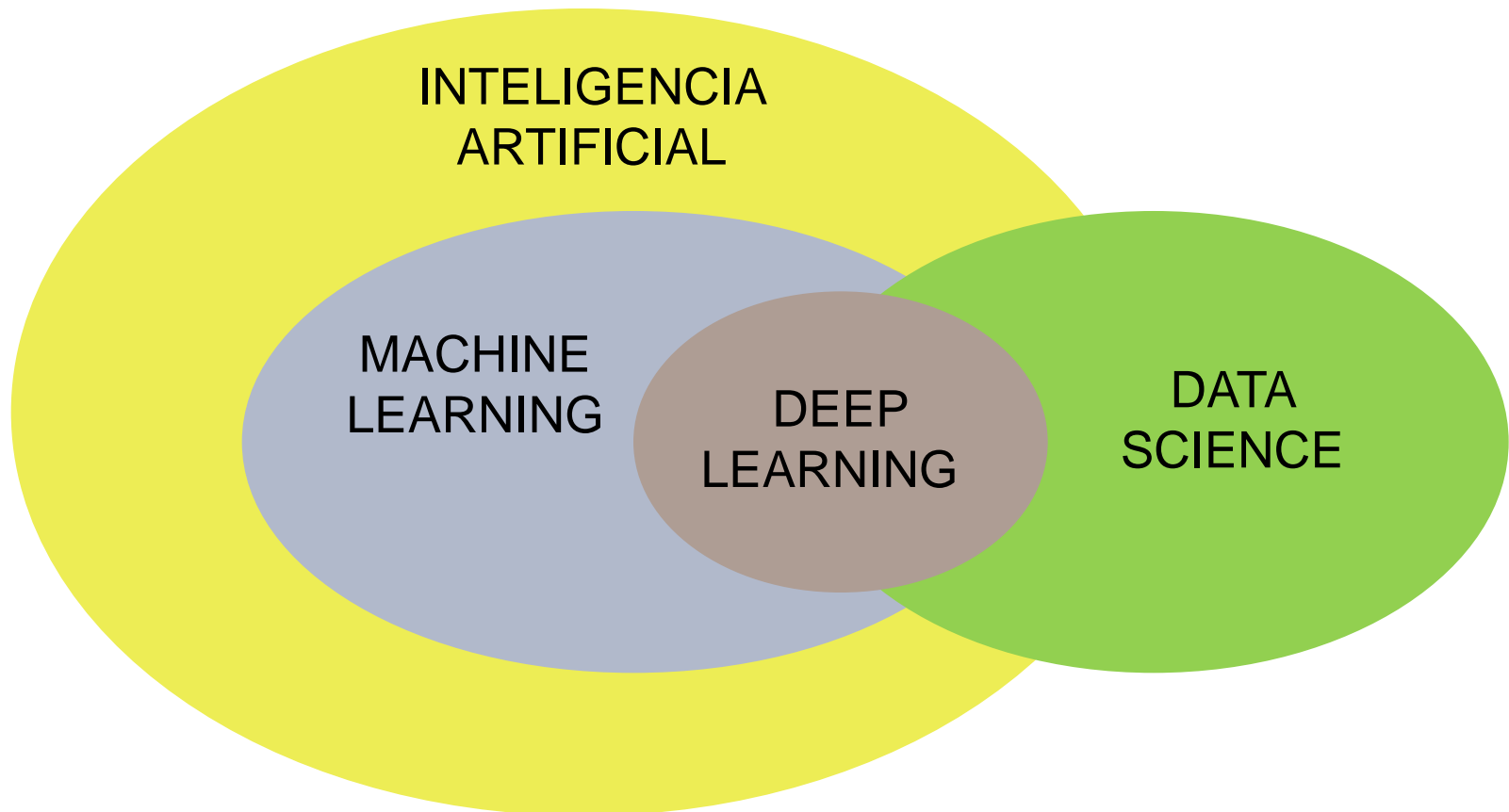


# **Aprendizaje Automatizado: Introducción**

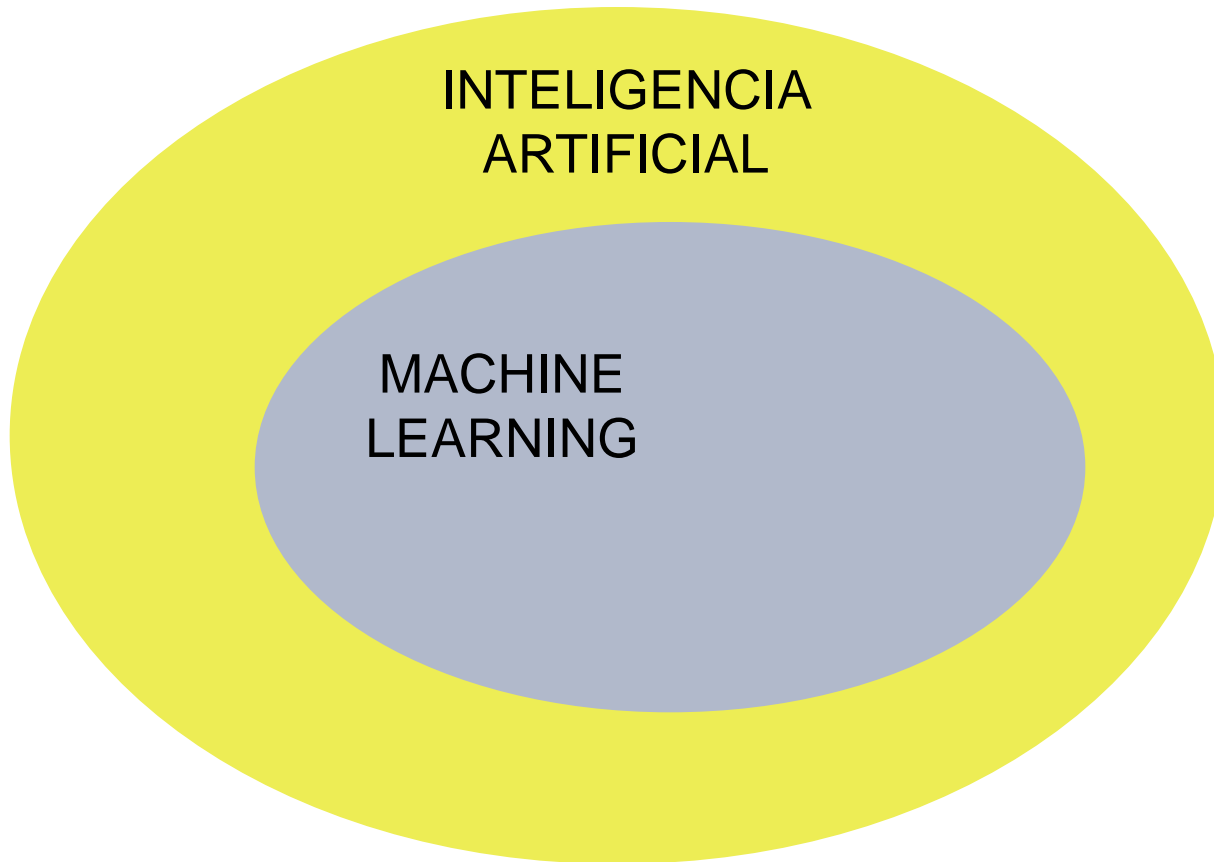
**IIA - LCC**

---

# Aprendizaje Automatizado



# Aprendizaje Automatizado



# Aprendizaje Automatizado

Trata problemas que pueden ser simples de “entender o reconocer” pero muy difíciles de “definir” y convertir en algoritmo

- Ejemplo: Detectar una sonrisa en una cara, reconocer un gato, interpretar una secuencia de sonidos para traducirlo en palabras
- El Aprendizaje Automatizado introduce métodos que pueden resolver alguna de esas tareas “aprendiendo” la solución a partir de ejemplos de cómo se realizan

# Aprendizaje Automatizado

## Problemas en AA

- Clasificación
- Regresión
- Ranking-Retrieval
- Detección de novedades
- Clustering
- Etc.

# Clasificación

Problema:

Dado un objeto (descrito por un conjunto de características medidas de alguna forma) asignarle una (o varias) etiqueta/s de un conjunto finito.

Ejemplo:

- Asignar un símbolo alfanumérico a una secuencia de movimientos del lápiz en la pantalla táctil

