

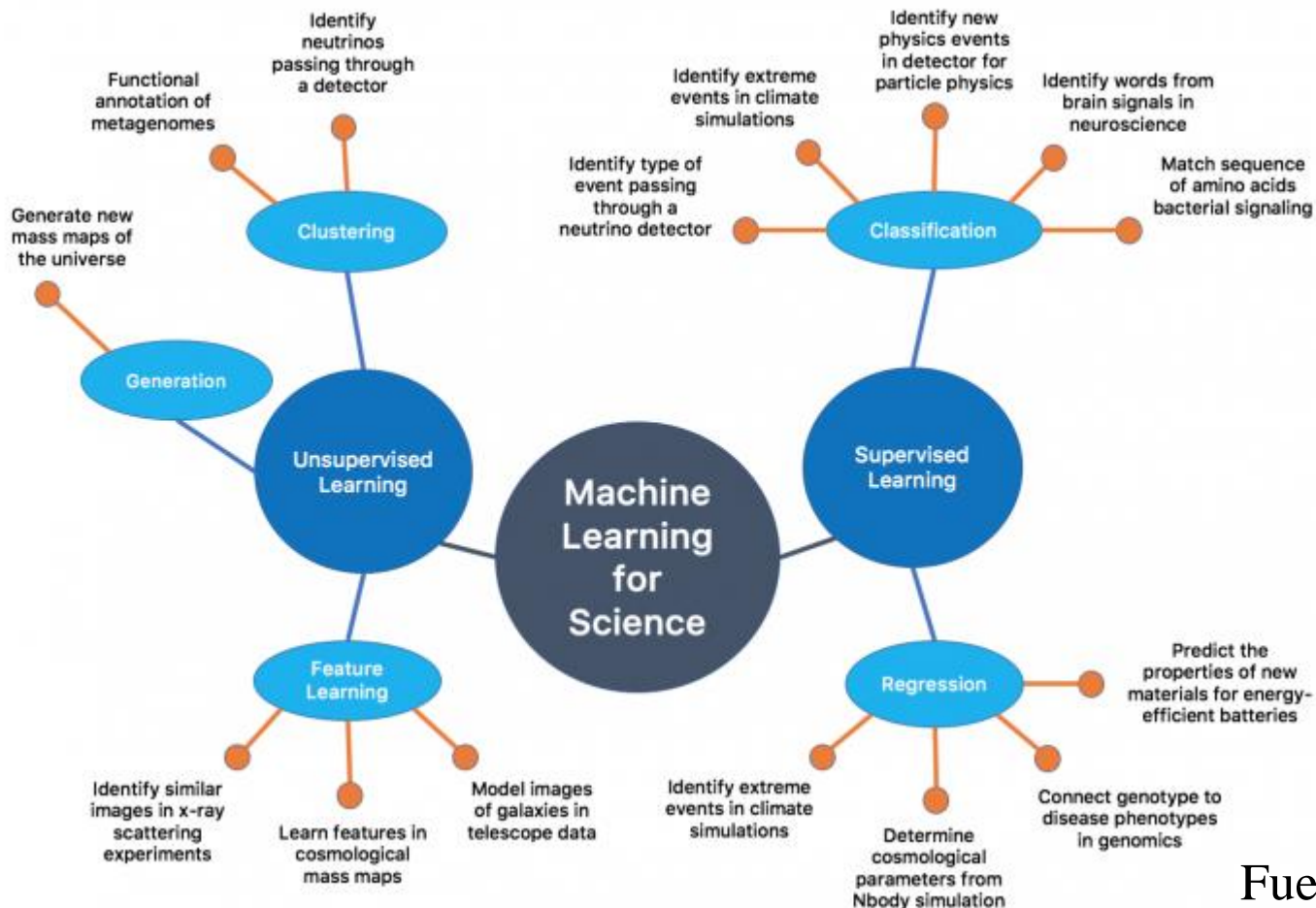
Aprendizaje Automatizado



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.**

Aprendizaje Automatizado



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

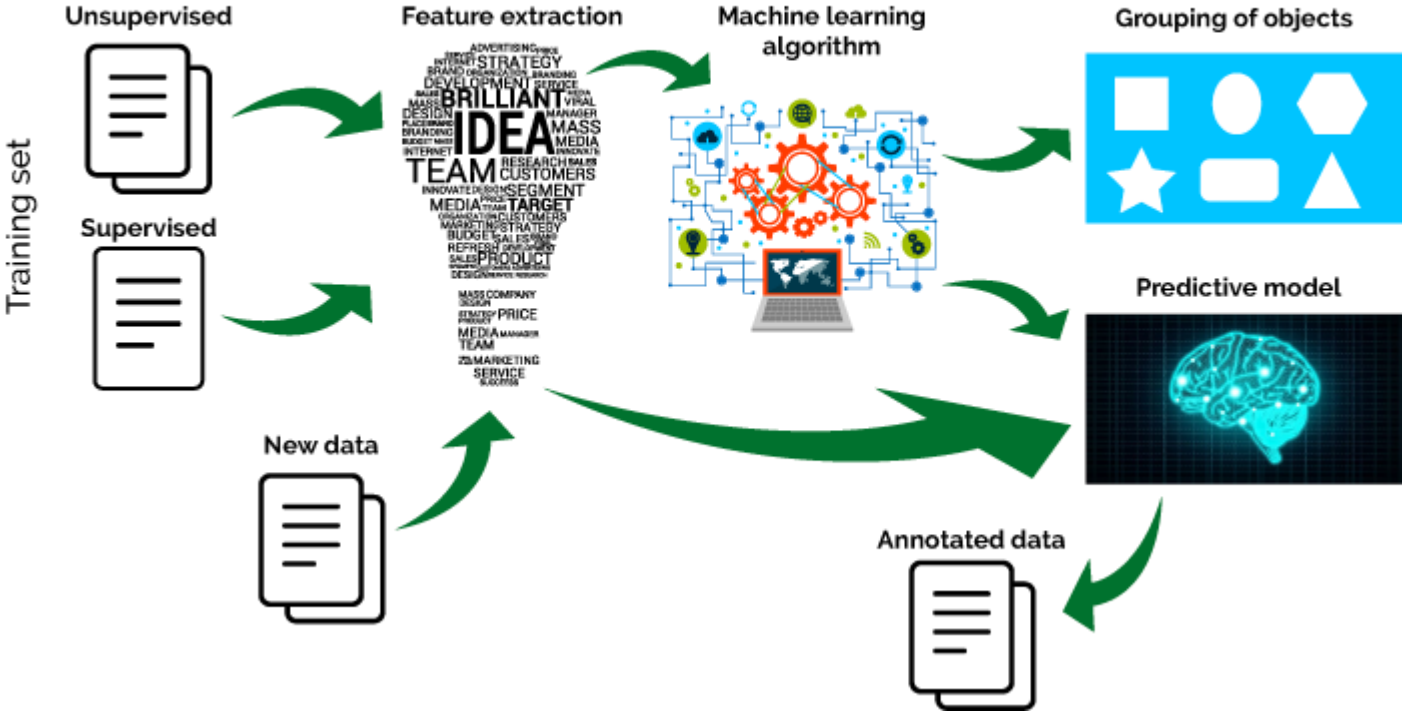
Aprendizaje Inductivo

- 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

Machine Learning



Definiciones

- **Resultado**: modelo que se infiere a partir de los ejemplos (también llamado **hipótesis**).
- **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 / la máquina 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.
 - Si *nivel_vibraciones = alto* Y *temp = alta* ENTONCES *fallará*.
 - Red Neuronal para clasificación

