

Intro al Machine Learning

Dr. Ángel Díaz Pacheco





Tipos de aprendizaje

Aprendizaje Supervisado (Supervised Learning)

Definición:

Es un enfoque en el que el modelo aprende a partir de un conjunto de datos etiquetado. Es decir, cada entrada del conjunto de datos está asociada a una salida deseada (etiqueta o valor).

Objetivo:

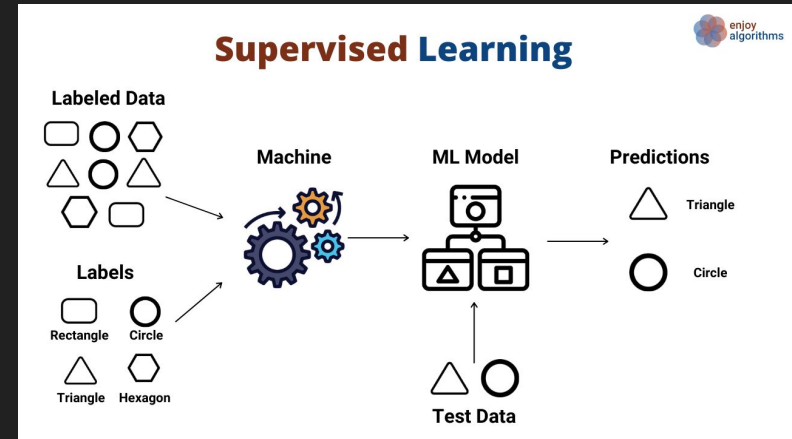
Aprender una función que, a partir de los datos de entrada, pueda predecir correctamente la salida.

Tipos de tareas:

- Clasificación: predecir una categoría (e.g., detectar si un correo es spam o no).
- Regresión: predecir un valor continuo (e.g., predecir el precio de una casa).

Ejemplos comunes:

- Detección de fraudes bancarios.
- Reconocimiento de imágenes etiquetadas (gato, perro, coche...).
- Predicción del rendimiento escolar con base en características de los estudiantes.



Aprendizaje No Supervisado (Unsupervised Learning)

Definición:

No se cuenta con etiquetas en los datos. El modelo debe encontrar patrones, estructuras o agrupamientos por sí mismo.

Objetivo:

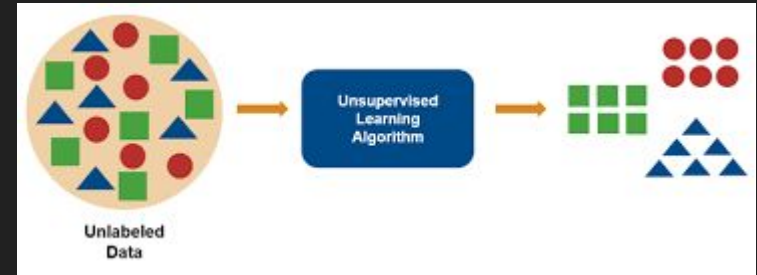
Descubrir relaciones ocultas o estructuras subyacentes en los datos.

Tareas comunes:

- Clustering (agrupamiento): agrupar datos similares (e.g., segmentación de clientes).
- Reducción de dimensionalidad: simplificar los datos preservando su estructura (e.g., PCA).
- Detección de anomalías: identificar comportamientos atípicos.

Ejemplos comunes:

- Agrupamiento de usuarios según su comportamiento en una tienda en línea.
- Análisis de temas en un conjunto de textos (topic modeling).
- Detección de errores en sensores industriales.



supervised learning

Input data



Annotations

These are
apples



Model



Prediction

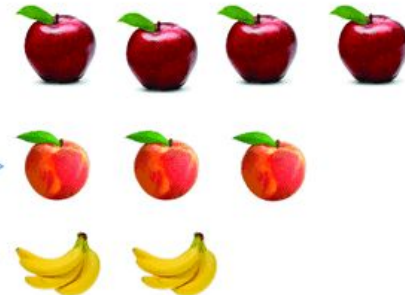
Its an
apple!

unsupervised learning

Input data



Model



Aprendizaje por Refuerzo (Reinforcement Learning)

Definición:

Un agente aprende a tomar decisiones realizando acciones en un entorno, recibiendo recompensas o penalizaciones según el resultado de sus acciones.

Objetivo:

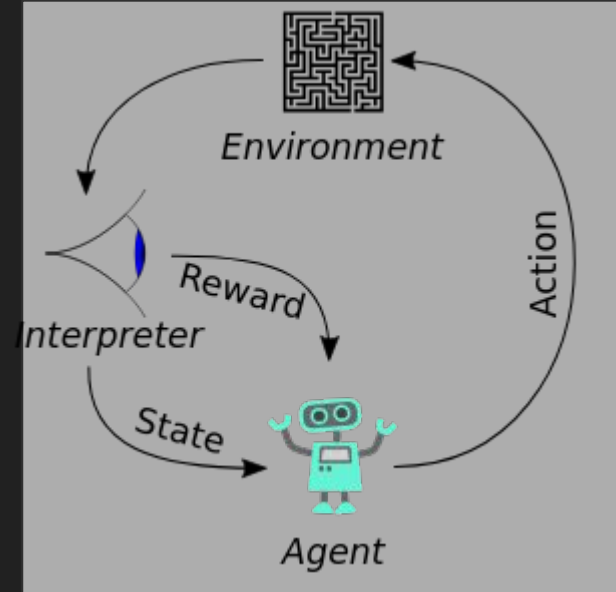
Maximizar la recompensa acumulada a lo largo del tiempo aprendiendo una política óptima de acción.

Componentes clave:

- Agente: el que toma decisiones.
- Entorno (environment): donde actúa el agente.
- Recompensa: retroalimentación que guía el aprendizaje.
- Política: estrategia que sigue el agente para tomar decisiones.

Ejemplos comunes:

- Juegos (ajedrez, Go, videojuegos).
- Robots aprendiendo a caminar o manipular objetos.
- Optimización de rutas de entrega.
- Sistemas de recomendación que se adaptan al usuario.



