


Introducción al aprendizaje de máquina y preprocesamiento de datos

 por Kibernetum Capacitación S.A.

¿Qué es el aprendizaje de máquina?

El aprendizaje de máquina o Machine Learning (ML) es una subdisciplina de la inteligencia artificial (IA) que permite a las computadoras detectar patrones en datos y hacer predicciones o tomar decisiones sin estar explícitamente programadas para cada tarea. En lugar de seguir reglas rígidas, un modelo de ML aprende a partir de la experiencia (datos) y mejora su desempeño con el tiempo.

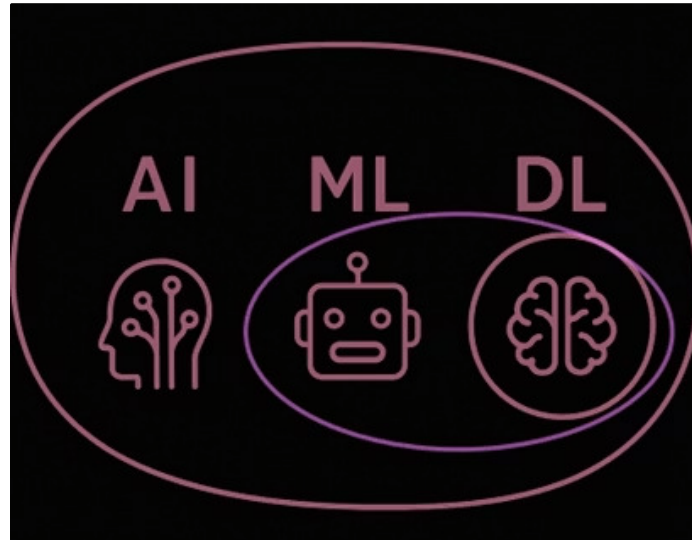


Imagen 1: *Toda inteligencia artificial no es machine learning, pero todo machine learning es IA. A su vez, el deep learning es una técnica avanzada dentro del machine learning.*

Una plataforma como Netflix o YouTube utiliza ML para recomendar contenido personalizado. El sistema analiza tus hábitos de consumo, los compara con otros usuarios similares, y predice qué películas o videos podrían gustarte más.

¿Por qué es importante el aprendizaje de máquina?

Según IBM, el ML es fundamental en el mundo actual porque permite automatizar tareas complejas, descubrir información oculta en grandes volúmenes de datos y tomar decisiones con mayor rapidez y precisión. Algunos beneficios clave incluyen:

Beneficio	Ejemplo de Aplicación
Automatización inteligente	Detección automática de fraudes bancarios
Mejora de procesos	Optimización logística en tiempo real
Experiencia personalizada	Recomendaciones en e-commerce
Descubrimiento de patrones	Diagnóstico médico basado en imágenes

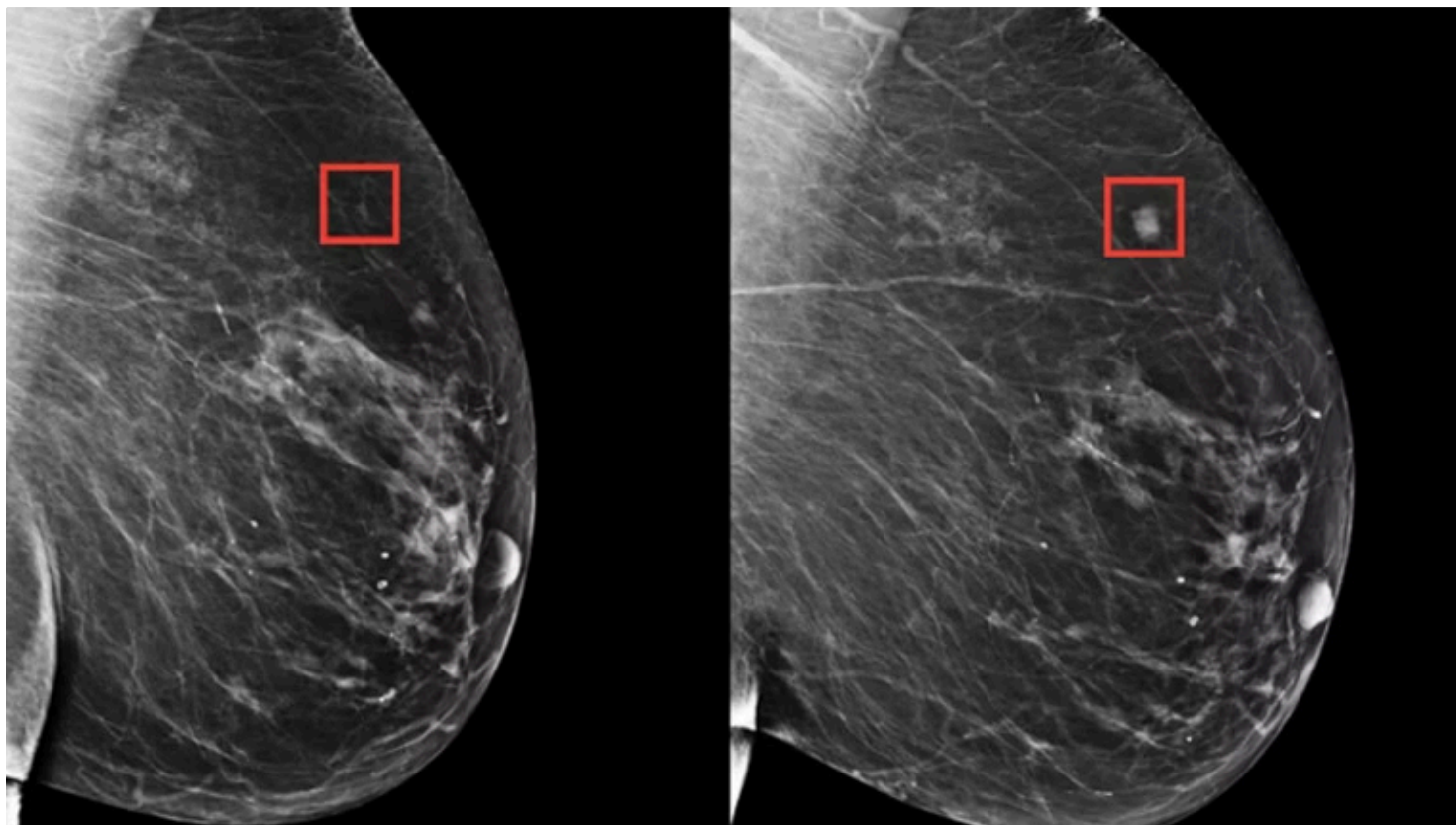


Imagen 2: Investigadores han podido desarrollar una serie de algoritmos que aprenden los sutiles patrones del tejido mamario que son precursores de un tumor maligno.

¿Cómo funciona un modelo de ML?

