Informática y comunicaciones
ies azarquiel

**Curso de especialización inteligencia artificial y big data**

# Capitulo 2: Desarrollo de Sistemas de Aprendizaje Automático

Profesor: Sebastián Rubio

Septiembre 2025

**Sistemas de aprendizaje automático**

# 1 FASES EN EL DESARROLLO DE UN SISTEMA DE APRENDIZAJE AUTOMÁTICO

## 1.1 FASES

A continuación vamos a dar la relación de puntos más importantes en el flujo de trabajo de proyectos relacionados con Machine Learning. Cada uno de estos puntos se desarrollan en diferentes módulos del curso

- 1.- Enmarcar el problema
- 2.- Obtener los datos
- 3.- Explotar y visualización de los datos.
- 4.- Preparación de los datos para el modelo
- 5.- Seleccionar el modelo y entrenarlo
- 6.- Perfeccionar el modelo
- 7.- Presentar solucionar.
- 8.- Lanzar, monitorizar y mantener el sistema

# 2 ESTRUCTURA

Los datos que vamos usar en el mdelo pueden estar etiquetado o no. Cuando no exista el etiquetado previo (conocido con el nombre de aprendizaje no supervisado) se agrupan las observaciones basándose en criterios de semejanza. Una vez que se han obtenido nuevas descripciones se establecen mecanismos de evaluación o valoración, de tal forma que sólo estarán disponibles aquellos que pasen cierto umbral de corrección.

## 2.1 Instancias u observaciones

Los datos utilizados durante el aprendizaje por cualquier estrategia recibe diferentes nombres: muestras, instancias, vectores característicos, o simplemente ejemplos. El conjunto de muestras necesarios para el entrenamiento va a depender del espacio de hipótesis, de forma que cuanto mayor sea, mayor será necesario el conjunto de muestras necesarias. El conjunto de muestras debe ser representativo de modo que los atributos deben ser relevantes, aunque esta tarea también puede ser realizada por

un módulo externo. En la mayoría de las tareas de aprendizaje se procura que los ejemplos tratados sean representativos del concepto buscado. El problema de seleccionar atributos relevantes puede considerarse como una labor de ingeniería, aunque también puede ser una tarea realizada por otras técnicas de aprendizaje.

## 2.2 Representación

La representación de las entradas es clave en cualquier problema de aprendizaje,es decir hay que codificarlas en un lenguaje concreto. Esto puede ser simbólica o numérica.

- Clases: Hay que determinar que se quiere aprender y qué posibles valores puede tomar:Ejemplo
- Atributos:Los atributos con los que se va a caracterizar lo que quiere aprender
- Valores de los atributos: Un mismo atributo puede tomar diferentes valores.

La inclusión u omisión de ciertos atributos o valores condiciona fuertemente lo que se va aprender, la corrección y la complejidad del método aplicado.

## 2.3 Etiquetado

Cada instancia que se toma para el aprendizaje se le debe asignar una clase. A esto se le conoce como etiquetado. En los sistemas aprendices,son capaces de aprender de los gustos, preferencias o necesidades de un usuario final a partir de la mera observación de aquéllos en la realización de una tarea. En todos los casos en que exista una clasificación o etiquetado previo de instancias, ya sea por otra persona, programa o entidad, se habla de aprendizaje supervisado. Es decir los ejemplos están previamente etiquetados como instancias positivas o negativas del concepto objetivo. En el aprendizaje por descubrimiento (no supervisado) no se dispone de la clase a la que pertenece cada instancia.

## 2.4 Fuente de los ejemplos

Se distinguen dos tipos principales de fuentes:

- Fuente Externa:En este caso, existe una entidad externa al sistema de aprendizaje que proporciona los ejemplos ya etiquetados. Esta entidad actúa como una guía o ejemplo a seguir, y el sistema aprende clasificando los ejemplos presentados. Se menciona que este es el escenario típico de los llamados "sistemas aprendices".

