

Curso Aprendizaje Automático (ML) con Python

Nelson López Centeno

Introducción al Aprendizaje Automático (ML)

- ☐ ¿Qué es el Aprendizaje Automático (ML)?
- ☐ Flujo de trabajo
- ☐ *Overfitting y underfitting*
- ☐ Tareas del Aprendizaje Automático (ML)
- ☐ Métodos de aprendizaje
- ☐ Algoritmos
- ☐ Métricas de evaluación de modelos



Ejercicios

- ☐ Regresión lineal de datos sintéticos
- ☐ Clasificación con el conjunto de datos Iris
- ☐ Clustering con el conjunto de datos Iris



¿Qué es el Aprendizaje Automático (ML)?

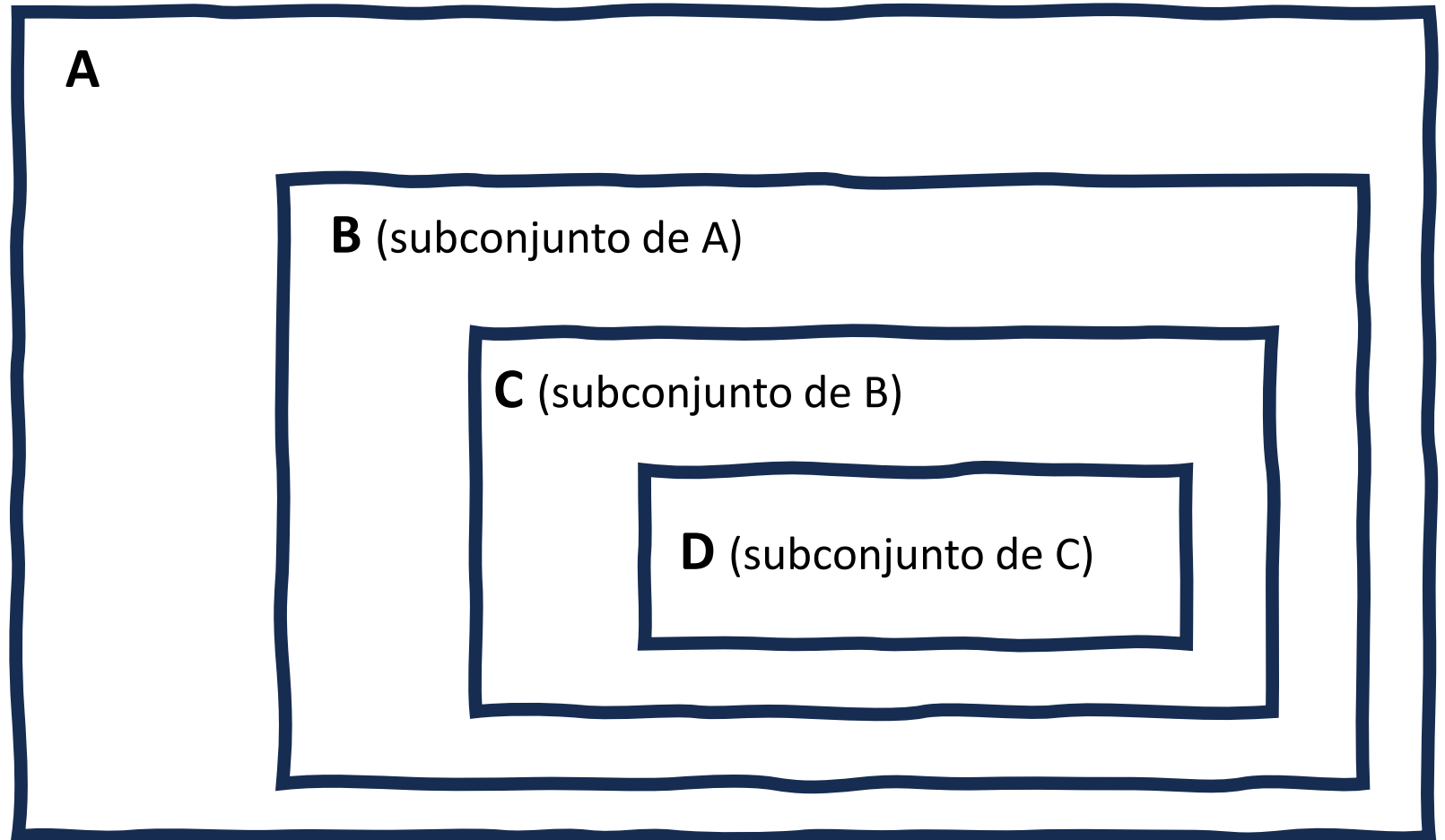
Ubica los términos de la izquierda en el diagrama de la derecha

Aprendizaje Automático (ML)

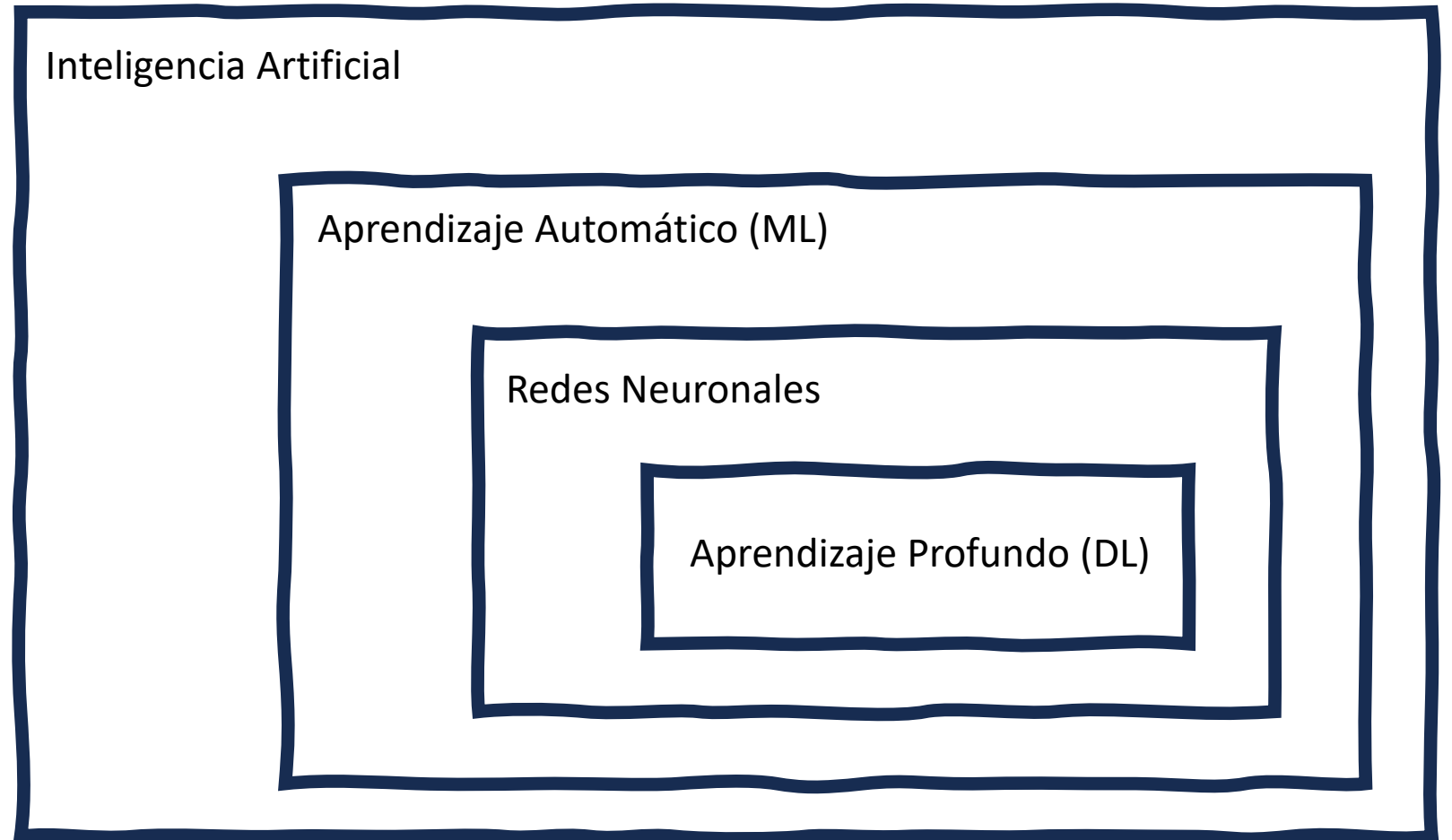
Aprendizaje Profundo (DL)

Inteligencia Artificial

Redes Neuronales



¿Qué es el Aprendizaje Automático (ML)?



¿Qué es el Aprendizaje Automático (ML)?

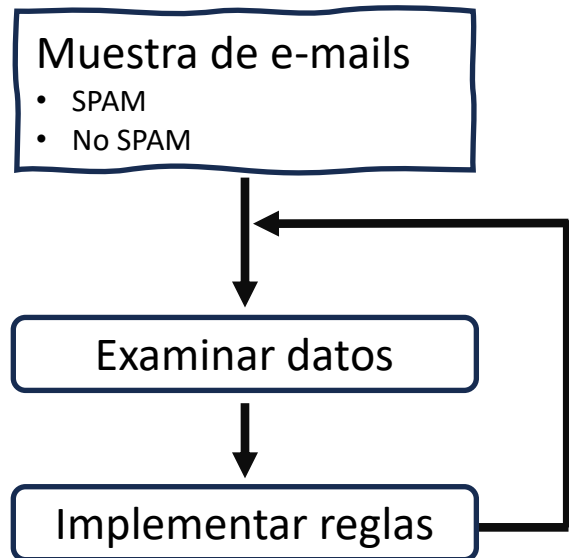
“El aprendizaje automático es el ámbito de estudio que aporta a los **ordenadores** la capacidad de **aprender sin estar explícitamente programados**”

Arthur Samuel, 1959

¿Qué es el Aprendizaje Automático (ML)?

Implementar un filtro de SPAM sin ML y con ML

Sin Aprendizaje Automático (ML)



Nuevos e-mails

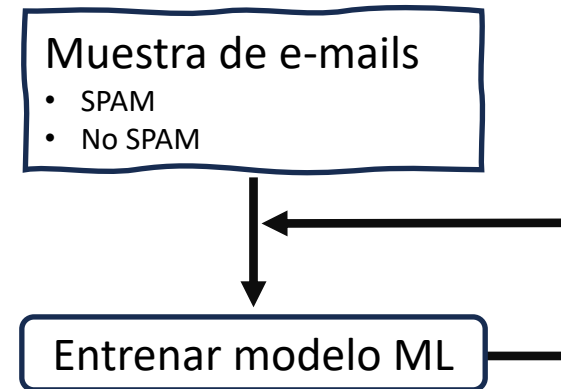


Aplicar reglas

SPAM

No SPAM

Con Aprendizaje Automático (ML)