Un sistema de ML atraviesa distintas etapas para poder aprender y predecir:

Entrada de datos → Preprocesamiento → Entrenamiento → Evaluación → Predicción

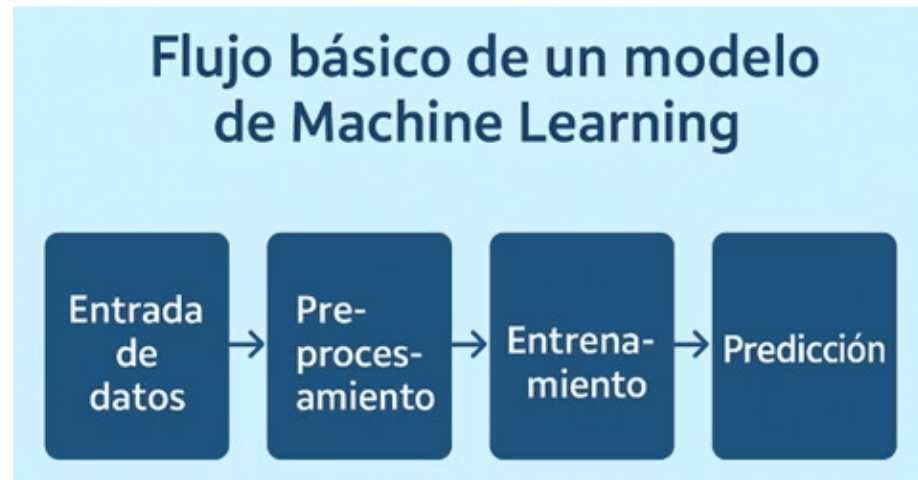


Imagen 3: *Flujo básico de un modelo de Machine Learning*

Entrada de datos → Preprocesamiento → Entrenamiento → Evaluación → Predicción

Obtención de datos

¿Recuerdas la clase de la sesión 1 sobre introducción a la arquitectura de datos, cuando hablamos de las fuentes de datos?

Los datos son el insumo fundamental para cualquier sistema de análisis o decisión. Pueden originarse en diversas fuentes como sensores IoT, formularios digitales, redes sociales, registros (logs) de sistemas, aplicaciones empresariales, entre otros.

Identificar correctamente estas fuentes es clave para diseñar una arquitectura eficiente y robusta.

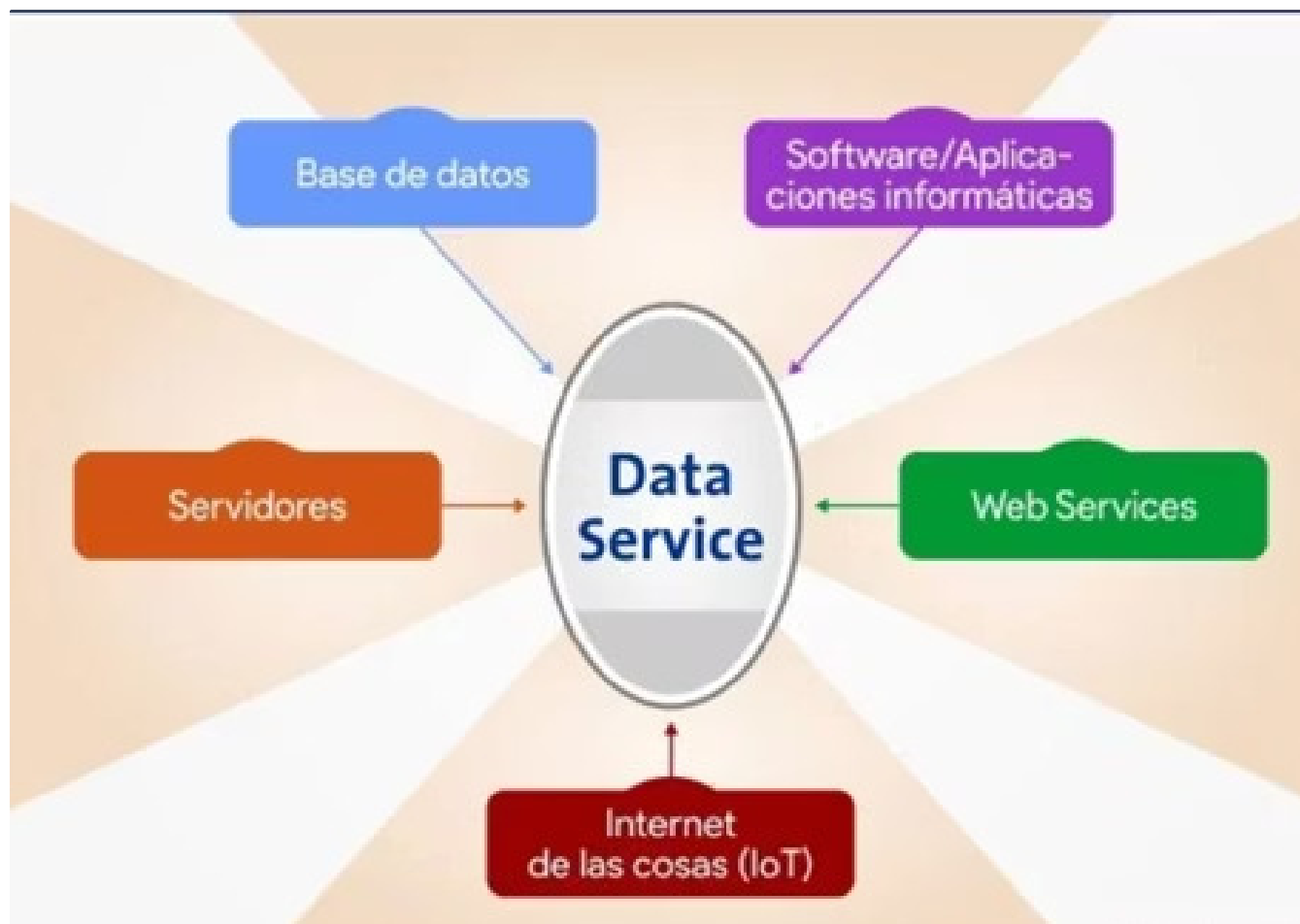


Imagen 4: Representación visual de obtención de datos desde múltiples fuentes.

Preprocesamiento y limpieza

En esta etapa, los datos crudos se transforman a un formato adecuado para el análisis. Esto incluye tareas como la eliminación de valores nulos, la codificación de variables categóricas, la normalización o estandarización de números, entre otras técnicas esenciales.

Aunque parezca sorprendente, el preprocesamiento y la limpieza de datos pueden consumir hasta el 80% del tiempo total en un proyecto de ciencia de datos o Machine Learning.

Este punto se abordará en mayor profundidad más adelante en esta misma sesión.



Imagen 5: La limpieza de datos es un proceso esencial en la Data Science y en Machine Learning. Consiste en resolver anomalías en conjuntos de datos (Datasets), para poder explotarlos después.

Entrenamiento del modelo

En esta etapa se selecciona un algoritmo (como regresión lineal, árboles de decisión o redes neuronales) y se entrena para que aprenda a reconocer patrones en los datos.

Este entrenamiento permite que el modelo haga predicciones o clasificaciones basadas en ejemplos previos.

Puedes ver ejemplos de esto en la vida cotidiana:

- Cuando consultas el pronóstico del tiempo, estás viendo el resultado de un modelo entrenado con datos meteorológicos.
- Si visitas una librería virtual, notarás que te recomienda libros en función de tus compras anteriores y tus preferencias de lectura.
- Incluso en aplicaciones de inversión, como las que analizan el comportamiento de criptomonedas, se utilizan modelos entrenados para predecir tendencias o sugerir decisiones.

