

APRENDIZAJE AUTOMÁTICO I

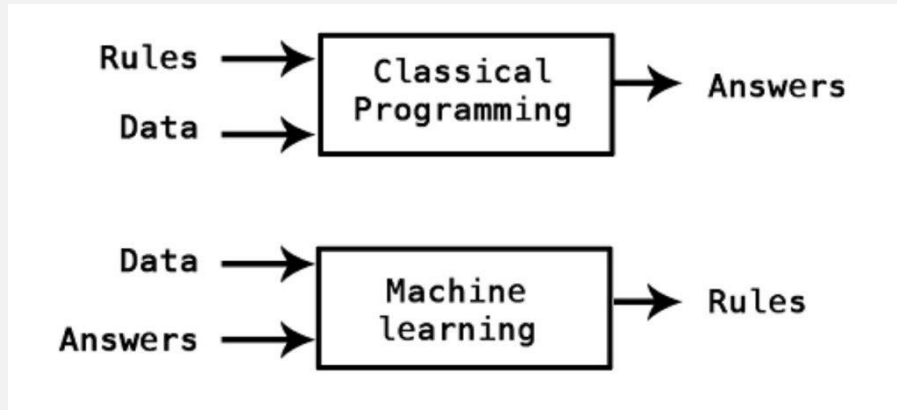
UDESА – Maestría en IA - 2025

MATIAS LEONI

INTRODUCCIÓN AL MACHINE LEARNING: CONCEPTOS FUNDAMENTALES

- ¡Bienvenidos al curso de Machine Learning!
- **Docente:** Matías Leoni
- **Agenda de Hoy:**
 - ¿Qué es el Machine Learning?
 - Problemas comunes: Clasificación y Regresión.
 - Cómo evaluamos nuestros modelos.
 - Los peligros del sobreajuste (Overfitting).
 - Taller práctico introductorio.
- **Objetivo de la clase:** Comprender el vocabulario y los conceptos básicos que nos acompañarán durante todo el curso.

¿QUÉ ES MACHINE LEARNING?



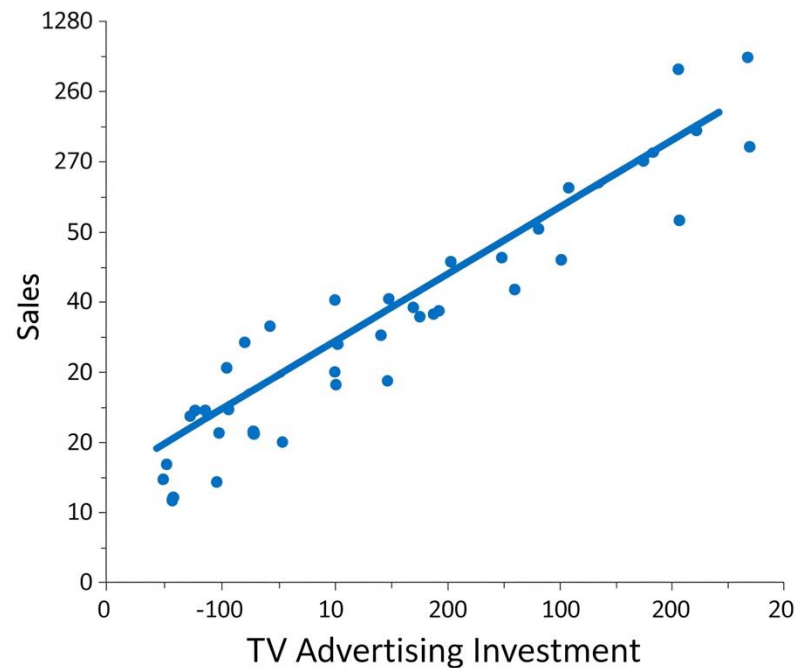
- **Definición formal:** El ML es un campo de la inteligencia artificial que utiliza técnicas estadísticas para dar a los sistemas informáticos la capacidad de "aprender" de los datos, sin ser explícitamente programados.
- **Diferencia clave con la programación tradicional:**
 - **Tradicional:** Reglas + Datos → Respuestas.
 - **Machine Learning:** Respuestas + Datos → Reglas.
- El objetivo es descubrir patrones en los datos para hacer predicciones sobre datos nuevos y nunca antes vistos.

