Evaluación y Optimización de de Modelos: Validación Cruzada Cruzada y Afinamiento de Hiperparámetros

¿Qué es el Underfitting y el Overfitting?

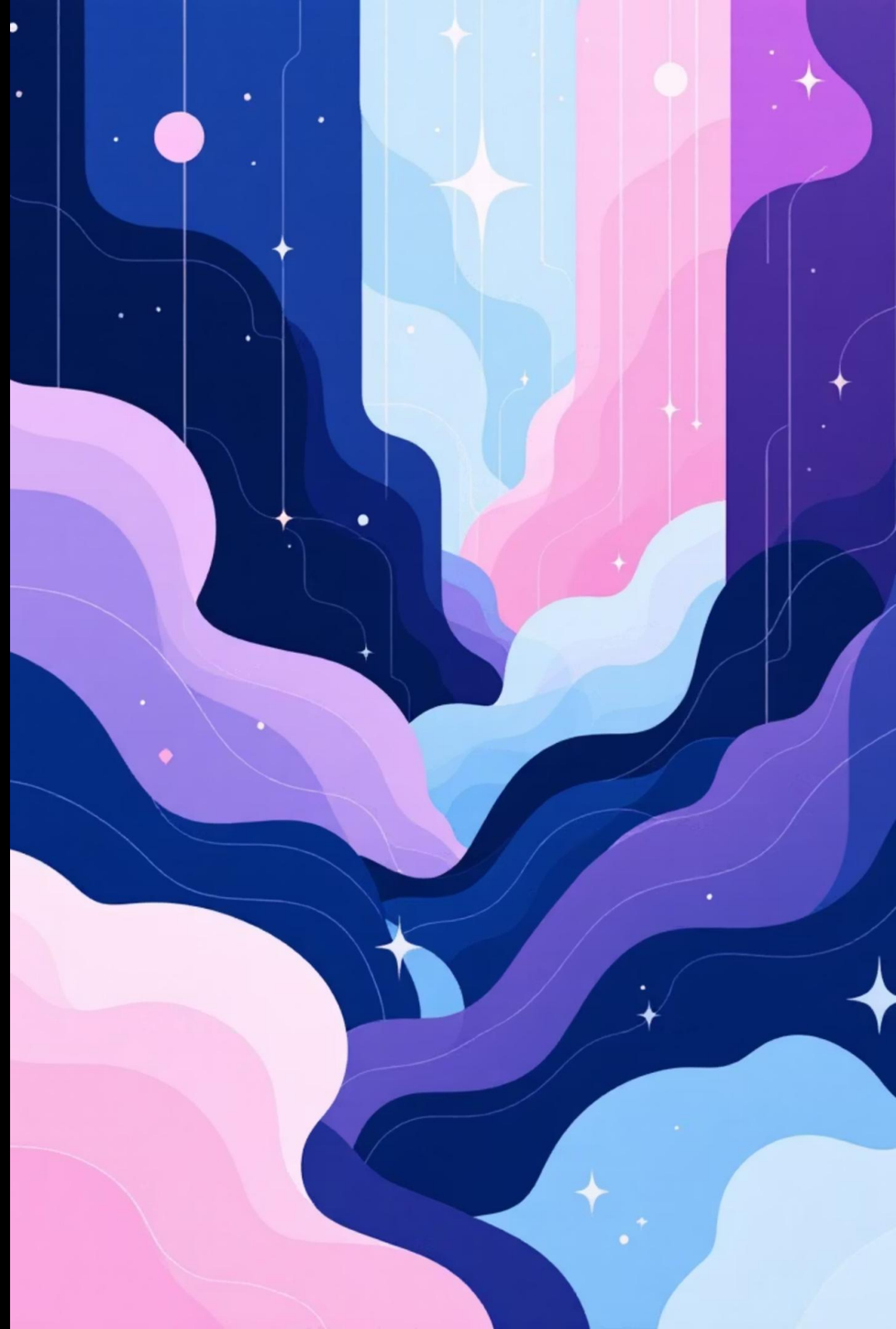
Underfitting

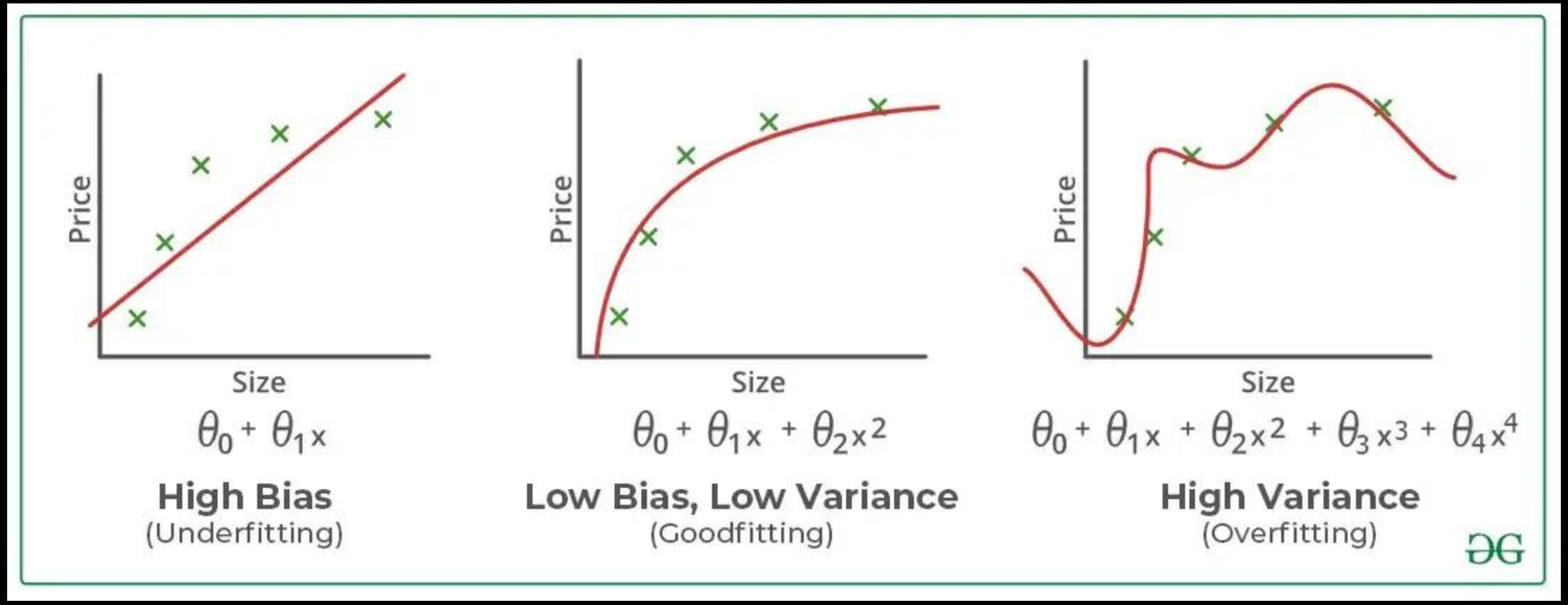
Modelo demasiado simple que no logra capturar los patrones subyacentes de los