Nuevos e-mails

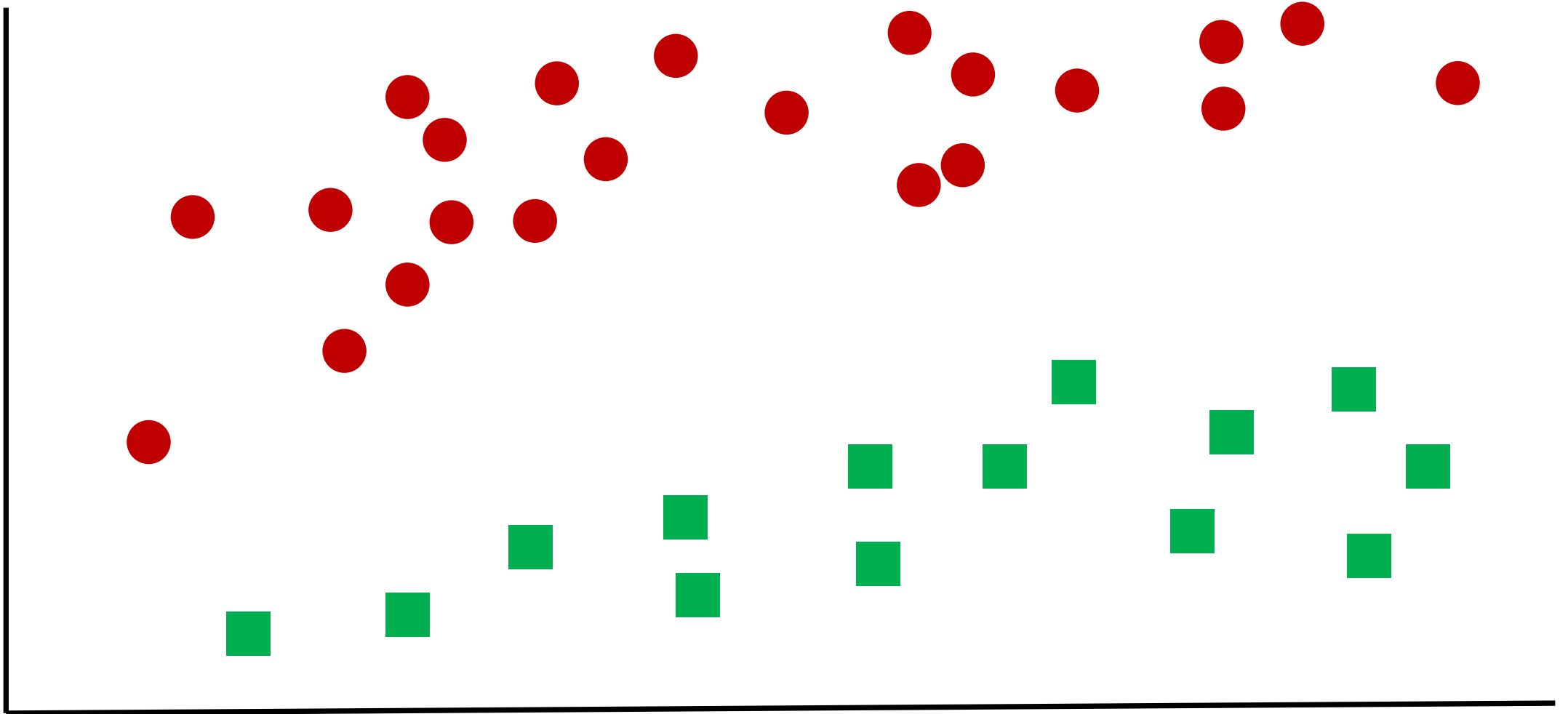


Aplicar modelo ML

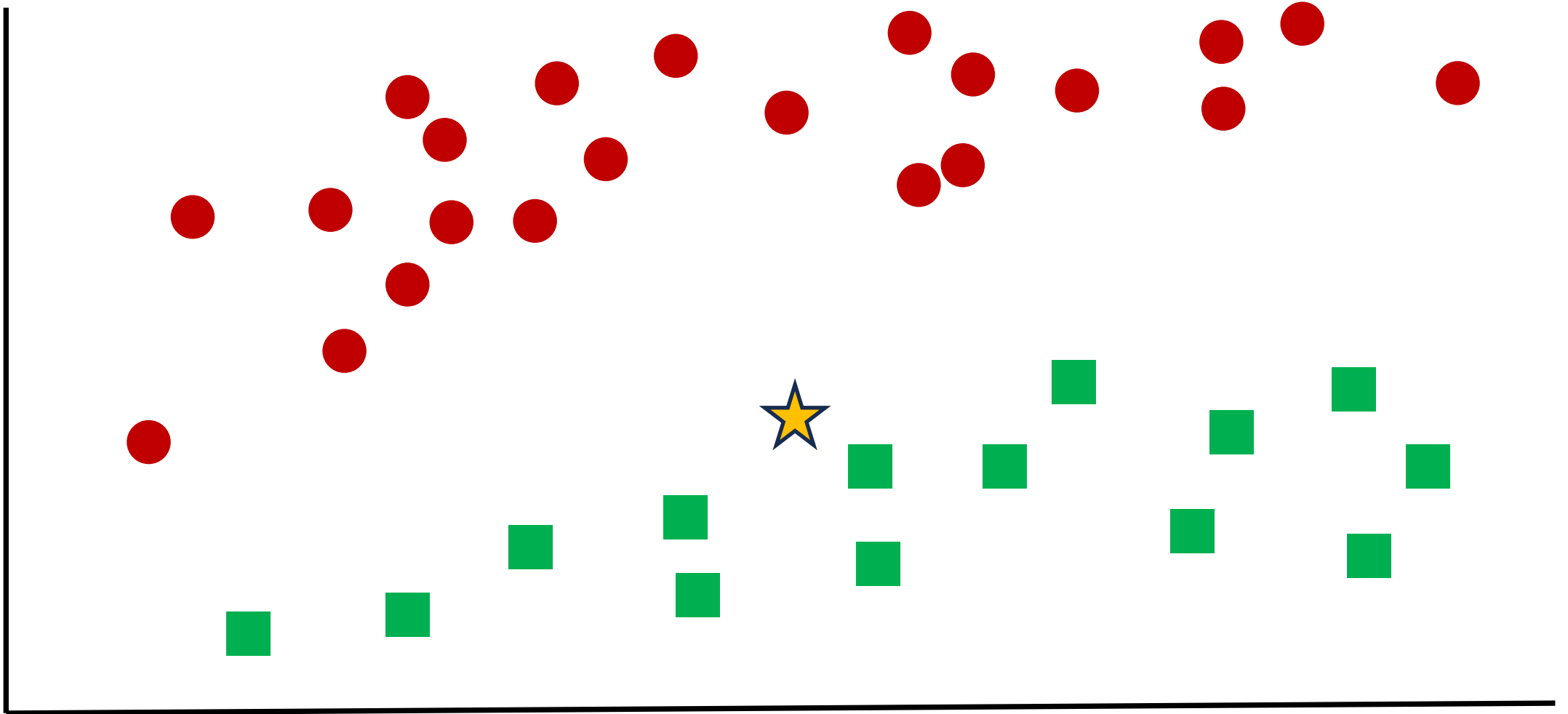
SPAM

No SPAM

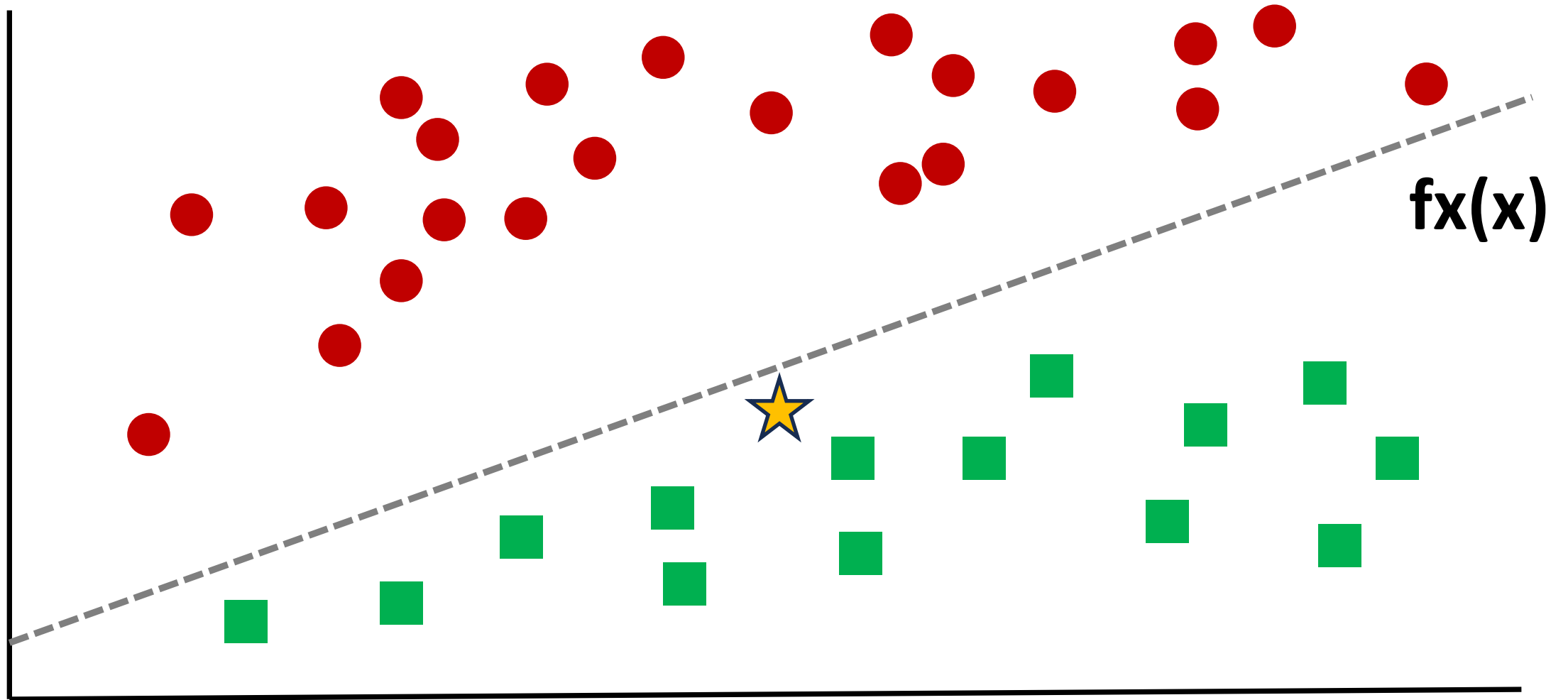
¿Qué es el Aprendizaje Automático (ML)?



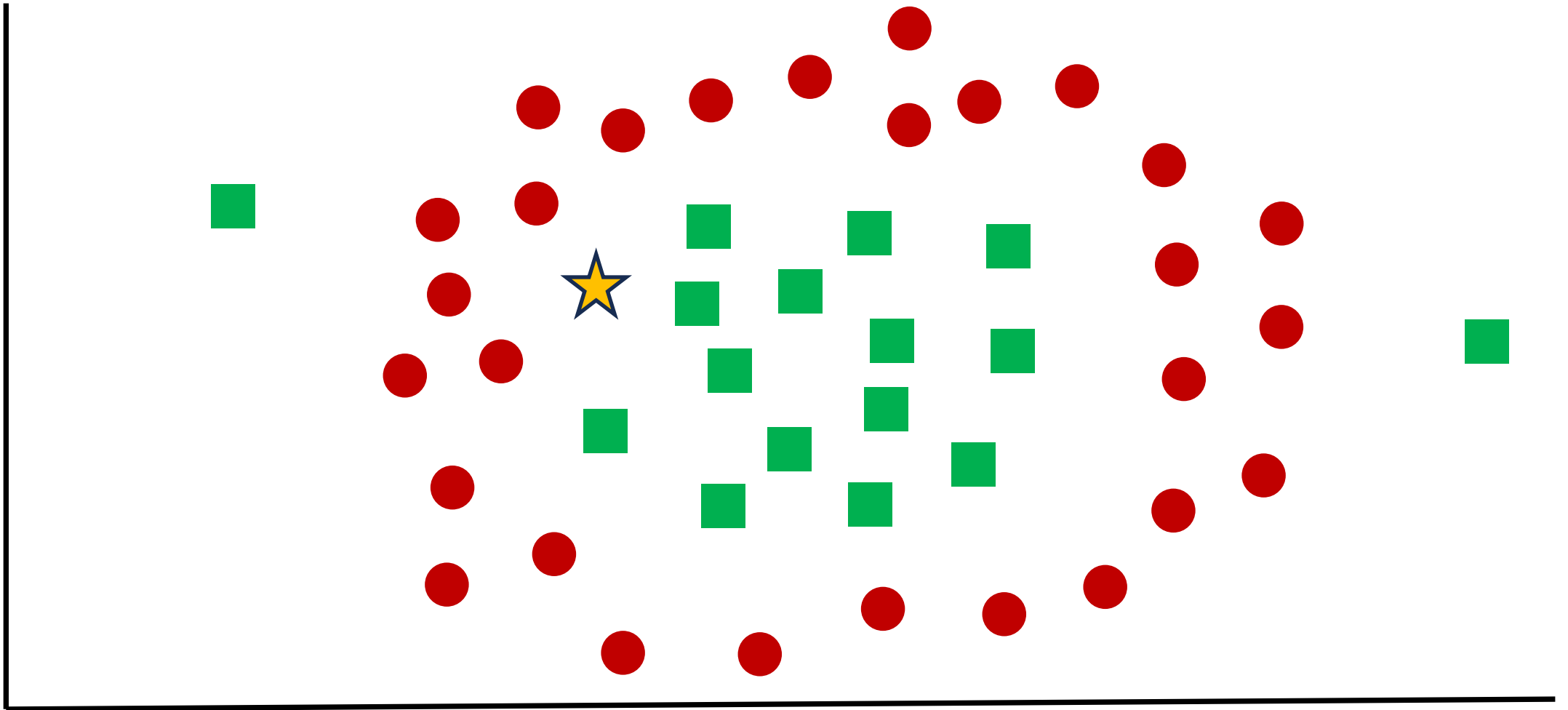
¿Qué es el Aprendizaje Automático (ML)?



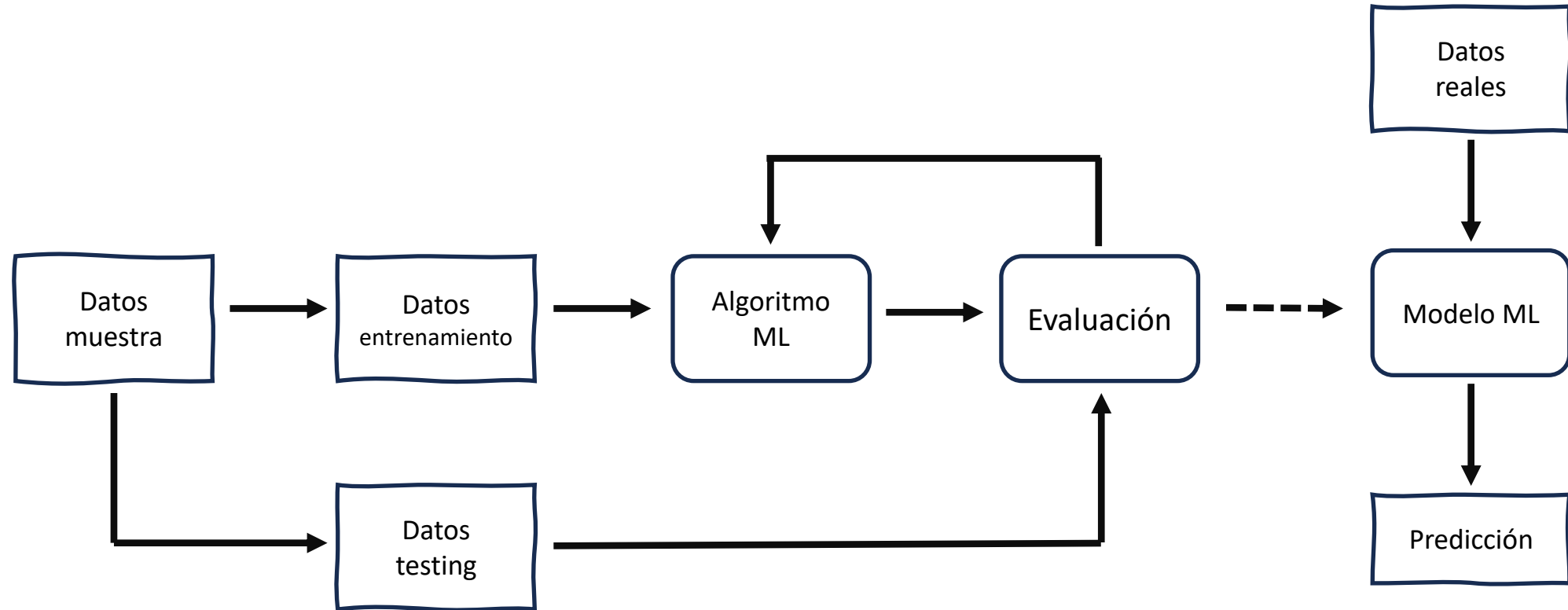
¿Qué es el Aprendizaje Automático (ML)?



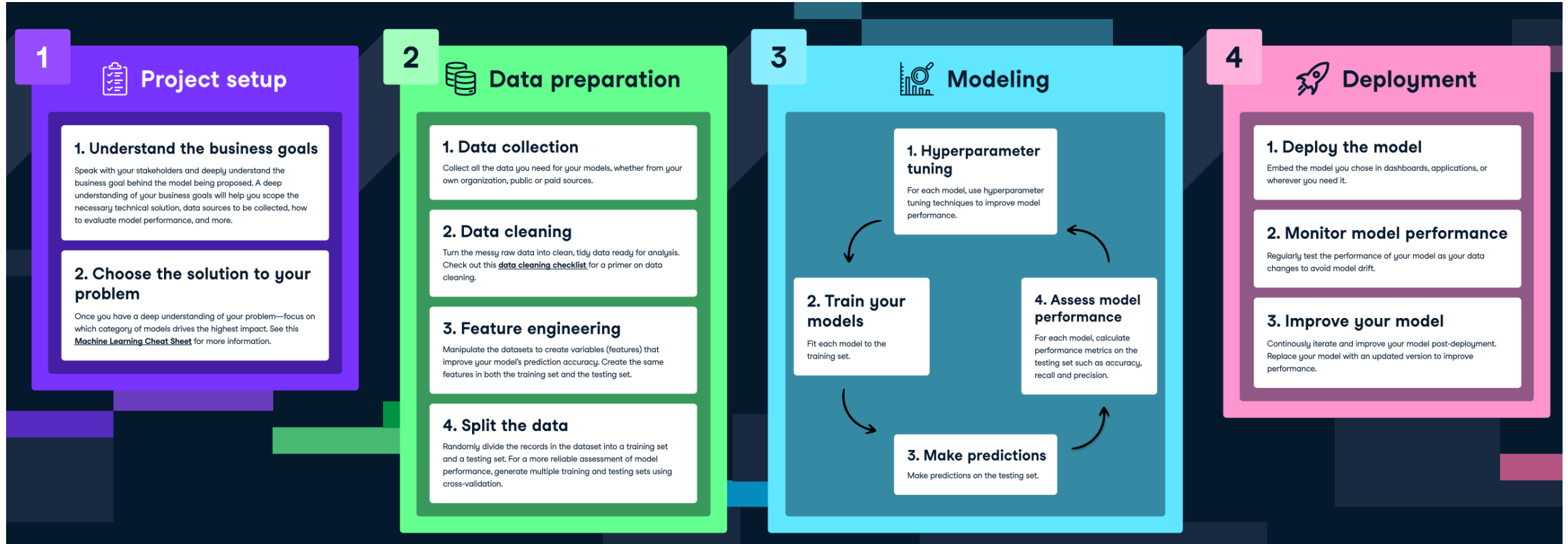
¿Qué es el Aprendizaje Automático (ML)?



Flujo de trabajo



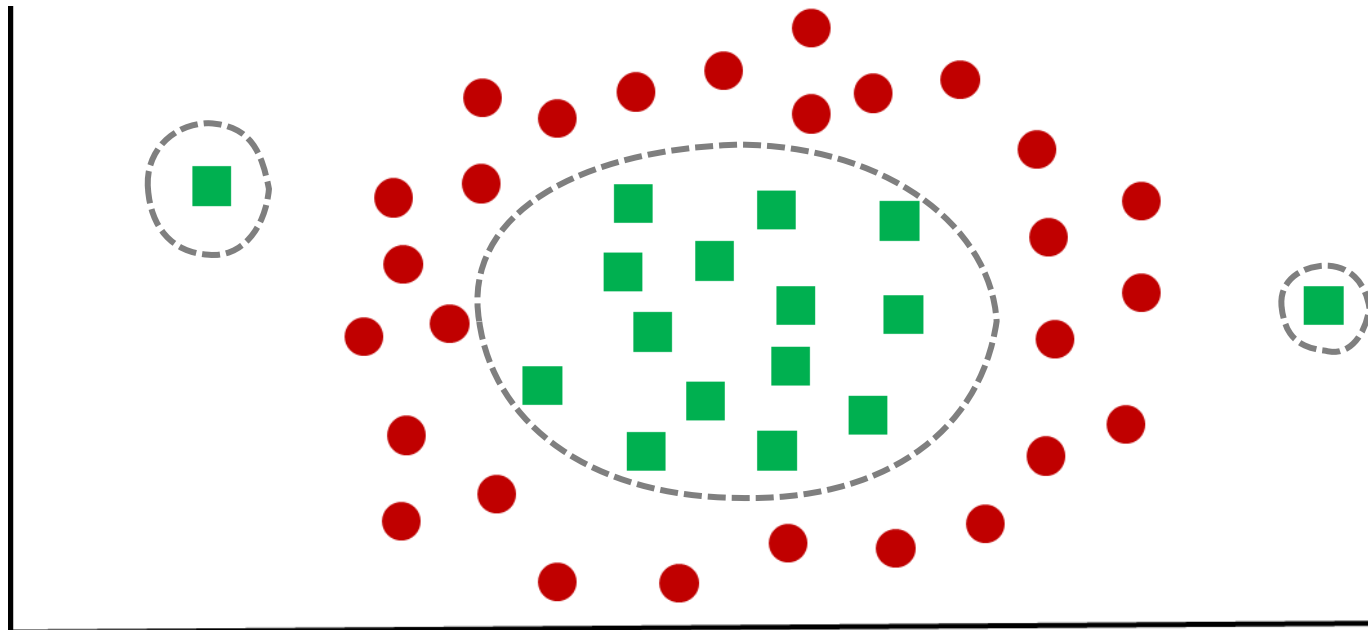
Flujo de trabajo



<https://www.datacamp.com/blog/a-beginner-s-guide-to-the-machine-learning-workflow>

Overfitting (sobreajuste)

El modelo ML **predice muy bien** con los **datos de entrenamiento** mientras que **predice mal** con los **datos de testing** o los **datos reales**.

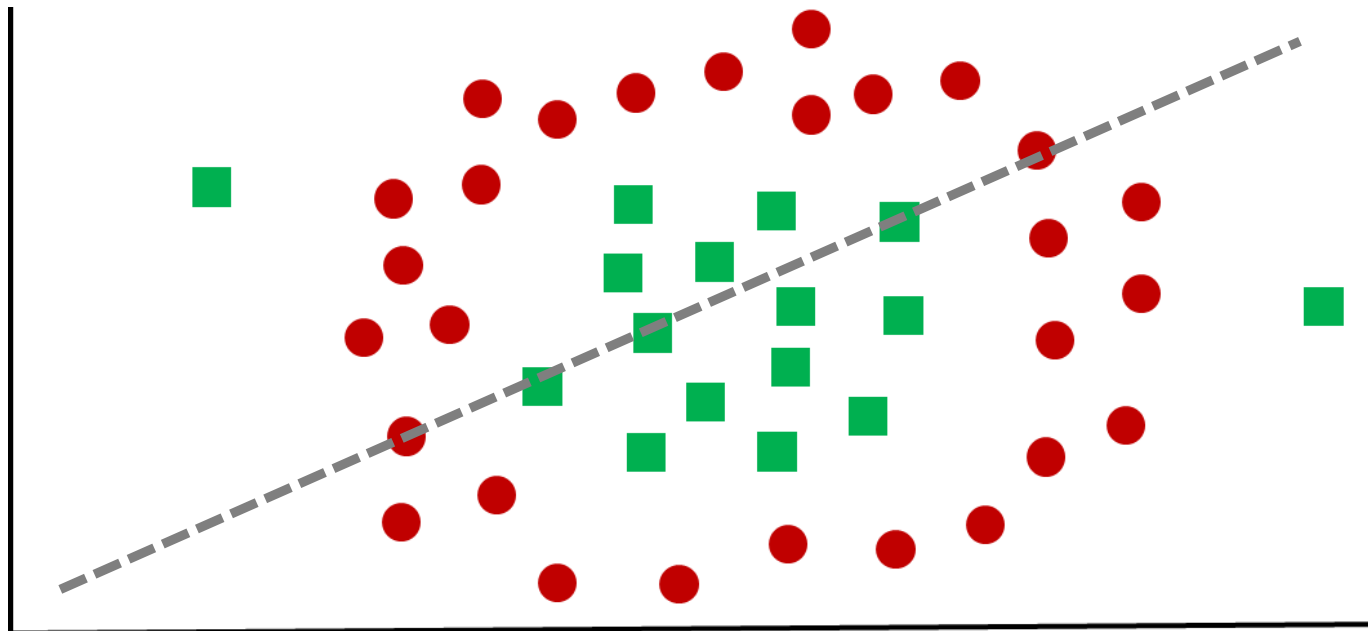


Posibles soluciones

- Algoritmo más sencillo
- Regularización de los parámetros del modelo
- Reducir los atributos en los datos de entrenamiento
- Aumentar la cantidad de los datos de entrenamiento
- Reducir el ruido en los datos de entrenamiento

Underfitting

El modelo ML no es capaz de capturar la relación entre las variables por lo que **predice mal** tanto los **datos de entrenamiento** como los **datos reales**.