Atributos

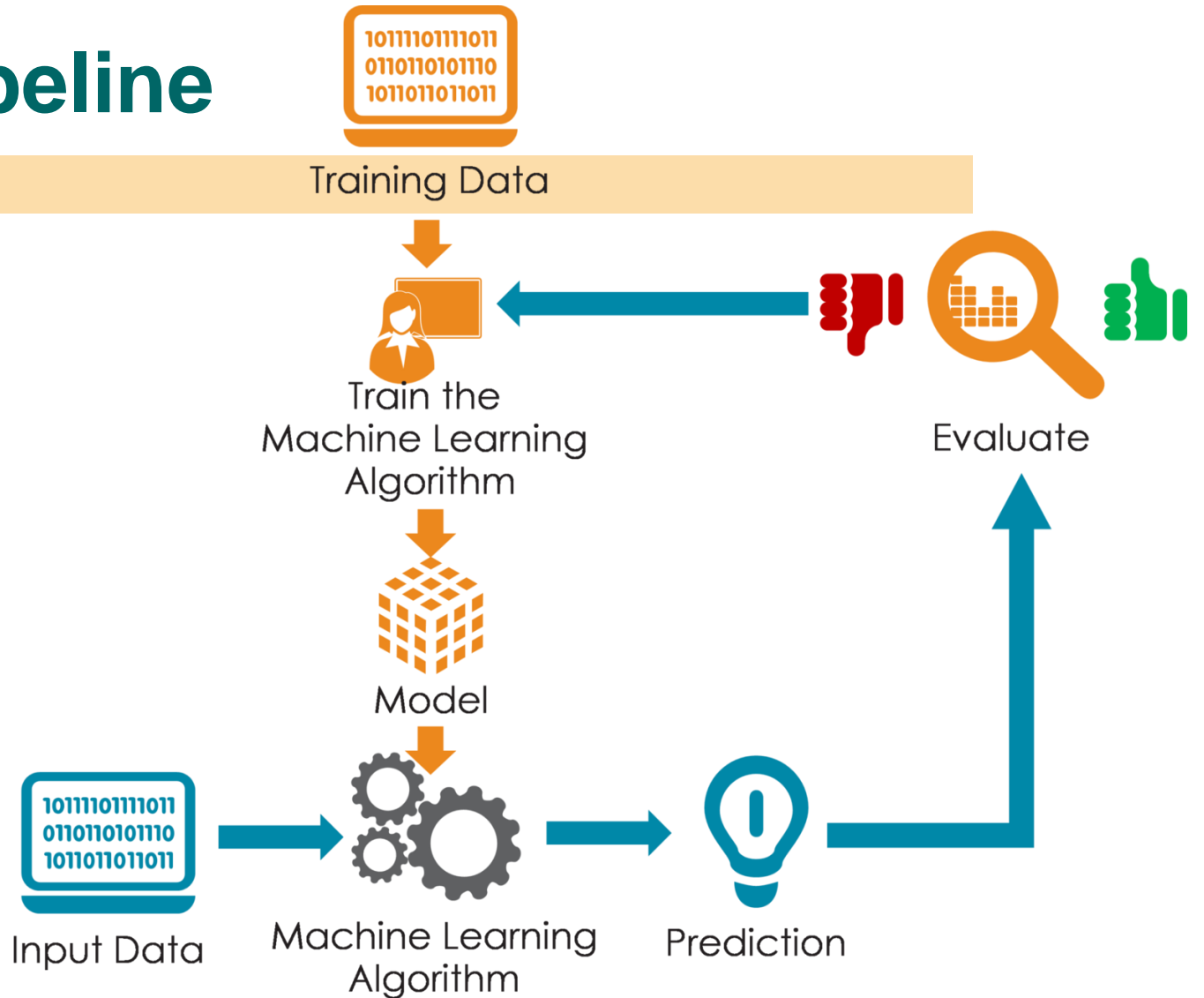
Hay múltiples **tipos** de atributos:

- **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

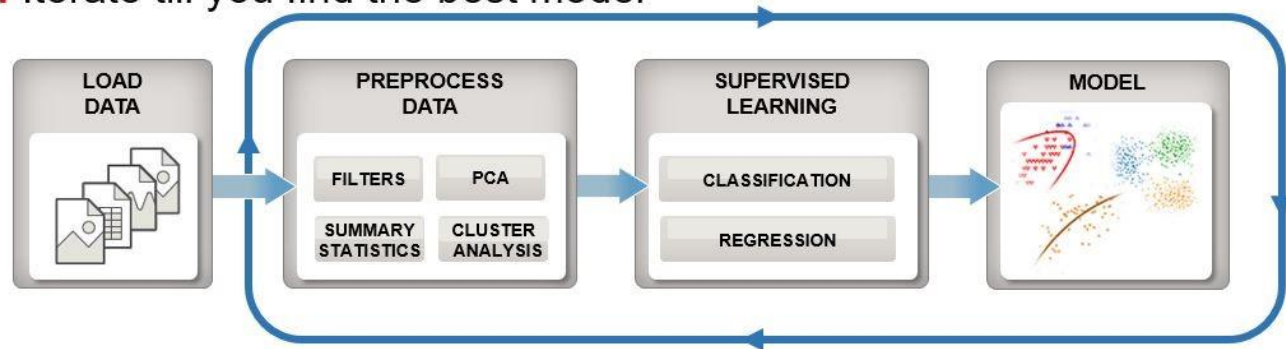
- Los **modelos** se pueden expresar de diversas formas:
 - Árboles de decisión.
 - Listas de reglas.
 - Redes neuronales.
 - Modelos bayesianos o probabilísticos.
 - Etc.

