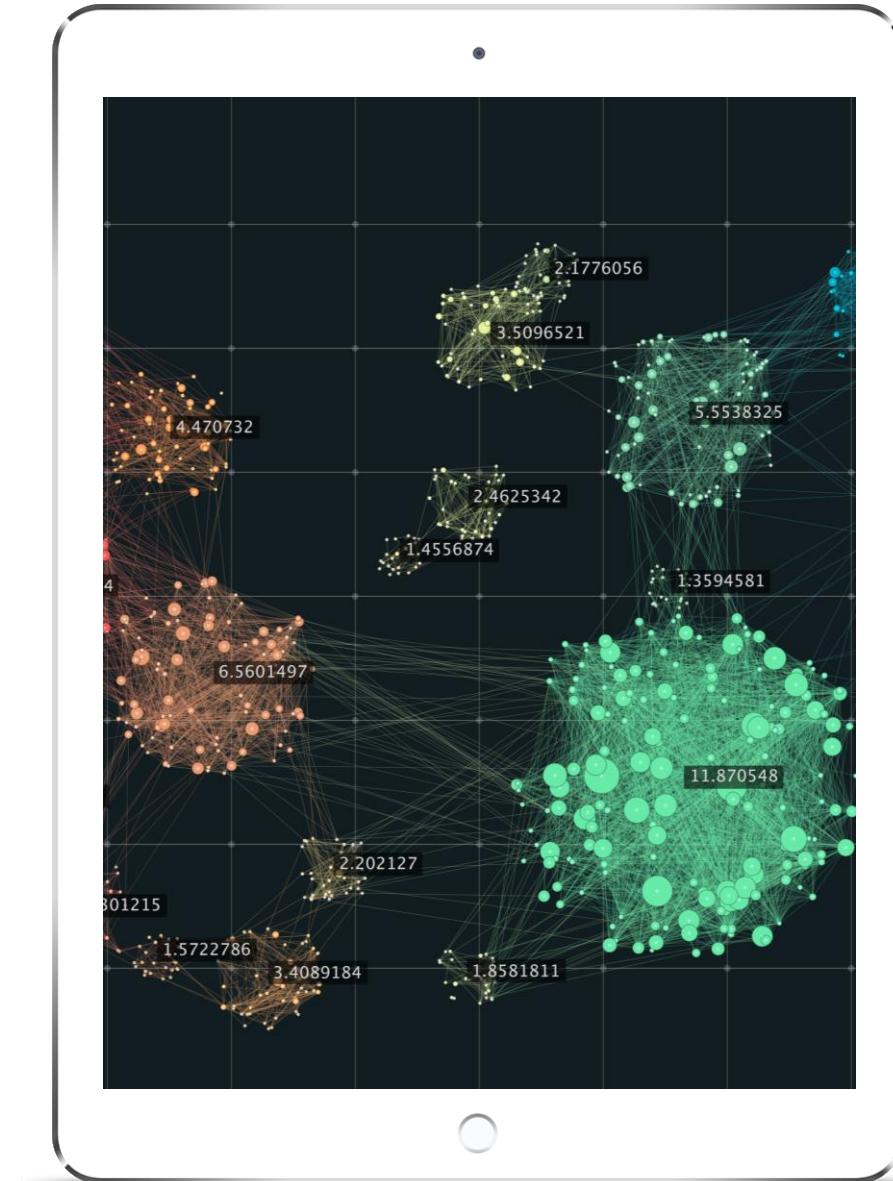
## Cluster

Un número de cosas o personas similares o cercanas, agrupadas



## Clustering

Es el proceso de particionar un conjunto de objetos (datos) en un conjunto de sub-clases con cierto significado



## Por tanto

Es un proceso de aprendizaje no supervisado:

Las clases no están predefinidas sino que deben ser descubiertas dentro de los ejemplos

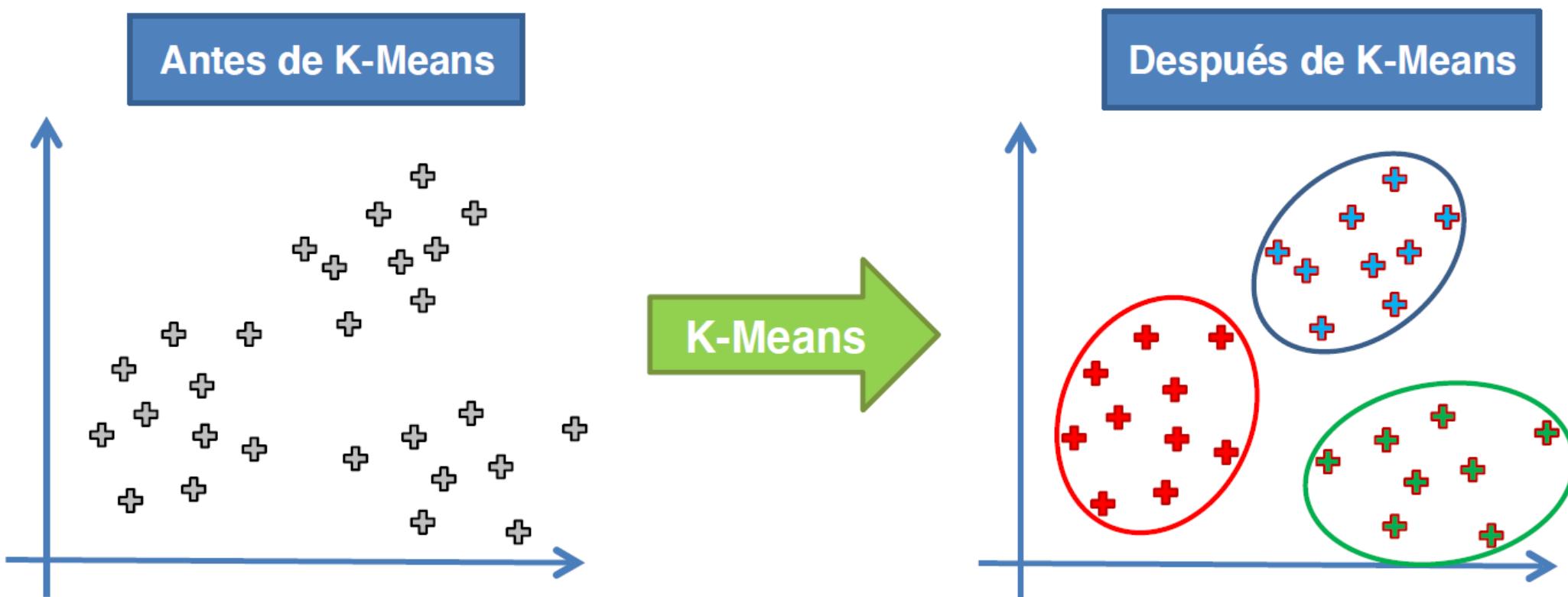
- Primariamente es un método descriptivo para interpretar un conjunto de datos
- Particionar ejemplos de clases desconocidas en subconjuntos disjuntos de clusters tal que:
  - Ejemplos en un mismo cluster sean altamente similares entre sí
  - Ejemplos en diferentes clusters sean altamente disimiles entre sí

# K-means

No Supervisado

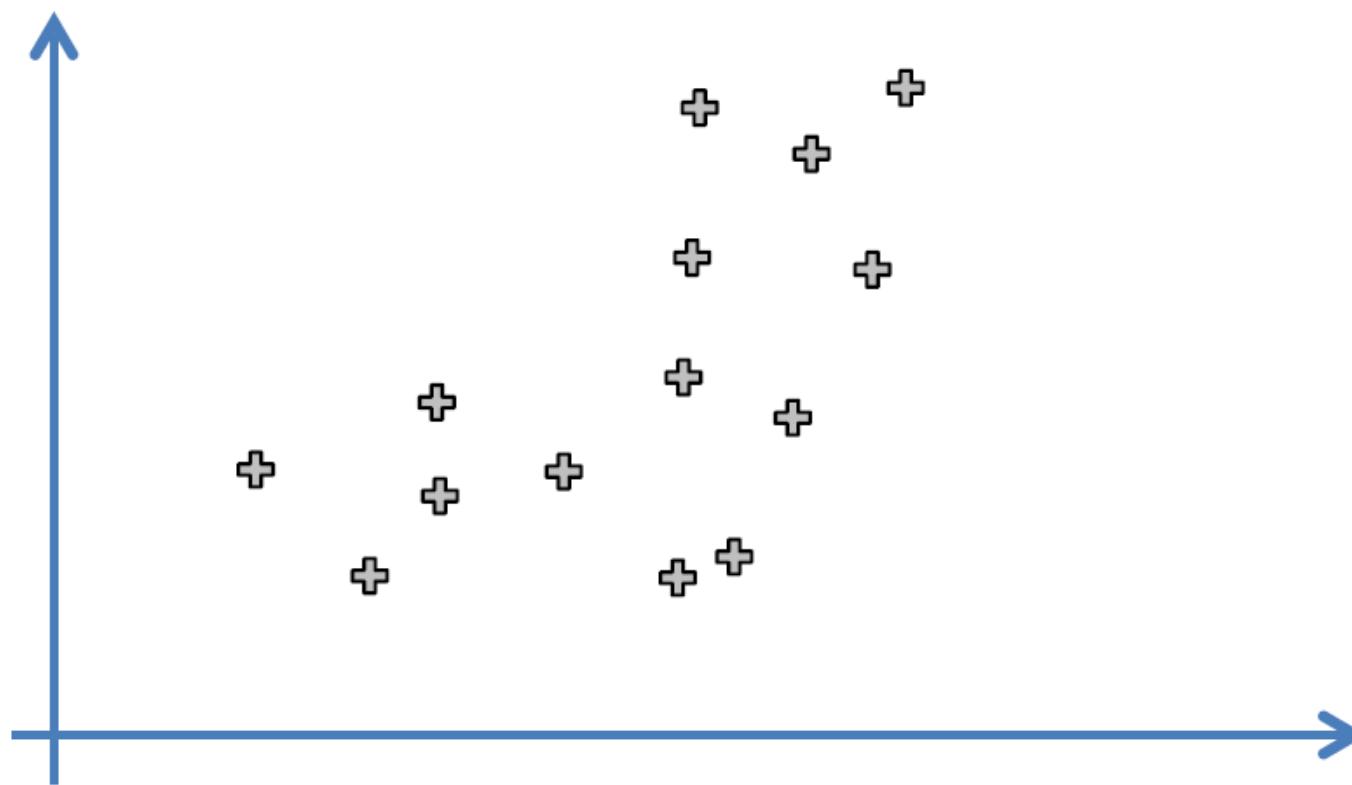
K-Means es un algoritmo **no supervisado** de Clustering. Se utiliza cuando tenemos un montón de datos sin etiquetar. El objetivo de este algoritmo es el de encontrar «K» grupos (clusters) entre los datos crudos. En este artículo repasaremos sus conceptos básicos y veremos un ejemplo paso a paso en python que podemos descargar.

# K-means



## K-means

**PASO 1:** Elegir el número K de clusters:  $K = 2$



## K-means

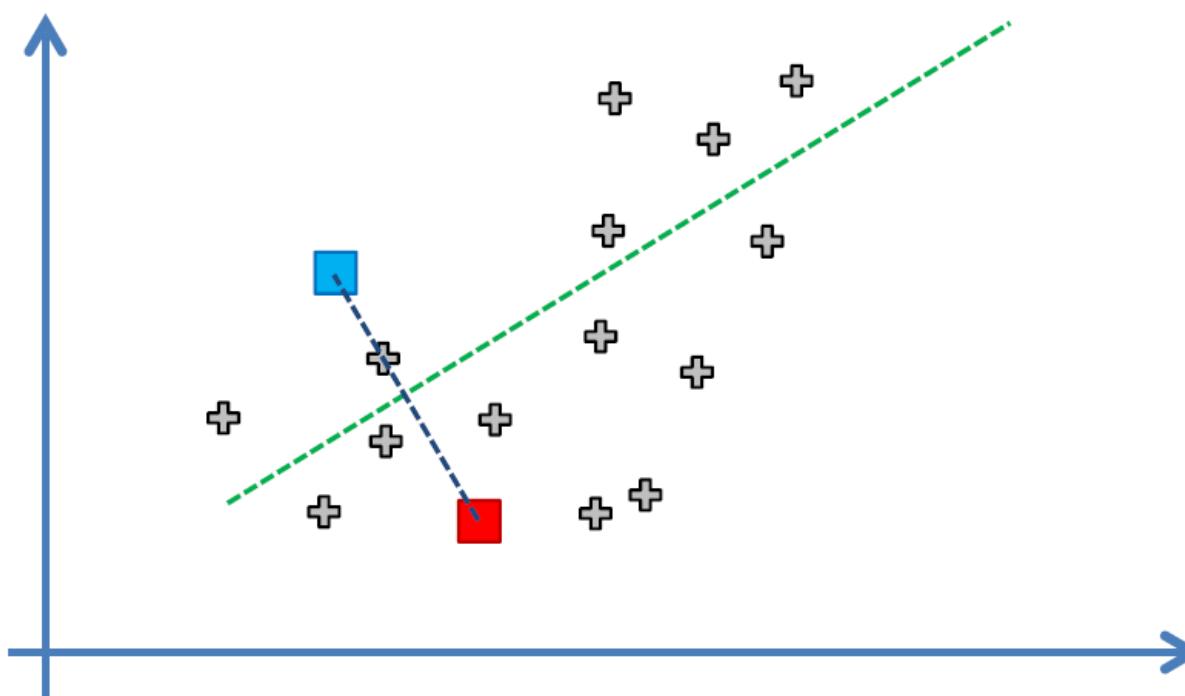
**PASO 2:** Seleccionar al azar K puntos, los baricentros (no necesariamente de nuestro dataset)



## K-means

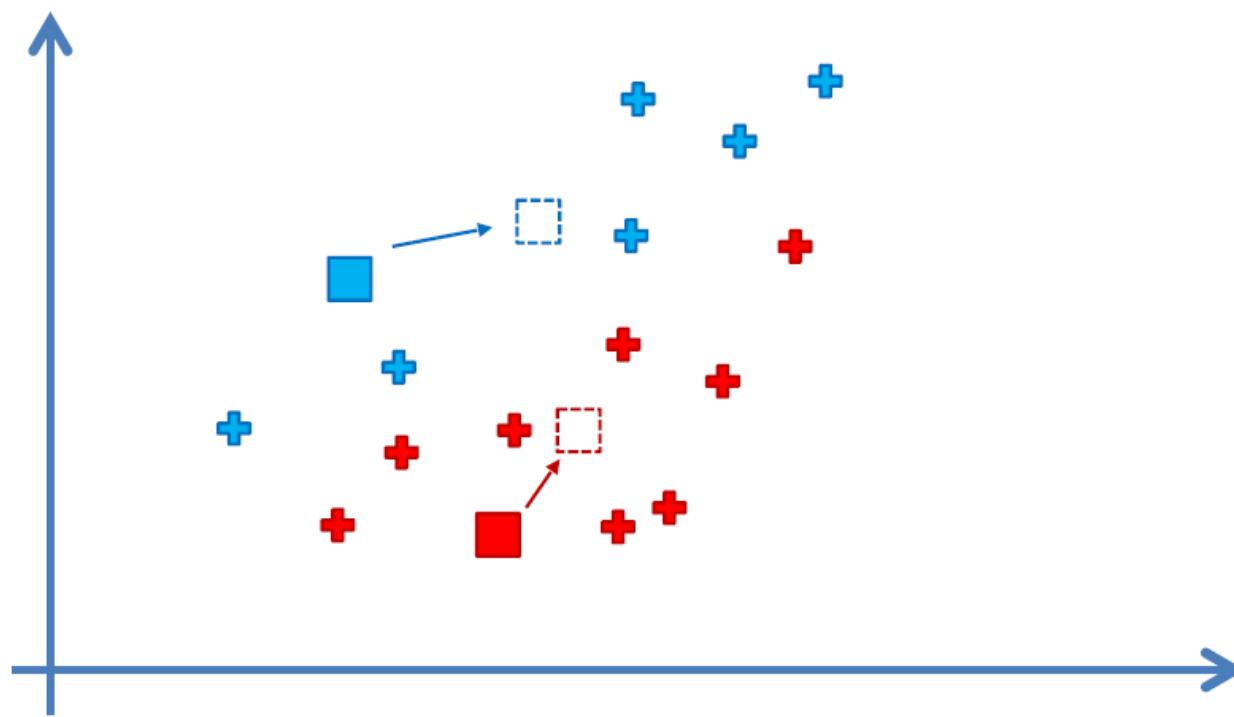
**PASO 3:** Asignar cada punto al baricentro más cercano