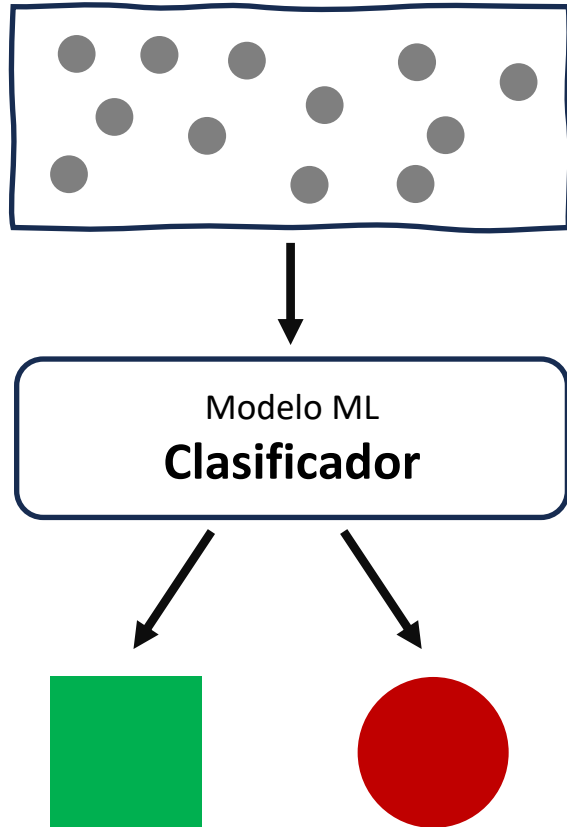
Posibles soluciones

- Algoritmo más complejo
- Reducir la regularización
- Mejorar los atributos en los datos de entrenamiento

Tareas del Aprendizaje Automático (ML)

- ☐ Clasificación
- ☐ Regresión
- ☐ *Clustering*
- ☐ Reducción de dimensionalidad

Clasificación



Asignar clases (categorías) a una instancia

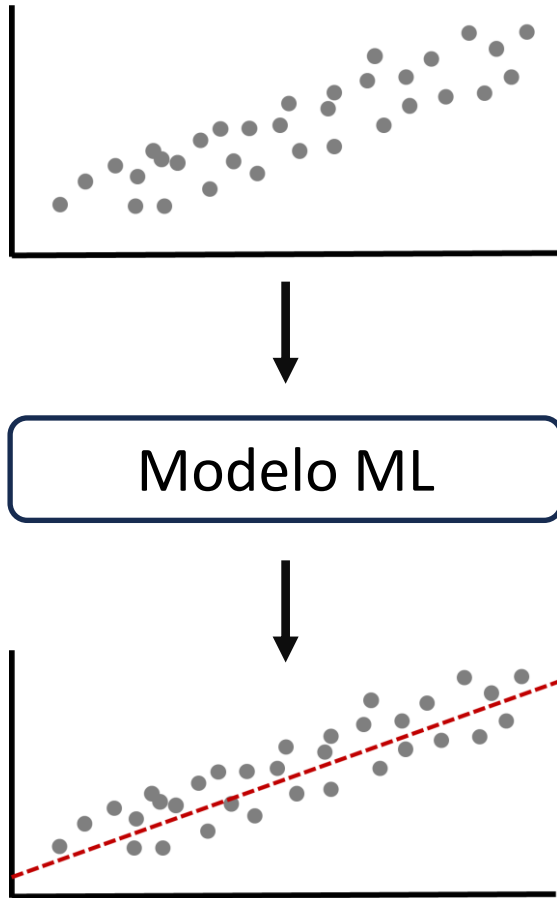
Tipos

- Clasificación **Binaria**: Escoger una de dos clases
- Clasificación **Multi-clase**: Escoger una de varias clases
- Clasificación **Multi-etiqueta**: Una instancia puede pertenecer a varias clases simultáneamente

Algoritmos

- Regresión logística
- nearest-neighbor, k-nearest-neighbor
- Árboles de decisión
- *Random forests*
- Support Vector Machine (SVM)

Regresión



Predecir un valor continuo (numérico)

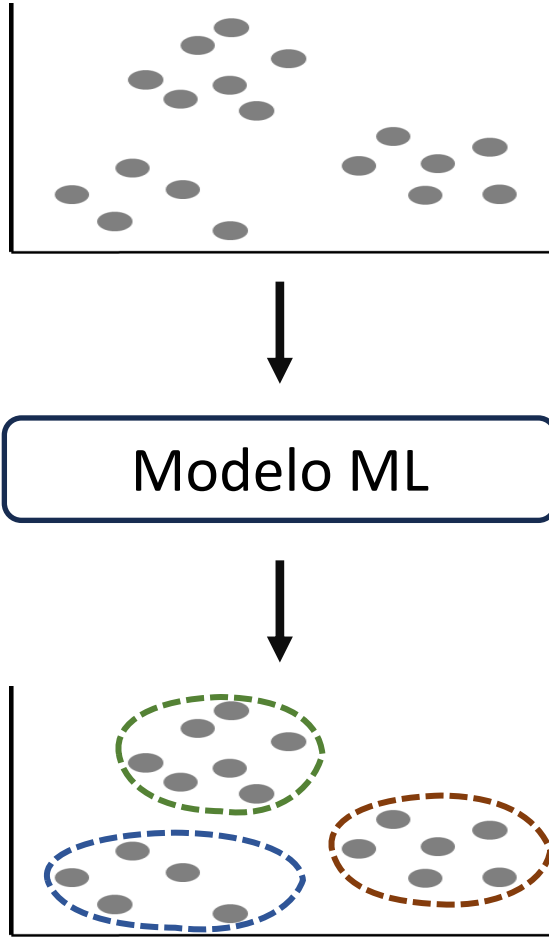
Ejemplos de predicciones

- Precio de una vivienda a partir de atributos como el número de habitaciones, la ubicación o el tamaño
- Precios futuros de las acciones basándose en datos históricos y en las tendencias actuales del mercado
- Ventas de un producto basándose en los presupuestos publicitarios

Algoritmos

- Regresión lineal
- Lasso
- Gradient boosting
- Árbol de decisión

Clustering



Agrupar elementos similares

Ejemplos

- Comprender los segmentos de clientes de hoteles en función de los hábitos y las características de las elecciones hoteleras
- Identificar segmentos de clientes y datos demográficos para ayudar a crear campañas publicitarias específicas
- Categorizar el inventario en función de las métricas de fabricación

Algoritmos

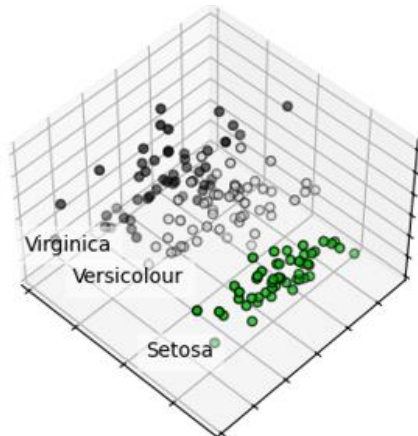
- k-means
- Gaussian mixture

Reducción de dimensionalidad

Sepal length	Sepal width	Petal Length	Petal Width
5,1	3,5	1,4	0,2
4,9	3,0	1,4	0,2
4,7	3,2	1,3	0,2



Modelo ML



Simplificar los datos sin perder la información relevante para el caso de uso

Ejemplo

- Reducir un conjunto de datos de 10 dimensiones a un conjunto de datos de 2 dimensiones para facilitar su visualización en forma de gráfico de dispersión, pero conservando las agrupaciones naturales entre las instancias de entrada

Algoritmos

- Análisis de componentes principales (PCA)
- Análisis de componentes independientes (ICA)

Métodos de aprendizaje

- ☐ Supervisado
- ☐ No supervisado
- ☐ Semisupervisado
- ☐ Aprendizaje por refuerzo (RL)

Métodos de aprendizaje

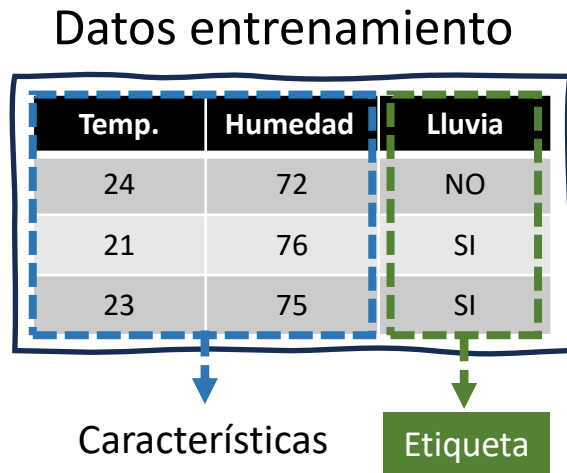
Supervisado

Datos entrenamiento

Temp.	Humedad	Lluvia
24	72	NO
21	76	SI
23	75	SI

Características

Etiqueta



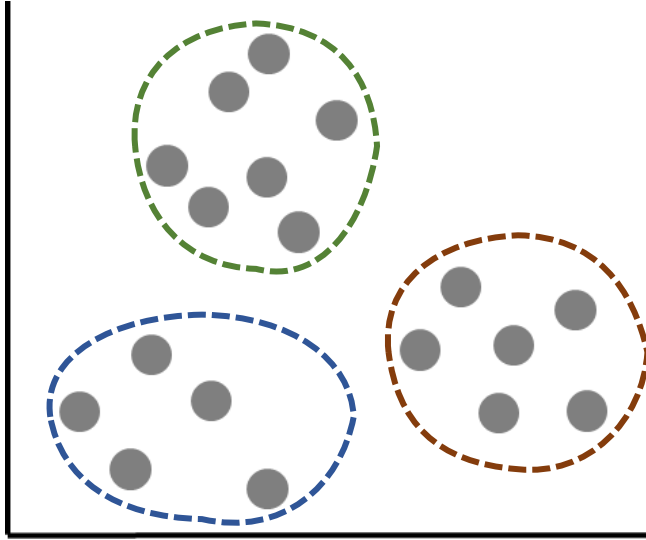
El aprendizaje supervisado utiliza datos de entrenamiento que incluyen la respuesta deseada y a dicha respuesta se le llama **etiqueta**.

Tipos de tareas

- Clasificación
- Regresión

Métodos de aprendizaje

No supervisado



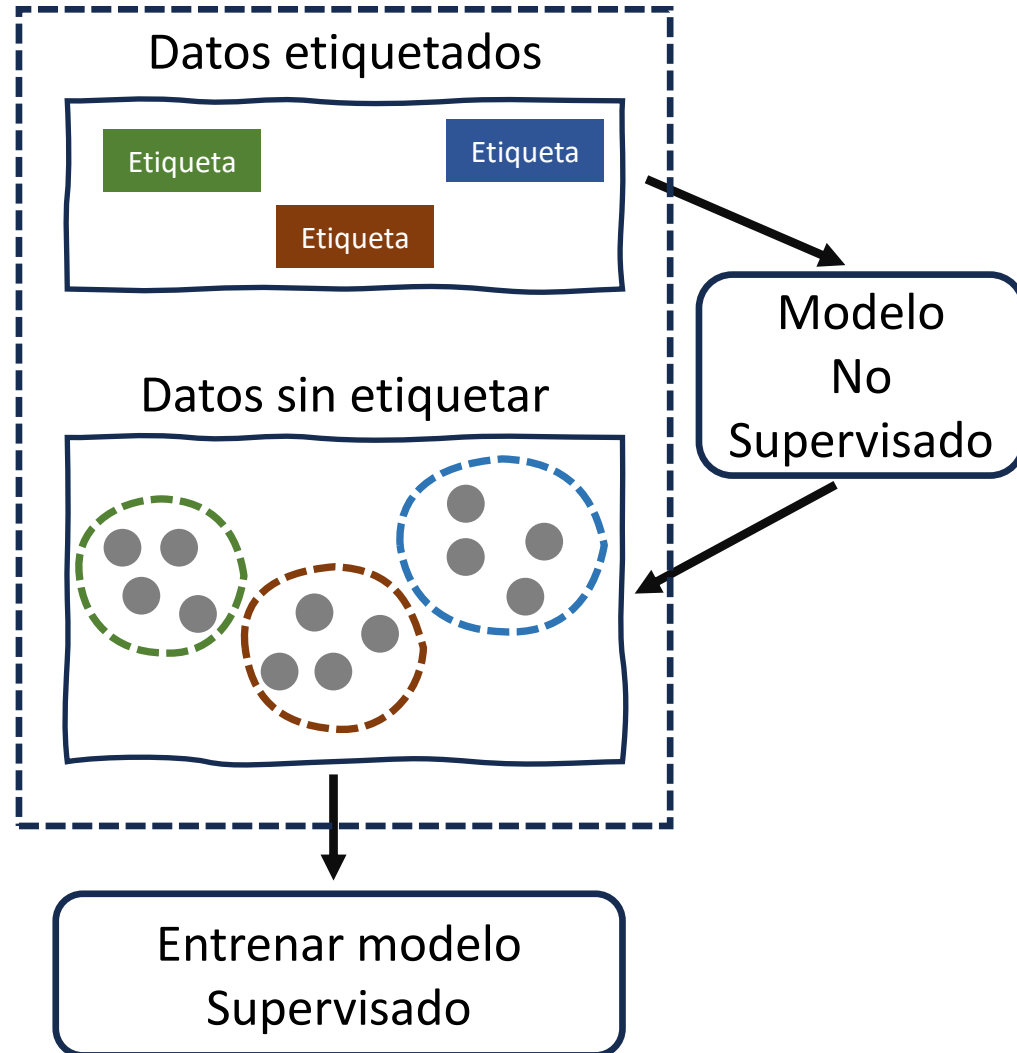
El aprendizaje no supervisado analiza y **agrupa** datos **sin etiquetar**.

Tipos de tareas

- *Clustering*
- Reducción de dimensiones

Métodos de aprendizaje

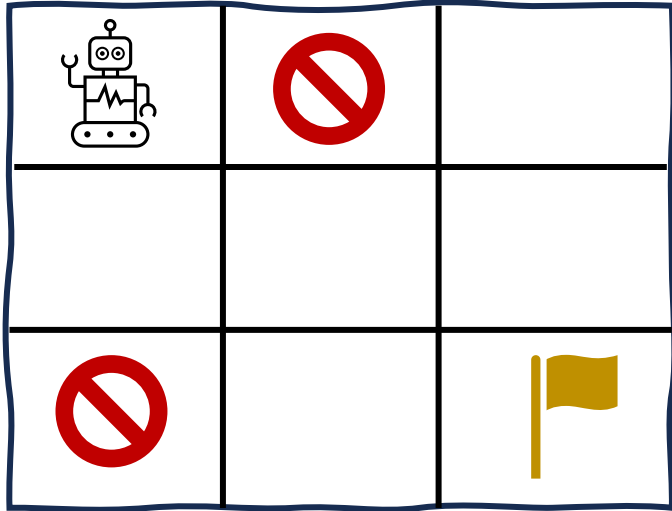
Semisupervisado



El aprendizaje semisupervisado se utiliza cuando se tiene **poca cantidad** de datos **etiquetados** y una **gran cantidad** de datos **sin etiquetar**.

Por lo general se utiliza una combinación de un modelo no supervisado que sea capaz de asignar las etiquetas, y los datos ya etiquetados se utilizan para entrenar un modelo supervisado.

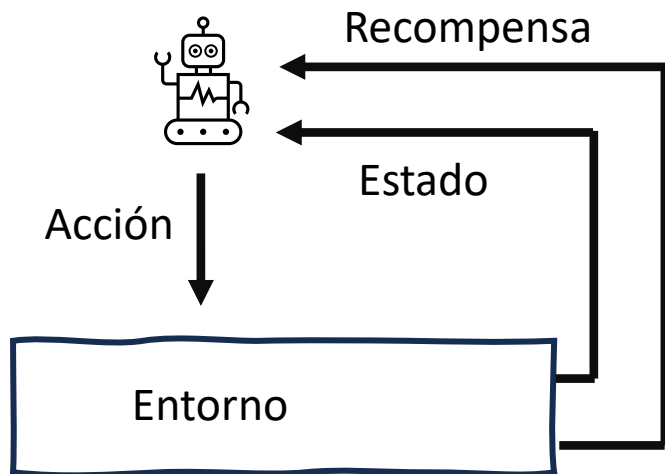
Aprendizaje por refuerzo (RL)



Un **agente** explora un entorno para ejecutar las **acciones** posibles en el **estado** actual y es **recompensado** o **castigado** (recompensa negativa).

El agente va **aprendiendo** cuales son las acciones que generan recompensa para cada estado.

El agente debe tener margen para **explorar** nuevas acciones desde estados por los que ya haya pasado.



