3

COMPRENSIÓN DE LOS PRINCIPALES CONCEPTOS SOBRE LA CIENCIA DE DATOS E IA

Hoy en día, los datos no solo se almacenan: se analizan, se interpretan y, sobre todo, se utilizan para tomar decisiones inteligentes. Este tercer capítulo abre la puerta a la ciencia de datos y a la inteligencia artificial, mostrando cómo se conectan con el Big Data para obtener valor real a partir de grandes volúmenes de información.

Se explica qué son los algoritmos supervisados y no supervisados, cómo funciona el aprendizaje profundo (deep learning) y qué papel juega el procesamiento de textos e imágenes. También se presentan herramientas de visualización de datos que permiten comunicar resultados de forma comprensible y visual.

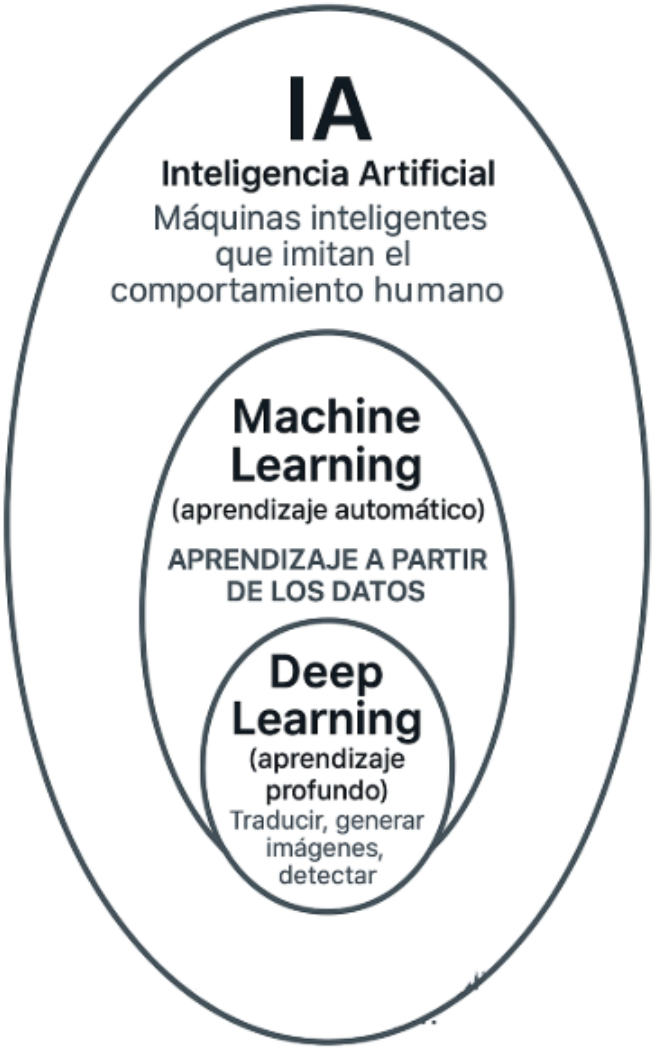
3.1 INTRODUCCIÓN A LA "CIENCIA DE DATOS" Y LA INTELIGENCIA ARTIFICIAL

La **ciencia de datos** es una disciplina que busca **extraer valor y conocimiento útil a partir de grandes cantidades de información**. No se trata solo de mirar números o hacer estadísticas, sino de combinar distintas técnicas —como el análisis de datos, la programación y el conocimiento del contexto— para obtener respuestas que ayuden a entender mejor lo que ocurre y a tomar decisiones más acertadas. En un mundo donde casi todo lo que hacemos genera datos (desde las compras que realizamos hasta lo que compartimos en redes sociales), tener personas y sistemas capaces de interpretar esa información se ha convertido en una necesidad en muchos sectores.

Esta disciplina ha crecido mucho gracias al **Big Data**, ya que ahora disponemos de datos en cantidades enormes y de todo tipo: texto, imágenes, audios, señales de sensores, etc. Al haber tanta información disponible y tan variada, la ciencia de datos ha encontrado el escenario perfecto para desarrollarse. Cuanto más volumen y variedad hay, más herramientas hacen falta para ordenar, limpiar, analizar y visualizar esa información de forma útil.

Dentro de este contexto también aparece la **Inteligencia Artificial (IA)**, que es el conjunto de tecnologías capaces de realizar tareas que, hasta hace poco, estaban reservadas solo a las personas. Por ejemplo, entender un texto, reconocer una cara en una foto, o responder a una pregunta con sentido. En otras palabras, **la IA busca imitar la forma en que los humanos pensamos, decidimos o aprendemos**, aunque lo haga con otros métodos.

Es común escuchar términos como IA, **Machine Learning** y **Deep Learning**, y aunque están relacionados, no significan lo mismo. La IA es el campo más general, que abarca todo lo que tenga que ver con que una máquina se comporte de forma inteligente. Dentro de la IA está el **Machine Learning (aprendizaje automático)**, que es la parte que enseña a las máquinas a aprender a partir de los datos, sin necesidad de que se les programe cada detalle. Y dentro del Machine Learning está el **Deep Learning (aprendizaje profundo)**, que utiliza redes neuronales con muchas capas para resolver tareas aún más complejas, como traducir idiomas, generar imágenes o detectar emociones en una voz.