Algoritmos en cada caso

Supervised learning

- Regresión lineal/logística
- Árboles de decisión.
- SVM

Unsupervised learning

- k-means
- DBSCAN
- Clustering jerárquico

Reinforcement learning

- Q-Learning.
- Deep Q Network (DQN).
- Proximal Policy Optimization (PPO).

Comparación General

Característica	Supervisado	No Supervisado	Por Refuerzo
¿Hay etiquetas?	Sí	No	No (pero hay recompensas)
Tipo de salida	Conocida	Desconocida	Política óptima
Ejemplos de tareas	Clasificación, regresión	Clustering, reducción	Toma de decisiones
Aplicaciones	Diagnóstico médico, predicción financiera	Segmentación de clientes, análisis de temas	Juegos, robótica, optimización de procesos

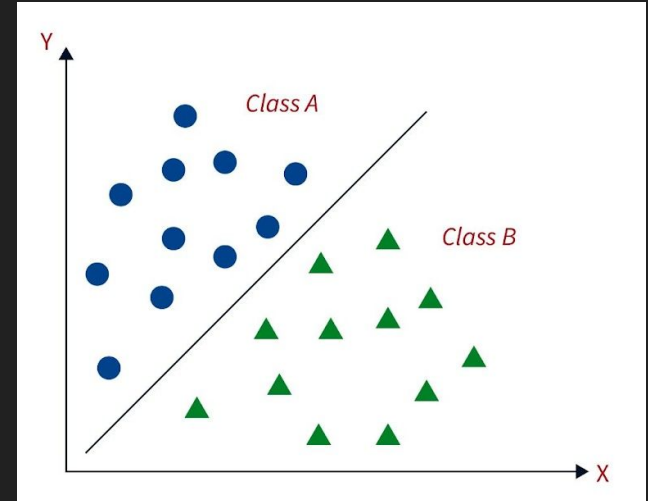


Clasificación y regresión

Clasificación (Classification)

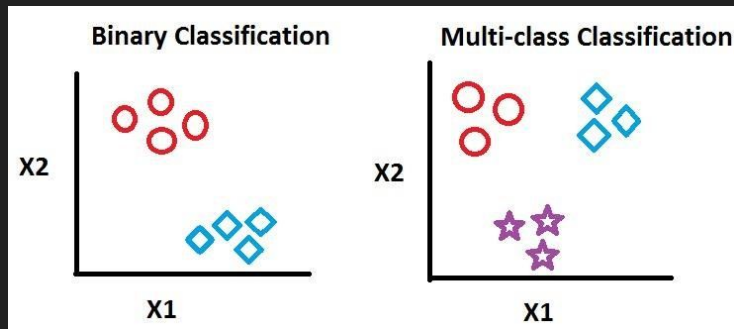
Definición:

La clasificación consiste en predecir una etiqueta o clase discreta a partir de los datos de entrada. El modelo aprende a asignar cada instancia de entrada a una de un conjunto finito de clases.



Tipos de clasificación:

- Binaria: dos clases (e.g., spam / no spam).
- Multiclase: más de dos clases (e.g., tipos de flores: rosa, lirio, margarita).
- Multietiqueta (multi-label): una instancia puede pertenecer a varias clases a la vez.



Ejemplos:

- Diagnóstico médico: sano vs enfermo.
- Reconocimiento facial: identificar a una persona entre varias.
- Clasificación de correos: spam o no spam.
- Clasificación de sentimientos: positivo, negativo o neutro.

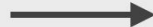
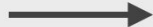
Métricas de evaluación (accuracy)

AN EXAMPLE OF A CLASSIFIER THAT DETECTS CATS
WITH A LABEL OF "POSITIVE" OR "NEGATIVE"

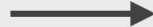
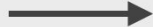
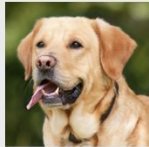
INPUT

CLASSIFIER

OUTPUT

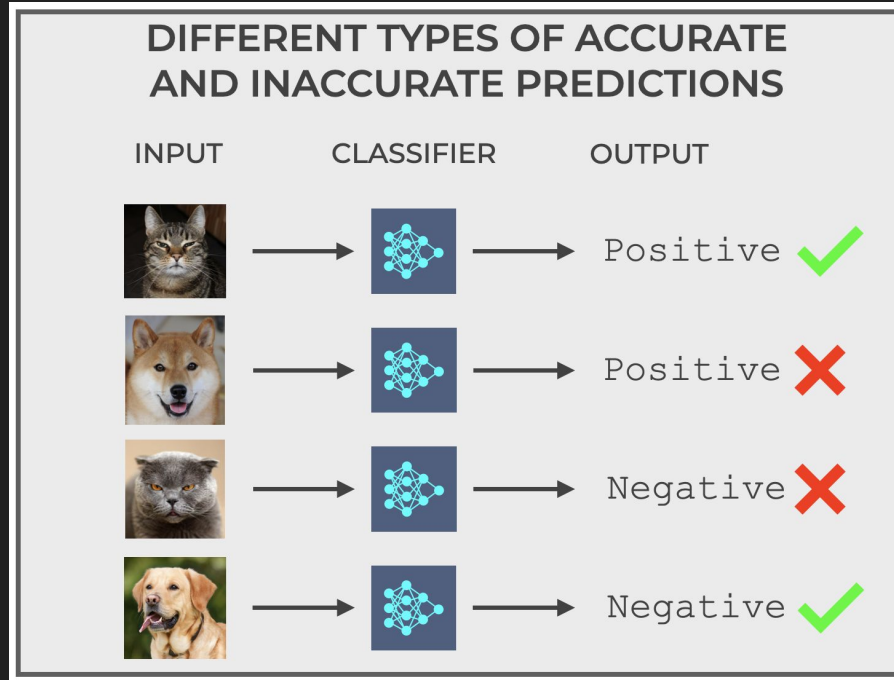


Positive



Negative

Métricas de evaluación (accuracy)