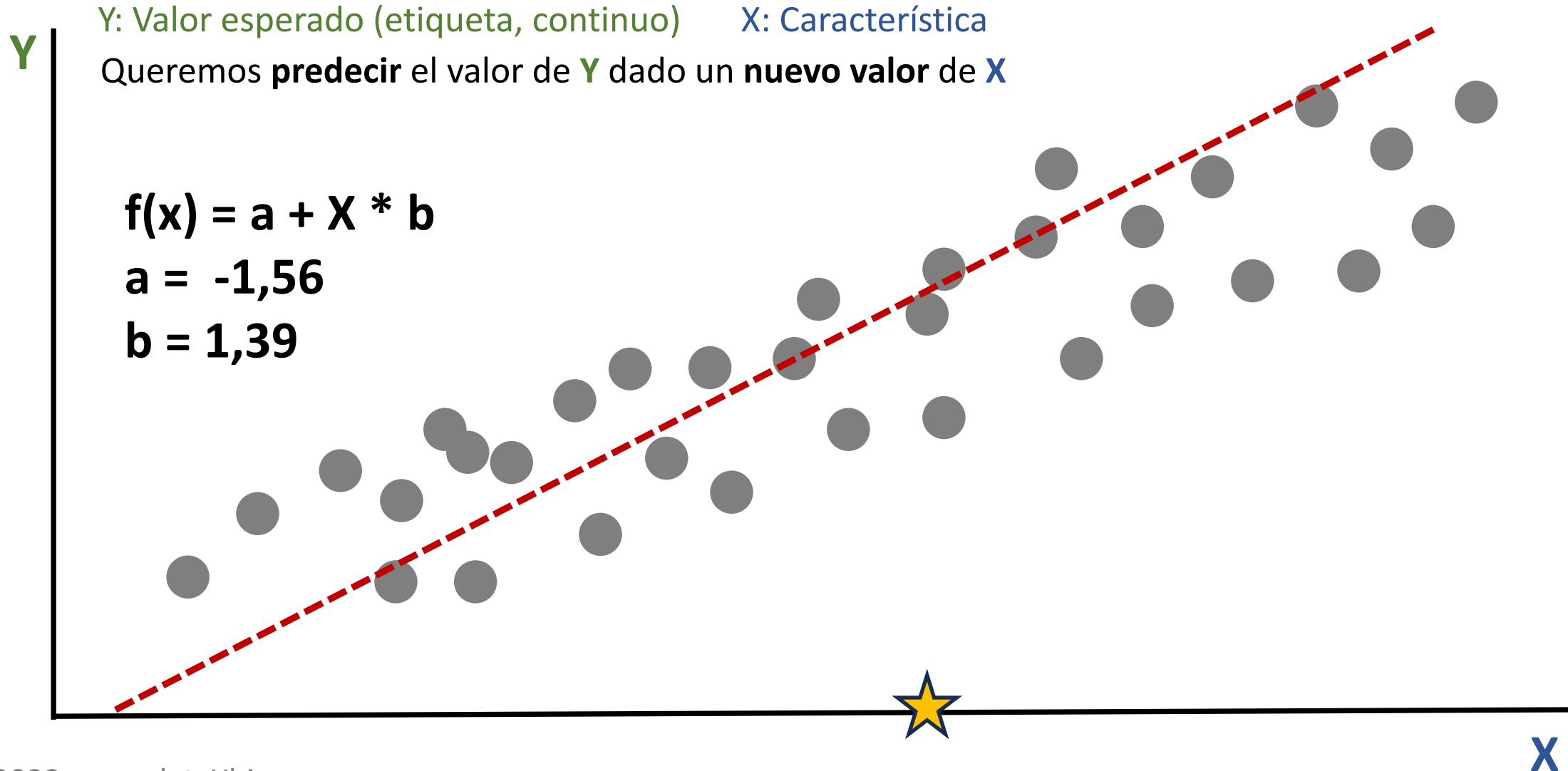
Algoritmos

- ☐ Regresión lineal
- ☐ nearest-neighbor
- ☐ k-nearest-neighbor
- ☐ Árbol de decisión
- ☐ Random forests
- ☐ k-means