AA pipeline

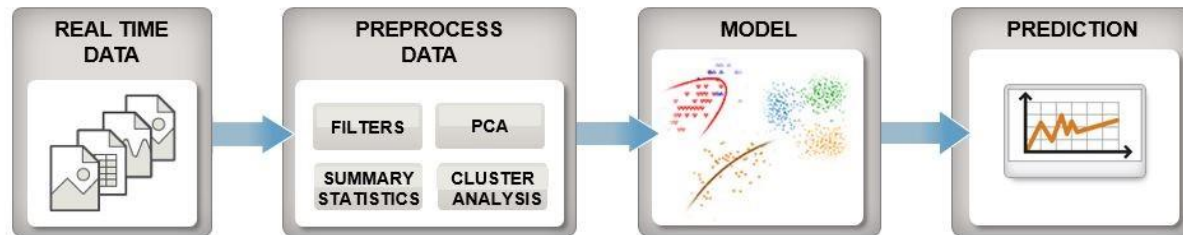


AA pipeline

Train: Iterate till you find the best model

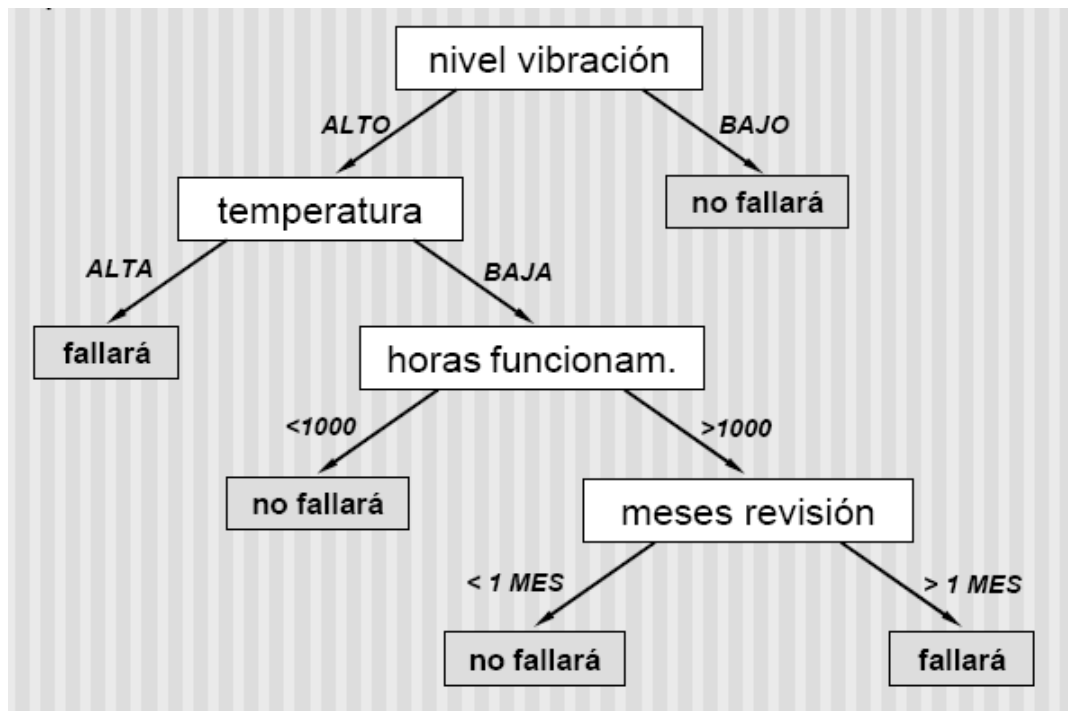


Predict: Integrate trained models into applications



Árboles de Decisión (I)

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



Árboles de Decisión (II)

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

Árboles de Decisión (III)