¿Dónde más crees que se aplican estos modelos en tu vida diaria?

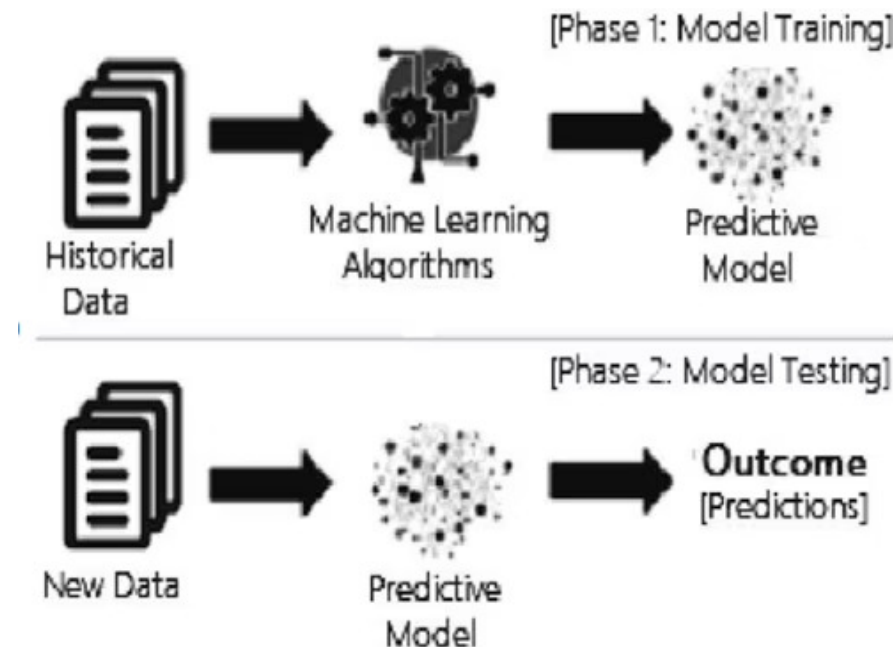


Imagen 6: La imagen muestra cómo un modelo aprende a partir de datos. Se entrena con ejemplos históricos, se ajusta para mejorar su precisión y luego se evalúa para asegurar que pueda hacer buenas predicciones con nuevos datos.

Evaluación del modelo

Una vez entrenado el modelo, es fundamental evaluar su desempeño utilizando métricas específicas que varían según el tipo de problema. Algunas de las más comunes incluyen la precisión, el error cuadrático medio (MSE) y el área bajo la curva ROC (ROC-AUC), entre otras.

Estas métricas serán analizadas en profundidad en las Sesiones 3 y 4 de este módulo, donde aprenderás a aplicarlas correctamente según el contexto.

Predicción y despliegue

Una vez que el modelo ha sido entrenado y evaluado satisfactoriamente, llega el momento de su aplicación en el mundo real. Esto implica utilizar el modelo para generar predicciones sobre nuevos datos, ya sea para recomendar productos, detectar fraudes, estimar ventas futuras, entre muchas otras aplicaciones prácticas.

Sin embargo, para que el modelo esté disponible fuera del entorno de desarrollo (por ejemplo, en una aplicación web o una herramienta de negocios), es necesario llevar a cabo un proceso conocido como despliegue. Esto puede implicar:

- Convertir el modelo en un servicio accesible a través de una API.
- Integrarlo dentro de una aplicación o flujo automatizado.
- Monitorearlo constantemente para asegurar que su desempeño no se degrade con el tiempo.

El despliegue es un paso clave para convertir un modelo de Machine Learning en una solución real y útil para la organización.

Ejemplo de despliegue: Netflix

Pensemos en el caso de Netflix. Una vez entrenado el modelo con datos de visualización de miles de usuarios (géneros preferidos, historial de vistas, calificaciones), se despliega en producción a través de una API.

Cada vez que un usuario inicia sesión:

- La app llama a la API del modelo.
- El modelo predice qué contenidos podrían gustarle.
- La plataforma muestra automáticamente esas recomendaciones en su dashboard.

Todo esto sucede en segundos, de forma invisible, y representa un ejemplo claro de modelo predictivo en producción.

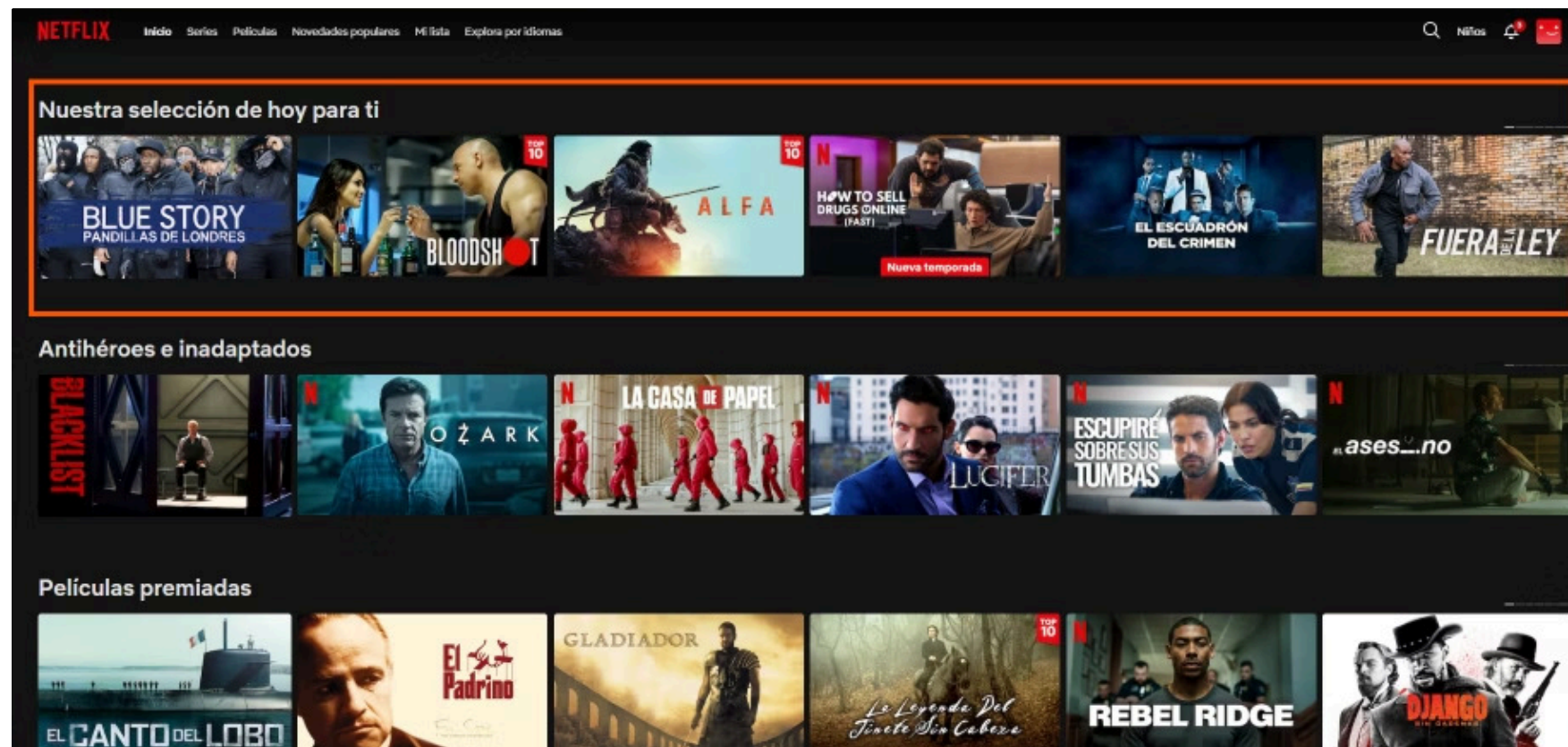


Imagen 7

Este proceso será abordado en profundidad en la **Sesión 5 del módulo**, donde aprenderás cómo llevar un modelo desde el laboratorio hasta un entorno de producción.