# Regresión

Problema:

Dado un objeto asignarle un número real.

Ejemplos: predecir...

- la relación euro-dolar de mañana.
- niveles de stock/ventas a futuro.
- cuestiones climáticas...

# Búsqueda-recomendación

Problema:

Dado un objeto (necesidad de información/query), asignarle y ordenar las respuestas más adecuadas dentro de una base de datos.

Ejemplo:

- Buscadores en Internet
- Sistemas de recomendación



# Detección de novedades

Problema:

Detectar "outliers", objetos que son diferentes a los demás.

Ejemplos:

- Alarmas de comportamiento en compras con tarjeta.
- Detección de fallas en equipos críticos.

# Cuándo un programa aprende?

“Se dice que un programa aprende si mejora su performance en una cierta tarea al incorporar experiencia”

- ✓ Memorizar no es aprender
- ✓ Generalizar es aprender

# Cómo generalizar?

Tengo estos datos:

8 – T

2 – T

5 – F

9 – F

4 – T

13 – F

Cual es la respuesta  
para 12?

Y si agrego los datos:

14 – F

16 - T

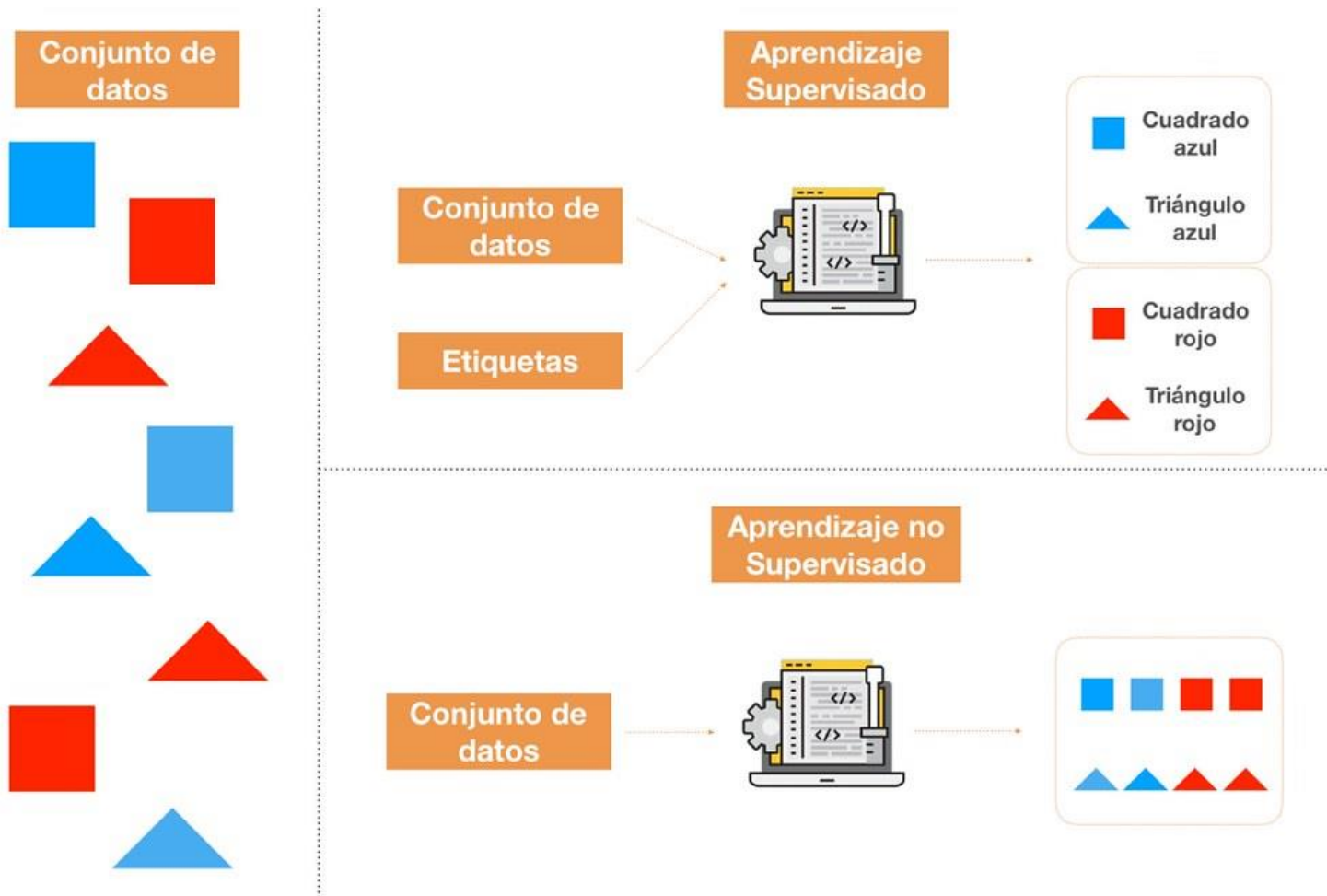
# Cómo generalizar?