- 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

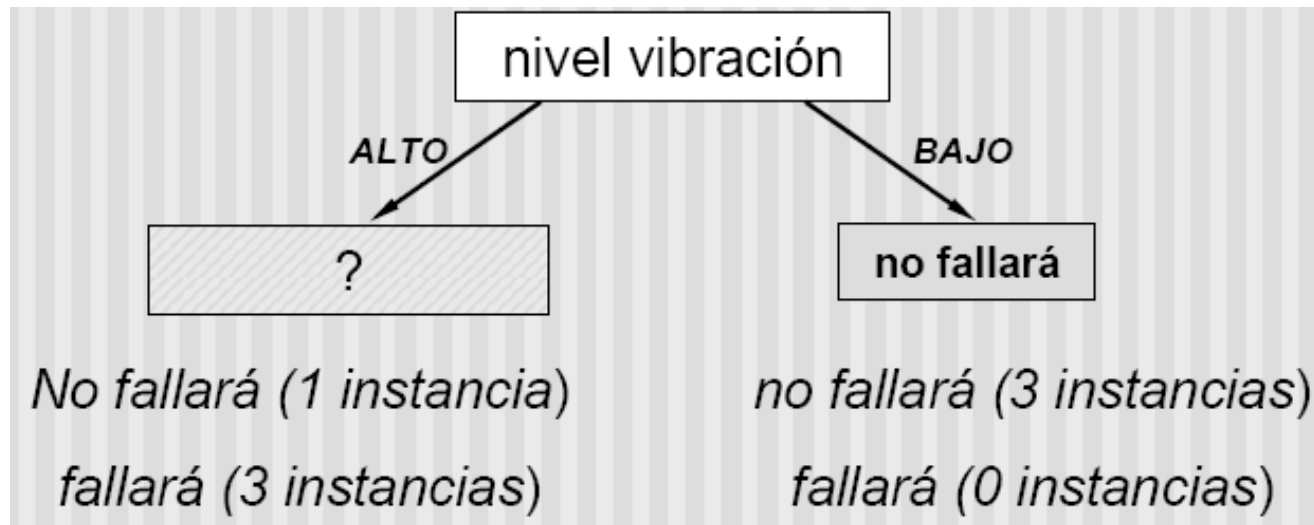
Árboles de Decisión (IV)

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 (V)

- Árbol construido hasta el momento:



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

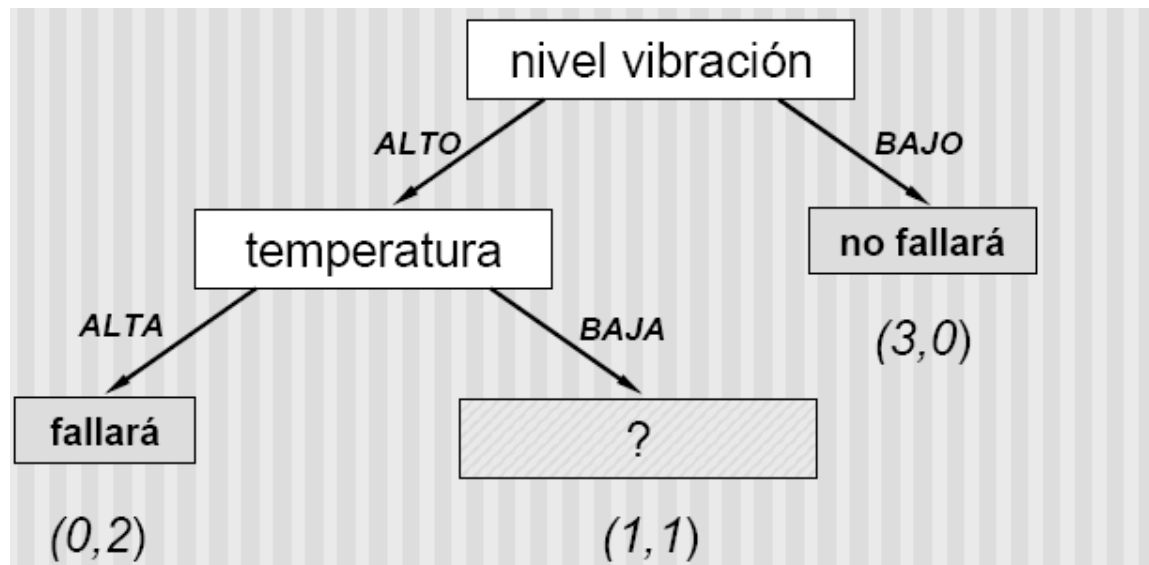
Árboles de decisión (VI)

- 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 (VI)

- 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.
- A continuación se indican algunos criterios para elegir modelos.

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.)

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.