datos. Muestra baja precisión tanto en entrenamiento como en prueba.

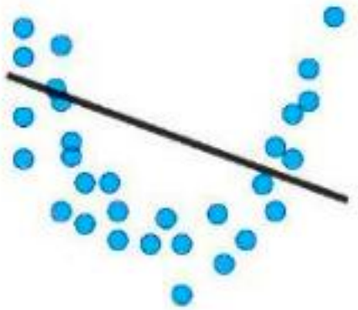
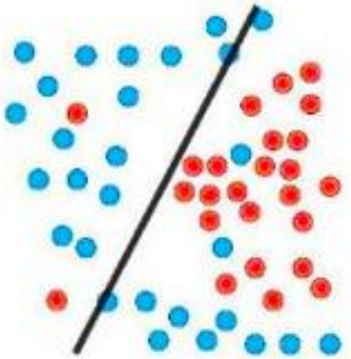
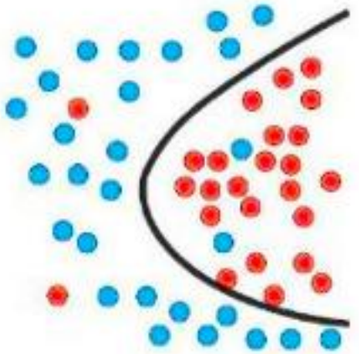
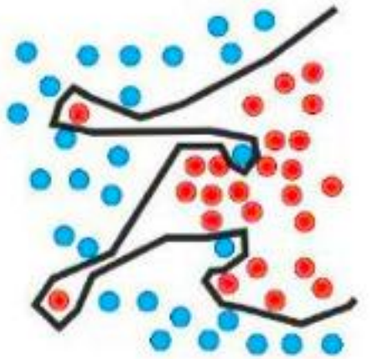
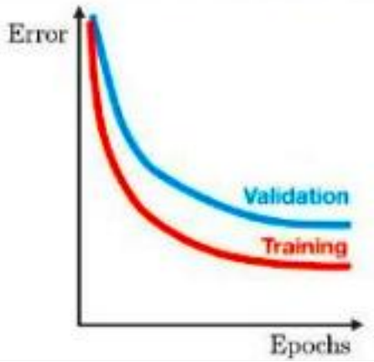
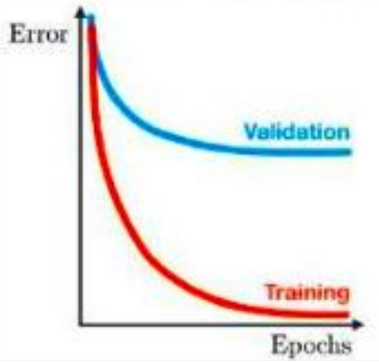
Overfitting