TIPOS PRINCIPALES DE MACHINE LEARNING

- **Aprendizaje Supervisado (Supervised Learning):**
 - El más común.
 - Aprendemos de datos **etiquetados** (cada dato tiene una respuesta correcta).
 - **Ejemplo:** Predecir si un email es 'spam' o 'no spam' basándose en emails previamente clasificados.
- **Aprendizaje No Supervisado (Unsupervised Learning):**
 - Trabajamos con datos **no etiquetados**.
 - El objetivo es encontrar estructura o patrones ocultos.
 - **Ejemplo:** Agrupar clientes con comportamientos de compra similares (Clustering).
- **Aprendizaje Semi-Supervisado:**
 - Una mezcla de los dos anteriores. Se utiliza una gran cantidad de datos no etiquetados y una pequeña cantidad de datos etiquetados.
 - Útil cuando etiquetar datos es caro o consume mucho tiempo.

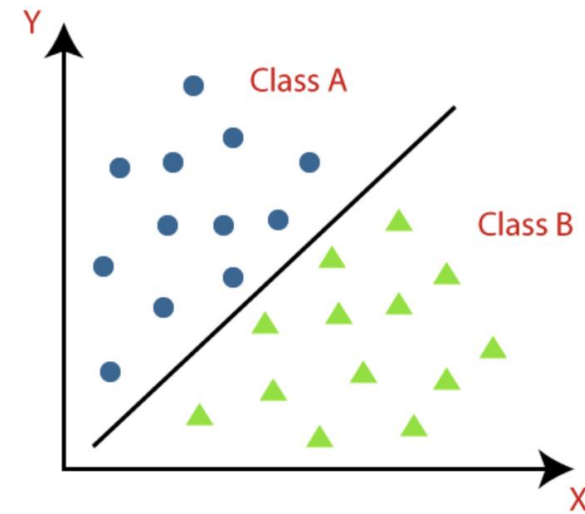
PROBLEMA SUPERVISADO: REGRESIÓN

- **Objetivo:** Predecir un valor **continuo** o numérico.
- La "respuesta" o variable objetivo (Y) es una cantidad.
- **Preguntas que responde:** "¿Cuánto?" o "¿Qué valor?".
- **Ejemplos Clave:**
 - Predecir el precio de una casa según sus características (tamaño, ubicación, etc.).
 - Estimar las ventas de un producto en función de la inversión en publicidad.
 - Pronosticar la temperatura de mañana.



PROBLEMAS MOTIVADORES: CLASIFICACIÓN

- **Objetivo:** Predecir una etiqueta **discreta** o una categoría.
- La variable objetivo pertenece a un conjunto finito de clases.
- **Preguntas que responde:** "¿De qué tipo es?" o "¿A qué grupo pertenece?".
- **Ejemplos Clave:**
 - Clasificar un correo como spam o no spam (Clasificación Binaria).
 - Diagnosticar si un tumor es benigno o maligno.
 - Identificar la raza de un perro en una foto (Husky, Golden Retriever, Poodle, etc.) (Clasificación Multiclase).



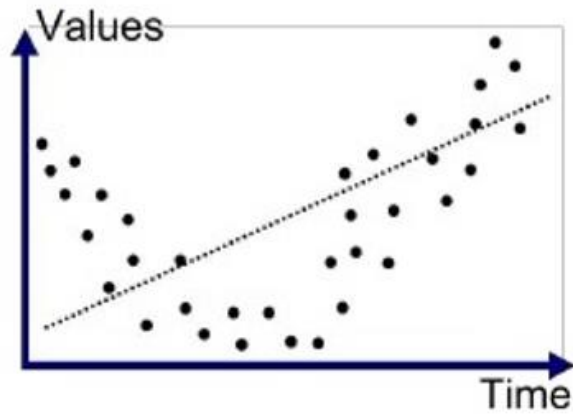
LA IMPORTANCIA DE LA GENERALIZACIÓN: TRAIN, VALIDATION Y TEST

- No nos interesa qué tan bien funciona el modelo en los datos que ya vio.
- El verdadero desafío es que el modelo funcione bien con **datos nuevos y nunca antes vistos**. Esto se llama **generalización**.
- Para simular este escenario, dividimos nuestros datos en (al menos) dos conjuntos:
 - **Conjunto de Entrenamiento (Training Set):** Los datos que usamos para "enseñarle" al modelo.
 - **Conjunto de Prueba (Test Set):** Datos que mantenemos ocultos. Los usamos **una sola vez** al final para obtener una evaluación honesta del rendimiento del modelo.
- **Analogía:** Es como estudiar para un examen. El *training set* son los ejercicios que resuelves para aprender. El *test set* es el examen final, con preguntas que nunca has visto antes.