Comparación con la programación tradicional

Para entender mejor qué hace único al Machine Learning, es útil compararlo con la programación tradicional. Aunque ambos enfoques utilizan datos como insumo, difieren profundamente en su lógica y funcionamiento. La siguiente tabla resume las principales diferencias entre ambos paradigmas, destacando cómo el aprendizaje automático permite que los sistemas aprendan patrones a partir de los datos, en lugar de seguir instrucciones rígidas predefinidas.

Característica	Programación Tradicional	Machine Learning
Entrada	Datos + Reglas explícitas	Datos + Resultados esperados
Salida	Resultado	Reglas implícitas (modelo aprendido)
Adaptabilidad	Baja	Alta (aprende y mejora)
Ejemplo	Calcular impuesto con fórmula	Predecir si un cliente se dará de baja

Tipos de Aprendizaje de Máquina

Una vez comprendido qué es el aprendizaje automático y su relevancia, es fundamental conocer cómo se clasifica según la forma en que el modelo accede y aprende de los datos. Existen principalmente dos enfoques:

Aprendizaje supervisado

En este enfoque, el modelo aprende a partir de un conjunto de datos que incluye tanto las entradas como las salidas esperadas. Se le "supervisa" durante el entrenamiento, indicándole cuál es la respuesta correcta para cada caso.

Ejemplo: Un modelo que predice si un correo es spam o no, utilizando correos etiquetados previamente como "spam" o "no spam".

Tareas comunes: Clasificación (spam/no spam) y regresión (predecir el precio de una casa).



Imagen 8

Este tipo de aprendizaje es central en el desarrollo de modelos que requieren evaluación cuantitativa. Profundizaremos en modelos supervisados de regresión en la Sesión 3, y de clasificación en la Sesión 4 del módulo.

Aprendizaje no supervisado

En este caso, los datos no contienen etiquetas ni respuestas correctas. El objetivo del modelo es descubrir estructuras, patrones o agrupaciones naturales en los datos, sin intervención humana directa.

Ejemplo: Agrupar clientes según sus hábitos de compra para personalizar campañas de marketing.

Tareas comunes: Clustering (agrupamiento), reducción de dimensiones, detección de anomalías.



Imagen 9

Este enfoque es muy útil en etapas exploratorias o cuando no se dispone de información categorizada.

Comparación entre tipos de aprendizaje

Característica	Aprendizaje Supervisado	Aprendizaje No Supervisado
Datos de entrada	Etiquetados (entrada + salida conocida)	No etiquetados (solo entrada)
Objetivo	Predecir un valor o clase	Encontrar estructura o patrones ocultos
Ejemplos típicos	Clasificación de correos, regresión de precios	Segmentación de clientes, detección de fraudes
Aplicaciones comunes	Predicción, detección de enfermedades	Agrupación, análisis exploratorio

Si quieres reforzar lo aprendido, te recomiendo este video del canal DotCSV, donde se explican de forma clara y sencilla las diferencias entre el aprendizaje supervisado y no supervisado, con ejemplos muy ilustrativos.

Aprendizaje por refuerzo

Existe un tercer enfoque, conocido como aprendizaje por refuerzo (Reinforcement Learning), en el que el modelo aprende a través de ensayo y error, recibiendo recompensas o penalizaciones en función de sus acciones. Es ampliamente utilizado en robótica, juegos y control de sistemas autónomos.

Este tipo de aprendizaje no será abordado en profundidad en este módulo, pero es importante conocerlo como parte del ecosistema de ML.

En las próximas sesiones trabajaremos con ejemplos prácticos tanto de clasificación como de regresión, que forman parte del aprendizaje supervisado.

El aprendizaje no supervisado, especialmente técnicas de agrupamiento, puede aplicarse en la etapa de preprocesamiento exploratorio o como complemento de modelos supervisados.

Tareas de regresión y clasificación

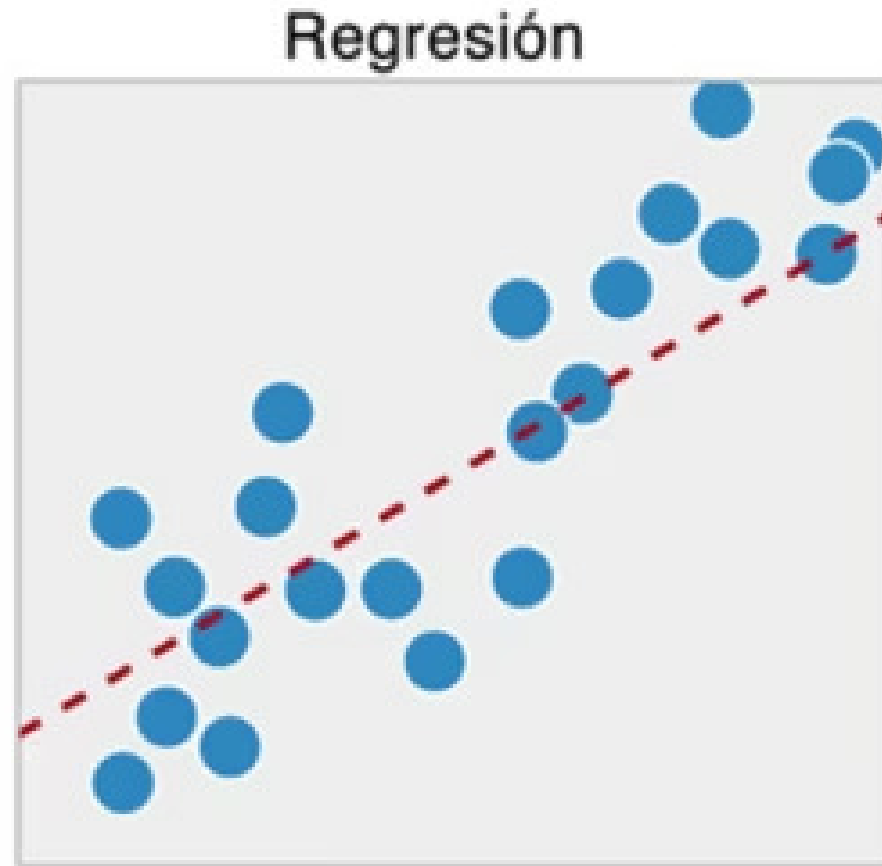
Dentro del aprendizaje supervisado, existen dos grandes tipos de tareas que permiten abordar distintos tipos de problemas: regresión y clasificación. Comprender sus diferencias es esencial para elegir el modelo adecuado según el tipo de salida que queremos obtener.