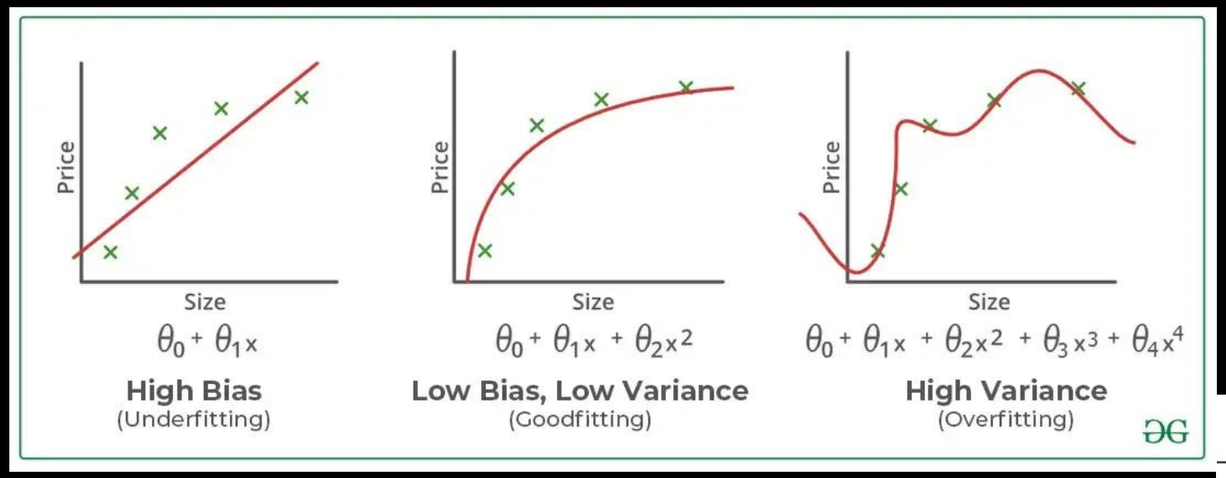
Modelo excesivamente complejo que memoriza ruido y detalles específicos del conjunto de entrenamiento, fallando al generalizar a datos nuevos.

Identificación: Se detectan mediante curvas de error, análisis del desempeño en datos de validación y visualización del ajuste del modelo.



