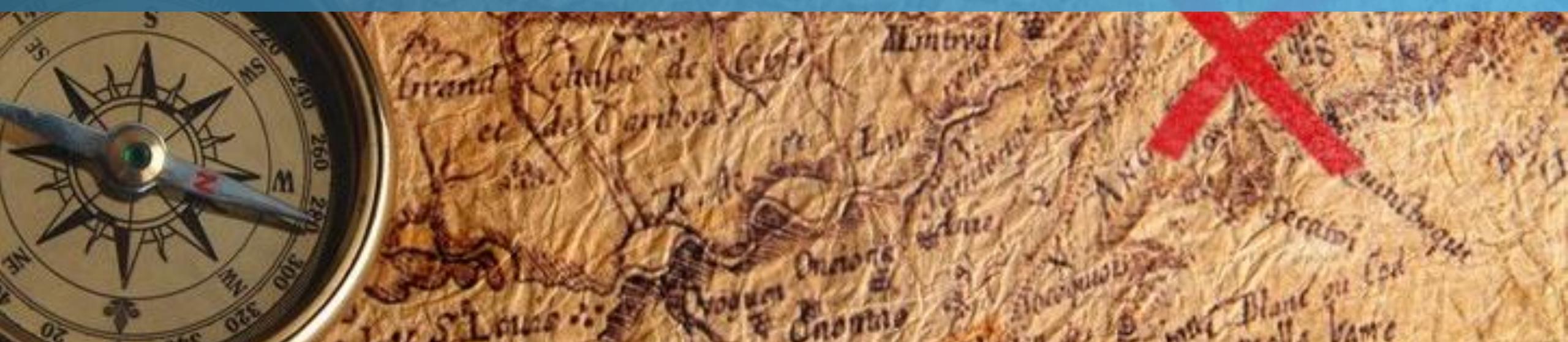


# 4. Algoritmos de aprendizaje automático en SBC



**MD005 – Sistemas Basados en el Conocimiento**

Manel Cerezo García- [manel.cerezo@salle.url.edu](mailto:manel.cerezo@salle.url.edu)

David Larrosa- [david.larrosa@salle.url.edu](mailto:david.larrosa@salle.url.edu)

# Organización

2025							
OCTUBRE							
L	M	X	J	V	S	D	
		1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
21	22	23	24	25	26	27	
28	29						

NOVIEMBRE							
L	M	X	J	V	S	D	
				1	2		
4	5	6	7	8	9		
11	12	13	14	15	16		
18	19	20	21	22	23		
25	26	27	28	29	30		

DICIEMBRE							
L	M	X	J	V	S	D	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31					

2026							
ENERO							
L	M	X	J	V	S	D	
				1	2	3	4
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	31		

FEBRERO							
L	M	X	J	V	S	D	
						1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28		

# Organización

- **Aprendizaje no supervisado (Agrupamiento)**
  - Qué es y en qué consiste.
  - Algoritmos a tener en cuenta: K-means, jerárquico, densidades.
  - Teoría de los algoritmos.
  - Ejemplos prácticos: ejercicio de ejemplo y ejercicio guiado.
- **Aprendizaje supervisado (clasificación)**
  - Qué es y en qué consiste.
  - Algoritmos a tener en cuenta: Nearest neighbours (KNN), árboles de decisión.
  - Teoría de funcionamiento.
  - Ejemplos prácticos: ejercicio de ejemplo y ejercicio guiado.

# Organización

- **Aprendizaje supervisado (regresión)**
  - Qué es y en qué consiste.
  - Algoritmos a tener en cuenta: regresiones polinomiales, árboles de regresión.
  - Teoría de funcionamiento.
  - Ejemplos prácticos: ejercicio de ejemplo y ejercicio guiado.
- **Parametrización de los algoritmos y su evaluación en profundidad**
  - Qué nos aporta parametrizar.
  - Métodos de parametrización: gridsearch, randomsearch, parametrización métodos estadísticos condicionados (Bayes).
  - Teoría de funcionamiento.
  - Ejemplos prácticos: ejercicios de ejemplo y ejercicios guiados para los diferentes algoritmos vistos.

# Organización

- **Redes Neuronales**
  - Qué son y porqué funcionan. Conceptos básicos.
  - Arquitectura simple: perceptrón.
  - Ejercicios de ejemplo y guiado para 3 casos de uso:
    - Clasificación entre 2 clases (binaria)
    - Clasificación entre más clases (multiclas)
    - Regresiones
- **Redes Neuronales**
  - Evolución del perceptrón: perceptrón multicapa.
  - Ejercicios de ejemplo y guiado para 3 casos de uso:
    - Clasificación entre 2 clases (binaria)
    - Clasificación entre más clases (multiclas)
    - Regresiones
  - Comparativa entre perceptrón y perceptrón multicapa.

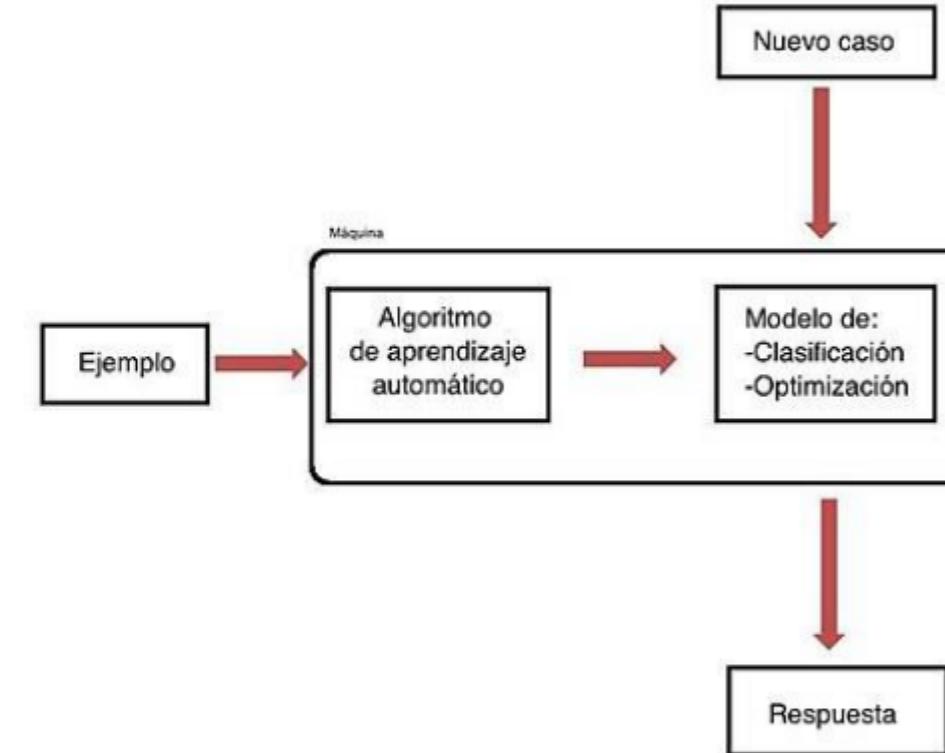
## Hoy veremos...

- Objetivo del aprendizaje automático
- El ciclo de los datos
- Caso real de aplicación del ML
- Paradigmas (clasificaciones)
- Introducción a los algoritmos que veremos a lo largo de la asignatura
- Características deseables
- Posibles problemas en el entrenamiento
- ¿Como evaluamos un método que aprende?

# Objetivo

# ¿Qué es el aprendizaje automático (ML)?

- **Objetivo:** Desarrollar sistemas que se adapten a nuevas situaciones y resolver problemas en ellas sin un entrenamiento previo. El entrenamiento se basará en situaciones ya conocidas, con las que se generará un conocimiento que se extrapolará a las nuevas situaciones.





Cómo planteamos  
un problema

# Ciclo de desarrollo



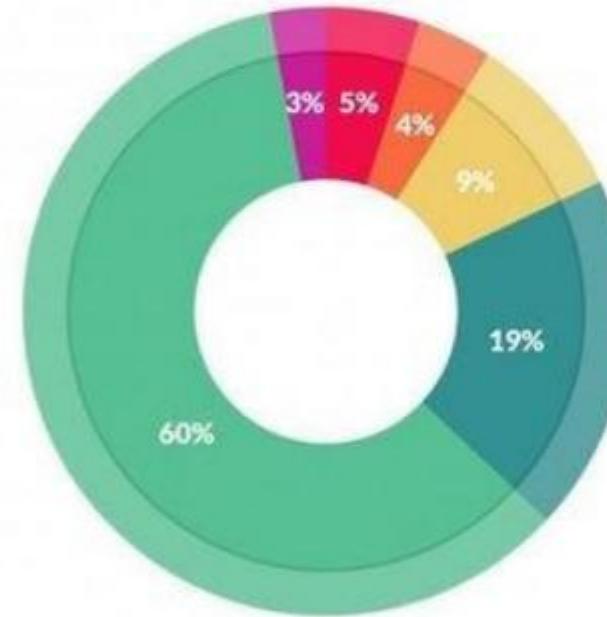
## Ciclo de desarrollo



## Ciclo de los datos

- Los datos se generan en los dispositivos (Cultivo).
- Recolección y almacenaje de éstos (Cosecha).
- Fases de filtrado (Post-Cosecha).
- Dataset 'limpio' (Post-Cosecha).
- Fase exploratoria (Tostar).
- Preparación para el uso de la información relevante (Preparación).
- Entreno del modelo (Preparación).
- Realización de predicciones (Consumo)

# Ciclo de los datos: tiempo invertido



What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

# "Pasos" para construir un modelo de ML

## 1. Entendimiento del dominio y problema

- ¿Supervisado o no supervisado?

## 2. Análisis exploratorio de Datos (EDA)

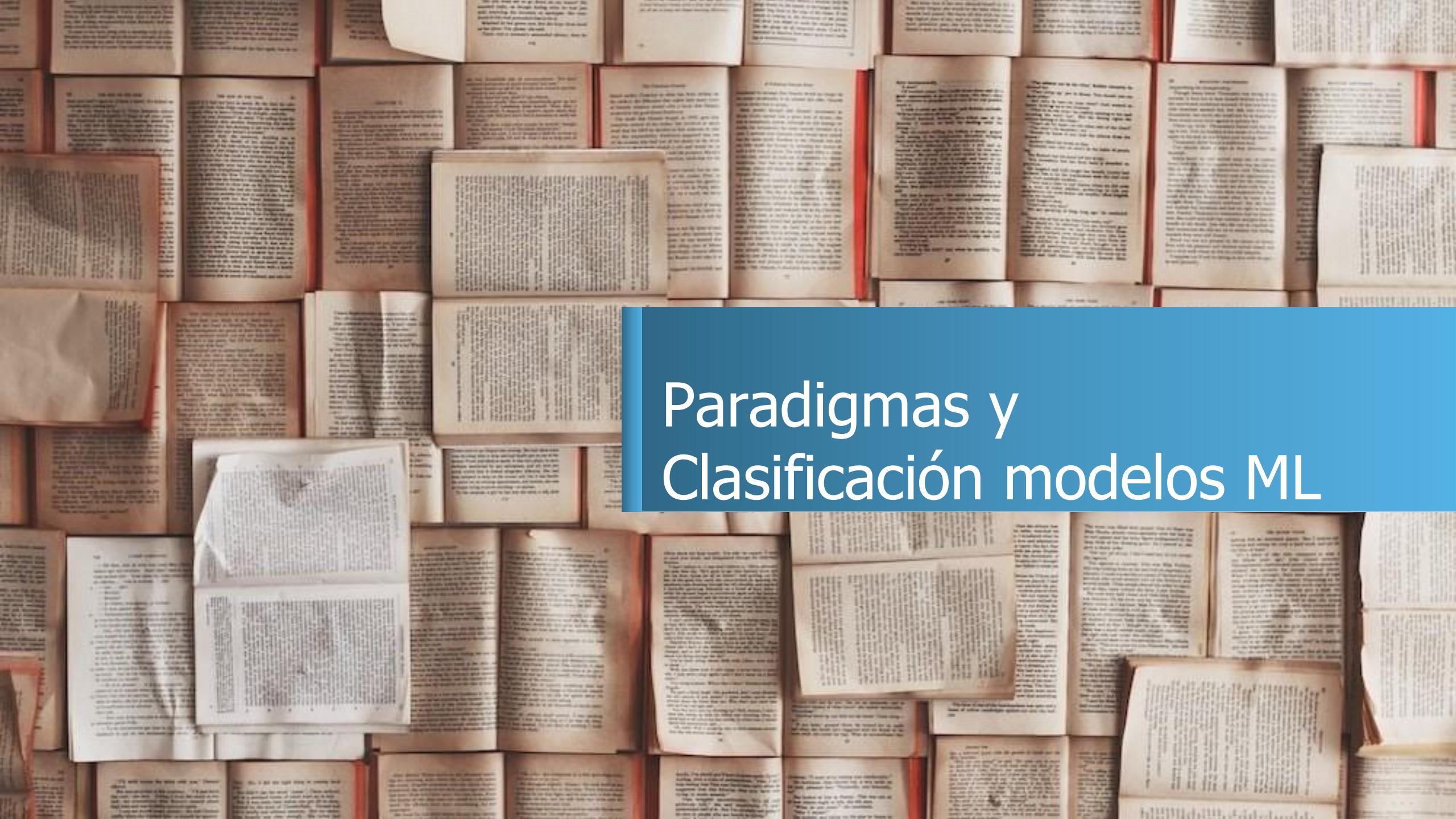
- **Calidad** general de los datos y **estadística descriptiva**.
- Visualizaciones que ayuden a entender la **distribución** de las variables continuas y categóricas.
- Visualizaciones que ayuden a entender la **relación** entre los atributos y la variable objetivo (si la hay).
- **Estadísticos** que nos ayuden a entender mejor la **relación** (Correlación, PCA, ANOVA, ...).