Selección del modelo y/o algoritmo

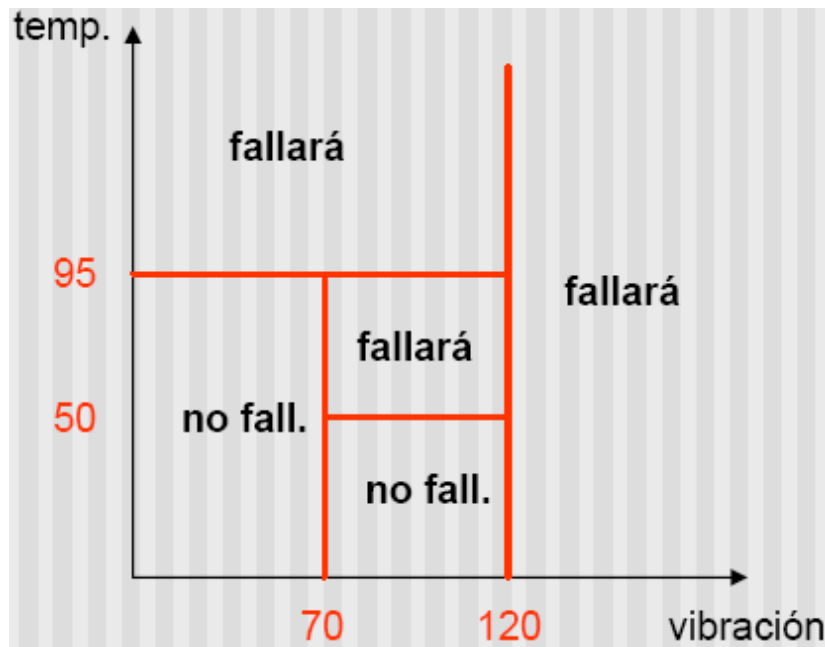
Capacidad de representación (I)

- Relacionado con el tipo de **fronteras de decisión** que se pueden expresar.
- **Fronteras de decisión**: separación de clases distintas.
- Cada modelo crea diferentes fronteras.

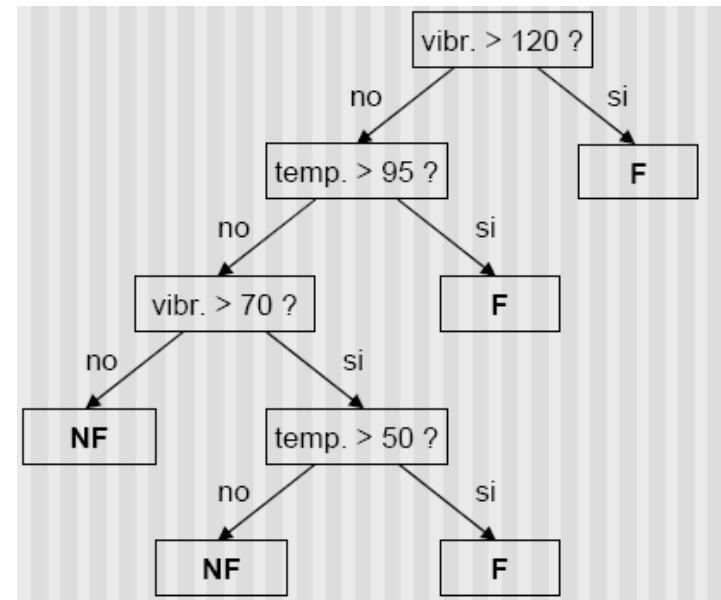
Selección del modelo y/o algoritmo

Capacidad de representación (II)

- Ejemplo con sólo dos atributos:

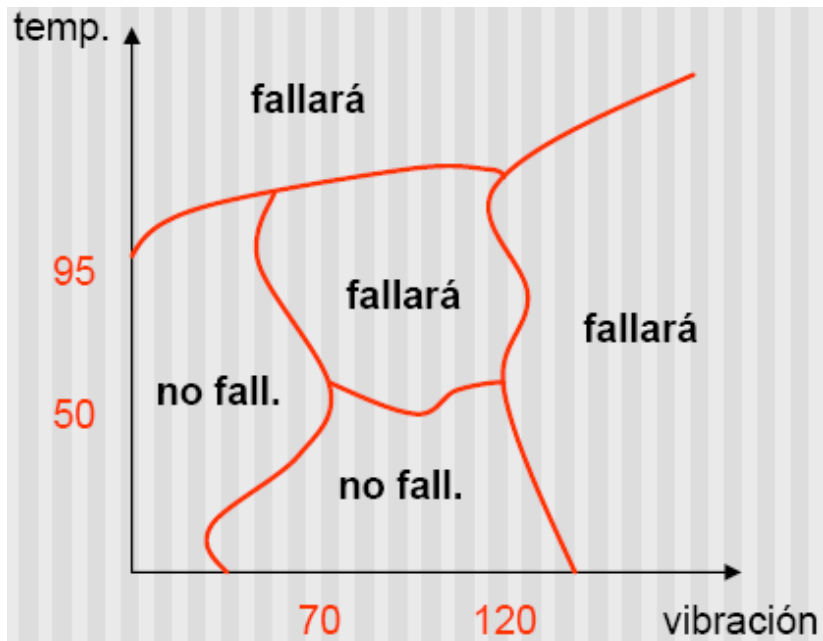


Árboles de decisión:
fronteras perpendiculares a
los ejes



Selección del modelo y/o algoritmo

Capacidad de representación (III)



Redes Neuronales (NN),
fronteras no lineales:

- Mayor capacidad de representación.
- Permiten representar conceptos más complejos que los árboles de decisión.
- Se estudiarán más adelante.