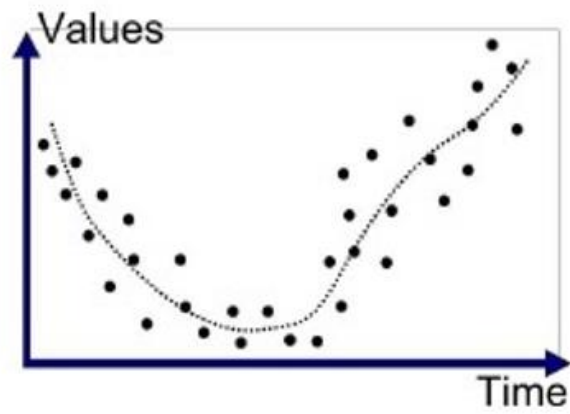
LOS PELIGROS DEL CAMINO: OVERFITTING Y UNDERFITTING

- **Underfitting (Subajuste):**
 - El modelo es **demasiado simple** y no captura la estructura de los datos.
 - Tendrá un mal rendimiento tanto en el entrenamiento como en la prueba.
 - **Causa:** Usar un modelo poco flexible para un problema complejo (ej. una línea recta para datos con forma de curva).
- **Overfitting (Sobreajuste):**
 - El modelo es **demasiado complejo**. Se "memoriza" los datos de entrenamiento, incluyendo el ruido.
 - Funciona perfectamente en el entrenamiento, pero **muy mal en la prueba**.
 - **Causa:** Usar un modelo muy flexible que se ajusta a las casualidades de los datos de entrenamiento.
- **El punto ideal (Good Fit):** Un modelo que captura la tendencia general de los datos y generaliza bien a nuevos datos.

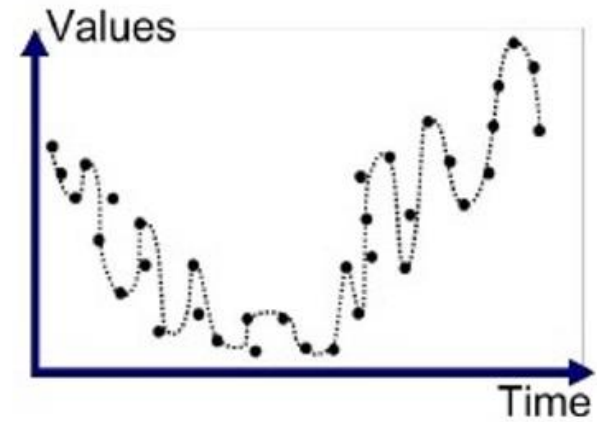
LOS PELIGROS DEL CAMINO: OVERFITTING Y UNDERFITTING