Hoy en día convivimos con la inteligencia artificial sin darnos cuenta. Por ejemplo, cuando pedimos algo a **un asistente virtual como Alexa o Siri**, estamos hablando con un sistema que entiende nuestra voz y responde usando modelos entrenados con miles de datos. Cuando **una plataforma como Netflix o Spotify nos sugiere una serie o canción**, lo hace analizando nuestro comportamiento y comparándolo con el de otros usuarios. O cuando **el correo electrónico detecta automáticamente un mensaje como spam**, también está aplicando IA, basada en patrones aprendidos de millones de correos anteriores.

Saber más

La Estrategia de Inteligencia Artificial aprobada por el Gobierno en 2024 señala que la implantación de esta tecnología en el tejido empresarial español ha crecido de forma considerable en los últimos dos años. Según los datos manejados por el Ministerio de Transformación Digital, un 11,8 % de las empresas con diez o más empleados ya estaban utilizando alguna solución de IA.

Sin embargo, estos datos también muestran una brecha tecnológica evidente en función del tamaño de las empresas. Así, mientras que el 41,2 % de las grandes empresas ya aplicaban la inteligencia artificial, el porcentaje bajaba al 20 % en las medianas, y apenas llegaba al 9,4 % en las pequeñas.

Las cifras publicadas por el Instituto Nacional de Estadística en el primer trimestre de 2024 coincidían bastante con las del ministerio, aunque permiten ir un paso más allá en el análisis. Por ejemplo, se observa que la presencia de la IA es más notable en el sector de las tecnologías de la información y la comunicación, donde el 45 % de las empresas la usan. En cambio, este porcentaje baja al 16 % en servicios, al 10 % en industria y al 4 % en construcción.

Entre las empresas que ya han incorporado soluciones de IA, casi la mitad (46 %) han optado por adquirir productos ya desarrollados y listos para su uso. Un 33 % prefirió contratar a proveedores externos para que les desarrollaran o personalizaran sistemas a medida. En cuanto a las funcionalidades concretas que emplean, destaca el análisis del lenguaje escrito (presente en el 45 %), la automatización de tareas y apoyo en decisiones (39 %) y la generación automática de texto o voz (38 %).

Respecto al grupo de empresas (7 %) que habían valorado usar IA pero no llegaron a implementarla, la mayoría (79 %) lo achacó a la falta de personal con conocimientos adecuados. Además, un 49 % expresó dudas sobre los posibles riesgos legales y un 48 % reconocía tener problemas con la calidad o disponibilidad de los datos necesarios.

En cuanto a los estudiantes, el informe *"Inteligencia artificial y empleabilidad del futuro"*, elaborado por GAD3 para Planeta Formación y Universidades, ofrece datos interesantes. Entre los 800 estudiantes universitarios de entre 18 y 35 años que participaron en el estudio, el 65 % declaró haber utilizado herramientas de inteligencia artificial como usuarios. Más de la mitad (53 %) decía entender cómo y en qué contextos podían usarse estas tecnologías, y un 28 % afirmaba saber desarrollarlas y aplicarlas.

Dentro del tipo de herramientas que más usaban destacaban las soluciones de IA generativa (78 %), las de creación y edición de contenidos (63 %), los sistemas de recomendación (31 %), los asistentes virtuales (22 %) y las plataformas con elementos de gamificación (11 %).

En general, el 67 % del alumnado en España se mostraba interesado en aprender más sobre inteligencia artificial. A pesar de este interés, el 72 % admitía no haber recibido formación específica en este campo. Además, cuando se les preguntó por el impacto de la IA en el mundo laboral, la opción más elegida (41 %) fue la preocupación por la posible eliminación de empleos en ciertos sectores. Para reducir este riesgo, un 43 % consideraba necesario establecer reglas claras que regulen el uso de la inteligencia artificial en el entorno profesional.

3.2 PRINCIPALES LENGUAJES DE PROGRAMACIÓN UTILIZADOS: R Y PYTHON

Python se ha convertido en uno de los lenguajes más utilizados en el mundo de la inteligencia artificial y la ciencia de datos, y buena parte de su éxito se debe a lo **fácil que es empezar a usarlo**. Su sintaxis es clara, directa y muy parecida al lenguaje natural, lo que hace que muchas personas sin experiencia en programación puedan aprenderlo en poco tiempo. Esto lo convierte en una herramienta muy accesible, tanto para perfiles técnicos como para quienes vienen de otras áreas, como la economía, la salud o la sociología.

Además de ser fácil de aprender, **Python destaca por su ecosistema de librerías**. Hay herramientas ya desarrolladas para casi cualquier cosa:

- **NumPy** y **Pandas** son perfectas para trabajar con datos numéricos y tablas.
- **Scikit-learn** permite aplicar modelos de machine learning de forma sencilla.
- **TensorFlow** y **Keras** se usan para construir redes neuronales y hacer deep learning.
- **Matplotlib** o **Seaborn** son excelentes para crear gráficos y visualizar datos. Esta variedad de librerías hace que Python se use en todo tipo de proyectos, desde análisis de mercado hasta visión por computador o chatbots.

Por otro lado, **R** es un lenguaje que, aunque menos popular entre los programadores puros, **sigue siendo muy valorado en ámbitos donde se necesita un enfoque más estadístico**. Es especialmente fuerte en análisis exploratorio, pruebas estadísticas y visualización de datos. R se creó desde el principio con ese objetivo, y eso se nota: su entorno y sus paquetes están pensados para realizar cálculos complejos de forma cómoda, como regresiones, análisis multivariante o modelos de series temporales. **Ggplot2**, por ejemplo, es una de las librerías más potentes de visualización en este campo.

R es muy utilizado en el mundo académico, donde los estudios estadísticos son más habituales, y también en sectores como la sanidad, la economía o las finanzas, donde se manejan modelos matemáticos más tradicionales y se requiere precisión en el análisis estadístico clásico.