Selección del modelo y/o algoritmo

Legibilidad (I)

- Capacidad de ser leído e interpretado por un humano.
- **Árboles de decisión**: fáciles de entender e interpretar:
 - conjunto de reglas.
 - en los niveles más altos están los atributos más importantes.
- **Redes neuronales**: difíciles (o imposibles) de interpretar: - pesos de conexiones entre neuronas.

Selección del modelo y/o algoritmo

Legibilidad (II)

- Un modelo legible puede **ofrecer información** sobre el problema que se estudia (ej. indicar qué atributos afectan la probabilidad de fallo de una máquina y cómo).
- Un modelo no legible sólo puede ser usado como un **clasificador** (ej. permite predecir si una máquina fallará o no aplicando el modelo).

Selección del modelo y/o algoritmo

Tiempo de cómputo on-line (I)

- Es el tiempo necesario para clasificar una instancia:
 - **Árboles de decisión**: tiempo necesario para recorrer el árbol, evaluando las funciones lógicas de cada nodo.
 - **Redes neuronales**: tiempo necesario para realizar las operaciones (sumas, productos, sigmoides) incluidas en la red.

Selección del modelo y/o algoritmo

Tiempo de cómputo on-line (II)

- Este tiempo se consume cada vez que se debe clasificar una nueva instancia.
- Algunas aplicaciones requieren clasificar miles de instancias.
 - Ejemplo: clasificación de cada uno de los píxeles de una imagen área de un cultivo, río, ruta, etc.
 - Se requiere clasificar millones de instancias.
 - El tiempo de cómputo es muy importante.

Selección del modelo y/o algoritmo

Tiempo de cómputo off-line (I)

- Es el tiempo necesario para construir o ajustar el modelo a partir de los ejemplos de entrenamiento.
 - **Árboles de decisión**: tiempo necesario para elegir la estructura del árbol, los atributos a situar en cada nodo y la optimización mediante la poda.
 - **Redes neuronales**: tiempo necesario para ajustar los pesos de las conexiones (puede tomar valores muy grandes).

Selección del modelo y/o algoritmo

Tiempo de cómputo off-line (II)

- Sólo se consume una vez, cuando mediante la utilización de los ejemplos de entrenamiento se genera y selecciona el resultado (modelo o hipótesis) más adecuado.
- Dependiendo de la aplicación no es un problema que el tiempo de cómputo off-line sea elevado (se deja una computadora procesando uno o tres días enteros).

Selección del modelo y/o algoritmo

Dificultad de ajuste de parámetros

- Se prefieren los algoritmos con pocos (o ninguno) parámetros que ajustar.
- Se prefieren algoritmos con muy poca sensibilidad a la modificación de sus parámetros.
- Hay modelos muy difíciles de ajustar mediante parámetros (puede ocurrir con redes neuronales).

Selección del modelo y/o algoritmo

Robustez ante el ruido

- Instancia de entrenamiento ruidosa:
 - etiquetada incorrectamente (ejemplo: una máquina que no falló, etiquetada como que sí falló).
 - algún atributo no está valorizado.
- Algunos algoritmos pueden funcionar adecuadamente aunque haya instancias ruidosas en el conjunto de entrenamiento (ej. árboles de decisión, redes neuronales).
- Otros algoritmos no ofrecen buenos resultados (ej. k-vecinos más cercanos).