## 3. Construcción del modelo

- **Elección** de los **modelos** que creemos que mejor funcionarán.
- **Elección** del método de **validación** (split train-test, k-fold cross validation, ...).
- **Preprocesado de datos** para que el modelo pueda ser entrenado (one-hot encoding, ...).
- **Hiperoptimización de parámetros** (grid search, random grid search, ...).
- **Entrenamiento** de los modelos.

## 4. Análisis e interpretación de los resultados

- **Análisis de rendimiento** de los modelos (accuracy,  $R^2$ , matriz de confusión, ...).
- **Comparación** de resultados obtenidos con distintos modelos.
- **Interpretación** de los resultados del modelo candidato para poner en producción.

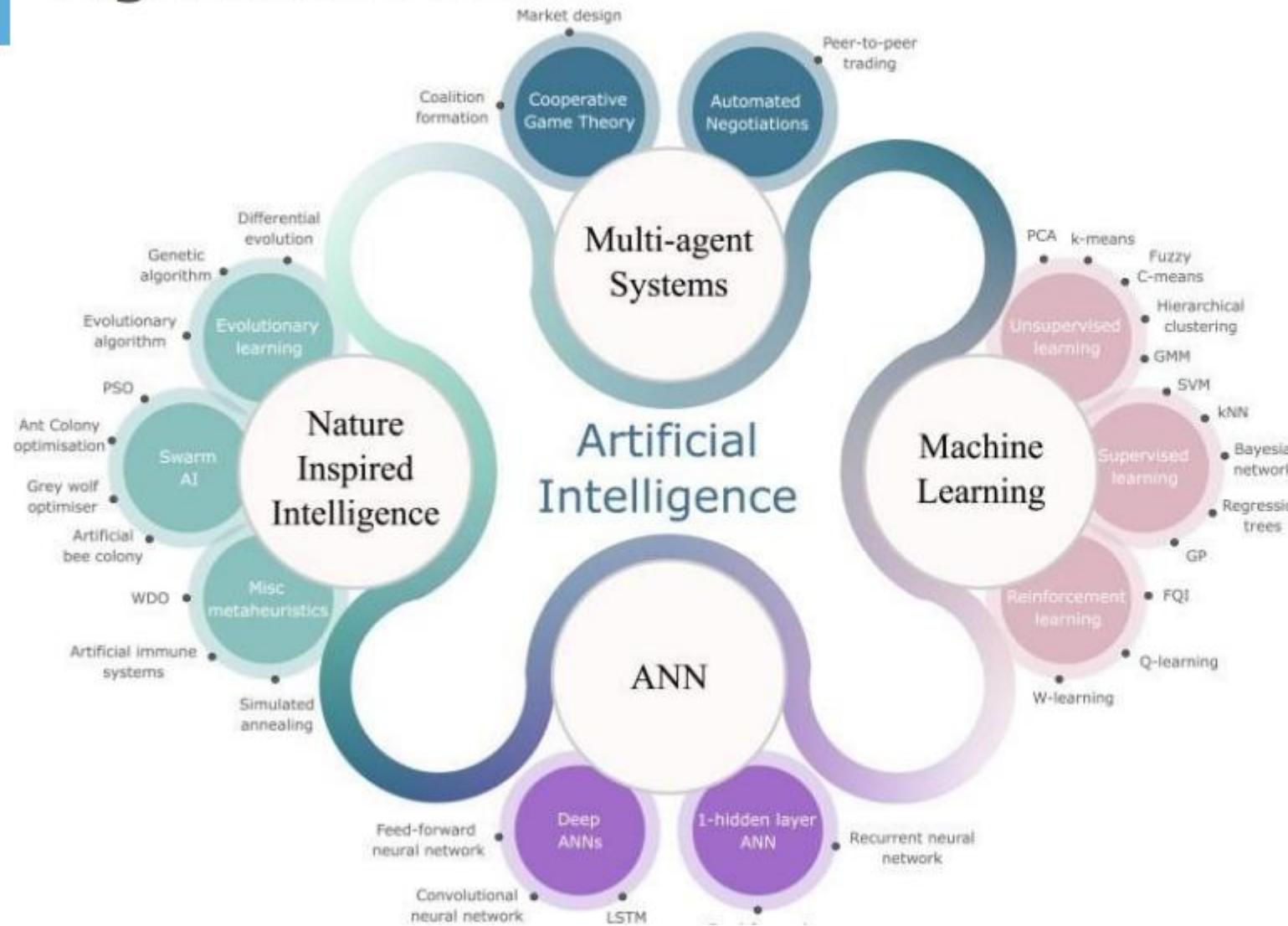


# Paradigmas y Clasificación modelos ML

# Paradigmas (clasificaciones) del ML

- **Tipos y clasificaciones:** En función del criterio a escoger, se pueden clasificar de 3 formas distintas:
  - **Según su selección y adaptación:** deductivo, inductivo, analogico, evolutivo, conexionista...
    - **Análogico:** inspirado en la memoria humana, resolvemos problemas a partir de la similitud con problemas ya vistos, adaptamos soluciones que funcionan para los similares al nuevo caso concreto.
    - **Conexionista:** Se inspira en la neurobiología, las neuronas se conectan entre ellas propagando valores que aprenden a partir de conocer ejemplos y bien hecho, generaliza el conocimiento.
  - **Según la estrategia:**
    - **Supervisado:** Los ejemplos se utilizan para extraer conocimiento, necesitamos todos los datos posibles para aplicarlos.
    - **No supervisado:** Extrae conocimiento a partir de descubrir patrones en los datos disponibles.
    - **Por refuerzo:** Entre las dos estrategias anteriores, se utiliza una función que evalúa qué tan bien se ajustas a modo de refuerzo para dirigir el aprendizaje del algoritmo.
  - **Según el problema que resuelve:** Asociación, clasificación, agrupación, regresión...

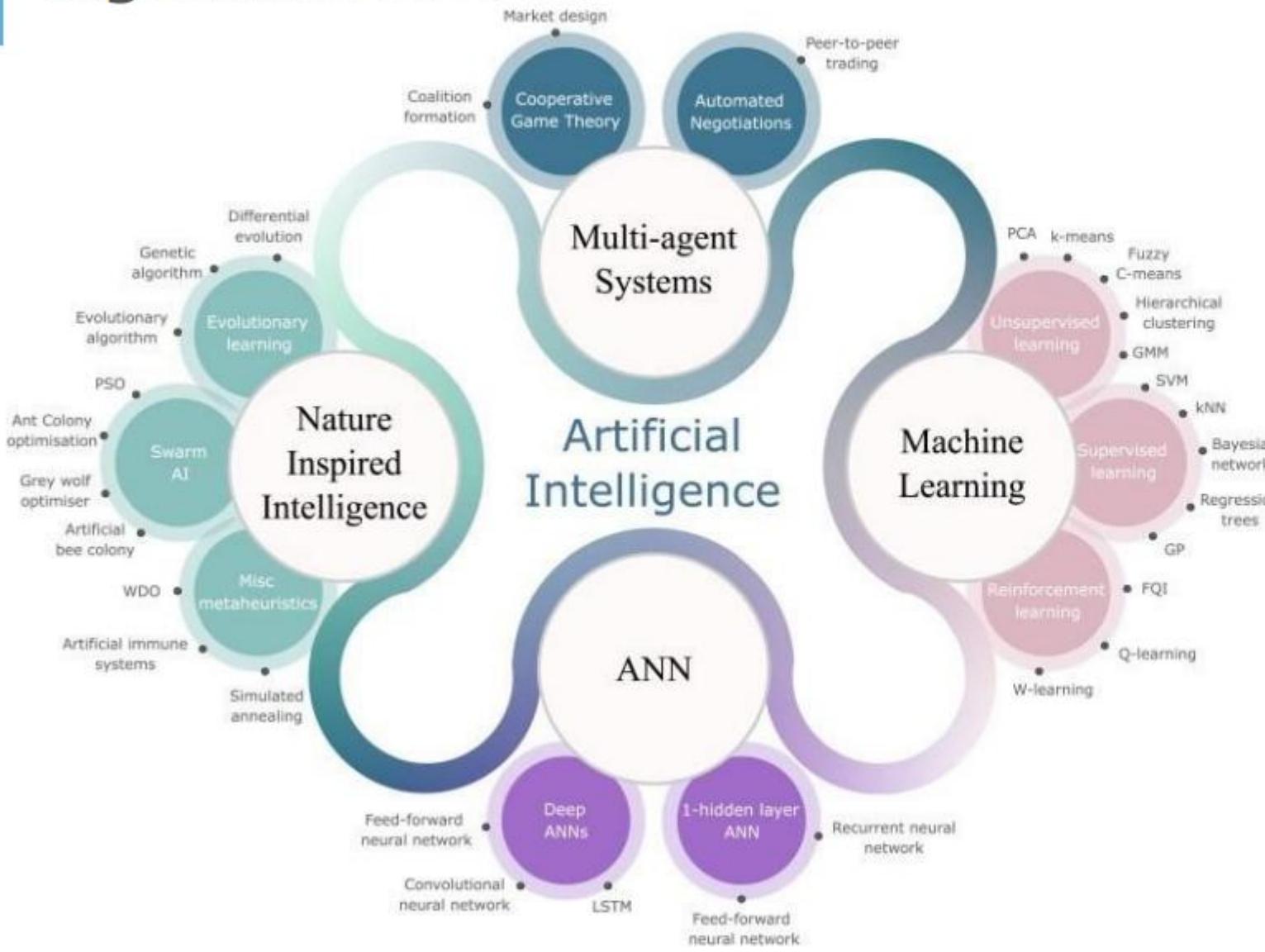
# Algoritmos MD005



**¿Qué podemos utilizar para cada problema?**

- **¿Identificador Melanoma?**
- **¿Historial Crediticio?**
- **¿Ruta del comerciante?**