Matriz de confusión

number of correct predictions
number of total predictions

$$\frac{TP + TN}{TP + TN + FP + FN}$$

$$\frac{40 + 35}{40 + 35 + 10 + 15} = \frac{75}{100} = .75$$

		Predicted	
		CAT	NOT CAT
Actual	CAT	 40	 10
	NOT CAT	 15	 35

Pros y contras de usar Accuracy

Pros:

- Simplicidad y comprensibilidad
- Eficacia en conjuntos de datos balanceados
- Ideal para la evaluación inicial
- Visión general integral

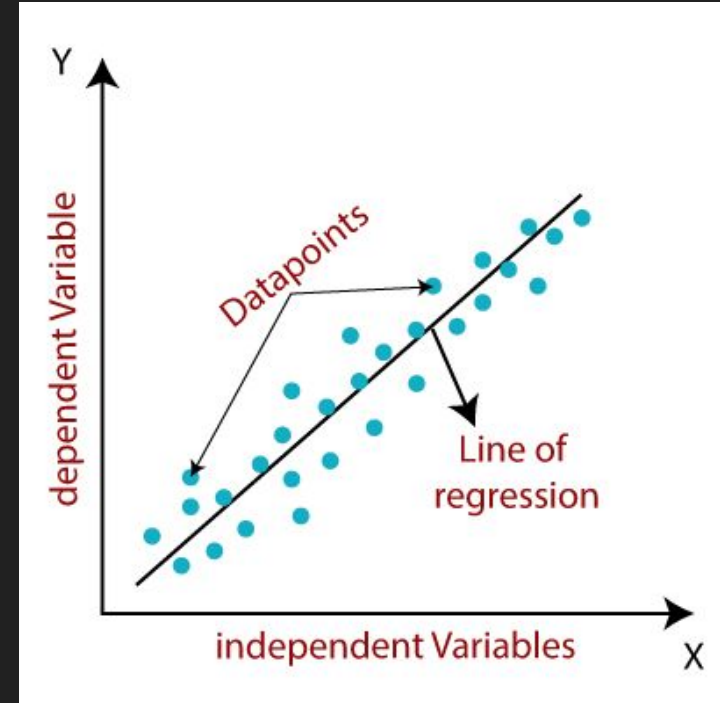
Contras:

- Engañoso en conjuntos de datos desequilibrados
- No diferencia entre tipos de error
- No tiene en cuenta el coste de los errores
- Poco informativo para problemas multiclase
- Falta de comprensión de la confianza en las predicciones
- A menudo fomenta el sobreajuste

Regresión (Regression)

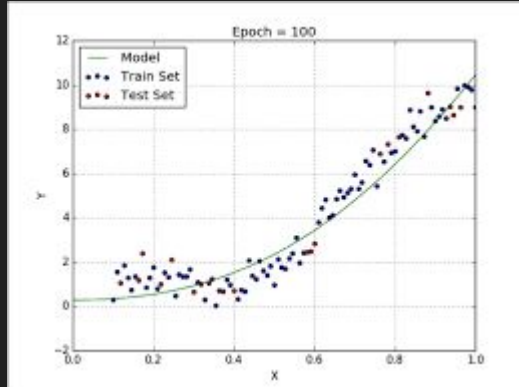
Definición:

La regresión busca predecir un valor numérico continuo a partir de variables de entrada. El modelo aprende una función que mapea entradas a salidas continuas.



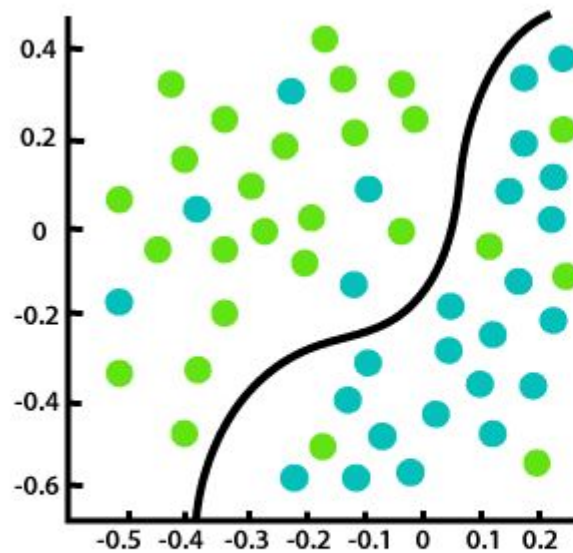
Tipos de regresión:

- Lineal: relación lineal entre variables ($y = mx + b$).
- Polinómica: relación no lineal modelada con polinomios.
- Regresión múltiple: múltiples variables independientes.
- Regresión logística: para clasificación binaria (aunque es un modelo de regresión).

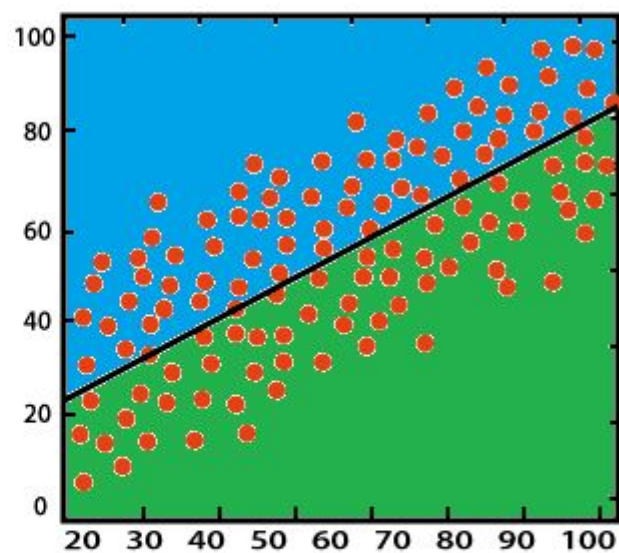


Ejemplos:

- Predicción del precio de una vivienda en función del número de habitaciones y ubicación.
- Pronóstico del valor de acciones financieras.
- Estimación del tiempo que tomará realizar una tarea.
- Predicción de la demanda eléctrica.