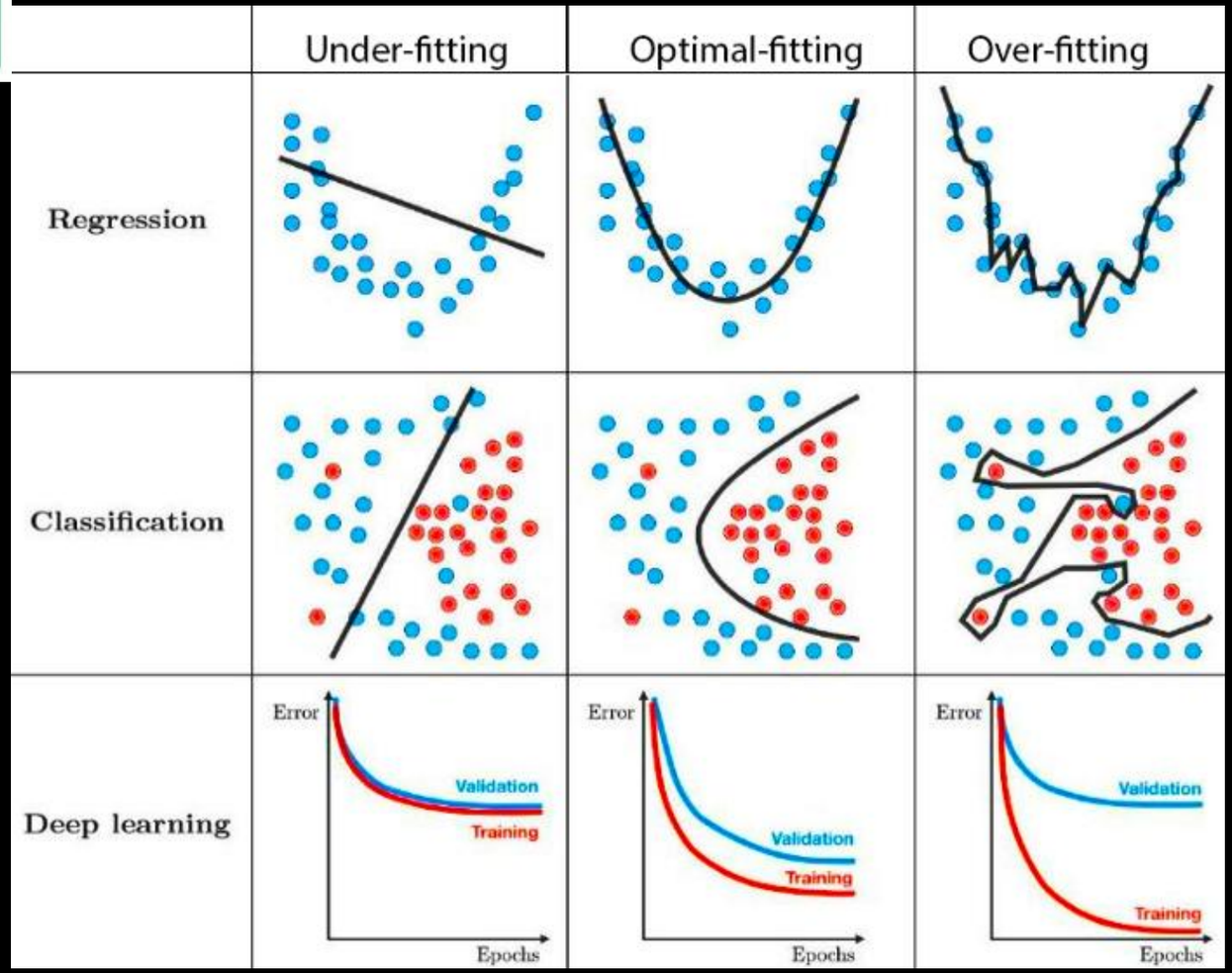

Price
Size
$\theta_0 + \theta_1 x$
High Bias
(Underfitting)
Price
Size
$\theta_0 + \theta_1 x + \theta_2 x^2$
Low Bias, Low Variance
(Goodfitting)
Price
Size
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
High Variance
(Overfitting)

	Under-fitting	Optimal-fitting	Over-fitting
Regression			
Classification			
Deep learning			



Bias
El sesgo es un peso desproporcionado a favor o en contra de una cosa

Varianza
cuantifica la **dispersión** de un conjunto de valores en relación con su media



- **High bias + Low variance → Underfitting**
  Modelo simple que no captura el patrón → errores altos en train y test.

- **Low bias + High variance → Overfitting**
  Modelo muy complejo, memoriza el train pero falla en test.

- **Low bias + Low variance → Ideal**
  Buen balance: generaliza bien.

- **High bias + High variance → Peor escenario**
  Ni aprende ni generaliza.

Cómo Evitar Underfitting y Overfitting

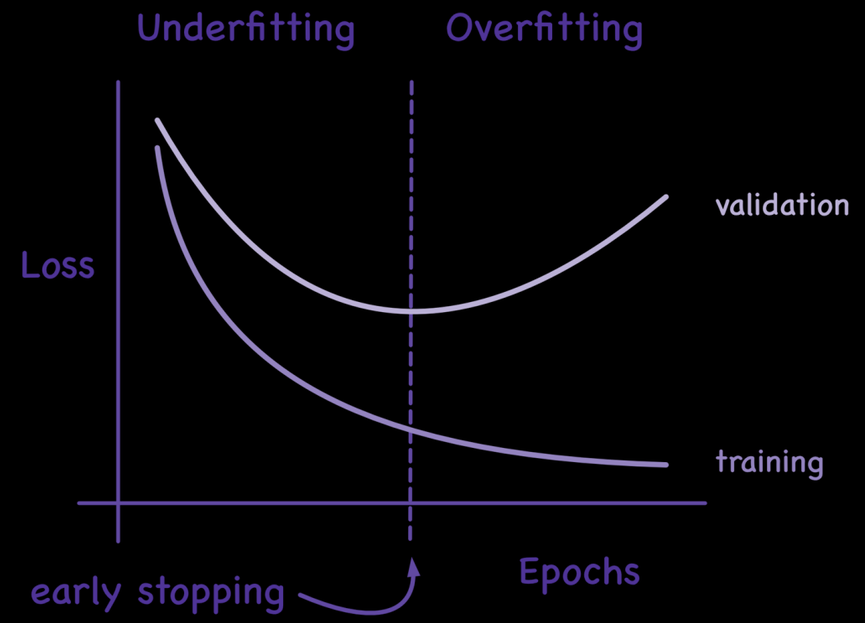
Estrategias Anti-Underfitting

- Aumentar complejidad del modelo
- Agregar características relevantes
- Reducir regularización
- Entrenar por más tiempo

Estrategias Anti-Overfitting

- Aplicar regularización (L1, L2)
- Obtener más datos de entrenamiento
- Usar validación cruzada
- Implementar dropout o early stopping

El equilibrio perfecto requiere ajustar hiperparámetros cuidadosamente y validar el rendimiento en datos no vistos.