→ Esto formará los K clusters



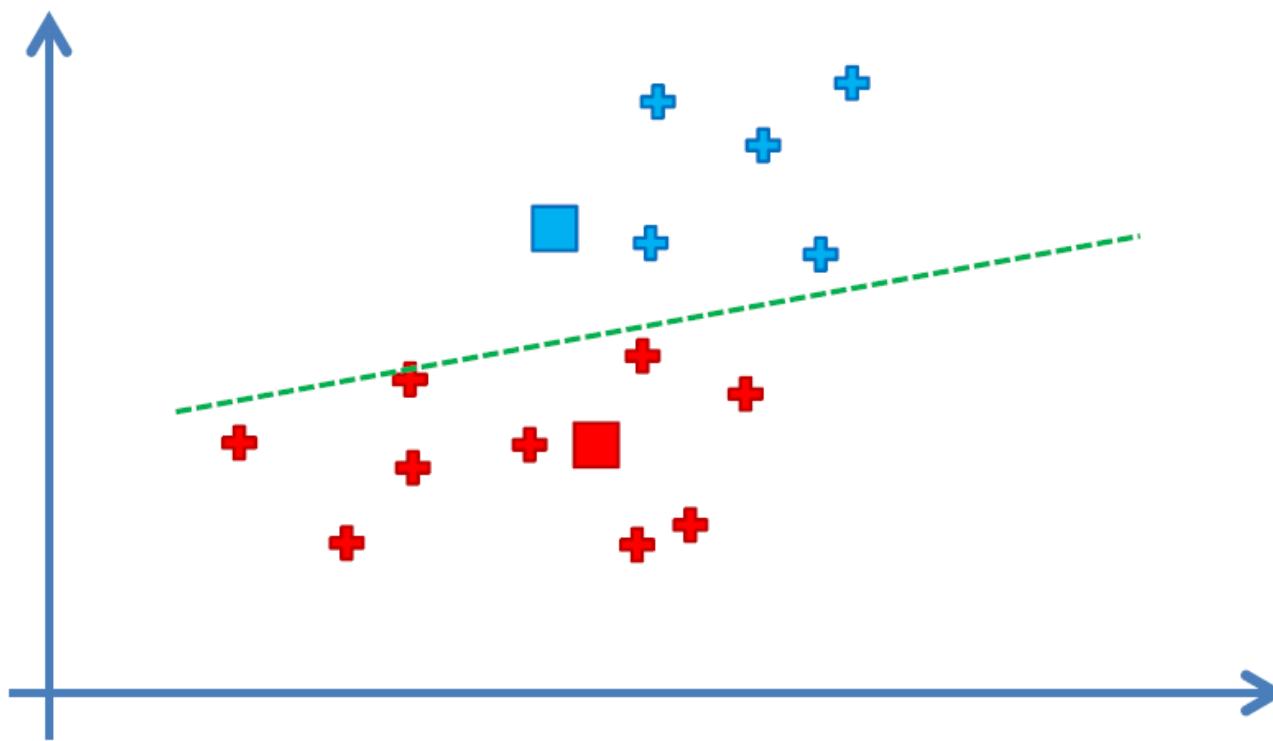
## K-means

**PASO 4:** Calcular y asignar el nuevo baricentro de cada cluster

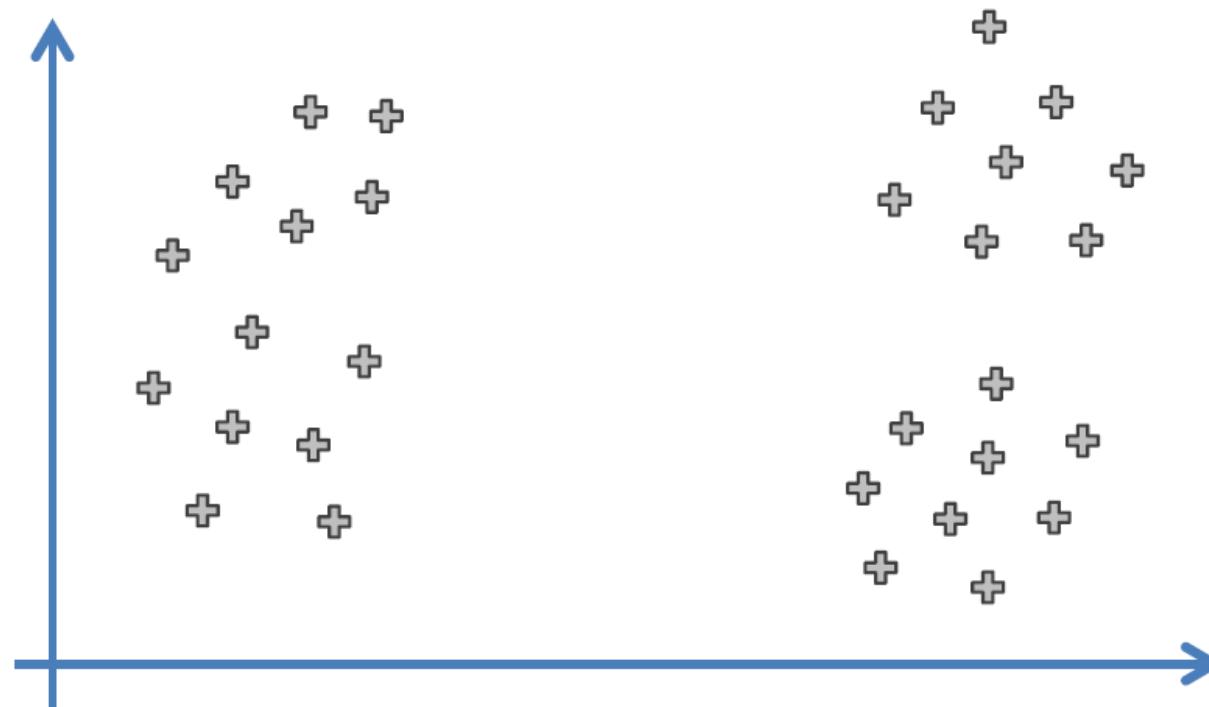


## K-means

**PASO 5:** Reasignar cada punto de los datos a su baricentro más cercano.  
Si ha habido nuevas asignaciones, ir al PASO 4, si no ir FIN.



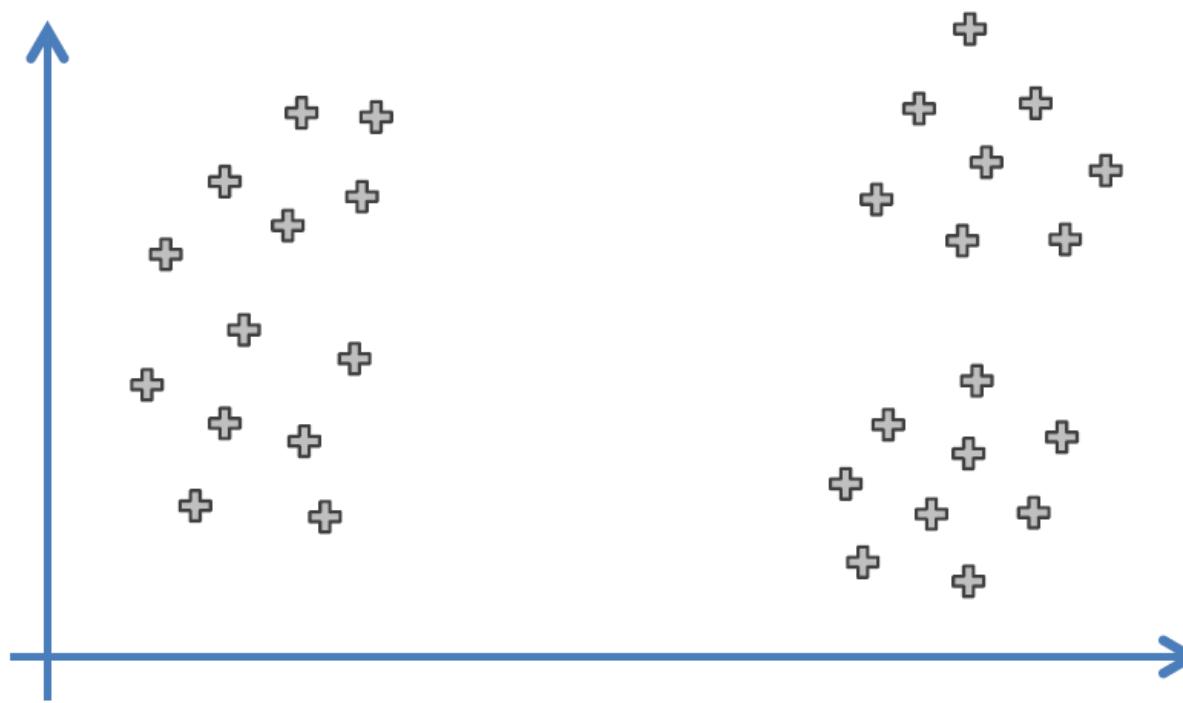
## La Trampa de la Inicialización Aleatoria



Si elegimos  $K = 3$  clusters...

# La Trampa de la Inicialización Aleatoria

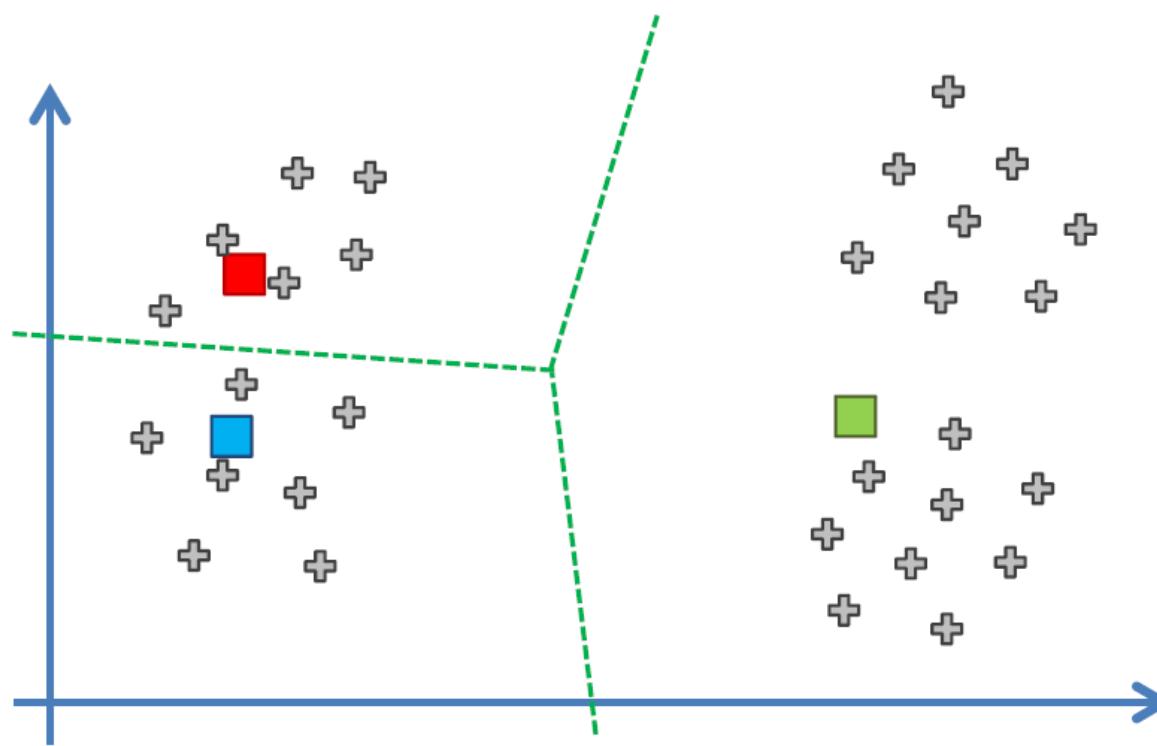
**PASO 1:** Elegir el número K de clusters:  $K = 3$



# La Trampa de la Inicialización

Aleatoria

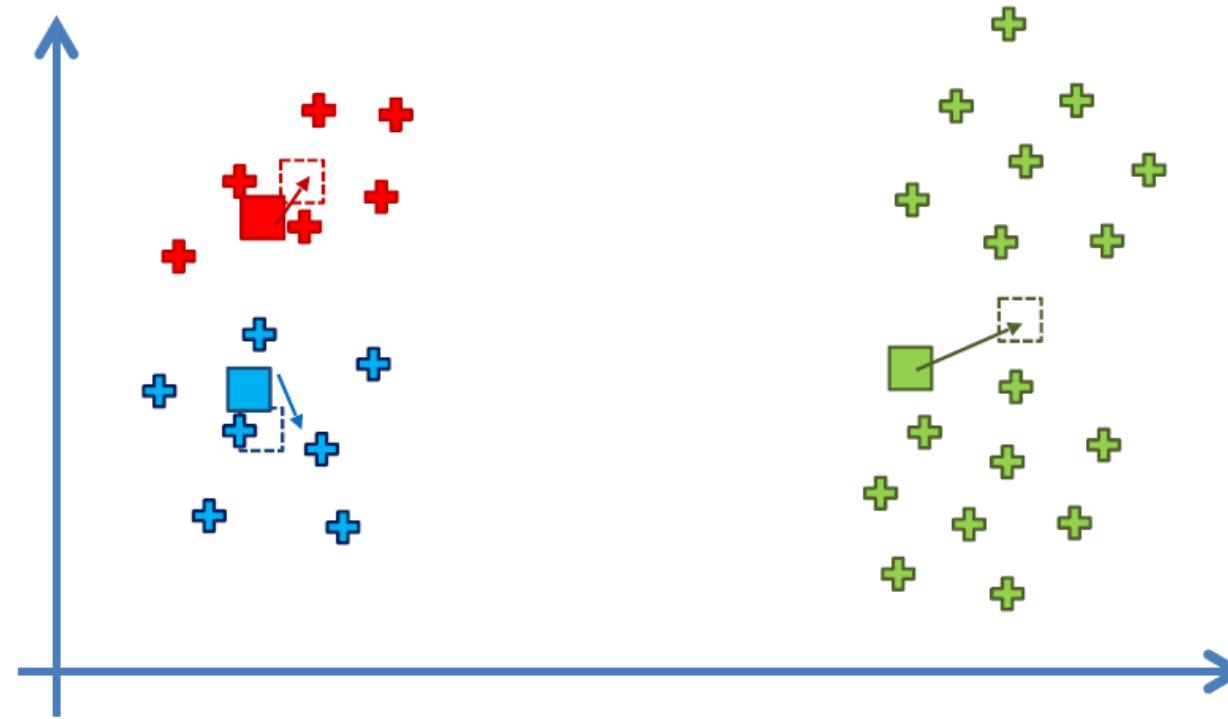
PASO 2: Seleccionar al azar K puntos, los baricentros (no necesariamente de nuestro dataset)