Regresión

La regresión se utiliza cuando el objetivo es predecir un valor numérico continuo. Se aplica en situaciones donde queremos estimar cantidades como precios, temperaturas, tiempos o cualquier variable que varíe en una escala numérica.

Un ejemplo sería predecir el precio de una casa a partir de características como los metros cuadrados, la ubicación, el número de habitaciones y baños, entre otros factores.

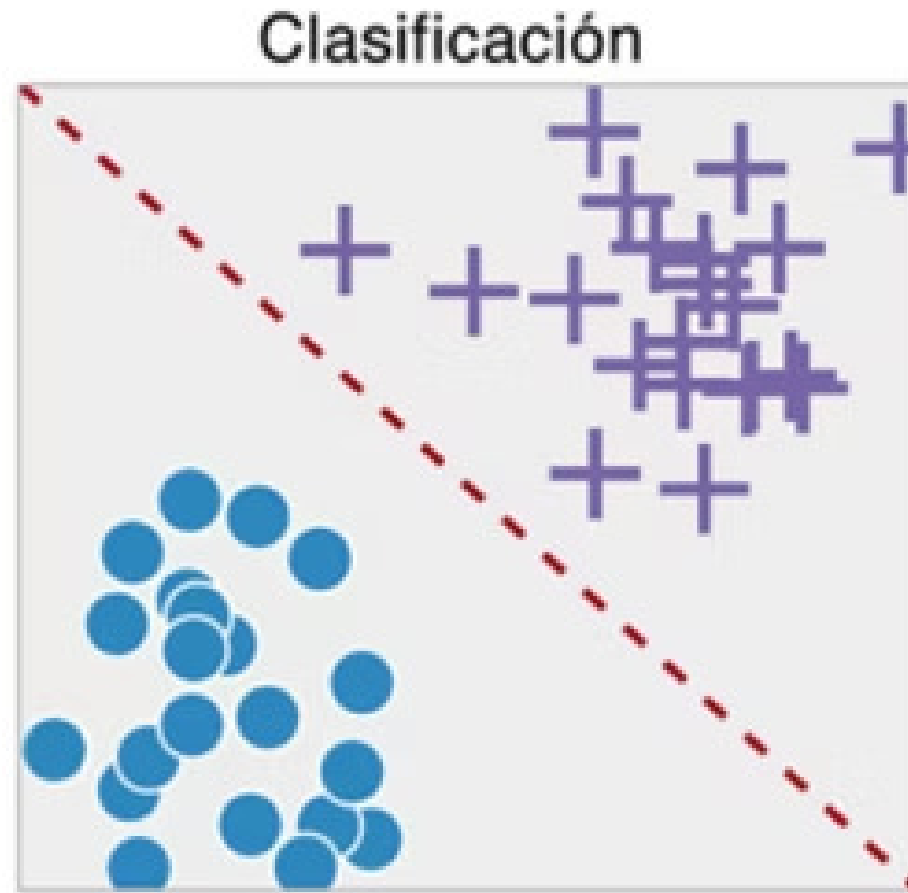
Este tipo de tarea se aborda en profundidad en la Sesión 3, donde exploraremos diferentes modelos de regresión (lineal, polinómica, etc.) y sus métricas de evaluación.



Clasificación

La clasificación se utiliza cuando el objetivo es predecir una categoría o clase. En lugar de entregar un valor numérico, el modelo selecciona una etiqueta entre un conjunto de posibles opciones.

Determinar si un correo electrónico es spam o no spam, basándose en su contenido, título y origen sería un ejemplo de Clasificación.



Tipos de clasificación

Tipo	Descripción	Ejemplo
Binaria	Solo existen dos clases posibles.	Correo spam / no spam
Multiclase	Existen más de dos categorías posibles y se asigna solo una por instancia.	Clasificación de tipos de flores (rosa, tulipán...)
Multi-etiqueta	Se pueden asignar múltiples etiquetas a una misma instancia.	Imagen etiquetada como "playa", "atardecer", "vacaciones"

Dato importante: Cada tipo de clasificación requiere diferentes métricas y enfoques de evaluación, los cuales serán tratados en detalle durante la Sesión 4.

Comparación entre regresión y clasificación

Tipo de tarea	Objetivo principal	Salida esperada	Ejemplo típico
Regresión	Predecir un valor continuo	Número real (float)	Precio de una vivienda
Clasificación	Predecir una clase o categoría	Etiqueta/categoría	Tipo de cliente (fiel/nuevo)

¿Cómo elegir entre regresión o clasificación?

Una buena forma de decidir qué tipo de tarea estás enfrentando es preguntarte:

- ¿La salida que espero es un número? → Regresión
- ¿La salida que espero es una categoría? → Clasificación

En algunos casos, un mismo problema puede plantearse como regresión o clasificación, dependiendo de cómo se estructuren los datos.

Por ejemplo, predecir la edad exacta de una persona sería una regresión, pero agruparla en rangos etarios (18–25, 26–35, etc.) sería clasificación multiclase.

En resumen

Las tareas de **regresión** y **clasificación** son los pilares del **aprendizaje supervisado**. Elegir correctamente entre ambas y entender sus características es el primer paso para diseñar modelos robustos y adecuados para el problema a resolver.

Preprocesamiento de datos

El preprocesamiento de datos es una de las etapas más importantes (y muchas veces subestimadas) en cualquier proyecto de Machine Learning. Incluso con modelos sofisticados, si los datos no han sido correctamente preparados, los resultados serán poco confiables o incluso erróneos.

¿Por qué es necesario?

- Los modelos de ML no pueden trabajar directamente con datos crudos. Necesitan que la información esté limpia, estructurada y, en muchos casos, transformada numéricamente.
- Preprocesar adecuadamente puede marcar la diferencia entre un modelo mediocre y uno preciso.
- Recuerda que se estima que entre un 60% y un 80% del tiempo total de un proyecto de ciencia de datos se dedica al preprocesamiento.

PRINCIPALES TÉCNICAS DE PREPROCESAMIENTO

Las variables categóricas (como "rojo", "verde", "azul") deben transformarse a valores numéricos para que puedan ser utilizadas por los modelos. Las técnicas más comunes son:

Técnica	Descripción	Implementación en Python
Label Encoding	Asigna un número entero a cada categoría	LabelEncoder de scikit-learn
One-Hot Encoding	Crea una nueva columna binaria para cada categoría	OneHotEncoder de scikit-learn o pandas.get_dummies()
Variables Dummy	Igual que One-Hot, pero elimina una columna para evitar colinealidad	get_dummies(drop_first=True) de pandas

¿Cuándo usar una u otra?

- LabelEncoder (sklearn): útil para modelos basados en árboles de decisión, donde el orden numérico no distorsiona los resultados.
- OneHotEncoder (sklearn): preferido en regresión lineal o redes neuronales, donde los modelos interpretan valores numéricos como jerárquicos.
- pandas.get_dummies(): muy práctico en exploración y prototipado rápido, especialmente útil cuando se trabaja con DataFrames.

Escalado de datos

Cuando los datos numéricos están en escalas diferentes (por ejemplo, centímetros vs. grados), el modelo puede dar más importancia a una variable solo por tener números más grandes. Para evitar esto, se aplica escalado o normalización.

