



Sociedad Mexicana
de Materiales A.C.

TRONCO COMÚN

Módulo 1:

Introducción a la Minería de Datos

Dr. Irvin Hussein López Nava

Centro de Investigación Científica y de Educación Superior de Ensenada
Facultad de Ciencias, Universidad Autónoma de Baja California



Este módulo

Módulo del **tronco común**

- Introducción a la minería de datos
- Aprendizaje de máquina
- Aprendizaje profundo

8 horas (teoría + prácticas)

Propósito del módulo:

Establecer las bases conceptuales y metodológicas de la minería de datos como componente fundamental de la inteligencia artificial aplicada.



Relación con el diplomado

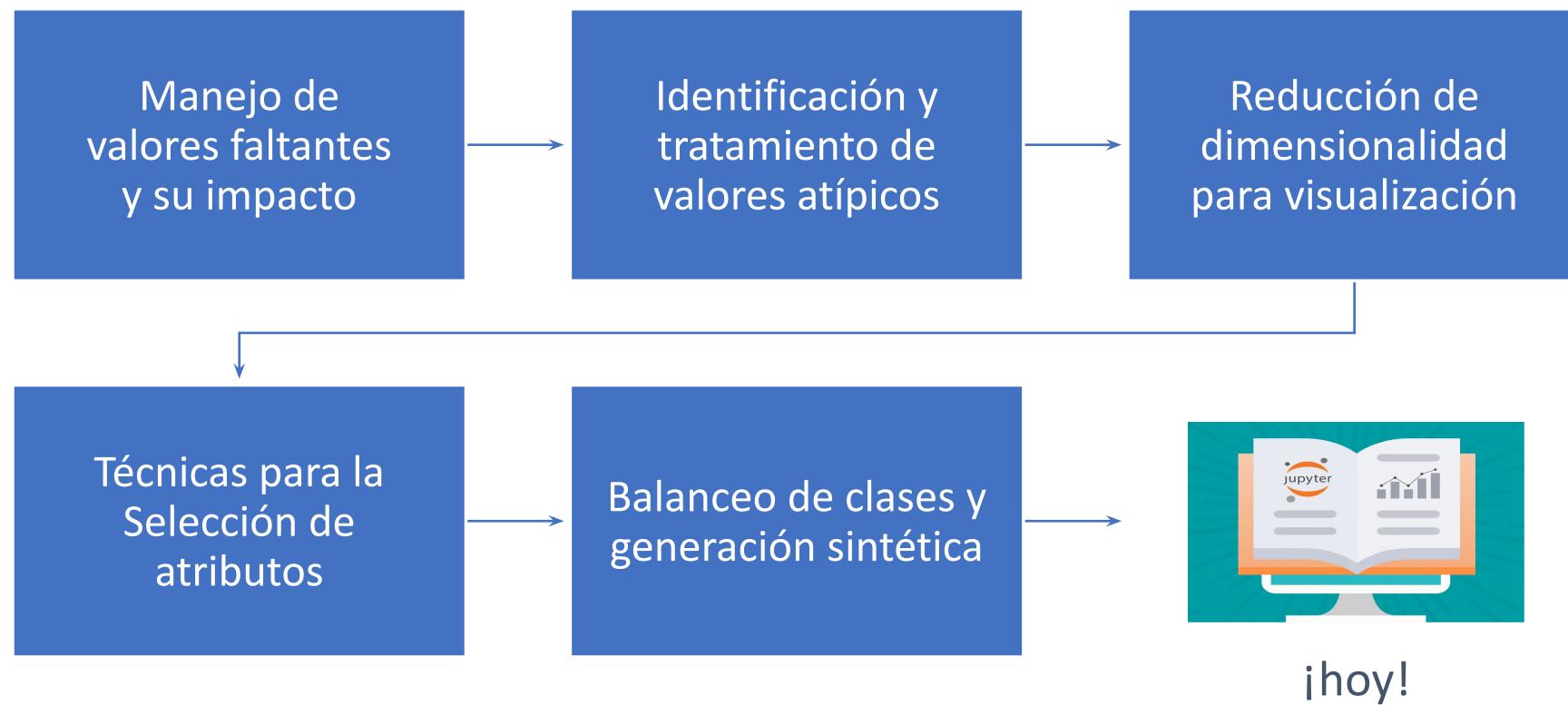
El **diplomado** busca:

- Brindar conocimientos multidisciplinarios en IA.
- Desarrollar competencias teóricas y prácticas.
- Enfocarse en aplicaciones en ciencia de materiales y procesos industriales.

La **minería de datos** contribuye directamente a:

- Identificación y análisis de problemas complejos.
- Preparación y estructuración de datos reales.
- Fundamento para modelos de aprendizaje automático y profundo.

¿En qué nos quedamos?



¡hoy!

TEMARIO

Martes 10/feb

Introducción (1 hora)

- ¿Qué es la minería de datos?
- Conceptos básicos
- Visualización

Ejercicios (2 horas)

- Visualización de datos genéricos
- Exploración de datos reales

Limpieza (1 hora)

- Preprocesamiento
- Reducción de dimensionalidad
- Selección de atributos
- Balanceo de clases

Ejercicios (2 horas)

- Limpieza
- Reducción
- Selección
- Balanceo

Martes 17/feb

Evaluación (1 hora)

- Partición de datos
- Métricas de rendimiento

Ejercicios (1 hora)

- Validación de modelo simple
- Análisis de resultados

FUNDAMENTOS DEL APRENDIZAJE SUPERVISADO

Formulación del problema

En **aprendizaje automático supervisado** se dispone de un conjunto de **observaciones etiquetadas** (x_i, y_i) , donde $x_i \in X$ representa el vector de atributos y $y_i \in Y$ la variable objetivo.

El objetivo consiste en estimar una función $f: X \rightarrow Y$ que permita predecir correctamente el valor de y para nuevas observaciones no vistas.

Esta formulación establece un problema de aproximación funcional bajo información incompleta sobre la distribución real de los datos.



Hipótesis y espacio de modelos

En la práctica, no se busca cualquier función posible, sino una **función** perteneciente a una familia paramétrica definida por un conjunto de parámetros θ .

La elección de esta familia —lineal, no lineal, basada en árboles, redes neuronales, etc.— restringe el espacio de soluciones posibles.

Este espacio de hipótesis determina la capacidad del modelo para capturar relaciones complejas en los datos.



Función de pérdida

La función de pérdida $L(y, f(x; \theta))$ mide el costo asociado a una predicción y define el criterio que conecta el modelo con el objetivo del problema.

- En **regresión**, la pérdida cuadrática penaliza desviaciones proporcionalmente al cuadrado del error, mientras que la pérdida absoluta introduce mayor robustez frente a valores extremos.
- En **clasificación**, no solo se penalizan errores de etiqueta, sino también la confianza incorrecta del modelo, incorporando información probabilística.

La minimización de la **pérdida promedio** sobre los **datos de entrenamiento** define el criterio de ajuste del modelo.



Riesgo empírico

El **riesgo empírico** se define como: $\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i; \theta))$

Este valor es una estimación del **desempeño del modelo** sobre los datos **observados** y constituye el objetivo directo de los algoritmos de entrenamiento.

