A. ¿Qué es Machine Learning?

El *Machine Learning* (ML), o aprendizaje automático, es una disciplina de la Inteligencia Artificial (IA) que se centra en el diseño y desarrollo de algoritmos capaces de **aprender automáticamente a partir de datos** y de **mejorar su rendimiento con la experiencia**, sin ser programados explícitamente para cada tarea.

A diferencia de la programación tradicional, donde el ser humano define paso a paso qué debe hacer un sistema, en ML el proceso cambia:

- Se proporciona al modelo un **conjunto de datos de entrenamiento**.
- El algoritmo busca **patrones y relaciones estadísticas** en esos datos.
- Con esos patrones construye una **función de predicción o decisión**.
- Dicha función se utiliza luego con datos nuevos para resolver problemas similares.

1. El núcleo del ML: la función de aproximación

En términos matemáticos, el aprendizaje automático busca **aproximar una función desconocida** que relaciona entradas (X) con salidas (Y).

- Por ejemplo:
    - Entrada X: características de una casa (tamaño, ubicación, número de habitaciones).
    - Salida Y: precio de venta.

El algoritmo de ML no conoce la regla exacta que determina el precio, pero **ajusta un modelo** (una función) para que, dadas unas entradas nuevas, la predicción sea lo más cercana posible a la realidad.

2. Proceso típico de aprendizaje automático

1. **Definición del problema:** identificar qué queremos predecir o clasificar.
2. **Recogida de datos:** sin datos no hay aprendizaje. La calidad y cantidad son determinantes.
3. **Preprocesamiento:** limpieza, transformación y selección de variables relevantes.

4. **Selección del modelo:** escoger el algoritmo más adecuado (regresión lineal, árboles de decisión, redes neuronales, etc.).
5. **Entrenamiento:** ajustar los parámetros del modelo minimizando un error o maximizando un criterio.
6. **Evaluación:** medir el rendimiento del modelo con datos que no ha visto (validación y test).
7. **Predicción/Despliegue:** aplicar el modelo en el mundo real.

3. Aprendizaje vs. Memorización

Un aspecto clave es la diferencia entre **aprender** y **memorizar**:

- Si un modelo simplemente memoriza los datos de entrenamiento, funcionará bien con ejemplos conocidos, pero fallará con casos nuevos.
- El verdadero aprendizaje consiste en **generalizar**: encontrar reglas útiles más allá de los datos concretos que se usaron para entrenar.

Este equilibrio se conoce como **trade-off sesgo-varianza**:

- **Sesgo alto:** el modelo es demasiado simple → no capta los patrones.
- **Varianza alta:** el modelo es demasiado complejo → se sobreajusta a los datos y no generaliza.

4. ML y su relación con la estadística

El ML comparte raíces con la estadística. Ambos buscan descubrir patrones en datos.

- La **estadística clásica** se enfoca en inferir propiedades de una población a partir de una muestra (explicación e interpretación).
- El **ML** se centra en construir sistemas que **predigan con precisión** y se adapten a datos cambiantes, incluso si el modelo no es fácilmente interpretable.

En la práctica moderna, ML y estadística se complementan:

- La estadística aporta rigurosidad en la inferencia.
- El ML aporta escalabilidad y capacidad de trabajar con grandes volúmenes de datos y alta complejidad.

5. Características principales del Machine Learning

1. **Dependencia de los datos:** sin datos adecuados, no hay aprendizaje.
2. **Capacidad de adaptación:** el modelo puede ajustarse a cambios en el entorno si se reentrena con nuevos datos.
3. **Automatización parcial:** reduce la necesidad de programación explícita, pero requiere intervención humana en la preparación de datos y la validación de modelos.
4. **Aplicabilidad transversal:** se utiliza en medicina, economía, ingeniería, arte, educación, etc.

6. Ejemplo ilustrativo

Supongamos que queremos construir un filtro de spam en el correo electrónico:

- **Entrada (X):** características de un correo (palabras usadas, frecuencia de enlaces, remitente).
- **Salida (Y):** "spam" o "no spam".
- El algoritmo analiza miles de correos etiquetados previamente.
- Aprende patrones: por ejemplo, que correos con palabras como *gratis* o *ganador* tienen mayor probabilidad de ser spam.
- Una vez entrenado, el sistema puede clasificar correos nuevos sin intervención humana.