Underfitted



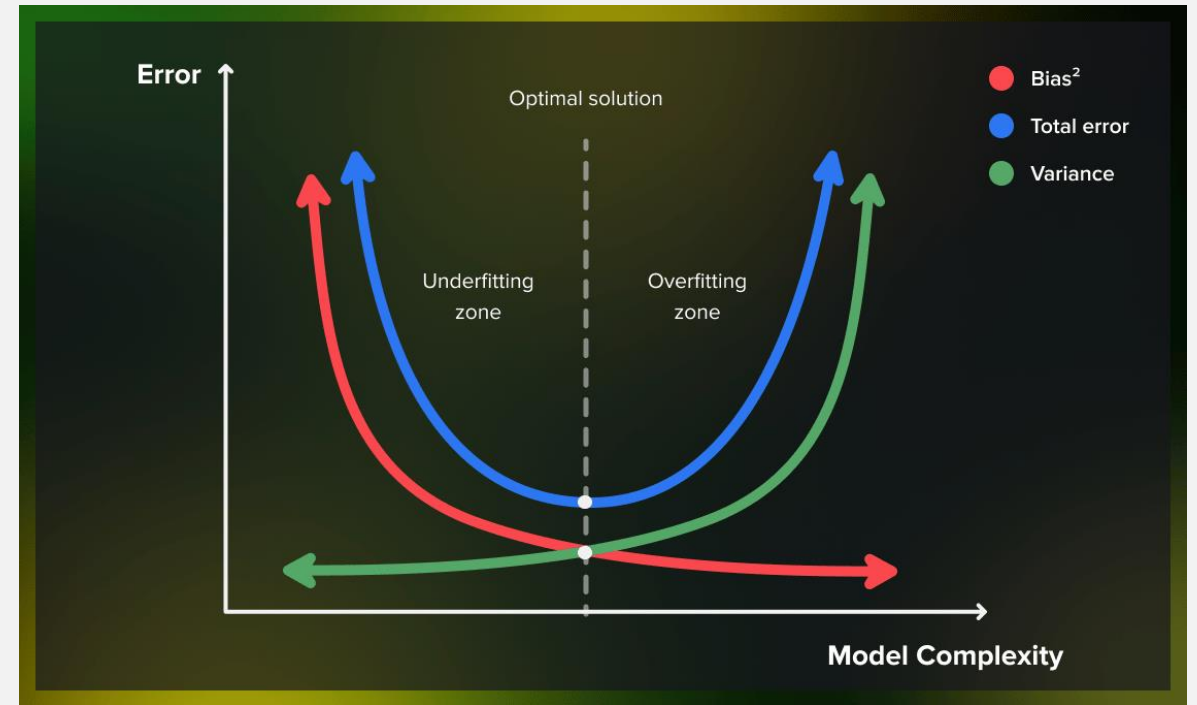
Good Fit/Robust



Overfitted

LA DESCOMPOSICIÓN DEL ERROR: SESGO VS. VARIANZA

- El error de predicción de un modelo se puede descomponer conceptualmente: $Error_{total} = (sesgo)^2 + Varianza + Error_{irreducible}$
- **Sesgo (Bias):** Error por suposiciones incorrectas del modelo. Un modelo de alto sesgo es demasiado simple (causa *underfitting*).
 - **Ejemplo:** Asumir que una relación compleja es lineal.
- **Varianza (Variance):** Sensibilidad del modelo a pequeñas fluctuaciones en los datos de entrenamiento. Un modelo de alta varianza es demasiado complejo (causa *overfitting*).
- **Trade-off Sesgo-Varianza:**
 - Aumentar la complejidad del modelo → Disminuye el sesgo, Aumenta la varianza.
 - Disminuir la complejidad del modelo → Aumenta el sesgo, Disminuye la varianza.
 - El objetivo es encontrar el balance óptimo.



¿CÓMO MEDIMOS EL ÉXITO? MÉTRICAS DE CLASIFICACIÓN

- Para saber si nuestro modelo es bueno, necesitamos métricas. La elección depende del objetivo del negocio.
- **Matriz de Confusión:** La base de todo. Es una tabla que visualiza el desempeño del clasificador.
- **Accuracy (Exactitud):** La métrica más simple.
Fórmula: $Accuracy = \frac{VP+VN}{VP+VN+FP+FN}$
- **Significado:** ¿Qué proporción de predicciones fueron correctas?
- **¡Cuidado!** Puede ser engañosa si las clases están desbalanceadas (ej. 99% de los emails no son spam. Un modelo que siempre dice "no spam" tiene 99% de accuracy, pero es inútil).

Clase Predicción	Positivo	Verdadero Positivo (VP)	Falso Positivo (FP)
	Negativo	Falso Negativo (FN)	Verdadero Negativo (VN)
		Positivo	Negativo
		Clase Real	

MÉTRICAS CLAVE: PRECISION Y RECALL

- **Precision (Precisión):**

- **Fórmula:** $Precision = \frac{VP}{VP+FP}$
- **Pregunta que responde:** De todos los que predijimos como "Positivo", ¿cuántos realmente lo eran?
- **Cuándo es importante:** Cuando el coste de un **Falso Positivo** es alto. (Ej: Marcar un email importante como spam).