## La Trampa de la Inicialización

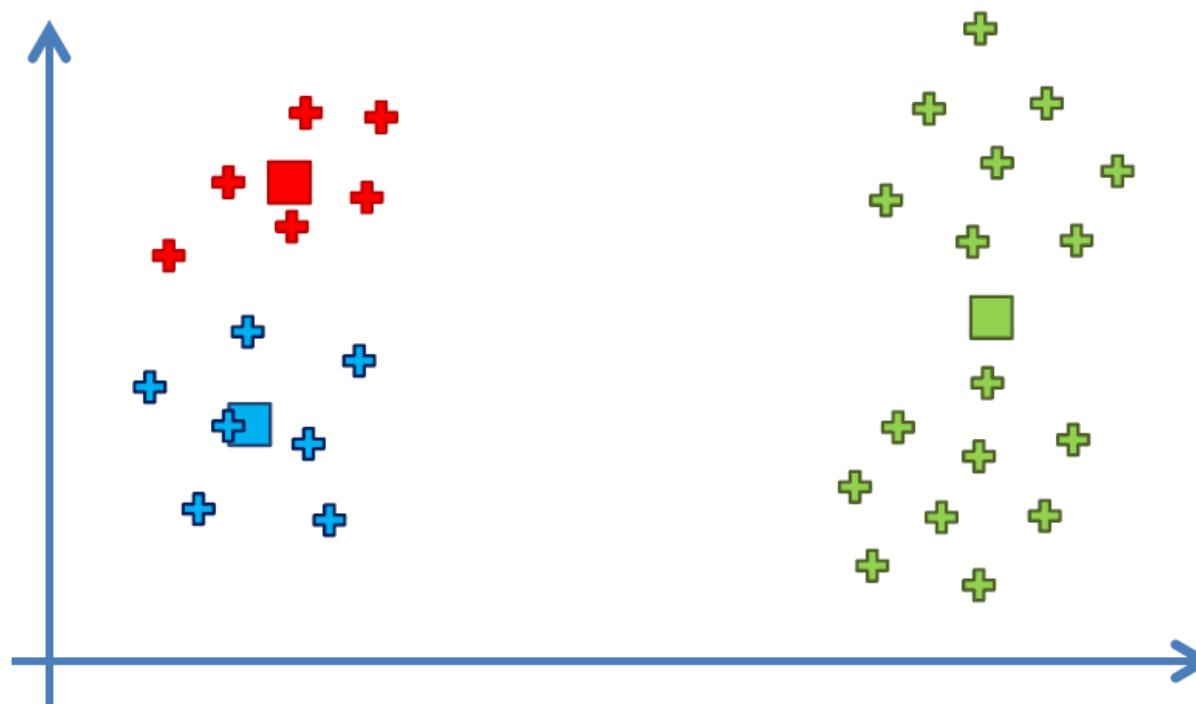
**PASO 3:** Asignar cada punto al baricentro más cercano

→ Esto formará los K clusters



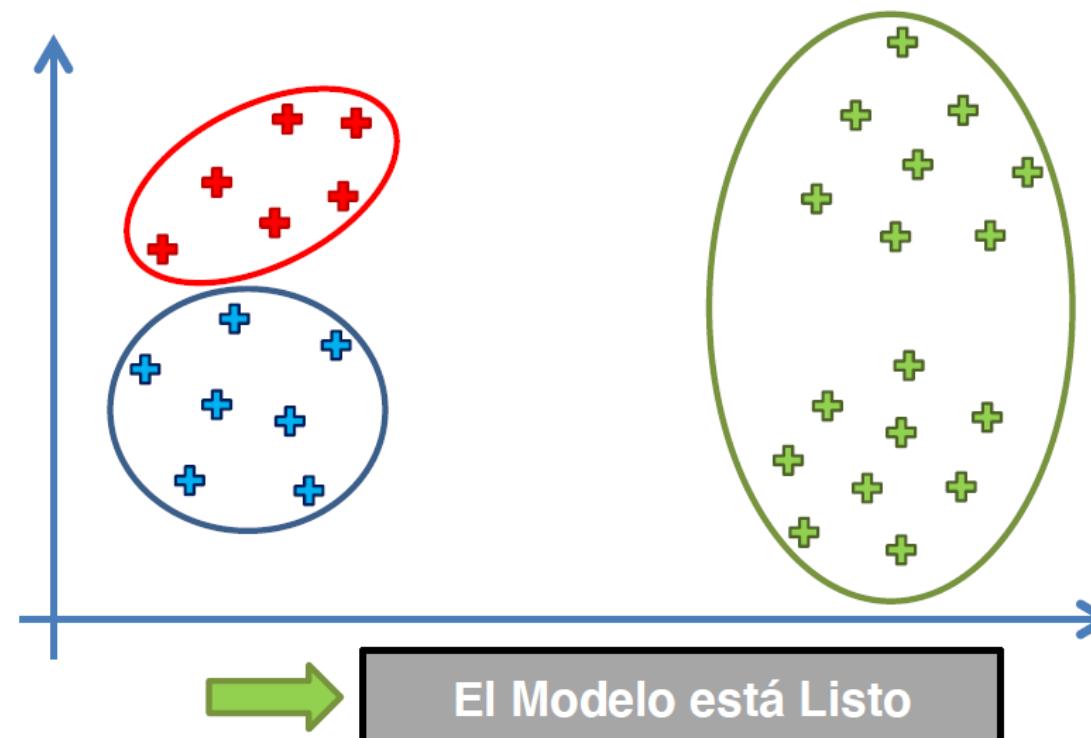
# La Trampa de la Inicialización

Aleatorio  
**PASO 4:** Calcular y asignar el nuevo baricentro de cada cluster



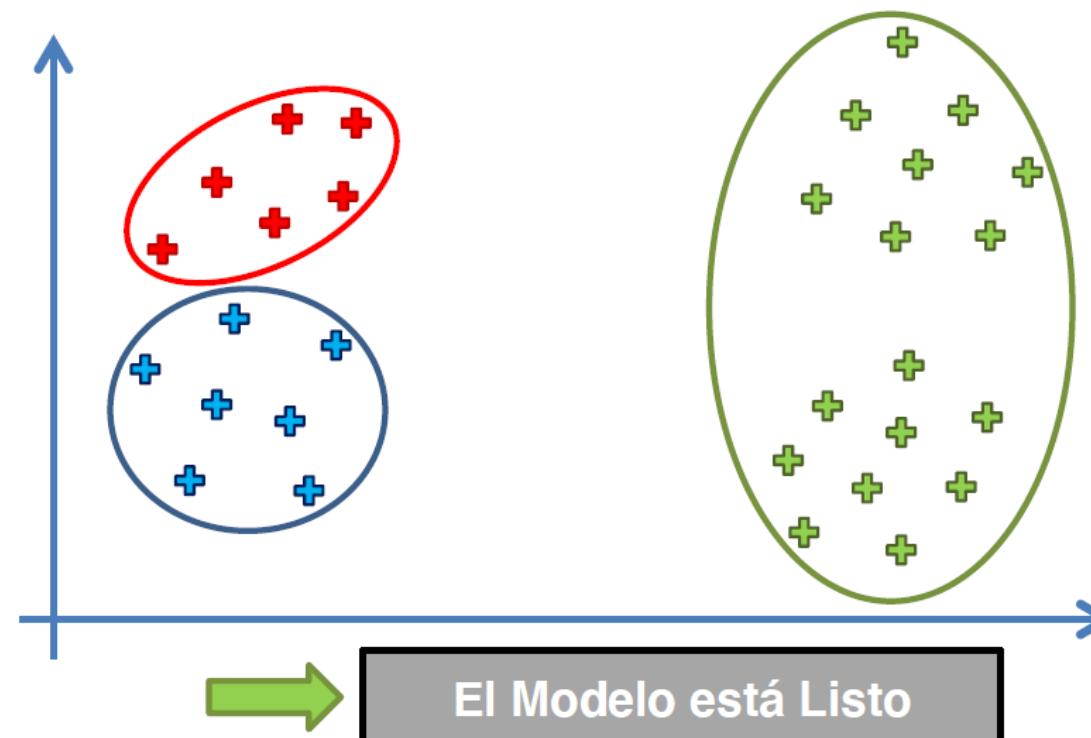
## La Trampa de la Inicialización

Ale PASO 5: Reasignar cada punto de los datos a su baricentro más cercano.  
Si ha habido nuevas asignaciones, ir al PASO 4, si no ir FIN.



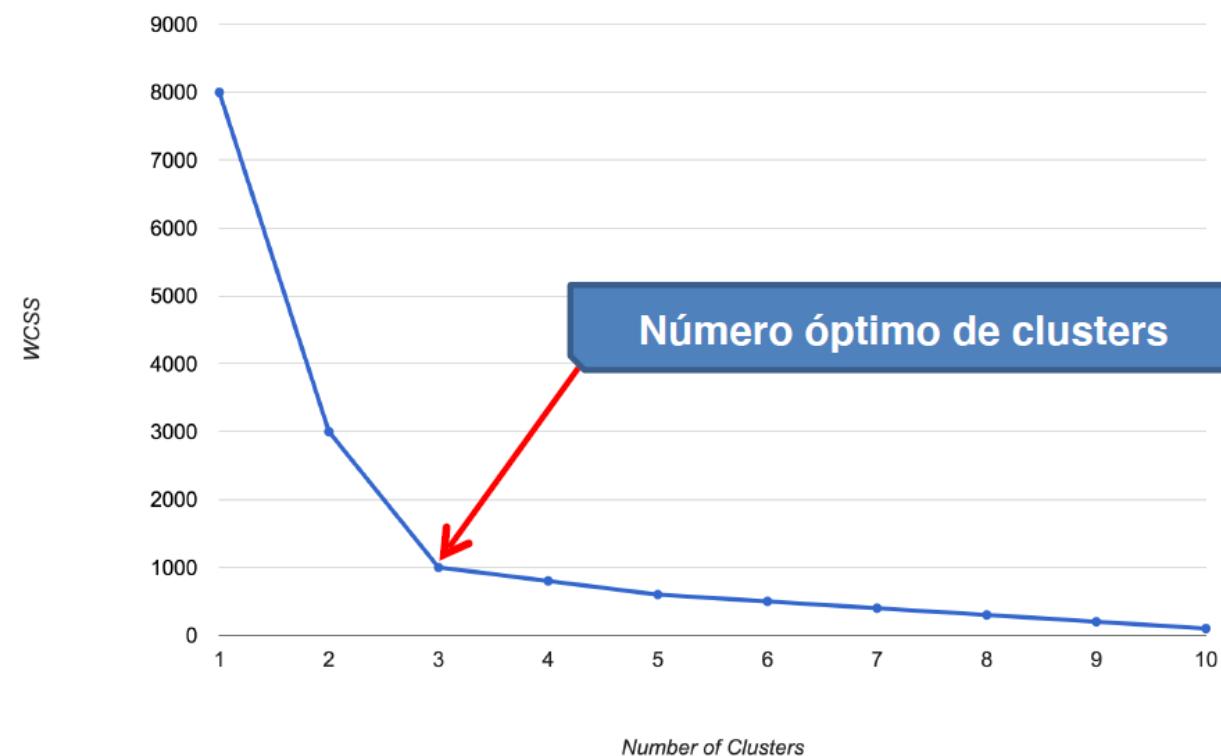
## La Trampa de la Inicialización

Ale PASO 5: Reasignar cada punto de los datos a su baricentro más cercano.  
Si ha habido nuevas asignaciones, ir al PASO 4, si no ir FIN.



# Elegir el número correcto de clusters

## La técnica del codo



# ¿DÓNDE Y CUÁNDO APLICARLA?

El algoritmo de **Clustering K-means** es uno de los más usados para encontrar grupos ocultos, o sospechados en teoría sobre un conjunto de datos no etiquetado. Esto puede servir para confirmar - o desterrar alguna teoría que teníamos asumida de nuestros datos. Y también puede ayudarnos a descubrir relaciones asombrosas entre conjuntos de datos, que de manera manual, no hubiéramos reconocido. Una vez que el algoritmo ha ejecutado y obtenido las etiquetas, será fácil clasificar nuevos valores o muestras entre los grupos obtenidos.



## Ventajas y Desventajas

### Algunas Ventajas:

- Entre los algoritmos de particionamiento es eficiente
- Implementación sencilla.

### Contras:

- Necesito conocer k de antemano
- Sensible a ruido
- El resultado puede variar en base a las semillas elegidas al inicio
- Algunas semillas pueden resultar en una tasa de convergencia menor
- La selección de semillas se puede basar en heurísticas o resultados obtenidos por otros métodos
- Puede caer en mínimos locales
- No trata datos nominales (K-Modes).

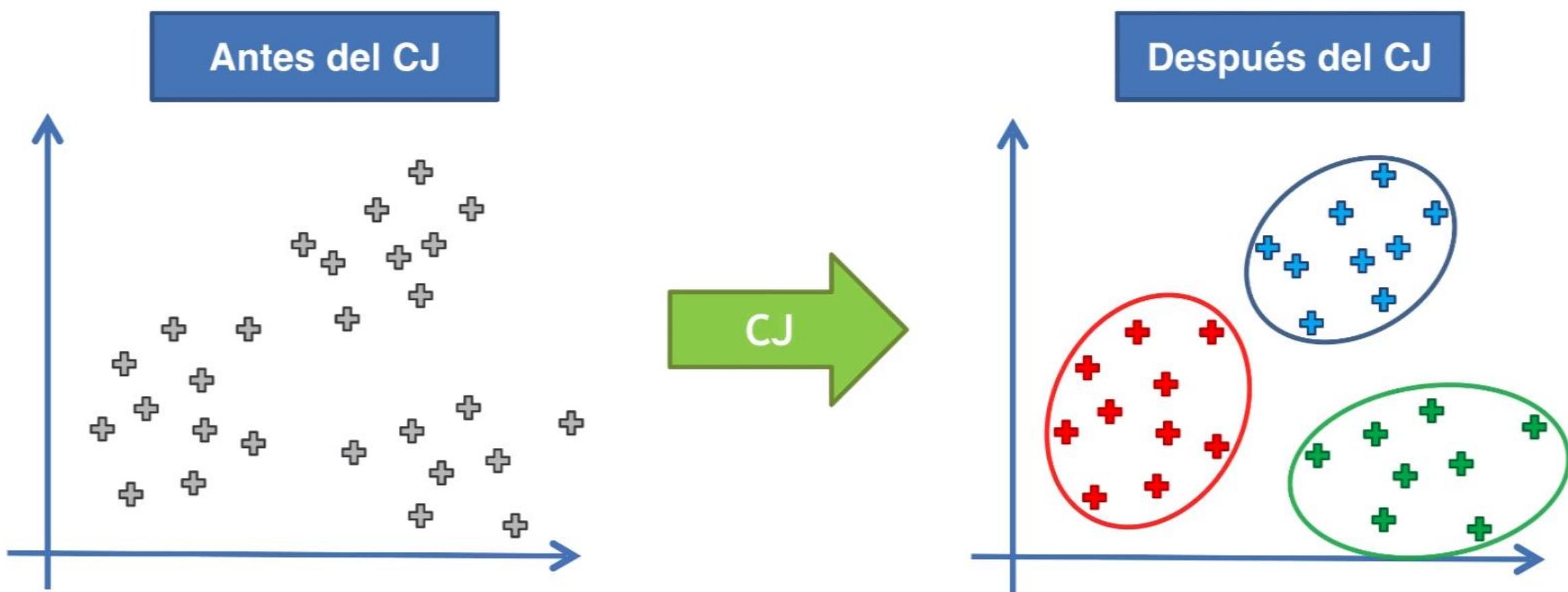


El algoritmo de **clúster jerárquico** agrupa los datos basándose en la distancia entre cada uno y buscando que los datos que están dentro de un clúster sean los más similares entre sí.