Sin embargo, el **riesgo empírico** es una estimación condicionada al conjunto disponible y puede subestimar el error real.



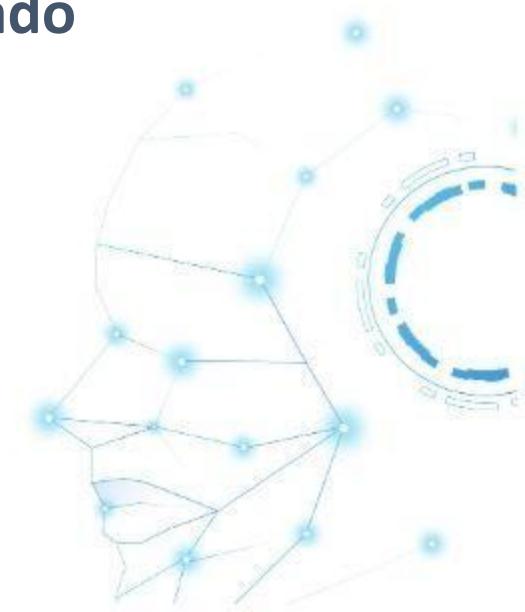
Riesgo esperado

El **riesgo esperado** se define como: $R(\theta) = \mathbb{E}_{(x,y) \sim P(X,Y)} [L(y, f(x; \theta))]$

Este valor representa el **desempeño promedio** del modelo sobre la **distribución real** de datos, la cual es desconocida.

El problema fundamental consiste en aproximar este **riesgo esperado** utilizando únicamente una **muestra finita**.

La diferencia entre $R(\theta)$ y $\hat{R}(\theta)$ es el núcleo del problema de **generalización**.



Brecha de generalización

La **brecha de generalización** depende tanto del **tamaño de la muestra** como de la **complejidad del modelo**.

Modelos más complejos pueden reducir el riesgo empírico, pero incrementan la varianza de la estimación del riesgo esperado.

Este fenómeno explica por qué un ajuste perfecto en entrenamiento puede corresponder a un desempeño pobre fuera de muestra.

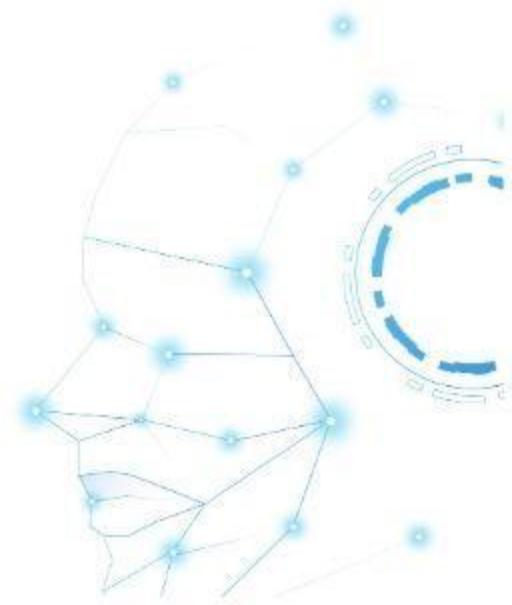
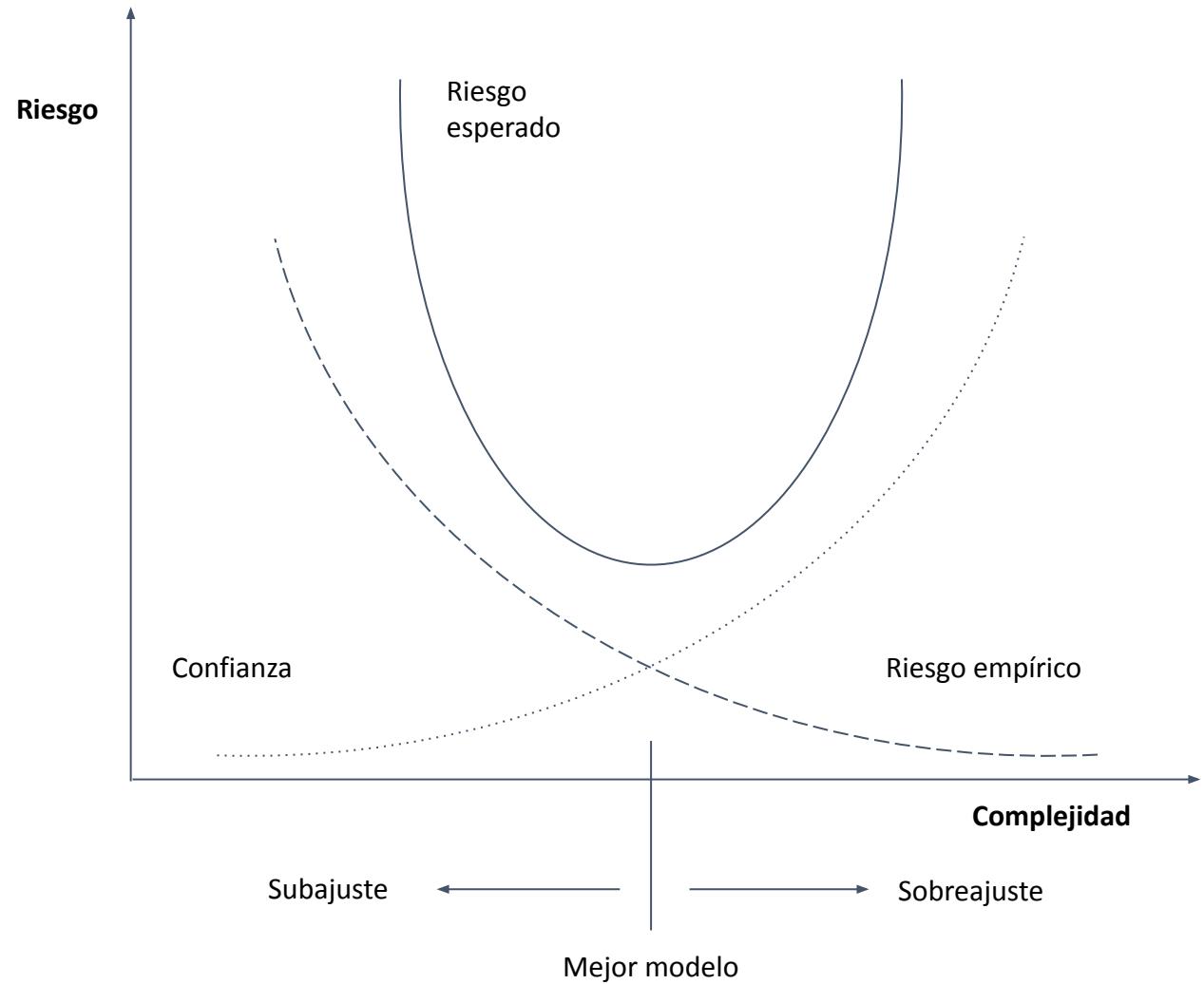


Sobreajuste y subajuste

El **sobreajuste** ocurre cuando el modelo reduce el riesgo empírico a costa de aumentar la brecha de **generalización**.