A la hora de elegir entre uno u otro, **la decisión depende del tipo de proyecto y del perfil del equipo**. Si se busca construir sistemas inteligentes, hacer análisis predictivos a gran escala o trabajar con grandes volúmenes de datos de distintas fuentes, **Python suele ser la mejor opción**. En cambio, si se va a trabajar con modelos estadísticos complejos, si se trata de proyectos de investigación o si ya se tiene experiencia con herramientas como SPSS o SAS, **R puede ofrecer ventajas más directas**.

En cuanto a ejemplos reales, Python está presente en aplicaciones de recomendación como las de YouTube o Amazon, en herramientas de análisis de texto como los filtros de contenido, o en modelos de predicción de riesgo en el ámbito bancario. R, por su parte, se utiliza en estudios epidemiológicos para analizar la evolución de enfermedades, en universidades para investigaciones sociales y en bancos para construir modelos de riesgo financiero.

Ambos lenguajes son herramientas muy potentes, y conocerlos permite tener más flexibilidad a la hora de afrontar distintos tipos de proyectos. Lo más habitual en equipos multidisciplinares es que convivan, y que cada profesional escoja el lenguaje que mejor se adapte a lo que necesita hacer.

Nota

Imagina que Big Data e Inteligencia Artificial (IA) son dos grandes mundos llenos de posibilidades, pero también muy complejos. Para poder movernos por esos mundos, necesitamos herramientas que nos ayuden a entender los datos, analizarlos, entrenar modelos y tomar decisiones. Y ahí es donde entran en juego Python y R, como si fueran dos superhéroes del análisis de datos.

Si estás metiéndote en el mundo del Big Data y la IA, Python es casi imprescindible por su versatilidad, escalabilidad y compatibilidad con sistemas modernos. Pero R sigue siendo muy valioso, sobre todo cuando necesitas un análisis estadístico profundo y una presentación elegante de resultados.

Situación	Lenguaje recomendado
Análisis de datos financieros en bolsa	Python
Visualización avanzada de datos epidemiológicos	R
Modelado predictivo para mantenimiento industrial	Python
Estudio estadístico de resultados académicos	R
Procesamiento de datos en tiempo real con sensores IoT	Python
Análisis de encuestas sociales	R
Desarrollo de dashboards interactivos empresariales	Python
Exploración de datos clínicos para ensayos médicos	R
Clasificación de correos electrónicos como spam o no	Python
Análisis de datos de experimentos psicológicos	R
Análisis de sentimiento en redes sociales	Python
Estudio de tendencias demográficas nacionales	R
Reconocimiento de imágenes para control de calidad	Python
Análisis multivariante de resultados de laboratorio	R
Automatización de informes de marketing	Python
Comparación de tratamientos médicos en estudios clínicos	R
Predicción de demanda energética	Python
Evaluación de políticas públicas usando datos censales	R
Modelos de recomendación en plataformas online	Python
Evaluación de rendimiento académico con modelos mixtos	R
Minería de textos legales	Python

Estudio de correlación entre factores ambientales	R
Procesamiento de logs en servidores web	Python
Análisis de datos experimentales en biología	R
Construcción de APIs de análisis de datos	Python
Evaluación de resultados en pruebas educativas estandarizadas	R
Segmentación de clientes en comercio electrónico	Python
Estadística descriptiva en estudios de salud pública	R
Implementación de modelos de deep learning	Python
Modelos de regresión lineal simple para publicaciones académicas	R

Ejemplo

Ejemplo en Python: predicción de abandono de clientes en una empresa de telefonía

Supón que trabajas en una empresa llamada Eledra Telcom y quieres predecir qué clientes podrían abandonar el servicio (churn). Tienes un montón de datos: edad, facturación mensual, quejas, duración del contrato, etc.

Objetivo:

Entrenar un modelo de IA que detecte patrones en los clientes que ya se han dado de baja y usarlo para anticiparse al abandono de nuevos clientes.

Paso a paso en Python (usando scikit-learn):

Python

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
```

```
# 1. Cargar datos
df = pd.read_csv('clientes_telefonia.csv')

# 2. Preprocesamiento básico
df = df.dropna()  # Eliminar filas con valores nulos
df['Genero'] = df['Genero'].map({'Masculino': 0, 'Femenino': 1})  # Codificar
variables

# 3. Dividir variables
X = df.drop('Abandono', axis=1)  # Variables predictoras
y = df['Abandono']               # Variable objetivo

# 4. Dividir en entrenamiento y test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_
state=42)

# 5. Entrenar modelo de IA
modelo = RandomForestClassifier(n_estimators=100, random_state=42)
modelo.fit(X_train, y_train)