- Para generalizar incorporamos “algo” a los datos: un bias.
- En general usamos la “navaja de Occam”: La respuesta más simple que explica las observaciones es la válida
- Distintos métodos de ML usan distintos bias  
«en igualdad de condiciones, la explicación más sencilla suele ser la más probable».

# Aprendizaje Automatizado

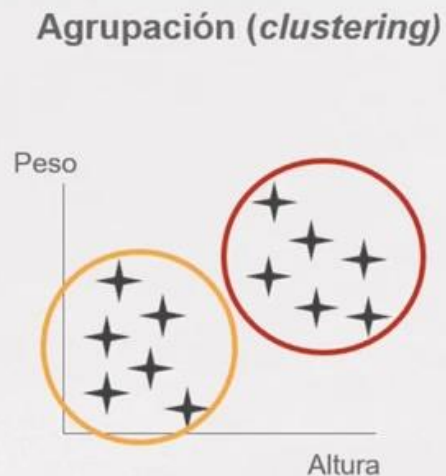
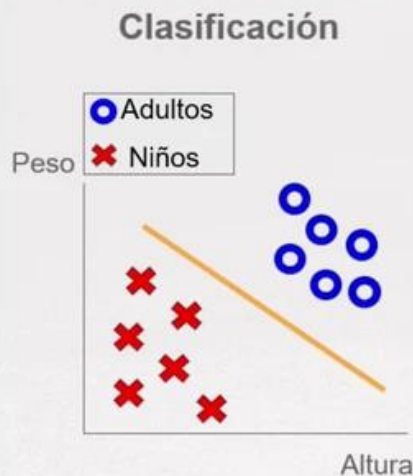
- Programas que mejoran “su comportamiento” con la experiencia.
- Dos formas de adquirir experiencia:
  - A partir de ejemplos suministrados por un usuario (un conjunto de ejemplos clasificados o etiquetados). **APRENDIZAJE SUPERVISADO.**
  - Mediante exploración autónoma (ej. software que aprende a jugar al ajedrez mediante la realización de miles de partidas contra sí mismo). **APRENDIZAJE NO SUPERVISADO.**

# AA: Supervisado / No supervisado



# AA: Supervisado / No supervisado

## Técnicas de Machine Learning



APRENDIZAJE SUPERVISADO

APRENDIZAJE NO SUPERVISADO

# Tipos de Aprendizaje

- Aprendizaje inductivo.
  - Datos de entrada específicos: un usuario provee un subconjunto de todas las posibles situaciones.
  - Datos de salida generales: regla o modelo que puede ser aplicada a una nueva situación.
- Aprendizaje por refuerzo.
  - Sistemas que aprenden mediante prueba y error.
  - Exploración autónoma para inferir reglas de comportamiento.
- Otros: Aprendizaje deductivo (EBL), Razonamiento basado en casos (CBR)...

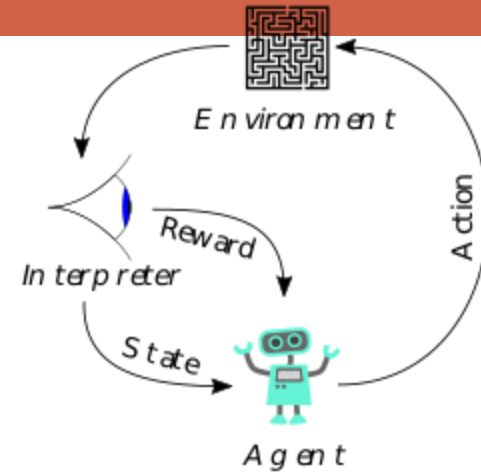


# Aprendizaje por Refuerzo

- No hay fuente de información (no hay datos de entrada).
- El sistema aprende mediante **prueba y error**.
- Se realiza una exploración autónoma para inferir reglas de comportamiento (aprendizaje no supervisado).
- El sistema realiza una determinada tarea repetidamente, para adquirir experiencia y mejorar su comportamiento.
- Se requiere un número de repeticiones muy elevado.