El **subajuste** ocurre cuando la capacidad del modelo es insuficiente para capturar la estructura subyacente.

Ambos fenómenos deben evaluarse en función del desempeño fuera de muestra, no del error de entrenamiento.



¿Qué significa realmente “aprender”?

El **aprendizaje automático supervisado** puede entenderse entonces como un proceso de optimización basado en una muestra finita.

- El **entrenamiento** minimiza una función de pérdida sobre los datos disponibles, pero el objetivo real es capturar regularidades que se mantengan fuera del conjunto observado.

El desafío central consiste en equilibrar:

- ajuste adecuado a los datos disponibles,
- capacidad de generalización a datos nuevos.

Este **equilibrio** depende de la complejidad del modelo y de las decisiones adoptadas durante el diseño del experimento.



DISEÑO EXPERIMENTAL Y PARTICIÓN DE DATOS

¿Por qué particionar los datos?

En la sección anterior vimos que el objetivo del aprendizaje es aproximar el riesgo esperado $R(\theta)$, el cual depende de la **distribución real** de los datos.

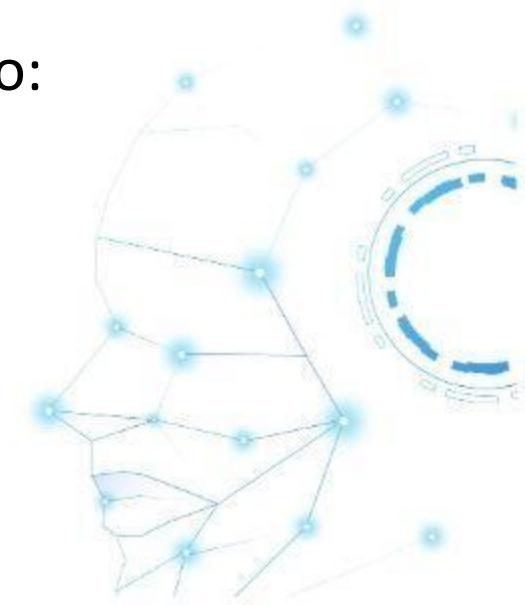
Sin embargo:

- El modelo se ajusta minimizando el riesgo empírico.
- Este se calcula sobre los mismos datos utilizados para entrenar.

Si evaluamos el modelo sobre el mismo conjunto de entrenamiento:

- El error será optimista
- No estimaremos la capacidad real de generalización.

La **partición de datos** es el mecanismo práctico para aproximar el desempeño fuera de la muestra.



Partición entrenamiento / prueba

La estrategia más básica consiste en dividir el conjunto disponible D en dos subconjuntos disjuntos:

- **Conjunto de entrenamiento:** Se utiliza para ajustar los parámetros θ .
- **Conjunto de prueba:** Se utiliza exclusivamente para estimar desempeño.

Formalmente: $D = D_{\text{train}} \cup D_{\text{test}}$ $D_{\text{train}} \cap D_{\text{test}} = \emptyset$

El supuesto clave es que ambos subconjuntos provienen de la misma distribución.

Esquemas de evaluación

Hold-out simple

k-fold
cross-validation

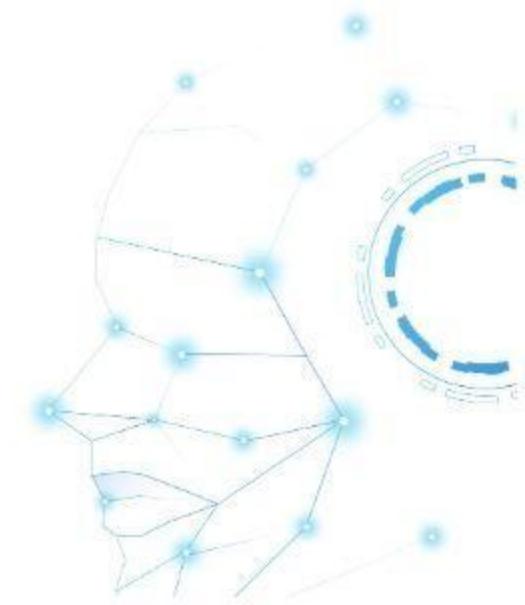
Leave-One-Out
cross-validation

Nested
cross-validation

La diferencia entre ellos radica en:

- Cómo se reutilizan los datos.
- El balance entre sesgo y varianza en la estimación.
- El costo computacional.

La elección del esquema es una decisión metodológica,
no meramente técnica.



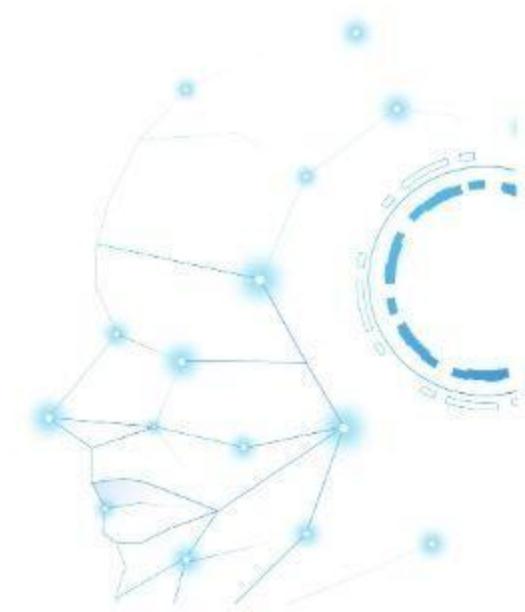
Hold-Out simple

Una sola partición implica que:

- El desempeño estimado depende de cómo se dividieron los datos.
- Cambiar la semilla puede cambiar el resultado.
- Con pocos datos, el estimador puede tener alta varianza.

Por lo tanto, el desempeño observado no es un número absoluto, sino una realización aleatoria condicionada a la partición.

Esto introduce el problema de variabilidad en la estimación del desempeño.



k-Fold Cross-Validation

1. Se divide el conjunto en k subconjuntos del mismo tamaño.
2. En cada iteración:
 - Uno se usa como prueba.
 - Los restantes $k-1$ como entrenamiento.
3. Se promedian los resultados.

Ventajas:

- Cada observación se usa tanto para entrenamiento como para evaluación
- Reduce varianza respecto al hold-out simple.
- Es el estándar práctico en evaluación comparativa..

El valor típico es $k=5$ (cinco iteraciones: 80% *training*, 20% *testing*).

Leave-One-Out Cross-Validation

LOOCV es el caso extremo donde $k = n$.

Cada iteración:

- Se entrena con $n-1$ observaciones.
- Se evalúa con una sola.

Propiedades:

- Bajo sesgo en la estimación.
- Alta varianza.
- Alto costo computacional.

Se utiliza principalmente en contextos con datasets pequeños.