# Algoritmos MD005



**¿Qué podemos utilizar para cada problema?**

- ¿Identificador Melanoma?**

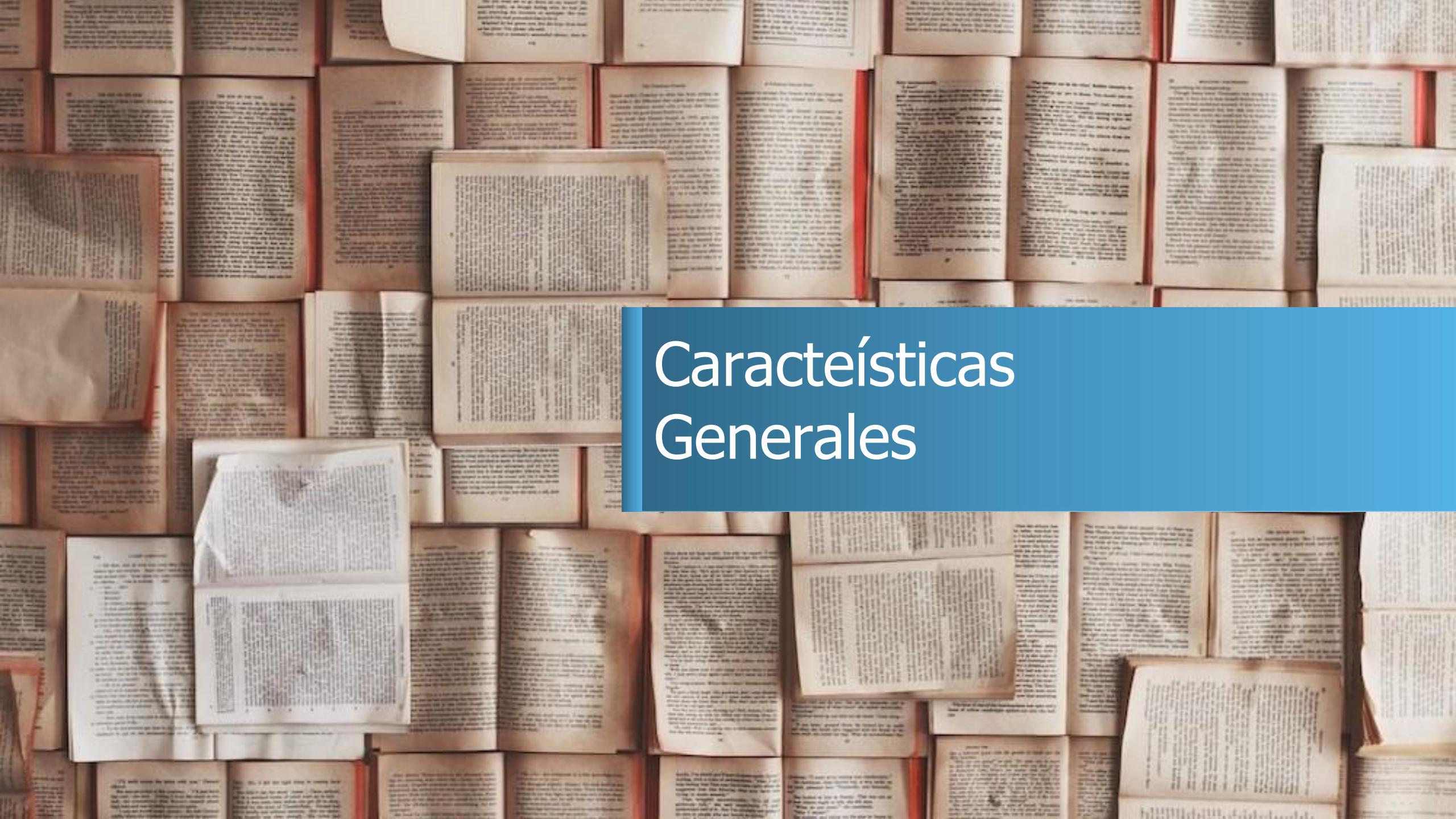
Árboles de decisión, Random Forest, XGBoost, CatBoost ... incluso agrupación Redes Neuronales

- ¿Historial Crediticio?**

Clasificadores / Regresiones  
Redes Neuronales

- Ruta del comerciante?**

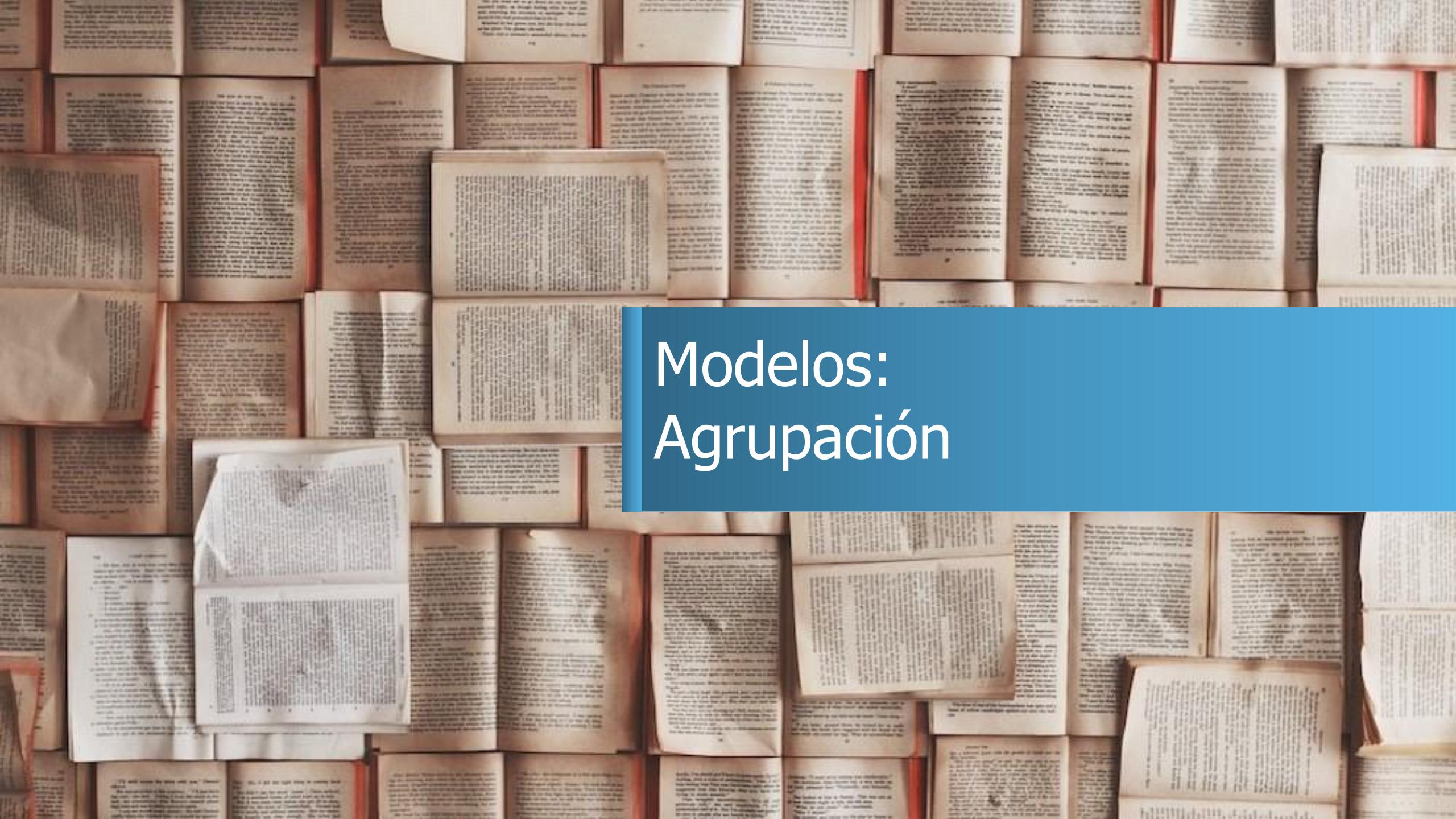
Ant colony optimization  
Redes Neuronales



# Características Generales

## Características deseables

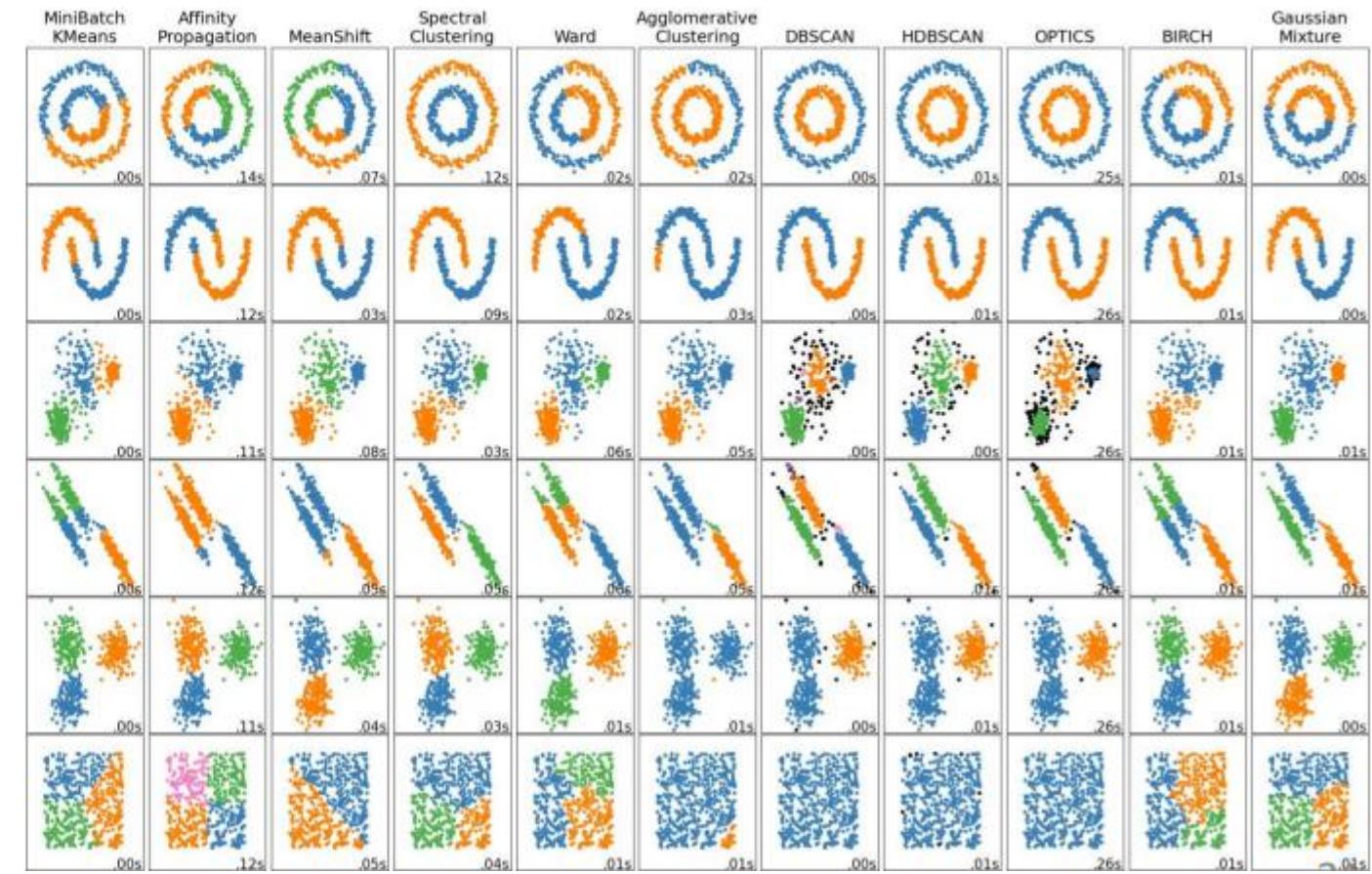
- **Generalidad/capacidad de abstracción:** Que sean capaces de extraer un conocimiento general y no solamente aprendan los ejemplos del entrenamiento.
- **Robustez:** Tanto el funcionamiento como el resultado ha de ser robusto y proporcionar resultados consistentes incluso si hay posibles ruidos.
- **Aprendizaje:** Ha de ser razonable a nivel de recursos (costes computacionales y de memoria) tanto a nivel de resultados. Utilizar algoritmos y métodos a fuerza bruta no es una solución viable si el coste es excesivamente elevado.
- **Explicabilidad:** Los modelos han de ser explícitos o por lo menos, maximizar el entendimiento de éste. Utilizar un algoritmo sin saber cómo funciona ni por qué nos hará llegar a un callejón sin salida.
- **Determinista:** Para cualquier conjunto de datos, el resultado ha de ser siempre el mismo. En caso de no ser así, el algoritmo no es nada fiable.