**EXPLORACIÓN AUTÓNOMA -> MODELOS**

# Aprendizaje por Refuerzo



Aplicaciones: en procesos que se realizan como una secuencia de acciones:

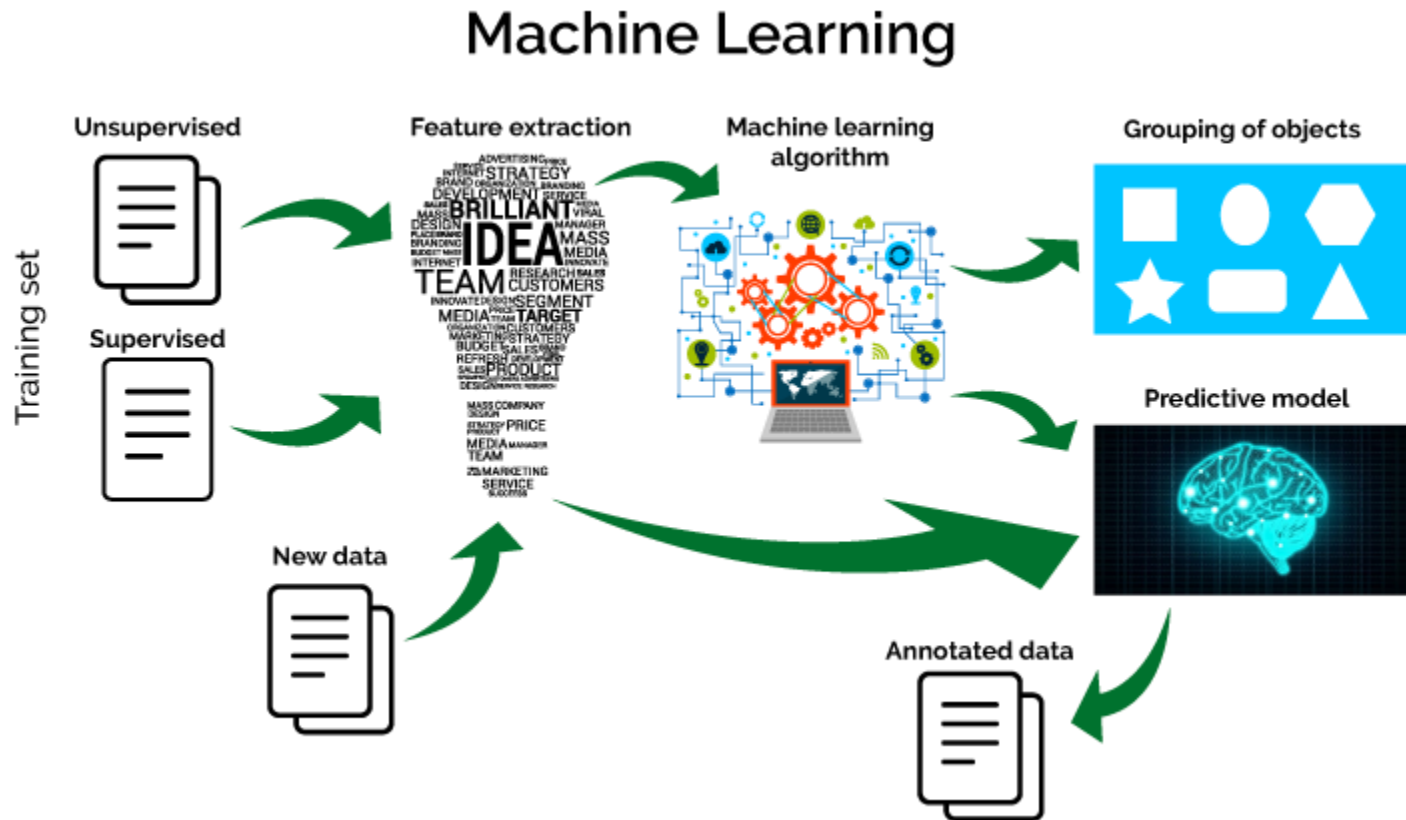
- Robots móviles: aprendizaje de la forma de escapar de un laberinto.
- Juego de ajedrez: aprendizaje de la mejor secuencia de movimientos para ganar un juego.
- Brazo robot: aprendizaje de la secuencia de acciones a aplicar a las articulaciones para conseguir un cierto movimiento.
- Separación de imágenes de grandes grupos: caras de personas – animales – ninguno de ellos

# Aprendizaje Inductivo

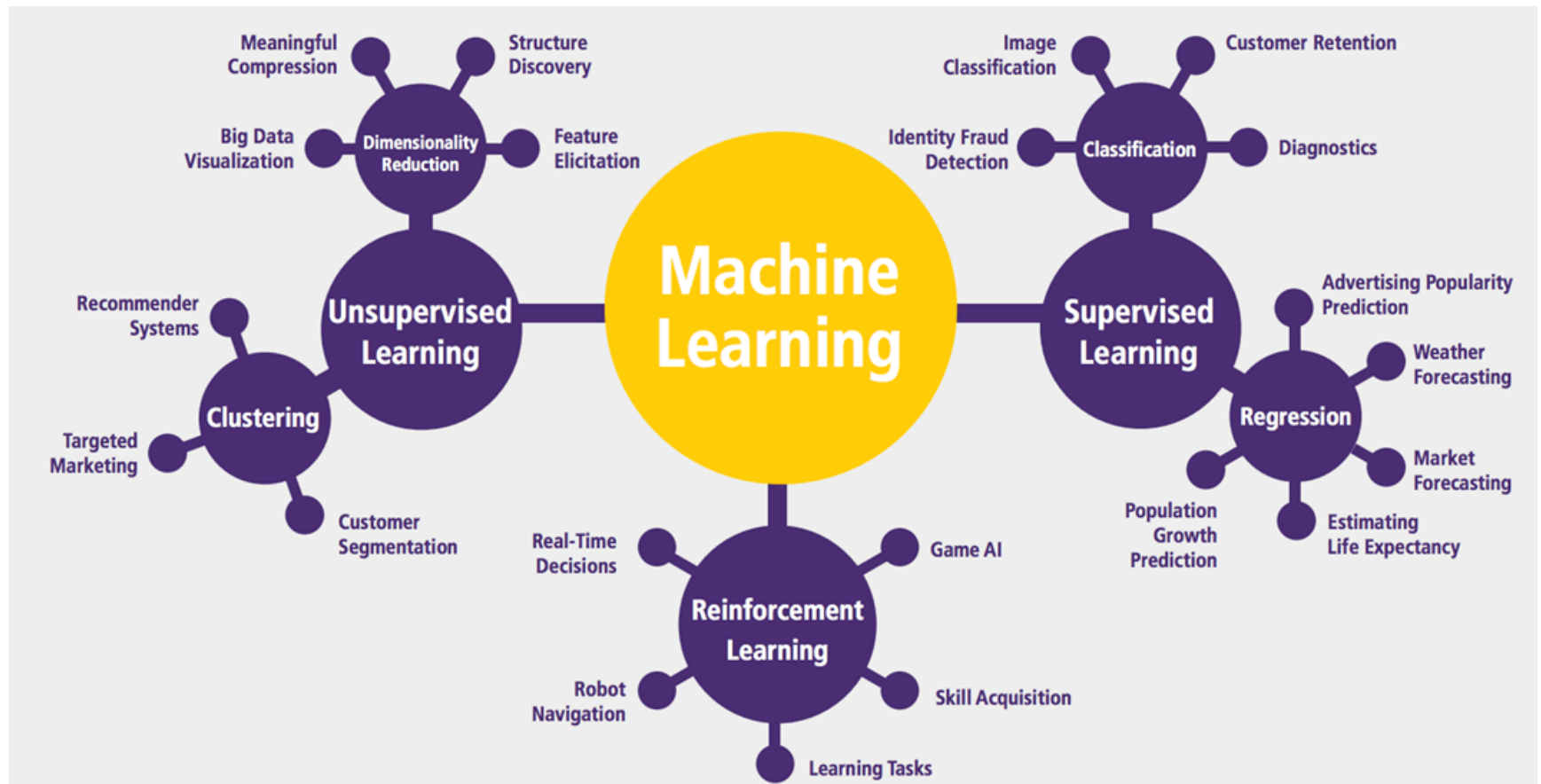
- El objetivo es generar un modelo a partir de ejemplos.
- El conjunto de ejemplos usados se llama **conjunto de entrenamiento**.
- Cuatro elementos fundamentales: modelo resultante (hipótesis), instancias, atributos y clases.