# 6. Evaluar
y_pred = modelo.predict(X_test)
print(classification_report(y_test, y_pred))
```

¿Qué conseguimos con este código?

- Se analiza un dataset grande con miles de clientes (esto sería parte del Big Data si usáramos versiones distribuidas como Spark).
- Se entrena un modelo de IA que aprende a detectar quién se podría dar de baja.
- Luego se puede integrar ese modelo en la web de atención al cliente para que el sistema lance alertas preventivas.

Ahora un ejemplo en R: análisis estadístico con visualización.

Queremos entender cómo varía la facturación media por edad y género entre los clientes. Esto se usa como análisis exploratorio antes de construir un modelo.

Código en R:

R

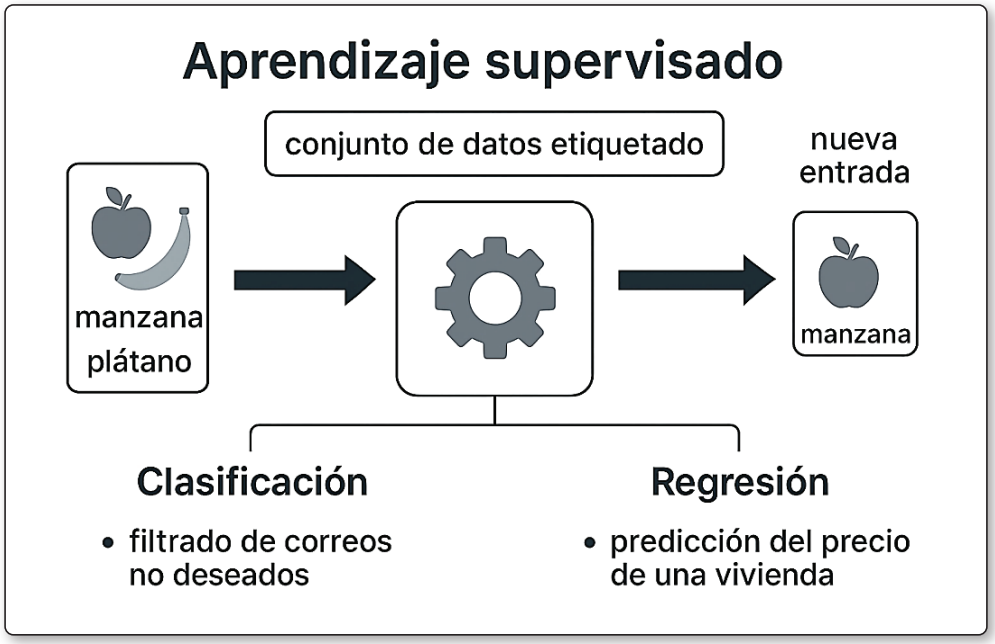
```
# Cargar librerías
library(tidyverse)

# Leer datos
clientes <- read.csv("clientes_telefonia.csv")

# Agrupar y calcular media
resumen <- clientes %>%
  group_by(Edad, Genero) %>%
  summarise(FacturacionMedia = mean(Facturacion, na.rm = TRUE))