En síntesis: *Machine Learning* es la ciencia y el arte de que las máquinas aprendan de los datos para realizar predicciones o tomar decisiones. No reemplaza la programación tradicional, sino que la complementa en problemas donde definir reglas exactas es inviable o demasiado costoso.

B. Diferencias entre programación, M-L, D-L y IA

Programación (software en general)

└── Inteligencia Artificial (IA)

  └── Machine Learning (ML)

    └── Deep Learning (DL)

Mapa mental rápido

Programación (software en general)
└── Inteligencia Artificial (IA)
  └── Machine Learning (ML)
    └── Deep Learning (DL)

Todo ML y DL es IA, y todo IA/ML/DL se implementa con programación, pero **no toda programación es IA** ni **toda IA aprende de datos**.

1) Programación clásica (reglas explícitas)

Qué es Desarrollar software definiendo paso a paso qué debe hacer el sistema (**reglas deterministas**). El conocimiento está **codificado por personas**.

Ejemplo Un validador de DNI/NIF: reglas de formato, cálculo de letra de control, mensajes de error. Un algoritmo de ordenación (quicksort) que siempre produce el mismo resultado para la misma entrada.

Rasgos clave

- **Determinismo**: misma entrada → misma salida.

- **Conocimiento**: reglas y casos escritos a mano.
- **Datos**: no son necesarios para "aprender"; sí para operar.
- **Evaluación**: tests unitarios, casos borde, cobertura de código.
- **Mantenimiento**: si cambian los requisitos, cambias el código.

Cuándo usarla

- Problemas con **reglas claras y estables** (fiscalidad simple, validaciones, CRUD, ETL, UIs, backends).
- Exige **explicabilidad total** y trazabilidad paso a paso.
- **Pocos datos** o no hay patrones que "descubrir".

Limitación

- Escala mal cuando las reglas son **demasiadas, ambiguas o cambiantes** (p. ej., reconocer gatos en fotos).

2) Inteligencia Artificial (IA)

Qué es Campo que busca sistemas que realicen tareas que asociamos a la inteligencia humana: **razonamiento, búsqueda, planificación, percepción, lenguaje**. **No toda IA aprende de datos**.

Dos grandes familias

1. **IA simbólica / GOFAI** (basada en conocimiento): reglas lógicas, ontologías, motores de inferencia, planificación, satisfacibilidad de restricciones (CSP), grafos de conocimiento.
    a. Ej.: un **sistema experto** médico con cientos de reglas "SI… ENTONCES…".
2. **IA basada en datos**: aquí entra **ML** (y dentro, **DL**).

Rasgos clave de IA simbólica

- **Explicabilidad alta** (árbol de inferencias).
- Requiere **ingeniería del conocimiento** (extraer reglas de expertos).
- Frágil ante **incertidumbre y ruido**; difícil en percepción (visión/audio) pura.

Cuándo usar IA (no-ML)

- Necesitas **razonar** con reglas, **planificar** (p. ej., rutas con restricciones), **probar** propiedades, o combinar conocimiento estructurado con consultas complejas.

3) Machine Learning (ML)

Qué es Subcampo de la IA que aprende **funciones** a partir de **datos**. Busca una aproximación $\hat{f}: X \rightarrow Y$ que **generalice** a ejemplos no vistos, optimizando una **métrica** (pérdida) con **validación** en datos separados.

Familias típicas

- **Supervisado** (clasificación, regresión).
- **No supervisado** (clustering, reducción de dimensión).
- **Semisupervisado** y **aprendizaje por refuerzo** (este último, a veces se trata aparte).

Rasgos clave

- **Estadístico y probabilístico**: salidas con incertidumbre (p. ej., probabilidad de fraude).
- **Ingeniería de características** ("features") es crítica (en ML clásico).
- **Evaluación**: accuracy, F1, AUC, MAE/MSE, curva PR, calibración, validación cruzada.
- **Riesgos**: **sobreajuste**, **fuga de variables**, **shift de distribución**, **sesgo**.

Algoritmos representativos

- Regresión lineal/logística, **árboles**, **random forests**, **gradient boosting** (XGBoost/LightGBM/CatBoost), **SVM**, **k-NN**, **PCA**, **k-means**.

Cuándo usar ML

- Hay **datos históricos** que reflejan la tarea.
- Reglas manuales serían **demasiadas** o **no precisas**.
- Necesitas **probabilidades** y priorización (riesgo de impago, propensión a compra, detección de fraude).