Técnica	Descripción
MinMax Scaling	Escala los valores a un rango fijo, generalmente [0, 1]
Estandarización	Centra los datos en media 0 con desviación estándar 1 (Z-score)

El escalado es especialmente importante en modelos que utilizan distancias (como KNN, clustering), que veremos en la siguiente sección.

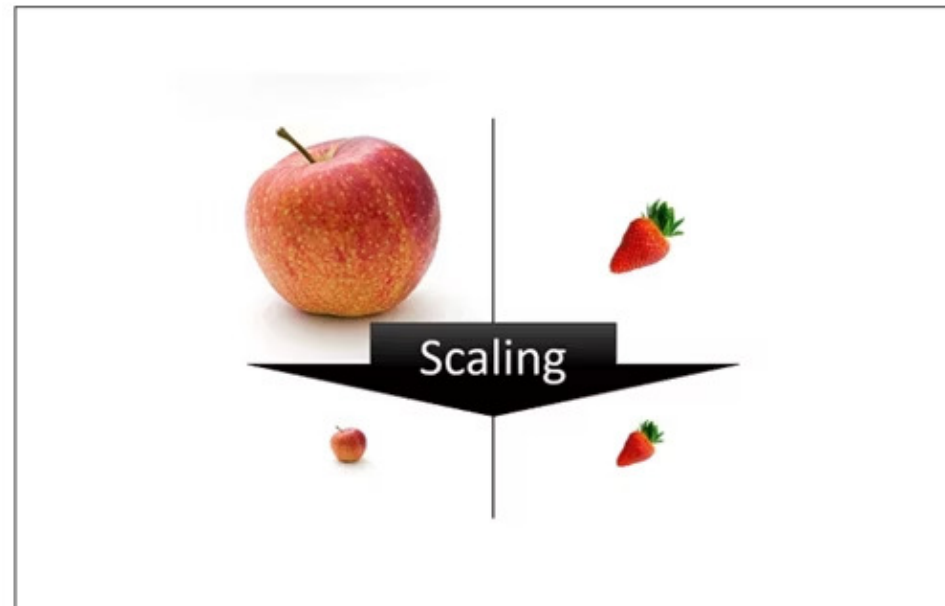
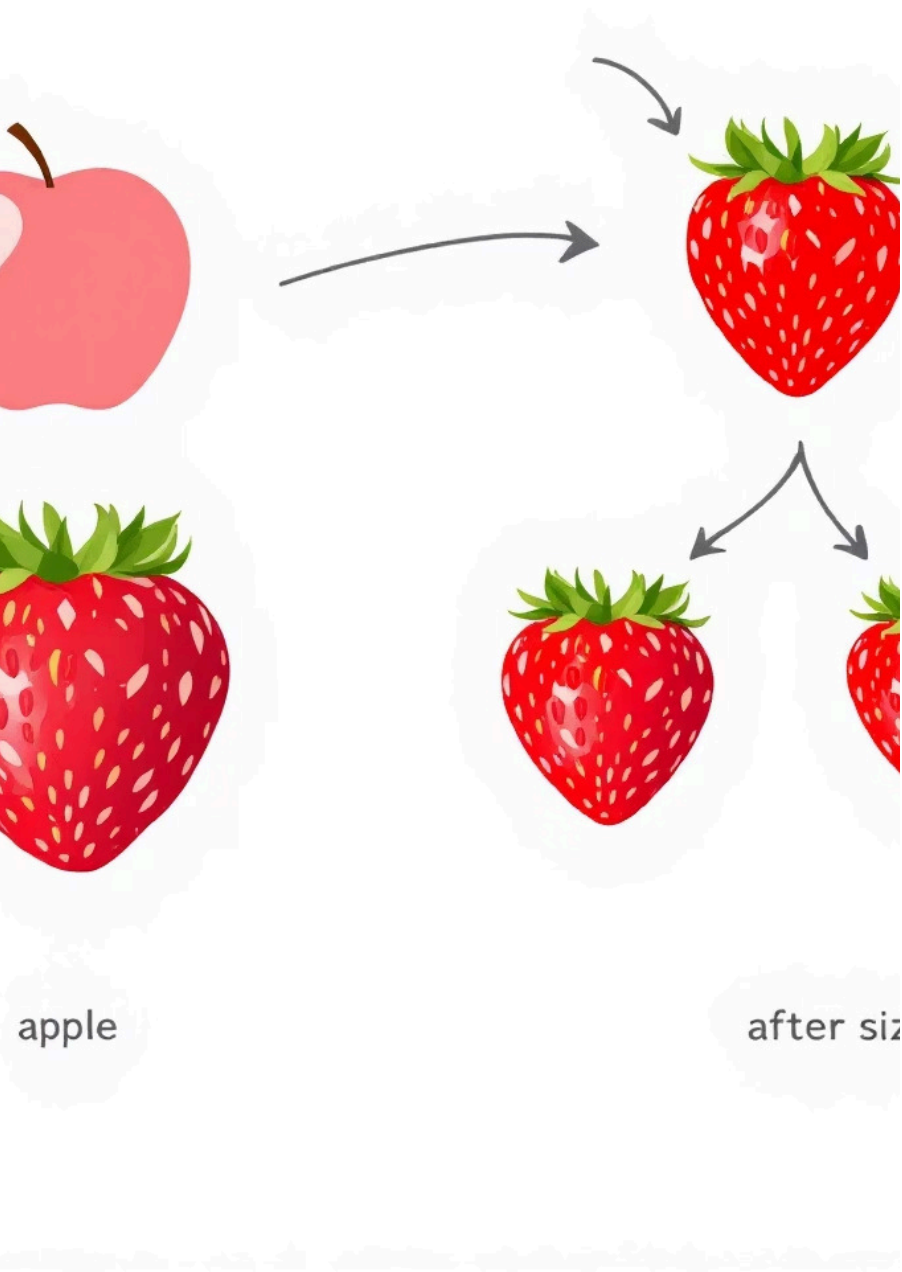


Imagen 14: Representación visual del escalado de variables



Importancia del escalado en variables

A la izquierda vemos una manzana y una frutilla en sus tamaños reales. Si un algoritmo de Machine Learning comparara estas imágenes en función del tamaño, daría más peso a la manzana simplemente por su escala.

Sin embargo, el tamaño no necesariamente representa mayor importancia, solo es una característica.

Tras aplicar scaling (escalado), ambas frutas aparecen en proporciones comparables, permitiendo al modelo tratarlas con equidad.

Este proceso es exactamente lo que hacemos cuando escalamos variables numéricas como peso, edad o ingresos en un modelo de ML.