# Visualizar
ggplot(resumen, aes(x = Edad, y = FacturacionMedia, color = Genero)) +
  geom_line(size = 1.2) +
  labs(title = "Facturación media por edad y género",
       x = "Edad",
       y = "Facturación (€)")
```

¿Qué conseguimos con esto?

- Vemos de forma clara si hay patrones de consumo por grupo de edad o por género.
- Podemos detectar si hay perfiles más rentables o en riesgo de abandono.
- Esto puede alimentar una estrategia de segmentación de clientes o personalización de ofertas.

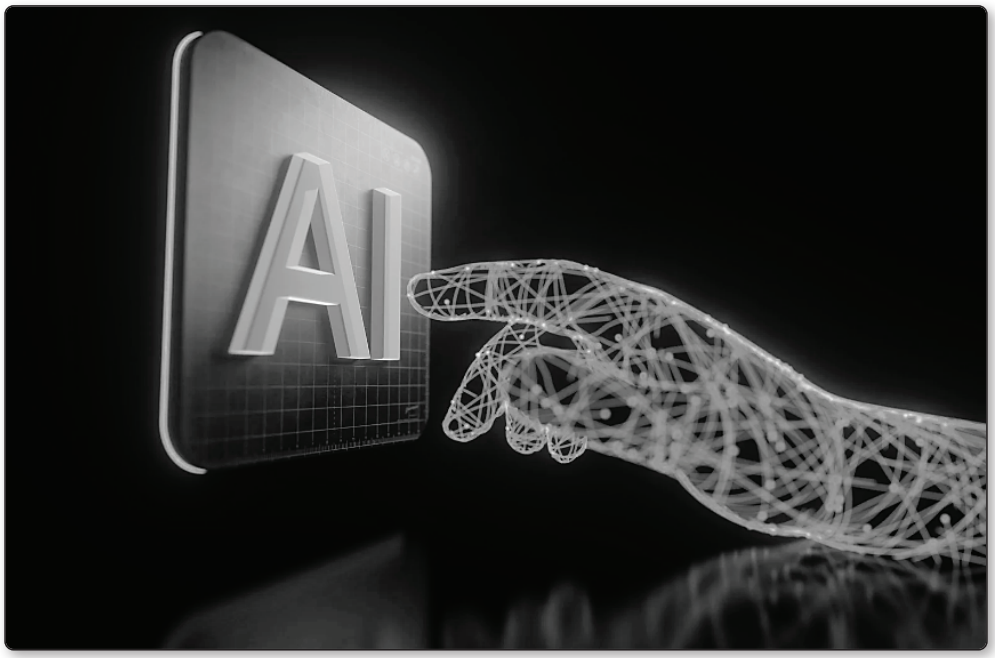
3.3 ALGORITMOS SUPERVISADOS: ¿QUÉ SON? ALGUNOS EJEMPLOS

El **aprendizaje supervisado** es una de las formas más comunes de entrenar a un sistema de inteligencia artificial. La idea es bastante sencilla: el modelo aprende a partir de ejemplos que ya tienen la respuesta correcta. Es como si le enseñaras a alguien a reconocer frutas mostrándole muchas fotos de manzanas y plátanos, cada una con su etiqueta correspondiente. Con el tiempo, esa persona (o en este caso, el algoritmo) aprende a distinguirlas por sí misma, incluso cuando aparecen en imágenes nuevas.



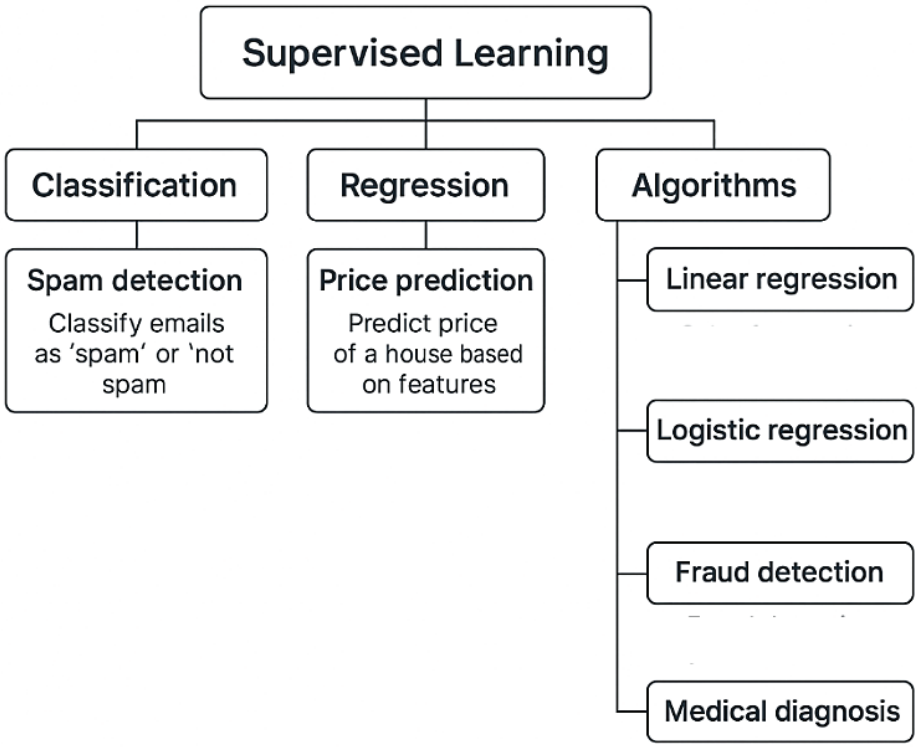
* Esta imagen representa de forma clara y sencilla cómo funciona el aprendizaje supervisado. En la parte central se muestra un modelo de inteligencia artificial que aprende a partir de un conjunto de datos etiquetado, es decir, ejemplos que ya vienen con su respuesta correcta. En este caso, se le muestran frutas como manzanas y plátanos, cada una identificada con su nombre. El modelo analiza esas etiquetas y extrae patrones. Después, cuando se le presenta una nueva entrada (una imagen sin etiqueta), es capaz de predecir la categoría correcta, como decir que se trata de una manzana. Debajo se muestran dos tipos principales de tareas supervisadas: la clasificación, como el filtrado de correos no deseados (spam), y la regresión, como la predicción del precio de una vivienda.

Este tipo de aprendizaje se basa en un **conjunto de datos etiquetado**, es decir, con información clara sobre qué representa cada caso. Por ejemplo, si queremos que un modelo detecte correos spam, se le entrena con muchos correos ya clasificados como "spam" o "no spam". Durante este proceso, el sistema va encontrando patrones comunes en los correos de cada tipo: palabras, estructuras, remitentes… Con esos patrones aprende a **generalizar**, de forma que cuando se le presenta un correo nuevo, pueda decidir con bastante acierto si es basura o no, incluso aunque nunca lo haya visto antes.



Las **aplicaciones prácticas del aprendizaje supervisado** son muchísimas. Una de las más conocidas es la **clasificación**, que consiste en asignar una categoría a cada caso. Esto se usa en cosas como el filtrado de correos no deseados, los diagnósticos médicos basados en síntomas o imágenes, o incluso en redes sociales, para decidir si un contenido debe ser moderado. Otra aplicación muy extendida es la **regresión**, que sirve para predecir un valor numérico. Por ejemplo, cuánto costará un piso según su ubicación y características, o cuánta energía se consumirá en una fábrica en las próximas semanas.

Dentro del aprendizaje supervisado hay **varios algoritmos muy conocidos**, cada uno con sus ventajas según el tipo de datos. La **regresión lineal**, por ejemplo, es muy útil cuando se quiere predecir una variable continua y se busca una relación directa entre variables. La **regresión logística** se utiliza más cuando la respuesta es binaria, como "sí" o "no", "positivo" o "negativo". Los **árboles de decisión** funcionan como un juego de preguntas que se van ramificando según las respuestas, y son muy fáciles de interpretar. Y las **máquinas de soporte vectorial (SVM)** son capaces de trazar una línea (o superficie) que separe grupos de datos de forma muy precisa, especialmente útil en casos complejos.



Esta imagen representa de forma esquemática el concepto de aprendizaje supervisado (Supervised Learning), que se divide en tres grandes bloques: clasificación, regresión y algoritmos. En el apartado de clasificación, se pone como ejemplo la detección de correos no deseados (spam detection), donde el sistema aprende a clasificar los correos como "spam" o "no spam". En regresión, se muestra cómo se puede predecir el precio de una vivienda (price prediction) basándose en sus características. Finalmente, en la columna de algoritmos, se mencionan algunos de los más usados: la regresión lineal para prever ventas (sales forecasting), la regresión logística para saber si un cliente se dará de baja (customer retention), la detección de fraudes (fraud detection) para clasificar transacciones como fraudulentas o no, y el diagnóstico médico (medical diagnosis), donde el sistema ayuda a predecir enfermedades a partir de datos clínicos. Todo esto parte de un conjunto de datos etiquetado con ejemplos previos, que permiten al modelo aprender y luego generalizar a nuevos casos.