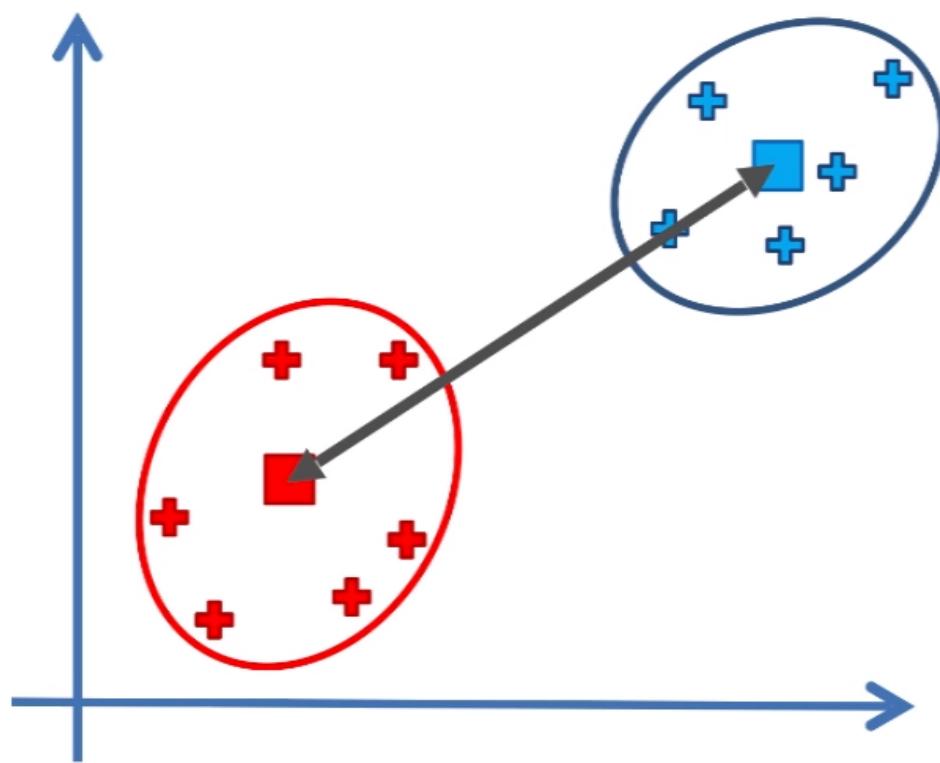
## Clustering Jeraquico

Una cantidad de Arboles de  
Decisión

# Cómo funciona el Clustering



# Cómo funciona el Clustering Jeraquico



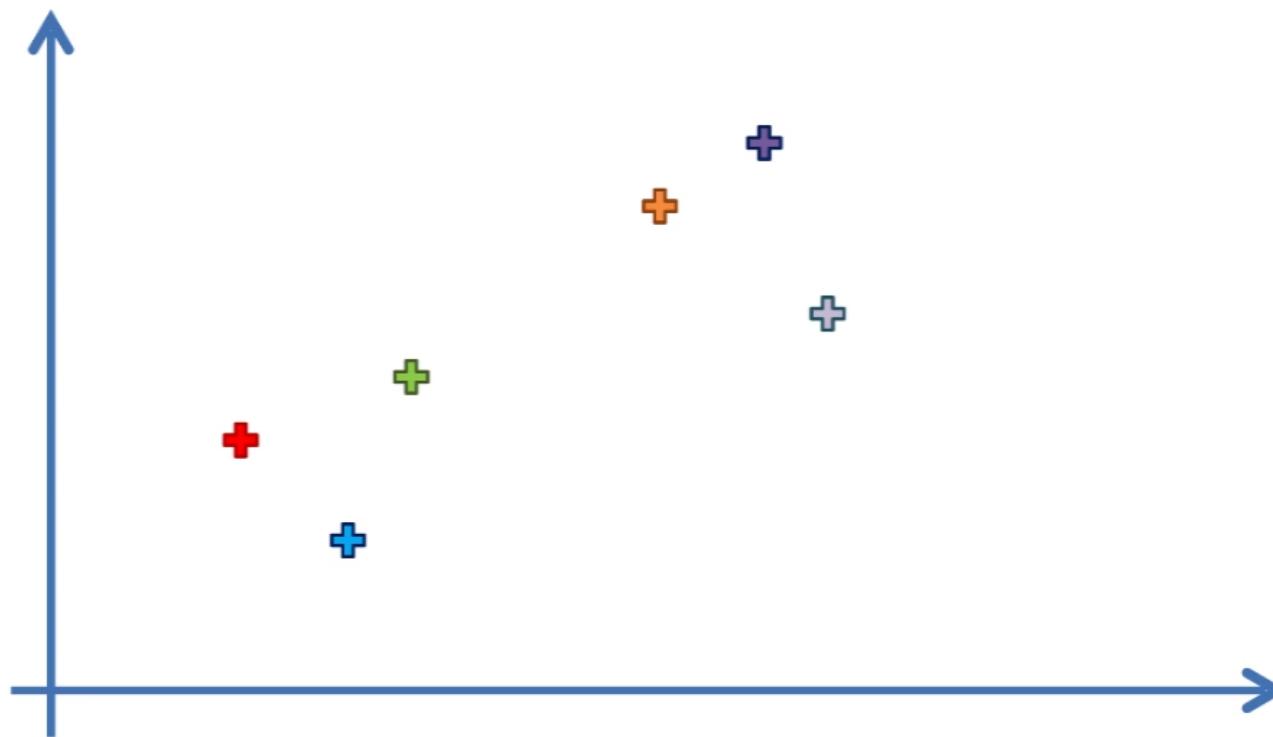
## Distancia entre dos Clusters:

- Opción 1: Puntos más cercanos
- Opción 2: Puntos más alejados
- Opción 3: Distancia media
- Opción 4: Distancia entre sus baricentros

# Cómo funciona el Clustering

PASO 1: Hacer que cada punto sea un propio cluster.

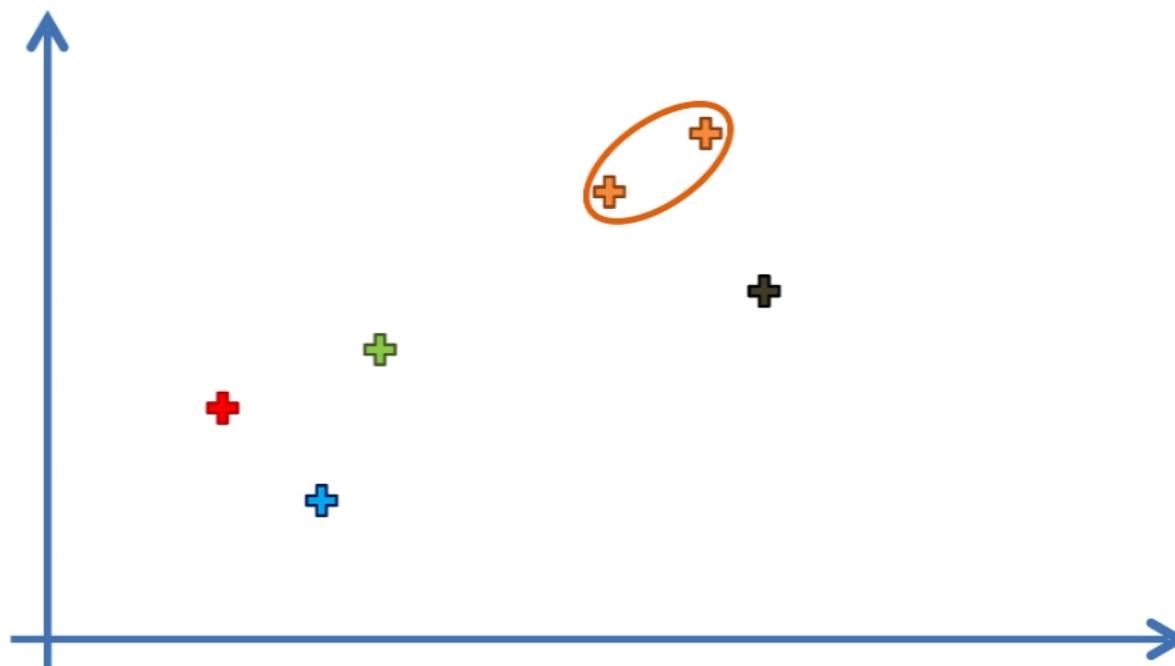
→ Así tendremos 6 clusters



# Cómo funciona el Clustering Jerárquico

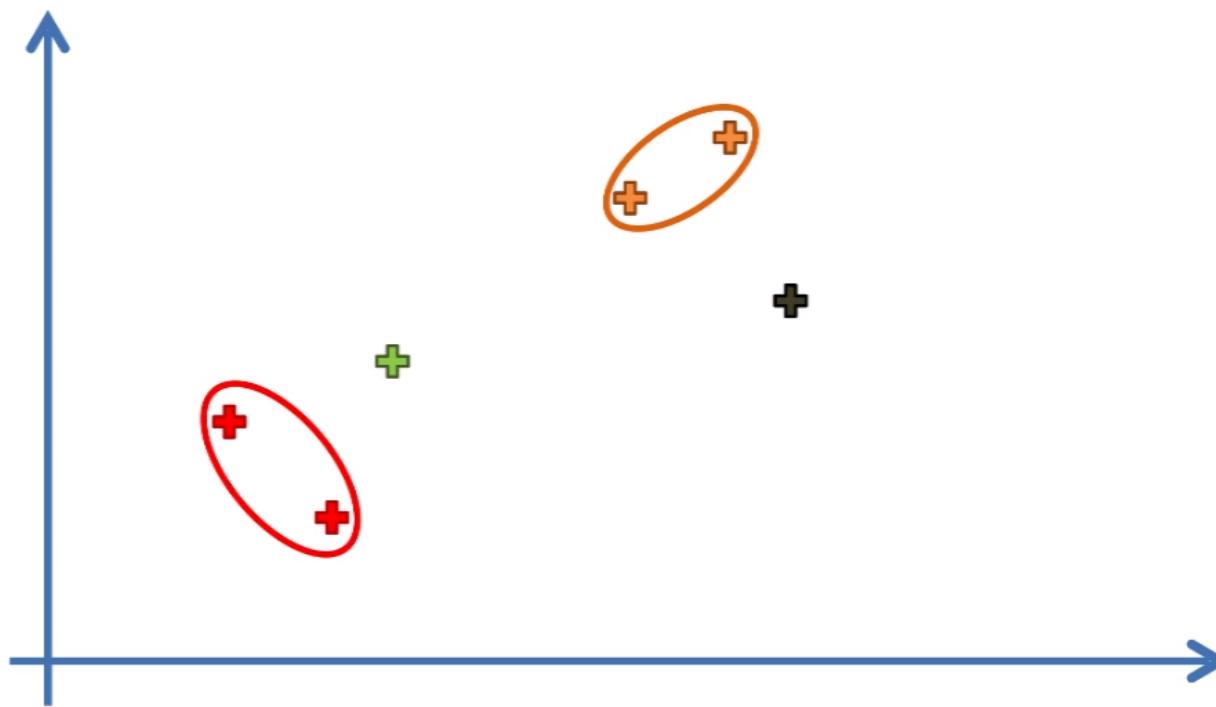
PASO 2: Elegir los dos puntos más cercanos y juntarlos en un único cluster

→ Así nos quedan 5 clusters



## Cómo funciona el Clustering

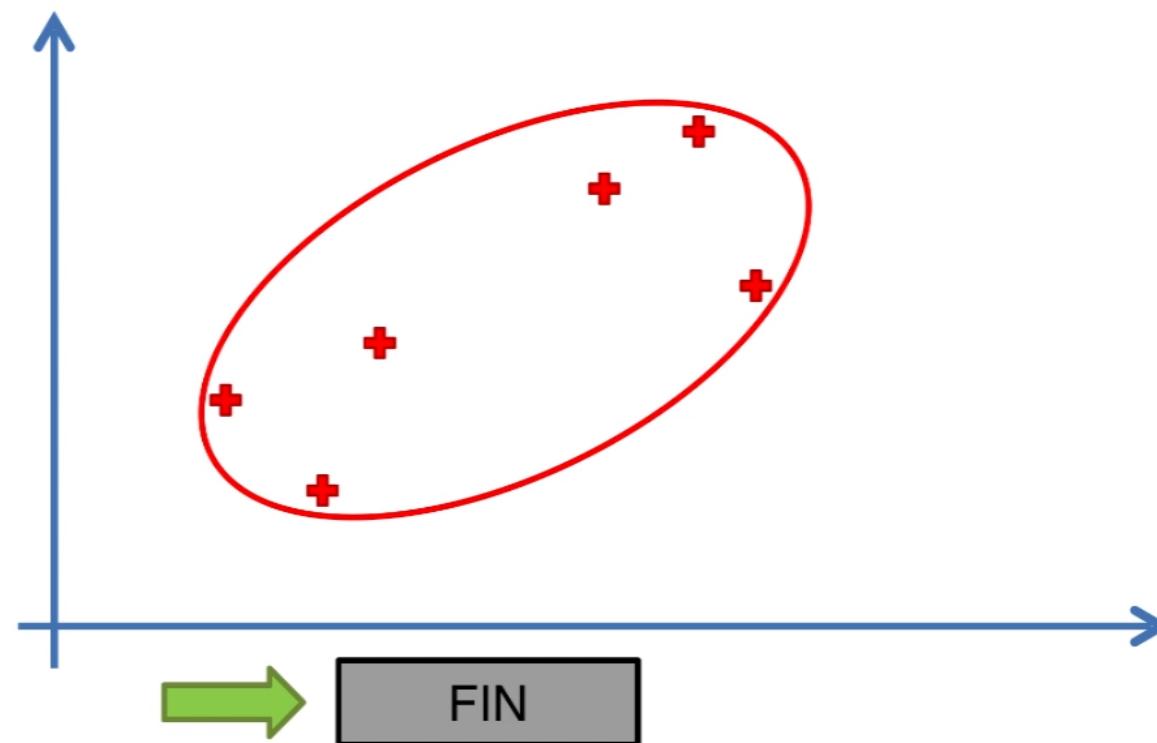
**JPASO 3:** Elegir los dos clusters más cercanos y juntarlos en un único cluster  
→ Así tenemos 4 clusters