4) Deep Learning (DL)

Qué es Subconjunto de ML basado en **redes neuronales profundas** que **aprenden representaciones** automáticamente (representational learning). Reducen (o desplazan) la ingeniería manual de features.

Arquitecturas comunes

- **CNN** (visión),
- **RNN/LSTM** (secuencias),
- **Transformers** (texto, visión, audio, multivariado).

Rasgos clave

- **Escala con datos y cómputo** (GPU/TPU).
- Muy fuerte en **percepción** y **señales complejas** (imagen, audio, lenguaje).
- **Menor interpretabilidad** (aunque existen SHAP, Grad-CAM, attributions).
- Sensible a **datos y distribución**; necesita **regularización**, **early stopping**, **data augmentation**.

Cuándo usar DL

- Problemas de **alta dimensión** y **patrones complejos**: visión artificial, ASR, NLU, recomendadores masivos, series temporales complejas.
- Cuando ML clásico **satura** en rendimiento y hay **datos suficientes**.

5) Comparativa directa

Perspectiva “fuente de inteligencia”

Aspecto	Programación	IA (simbólica)	ML	DL
Origen del conocimiento	Reglas humanas	Reglas + ontologías	**Datos etiquetados/no etiquetados**	**Datos masivos**
Determinismo	Alto	Alto (con lógica)	Probabilístico	Probabilístico
Ingeniería de características	N/A	N/A	**Alta**	**Baja–media** (las aprende)
Interpretabilidad	Muy alta	Alta	Media	Baja (salvo técnicas específicas)
Datos necesarios	—	—	**De decenas a millones**	**Habitualmente muchos**
Cómputo	CPU	CPU	CPU/GPU	**GPU/TPU**
Robustez a ruido	Baja–media (si no se prevé)	Baja	Media	Media–alta (con datos/regularización)
Mantenimiento	Cambiar reglas	Mantener base de conocimiento	**Reentrenar y monitorizar**	**Reentrenar y monitorizar**

Perspectiva “ciclo de vida”

- **Programación/IA simbólica**: análisis → diseño reglas → implementación → pruebas.
- **ML/DL**: problema → datos → *split* (train/val/test) → **modelo** → **optimización** → **evaluación** → **despliegue** → **monitorización y reentrenos**.

Perspectiva “pruebas y métricas”

- **Programación**: tests unitarios/integración, invariantes.
- **IA simbólica**: cobertura de reglas, consistencia lógica.
- **ML/DL**: métricas predictivas, *offline validation*, *online A/B*, **drift detection**, fairness.

6) El mismo problema visto desde cada paradigma

Tarea: detectar transacciones fraudulentas.

1. **Programación clásica (reglas)**
   Ejemplo: si (importe > 1000) y (país no habitual) y (hora extraña) entonces marcar_fraude()
- Fácil de explicar, **falso positivos** si los defraudadores cambian el patrón.
2. **IA simbólica (conocimiento + reglas ponderadas)**
- Ontología de comercios, reglas con grados de certeza, motor de inferencia.
- Mejor estructuración, pero sigue **dependiendo de expertos** para actualizar.
3. **ML supervisado (gradient boosting)**
- Entrena con miles de casos etiquetados (fraude/no fraude).
- Aprende **combinaciones no triviales**; devuelve **probabilidad**.
- Requiere **monitorizar drift** y reentrenar.
4. **DL (transformer sobre secuencias de transacciones)**
- Aprende **patrones temporales complejos** y representaciones del cliente/comercio.
- Suele rendir más alto con mucho dato y compute; menos interpretable.

7) Decidir qué usar (regla práctica)

1. **¿Existen reglas claras y cerradas?** Sí → **Programación clásica**.
2. **¿Necesito razonar con conocimiento estructurado y explicable?** Sí → **IA simbólica** (reglas, planificación).
3. **¿Hay datos históricos relevantes y la tarea es predecir?** Sí → **ML** (empieza por modelos tabulares robustos: árboles/boosting).
4. **¿El input es percepción o señal compleja (imagen, audio, lenguaje) o ML clásico se queda corto y tengo datos/cómputo?** Sí → **DL**.

A menudo la solución **ganadora es híbrida**: reglas de negocio + ML (para priorizar) + validaciones programadas + supervisión humana. También emergen enfoques **neuro-simbólicos** que combinan conocimiento y aprendizaje.