REGRESION

Métrica / Error	Qué mide	Interpretación	Buena medida (ejemplo)	Mala medida (ejemplo)
MAE	Error medio absoluto	Promedio del error en las mismas unidades que la variable	MAE = 2 → en promedio el modelo se equivoca solo 2 unidades	MAE = 50 → el modelo se equivoca 50 unidades en promedio
MSE	Error cuadrático medio	Similar al MAE, pero penaliza más los errores grandes	MSE = 4 → errores pequeños y consistentes	MSE = 2500 → errores muy grandes frecuentes
RMSE	Raíz cuadrada del MSE	Error promedio en las mismas unidades, más interpretable que el MSE	RMSE = 3 → se equivoca en ±3 unidades aprox.	RMSE = 40 → errores muy alejados de lo real
R^2	Proporción de varianza explicada	Qué tan bien el modelo explica la variabilidad de la variable dependiente	R^2 = 0.90 → explica el 90% de la variación de los datos	R^2 = 0.20 → explica solo el 20%, casi azar
MAPE	Error porcentual medio	Promedio del error en porcentaje respecto al valor real	MAPE = 5% → el modelo falla solo un 5%	MAPE = 60% → falla más de la mitad del valor real

CLASIFICACION

Métrica / Error	Qué mide	Interpretación	Buena medida (ejemplo)	Mala medida (ejemplo)
Accuracy	Exactitud global	Porcentaje de predicciones correctas	0.95 → el 95% de predicciones son correctas	0.50 → como lanzar una moneda
Precision	Correctos entre los predichos como positivos	De los que dije “positivo”, cuántos realmente lo son	0.92 → casi todos los positivos predichos son reales	0.40 → más de la mitad son falsos
Recall (Sensibilidad)	Cobertura de positivos reales	De todos los positivos reales, cuántos detecta el modelo	0.90 → detecta el 90% de los positivos	0.30 → se le escapan 70% de positivos
F1-Score	Balance entre precision y recall	Útil si hay clases desbalanceadas	0.88 → buen equilibrio entre detectar y acertar	0.45 → bajo, falla mucho en balance
Matriz de confusión	Distribución de aciertos y errores	Permite ver TP, FP, FN, TN	TP=95, FN=5, FP=10, TN=90 → muy pocos errores	TP=30, FN=70, FP=50, TN=50 → muchos errores
AUC-ROC	Capacidad de separar clases	Mide qué tan bien distingue positivos y negativos	0.95 → separa muy bien las clases	0.55 → apenas mejor que adivinar
Log Loss	Penaliza predicciones poco seguras	Evalúa la calidad probabilística de las predicciones	0.15 → bajo, predicciones seguras y correctas	0.80 → alto, predicciones poco confiables

Validación Cruzada: ¿Qué es y por qué importa?

01

División Estratégica

Divide el conjunto de datos en k subconjuntos (folds) de tamaño similar, manteniendo la distribución de clases.

02

Entrenamiento Iterativo

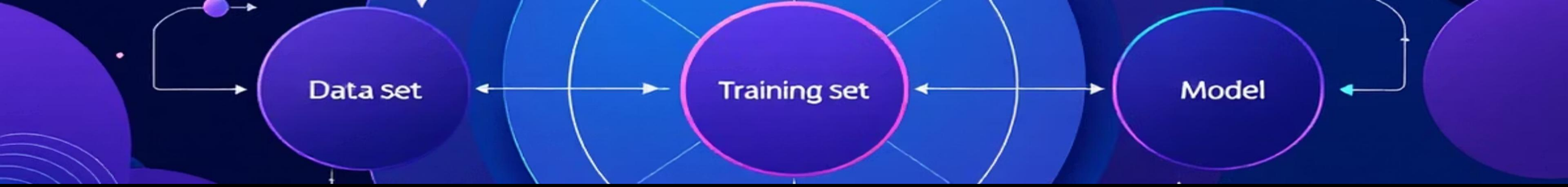
Entrena el modelo k veces, usando k-1 folds para entrenamiento y 1 fold para validación en cada iteración.

03

Evaluación Robusta

Promedia los resultados de todas las iteraciones para obtener una métrica de rendimiento más confiable y menos sesgada.

Especialmente valiosa para conjuntos de datos pequeños o medianos donde cada observación cuenta.



Visualización del Proceso de Validación Cruzada k-Fold

La validación cruzada k-Fold evalúa la calidad de un modelo dividiendo el dataset en *k* partes iguales y entrenándolo *k* veces: en cada iteración se usan *k-1* folds para entrenar y el fold restante para probar, rotando hasta que cada fold se use como test; al final, se promedian las métricas obtenidas (accuracy, MAE, R^2), logrando una estimación más robusta y confiable del rendimiento real del modelo.

Fold 1

Entrena con folds 2-5, valida con fold 1

Fold 3

Entrena con folds 1-2,4-5, valida con fold 3

1 — 2 — 3 — 4

Fold 2

Entrena con folds 1,3-5, valida con fold 2

Promedio

Combina resultados para métrica final

ⓘ Cada parte se usa para validar exactamente una vez, maximizando el uso de datos y la confiabilidad de la evaluación.

Hiperparámetros: ¿Qué Son y Por Qué Son Clave?