# Cómo funciona el Clustering

## Jerárquico

**PASO 4:** Repetir el PASO 3 hasta que quede un solo cluster



# ¿DÓNDE Y CUÁNDO APLICARLA?

**Opción 1:** Segmentar grupos de personas de acuerdo a sus intereses de compras.

**Opción 2:** Determinar el comportamiento de votación del senado de una comunidad.

**Opción 3:** Separar a personas reales de los bots presentes en redes sociales.



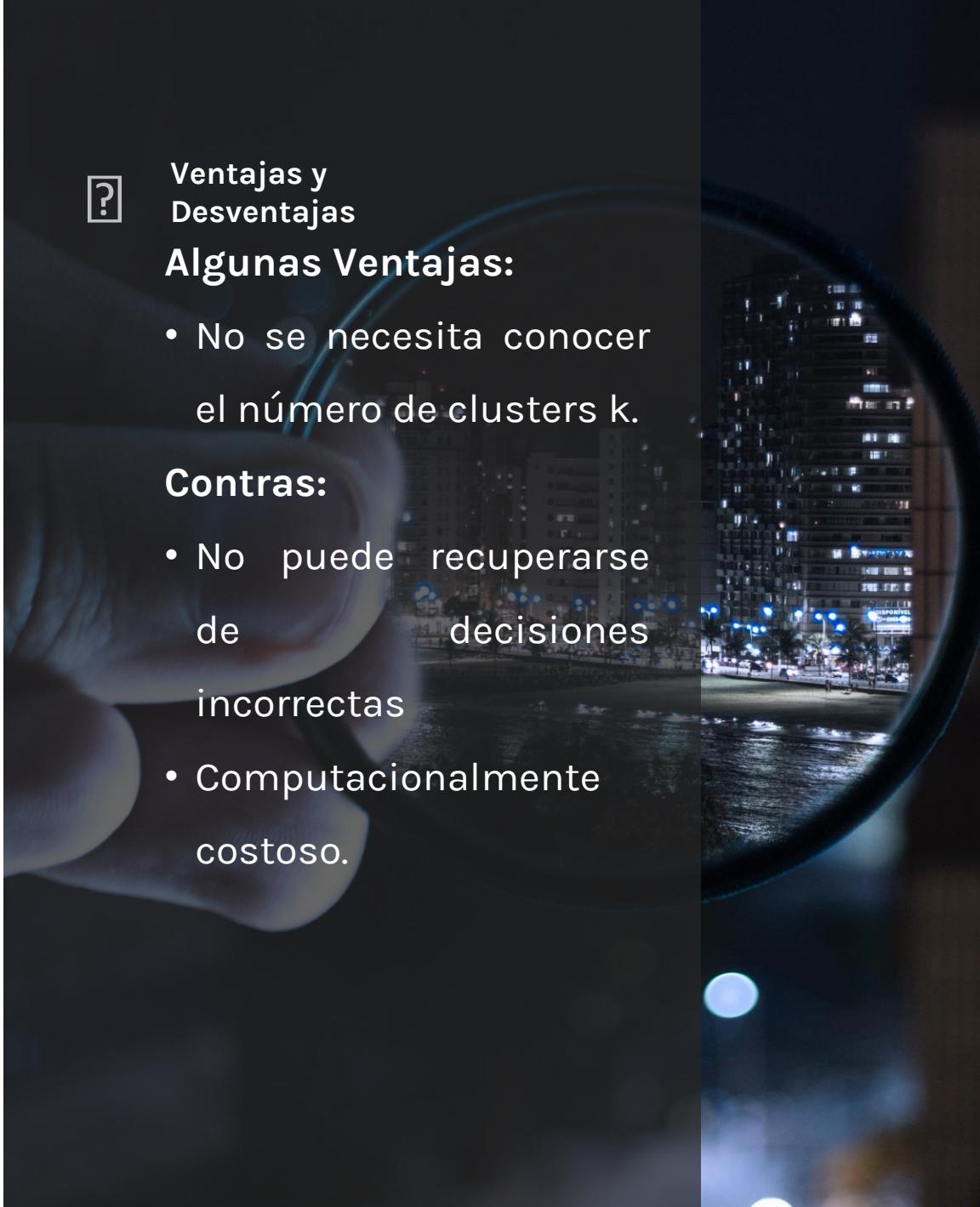
Ventajas y Desventajas

## Algunas Ventajas:

- No se necesita conocer el número de clusters k.

## Contras:

- No puede recuperarse de decisiones incorrectas
- Computacionalmente costoso.



## Aprendizaje por Reglas de Asociación

Reglas de asociación es una técnica de inteligencia artificial ampliamente utilizada en Data Mining.



```

5 #author: helmer
6 """
7 # Regresión Logística
8
9 # Cómo importar las librerías
10 import numpy as np
11 import matplotlib.pyplot as plt
12 import pandas as pd
13
14 # Importar el data set
15 dataset = pd.read_csv('Social_Network_Ads.csv')
16
17 X = dataset.loc[:,['User ID','Gender','EstimatedSalary','Purchased']].values
18 y = dataset.iloc[:, 4].values
19
20
21
22

```

dataset - DataFrame					
Índice	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0

X - Arreglo de NumPy (sólo lectura)				
0	1	2	3	
0	15624510	Male	19000.0	0
1	15810944	Male	20000.0	0
2	15668575	Female	43000.0	0
3	15603246	Female	57000.0	0
4	15804002	Male	76000.0	0
5	15728773	Male	58000.0	0
6	15598044	Female	84000.0	0
7	15694829	Female	150000.0	1
8	15600575	Male	33000.0	0
9	15727311	Female	65000.0	0
10	15570769	Female	80000.0	0

# Apriori

Árboles en cantidad

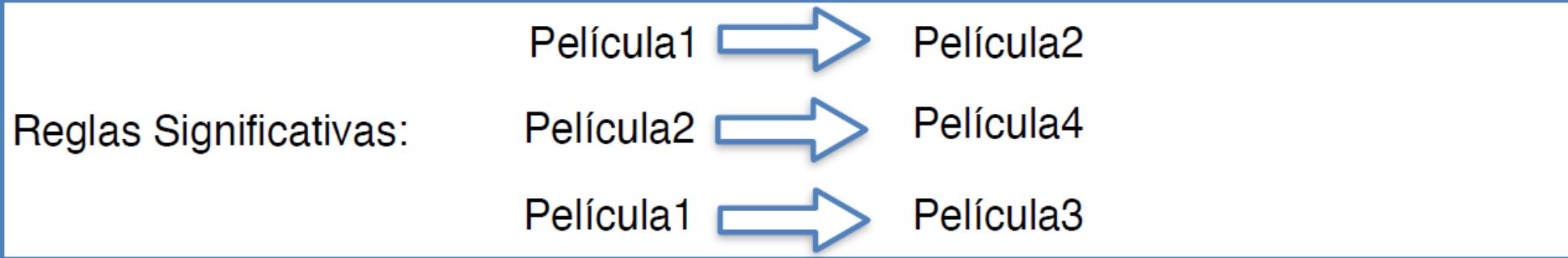
Es la combinación de **árboles predictores** tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

# Apriori



# Apriori

User ID	Películas que le han gustado
46578	Película1, Película2, Película3, Película4
98989	Película1, Película2
71527	Película1, Película2, Película4
78981	Película1, Película2
89192	Película2, Película4
61557	Película1, Película3



# Apriori

Transaction ID	Productos comprados
46578	Hamburguesas, Patatas, Verduras
98989	Hamburguesas, Patatas, Ketchup
71527	Verduras, Fruta
78981	Pasta, Fruta, Mantequilla, Verduras
89192	Hamburguesas, Pasta, Patatas
61557	Fruta, Zumo de Naranja, Verduras
87923	Hamburguesas, Patatas, Ketchup, Mayo

Reglas Significativas:

Hamburguesas	→	Patatas
Verduras	→	Fruta
Hamburguesas, Patatas	→	Ketchup

# Apriori

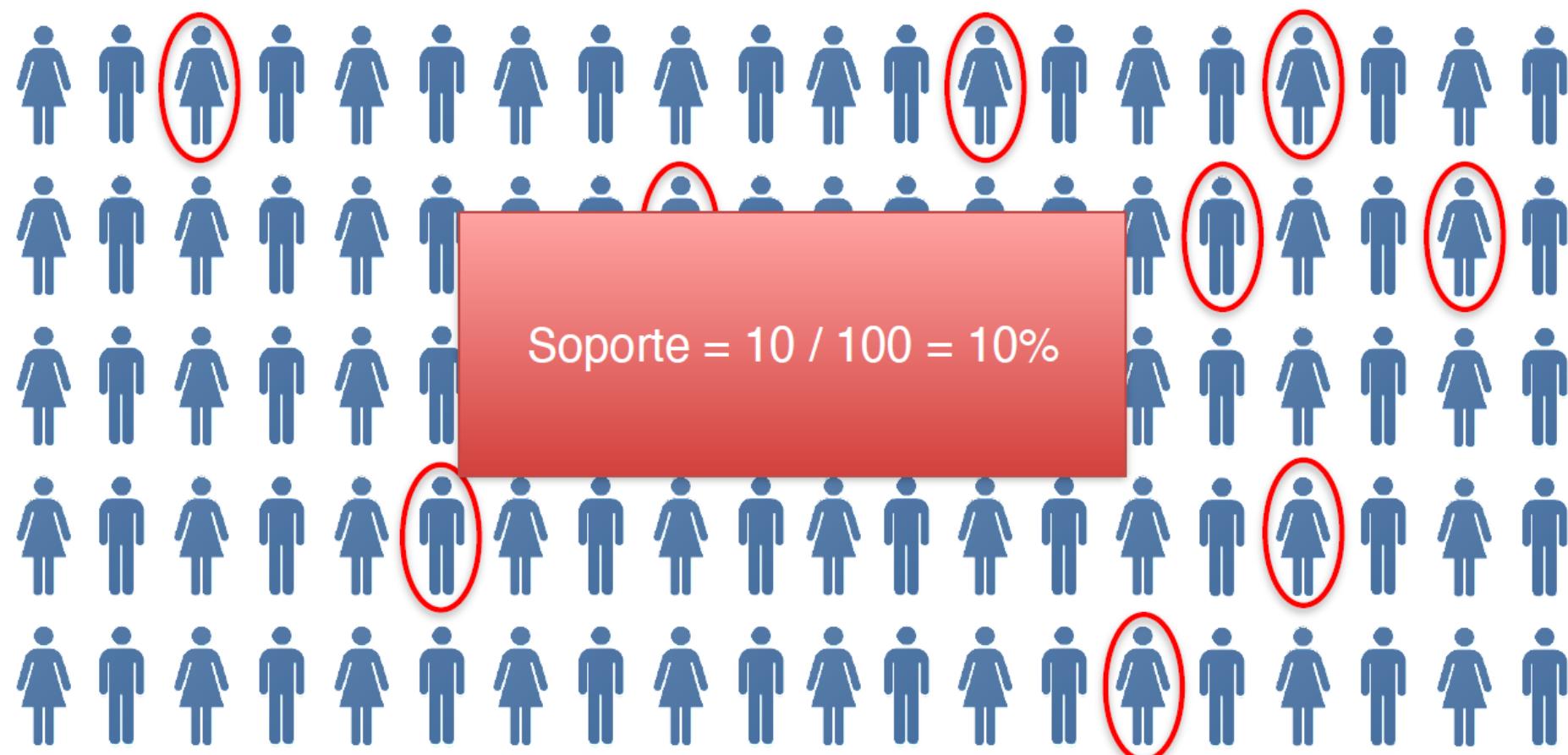
Recomendación de Películas:

$$sop(\mathbf{M}) = \frac{|\text{usuarios que vieron } \mathbf{M}|}{|\text{usuarios}|}$$

Optimización de la Cesta de la Compra:

$$sop(\mathbf{I}) = \frac{|\text{transacciones que contienen } \mathbf{I}|}{|\text{transacciones}|}$$

## Apriori



# Apriori

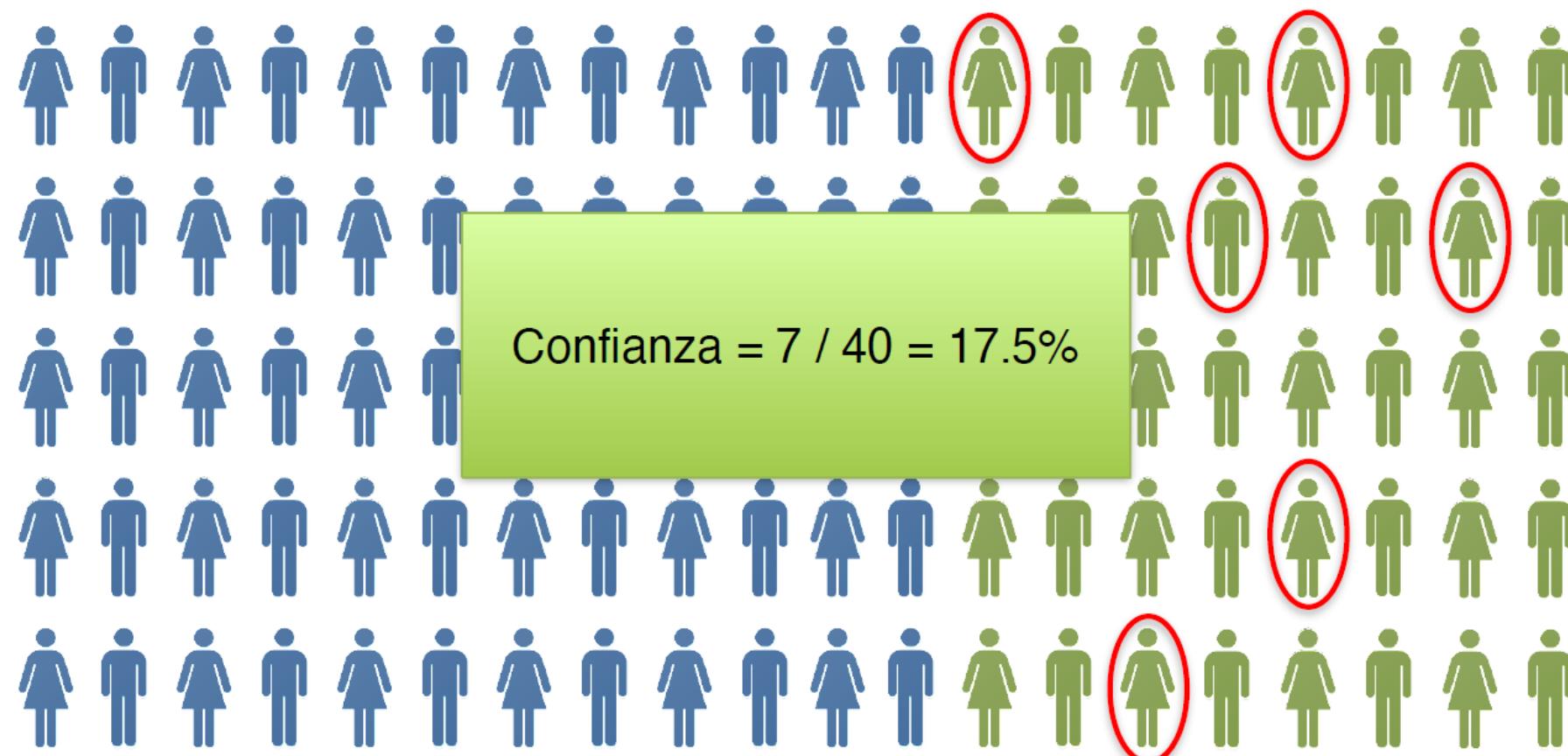
Recomendación de Películas:

$$conf(\mathbf{M}_1 \Rightarrow \mathbf{M}_2) = \frac{|\text{usuarios que vieron } \mathbf{M}_1 \text{ y } \mathbf{M}_2|}{|\text{usuarios que vieron } \mathbf{M}_1|}$$