Nested Cross-Validation

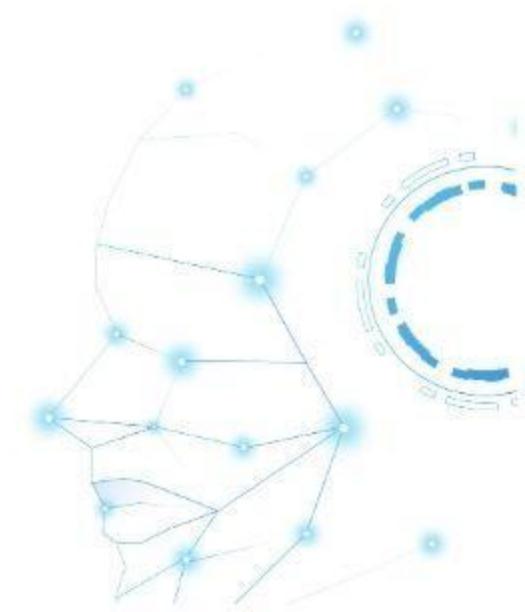
Cuando se optimizan hiperparámetros:

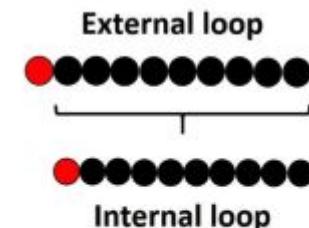
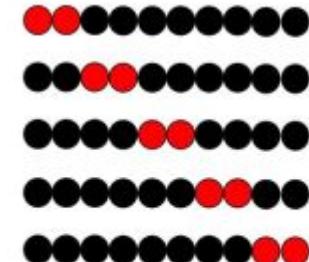
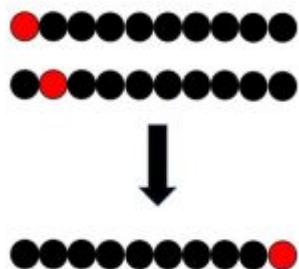
- Se selecciona el mejor modelo según desempeño.
- Si se usa el mismo esquema para evaluar, se introduce optimismo.

Nested CV separa:

- Bucle interno: selección de hiperparámetros.
- Bucle externo: estimación de desempeño.

Esto evita que el proceso de selección contamine la evaluación final.
Es el estándar en estudios comparativos rigurosos.





Nested



Estratificación

En problemas de clasificación con desbalance:

- Una partición aleatoria puede alterar las proporciones de clase.
- Esto distorsiona métricas como precisión o recall.

La estratificación garantiza que:

- Cada subconjunto preserve aprox. la proporción original de clases.
- La estimación del desempeño sea más estable.

Es especialmente importante cuando la clase minoritaria es pequeña o cuando el *dataset* es reducido.



Estimación de varianza del desempeño

El desempeño de un modelo **NO es un valor fijo**.

Es una variable aleatoria dependiente de:

- La muestra disponible.
- La partición generada.
- El modelo utilizado.

Por ello es recomendable reportar al menos:

- Métrica de tendencia central, e.g., media.
- Métrica de dispersión, e.g., desviación estándar.

Un modelo con desempeño ligeramente mayor pero alta varianza puede no ser preferible.



Data Leakage

La fuga de información ocurre cuando la información del **conjunto de prueba** influye en el **entrenamiento**.

Ejemplos frecuentes:

- Escalar antes de dividir.
- Seleccionar variables con todo el dataset.
- Aplicar aumento de datos antes de la partición.

Consecuencia:

- Se subestima el error real.
- Se obtiene una evaluación artificialmente optimista.

La prevención del *leakage* es parte del diseño experimental.



MÉTRICAS DE RENDIMIENTO

Evaluar modelos de regresión

En problemas de regresión:

- La variable objetivo es continua.
- El modelo produce una predicción numérica $\hat{y}_i = f(x_i; \theta)$

La evaluación consiste en cuantificar la discrepancia entre:

y (*valor real*)

\hat{y} (*predicción*)

Las métricas se derivan de la **función de pérdida** definida en el marco teórico del riesgo empírico.



Error absoluto medio (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Interpretación:

- Promedio de las desviaciones absolutas.
- Penaliza errores linealmente.
- Está en la misma unidad que la variable objetivo.

Propiedad importante:

- Es más robusto a valores atípicos que métricas cuadráticas.



Error Cuadrático Medio (MSE)

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

Interpretación:

- Penaliza errores grandes de forma cuadrática.
- Sensible a outliers.
- Relacionado directamente con la pérdida cuadrática en el entrenamiento.

El MSE es la métrica natural cuando el modelo se ajusta mediante mínimos cuadrados.



Raíz del Error Cuadrático Medio (RMSE)

$$\text{RMSE} = \sqrt{\text{MSE}}$$

Ventaja principal:

- Devuelve la métrica a la escala original de la variable objetivo.
- Facilita interpretación comparativa.

Sin embargo:

- Mantiene la sensibilidad a errores grandes heredada del MSE.



Coeficiente de Determinación R^2

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

Interpretación:

- Mide la proporción de varianza explicada por el modelo.
- $R^2 = 1 \rightarrow$ ajuste perfecto.
- $R^2 = 0 \rightarrow$ equivalente a predecir la media.
- Puede ser negativo si el modelo es peor que la media.

No mide error absoluto, sino capacidad explicativa relativa.



Sensibilidad a valores atípicos

Comparación conceptual:

- MAE → penalización lineal.
- MSE / RMSE → penalización cuadrática.

Si existen valores extremos:

- El MSE puede estar dominado por pocos puntos.
- El MAE puede ser más representativo del comportamiento general.

Métricas cuadráticas amplifican errores extremos;
métricas lineales los penalizan proporcionalmente.



Evaluar modelos de clasificación

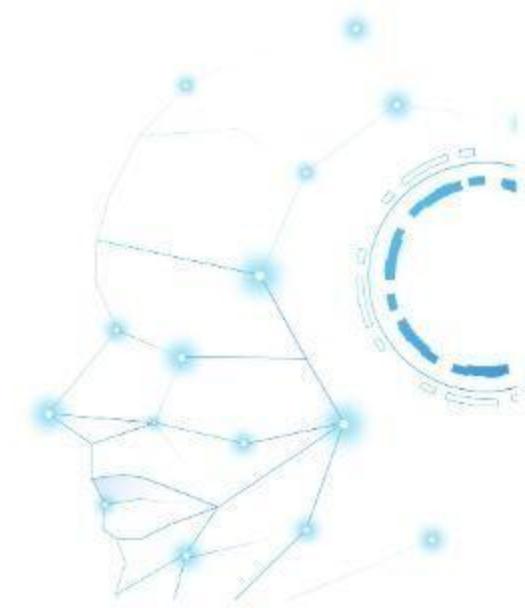
En problemas de clasificación:

- La variable objetivo es categórica.
- El modelo produce una etiqueta predicha \hat{y} o una probabilidad estimada $\hat{p}(y | x)$

La evaluación puede basarse en:

- Comparación directa de etiquetas.
- Análisis de probabilidades.
- Curvas de desempeño según umbral.

Las métricas deben capturar no solo la exactitud, sino la estructura del error.



Matriz de Confusión (caso binario)

Para clasificación binaria, definimos:

- Verdaderos positivos (TP).
- Verdaderos negativos (TN).
- Falsos positivos (FP).
- Falsos negativos (FN).