8) Fallos típicos y cómo evitarlos

- **Sobreajuste (ML/DL)**: usa validación cruzada, regularización, *early stopping*, *dropout*, *augmentation*.
- **Fuga de variables**: estricta separación temporal y de conjuntos; revisa canalizaciones.
- **Shift de distribución**: monitoriza datos y performance; *retraining* programado.
- **Sesgo y fairness**: auditorías por subgrupos; métricas de equidad; *post-processing*.
- **Espagueti de reglas (programación/IA simbólica)**: refactorizar, pruebas de regresión, linters de conocimiento.

9) Resumen en una frase cada uno

- **Programación**: “Te digo exactamente qué hacer”.
- **IA (simbólica)**: “Te doy conocimiento/razónalo con reglas”.
- **ML**: “Aprende de ejemplos para predecir”.
- **DL**: “Aprende también **cómo** representar los datos”.

C. Tipos de aprendizaje en Machine Learning

En *Machine Learning*, los algoritmos aprenden de los datos siguiendo diferentes paradigmas, según la información que tengan disponible y el objetivo que persigan. Los principales tipos son:

1️⃣ Aprendizaje supervisado

Definición: En el aprendizaje supervisado, el modelo se entrena con datos **etiquetados**, es decir, cada ejemplo de entrada tiene una salida conocida que sirve como referencia. El objetivo es **aprender la relación entre las entradas y las salidas** para predecir resultados de nuevos datos.

Componentes principales:

- **Entrada (X):** variables o características del problema.
- **Salida (Y):** etiqueta o valor a predecir.
- **Función objetivo:** minimizar el error entre la predicción y la salida real.

Tipos de tareas:

- **Clasificación:** la salida es una categoría (spam/no spam, tumor benigno/maligno, aprobado/reprobado).
- **Regresión:** la salida es un valor continuo (precio de una casa, temperatura, ventas futuras).

Ejemplo real:

- Predecir si un correo electrónico es spam usando ejemplos de correos ya clasificados.
- Estimar el precio de venta de un inmueble a partir de su tamaño, ubicación y número de habitaciones.
- Diagnóstico médico: clasificar imágenes médicas como sanas o con patología.

2️⃣ Aprendizaje no supervisado

Definición: Aquí, los datos **no tienen etiquetas**, y el modelo intenta **descubrir patrones o estructuras ocultas** en la información. No se le dice qué debe predecir; debe **organizar o resumir los datos** por sí mismo.

Objetivos principales:

- **Agrupamiento (clustering):** encontrar grupos de datos similares.
- **Reducción de dimensionalidad:** simplificar datos complejos para visualización o preprocesamiento.
- **Detección de anomalías:** identificar datos que no siguen patrones comunes.

Ejemplo real:

- Segmentación de clientes según su comportamiento de compra para marketing.
- Agrupar noticias por temática sin etiquetas previas.
- Detectar transacciones bancarias sospechosas que se alejan del comportamiento normal.

3️⃣ Aprendizaje por refuerzo (solo mención)

Definición:

- En el aprendizaje por refuerzo, un **agente** interactúa con un **entorno** tomando decisiones.
- Cada acción recibe una **recompensa** o **castigo**, y el agente aprende a maximizar la recompensa acumulada.

Ejemplo real:

- Juegos: AlphaGo, videojuegos tipo Atari, donde el agente mejora jugando repetidamente.
- Robótica: un robot aprende a caminar o a manipular objetos sin recibir instrucciones explícitas.

🔹 Ejemplos reales de uso de Machine Learning

Los algoritmos de ML se aplican en muchos sectores y tareas, mostrando su **versatilidad**:

1. **Salud:** detección de enfermedades a partir de imágenes médicas, predicción de riesgo de enfermedades.
2. **Finanzas:** detección de fraude en tarjetas de crédito, predicción de riesgo crediticio.
3. **Marketing:** recomendaciones personalizadas en tiendas online, segmentación de clientes, análisis de comportamiento.
4. **Industria:** mantenimiento predictivo de maquinaria, optimización de procesos industriales.
5. **Transporte:** coches autónomos, optimización de rutas de entrega.
6. **Tecnologías de consumo:** asistentes virtuales, reconocimiento de voz y facial, filtros de spam en correos electrónicos.