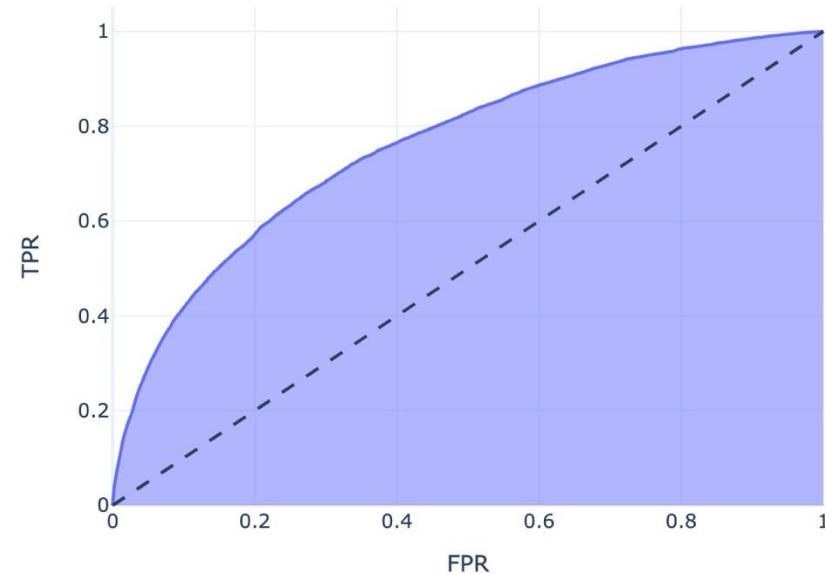
- **Recall (Sensibilidad o Cobertura):**

- **Fórmula:** $Recall = \frac{VP}{VP+FN}$
- **Pregunta que responde:** De todos los que eran "Positivo" en la realidad, ¿cuántos fuimos capaces de identificar?
- **Cuándo es importante:** Cuando el coste de un **Falso Negativo** es alto. (Ej: No detectar una enfermedad grave).

BALANCEANDO PRECISION Y RECALL: F1-SCORE Y AUC-ROC

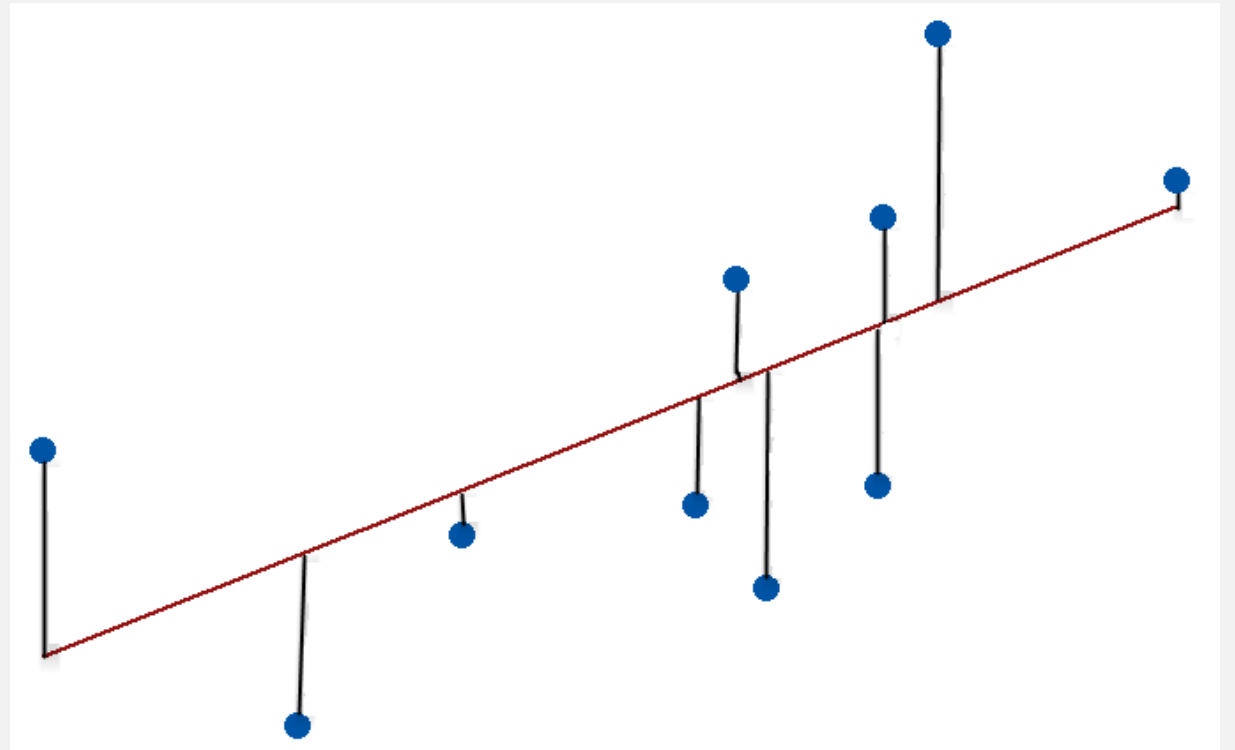
- **F1-Score:** La media armónica de Precision y Recall.
 - **Fórmula:** $F_1 = 2 \frac{Precision \times Recall}{Precision + Recall}$
 - **Significado:** Una única métrica que busca un balance entre Precision y Recall. Es alta solo si ambas son altas. Útil para clases desbalanceadas.
- **Curva ROC (Receiver Operating Characteristic):**
 - Es un gráfico que muestra el rendimiento de un clasificador para todos los umbrales de clasificación.
 - Eje Y: Tasa de Verdaderos Positivos (Recall).
 - Eje X: Tasa de Falsos Positivos ($FP / (FP + VN)$).
- **AUC (Area Under the Curve):**
 - El área bajo la curva ROC: 0.5 (azar) y 1.0 (perfecto).
 - Mide la capacidad del modelo para distinguir entre clases.