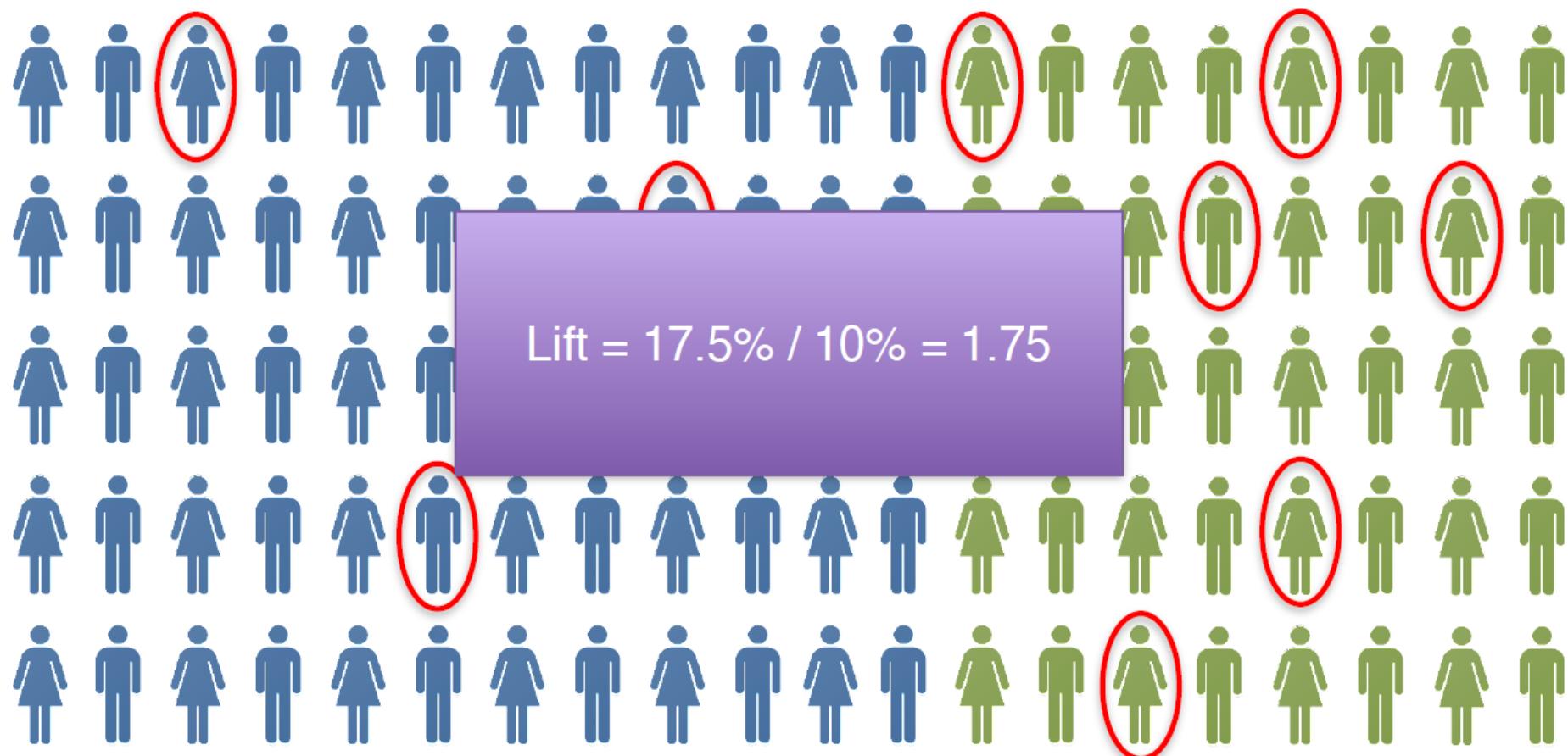
Optimización de la Cesta de la Compra:

$$conf(\mathbf{I}_1 \Rightarrow \mathbf{I}_2) = \frac{|\text{transacciones que contienen } \mathbf{I}_1 \text{ y } \mathbf{I}_2|}{|\text{transacciones que contienen } \mathbf{I}_1|}$$

## Apriori



## Apriori



# ¿DÓNDE Y CUÁNDO APLICARLA?

- Desarrollo de investigación Genética.
- Patologías de enfermedades
- Administración Financiera
- Estudio de Patrones laborales
- Efectos de la causalidad en áreas de trabajo

## ?

### Ventajas y Desventajas

#### Algunas Ventajas:

- Este es el más simple y algoritmo fácil de entender entre algoritmos de aprendizaje de reglas de asociación
- Las reglas resultantes son intuitivas y fáciles de comunicar a un usuario final.
- No requiere datos etiquetados ya que está completamente sin supervisión;
- Se propusieron muchas extensiones para diferentes casos de uso basados en esta implementación.
- El algoritmo es exhaustivo, por lo que encuentra todas las reglas con el soporte y la confianza especificados.

#### Contras:

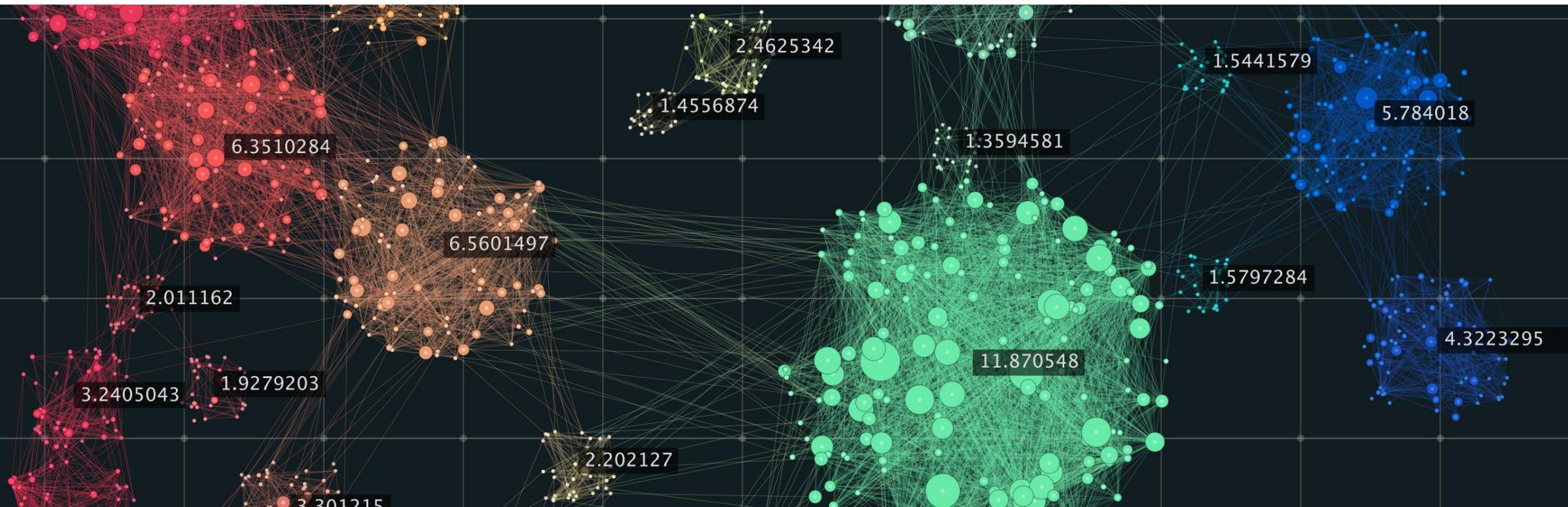
- Muchas asociaciones falsas si el dataset es pequeño.
- Todos los subconjuntos de un conjunto de elementos frecuentes deben ser frecuentes



Este Algoritmo significa **Agrupación de clase de equivalencia** y **Recorrido de celosía ascendente**. Es uno de los métodos populares de minería de reglas de asociación . Es una versión más eficiente y escalable del algoritmo Apriori.

## Eclat

Apriori con esteroides



# Eclat

User ID	Películas que le han gustado
46578	Película1, Película2, Película3, Película4
98989	Película1, Película2
71527	Película1, Película2, Película4
78981	Película1, Película2
89192	Película2, Película4
61557	Película1, Película3

Reglas Significativas:

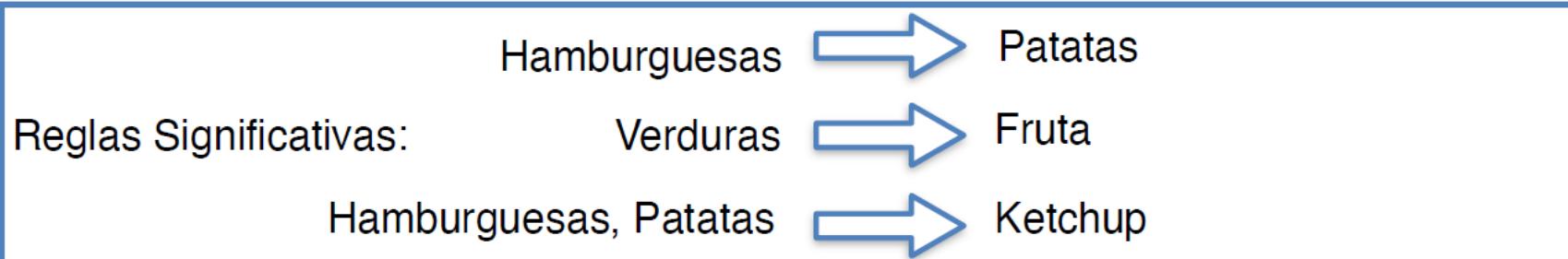
Película1 → Película2

Película2 → Película4

Película1 → Película3

# Eclat

Transaction ID	Productos comprados
46578	Hamburguesas, Patatas, Verduras
98989	Hamburguesas, Patatas, Ketchup
71527	Verduras, Fruta
78981	Pasta, Fruta, Mantequilla, Verduras
89192	Hamburguesas, Pasta, Patatas
61557	Fruta, Zumo de Naranja, Verduras
87923	Hamburguesas, Patatas, Ketchup, Mayo



## Eclat

Recomendación de Películas:

$$sop(\mathbf{M}) = \frac{|\text{usuarios que vieron } \mathbf{M}|}{|\text{usuarios}|}$$

Optimización de la Cesta de la Compra:

$$sop(\mathbf{I}) = \frac{|\text{transacciones que contienen } \mathbf{I}|}{|\text{transacciones}|}$$

# ¿DÓNDE Y CUÁNDO APLICARLA?

- Desarrollo de investigación Genética.
- Patologías de enfermedades
- Administración Financiera
- Estudio de Patrones laborales
- Efectos de la causalidad en áreas de trabajo

## ?

### Ventajas y Desventajas

#### Algunas Ventajas:

- Este es el más simple y algoritmo fácil de entender entre algoritmos de aprendizaje de reglas de asociación
- Las reglas resultantes son intuitivas y fáciles de comunicar a un usuario final.
- No requiere datos etiquetados ya que está completamente sin supervisión;
- Se propusieron muchas extensiones para diferentes casos de uso basados en esta implementación.
- El algoritmo es exhaustivo, por lo que encuentra todas las reglas con el soporte y la confianza especificados.

#### Contras:

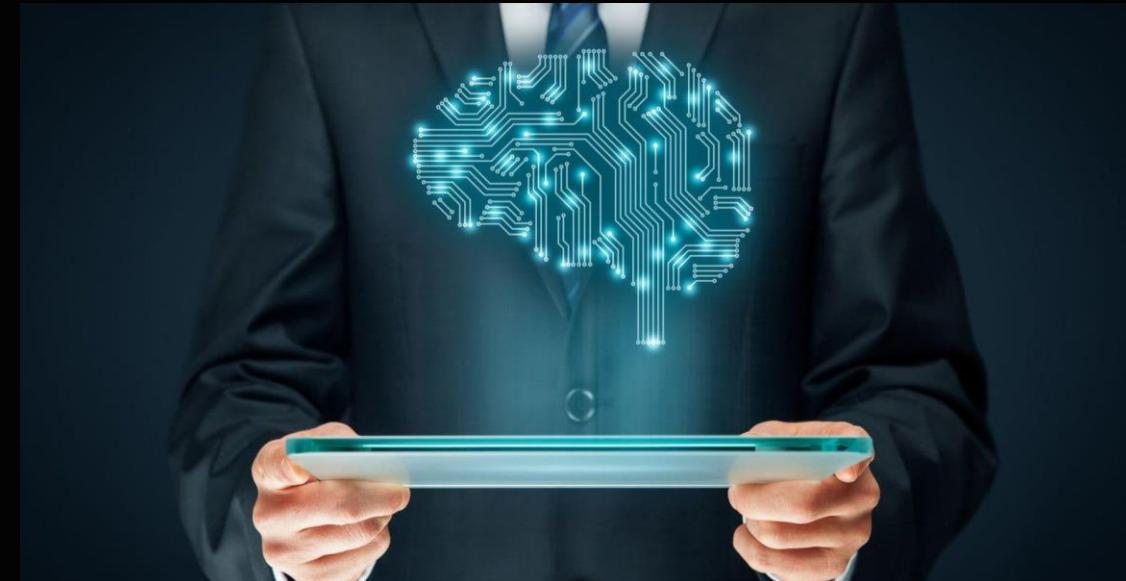
- Muchas asociaciones falsas si el dataset es pequeño.
- Todos los subconjuntos de un conjunto de elementos frecuentes deben ser frecuentes





## Reinforcement Learning

Se trata de tomar medidas adecuadas para maximizar la recompensa en una situación particular. Es utilizado por varios softwares y máquinas para encontrar el mejor comportamiento o ruta posible que debe tomar en una situación específica.



## Aprendizaje por Refuerzo

El aprendizaje por refuerzo difiere del aprendizaje supervisado en una forma en que en el aprendizaje supervisado los datos de entrenamiento tienen la clave de respuesta, por lo que el modelo se entrena con la respuesta correcta en sí misma, mientras que en el aprendizaje por refuerzo, no hay respuesta, pero el agente de refuerzo decide qué hacer.

# Upper Confidence Bound (UCB)

Reforzando el aprendizaje

En el **aprendizaje por refuerzo**, el agente o la persona que toma las decisiones genera sus datos de capacitación al interactuar con el mundo. El agente debe conocer las consecuencias de sus acciones a través de prueba y error, en lugar de que se le diga explícitamente la acción correcta.