# Modelos: Agrupación

Partiendo del conjunto de datos, contiene los que son similares dentro de un mismo grupo autoidentificado maximizando la compacidad del grupo y la diferenciación respecto a otros grupos.

Técnica/Algoritmo	Tipo
K-Means	Distancias
DBSCAN	Densidad
OPTICS	Densidad
Jerárquico	Distancias



# Algoritmos de agrupación

- **Segmentación de clientes**

- Input: Datos de comportamiento de compra, demografía, interacciones
- Output: Grupos de clientes con características similares

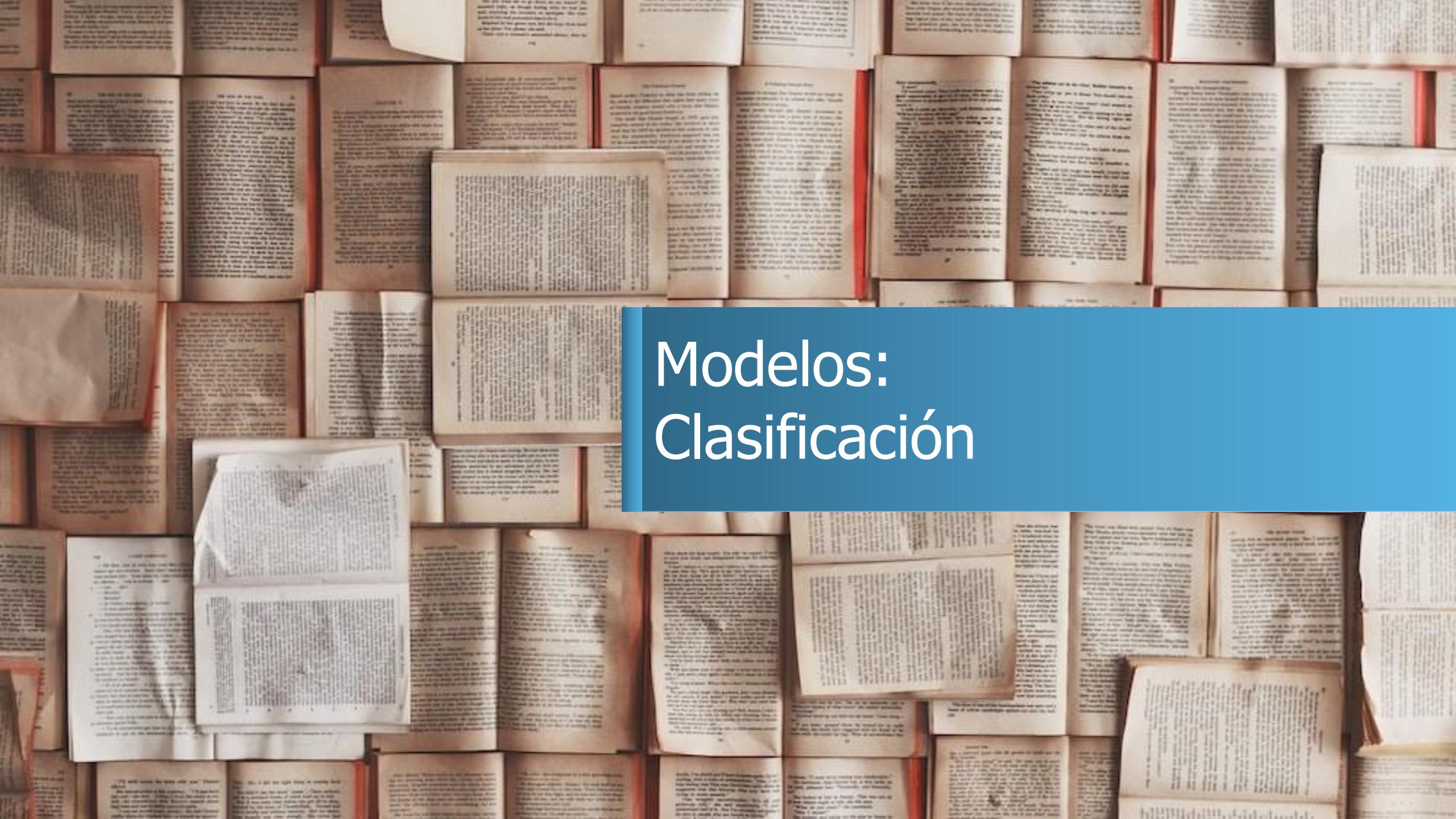
- **Agrupación de artículos**

- Input: Textos de artículos
- Output: Clusters temáticos automáticos

- **Identificación de patrones**

- Input: Datos de consumo horario de diferentes hogares
- Output: Perfiles típicos de consumo

Imágenes de Wikipedia



# Modelos: Clasificación

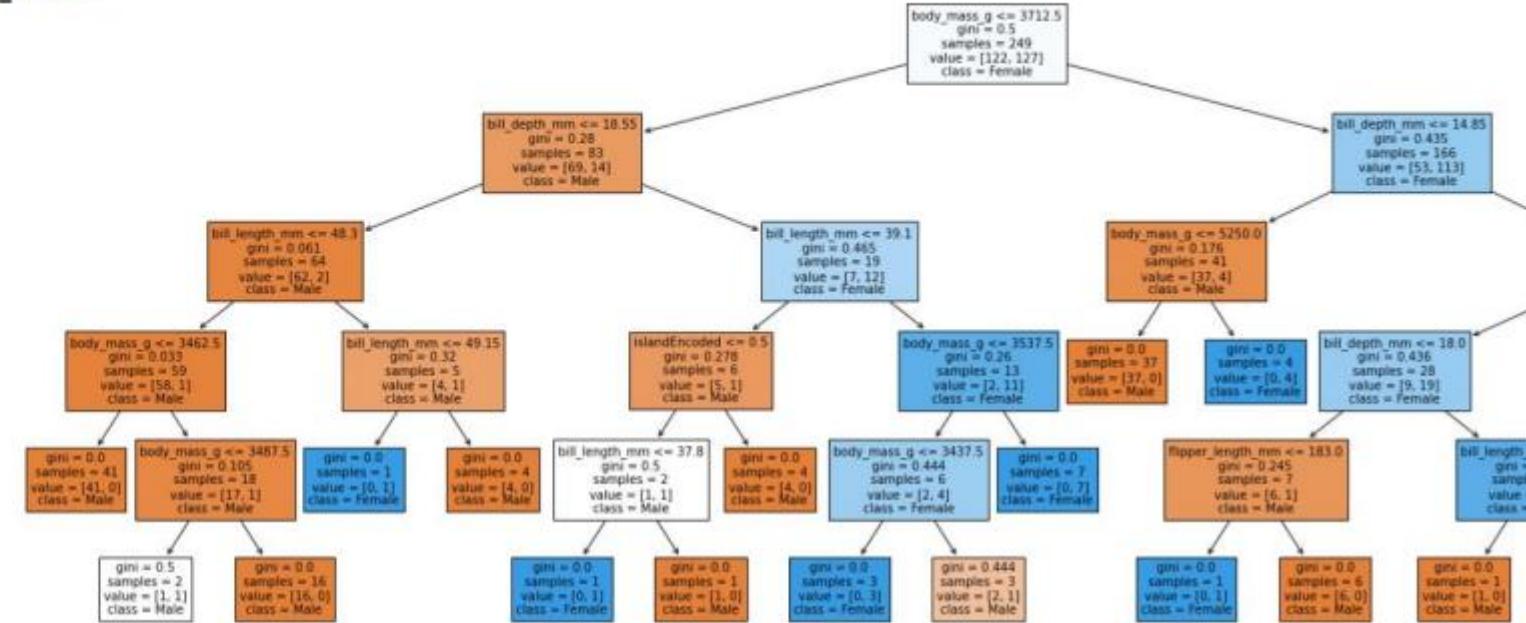
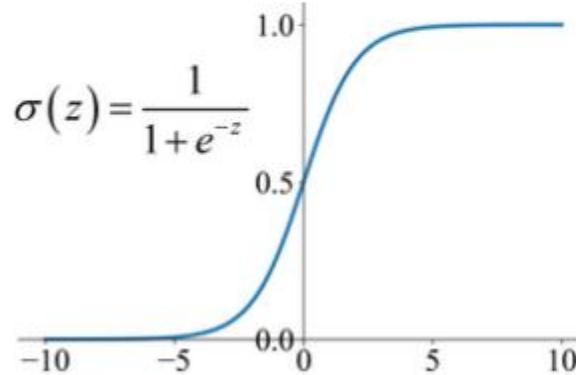
# Algoritmos de clasificación

Problemas típicamente de identificación. A partir de un conjunto de inputs (entradas de datos) asignar una clase u otra al elemento que estemos evaluando, siendo:

- **Binaria:** Casos en los que sólamente hay 2 opciones posibles.
- **Multiclasificación:** Más de 2 opciones disponibles.

Técnica/Algoritmo	Problema
Regresión Logística	Binario
KNN (k-nearest neighbors)	Binario/Multiclasificación
Árbol de decisión	Binario/Multiclasificación
Random Forest	Binario/Multiclasificación
SVC (Support Vector Classifier)	Binario/Multiclasificación

# Algoritmos de clasificación



Múltiples árboles de decisión y la salida es la combinada entre ellos por lo que cada árbol puede ser especialista en una sección concreta de los datos y por lo tanto, dar mejor resultado que uno solo.