**EJEMPLOS ESPECÍFICOS -> MODELO GENERAL**

# Aprendizaje Inductivo



# Aprendizaje Automatizado



# Definiciones

- **Resultado**: **modelo** que se infiere a partir de los ejemplos.
- **Instancia**: cada uno de los ejemplos.
- **Atributo**: cada una de las propiedades que se miden (observan) de un ejemplo.
- **Clase**: el atributo que debe ser deducido a partir de los demás.

# Ejemplo

Modelado de la estimación de fallo de una máquina.

- **Clases:** la máquina fallará / no fallará.
- **Atributos:**
  - Temperatura.
  - Nivel de vibraciones.
  - Horas de funcionamiento.
  - Meses desde la última revisión.

# Ejemplo

- **Instancias:** ejemplos pasados (situaciones conocidas). [Temp = alta, Nivel vibrac. = bajo, horas = 800, meses = 2, fallo = SÍ]
- **Resultado:** por ej. relación entre los valores de las variables y la clase resultante.
  - Regla: SI *nivel\_vibraciones* = *alto* Y *temp* = *alta* ENTONCES *fallará*.
  - Red Neuronal para clasificación



# Atributos

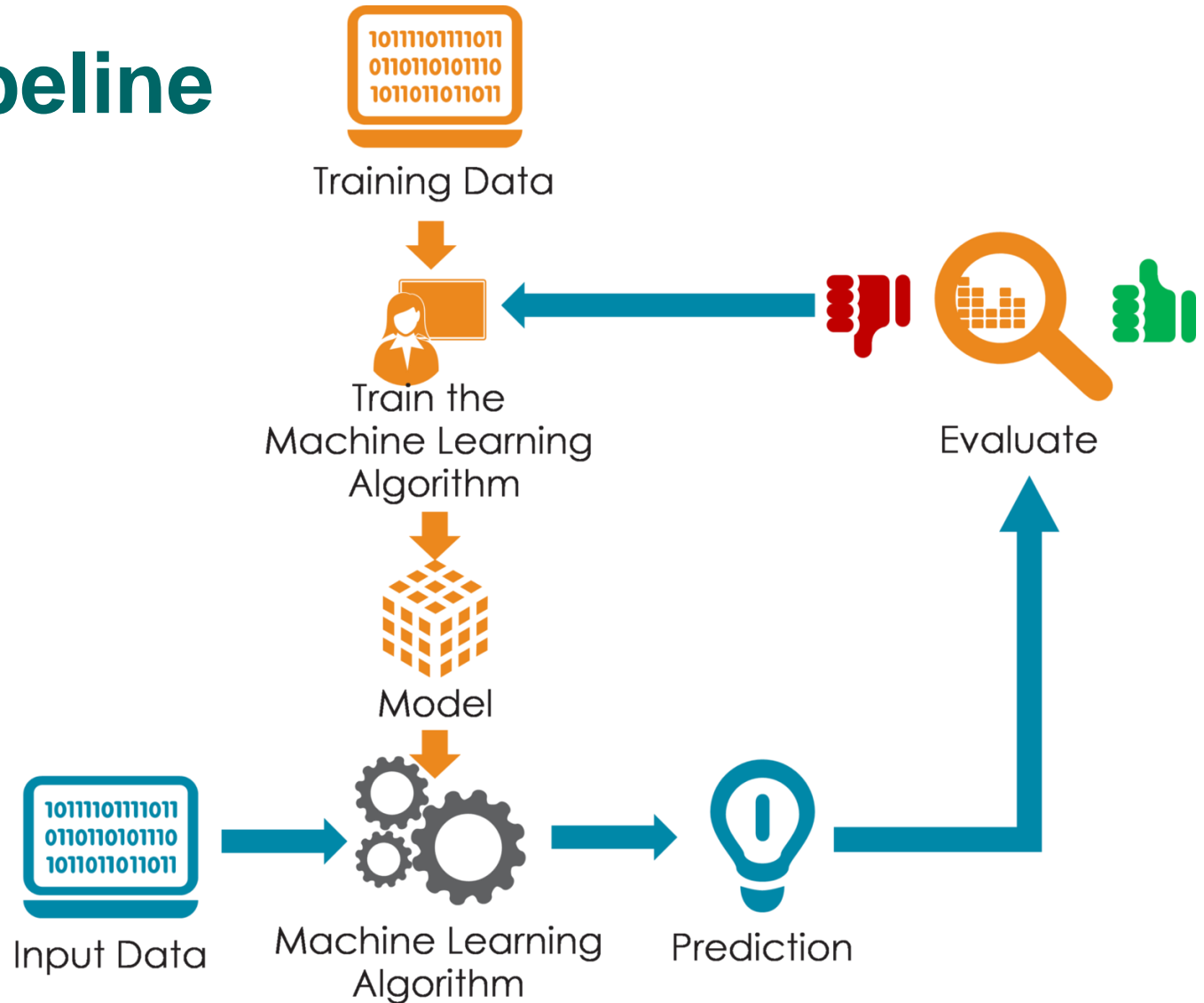
Hay múltiples **tipos** de atributos:

- **Real**: puede tomar cualquier valor dentro de un cierto rango. Ej. temperatura como un número real [grados].
- **Discreto**: Ej. horas de funcionamiento como un número natural.
- **Categorico**: Ej. color como {azul, rojo, amarillo}
  - Se puede pensar como '**discreto no ordenado**'.

# Resultados

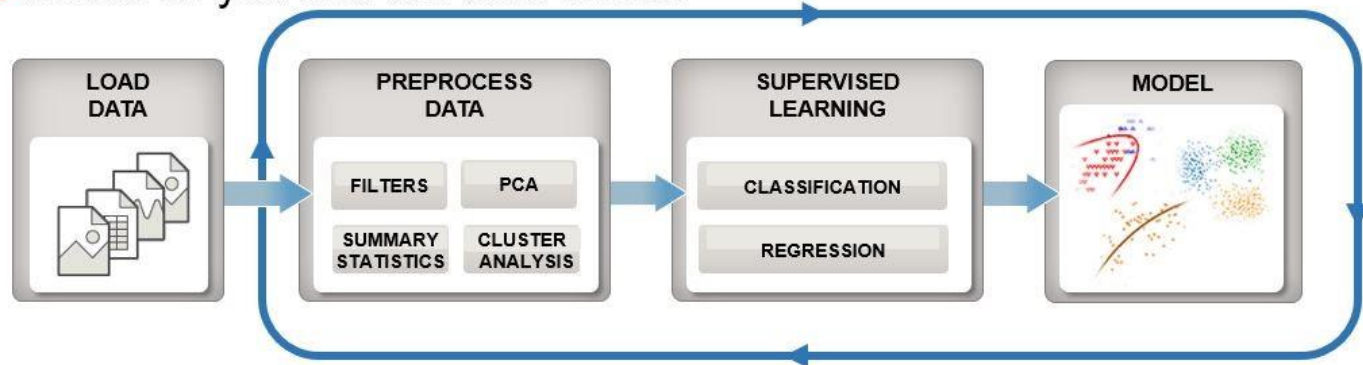
- Los modelos pueden ser de diversas formas:
  - Árboles de decisión.
  - Listas de reglas.
  - Redes neuronales.
  - Modelos bayesianos o probabilísticos.
  - Etc.

# AA pipeline

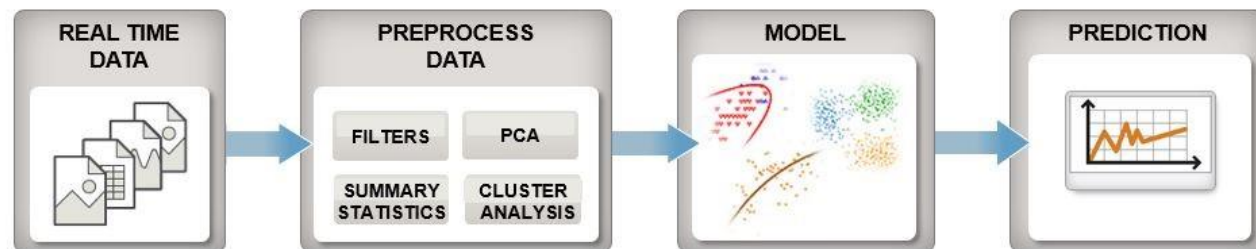


# AA pipeline

**Train:** Iterate till you find the best model



**Predict:** Integrate trained models into applications



# Cómo resolver un problema de ML?

- Identificar el problema
- Conseguir datos, muchos datos!
- Elegir un método adecuado (o varios)
- Entrenar varios modelos con el conjunto de entrenamiento, evaluarlos con el conjunto de validación
- Estimar el error con el conjunto de testeo

# Criterios de selección del modelo

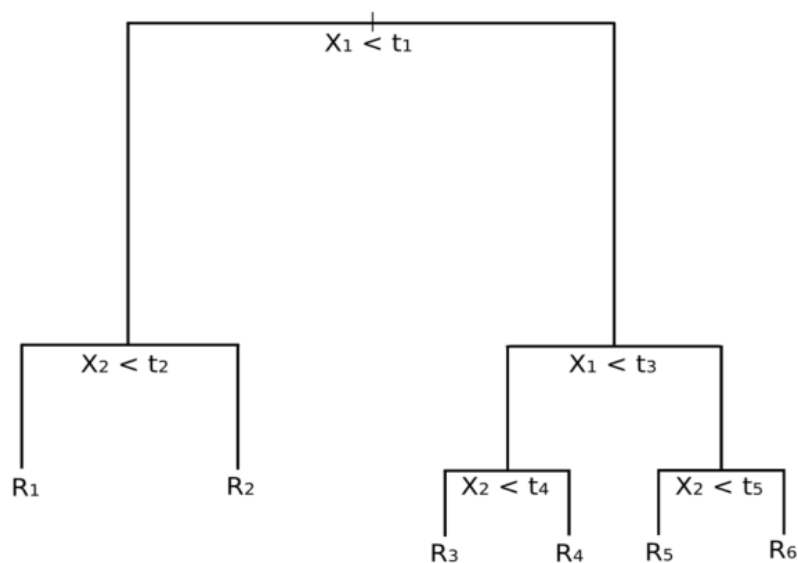
- Dos decisiones fundamentales:
  - El tipo de modelo (árboles de decisión, redes neuronales, modelos probabilísticos, etc.)
  - El algoritmo utilizado para construir o ajustar el modelo a partir de las instancias de entrenamiento (existen varias maneras de construir árboles de decisión, varias maneras de construir redes neuronales, etc.)