Ejemplo

1. Clasificación: detectar correos spam

Imagina que trabajas en una empresa y recibes cada día cientos de correos. Algunos son importantes, pero muchos otros son publicidad no deseada o estafas. Para evitar perder tiempo, decides entrenar un sistema que aprenda a separar los correos "spam" de los "no spam".

Para ello, usas un conjunto de ejemplos reales: 10.000 correos que ya han sido clasificados previamente por personas. Algunos están etiquetados como *spam* (por ejemplo, "Consigue dinero fácil en 24 horas"), y otros como *no spam* (como "Resumen de ventas del mes").

El sistema empieza a comparar los mensajes y encuentra patrones: los correos spam suelen tener muchas mayúsculas, enlaces sospechosos, o ciertas palabras como "gratis", "urgente", "premio". En cambio, los correos normales usan un lenguaje más formal, suelen venir de remitentes conocidos y contienen menos imágenes llamativas.

Después de entrenarlo, se le presenta un correo nuevo que nunca ha visto antes. El sistema lo analiza y, basándose en lo que aprendió, decide que sí es spam, porque contiene muchas palabras típicas de ese tipo de mensajes. ¡Así de simple funciona la clasificación!

2. Regresión: predecir el precio de un piso

Ahora imagina que estás buscando piso para comprar, pero quieres saber si te están cobrando de más. Para eso, decides entrenar un modelo que aprenda a predecir el precio aproximado de un piso basándose en sus características**:** metros cuadrados, número de habitaciones, si tiene ascensor, barrio, planta…

Recolectas una tabla con datos de 5.000 pisos vendidos en tu ciudad. Cada fila tiene la información del piso y el precio final por el que se vendió. Esa información sirve para que el modelo encuentre relaciones: por ejemplo, que en el centro los pisos son más caros, que tener ascensor sube el precio, o que los bajos son más baratos que los áticos.

Después de entrenar al modelo, le pasas la información de un piso concreto: 70 m², 2 habitaciones, en el barrio de Ruzafa (Valencia), planta 3 sin ascensor. El modelo calcula, basándose en los ejemplos anteriores, que ese piso debería costar unos 185.000 euros. Ya puedes comparar con el precio real que te piden y decidir con más criterio.

Esto es lo que se llama regresión, porque no se trata de elegir una categoría (como "caro" o "barato"), sino de predecir un valor numérico.

3. Regresión lineal: predecir ventas en función del gasto en publicidad

Supongamos que tienes una pequeña tienda online y quieres saber cómo influye el gasto en publicidad en tus ventas. Tienes datos de los últimos 12 meses: cuánto invertiste en anuncios cada mes y cuánto vendiste. Si haces un gráfico, verás que a más inversión, más ventas. Pero no sabes exactamente cuánta diferencia hay.

Ahí entra la regresión lineal: el modelo dibuja una recta que se ajusta lo mejor posible a esos puntos, y con eso puede predecir cuánto venderás si el próximo mes gastas, por ejemplo, 500 euros en publicidad. Te dice algo como: "si gastas 500, probablemente vendas unos 3.000 euros".

Es muy útil para detectar tendencias simples y tomar decisiones rápidas.

4. Regresión logística: predecir si un cliente comprará o no

Ahora imagina que trabajas para una empresa de seguros y quieres predecir si una persona va a contratar un seguro o no. Tienes muchos datos de antiguos clientes: edad, ingresos, tipo de vehículo, código postal, y si finalmente contrataron o no.

Como la respuesta solo puede ser "sí" o "no", usas una regresión logística, que es perfecta para este tipo de situaciones. El modelo aprende qué características suelen estar asociadas con personas que contratan, y cuáles no.

Así, cuando entra un nuevo cliente, el sistema te puede decir: "esta persona tiene un 82% de probabilidades de contratar". Y con esa información, tú puedes decidir si le haces una oferta especial o le das seguimiento personalizado.

5. Árboles de decisión: detectar fraude en tarjetas bancarias

Imagina que trabajas en un banco y te encargas de detectar posibles fraudes en las tarjetas. Sabes que cuando alguien hace una compra rara (como pagar 1.000 € desde otro país a las 3 de la madrugada), puede ser un robo de datos.

Un árbol de decisión funciona como un juego de "sí" o "no". El modelo va aprendiendo preguntas como:

- ¿La compra es en el mismo país del titular?
- ¿Es un importe alto?
- ¿Es un horario habitual?
- ¿Se parece a compras anteriores?

Cada respuesta lleva a una nueva rama del árbol, y al final se decide si esa operación parece normal o sospechosa. Lo mejor de los árboles es que son muy fáciles de interpretar, incluso para personas que no son técnicas. Se puede ver exactamente cómo llegó el sistema a su conclusión.