Classification

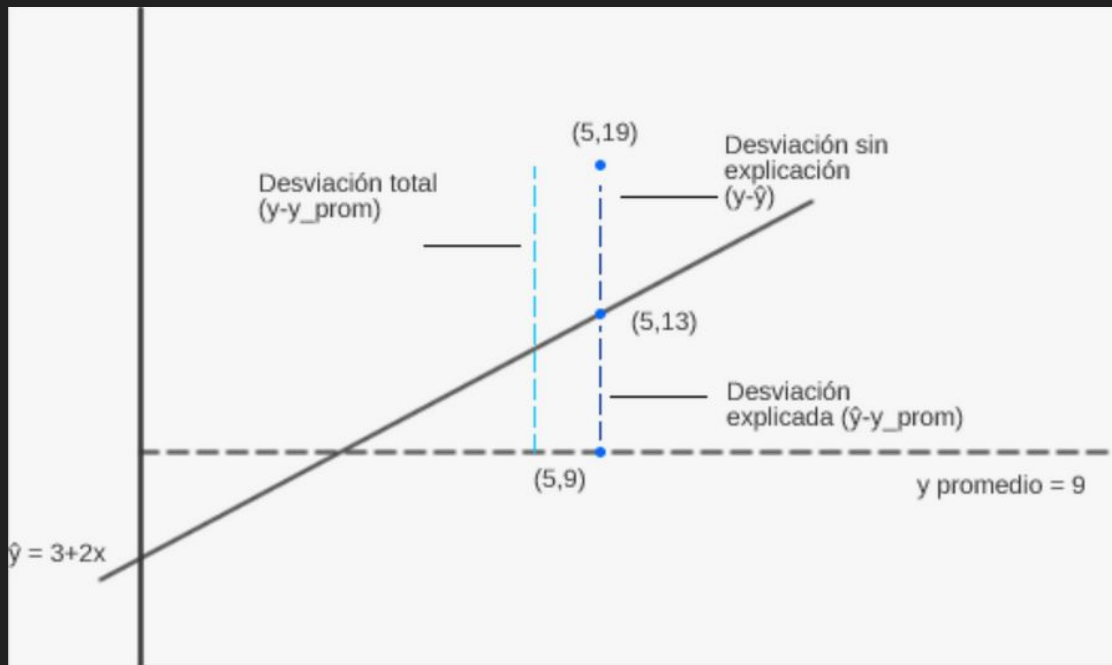


Regression

Coeficiente de determinación o R^2

El coeficiente de determinación es la cantidad de variación en y que se explica por la recta de regresión. Se calcula como:

$$r^2 = \frac{\text{variacionExplicada}}{\text{VariacionTotal}}$$

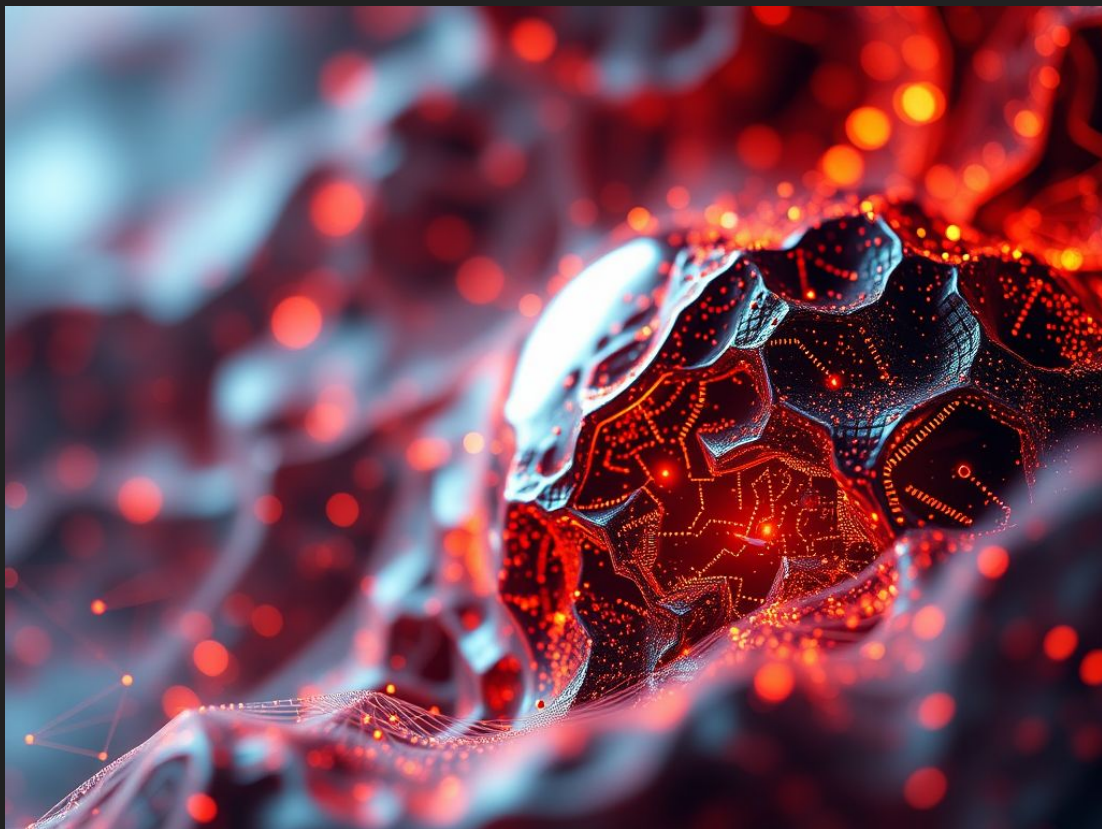


$$\begin{aligned} (\text{variación total}) &= (\text{variación explicada}) + (\text{variación sin explicar}) \\ \sum (y - \bar{y})^2 &= \sum (\hat{y} - \bar{y})^2 + \sum (y - \hat{y})^2 \end{aligned}$$

Root Mean Square Error o RMSE

RSME es la raíz cuadrada del valor del error cuadrático medio. Se utiliza para determinar la diferencia cuadrática media entre el valor predicho y el valor real de una característica o variable.

$$RSME = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$



Representación de Datos en Aprendizaje Automático

Representación de Datos en Aprendizaje Automático

 ¿Qué es?