Selección del modelo y/o algoritmo

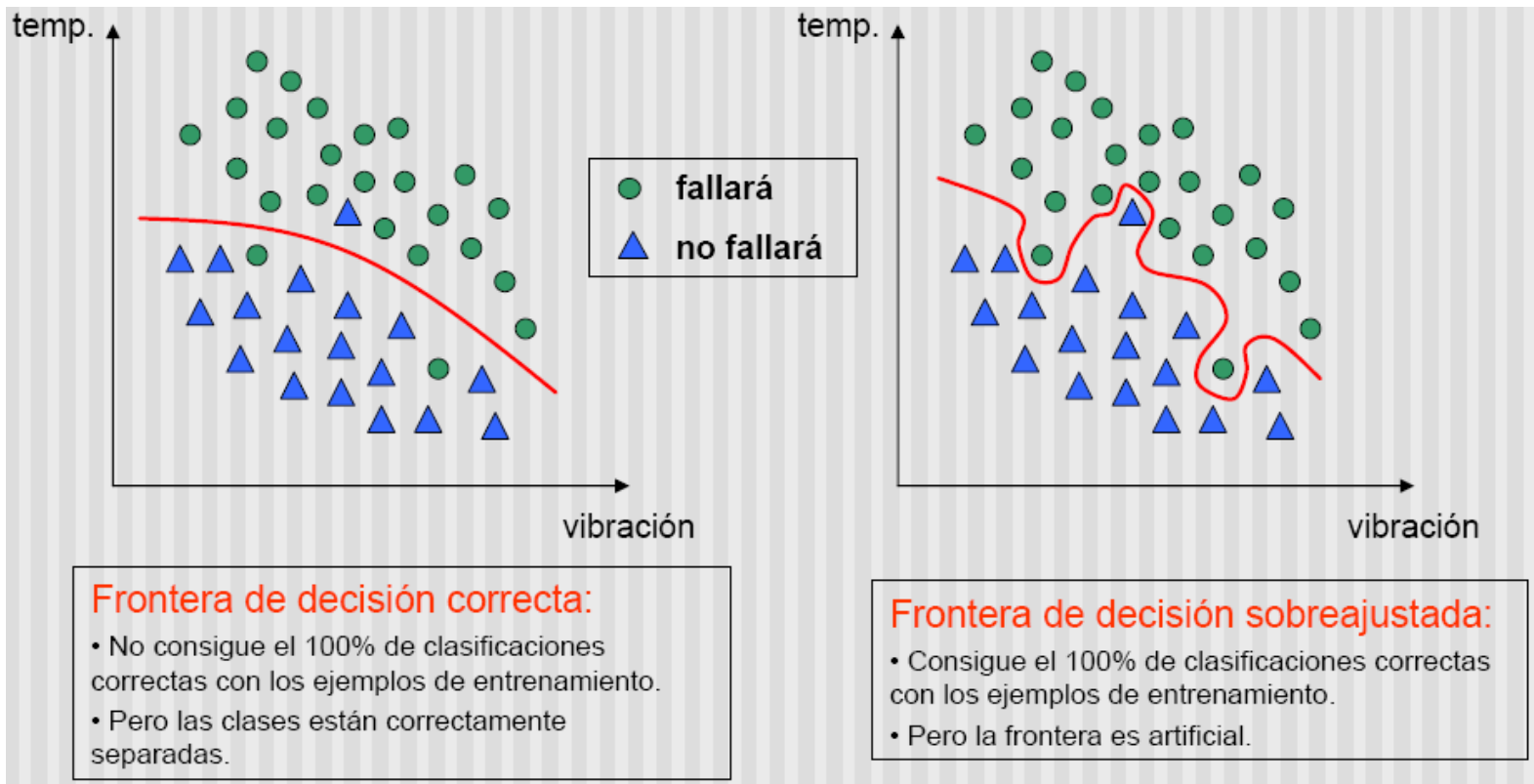
Sobreajuste (*overfitting*). (I)

- Problema muy común.
- El modelo está demasiado ajustado a las instancias y no funciona adecuadamente con nuevos casos.
- El modelo no es capaz de **generalizar**.
- Normalmente, fronteras de decisión muy complejas producen sobreajuste.

Selección del modelo y/o algoritmo

Sobreajuste (*overfitting*). (I)

- Ejemplo con dos atributos:



Selección del modelo y/o algoritmo

Minimización del error

Ejemplo: Resultados arrojados para el problema de clasificar los datos de Contraceptive Method Choice (CMC), con diferentes métodos.

	Árboles de decisión	Redes neuronales	Naive-Bayes con normales	Naive-Bayes con histogramas	k-primeros vecinos
Medias	26.41	23.711339	33.9851806	25.4706348	26.094287
Desviaciones estándar	2.96739691	4.40077215	4.67207077	3.70521908	3.50755602

Selección del modelo y/o algoritmo

- ¿Cuáles de los criterios anteriores se relacionan con la selección del modelo?
- ¿Cuáles de los criterios anteriores se relacionan con la selección del algoritmo?
- ¿Y con ambos?
- ¿Cuáles de los criterios anteriores se relacionan entre sí?

Bibliografía

- Machine Learning - Tom Mitchell – McGrawHill
- Curso de doctorado "Aprendizaje Automatizado y Data Mining" Grupo de Ingeniería de Sistemas y Automática (Universidad Miguel Hernández)
<http://isa.umh.es/asignaturas/aprendizaje/index.html>