# Algoritmos de clasificación

- **Detección de correo**
  - Input: Características del email (palabras clave, remitente, enlaces)
  - Output: Clasificación binaria (spam/no spam)
- **Diagnóstico médico**
  - Input: Síntomas, resultados de análisis, historial médico
  - Output: Probabilidad de diferentes enfermedades
- **Predicción de abandono de clientes (Customer Churn)**
  - Input: Historial de uso del servicio, quejas, pagos, tiempo como cliente
  - Output: Probabilidad de que el cliente abandone el servicio

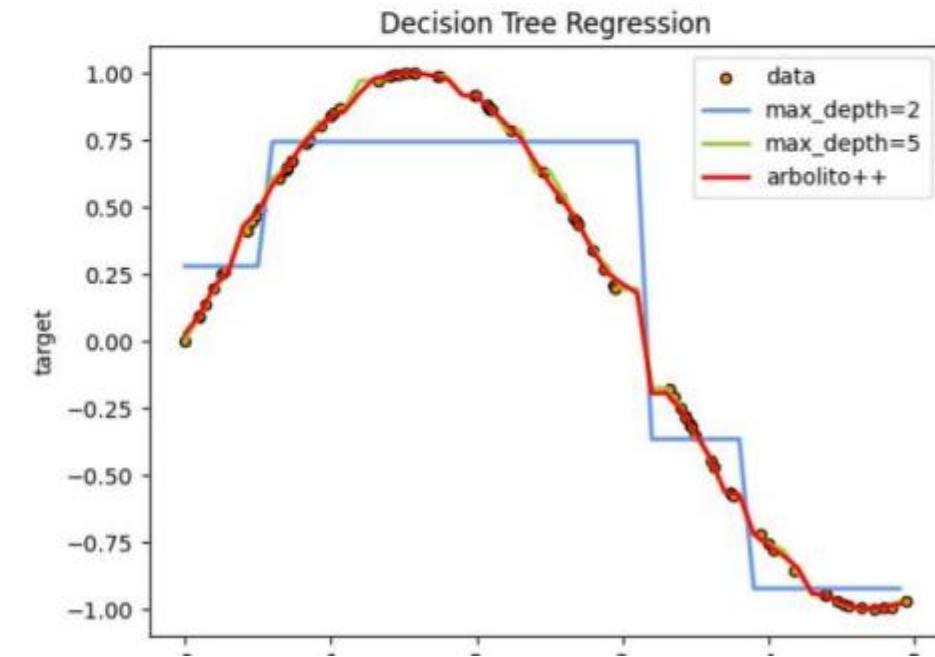
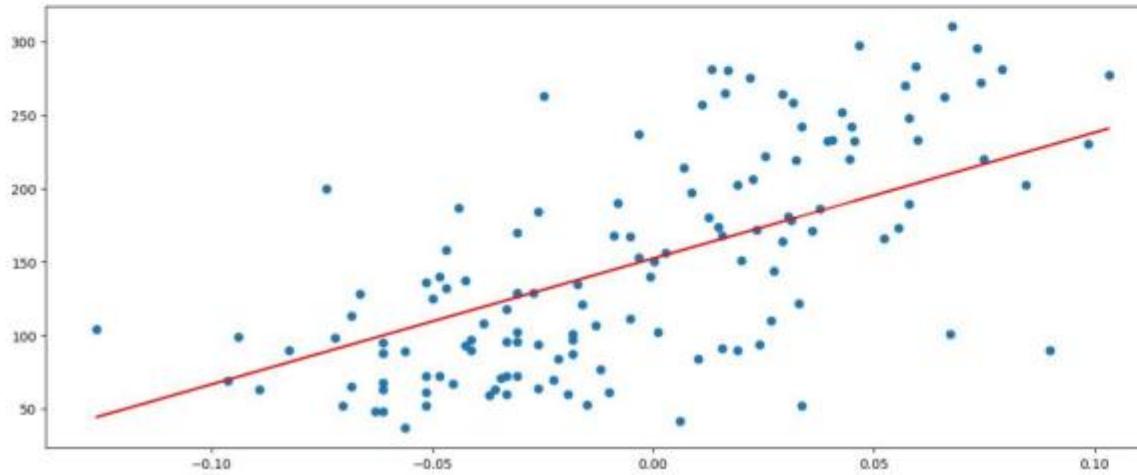


# Modelos: Regresión

## Algoritmos de regresión

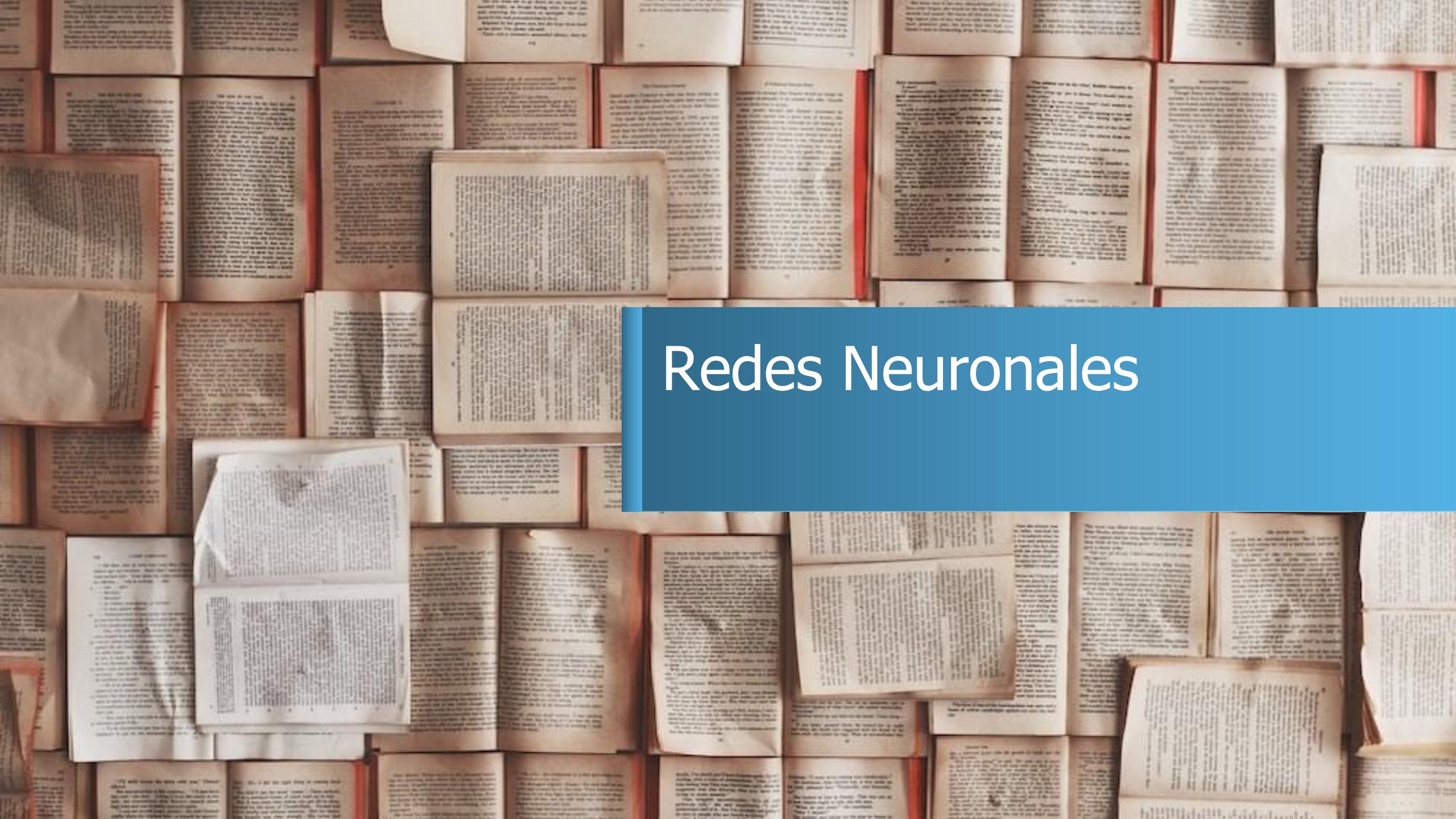
Partiendo del conjunto de datos, encontrar y recrear la función que modela a la variable dependiente en función del resto en todo el espacio latente.

- Técnica/Algoritmo
- Regresión lineal
- Regresión no lineal
- Árboles de regresión



# Algoritmos de regresión

- **Predicción de precios de viviendas**
  - Input: Características ( $m^2$ , ubicación, antigüedad, etc.)
  - Output: Precio estimado
- **Pronóstico de ventas mensuales**
  - Input: Histórico de ventas, temporada, eventos especiales
  - Output: Predicción de ventas futuras
- **Estimación de fallo del motor**
  - Input: Características del vehículo (peso, potencia, aerodinámica)
  - Output: Tiempo hasta necesitar nuevo motor

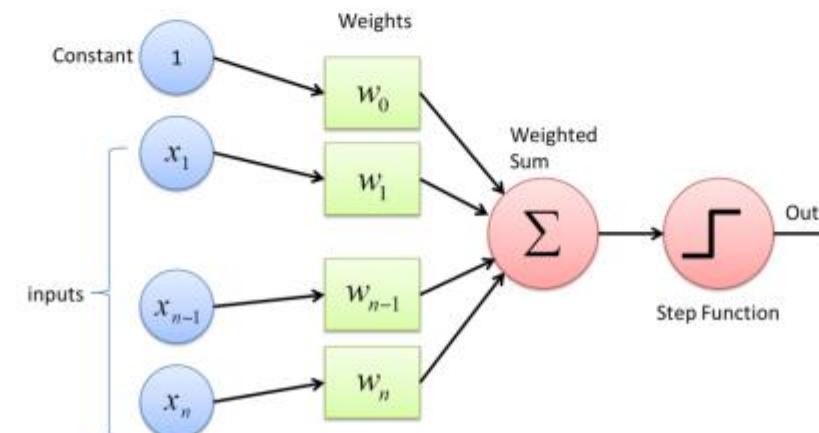


# Redes Neuronales

# Redes neuronales

En términos generales sirven para 'todo', solo necesitamos tener claro cómo adaptamos el problema y la arquitectura.

Técnica/Algoritmo	Uso
Perceptrón	Clasificación/regresión
Perceptrón multicapa	Clasificación/regresión

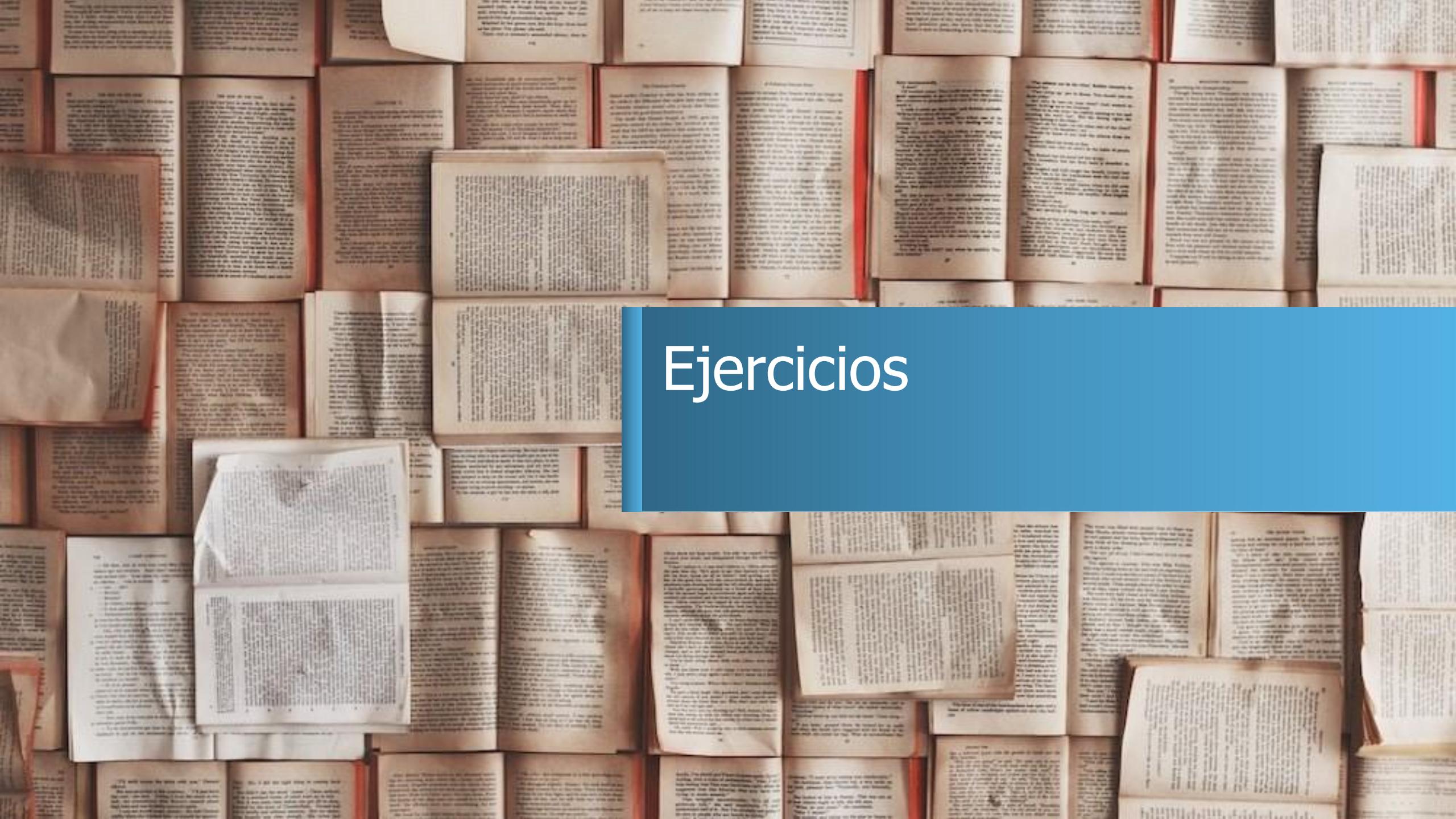


# Redes neuronales

Todo lo anterior!