La matriz resume todos los resultados posibles del clasificador.

Formalmente:

	$\hat{y} = 1$	$\hat{y} = 0$
$y = 1$	TP	FN
$y = 0$	FP	TN



Exactitud (Accuracy)

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Interpretación

- Proporción de predicciones correctas.

Limitación

- Puede ser engañosa en datasets desbalanceados.

Ejemplo

- Si el 95% pertenece a una clase, predecir siempre esa clase produce 95% de accuracy.

Precisión (*Precision*)

$$\text{Precision} = \frac{TP}{TP+FP}$$

Interpretación

- De todas las predicciones positivas, ¿cuántas son correctas?

Importante cuando

- Los falsos positivos son costosos.

Ejemplo

- Diagnóstico médico.
- Detección de fraude.

Sensibilidad (*recall*)

$$\text{Recall} = \frac{TP}{TP+FN}$$

Interpretación

- De todos los casos positivos reales, ¿cuántos detectamos?

Importante cuando

- Los falsos negativos son críticos.

Ejemplo

- Detección de enfermedades.
- Sistemas de alarma.

F1-Score

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Interpretación

- Media armónica entre precisión y recall.
- Penaliza desequilibrios extremos entre ambas.

Útil cuando

- Se requiere balance entre FP y FN.
- Existe desbalance moderado.

Otras métricas: ROC, AUC y MCC

Curva ROC: Evalúa el desempeño variando el umbral y graficando:

$$\text{TPR} = \frac{TP}{TP+FN} \quad \text{FPR} = \frac{FP}{FP+TN}$$

AUC (área bajo la curva): Resume la curva ROC en un valor entre 0 y 1 e indica la probabilidad de que el modelo asigne mayor puntuación a un positivo que a un negativo.

MCC (Matthews Correlation Coefficient): Incorpora todas las celdas de la matriz de confusión y es especialmente robusto en escenarios con clases desbalanceadas.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Selección de métrica según contexto

No existe una métrica universalmente superior.

La elección de métrica debe considerar:

- Naturaleza del problema.
- Distribución de clases.
- Consecuencias prácticas del error.
- Objetivo analítico del estudio.

En escenarios desbalanceados:

- La accuracy puede ser engañosa.
- Es preferible analizar precision, recall, F1 o MCC según el costo relativo de falsos positivos y falsos negativos.



MODELOS Y MARCO METODOLÓGICO

¿Dónde entran los modelos en el proceso?

El **modelado** aparece después de la preparación de los datos y del diseño experimental.

Un **modelo** es una hipótesis estructural sobre la relación entre variables.

No es el eje central del proceso, sino una pieza dentro de un diseño metodológico más amplio.

La **pregunta correcta** no es: ¿Qué algoritmo es mejor?
Sino: ¿Qué hipótesis funcional estoy imponiendo y
cómo la voy a evaluar fuera de la muestra?



Regresión lineal

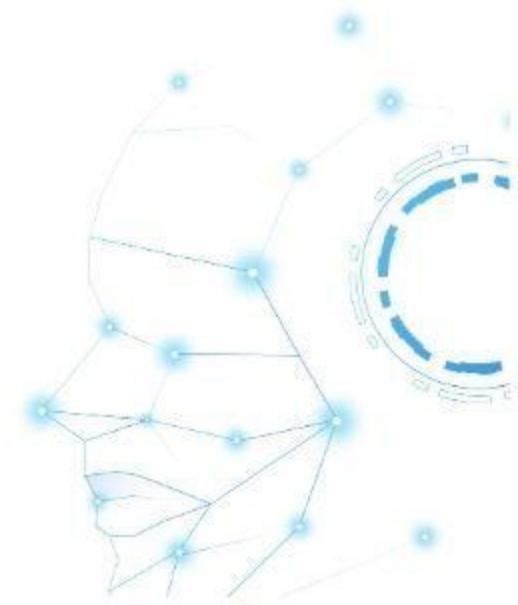
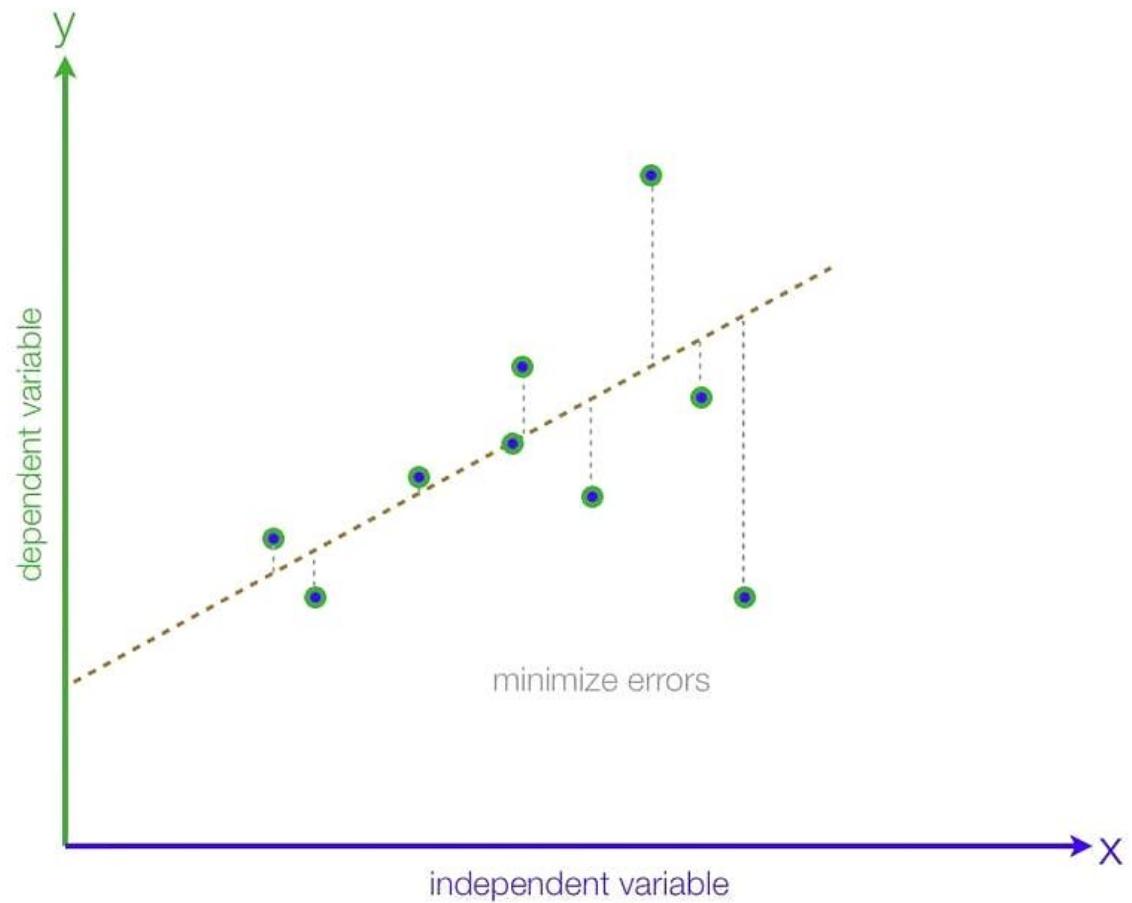
Modelo: $f(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\beta}$

Se estima minimizando el error cuadrático medio: $\min_{\boldsymbol{\beta}} \|\mathbf{y} - X\boldsymbol{\beta}\|^2$

- **Interpretación geométrica:** El vector de predicciones es la proyección ortogonal de \mathbf{y} sobre el subespacio generado por las columnas de X .
- **Hipótesis estructural:** La relación entre variables es lineal y global.