6. SVM (Máquinas de soporte vectorial): clasificar imágenes médicas

En un hospital se quiere desarrollar un sistema que distinga radiografías con tumores de las que no tienen. Se cargan miles de imágenes ya clasificadas por médicos (con y sin tumor), y se entrenan modelos para que aprendan a diferenciarlas.

Aquí las Máquinas de Soporte Vectorial (SVM) son muy útiles, porque son capaces de encontrar una "frontera" muy precisa que separa los casos positivos de los negativos, incluso cuando la diferencia es sutil.

Una vez entrenado, el modelo puede analizar una radiografía nueva y decir: "con un 94% de certeza, esta imagen corresponde a un caso con tumor", ayudando al personal médico a hacer diagnósticos más rápidos y seguros.

Otra cosa interesante del **aprendizaje supervisado** es que no se limita a tareas complejas. Muchas veces está detrás de funciones que parecen simples, pero que requieren un modelo entrenado. Por ejemplo, cuando una app de fotos **reconoce rostros** y los etiqueta automáticamente, lo hace porque ha aprendido con miles de ejemplos etiquetados previamente. Lo mismo ocurre con los sistemas de reconocimiento de voz: el modelo ha escuchado muchas grabaciones asociadas a su transcripción correcta y ha aprendido a interpretar lo que decimos.

Además, estos modelos pueden **mejorar con el tiempo** si se les sigue alimentando con nuevos datos y ejemplos reales. Esto se llama "entrenamiento continuo", y es especialmente útil en entornos donde los patrones cambian mucho, como en el comercio electrónico o la ciberseguridad. Por ejemplo, los correos de spam de hace diez años no se parecen nada a los actuales. Por eso, el modelo necesita ir actualizándose para seguir siendo eficaz.

En entornos profesionales, el aprendizaje supervisado se aplica con frecuencia para **automatizar tareas que requieren cierta inteligencia humana**, pero que serían lentas o costosas de hacer a mano. Pensemos en un sistema que revise automáticamente miles de solicitudes de crédito: con los datos de clientes anteriores y si devolvieron o no el préstamo, se puede entrenar un modelo para predecir si una nueva solicitud es de riesgo. Esto ayuda a **tomar decisiones más rápidas y con menos errores**, siempre que el modelo haya sido bien entrenado y validado.

Es importante tener en cuenta que estos modelos **necesitan datos de calidad y bien etiquetados para funcionar correctamente**. Si las etiquetas están mal o los ejemplos son poco representativos, el modelo puede aprender cosas incorrectas. Por eso, en muchos proyectos se dedica bastante tiempo a preparar los datos antes incluso de entrenar el modelo.

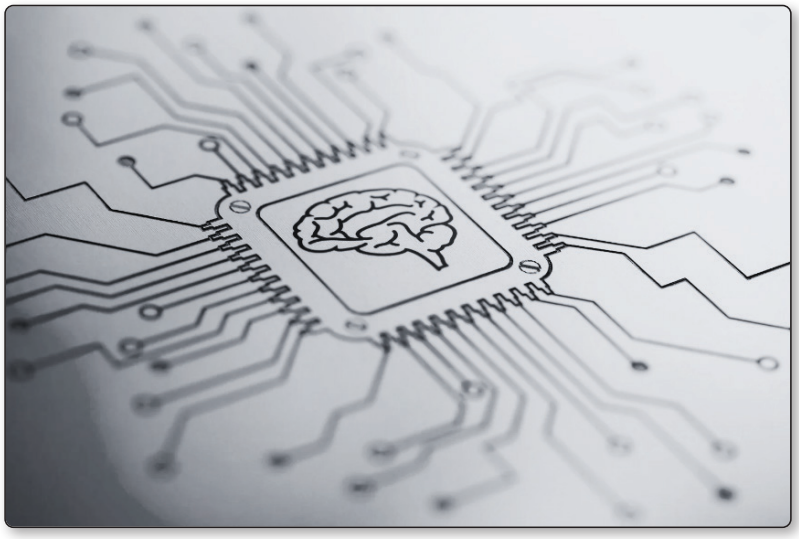
3.4 ALGORITMOS NO-SUPERVISADOS: ¿QUÉ SON? ALGUNOS EJEMPLOS

El **aprendizaje no supervisado** es un enfoque dentro de la inteligencia artificial en el que el modelo **aprende a identificar patrones por sí mismo**, sin necesidad de que los datos estén etiquetados previamente. Es decir, a diferencia del aprendizaje supervisado, aquí no le decimos al sistema cuál es la respuesta correcta. Simplemente le damos los datos tal y como están y le dejamos buscar relaciones, similitudes o diferencias por su cuenta. Es como si le entregáramos una caja con piezas mezcladas y el sistema se encargara de ordenarlas en grupos según sus características.

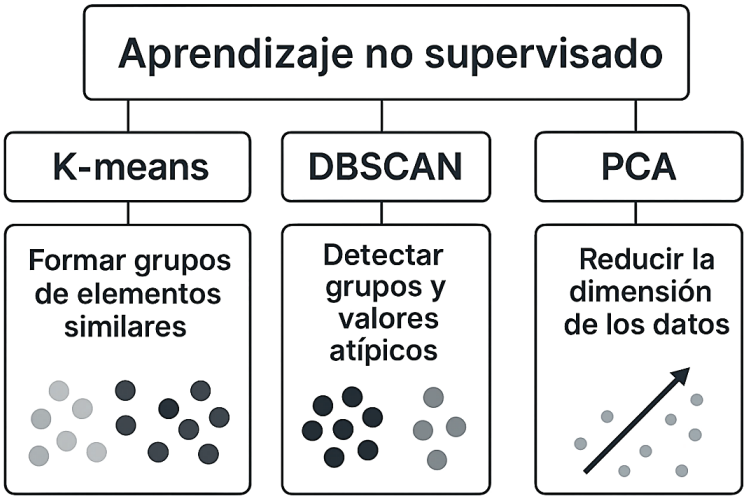
Este tipo de aprendizaje es muy útil para **explorar datos** cuando no sabemos muy bien qué hay dentro o qué deberíamos buscar. También se usa mucho en tareas de **agrupamiento**, donde se intenta encontrar elementos parecidos entre sí, o en **detección de anomalías**, para identificar cosas que se salen de lo común. Por ejemplo, si tenemos registros de miles de transacciones y una de ellas tiene un comportamiento completamente diferente, un algoritmo no supervisado puede detectarla como algo que merece atención.

En el mundo real, hay muchos usos prácticos para este tipo de algoritmos. En marketing, por ejemplo, se utilizan para hacer **segmentación de clientes**. Imagina una tienda online que quiere enviar promociones personalizadas, pero no sabe cómo clasificar a sus usuarios. Un modelo no supervisado puede analizar comportamientos de compra, hábitos de navegación o respuestas a campañas anteriores, y proponer grupos de clientes con perfiles similares. Esto permite diseñar estrategias más afinadas y efectivas, sin necesidad de etiquetar manualmente a cada usuario.