ROC Curve (AUC: 0.761, Gini: 0.522)



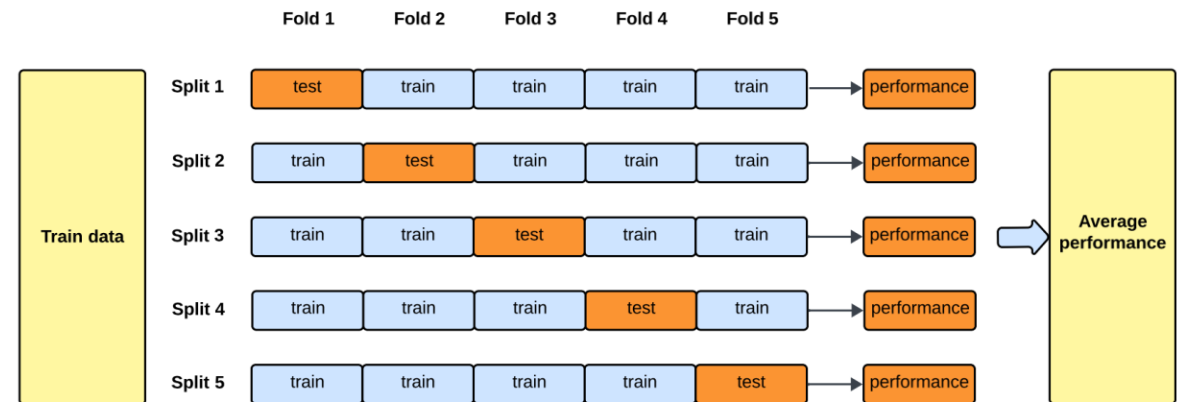
¿CÓMO MEDIMOS EL ÉXITO EN REGRESIÓN?

- En los problemas de regresión, necesitamos una forma de cuantificar qué tan cerca están nuestras predicciones (\hat{y}) de los valores reales y numéricos (y).
- La métrica más utilizada es el **Error Cuadrático Medio (Mean Squared Error - MSE)**. Su fórmula es:
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$$
 donde $\hat{f}(x_i)$ es la predicción que nuestro modelo genera para la i -ésima observación.
- **Intuición del MSE:**
 - Calcula la diferencia entre el valor real y el predicho para cada observación.
 - Eleva estas diferencias al cuadrado (lo que penaliza más los errores grandes y evita que los errores positivos y negativos se cancelen).
 - Calcula el promedio de estos errores al cuadrado. Un MSE más bajo es mejor, indicando que el modelo se ajusta bien a los datos.
- Es crucial diferenciar entre el **MSE de training** (calculado sobre los datos usados para entrenar) y el **MSE de testing** (calculado sobre datos nuevos). Nuestro objetivo final es siempre seleccionar el modelo que ofrezca el **MSE de testing más bajo**, ya que esto indica una mejor capacidad de generalización.