Este modelo sirve como punto de referencia conceptual para comprender otros métodos.





Regularización

Cuando el número de variables crece o existe **colinealidad**, el modelo lineal puede sobreajustar.

$$\text{Ridge (L2): } \min_{\beta} \|\mathbf{y} - X\beta\|^2 + \lambda \|\beta\|_2^2$$

$$\text{Lasso (L1): } \min_{\beta} \|\mathbf{y} - X\beta\|^2 + \lambda \|\beta\|_1$$

- **Interpretación:** La penalización limita la magnitud de los coeficientes y reduce la capacidad efectiva del modelo.
- Lasso puede anular coeficientes, conectando directamente con selección de variables.
- La complejidad deja de ser implícita y pasa a ser un parámetro controlable.



Otros modelos de regularización

Distintas estrategias implican distintas hipótesis:

- **k-NN regression:** la función es local y depende de vecindad.
- **Árboles de regresión:** el espacio se divide en regiones con valores constantes.
- **Support Vector Regression:** se ajusta una función que maximiza margen con tolerancia ϵ .

Cada modelo define una forma distinta para la función objetivo.

No existe modelo universalmente superior: depende de la estructura real de los datos.



Árboles de decisión

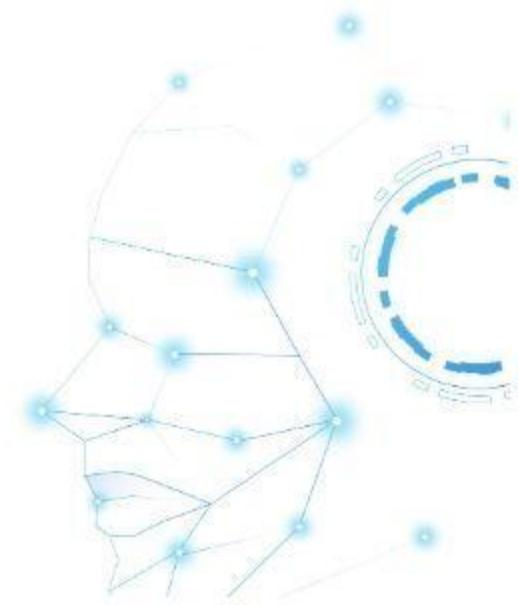
Divide el espacio de atributos mediante reglas binarias sucesivas.

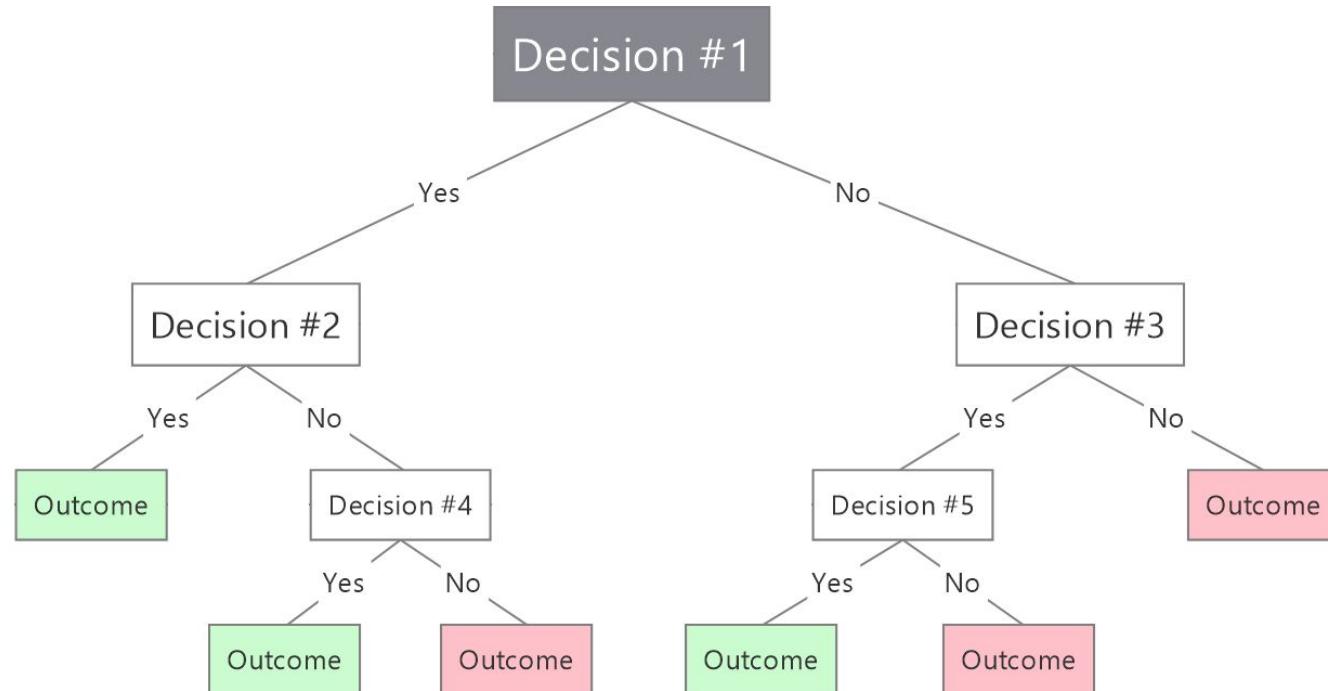
Cada nodo define una condición del tipo: $x_j < c$

Resultado: El espacio queda particionado en regiones.

Propiedades:

- Captura relaciones no lineales.
- Maneja interacciones automáticamente.
- Alta interpretabilidad estructural.





Criterios de división

Criterios típicos: Entropía, o índice de Gini.

Ambos cuantifican impureza de clase.

La complejidad del árbol depende de:

- Profundidad máxima.
- Número mínimo de muestras por hoja.
- Estrategias de poda.

A mayor profundidad, mayor capacidad del modelo.

Sin control estructural, el árbol puede memorizar el conjunto de entrenamiento.



Otros modelos de clasificación

- **Regresión logística:** frontera lineal probabilística.
- **Máquinas de soporte vectorial (SVM):** maximización de margen; posible no linealidad mediante kernels.
- **k-vecinos más cercanos (k-NN):** decisión basada en proximidad local.
- **Naive Bayes:** independencia condicional entre atributos.

Cada algoritmo impone una geometría distinta sobre el espacio de decisión.

Elegir un modelo implica elegir un supuesto estructural.