- Imagenes:
  - Clasificacion
  - Generacion
  - Deteccion
- Sonido
- Series temporales
- ...



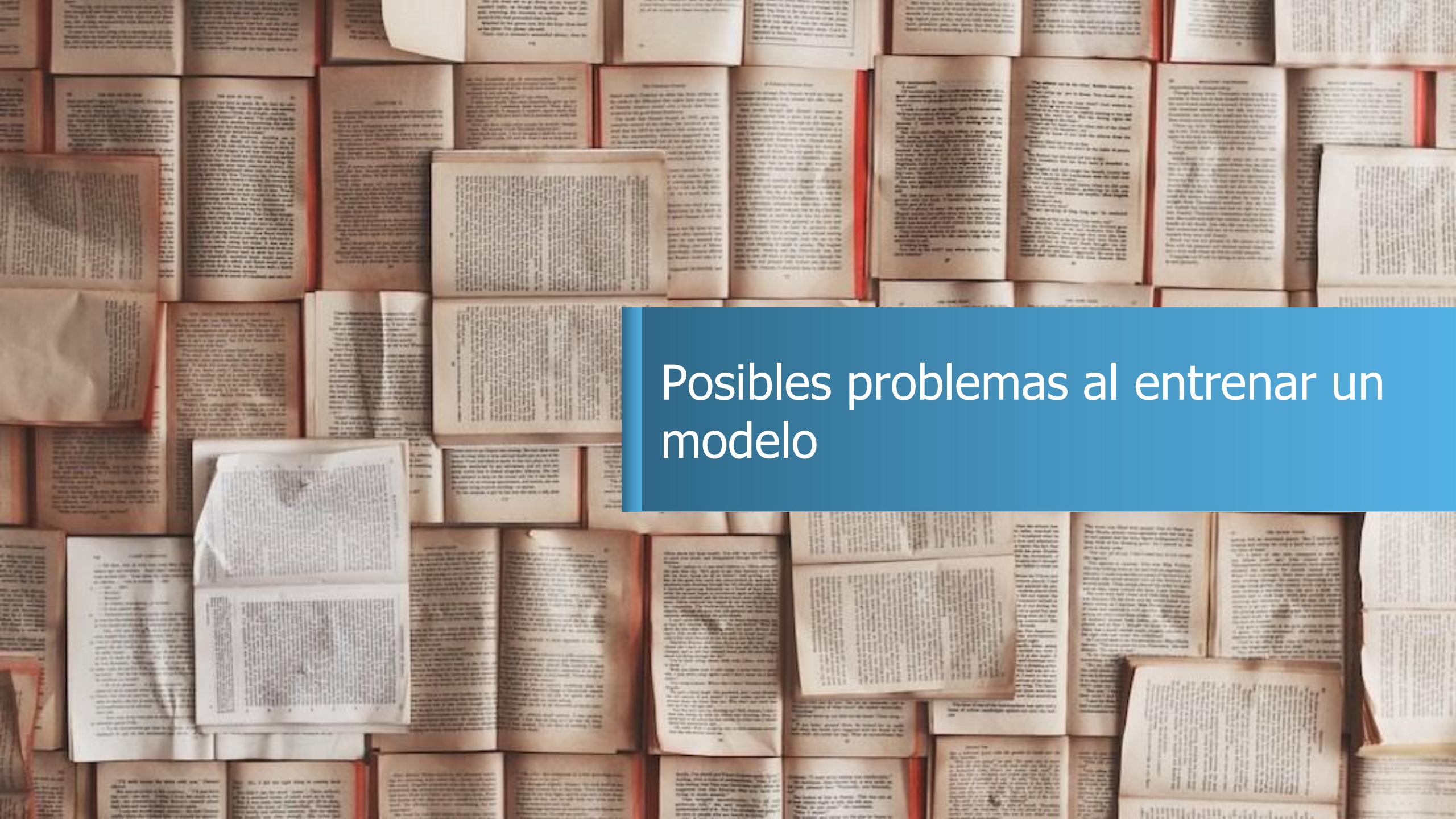
# Ejercicios

## Ejercicio 1

1. Vamos a elegir un modelo al azar para cada alumno.
2. Con el modelo que os haya tocado, tenéis que deciros :
  - Un problema a resolver en vuestra Empresa con dicho modelo
  - Un problema real que se haya resuelto con ese modelo
3. Defender porqué, para ese problema, dicho modelo es mejor opción que el resto:
  - Características de ese modelo diferente a los de su misma especie que lo hace buena elección
  - Porque la elección de esta tipología de modelo (clasificación, regresión,...)

## Ejercicio 2

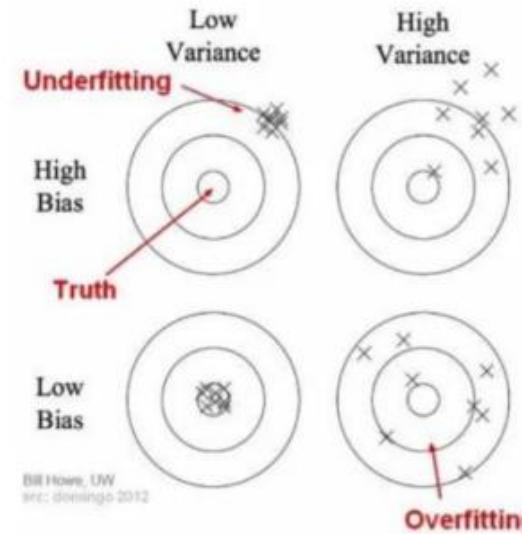
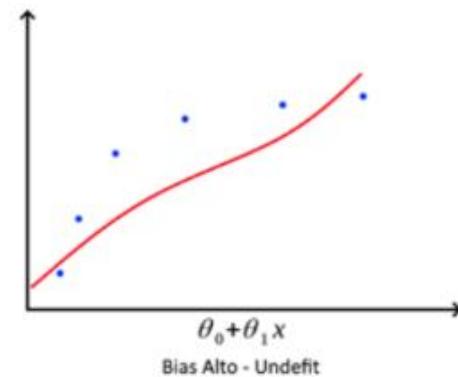
1. Os vamos a dar unos casos a resolver
2. Con el problema que os haya tocado, tenéis que decirnos qué modelo es la mejor opción y porqué



# Posibles problemas al entrenar un modelo

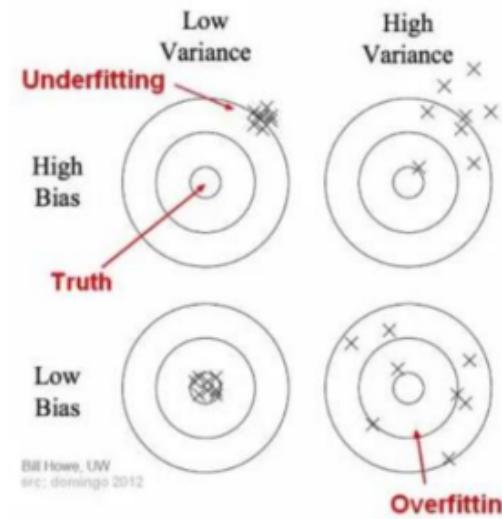
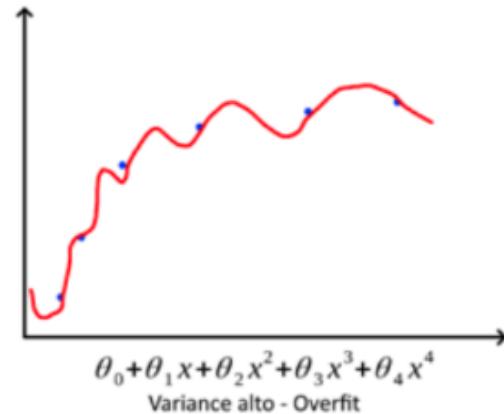
# Underfitting

- ¿De donde surge el **error** de nuestros modelos de ML?
- Principalmente: error = sesgo<sup>2</sup> + varianza +  $\epsilon$
- **Sesgo o bias (underfitting):**
  - Es la diferencia entre la predicción promedio de nuestro modelo y el valor correcto que estamos tratando de predecir.
  - Proviene de la rigidez del propio modelo para aprender las señales complejas del dataset (el modelo es demasiado simplista).
  - Los modelos con pocos grados de libertad son proclives a un alto bias (ej.: regresión lineal).
  - Si alto sesgo → error alto en los datasets de training y de test.