# ÁRBOLES DE DECISIÓN

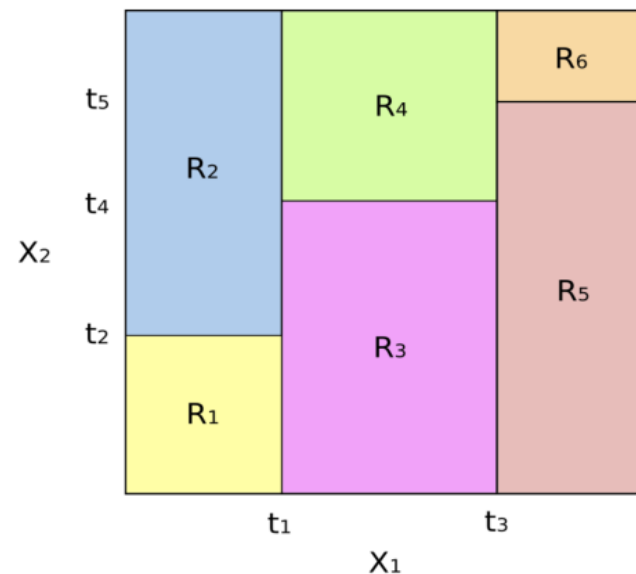
---

# Árboles de Decisión

- Cómo dividir una región en bloques...



A Decision Tree with six separate regions

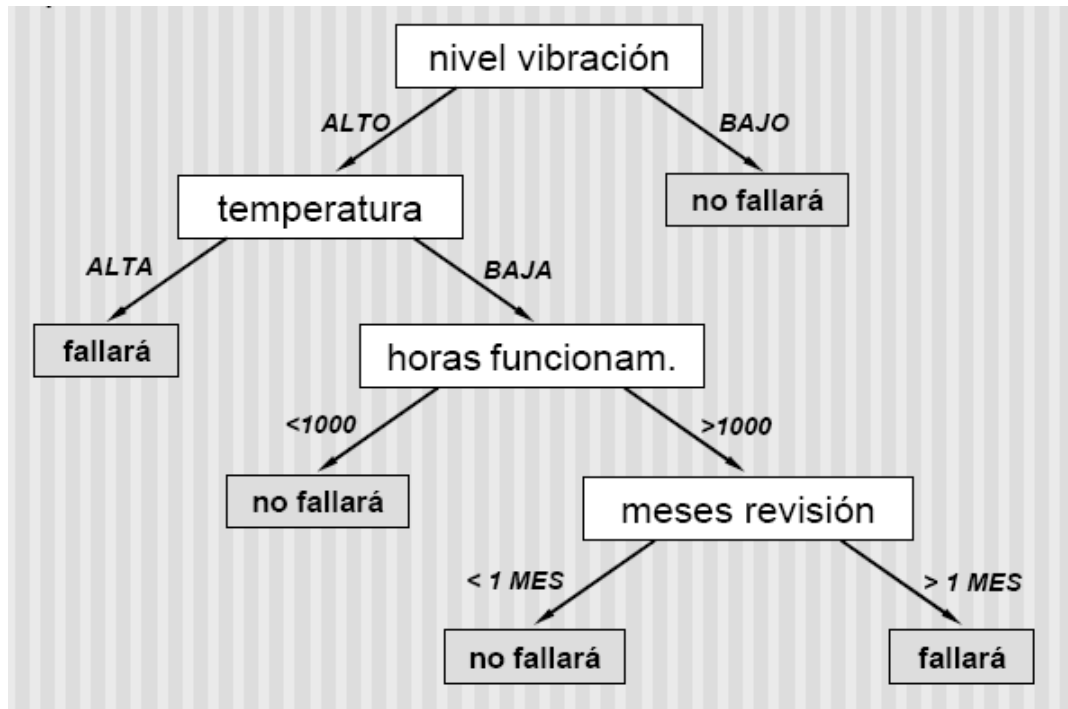


The resulting partition of the subset of  $\mathbb{R}^2$  into six regional "blocks"



# Árboles de Decisión

- Ejemplo: modelado de la posible falla de una máquina.



# Árboles de Decisión

- Compuestos de nodos y ramas.
- Representan reglas lógicas (if - then).
- Nodos internos = atributos (atributo-valor).
- Nodos hoja = clases.
- Nodo raíz = nodo superior del árbol.
- Objetivo en AA: Obtener un árbol de decisión (resultado) a partir de un conjunto de instancias o ejemplos.
- Bias: árbol mínimo

# Árboles de Decisión

- Ejemplo de un conjunto de entrenamiento.