- Ausencia de Fuente Externa: Cuando no existe una fuente externa que realice el etiquetado previo, se plantea una alternativa donde el propio sistema de aprendizaje, a través de un módulo interno de agrupamiento (clustering), se encarga de construir dicha clasificación. Esta solución requiere un criterio para discriminar las acciones deseables en tareas muy relacionadas.

La fuente de los ejemplos, ya sea externa o construida internamente, puede no considerar la ventaja relativa de proporcionar cierto tipo de ejemplos al sistema que aprende. Esto se debe a la incertidumbre inherente al proceso de aprendizaje y está relacionado con la capacidad expresiva del lenguaje de hipótesis y el conocimiento interno del nivel de corrección de lo aprendido.

## 2.5 Preprocesamiento

El preprocesamiento se define como un paso necesario cuando los ejemplos o datos no vienen descritos en un lenguaje apropiado para su tratamiento directo por los algoritmos de aprendizaje. Los puntos clave que se resaltan son:

- Necesidad del Preprocesamiento:Muchos conjuntos de datos requieren una etapa previa de transformación para que puedan ser utilizados eficazmente por los algoritmos de aprendizaje. Esto implica adecuar la representación de los ejemplos para que el sistema de aprendizaje pueda obtener una mejor respuesta al problema aplicado. La selección, organización y adecuación de las instancias asociadas a las entradas de la estrategia de aprendizaje se conoce como preprocesamiento.

- Unificación y Limpieza de Datos: El preprocesamiento se refiere principalmente a la unificación de la descripción de los datos y a los procesos de eliminación del ruido y errores presentes en ellos. En el contexto del Descubrimiento de Conocimiento en Bases de Datos (KDD) y la limpieza de datos (data cleaning), el objetivo es unificar la gran variedad de datos y de fuentes, así como resolver el problema de asignar una única nomenclatura para todos los datos, representar y tratar la forma unívoca los datos ausentes, el ruido y los errores.

- Tratamiento de Atributos con Muchos Valores: Un problema frecuente en el preprocesamiento es el tratamiento de atributos con un gran número de valores (a menudo inconmensurables), como el salario de los empleados. Para abordar esto, se sugiere agrupar los valores en rangos más reducidos (discretización), especialmente si la distinción producida por valores individuales no tiene una significación relativa con respecto al objetivo marcado. Esta técnica permite aplicar algoritmos para obtener patrones de discriminación adecuados.

- Tratamiento de Atributos Continuos: Además de la discretización mencionada anteriormente, otra solución para los atributos de valores continuos, utilizada en algunos algoritmos como C4.5, consiste en establecer dinámicamente (para cada atributo y en cada caso) el umbral que mejor discrimine el conjunto de ejemplos con respecto al valor de clase. Esto implica encontrar el punto de corte óptimo dentro del rango de valores continuos para separar los ejemplos de manera efectiva.

- Selección de Atributos Relevantes: En la mayoría de las tareas de aprendizaje, se busca que los ejemplos tratados sean representativos del concepto que se quiere aprender. Esto implica que los ejemplos deben ser similares entre sí y descritos por un conjunto de atributos que se espera sean relevantes para el objetivo de la tarea. Es decir, se buscan atributos que puedan formar parte de la descripción asociada a dicho objetivo. El problema de seleccionar cuáles son los atributos más importantes puede considerarse un problema de ingeniería (la forma en que se describe el problema), aunque también puede ser una tarea realizada por otras técnicas de aprendizaje automático.

En resumen, esta parte del texto profundiza en las estrategias para manejar atributos con valores continuos durante el preprocesamiento, destacando la discretización y la búsqueda dinámica de umbrales. Además, introduce la importancia de la selección de atributos relevantes para asegurar que los modelos de aprendizaje se basen en información significativa para la tarea en cuestión, señalando que esta selección puede ser un proceso de ingeniería o una tarea aprendida por otros algoritmos. En resumen, el preprocesamiento en IA es una etapa crucial que busca preparar y limpiar los datos brutos para que los algoritmos de aprendizaje puedan trabajar de manera más eficiente y efectiva, eliminando inconsistencias, ruido y transformando los datos a un formato adecuado para el análisis y la modelización.