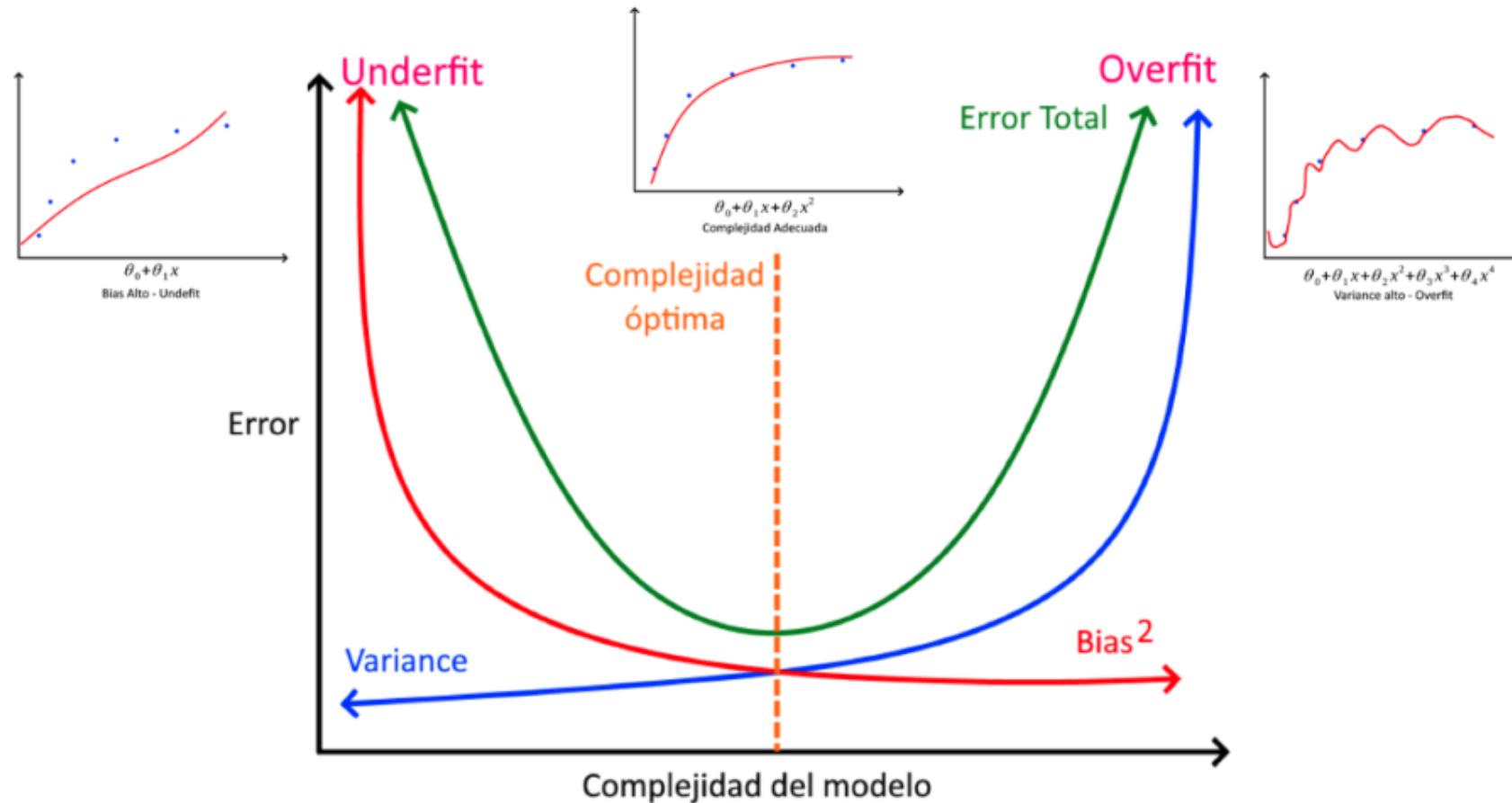
# Overfitting

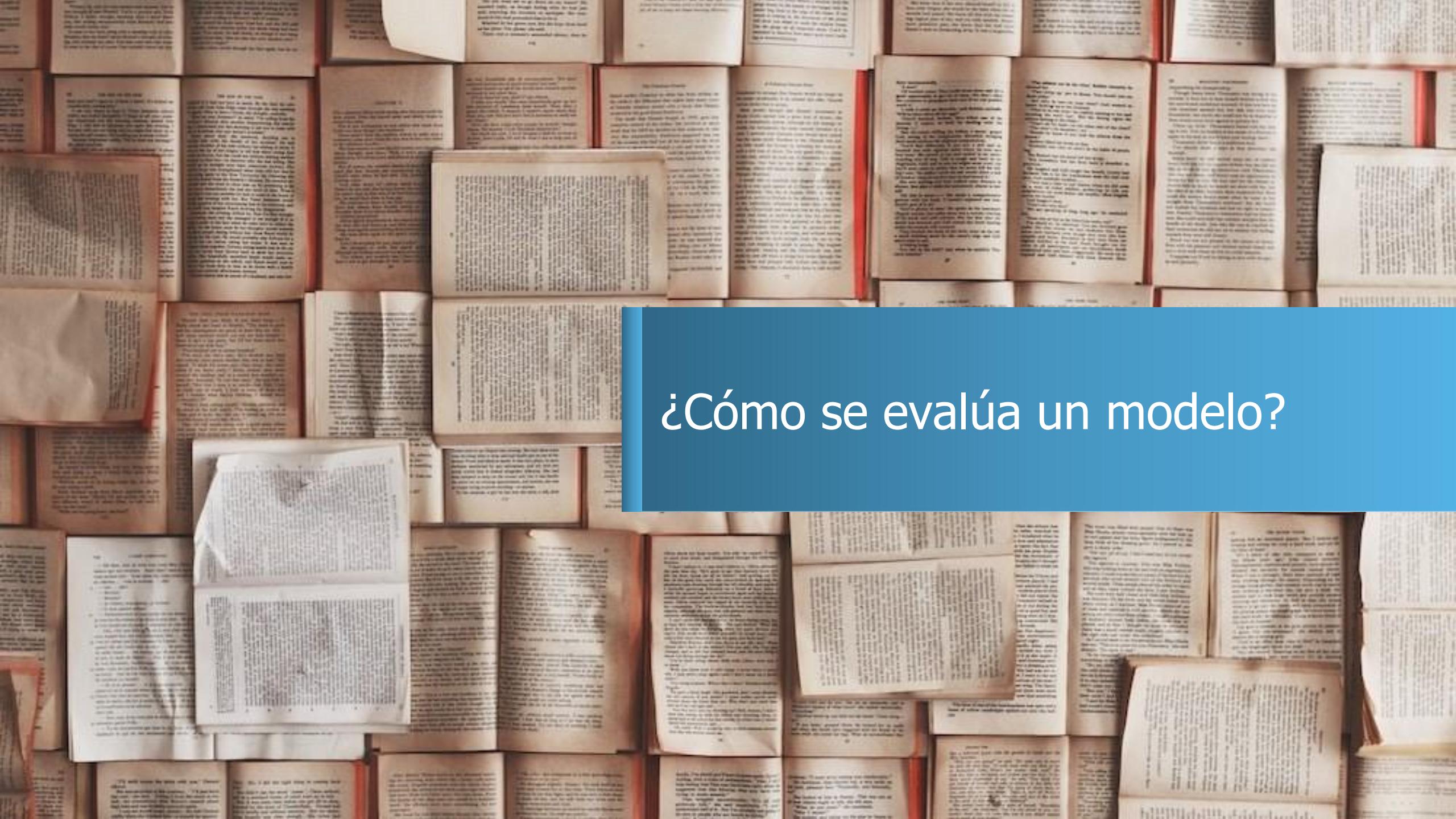
- ¿De donde surge el **error** de nuestros modelos de ML?
- Principalmente: error = sesgo<sup>2</sup> + varianza +  $\epsilon$
- **Varianza (overfitting):**
  - Es la variabilidad de la predicción del modelo para los puntos de datos dados.
  - Proviene de la sensibilidad (flexibilidad) de un modelo a las variaciones en los datos de entrenamiento.
  - Un modelo con alta varianza presta mucha atención a los datos de entrenamiento y no generaliza bien en datos que no ha visto antes.
  - Los modelos con muchos grados de libertad son proclives a una alta varianza (ej.: árbol de decisión con muchos nodos).
  - Si alta varianza → error bajo en los datos de training pero alto en test.



# Bias-variance trade-off

- Normalmente, los **modelos no rinden** bien por sí solos debido a que tienen un **sesgo alto** para ser precisos o una **varianza alta** para ser robustos





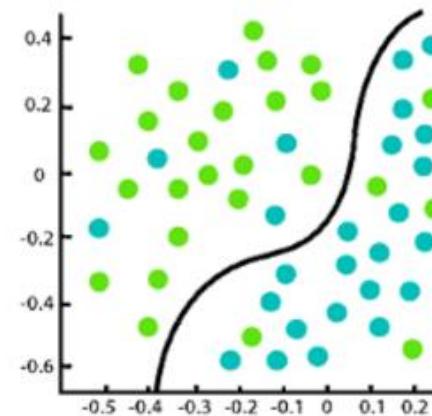
# ¿Cómo se evalúa un modelo?

# ¿Cómo los evaluamos?

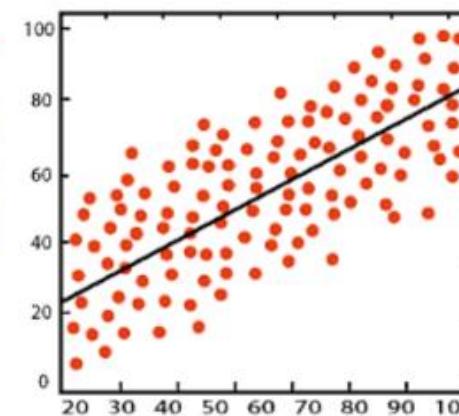
- **Evaluación de los resultados:** Hay infinidad de funciones que nos permiten evaluar qué tan bien se comportan los modelos, pero hay que hacer distinciones:

**¿Qué tipo de problema estamos solucionando?**

KNN  
Árboles de decisión  
Redes neuronales



Classification



Regression

Regresiones  
Árboles de regresión  
Redes neuronales

# Evaluación de modelos de clasificación

- **Métricas de clasificación**
  - % aciertos / accuracy
  - Matrices de confusión: 2 clases
  - Matrices de confusión: múltiples clases
  - Métricas derivadas de la matriz de confusión
  - Curvas ROC
- **Métricas de regresión**
- **Validación del método de aprendizaje (del modelo)**
- **Comparación de distintos métodos de aprendizaje**

# Evaluación de modelos de clasificación

**Exactitud (accuracy) = % Aciertos**

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

$$\text{Error} = 1 - \text{Accuracy}$$

- Es la más simple y fácil de interpretar
- No es representativa en conjuntos de datos desbalanceados

# Evaluación de modelos de clasificación

**Exactitud (accuracy) = % Aciertos**

**Ejemplo:**

- Tienes 100 imágenes de lunares:
  - 95 son lunares (no perjudicial para la salud)
  - 5 son melanomas (cáncer de piel)
- Un modelo podría comportarse de esta forma:
  - El modelo siempre predice "lunar", sin importar la imagen
- Resultado:
  - Clasifica correctamente las 95 imágenes de lunares (no peligrosos)
  - Falla en las 5 imágenes de melanomas (las predice como "lunar").
- A pesar de tener un **95% de accuracy**, el modelo es inútil para detectar melanomas, porque nunca acierta en esa clase

**La accuracy NO es válida para evaluar modelos entrenados con conjuntos de datos desbalanceados**

# Evaluación de modelos de clasificación

## Matrices de confusión

Nos permiten evaluar qué clases se han identificado correctamente, cuáles no y qué instancias se han clasificado mal para poder estudiar a fondo qué ha pasado, dando lugar a una reformulación del procesado inicial, arquitectura etc.

Además se pueden derivar en muchas métricas en función de los resultados que estamos buscando o qué errores nos podemos llegar a permitir.

	Predicted Positive	Predicted Negative
Actual Positive	TP <i>True Positive</i>	FN <i>False Negative</i>
Actual Negative	FP <i>False Positive</i>	TN <i>True Negative</i>

# Evaluación de modelos de clasificación

## Matrices de confusión

- **Recall (sensibilidad):**  
¿Qué porcentaje de los melanomas fueron detectados correctamente?
- **Specificity (especificidad):**  
¿Qué porcentaje de los lunares corrientes fueron identificados correctamente como no melanomas?
- **Precision:**  
Para las predicciones clasificadas como melanomas, ¿qué porcentaje realmente eran melanomas?
- **F1-Score:**  
¿Cuál es el balance entre precisión (precision) y sensibilidad (recall) para la detección de melanomas?

# Evaluación de modelos de clasificación

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

# Evaluación de modelos de clasificación

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	



$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

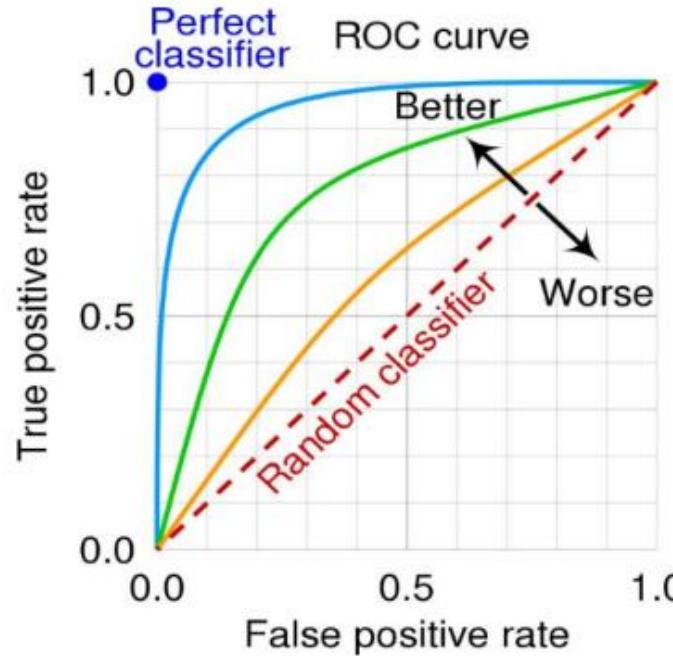


- $Precision = \frac{23}{23+5} = 0,821$
- $NPV = \frac{31}{31+8} = 0,794$
- $Recall = \frac{23}{23+8} = 0,791$
- $Specificity = \frac{31}{31+5} = 0,861$
- $Accuracy = \frac{23+31}{23+31+5+8} = 0,805$
- $F1 = 2 \frac{\frac{23}{23+5} * \frac{23}{23+8}}{\frac{23}{23+5} + \frac{23}{23+8}} = 2 \frac{0,649}{1,612} = 0,805$