Es la forma en la que estructuramos y organizamos los datos para que puedan ser utilizados eficazmente por los algoritmos de aprendizaje automático.

node	count	pv	pv index
N-000-000000	8	7.12	1.33
N-000-000001	5	6.87	1.04
N-000-000010	8	6.86	1.02
N-000-000011	3	6.49	0.71
N-000-000110	3	7.18	1.42
N-001-000000	313	6.88	1.05
N-001-000001	75	6.71	0.88
N-001-000010	276	7.14	1.35
N-001-000011	44	7.16	1.38
N-001-000110	130	7.68	2.34
N-001-000111	5	8.05	3.37
N-001-001000	5	8.07	3.45
N-001-001010	1	7.58	2.11
N-001-001110	1	7.35	1.67

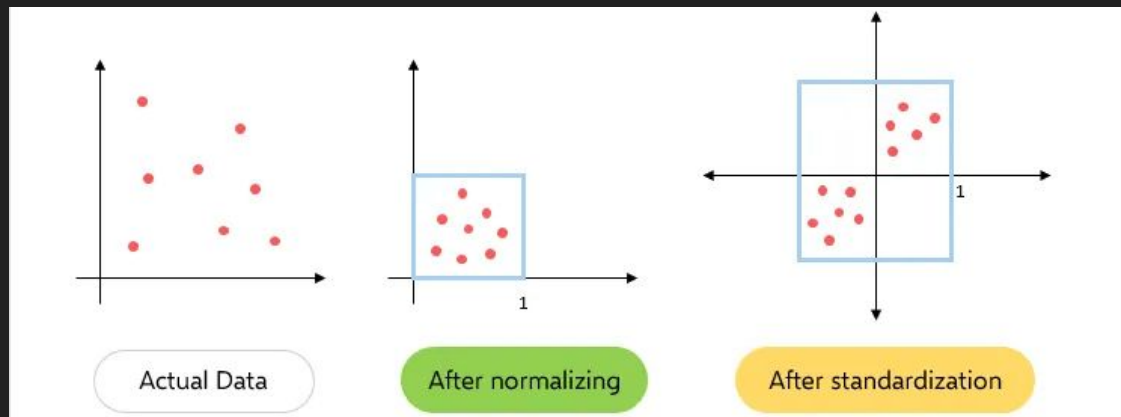


Tipos comunes de datos:

Tipo de dato	Ejemplo	Representación común
Numérico continuo	Altura, temperatura	Float o int
Categorico nominal	Color: rojo, verde, azul	One-hot encoding, label encoding
Categorico ordinal	Nivel: bajo, medio, alto	Etiquetas numéricas ordenadas
Texto	Reseñas de usuarios	Bag of Words, TF-IDF, embeddings
Imágenes	Fotografías, capturas de pantalla	Matriz de píxeles (arrays)
Series de tiempo	Datos de sensores	Secuencias con marcas de tiempo
Datos multimodales	Combinación de texto + imagen	Estructura combinada de vectores

Consideraciones clave:

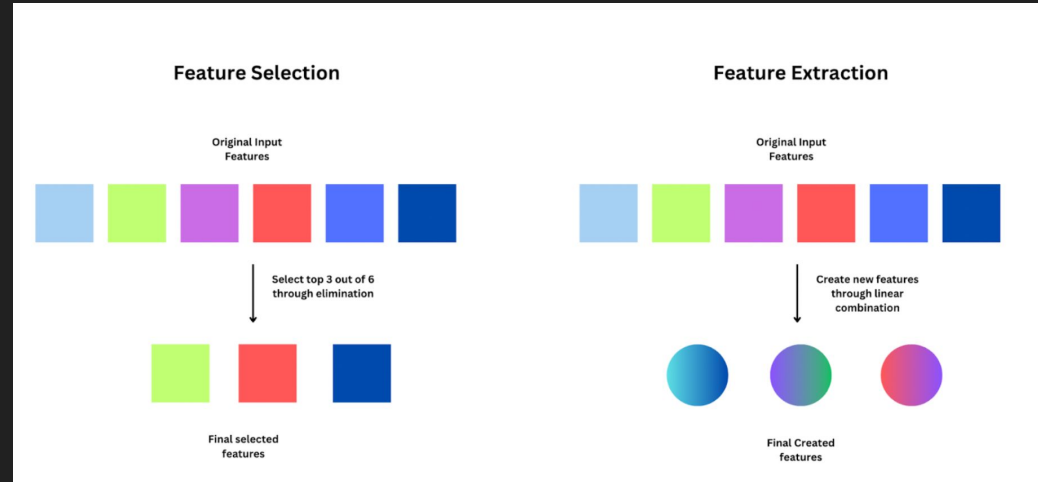
- Escalamiento y normalización: muchos algoritmos requieren que los datos estén en una escala común (e.g., de 0 a 1).
- Codificación de variables categóricas: esencial para convertir texto o categorías en números.
- Manejo de valores faltantes: imputación, eliminación, o modelado explícito.