Concepto de Distancia

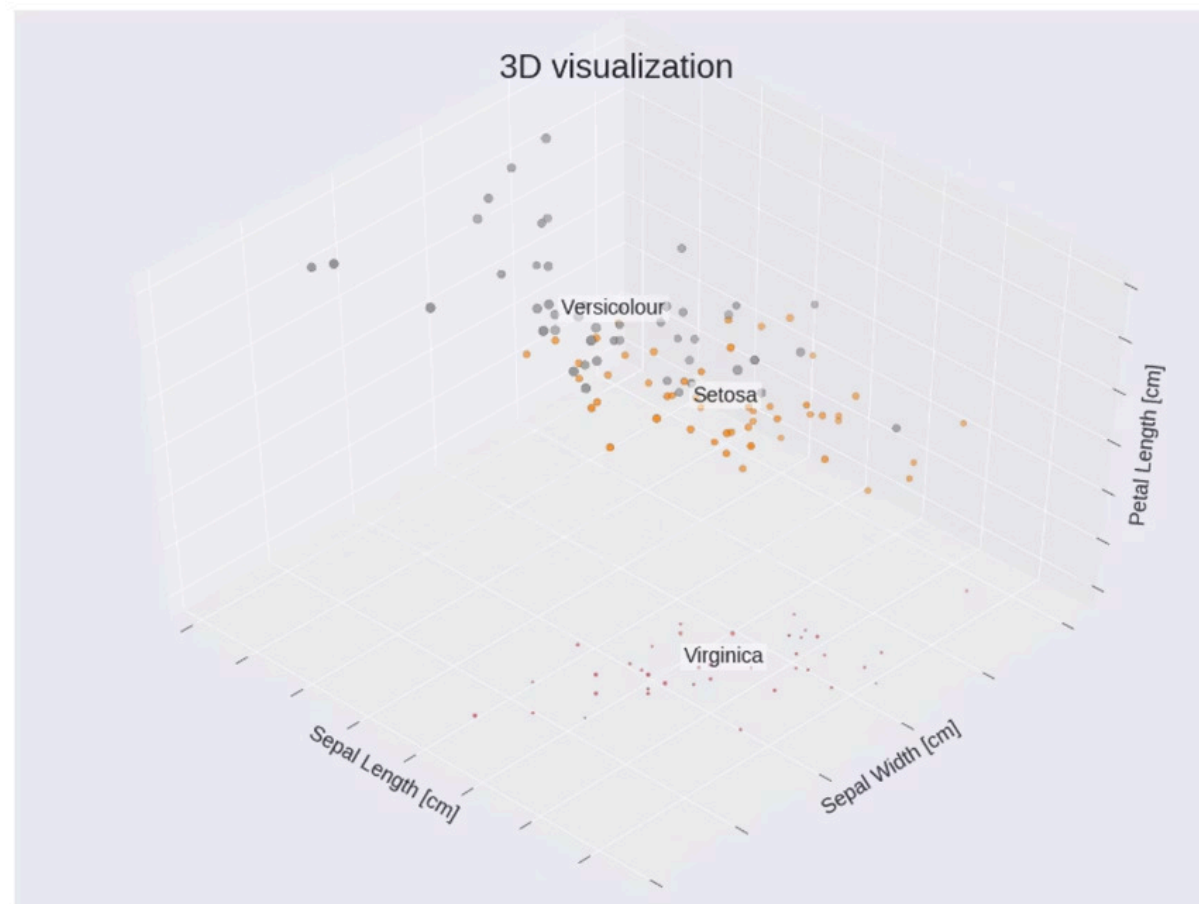
Algunos algoritmos de Machine Learning, como k-Nearest Neighbors (KNN) —un algoritmo de aprendizaje supervisado— o los métodos de clustering —típicamente usados en aprendizaje no supervisado—, necesitan una forma de "medir la cercanía" entre los datos. Para esto utilizan métricas de distancia, que permiten comparar cuánto se parecen o se diferencian dos puntos del espacio de características.

¿Qué significa "distancia" en Machine Learning?

En un espacio de datos, cada observación (fila del dataset) puede verse como un punto definido por sus características numéricas. Por ejemplo, una flor del dataset Iris puede representarse como un punto en un espacio de 4 dimensiones:

- Largo del sépalo
- Ancho del sépalo
- Largo del pétalo
- Ancho del pétalo

La distancia entre dos observaciones nos indica qué tan similares son.



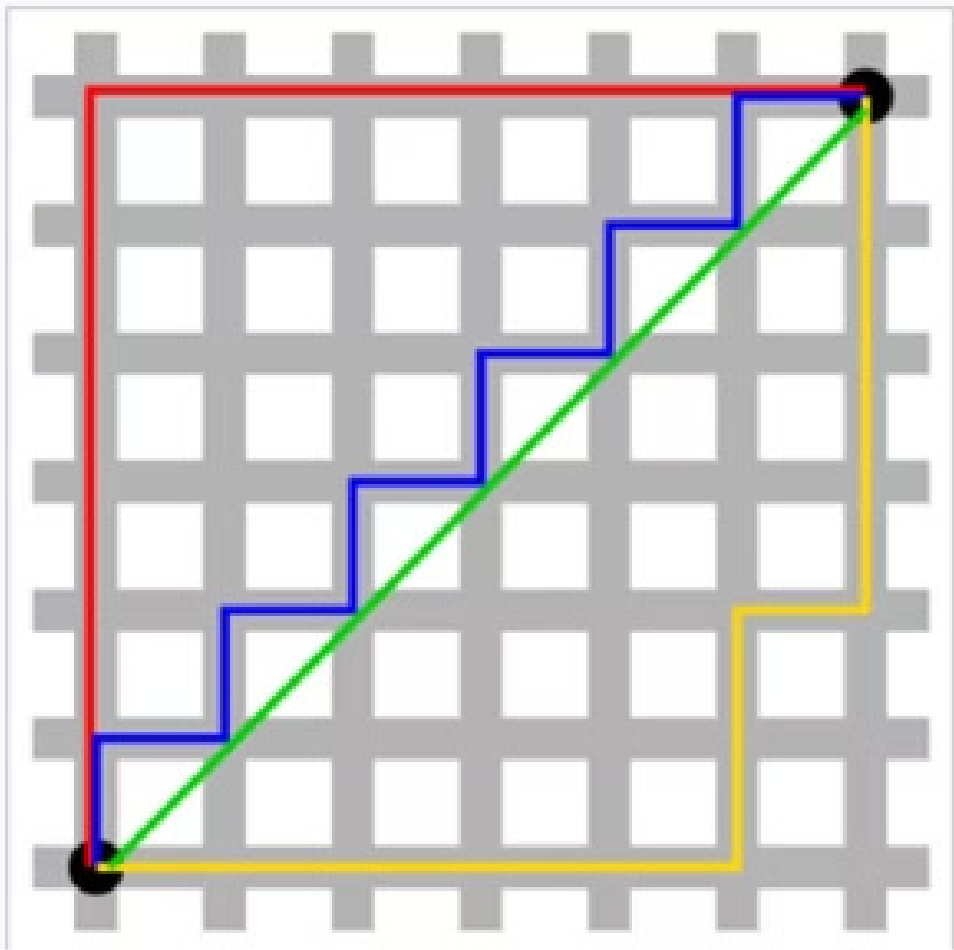
Cada punto en el gráfico representa una flor, posicionada según sus medidas de sépalo y pétalo. Esta vista en 3D nos permite imaginar el espacio de características donde se agrupan las clases, y entender cómo la distancia entre puntos refleja la similitud entre observaciones.