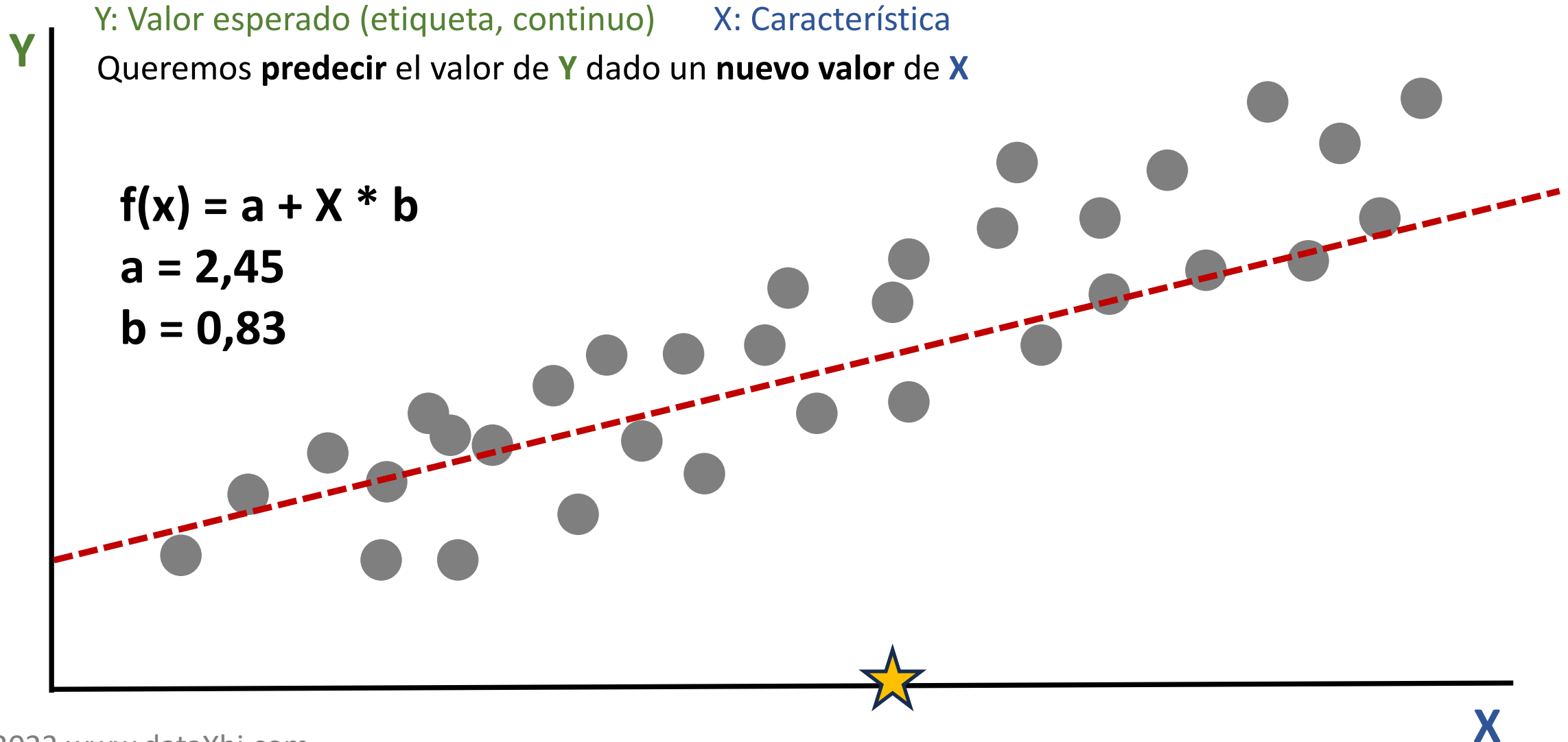
Regresión lineal



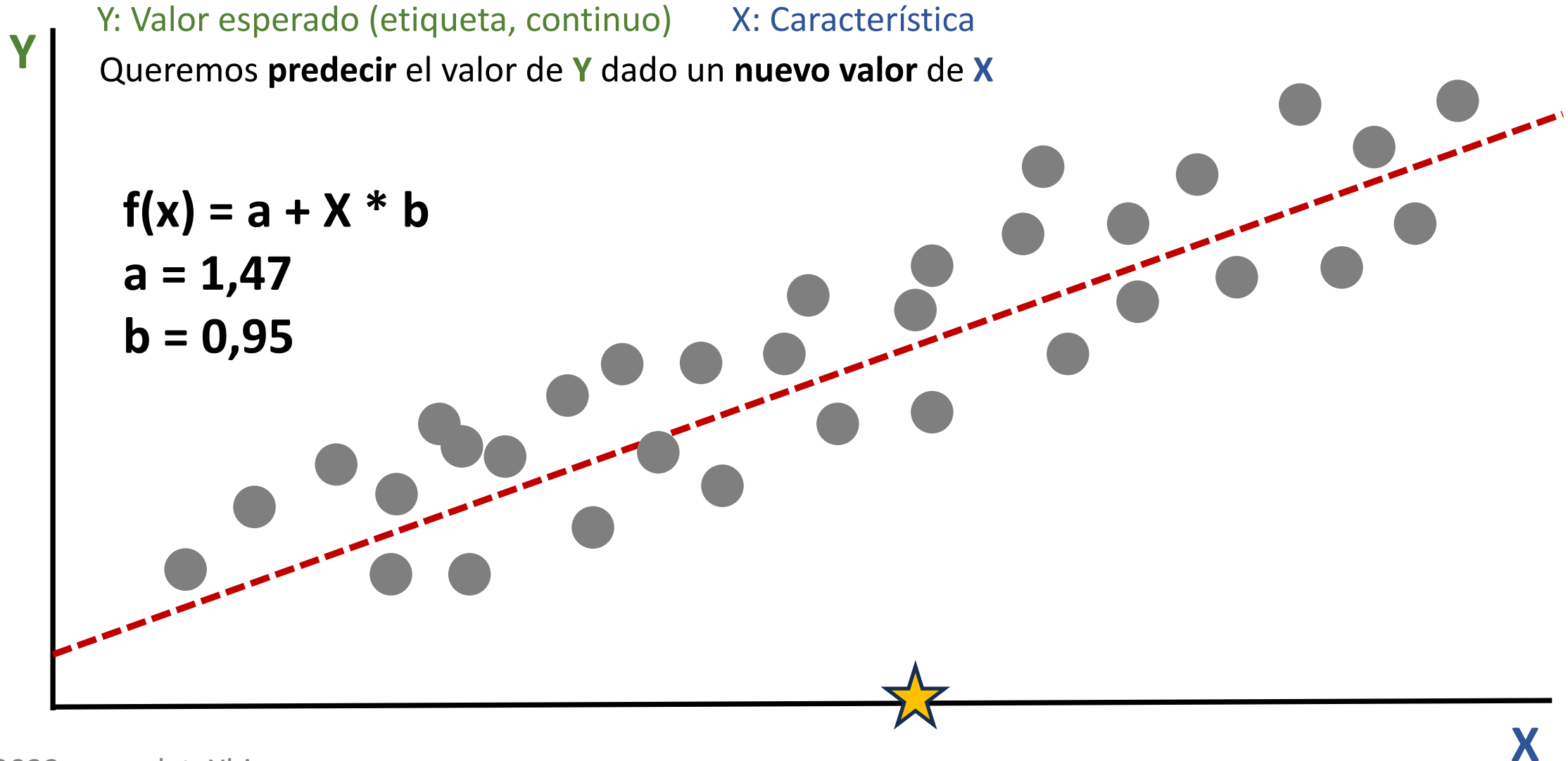
Regresión lineal



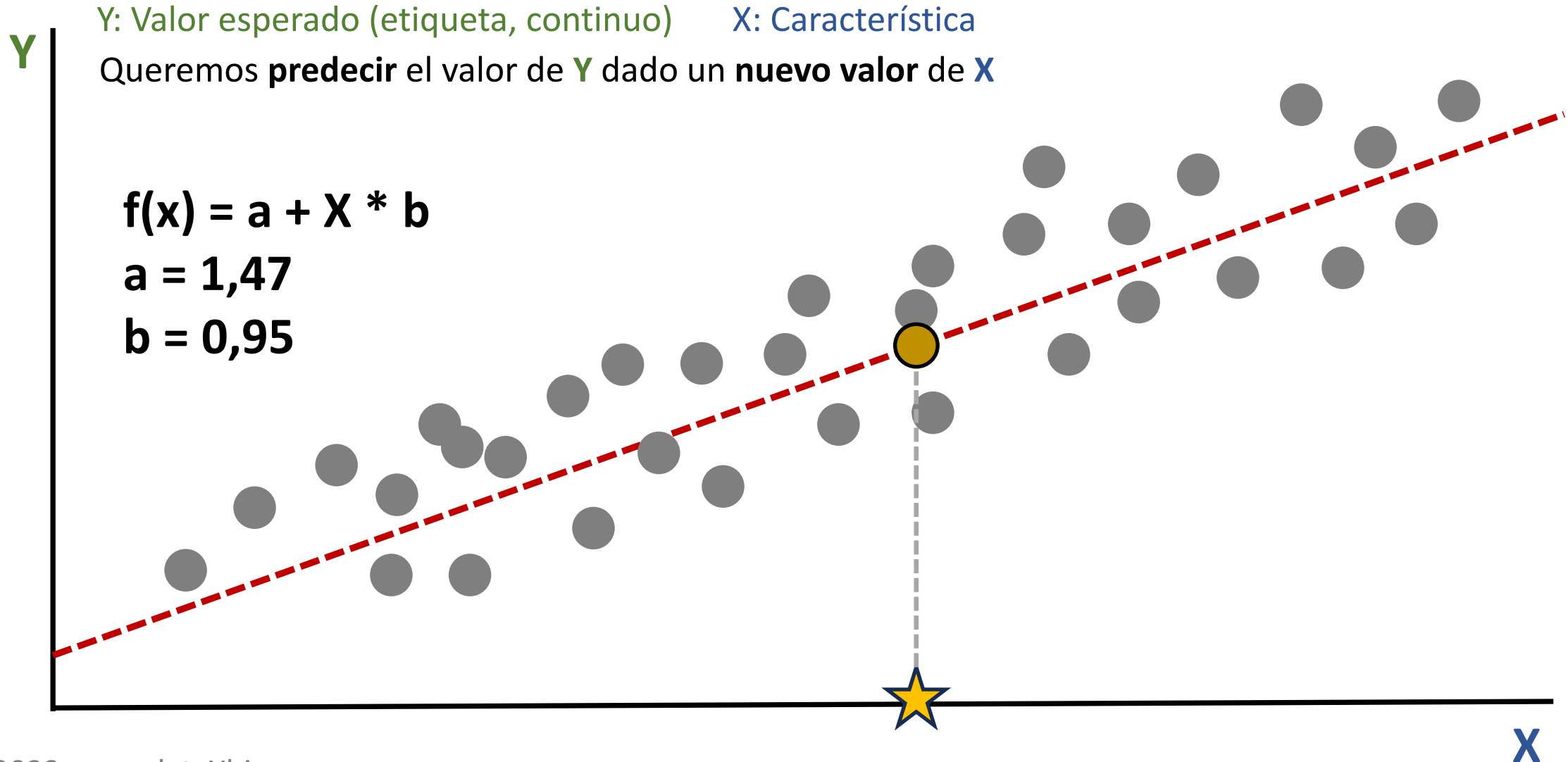
Regresión lineal



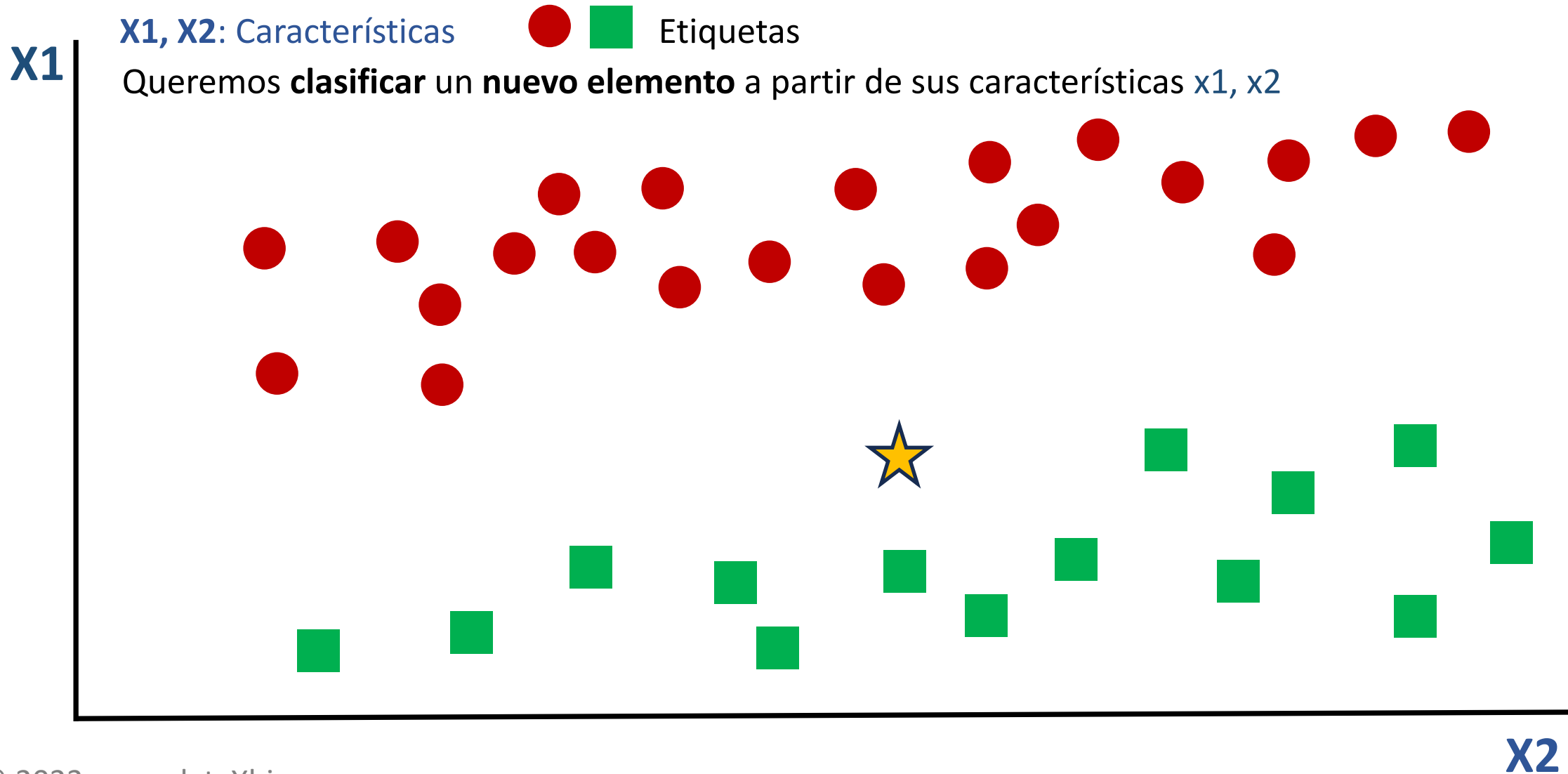
Regresión lineal



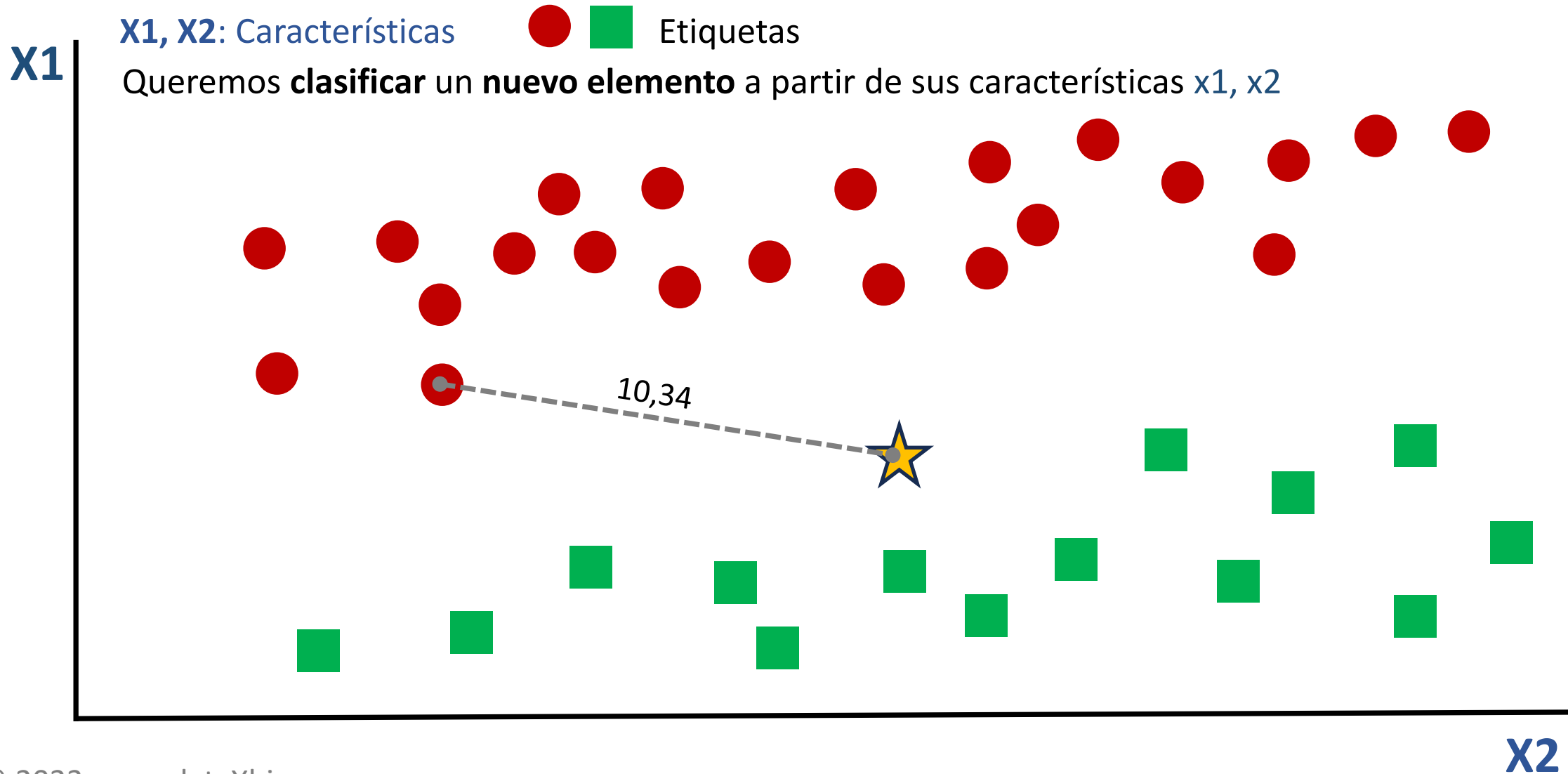
Regresión lineal



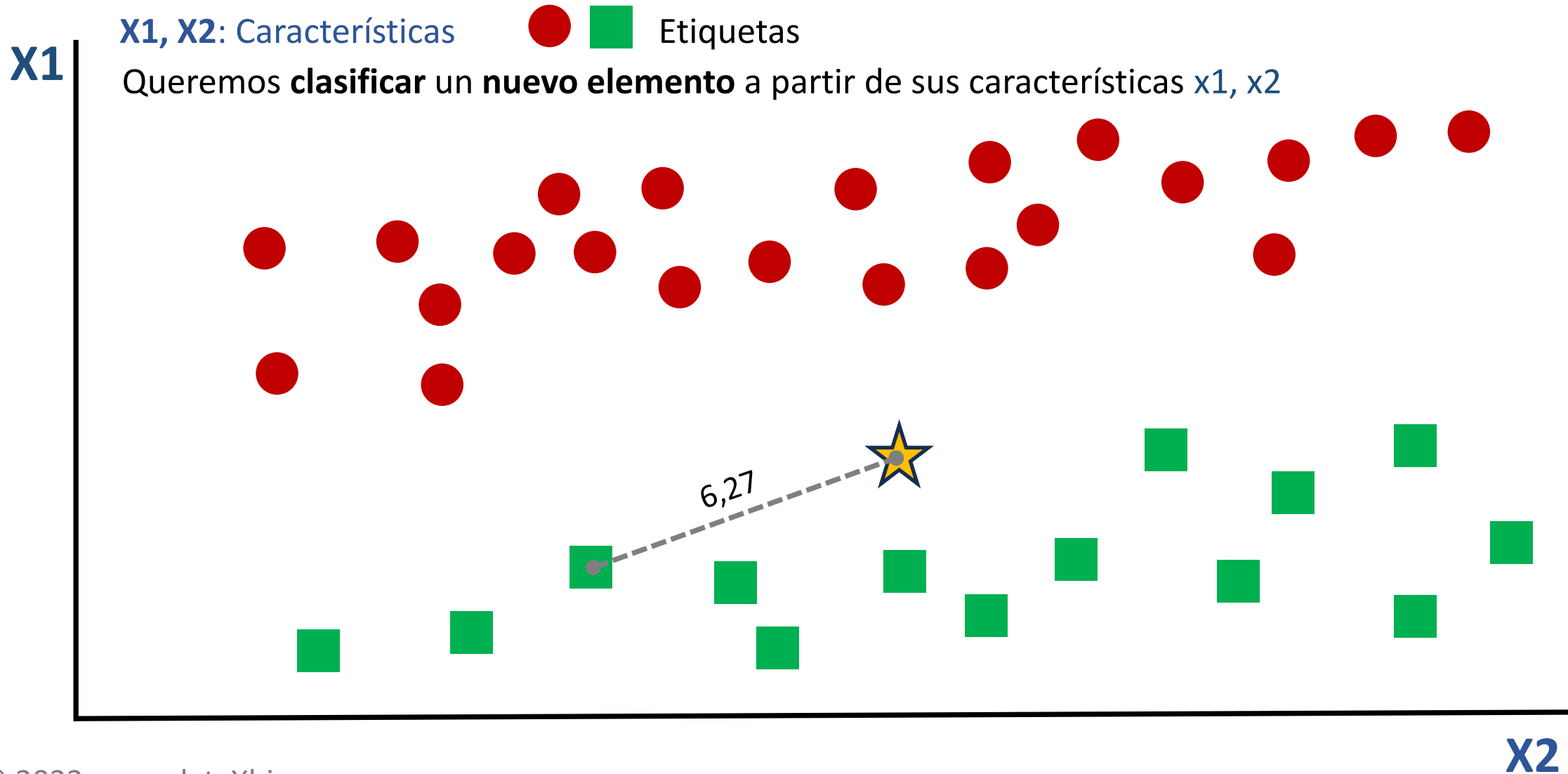
nearest-neighbor



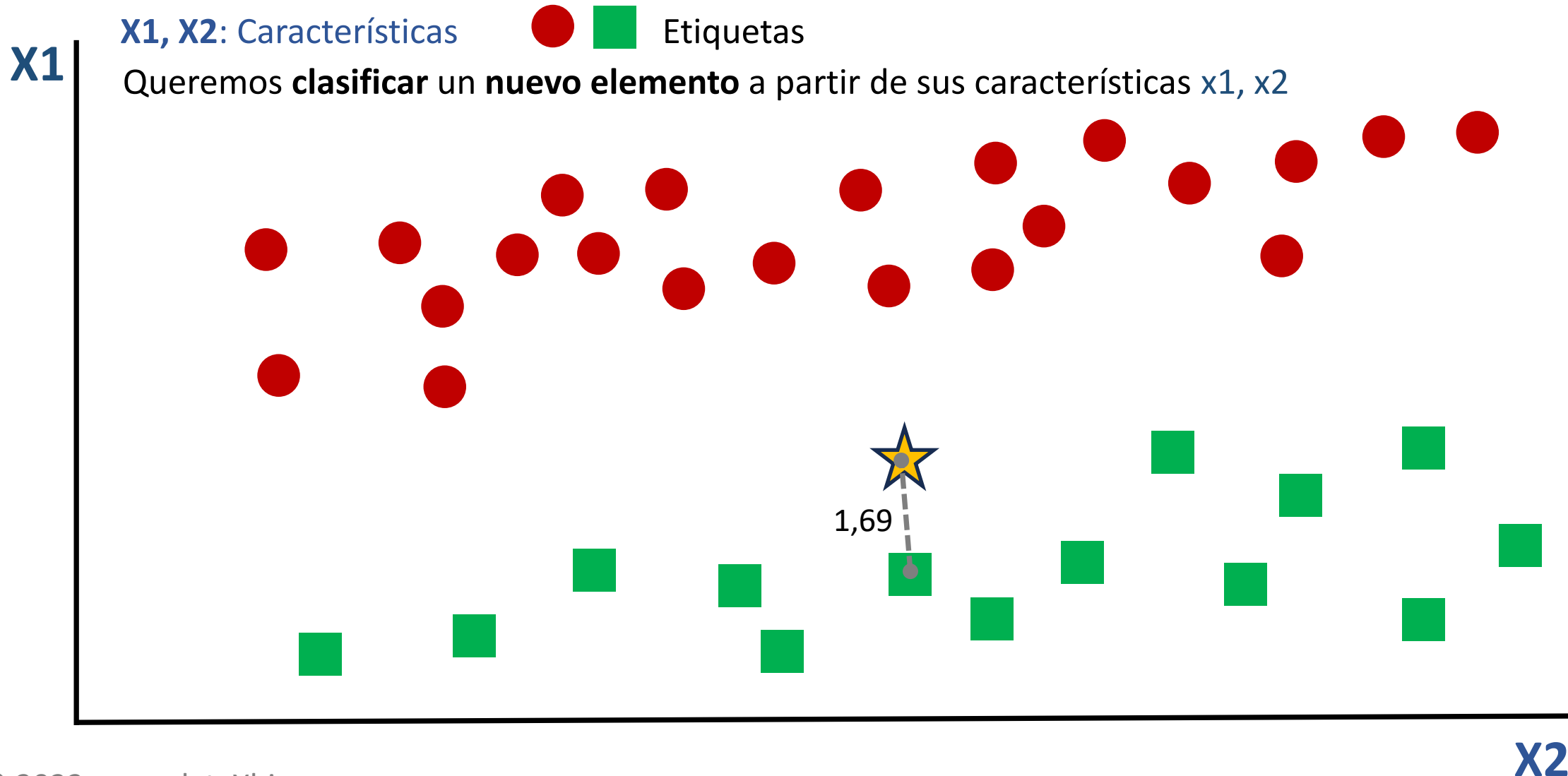
nearest-neighbor



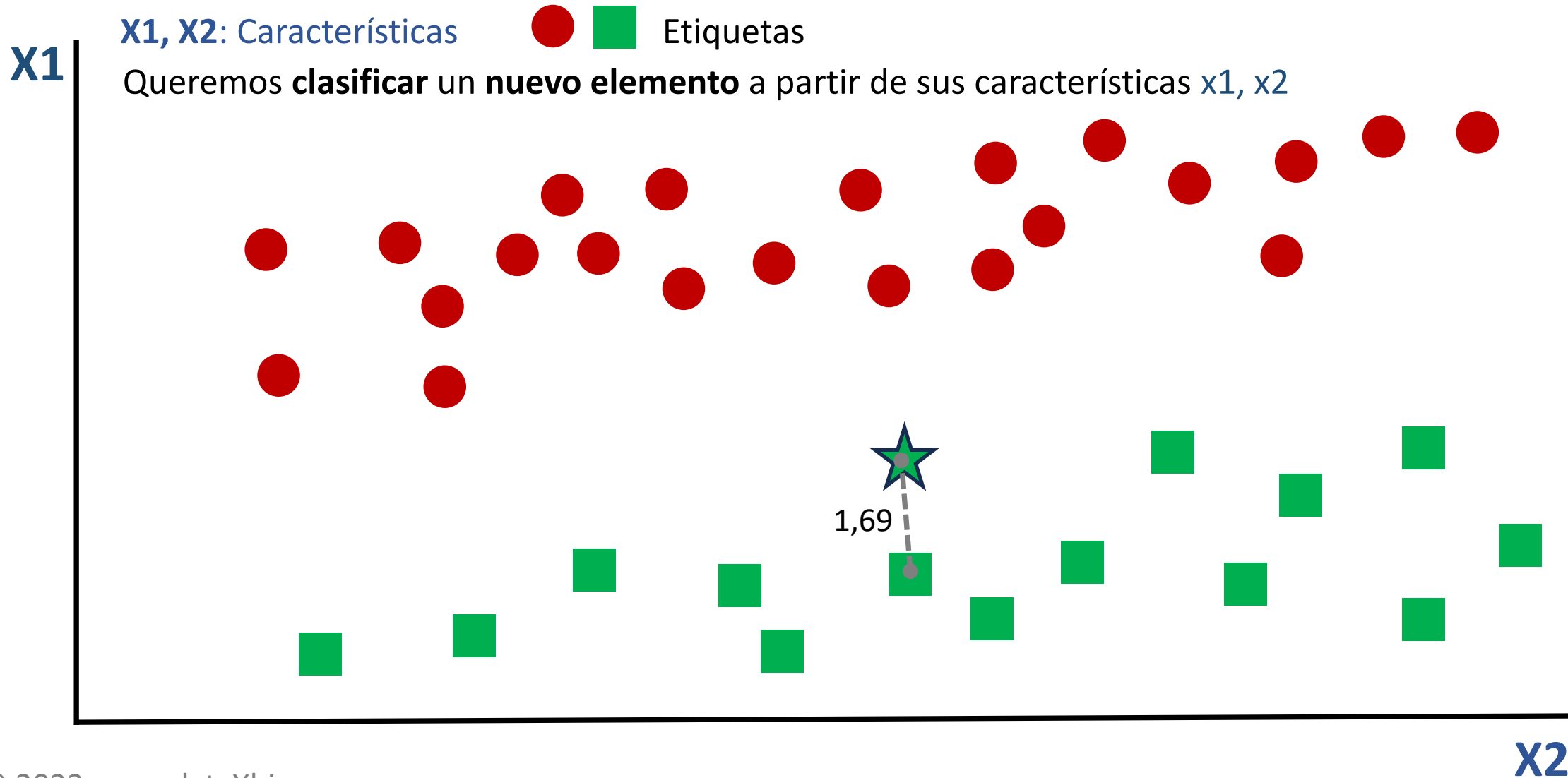
nearest-neighbor



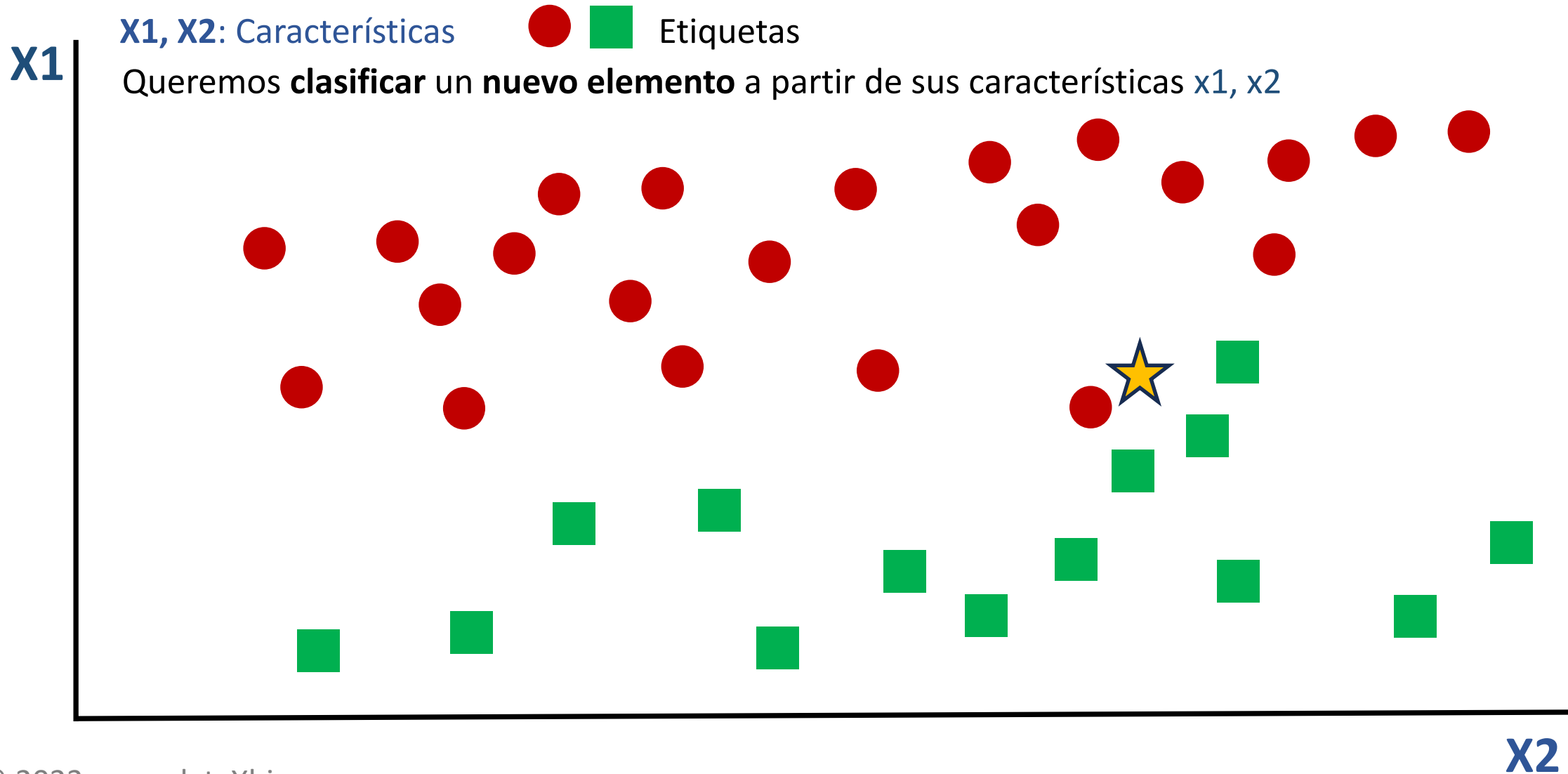
nearest-neighbor



nearest-neighbor



k-nearest-neighbor