Extracción de Características (Feature Extraction)

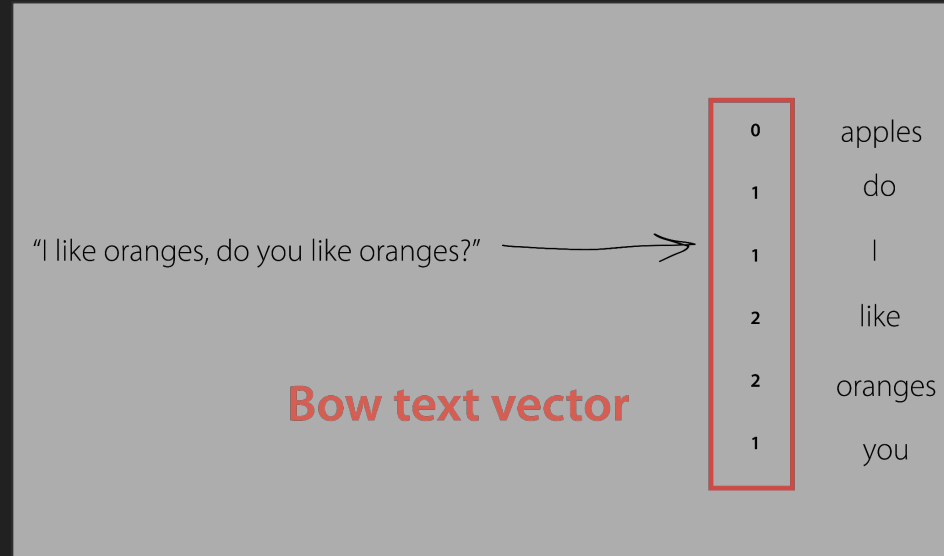
¿Qué es?

Es el proceso de transformar datos en características relevantes que pueden ser utilizadas por modelos de machine learning para mejorar su capacidad predictiva.



Objetivos principales:

- Reducir la dimensionalidad sin perder información importante.
- Destacar patrones o estructuras útiles.
- Preparar los datos para algoritmos que no funcionan bien con entradas "en crudo".





Técnicas por tipo de dato:



Datos tabulares:

- Selección de variables relevantes.
- Transformaciones estadísticas (log, z-score).
- Generación de nuevas variables a partir de otras.



Texto:

- **Bag of Words (BoW)**: frecuencia de palabras.
- **TF-IDF**: frecuencia ponderada por importancia.
- **Word embeddings**: representaciones densas (e.g., word2vec, GloVe).



Imágenes:

- Histogramas de color, bordes, texturas.
- Descriptores como SIFT, HOG.
- Extracción automática con redes convolucionales (CNNs).

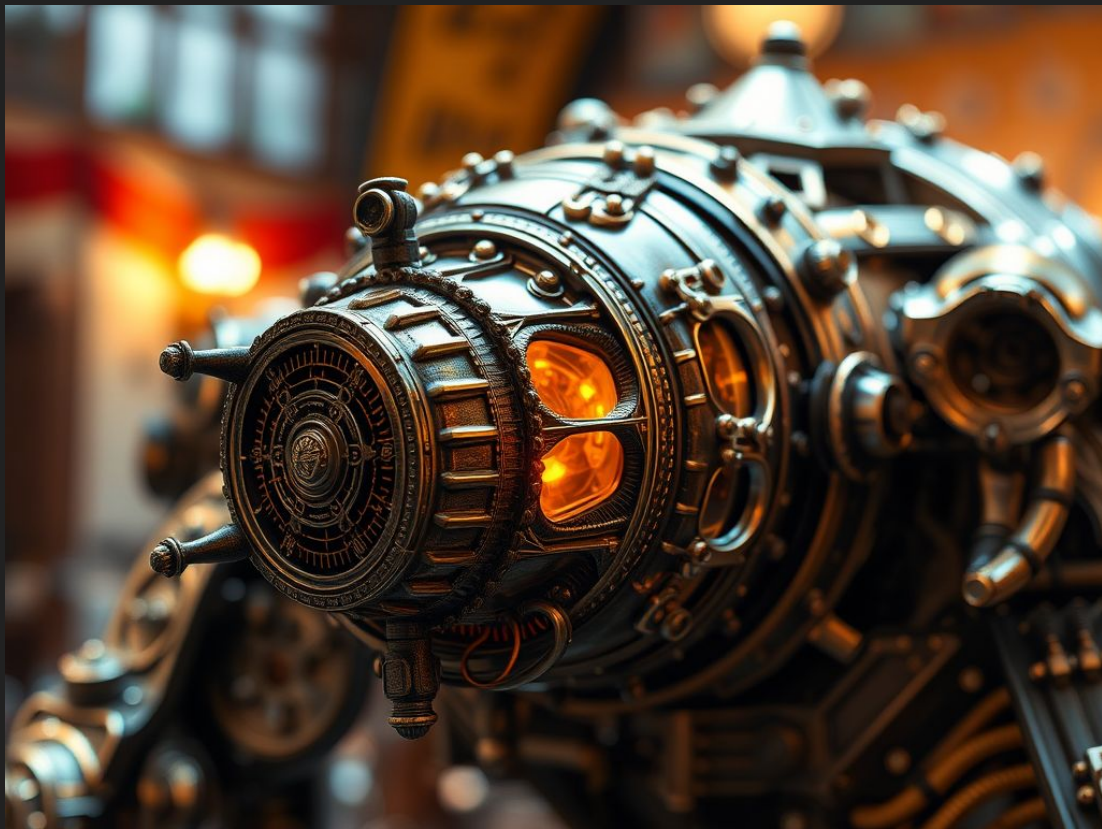


Video / series temporales:

- Promedios móviles, FFT, autocorrelación.
- Segmentación temporal.
- Modelos basados en ventanas o en aprendizaje profundo.