Definición

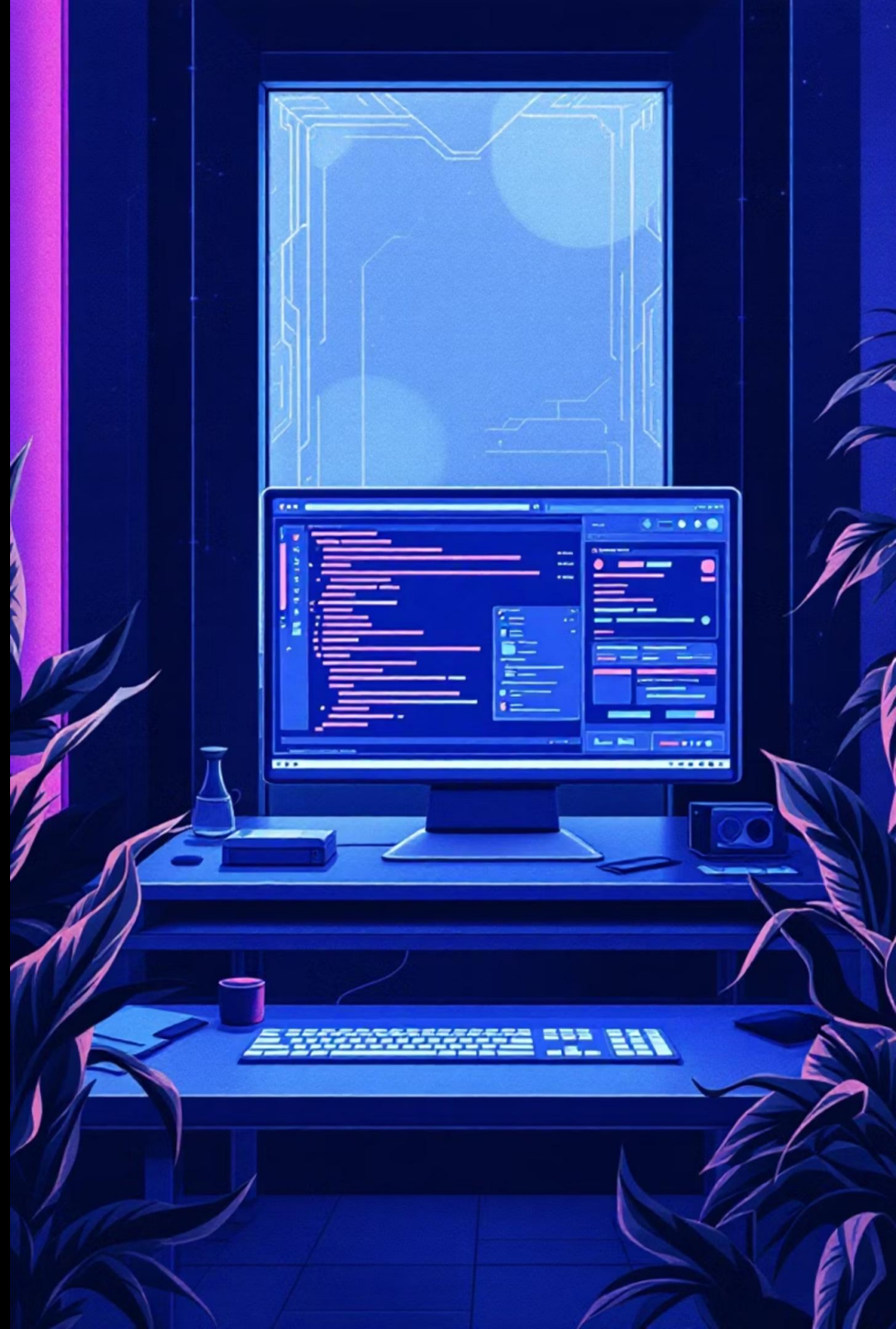
Parámetros externos que configuran el comportamiento del modelo antes del entrenamiento, no aprendidos de los datos.