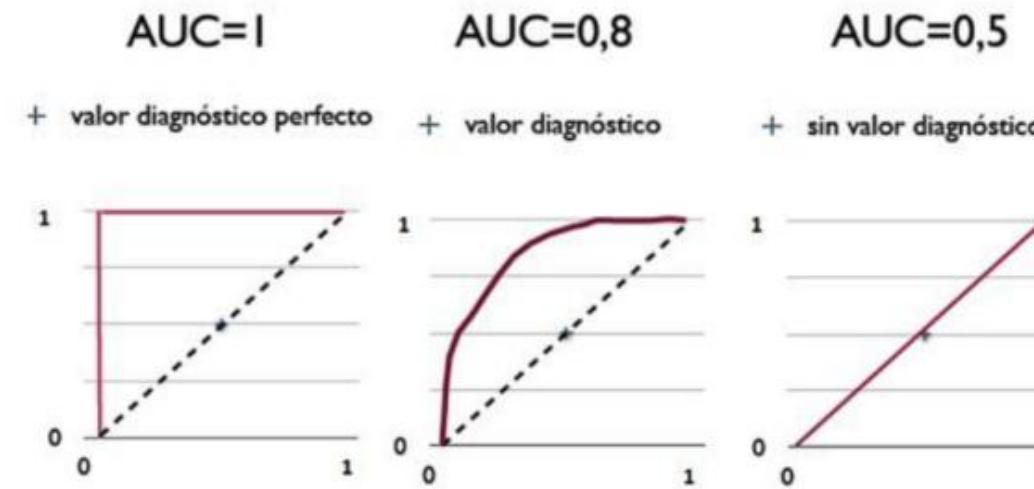
# Evaluación de modelos de clasificación

		Predicted class			Total
Actual class		Class a	Class b	Class c	
	Class a	88	10	2	100
	Class b	14	40	6	60
	Class c	18	10	12	40
	Total	120	60	20	

# Evaluación de modelos de clasificación



- **Curva ROC (Receiver Operating Characteristic):** Representa la sensibilidad y la tasa de falsos positivos (1-especificidad), eje y/x respectivamente.
- **Área bajo la curva (AUC):** Se integra la curva ROC a modo de medida de bonanza.

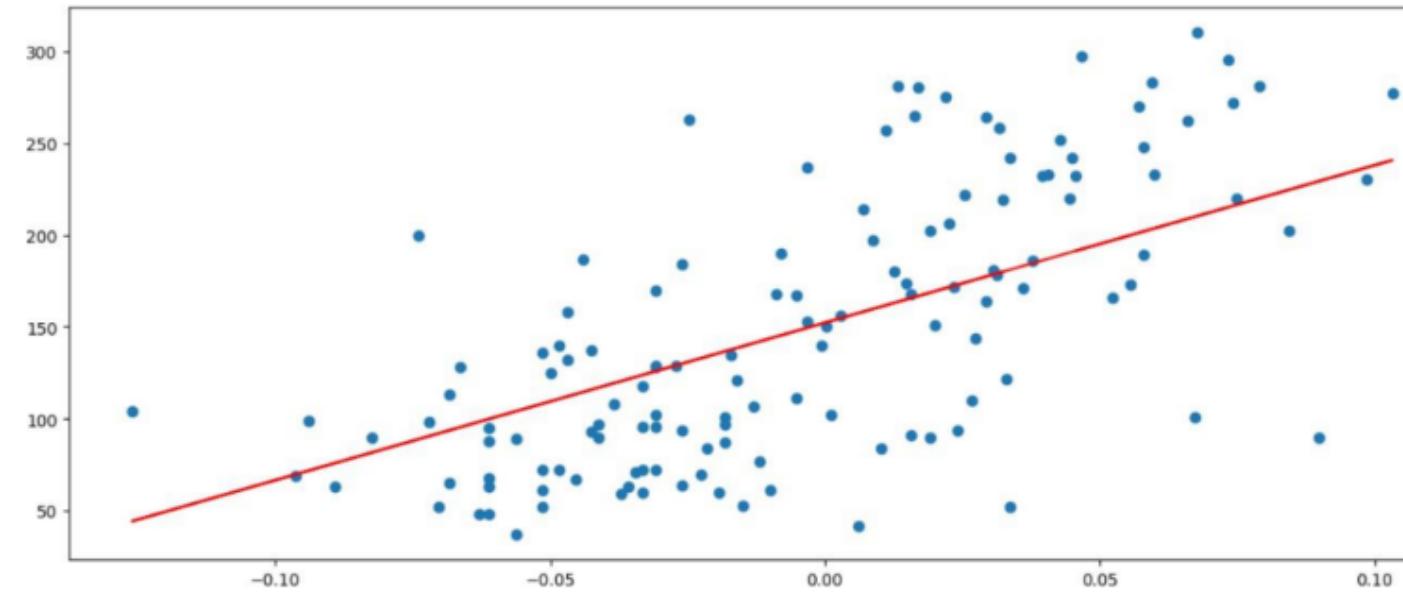


# Métodos de validación

## Métricas de regresión

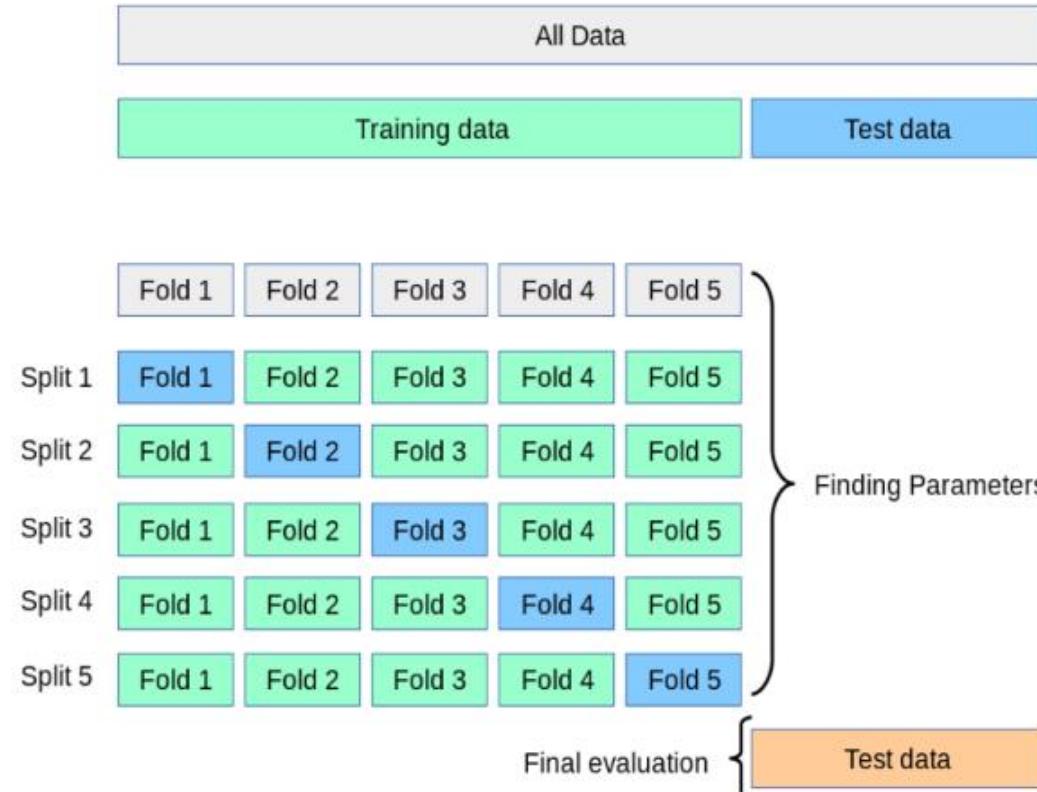
Funciones que nos permiten calcular de cuánto nos hemos equivocado en nuestra predicción y que la mayoría de ellas vienen derivadas de una misma función pero que nos permiten conocer ciertas características.

- Mean absolute error (MAE)
- Mean squared error (MSE)
- Root mean squared error (RMSE)
- ...



# Evaluación de modelos de regresión

- **K-fold cross validation:** Separar en K grupos y entrenar/evaluar con los diferentes grupos para tener una medida estadística de los resultados y darle consistencia.



# Comparando distintos métodos de aprendizaje

	Dataset 1	Dataset 2	....	Dataset K
Algorithm 1	% acc (std)	% acc (std)	....	% acc (std)
Algorithm 2	% acc (std)	% acc (std)	....	% acc (std)
:	:	:	...	:
Algorithm N	% acc (std)	% acc (std)	....	% acc (std)

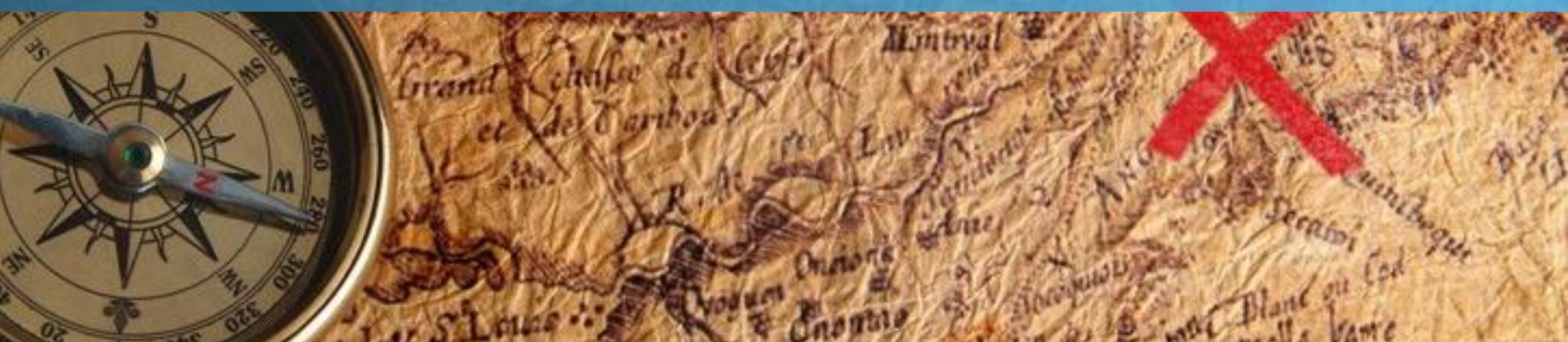
Which one is the best?

# Comparando distintos métodos de aprendizaje

Modelo	MMLU (5-shot)	HellaSwag	HumanEval	MATH	TruthfulQA	Avg. Score
GPT-4	86.4%	95.3%	67.0%	42.5%	59.2%	70.1%
Claude 3.5 Sonnet	85.7%	94.8%	65.5%	41.0%	58.7%	69.1%
Llama 3.1 405B	84.5%	93.2%	63.0%	39.5%	57.3%	67.5%
Gemini 1.5 Pro	83.0%	92.5%	61.5%	38.0%	56.0%	66.2%
Mistral 7B	80.0%	90.0%	58.0%	35.0%	52.0%	63.0%

- **MMLU (Massive Multitask Language Understanding):** Evalúa la comprensión multitarea en 57 temas diferentes.
- **HellaSwag:** Mide la capacidad de razonamiento de sentido común.
- **HumanEval:** Prueba la capacidad de generar código funcional.
- **MATH:** Evalúa la habilidad para resolver problemas matemáticos.
- **TruthfulQA:** Mide la veracidad de las respuestas generadas.

# 4. Algoritmos de aprendizaje automático en SBC



**MD005 – Sistemas Basados en el Conocimiento**

Manel Cerezo García- [manel.cerezo@salle.url.edu](mailto:manel.cerezo@salle.url.edu)

David Larrosa- [david.larrosa@salle.url.edu](mailto:david.larrosa@salle.url.edu)