Comparación entre ingeniería y extracción automática de características

Enfoque	Ventajas	Desventajas
Manual (ingeniería)	Interpretabilidad, control	Puede ser limitada o subjetiva
Automática (deep learning)	Escalable, detecta patrones complejos	Requiere más datos y cómputo



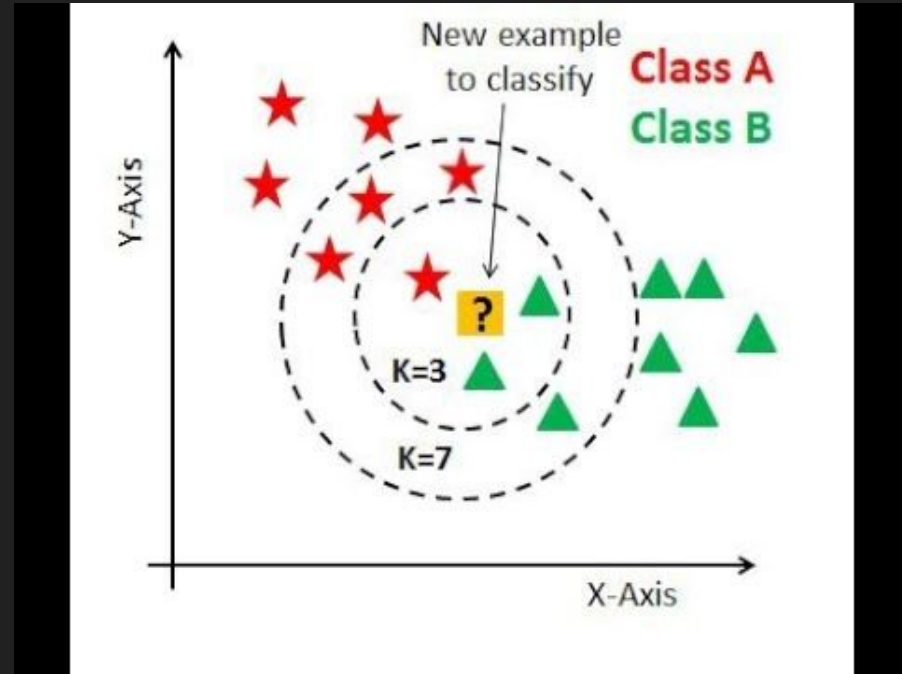
Algoritmos Clave

k-Nearest Neighbors (k-NN)

¿Qué es?

k-NN es un algoritmo de aprendizaje supervisado basado en instancia que clasifica un nuevo ejemplo comparándolo con los k ejemplos más cercanos en el conjunto de entrenamiento.

No tiene una fase de entrenamiento explícita (es un método perezoso): toda la "inteligencia" ocurre en la etapa de predicción.



¿Cómo funciona?

1. Se define un valor k (número de vecinos a considerar).
2. Para un nuevo dato a clasificar:
 - Se calcula la distancia entre este punto y todos los puntos de entrenamiento.
 - Se seleccionan los k vecinos más cercanos.
 - Se realiza la predicción:
 - Clasificación: se toma la clase más frecuente entre los vecinos.
 - Regresión: se promedia el valor numérico de los vecinos.

Naive Bayes

Es una familia de clasificadores probabilísticos basada en el Teorema de Bayes, con la suposición de independencia entre características (de ahí el “naive”).



Teorema de Bayes:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

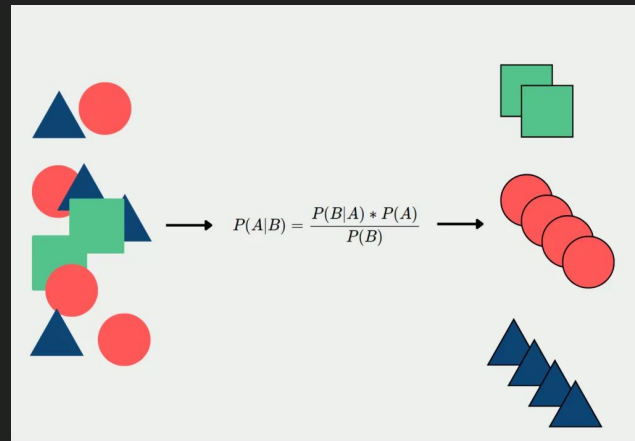
Donde:

- $P(C|X)$: probabilidad de la clase C dado el vector de atributos X.
- $P(X|C)$: verosimilitud.
- $P(C)$: probabilidad previa de la clase.
- $P(X)$: probabilidad de los atributos (se omite en la práctica).

Naive Bayes

Suposición clave:

Asume que las características son condicionalmente independientes dado la clase. Esto rara vez se cumple completamente, pero suele funcionar bien.



✓ Ventajas:

- Muy rápido y simple.
- Funciona bien con texto (clasificación de spam, análisis de sentimientos).
- Requiere poco entrenamiento.

✗ Desventajas:

- La independencia entre atributos rara vez es real.
- No modela bien relaciones complejas entre variables.

SVM (Máquinas de Vectores de Soporte)

¿Qué es?

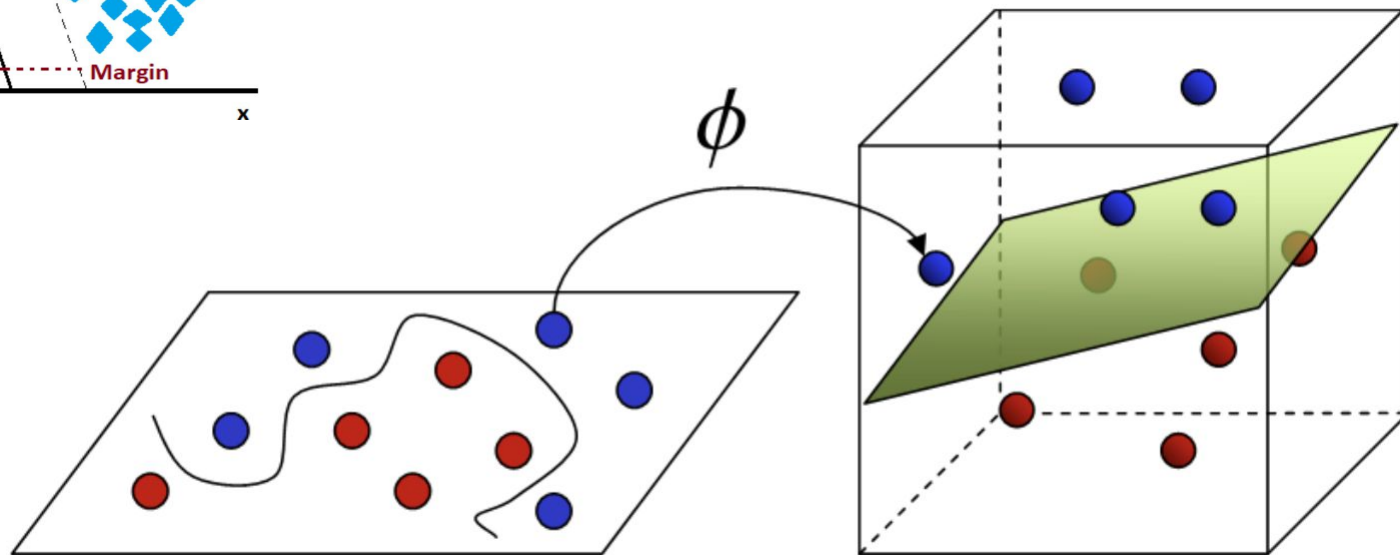
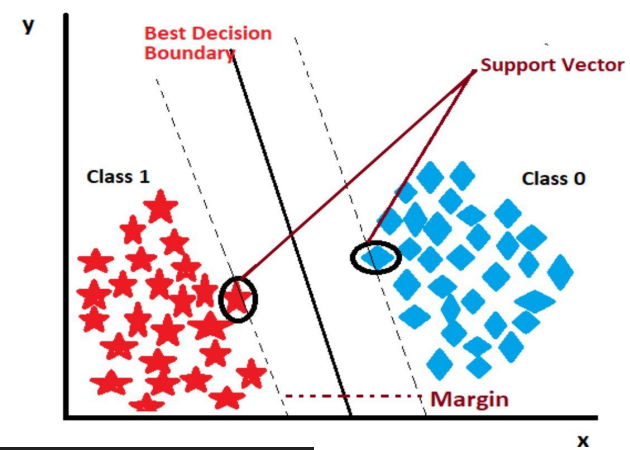
Es un algoritmo de clasificación (y regresión) que busca encontrar el hiperplano óptimo que separa las clases con el mayor margen posible.