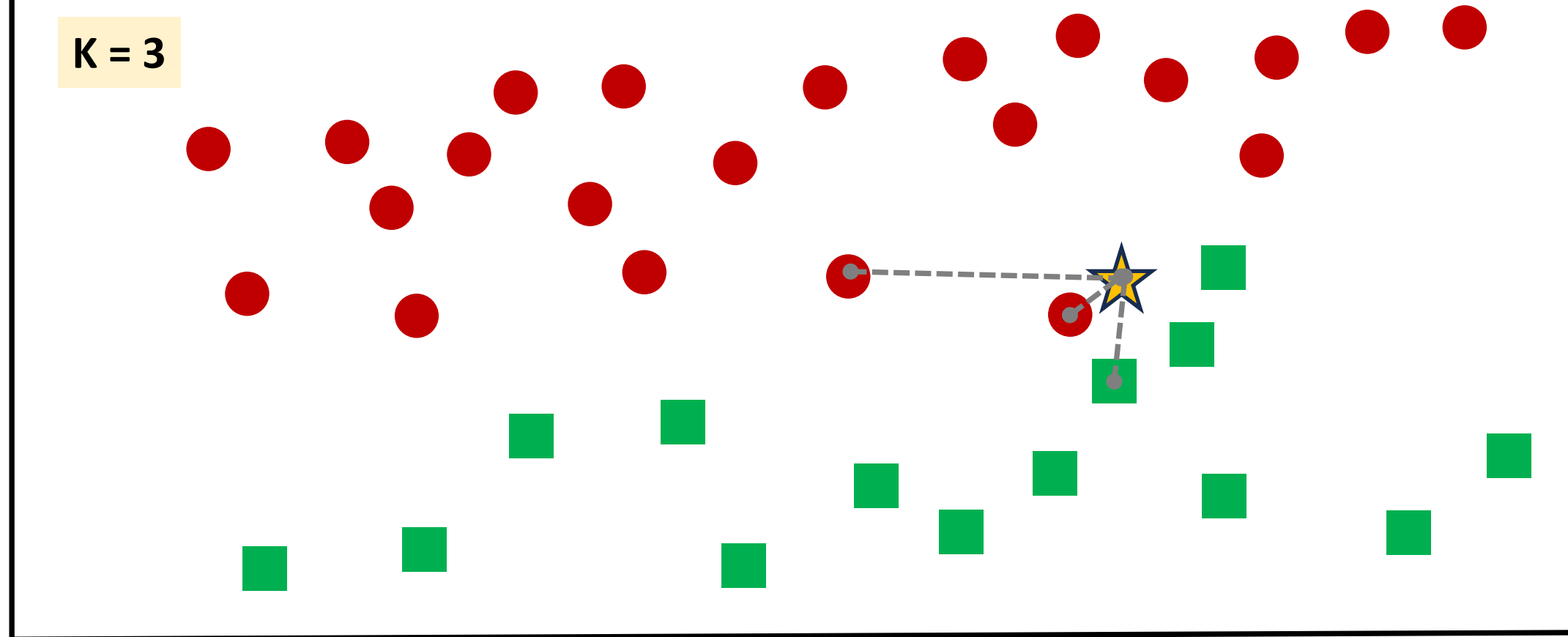


k-nearest-neighbor

 x_1 x_1, x_2 : Características

Etiquetas

Queremos **clasificar** un **nuevo elemento** a partir de sus características x_1, x_2

K = 3 x_2

k-nearest-neighbor

X1

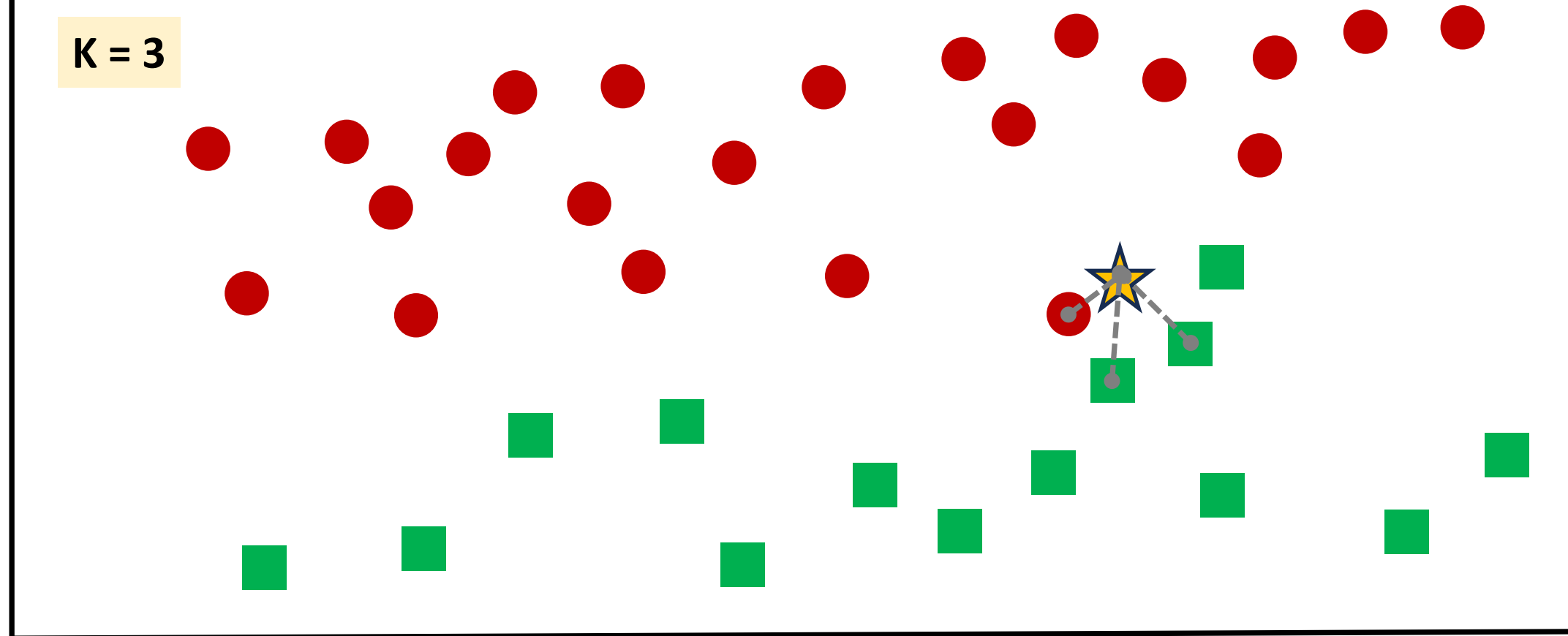
X1, X2: Características



Etiquetas

Queremos **clasificar** un **nuevo elemento** a partir de sus características x_1, x_2

K = 3



X2

k-nearest-neighbor

X1

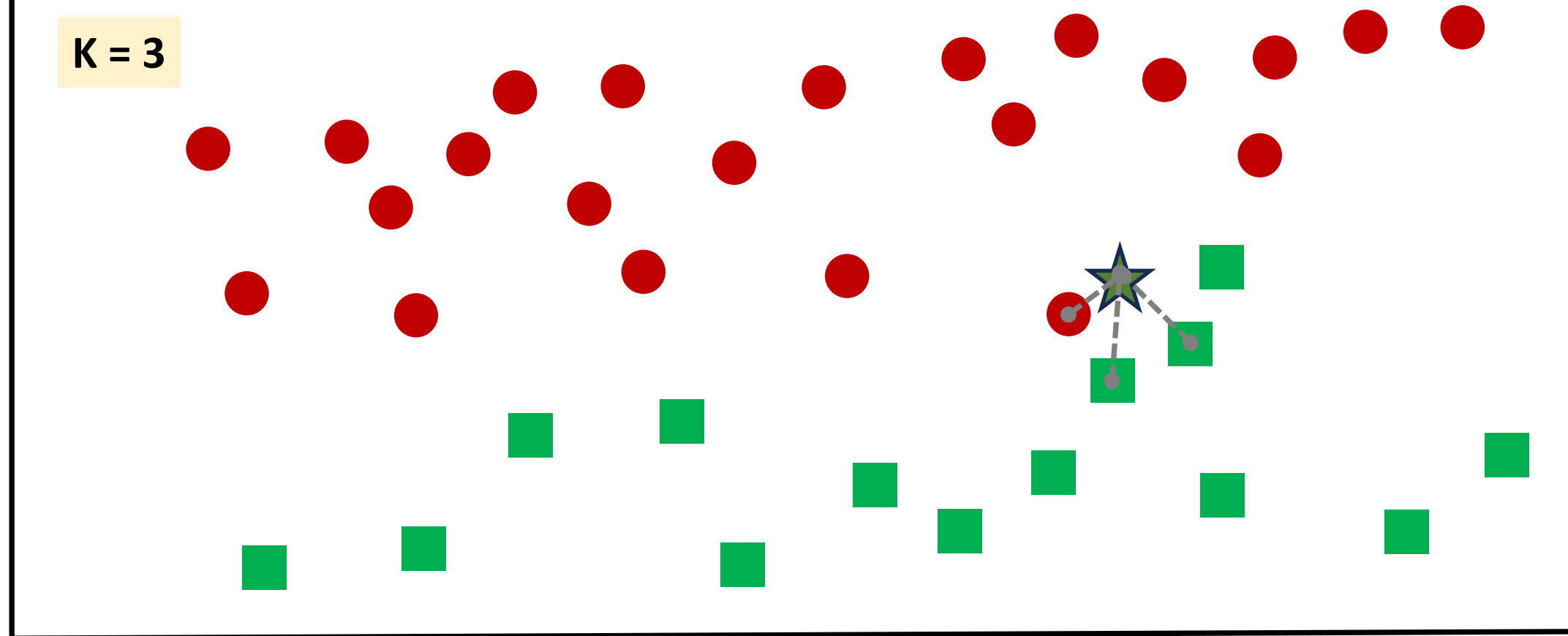
X1, X2: Características



Etiquetas

Queremos **clasificar** un **nuevo elemento** a partir de sus características x_1, x_2