Temperatura	Nivel de vibraciones	Horas de funcionamiento	Meses desde revisión	Probabilidad de fallo
ALTA	ALTO	< 1000	> 1 MES	fallará
BAJA	BAJO	< 1000	< 1 MES	no fallará
ALTA	BAJO	>1000	> 1 MES	no fallará
ALTA	BAJO	< 1000	> 1 MES	no fallará
BAJA	ALTO	< 1000	> 1 MES	no fallará
BAJA	ALTO	>1000	> 1 MES	fallará
ALTA	ALTO	< 1000	< 1 MES	fallará

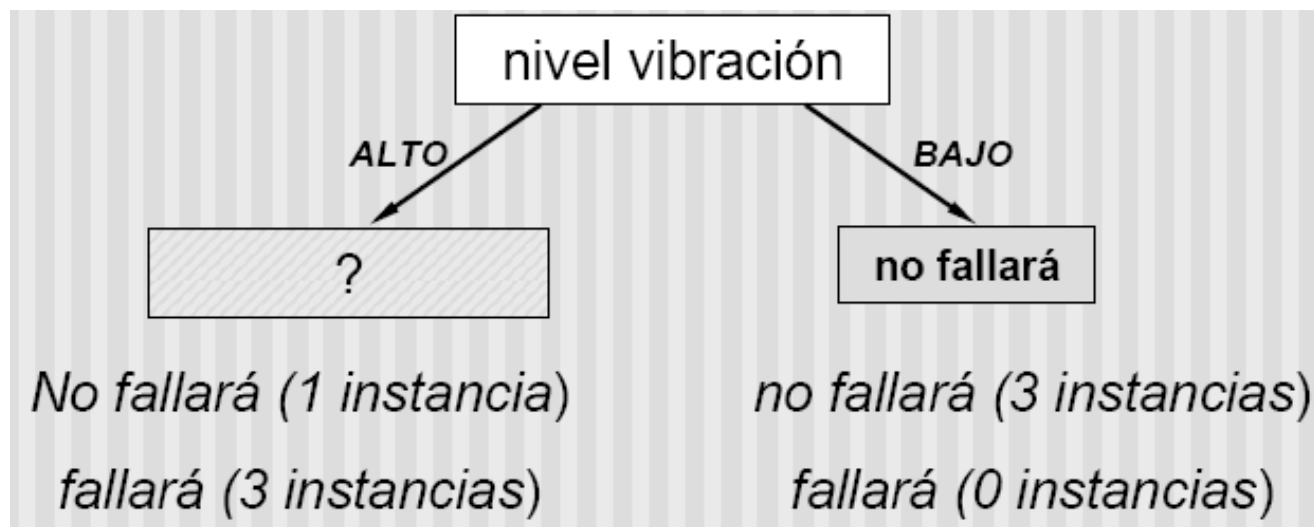
# Árboles de Decisión

Crearemos un árbol a partir de los ejemplos de entrenamiento anteriores. ¿Qué atributo elegir para el primer nodo?

ATRIBUTO	VALORES	CLASE	
		<i>fallará</i>	<i>no fallará</i>
Temperatura	Alto	2	2
	Bajo	1	2
Nivel de vibraciones	Alto	3	1
	Bajo	0	3
Horas defuncionamiento	< 1000	2	3
	>1000	1	1
Meses desde revisión	> 1 mes	2	3
	< 1 mes	1	1

# Árboles de decisión

- Árbol construido hasta el momento:



- ¿Qué atributo usamos en el siguiente nivel del árbol (rama izquierda)?

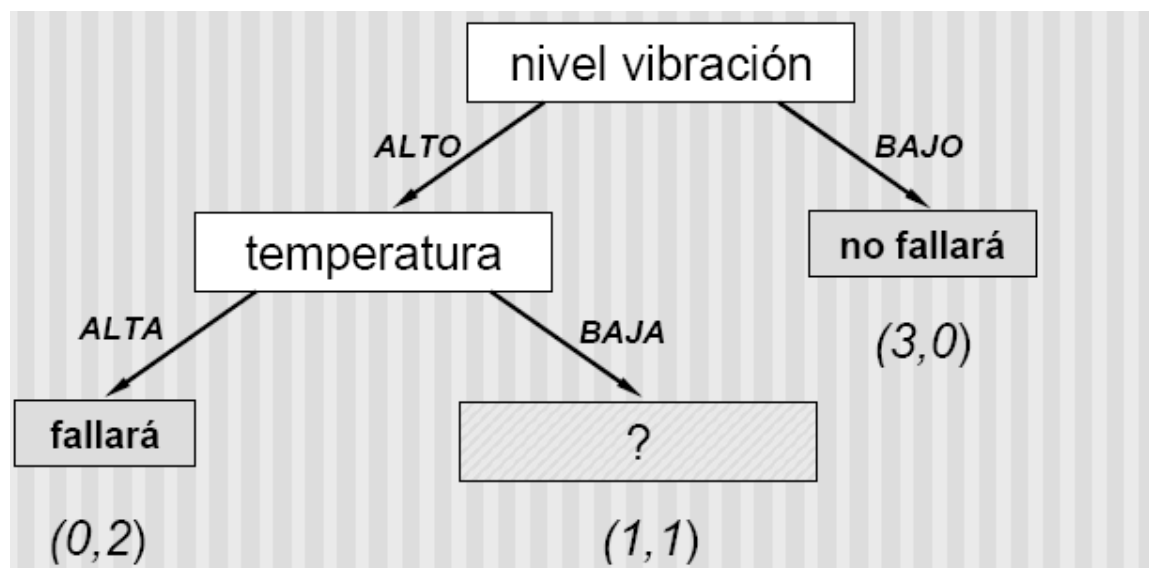
# Árboles de decisión

- Sólo aquellos ejemplos de entrenamiento que llegan al nodo (*nivel\_vibracion = alto*) se utilizan para elegir el nuevo atributo:

ATRIBUTO	VALORES	CLASE	
		<i>fallará</i>	<i>No fallará</i>
Temperatura	Alta	2	0
	BAja	1	1
Horas de funcionamiento	< 1000	2	1
	>1000	1	0
Meses desde revisión	> 1 mes	2	1
	< 1 mes	1	0

# Árboles de decisión

- Árbol construido hasta el momento:



- ¿Qué atributo usamos en el siguiente nivel del árbol (rama derecha)?

# Otros modelos

- Los árboles de decisión son sólo uno de los posibles modelos.
- Dependiendo de la aplicación se deberá elegir un modelo u otro.



# Selección del modelo y/o algoritmo

- Capacidad de representación.
- Legibilidad.
- Tiempo de cómputo on-line.
- Tiempo de cómputo off-line.
- Dificultad de ajuste de parámetros.
- Robustez ante el ruido.
- Sobreajuste.
- Minimización del error.