Concepto clave:

SVM selecciona los vectores de soporte, es decir, los ejemplos más cercanos al límite de decisión, y maximiza la distancia entre ellos y el hiperplano.

Soporte para casos no lineales:

Mediante el truco del kernel, proyecta los datos en un espacio de mayor dimensión donde se pueden separar linealmente.



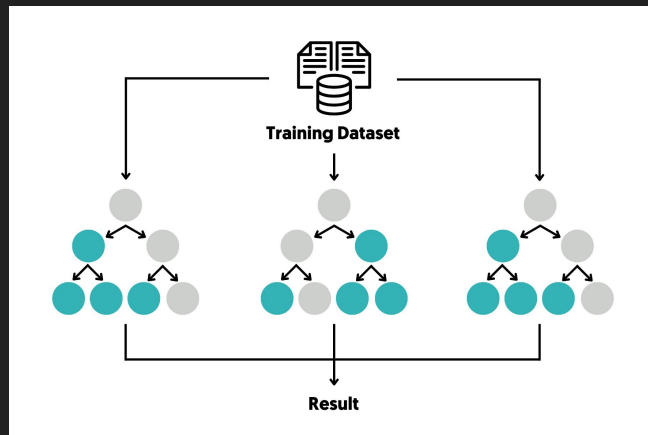
Random Forest

¿Qué es?

Es un ensamble de árboles de decisión. Construye múltiples árboles de decisión y predice por votación (clasificación) o promedio (regresión).

Idea central:

Cada árbol se entrena sobre una muestra aleatoria de los datos (con reemplazo, bagging), y en cada nodo se selecciona un subconjunto aleatorio de características.



Prevención de sobreajuste:

Al combinar múltiples árboles, reduce la varianza y mejora la generalización.

✓ Ventajas:

- Robusto contra sobreajuste.
- Soporta datos mixtos y valores faltantes.
- Requiere poca preparación de los datos.
- Permite calcular la **importancia de las características**.

✗ Desventajas:

- Puede ser lento en predicción con muchos árboles.
- Difícil de interpretar individualmente.

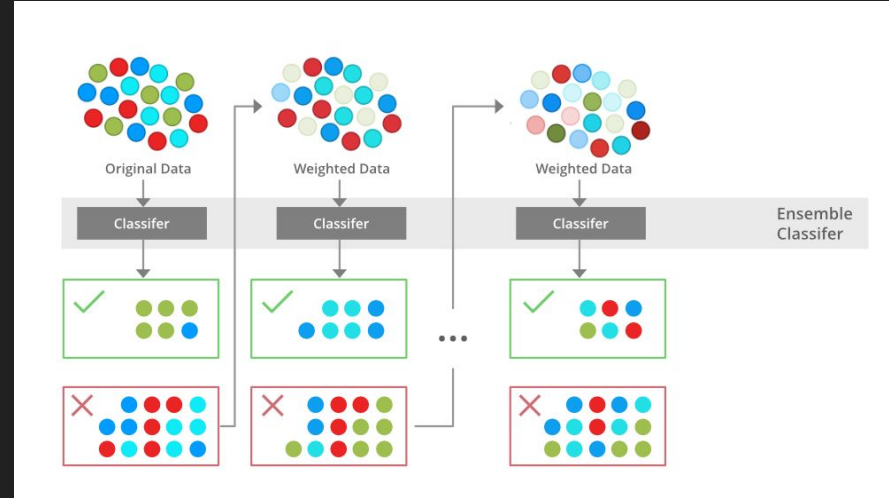
Boosting (e.g., AdaBoost, XGBoost, LightGBM)

¿Qué es?

Es una técnica de ensamble secuencial que combina modelos débiles (usualmente árboles pequeños) para formar un modelo fuerte.

Idea clave:

Cada nuevo modelo se entrena en los errores del anterior, con mayor énfasis en los ejemplos mal clasificados.



Boosting

✓ Ventajas:

- Alta precisión.
- Muy efectivo para conjuntos de datos tabulares.
- Flexibilidad para clasificar y predecir valores.

✗ Desventajas:

- Más sensible al sobreajuste si no se regula.
- Requiere más cómputo y tiempo de entrenamiento.
- Parámetros más complejos de ajustar.



GRACIAS!