K = 3

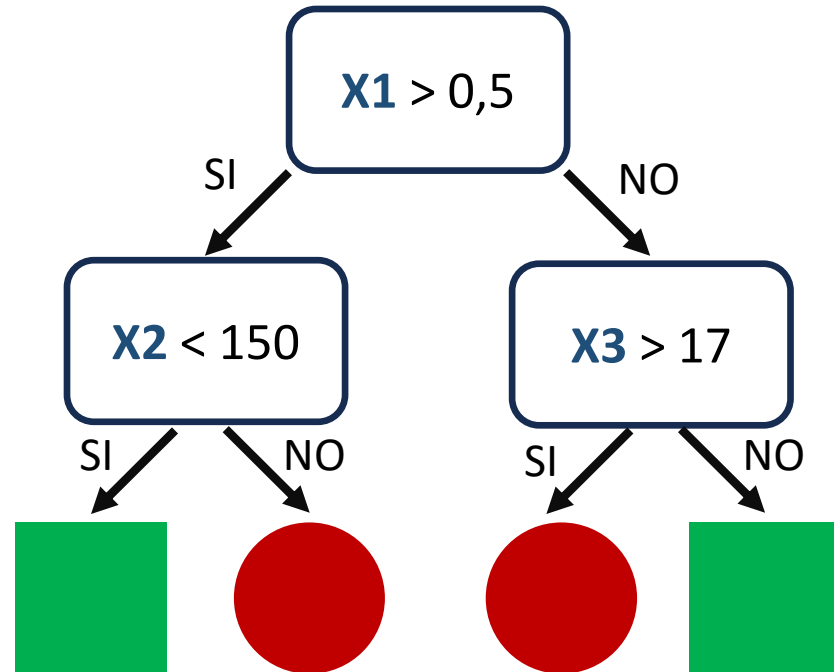


X2

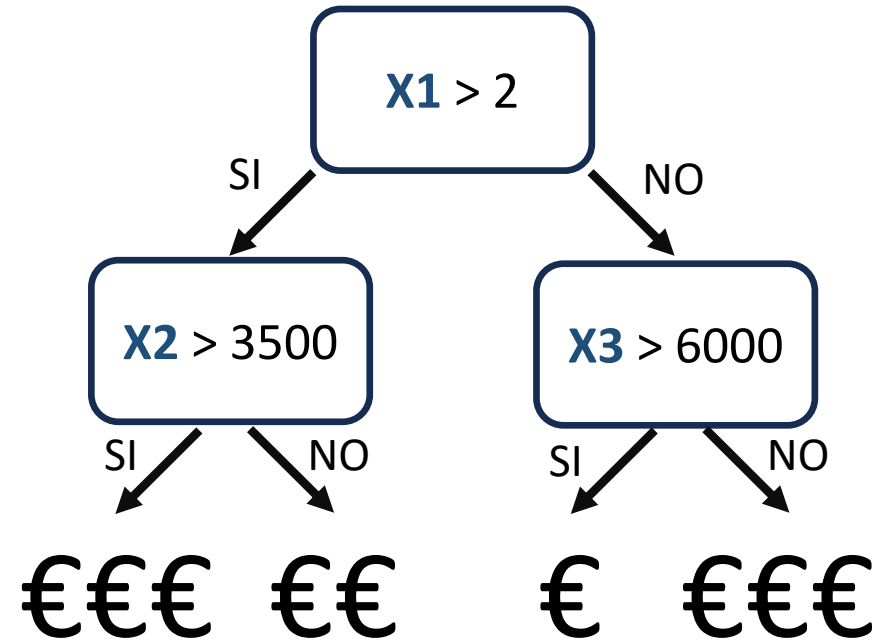
Algoritmos

Árbol de decisión

Queremos **clasificar** un nuevo elemento

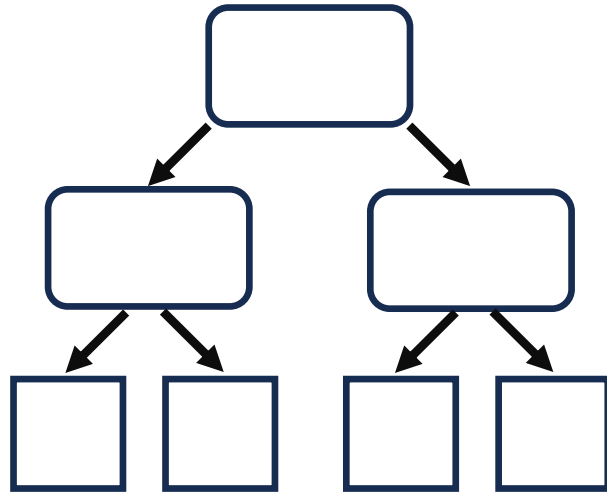


Queremos **predecir** el precio de un producto



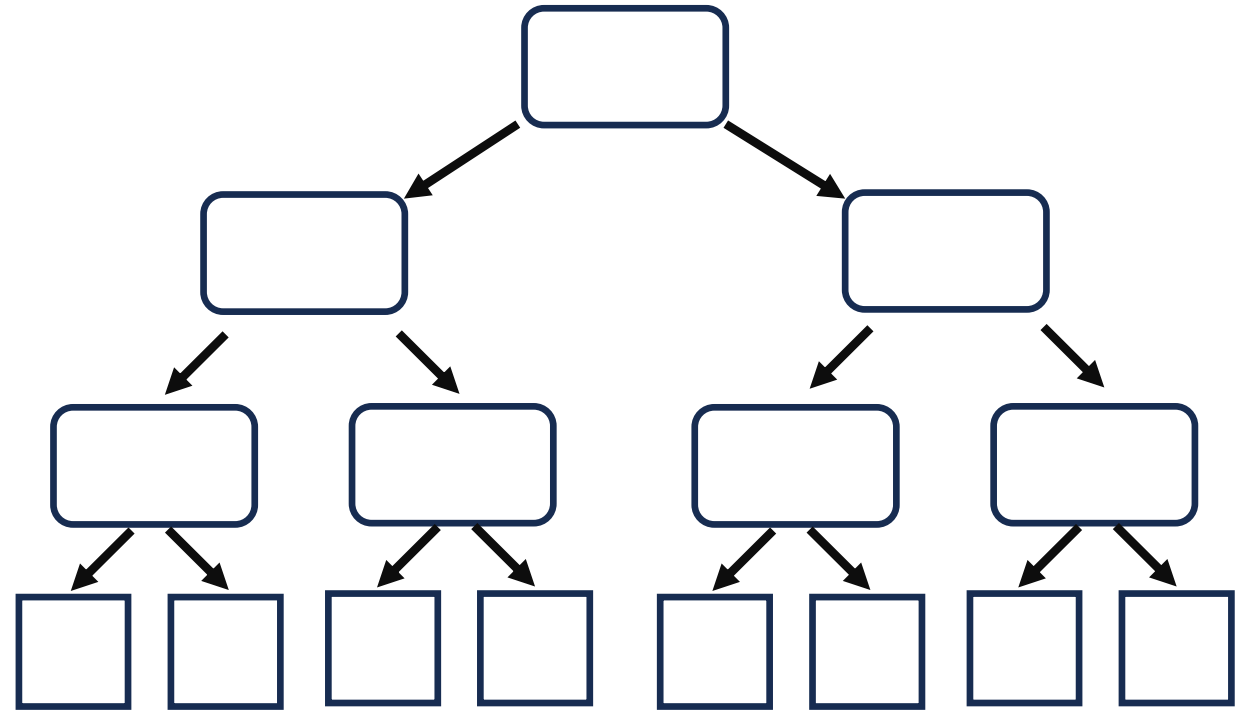
Árbol de decisión

Menor profundidad
Menos hojas



Más datos por hoja
Riesgo de *underfitting*

Mayor profundidad
Más hojas



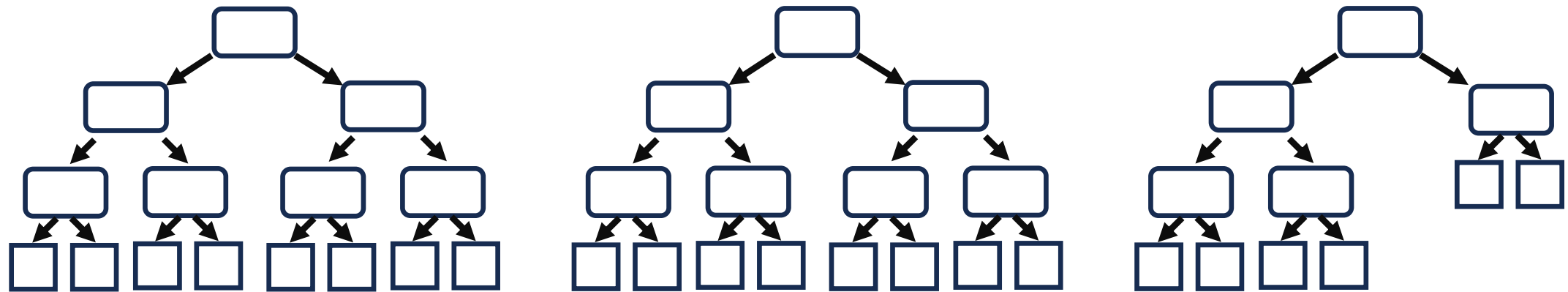
Menos datos por hoja
Riesgo de *overfitting*

Random forest

Se generan **muchos árboles** de decisión (un bosque) utilizando diferentes subconjuntos de los datos de entrenamiento que son seleccionados **aleatoriamente**.

Si es una **clasificación**, el **resultado final** es la **etiqueta que más se repita**.

Si es una **regresión**, el resultado final es el **promedio** de todas las predicciones.