Meta-modelos y ensambles

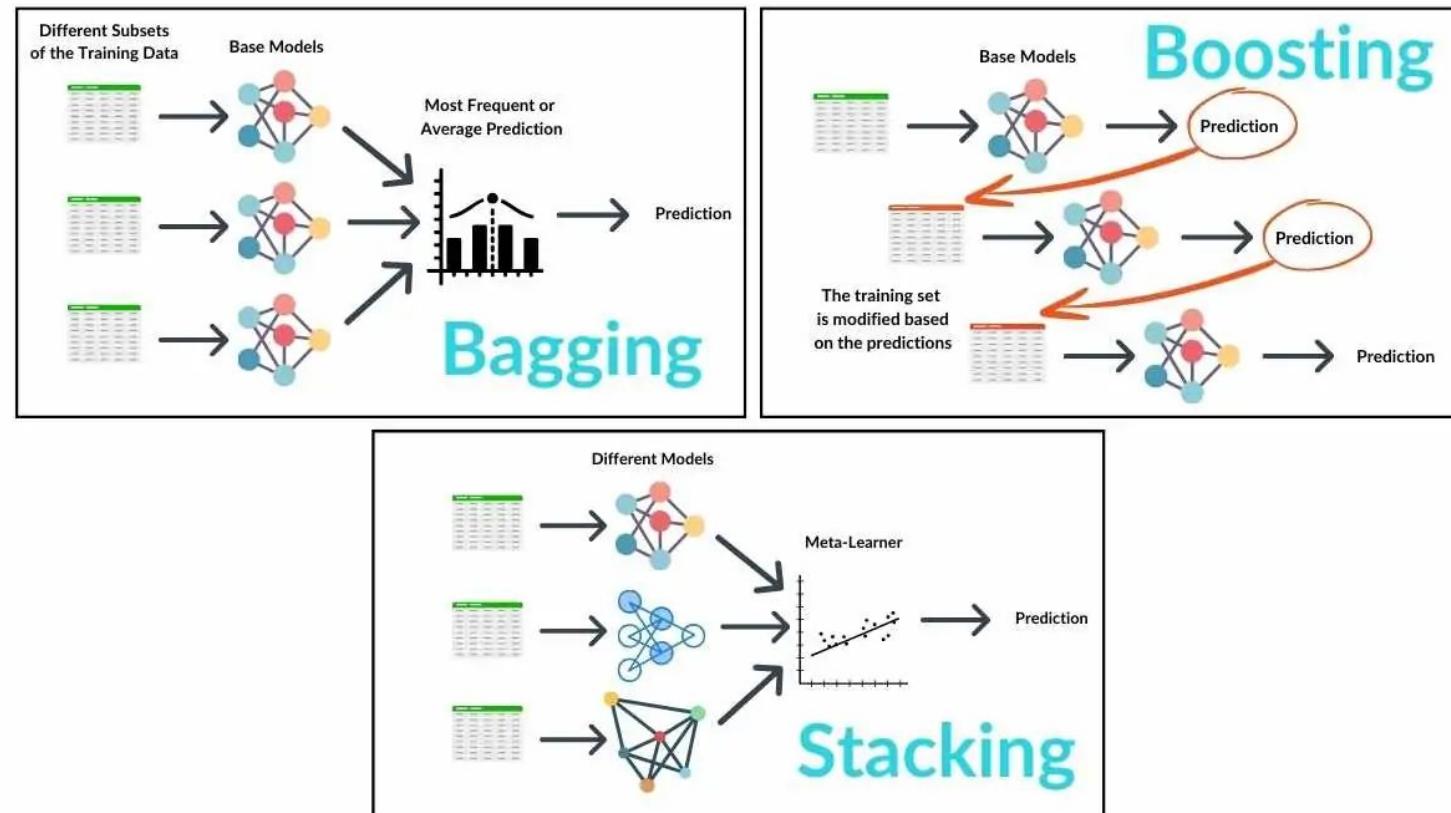
Idea central: Combinar múltiples modelos base para mejorar estabilidad o reducir error.

Ejemplos:

- **Bagging:** reducción de varianza.
- **Random Forest:** árboles + aleatoriedad.
- **Boosting:** ajuste secuencial que reduce sesgo
- **Stacking:** combinación jerárquica de predictores.

La mejora no proviene de un modelo más complejo, sino de la agregación estructurada de múltiples hipótesis.





Redes neuronales

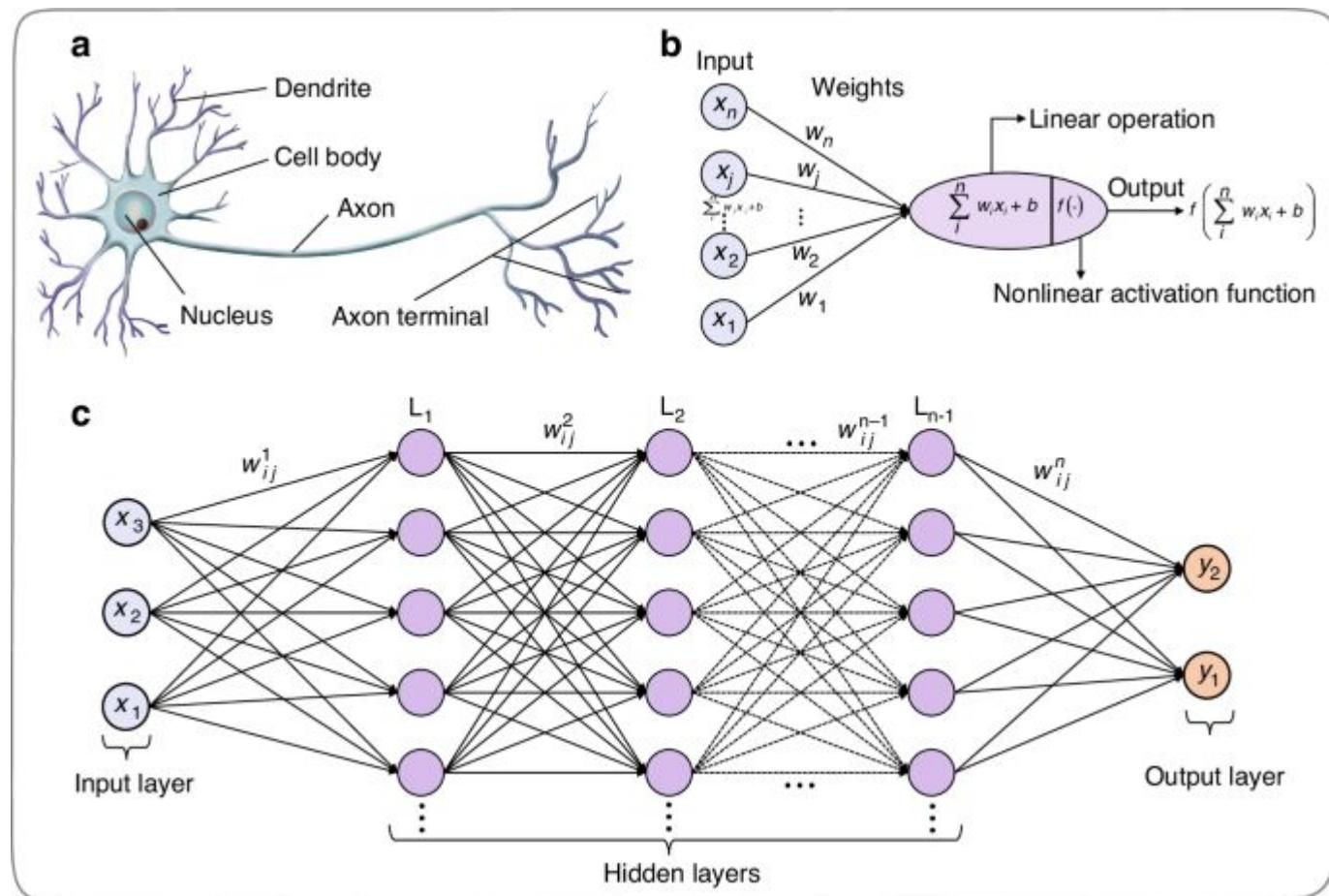
Modelos altamente flexibles capaces de aproximar funciones complejas.
Aprenden representaciones no lineales en múltiples capas.