# Upper Confidence Bound



# Upper Confidence Bound



D1



D2



D3

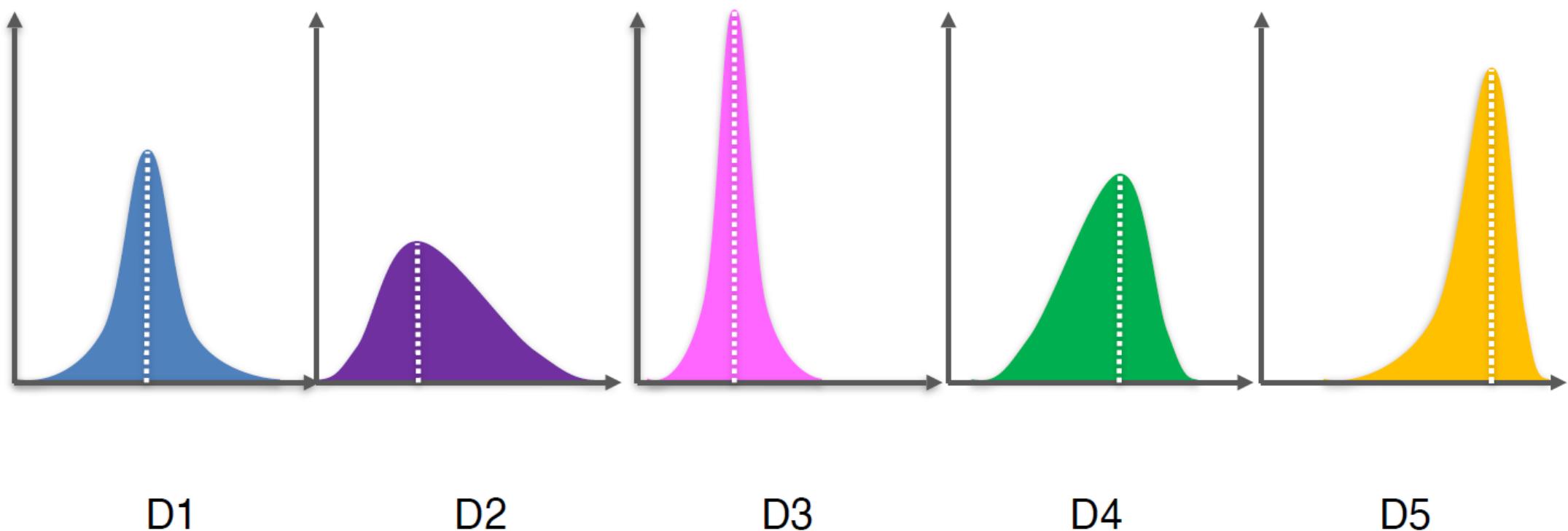


D4

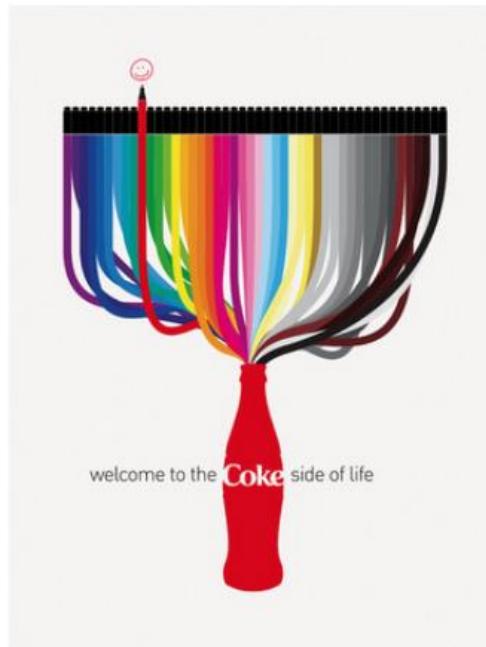


D5

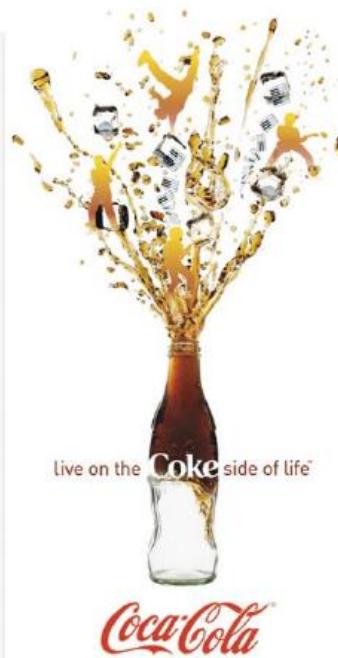
## Upper Confidence Bound



# Upper Confidence Bound



D1



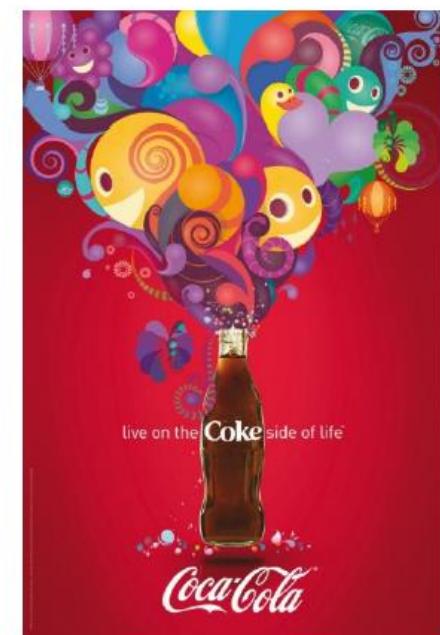
D2



D3

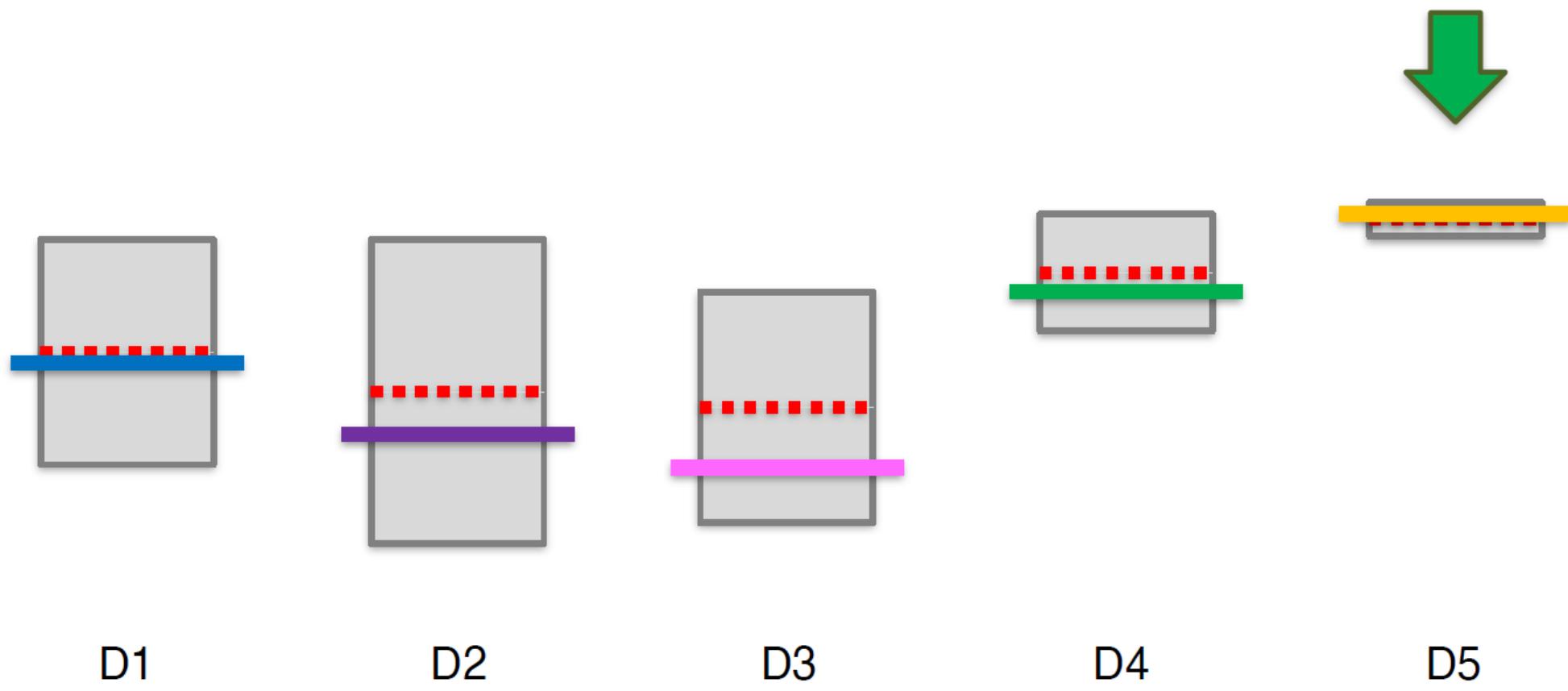


D4



D5

## Upper Confidence Bound



# ¿DÓNDE Y CUÁNDO APLICARLA?

**UCB** es un algoritmo determinista para el aprendizaje por refuerzo que se centra en la exploración y la explotación en función de un límite de confianza que el algoritmo asigna a cada máquina en cada ronda de exploración. (Una ronda es cuando un jugador tira del brazo de una máquina)



## Ventajas y Desventajas

### Algunas Ventajas:

- Sencillo de implementar.
- De un fácil trabajo en forma paralela con otros modelos de regresión y clasificación

### Contras:

- Límites de Arrepentimiento.

El muestreo de Thompson es un algoritmo que se puede usar para encontrar una solución a un problema Multi-Armed Bandit, un término derivado del hecho de que las máquinas tragaperras se denominan de manera informal "one-armed bandits".

## Muestreo de Thompson

Su objetivo es identificar la mejor máquina de la manera más eficiente posible.

## Muestreo de Thompson



D1



D2



D3



D4

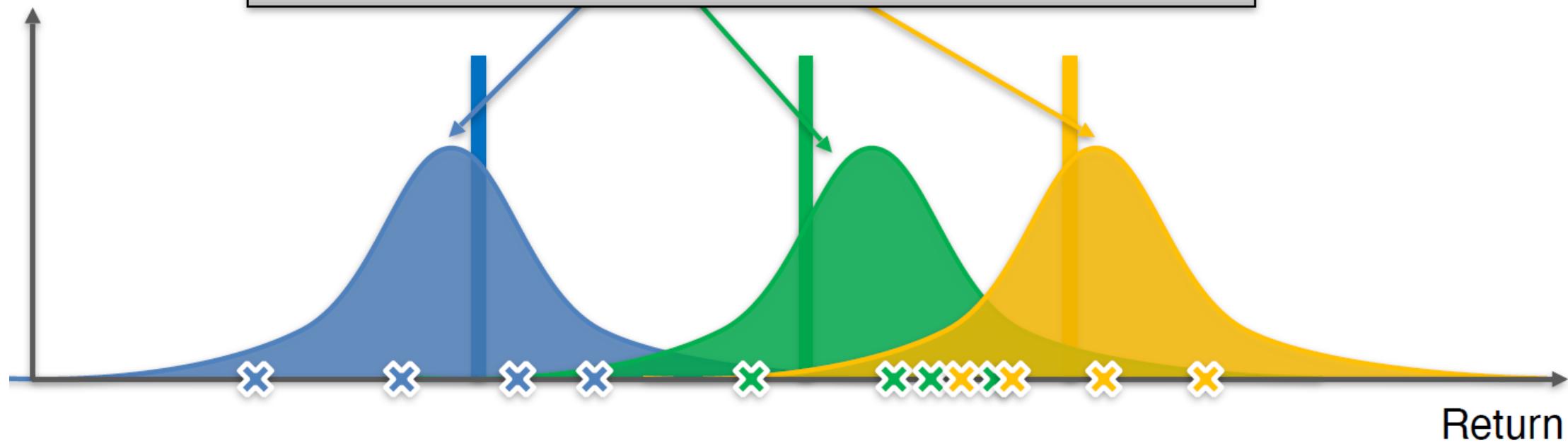


D5

# Muestreo de Thompson

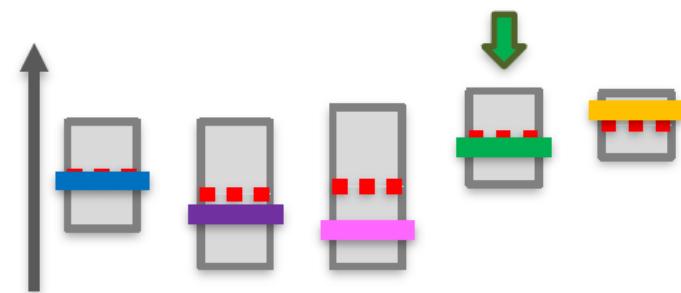
Donde pensamos que estarán los valores  $\mu^*$

i.e. NO queremos averiguar las distribuciones de las máquinas

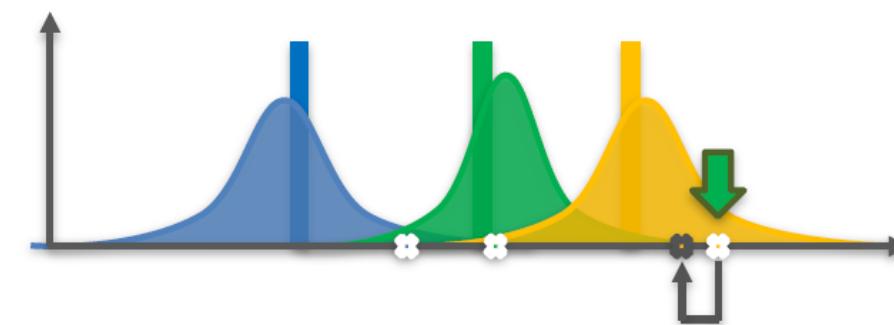


## Muestreo de Thompson

### UCB



### Muestreo Thompson



- Determinista
- Requiere actualizar a cada ronda

- Probabilístico
- Se amolda gracias al feedback a posteriori
- Más evidencias empíricas

# ¿DÓNDE Y CUÁNDO APLICARLA?

- Las actividades recientes del usuario visitante, como los artículos de noticias, donde el usuario ha leído recientemente.
- La información demográfica del usuario visitante, como la información del usuario, género y edad.
- La información contextual del usuario visitante, como la ubicación del usuario y el día de la semana.



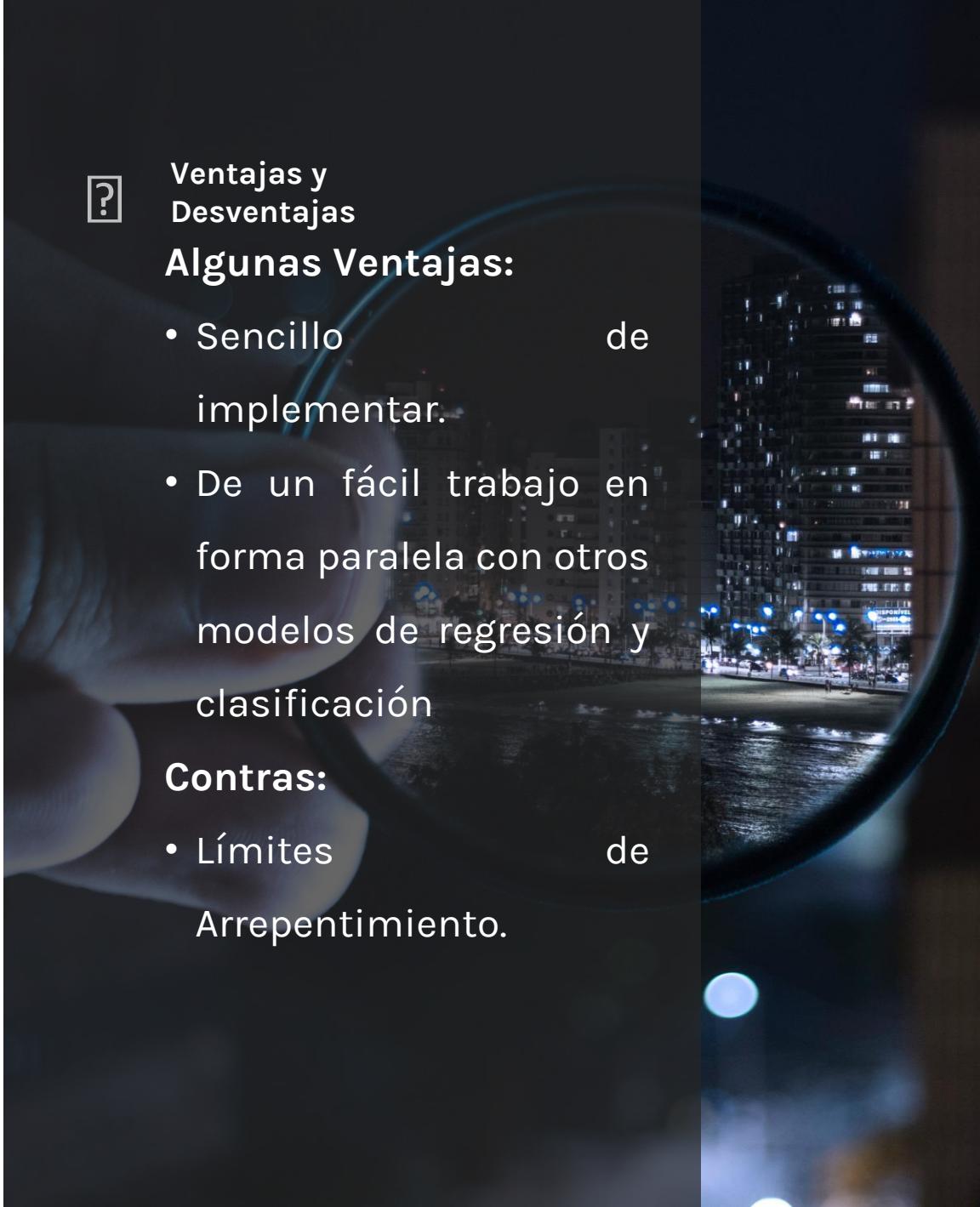
Ventajas y Desventajas

## Algunas Ventajas:

- Sencillo de implementar.
- De un fácil trabajo en forma paralela con otros modelos de regresión y clasificación.

## Contras:

- Límites de Arrepentimiento.





# REDUCCIÓN DE DIMENSIONES

## 01 Análisis de Componentes Principales (ACP)

Es un **método estadístico** que permite simplificar la complejidad de espacios muestrales con **muchas dimensiones** a la vez que conserva su información. Supóngase que existe una muestra con  $n$  individuos cada uno con  $p$  variables ( $X_1, X_2, \dots, X_p$ ).