Algoritmos

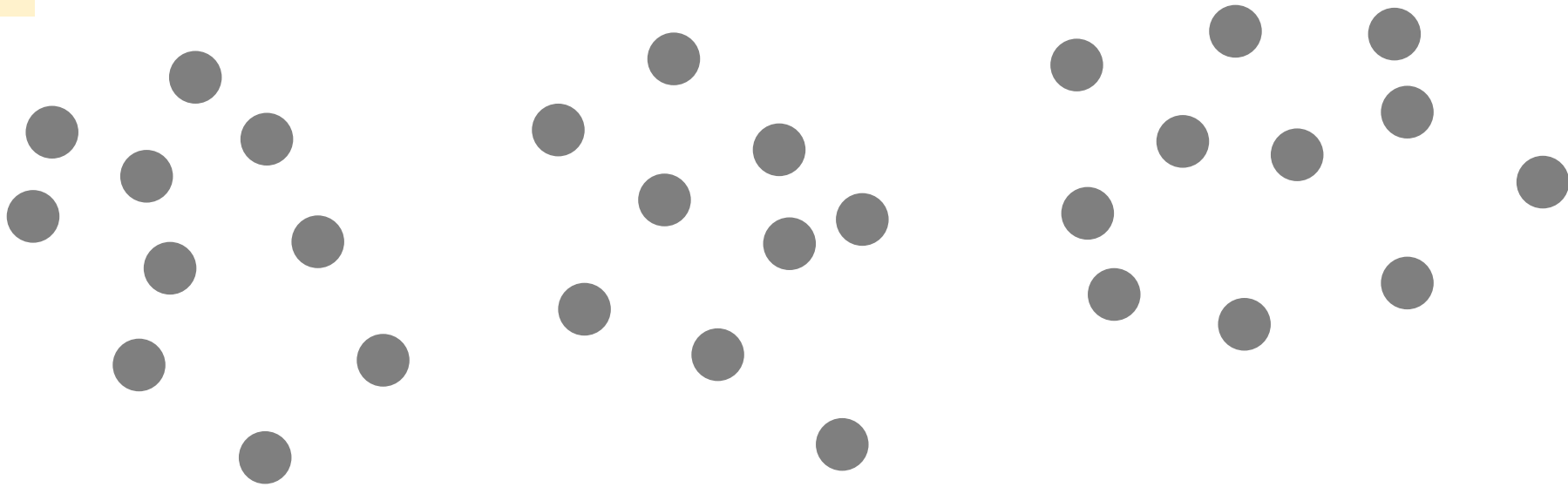
k-means

X1

X1, X2: Características

Queremos **dividir** los datos es **3** grupos

K = 3



X2

Algoritmos

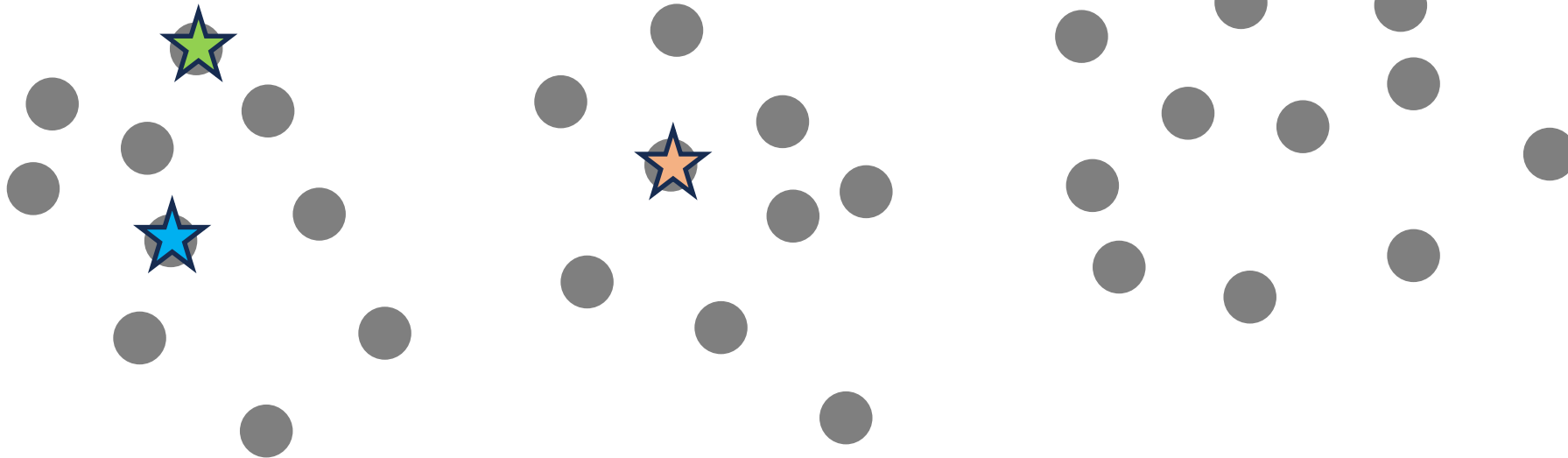
k-means

X1

X1, X2: Características

Queremos **dividir** los datos en **3** grupos

K = 3



K centroides que se posicionan en puntos **aleatorios**

X2

Algoritmos

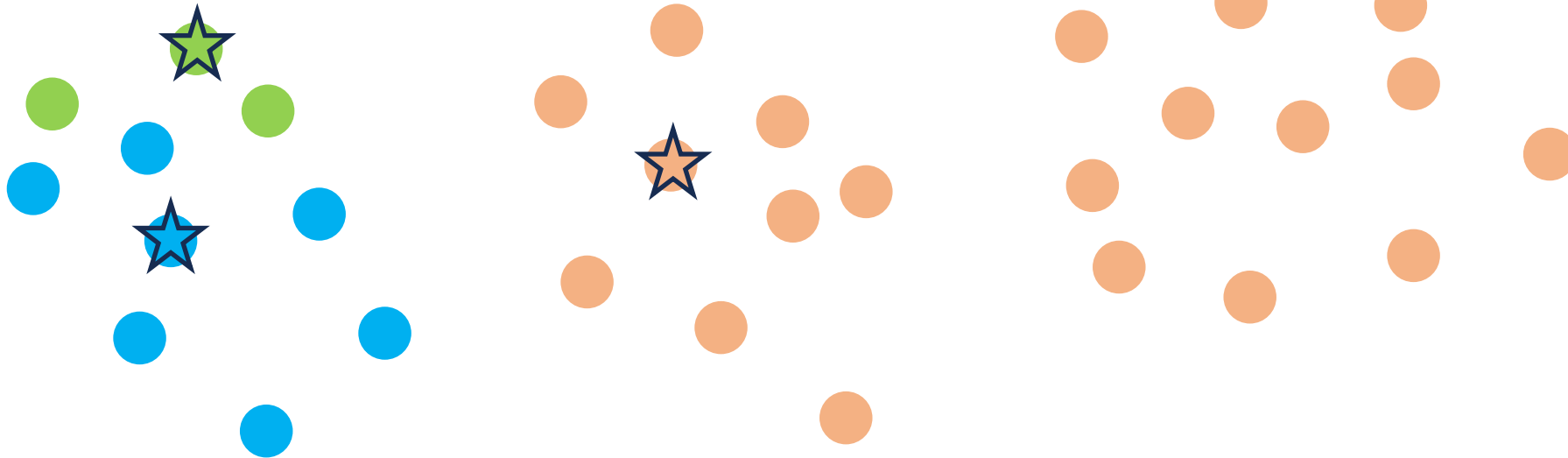
k-means

X1

X1, X2: Características

Queremos **dividir** los datos en **3** grupos

K = 3



Se **etiqueta** cada punto según la **distancia** a cada **centroide**

X2

Algoritmos

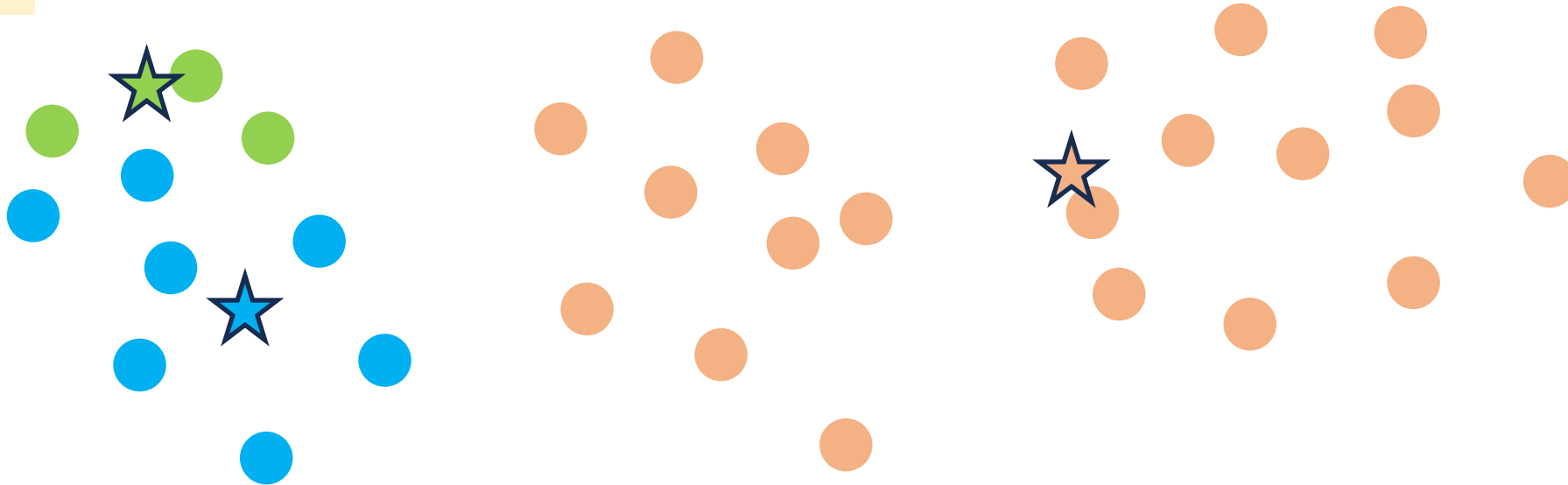
k-means

X1

X1, X2: Características

Queremos **dividir** los datos en **3** grupos

K = 3



Se **mueven** los centroides al punto medio de cada **cluster**

X2

Algoritmos

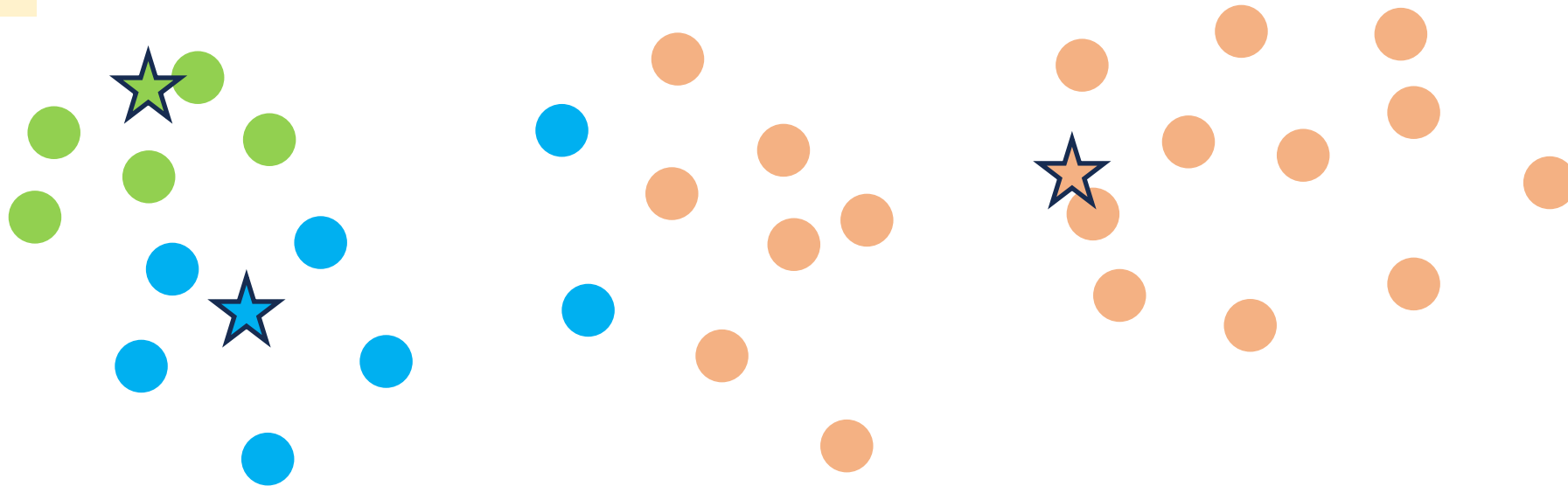
k-means

X1

X1, X2: Características

Queremos **dividir** los datos en **3** grupos

K = 3



Se vuelve a **etiquetar** cada punto según la **distancia** a cada **centroide**

X2

Algoritmos

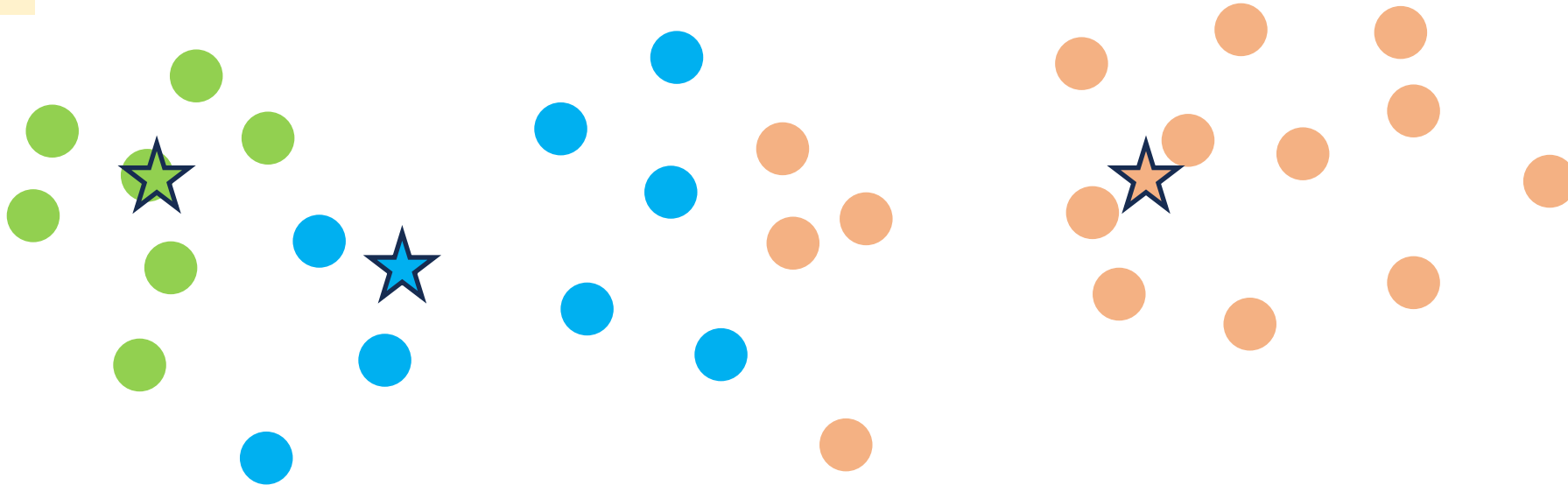
k-means

X1

X1, X2: Características

Queremos **dividir** los datos en **3** grupos

K = 3



Se vuelven a **mover** los centroides al **centro** de cada nuevo **cluster**

X2

Algoritmos

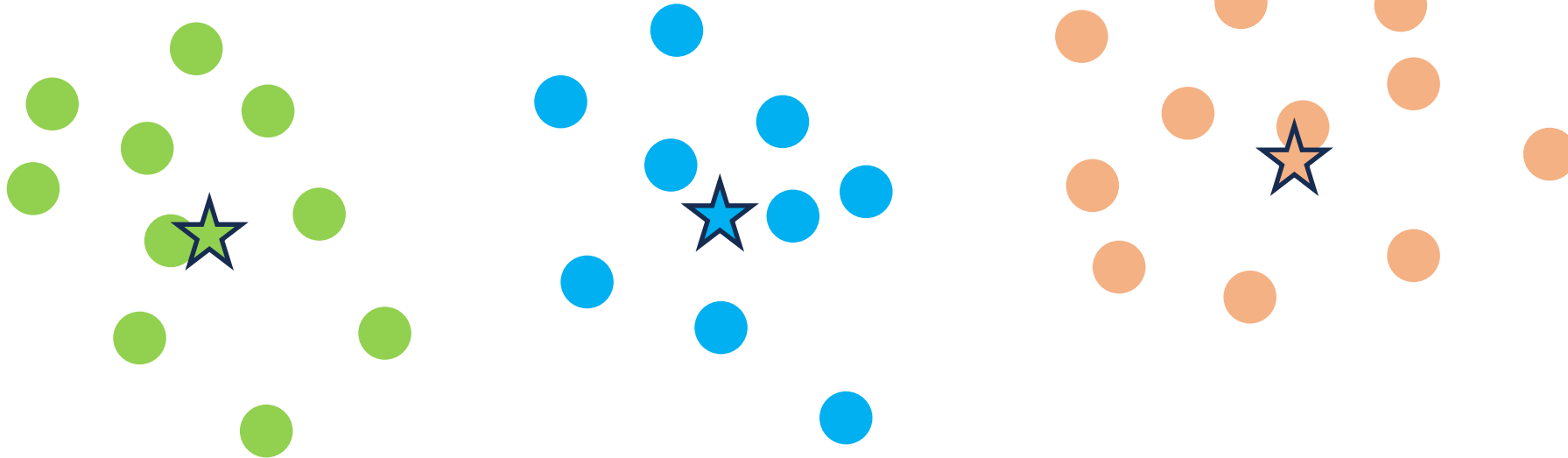
k-means

X1

X1, X2: Características

Queremos **dividir** los datos en **3** grupos

K = 3



El proceso se repite hasta que se **estabiliza**.

X2

Algoritmos

k-means

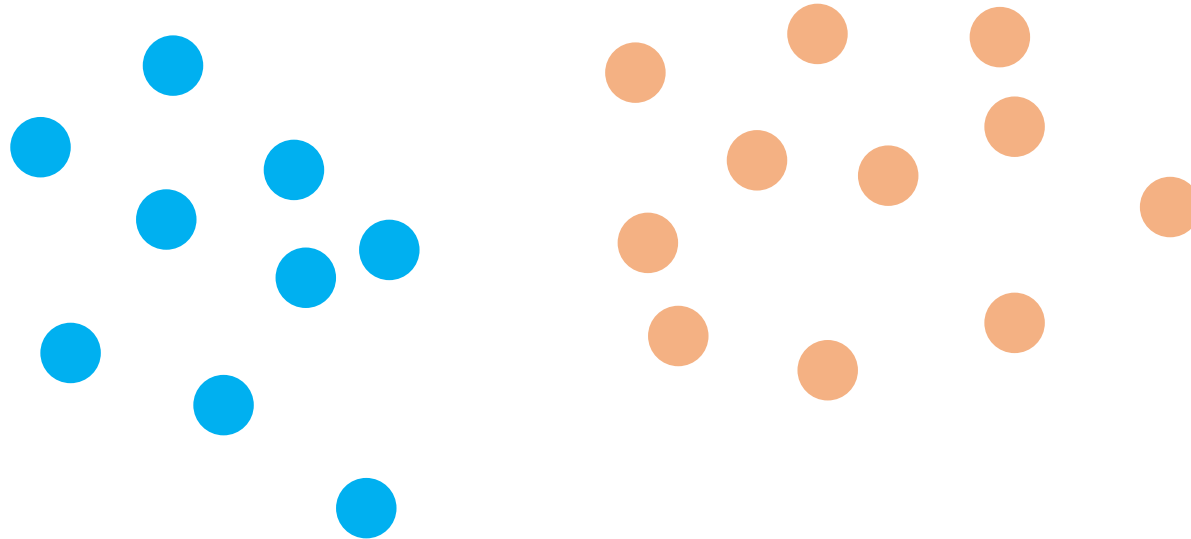
X1

X1, X2: Características

Queremos **dividir** los datos en **3** grupos

K = 3

Clusters finales



X2

Métricas de evaluación de modelos

☐ Hiperparámetros

☐ Métricas para Clasificación

- ☐ *Accuracy*

- ☐ *Precision*

- ☐ *Recall*

- ☐ *F1*

- ☐ *AUC-ROC*

- ☐ *Confusion Matrix*

☐ Métricas para Regresión

- ☐ *MEA (Mean Absolute Error)*

- ☐ *MSE (Mean Squared Error)*

- ☐ *RMSE (Root Mean Squared Error)*

- ☐ *R²*

☐ Métricas para *Clustering*

- ☐ *Silhouette Score*

- ☐ *Davies-Bouldin index (DB)*

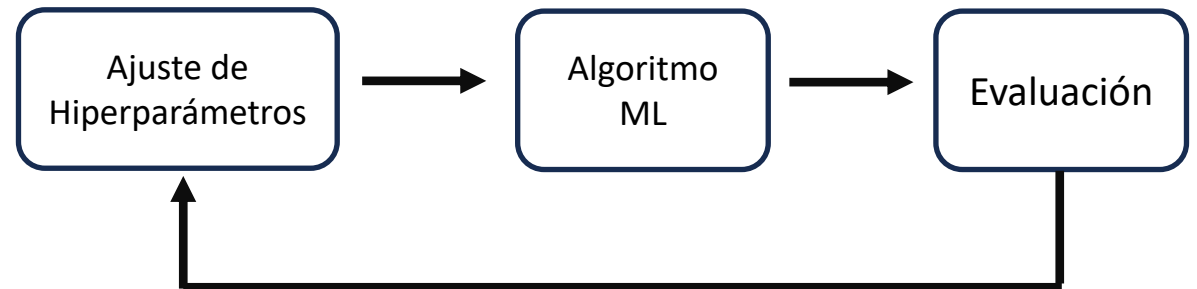
Hiperparámetros

Los algoritmos que utilizamos para entrenar un modelo tienen una serie de parámetros que los propios algoritmos van ajustando durante el entrenamiento.

Los **hiperparámetros** se establecen antes de comenzar el entrenamiento de un modelo y **controlan el comportamiento del algoritmo** y como consecuencia el **rendimiento de modelo** que se está entrenando.

Ejemplos:

- El número de vecinos (K) en k-nearest-neighbor
- La profundidad máxima en Random Forest
- El número de clusters (K) en k-means



Métricas de evaluación de modelos

Métricas para Clasificación

Accuracy

Relación de predicciones que coinciden exactamente con las etiquetas de clase

$$A = \text{Predicciones Correctas} / \text{Total Predicciones} * 100$$

Precisión

Capacidad de un modelo para evitar que las etiquetas negativas se etiqueten como positivas

$$P = \text{Verdaderos Positivos (TP)} / (\text{Verdaderos Positivos (TP)} + \text{Falsos Positivos (FP)})$$

Recall

Capacidad de un modelo para detectar todas las muestras positivas

$$R = \text{Verdaderos Positivos (TP)} / (\text{Verdaderos Positivos (TP)} + \text{Falsos Negativos (FN)})$$

F1

La fórmula utiliza una combinación de las métricas **precisión** y **recall**.

Mide el equilibrio entre Falsos Positivos (FP) y Falsos Negativos (FN). Sin embargo, no tiene en cuenta los Verdaderos Negativos (VN).

AUC-ROC

Área bajo la curva ROC (Característica Operativa del Receptor)

Todas estas métricas tienen un intervalo entre 0 y 1 y **cuanto más cerca del 1, mejor**.

Métricas para Clasificación

Matriz de Confusión

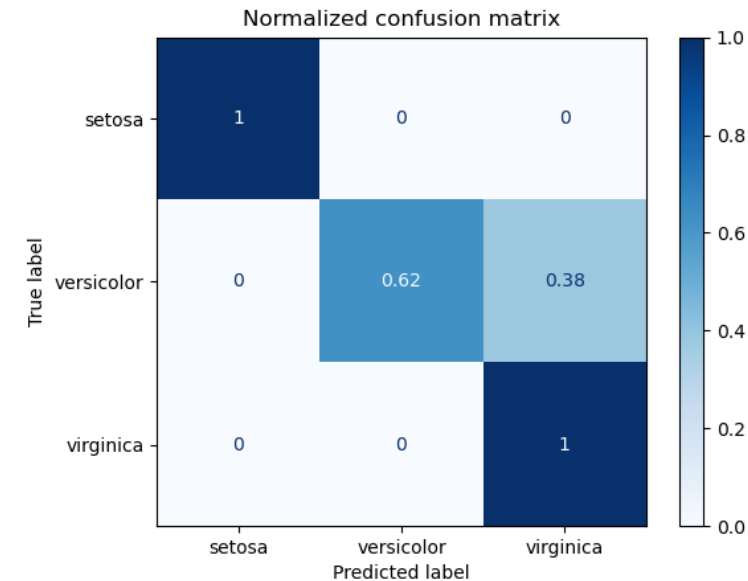
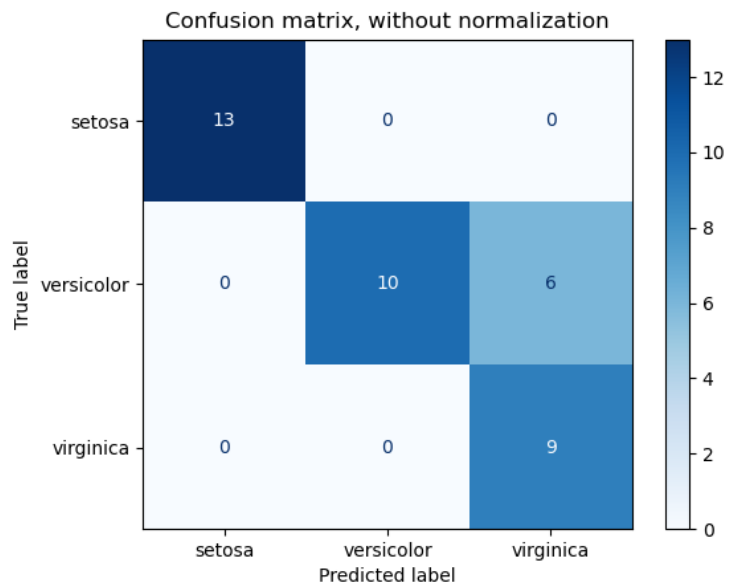
Compara los **valores de las etiquetas** (valores esperados) **versus** los valores de las **predicciones**.

Hay una fila y una columna para cada clase (categoría).

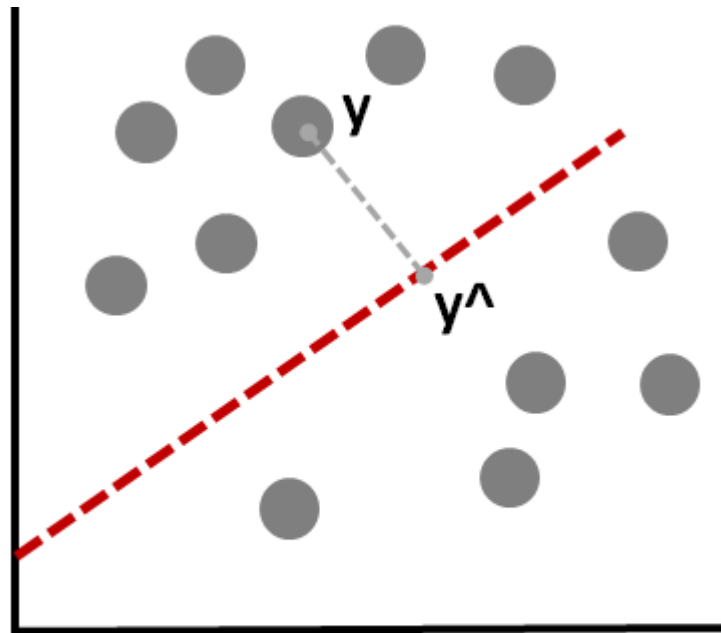
Las **filas** representan las **etiquetas** y las **columnas** las **predicciones**.

Los números de la **diagonal** indican las **predicciones correctas**.

En la **matriz de la derecha** los valores están **normalizados** por el número de muestras en cada clase



Métricas para Regresión



y es el valor esperado (target)
y-hat es el valor de la predicción

Mean Squared Error (MSE)

$$MSE = \frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2$$

Mean Absolute Error (MAE)

$$MAE = \frac{1}{N} \sum_{j=1}^N |y_j - \hat{y}_j|$$

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2}$$

R2

Indica que porcentaje de la variación en la estimación se puede explicar por la variación en las entradas X.

Tiene un intervalo entre 0 y 1, aunque puede ser negativo.

Por lo general un **R2** alto (cercano a 1) es bueno.

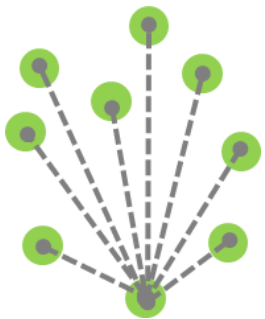
Métricas para Clustering

Silhouette Score

Evalúa la **cohesión interna** de cada cluster y la **separación entre clusters**.

Tiene un intervalo entre -1 y 1.

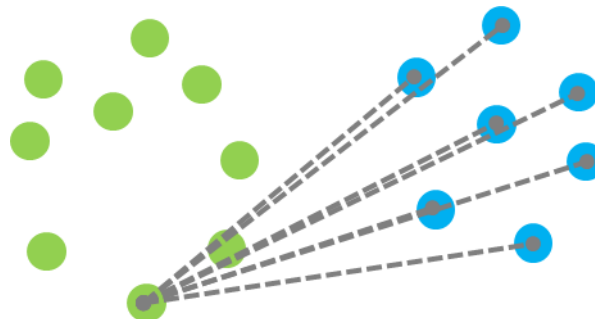
Un **valor** próximo a **1** es **bueno** e indica que los **puntos** están muy **cerca de su propio cluster** y **lejos de los otros clusters**.



Cohesión

Se calcula la **distancia** media entre los **puntos** de un **mismo cluster**.

A **menor distancia**, **mayor cohesión** y **mejor** es el **modelo**.



Separación

Se calcula la **distancia** media de un punto de **un cluster** a todos los puntos de los **otros clusters** más cercanos.

A **mayor separación**, **mejor** es el **modelo**.

Métricas para Clustering

Davies-Bouldin index (DB)

Evalúa la **compactación** de cada cluster y la **separación entre clusters**.

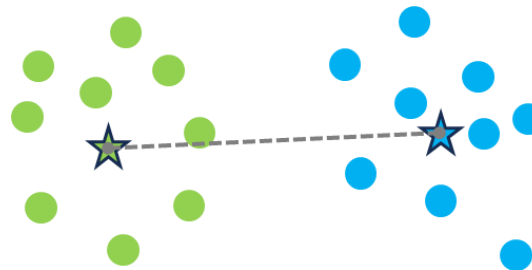
Tiene un intervalo entre 0 y sin límite superior .

Un **valor** bajo (próximo a 0) es **bueno** e indica que los **puntos** están muy **cerca de su propio cluster** y **lejos de los otros clusters**.



Compactación

Se calcula la **distancia** de todos los **puntos** de un **mismo cluster** a su **centroide**.



Separación

Se calcula la **distancia** entre nos.

A **mayor separación**, **mejor** es el **modelo**.



Ejercicios

- ☐ Regresión lineal de datos sintéticos
- ☐ Clasificación con el conjunto de datos Iris
- ☐ Clustering con el conjunto de datos Iris



Librería **scikit-learn**

`conda install -c conda-forge scikit-learn`