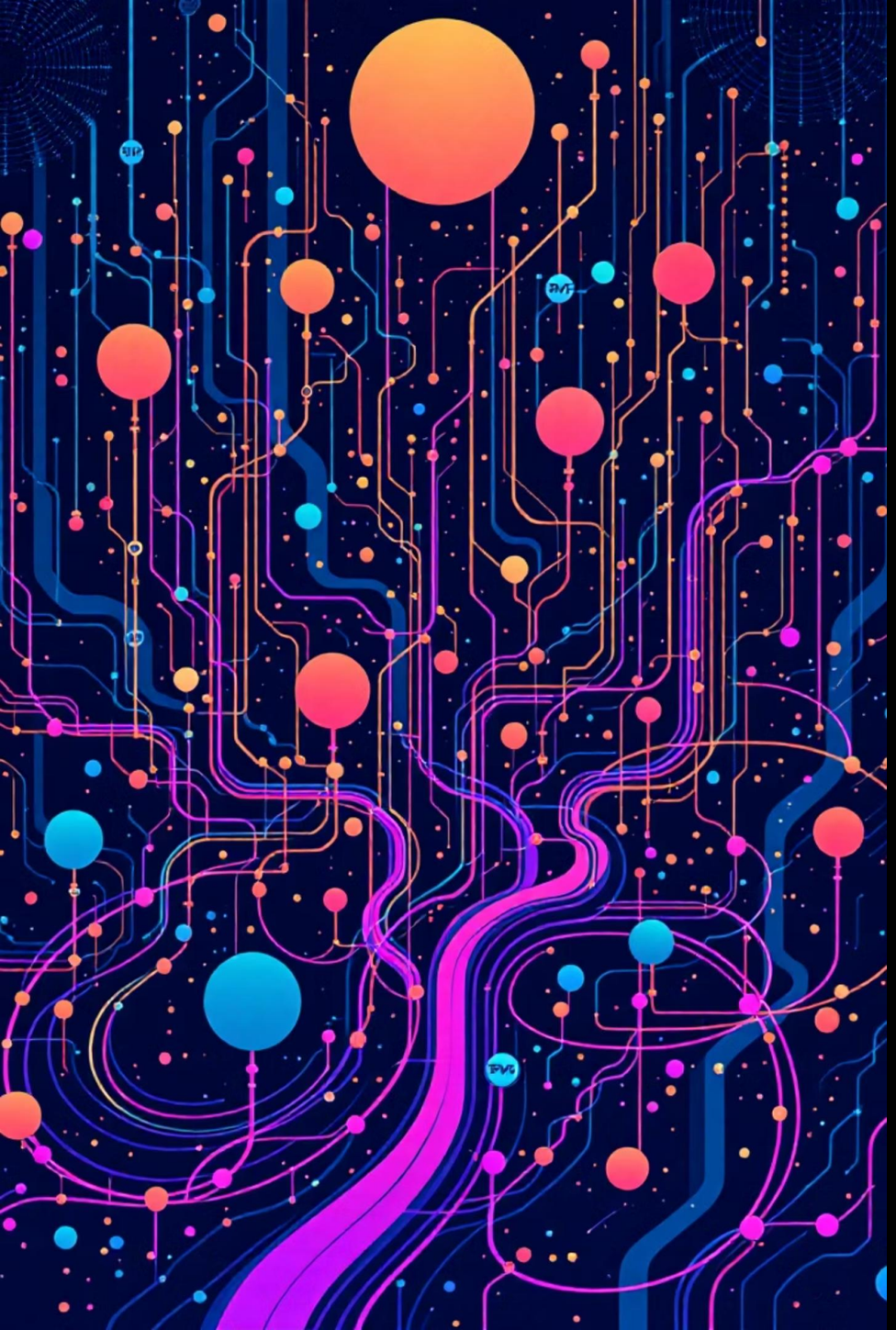
Ejemplos Comunes

Tasa de aprendizaje, profundidad de árbol, coeficiente de regularización, número de neuronas, kernel de SVM.

Impacto Directo

Determinan la precisión del modelo y su capacidad de generalización a datos nuevos no vistos durante el entrenamiento.





Métodos de Búsqueda y Optimización de Hiperparámetros Hiperparámetros

Búsqueda en Cuadrícula (Grid Search)

Prueba exhaustiva de todas las combinaciones predefinidas. Ideal para espacios pequeños de hiperparámetros.