Tipos de distancia más comunes

Tipo de distancia	Fórmula matemática	Característica principal
Euclidiana	$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$	Distancia "en línea recta" entre dos puntos (como con una regla)
Manhattan	$ x_1 - x_2 + y_1 - y_2 $	Suma de las diferencias absolutas (como moverse en calles rectas tipo cuadrícula)



Distancia euclidiana 2D



La imagen muestra diferentes formas de calcular la distancia entre dos puntos sobre una cuadrícula. La línea verde representa la distancia euclidiana (el camino recto más corto), mientras que las rutas en rojo, azul y amarillo simulan distintos recorridos tipo “L” que corresponden a la distancia de Manhattan. Esta métrica suma los pasos horizontales y verticales necesarios, como si uno caminara por calles en una ciudad.

¿Por qué es importante conocer esto?

Algunos algoritmos basan su lógica en la cercanía entre puntos. Por ejemplo:

- En KNN, para clasificar una observación nueva, el algoritmo busca las K observaciones más cercanas según la distancia (usualmente euclidiana).
- En clustering (como K-means), se agrupan puntos que están cerca entre sí según estas métricas.

Si tus variables están en diferentes escalas, la distancia puede distorsionarse. Por eso aplicamos escalado antes de usar estos algoritmos (como vimos en la sección anterior).

Dimensiones y la Maldición de la Dimensionalidad

En Machine Learning, cada variable o característica de un dataset representa una dimensión en el espacio de datos. Por ejemplo:

Observación	Largo del pétalo	Ancho del pétalo	Largo del sépalo	Ancho del sépalo
Flor A	3.5	1	5.1	1.4

Esta tabla muestra un espacio de 4 dimensiones. Ahora imagina un dataset con 100, 500 o incluso 10.000 variables. Aunque parezca que "más información es mejor", esto no siempre es cierto. A partir de cierto punto, añadir más dimensiones empeora el rendimiento de los modelos. Este fenómeno se conoce como la Maldición de la Dimensionalidad.

¿Qué es la Maldición de la Dimensionalidad?

La maldición de la dimensionalidad ocurre cuando el número de características de un conjunto de datos crece tanto que los modelos tienen dificultades para generalizar.

¿Por qué sucede?

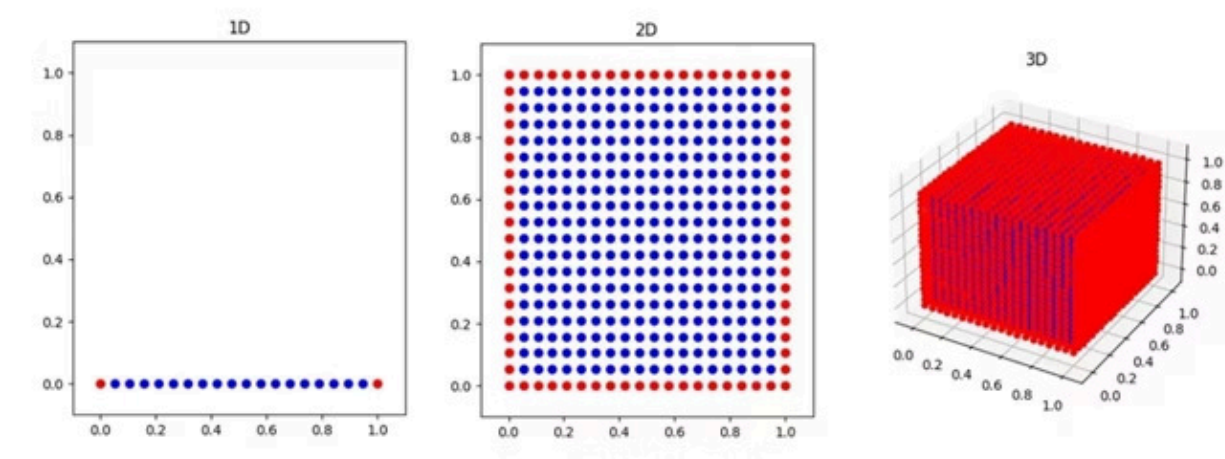
- A medida que aumentan las dimensiones, los datos se dispersan en el espacio: hay más combinaciones posibles de valores, pero la densidad de datos se diluye.
- Se necesitan más muestras para cubrir el espacio de forma efectiva.
- Algunas variables pueden aportar ruido en lugar de información útil.
- Muchas métricas (como la distancia Euclidiana) pierden su significado cuando hay demasiadas dimensiones.

Imagina un cubo de 1 metro cúbico. Ahora aumentemos las dimensiones:

Dimensiones	Volumen necesario para cubrir el “espacio” con datos
1D (una línea)	1 m
2D (un cuadrado)	1 m²
3D (un cubo)	1 m³
10D	1,000,000,000 m^10

El espacio crece **exponencialmente**, y tus datos se vuelven **escasos**. Los algoritmos no **“ven”** patrones porque no hay suficientes ejemplos cercanos para aprender.

The Curse of Dimensionality



La imagen muestra cómo, al aumentar las dimensiones (de 1D a 3D), el espacio de datos crece exponencialmente. Aunque se mantienen el mismo número de puntos, estos se dispersan cada vez más, haciendo que los extremos (en rojo) dominen el espacio. Esto ilustra cómo los datos se vuelven más escasos y menos representativos en espacios de alta dimensión

¿Cómo lo resolvemos?

La solución no es eliminar variables al azar, sino aplicar técnicas que reduzcan la dimensionalidad de forma **inteligente**, conservando la información relevante.

Técnicas comunes:

Técnica	Descripción	Uso común
PCA (Análisis de Componentes Principales)	Proyecta los datos en un nuevo espacio con menos dimensiones, manteniendo la mayor varianza posible	Exploración, visualización, preprocesamiento antes de <u>clustering</u>
Selección de características	Conserva solo las variables más relevantes según criterios estadísticos o del modelo	Clasificación, regresión, modelos interpretables

¿Cuándo aplicar reducción de dimensionalidad?

Cuando el número de variables es muy alto y el modelo rinde mal.

Si notas **overfitting** (el modelo aprende ruido).

Para visualizar datos complejos en 2D o 3D.

Antes de aplicar algoritmos que usan distancias (como KNN o clustering).

Glosario de términos

Término	Definición
Aprendizaje supervisado	Tipo de aprendizaje en el que se entrena un modelo con datos de entrada y salidas conocidas (etiquetas).
Aprendizaje no supervisado	Tipo de aprendizaje que busca patrones en datos sin etiquetas conocidas.
Clasificación	Tarea supervisada en la que el modelo predice una clase o categoría.
Regresión	Tarea supervisada en la que el modelo predice un valor continuo.
Preprocesamiento	Conjunto de técnicas aplicadas a los datos crudos para que puedan ser utilizados por modelos de ML.
Codificación	Transformación de variables categóricas en valores numéricos.
Escalado	Transformación de variables numéricas a una misma escala, para evitar sesgo en modelos sensibles a magnitud.
Distancia Euclidiana	Métrica que mide la distancia "en línea recta" entre dos puntos.
Distancia Manhattan	Métrica que mide la suma de diferencias absolutas entre coordenadas, como en una cuadrícula.
Dimensión	Cada característica numérica de un dataset representa una dimensión en el espacio de datos.
Maldición de la dimensionalidad	Fenómeno donde un alto número de dimensiones dificulta el aprendizaje del modelo.
PCA (Análisis de Componentes Principales)	Técnica de reducción de dimensionalidad que proyecta los datos en un espacio más compacto.
Varianza	Medida estadística que refleja cuánto se dispersan los datos respecto a la media.