## 02 Análisis discriminante lineal (LDA)

Es un método de clasificación supervisado de variables cualitativas en el que dos o más grupos son conocidos a priori y nuevas observaciones se clasifican en uno de ellos en función de sus características. Haciendo uso del teorema de Bayes

## 03 Kernel ACP

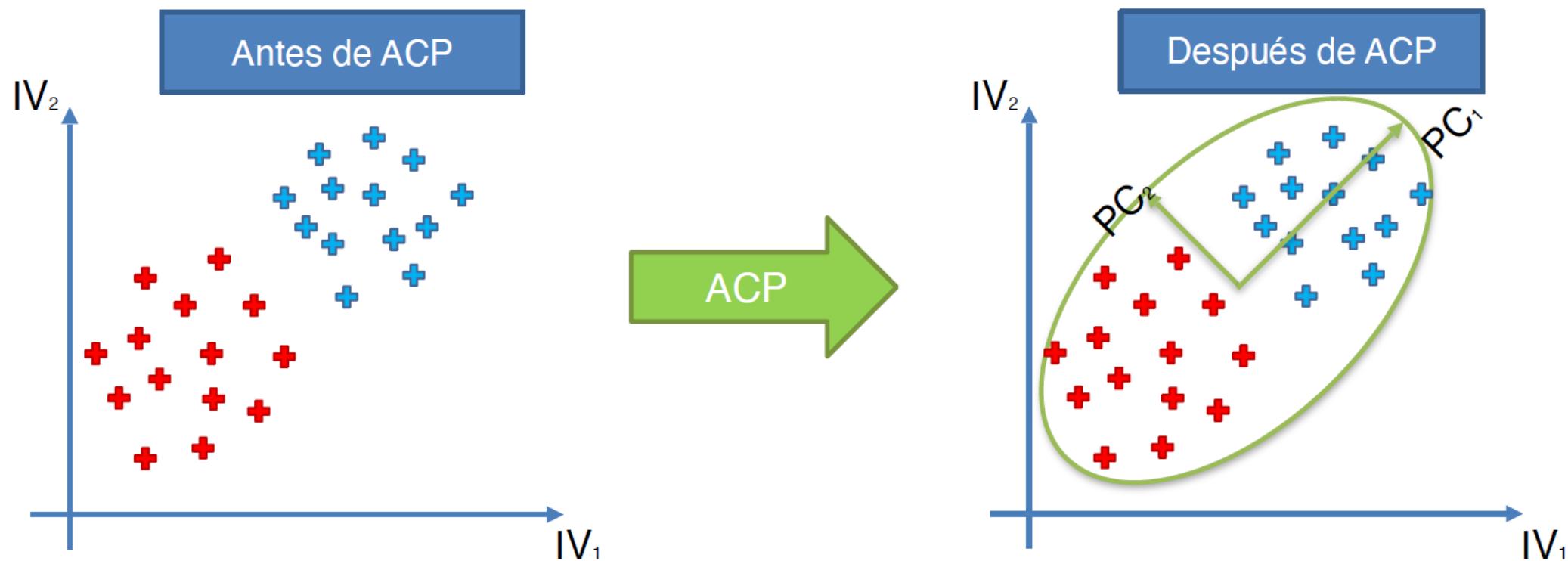
Implica el conglomeramiento espectral con algunos ejemplos ilustrativos. Se pretende estudiar los efectos de aplicar ACP como pre proceso sobre las observaciones que se desean agrupar, para lo cual se hacen experimentos con datos reales

# Análisis de Componentes Principales (ACP)

Selección de Características	Extracción de Características
Eliminación hacia atrás	ACP
Selección directa	LDA
Eliminación bidimensional	ACP con Kernel
Comparación de Scores	

De las  $m$  variables independientes del dataset, ACP extrae las  $p \leq m$  nuevas variables independientes que explicar la mayor parte de la varianza del dataset, **sin importar el valor de la variable dependiente.**

# Análisis de Componentes Principales (ACP)



$PC_1$  y  $PC_2$  son las direcciones de máxima varianza

# Análisis de Componentes Principales (ACP)

PASO 1: Aplicar escalado de variables a la matriz de características X, formada por m variables independientes.

PASO 2: Calcular la matriz de covarianzas de las m variables independientes de X.



PASO 3: Calcular los valores y vectores propios de la matriz de covarianzas.



PASO 4: Elegir un porcentaje P de varianza explicada y elegir los  $p \leq m$  valores propios más grandes tales que:

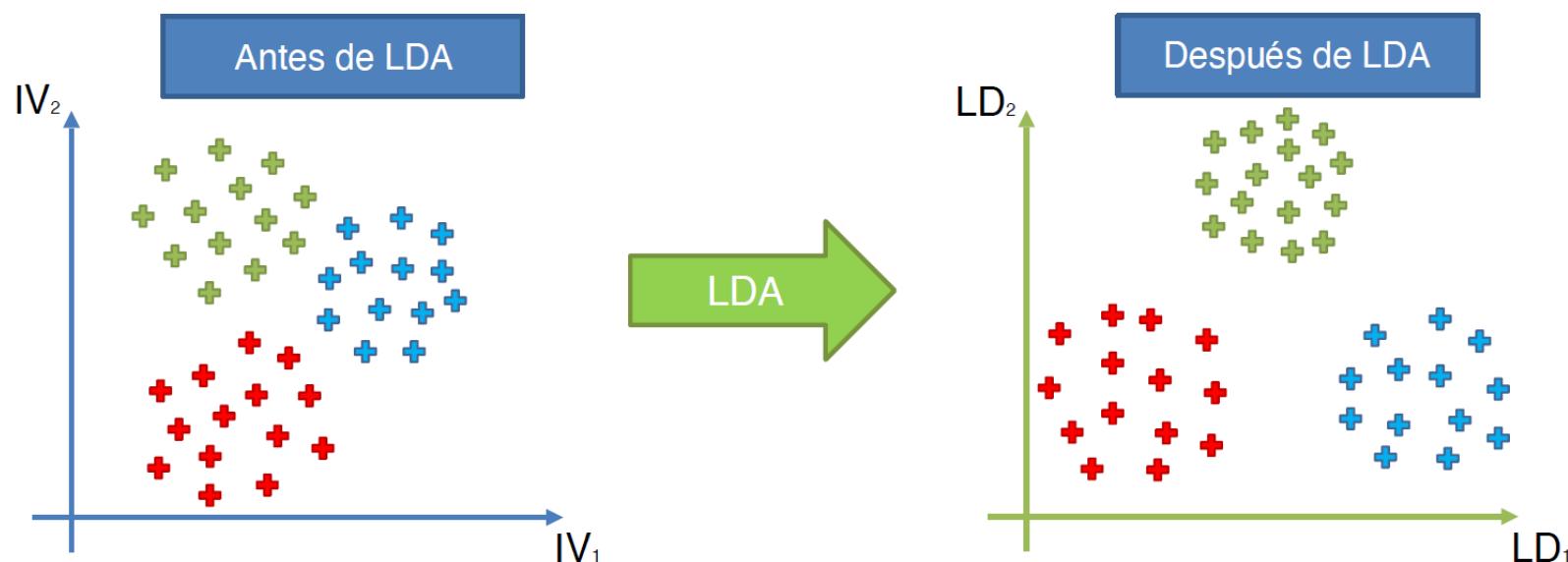


$$\frac{\sum_{j=1}^p \lambda_j}{\sum_{i=1}^m \lambda_i} \geq P$$

PASO 5: Los p vectores propios asociados a estos p valores más grandes son las componentes principales.

El espacio m-dimensional del dataset original se proyecta al nuevo subespacio p-dimensional de características, aplicando la matriz de proyecciones (que tiene los p vectores propios por columnas).

## LDA en dos palabras



$LD_1$  y  $LD_2$  son las direcciones de máxima separación de clases

De las  $n$  variables independientes del dataset, LDA extrae las  $p \leq n$  nuevas variables independientes que separan la mayoría de clases de la variable dependiente.



## XGBoost

Es una biblioteca de software de código abierto que proporciona un marco de aumento de gradiente para C++, Java, Python, R, Julia,

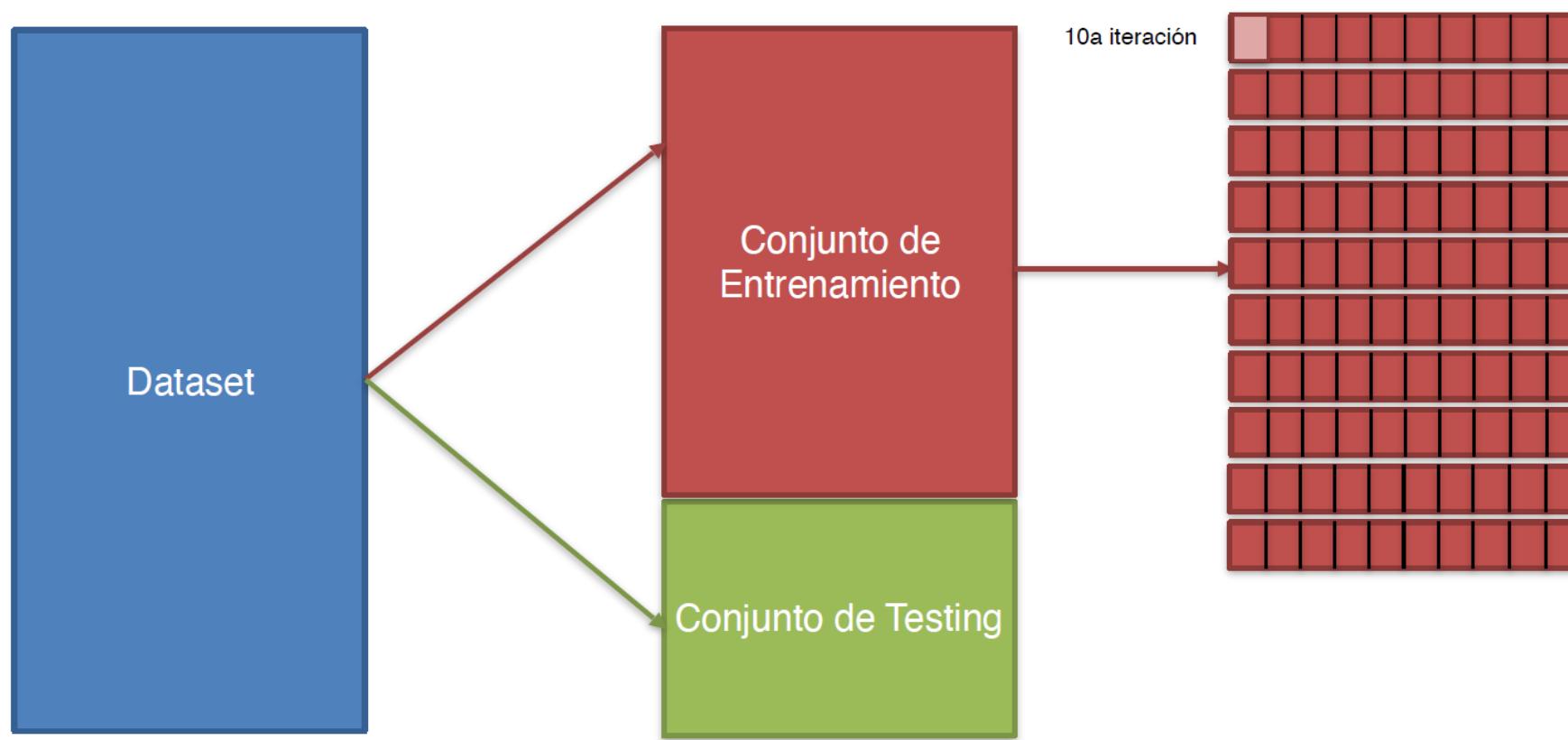
Perl y Scala



## K-fold Cross Validation

Es utilizado para proporcionar una evaluación relevante de la eficacia de nuestro modelo

# k-Fold Cross Validation



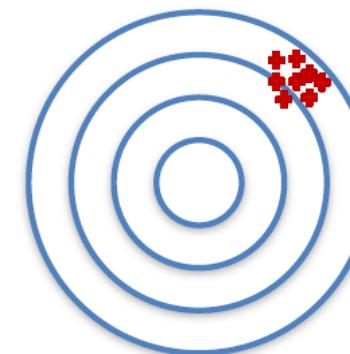
## Compensar el sesgo-varianza

**Sesgo bajo:** cuando el modelo elabora predicciones cercanas a los datos reales.

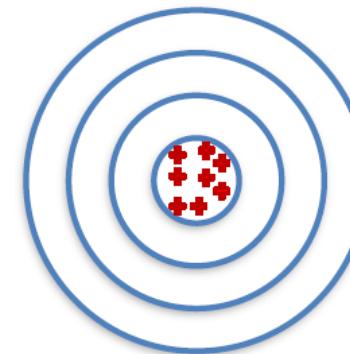
**Sesgo alto:** cuando el modelo elabora predicciones alejadas de los datos reales.

**Varianza baja:** cuando ejecutamos el modelo varias veces y las predicciones no varían demasiado.

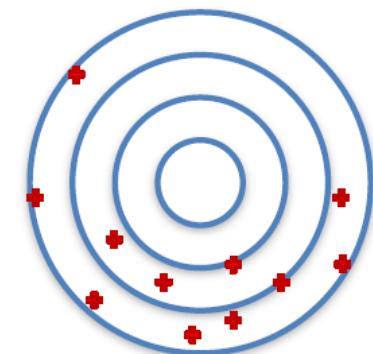
**Varianza elevada:** cuando ejecutamos el modelo varias veces y las predicciones varían demasiado.



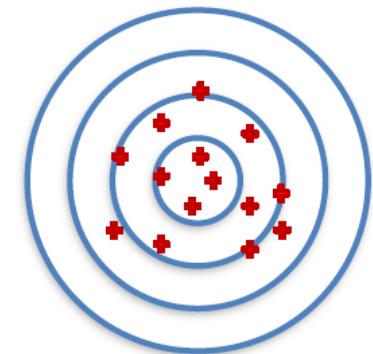
Sesgo alto Varianza baja



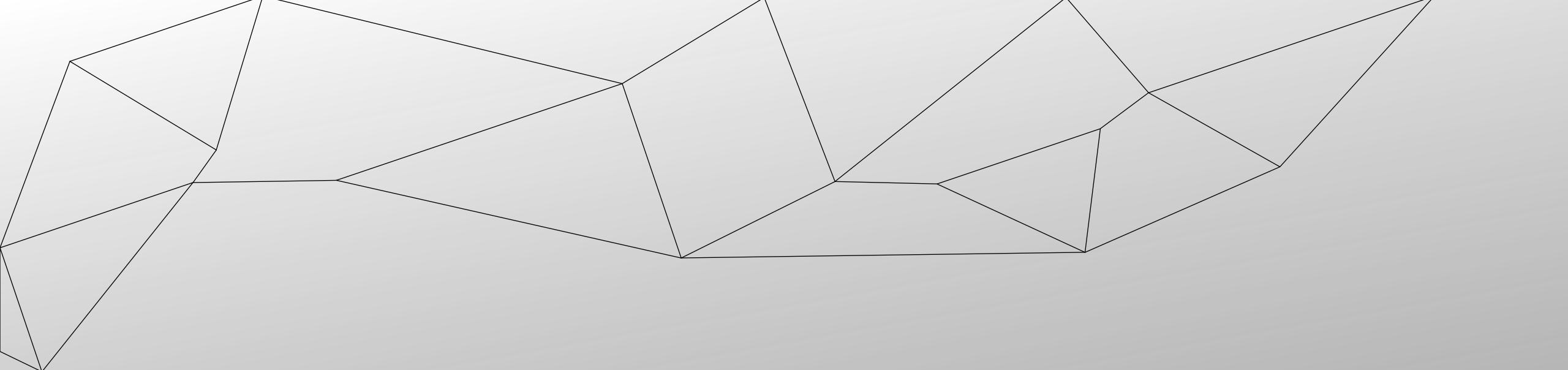
Sesgo bajo Varianza baja



Sesgo alto Varianza alta



Sesgo bajo Varianza alta



# Terminamos los Modelos

Es hora de los resultados...