Características:

- Gran capacidad de ajuste.
- Sensibles a tamaño de muestra.
- Requieren regularización y validación rigurosa.

La alta capacidad implica mayor riesgo de sobreajuste si el diseño experimental es débil.



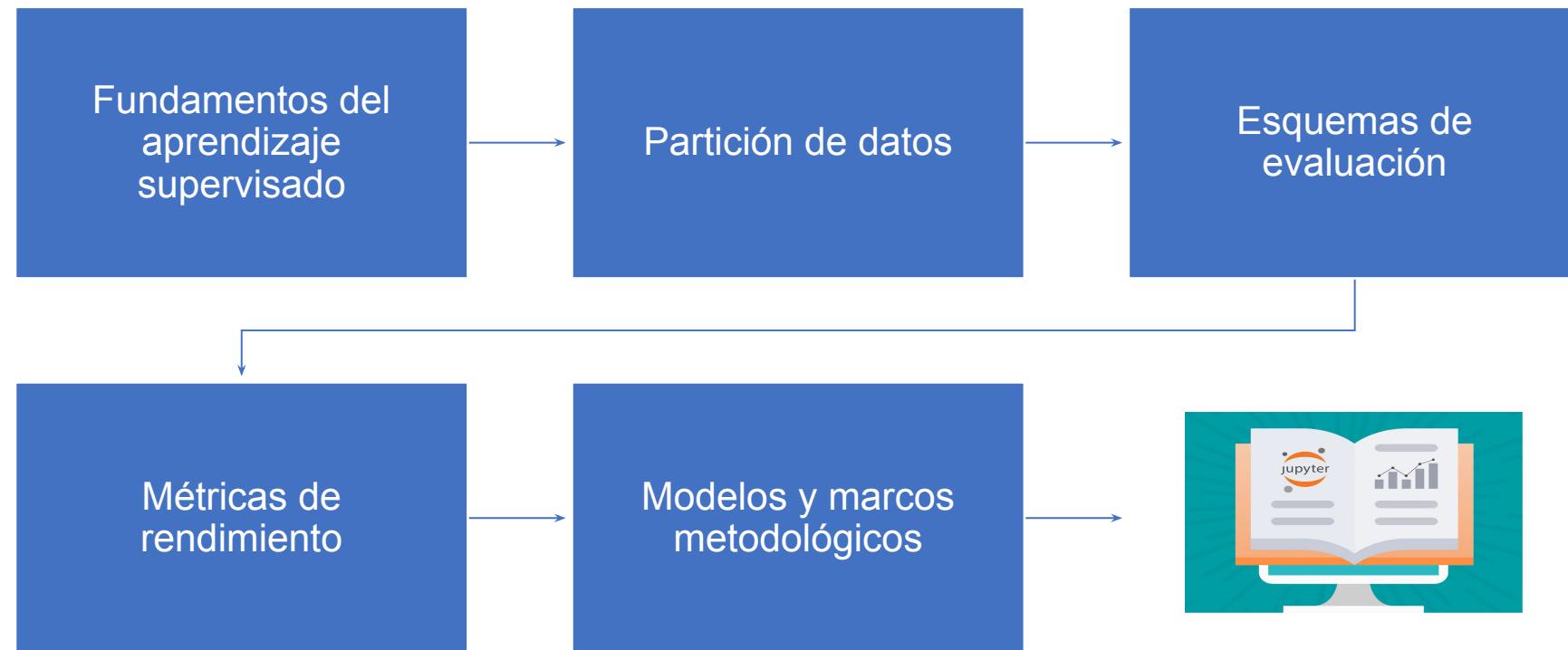


Evaluación como eje metodológico

Todos los modelos anteriores comparten algo:

- Se entrena minimizando una función de pérdida.
- Pero lo que importa es su desempeño fuera de muestra.
- No comparamos algoritmos por su error en entrenamiento.
- Comparamos estimaciones obtenidas bajo un esquema de validación adecuado.
- Sin validación rigurosa, no existe evidencia de generalización.

¿Qué se construyó en esta sección?



Idea central del Módulo 1

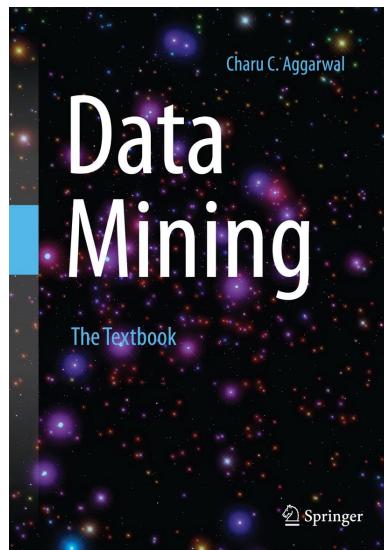
La **minería de datos** no es sólo el entrenamiento de modelos.
Es un **proceso estructurado** para estimar, con rigor, la capacidad de generalización de una hipótesis a partir de datos finitos.

- Formular correctamente el problema.
- Entender la estructura de los datos antes de modelar.
- Controlar la complejidad de las hipótesis.
- Diseñar esquemas de validación que eviten estimaciones optimistas.
- Elegir métricas coherentes con el objetivo analítico.

Este módulo establece la disciplina metodológica que sostiene todo el diplomado.

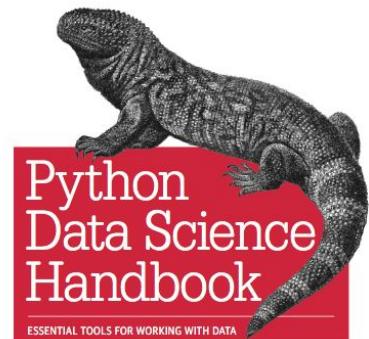


Bibliografía



Aggarwal, C. C. (2015). *Data mining: the textbook* (Vol. 1, No. 3). New York: Springer.

O'REILLY



Jake VanderPlas

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. O'Reilly Media, Inc.

<https://jakevdp.github.io/PythonDataScienceHandbook/>