Otro caso habitual es la **agrupación de productos o documentos**. Por ejemplo, una plataforma de libros electrónicos puede usar estos algoritmos para organizar automáticamente miles de títulos en categorías similares, incluso si no tienen etiquetas de género. El modelo detecta patrones en los textos o en los hábitos de lectura de los usuarios y crea agrupaciones que luego pueden usarse para recomendaciones o para mejorar la navegación en la web.



Entre los **algoritmos más conocidos de aprendizaje no supervisado**, uno de los más utilizados es **K-means**, que forma grupos de elementos parecidos entre sí. Por ejemplo, si tenemos datos sobre altura y peso de personas, K-means puede separar el conjunto en varios grupos que representen diferentes perfiles físicos. Otro algoritmo interesante es **DBSCAN**, que permite detectar grupos con formas menos uniformes y también identificar valores que están muy alejados del resto, lo que viene bien para detectar fraudes o errores. También está el **PCA** (análisis de componentes principales), que se usa mucho para **reducir la complejidad de los datos** y visualizar mejor lo que contienen, sin perder la información más importante.



Ejemplo

1. K-means (agrupación por similitud)

Ejemplo:

Agrupar clientes según su perfil físico.

Imagina que trabajas en un gimnasio y tienes una hoja de cálculo con los datos de altura y peso de 500 personas socias. No sabes a qué grupo pertenece cada una, pero te interesa detectar patrones comunes para ofrecer rutinas personalizadas.

Con el algoritmo K-means, le dices al sistema que quieres crear, por ejemplo, 3 grupos. El algoritmo empieza a analizar los datos y, tras varias iteraciones, consigue agrupar a las personas en tres perfiles:

- Grupo 1: personas de baja estatura y peso ligero.
- Grupo 2: personas de estatura media y peso moderado.
- Grupo 3: personas altas con peso más elevado.

Esto te permite ofrecer rutinas adaptadas según el grupo, sin haber tenido que etiquetar a cada persona de antemano. K-means lo deduce solo en base a los datos.

2. DBSCAN (agrupación con detección de valores atípicos)

Ejemplo:

Detección de fraudes en pagos con tarjeta.

Trabajas en un banco y tienes miles de registros de pagos con tarjeta: hora del día, importe, lugar, frecuencia… Sabes que algunos fraudes no siguen los patrones normales, pero no tienes una etiqueta clara para entrenar un modelo supervisado.

Con DBSCAN, puedes buscar grupos de transacciones con comportamientos similares, y al mismo tiempo identificar transacciones extrañas que no encajan en ningún grupo.

Por ejemplo, si una persona hace compras pequeñas y regulares en su ciudad y, de repente, aparece una compra grande desde otro país a las 3 de la madrugada, DBSCAN puede detectarlo como un valor atípico. Así se puede activar una alerta de posible fraude sin necesidad de ejemplos etiquetados.

3. PCA (análisis de componentes principales)

Ejemplo:

Visualizar mejor un conjunto de datos de vinos.

Supón que tienes un conjunto de datos sobre 1.000 tipos de vino, y cada uno tiene 13 características químicas distintas (acidez, nivel de alcohol, color, etc.). Son muchos datos y cuesta mucho analizarlos todos a la vez.

Con PCA, puedes reducir esos 13 factores a solo 2 o 3 componentes principales, que son combinaciones de los originales, pero que conservan la mayor parte de la información. Esto permite hacer un gráfico en dos dimensiones donde se visualizan los vinos según su perfil químico general.

Gracias a esto, puedes ver si hay vinos que se agrupan naturalmente (por ejemplo, tintos secos frente a blancos afrutados), o si hay alguno que destaca por ser muy diferente. Así, aunque no estés clasificando los vinos, puedes entender mejor la estructura de los datos y explorar su comportamiento.

Este tipo de modelos son especialmente potentes cuando trabajamos con datos muy grandes y variados, y queremos descubrir patrones que no están a simple vista. A veces el objetivo no es obtener una respuesta concreta, sino **comprender**

mejor cómo se comportan los datos, descubrir estructuras ocultas o preparar el terreno para otros análisis más específicos. Es una herramienta muy valiosa para explorar y organizar la información cuando todavía no sabemos bien qué buscar.

3.5 INTRODUCCIÓN AL DEEP LEARNING Y EL APRENDIZAJE POR REFUERZO

El **Deep Learning**, o aprendizaje profundo, es una rama del aprendizaje automático que se basa en el uso de **redes neuronales artificiales con muchas capas**, también llamadas redes profundas. La idea es que cuantas más capas tenga la red, más capaz será de **aprender representaciones complejas de los datos**. Estas capas actúan como filtros que transforman progresivamente la información, desde lo más simple hasta lo más abstracto. Por ejemplo, en una imagen, las primeras capas pueden detectar líneas o colores, y las últimas pueden identificar objetos como una cara, un coche o una señal de tráfico.