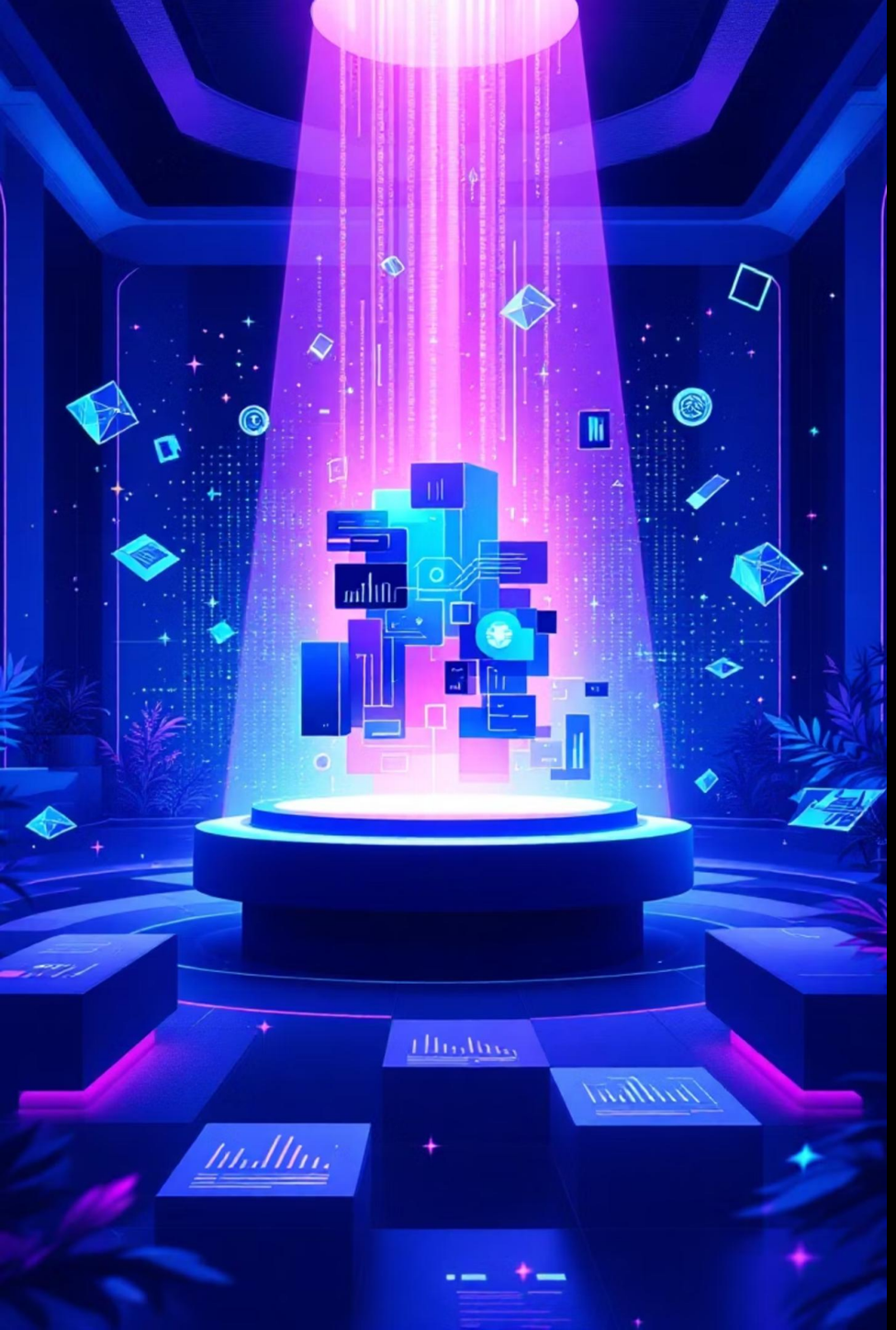
Búsqueda Aleatoria (Random Search)

Selecciona combinaciones aleatorias dentro de rangos definidos. Más eficiente en espacios de búsqueda grandes.

Reducir a la mitad (Halving Grid Search)

La idea principal es probar muchas combinaciones de parámetros, pero de manera progresiva, descartando las menos prometedoras en cada ronda

Método	Cómo funciona	Ventajas	Desventajas	Cuándo usarlo
GridSearchCV	Recorre todas las combinaciones de hiperparámetros	Encuentra el mejor dentro del grid definido Fácil de entender y explicar Bueno para clases/didáctica	Muy costoso si hay muchos parámetros o valores Escala mal con datasets grandes	- Pocos hiperparámetros y valores discretos - Dataset pequeño o de práctica
RandomizedSearchCV	Prueba combinaciones **aleatorias** de hiperparámetros dentro de distribuciones definidas	Mucho más rápido que GridSearch Explora rangos grandes Flexibilidad con distribuciones (randint, uniform)	No garantiza encontrar el mejor absoluto Puede necesitar muchas iteraciones para precisión	- Dataset mediano o grande - Cuando quieres probar muchos hiperparámetros con límite de tiempo
HalvingGridSearchCV (Successive Halving)	Empieza con muchas combinaciones, descarta las peores y refina las mejores	Más eficiente que GridSearch Usa recursos de forma inteligente Acelera búsquedas grandes	Más complejo de explicar No siempre encuentra el óptimo global	- Cuando quieres **balance entre exhaustividad y eficiencia** - Dataset medianos-grandes
Bayesian Optimization (Optuna, skopt, Hyperopt)	Aprende un modelo probabilístico de la función de score y elige inteligentemente los hiperparámetros	Explora de forma “inteligente” Encuentra buenos parámetros con menos evaluaciones Ideal para espacios de búsqueda grandes/continuos	Necesita librerías externas Más difícil de explicar en una clase básica	- Proyectos reales - Dataset grandes - Cuando buscas precisión con menos recursos



Conclusión: Claves para Modelos Robustos y Precisos

Comprende los Fundamentos

Dominar underfitting y overfitting es esencial para construir modelos que generalicen correctamente.

Valida Correctamente

La validación cruzada garantiza evaluaciones confiables y reduce el riesgo de conclusiones erróneas.

Optimiza Inteligentemente

El afinamiento de hiperparámetros desbloquea el verdadero potencial de tus modelos de machine learning.

¡Optimiza tu modelo para que aprenda bien y generalice mejor mejor!