UNA EVALUACIÓN MÁS ROBUSTA: VALIDACIÓN CRUZADA

- **Problema:** Si dividimos los datos solo una vez en *train/test*, el resultado puede depender mucho de la "suerte" de esa división particular.
- **Solución: K-Fold Cross-Validation.** Una técnica para evaluar la estabilidad del modelo.
- **¿Cómo funciona?**
 - Dividimos el conjunto de entrenamiento en K "pliegues" (folds) de igual tamaño (ej. K=5 o K=10).
 - Realizamos K iteraciones. En cada una:
 - Usamos K-1 folds para entrenar el modelo.
 - Usamos el fold restante para evaluar (como un mini-test set).
 - El rendimiento final es el promedio de las métricas obtenidas en las K iteraciones.
- **Ventajas:** Proporciona una estimación del rendimiento mucho más fiable y nos da una idea de la variabilidad (estabilidad) del modelo.

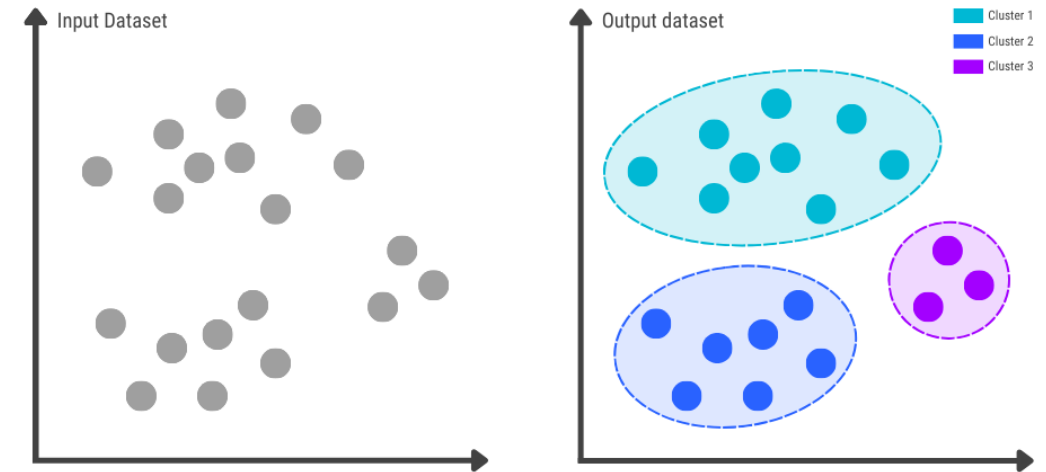


EVITANDO EL EXCESO DE CONFIANZA: REGULARIZACIÓN

- **Concepto:** Es una técnica para prevenir el *overfitting* al penalizar la complejidad del modelo.
- **Intuición:** Si un modelo tiene coeficientes (o "pesos") muy grandes, es probable que esté sobreajustando. La regularización añade un término a la función de coste del modelo para mantener estos coeficientes pequeños.
$$Costo_{nuevo} = Costo_{viejo} + Penalización$$
- **L2 Regularization (Ridge):**
 - Penaliza la **suma de los cuadrados** de los coeficientes.
 - Fórmula de penalización: $\lambda \sum_{j=1}^p \beta_j^2$
 - Efecto: Encoge los coeficientes hacia cero, pero raramente los hace exactamente cero.
- **L1 Regularization (Lasso):**
 - Penaliza la **suma de los valores absolutos** de los coeficientes.
 - Fórmula de penalización: $\lambda \sum_{j=1}^p |\beta_j|$
 - Efecto: Puede encoger algunos coeficientes hasta que sean **exactamente cero**, realizando así una selección automática de características.

¿Y SI NO HAY ETIQUETAS? INTRO A NO SUPERVISADO

- Recordemos: el aprendizaje no supervisado trabaja con datos sin una variable de respuesta (Y) definida.
- El objetivo es descubrir la estructura o patrones inherentes en los datos.
- **Clustering (Agrupamiento):** La tarea no supervisada más común.
- **Objetivo del Clustering:** Agrupar las observaciones de modo que los miembros de un mismo grupo (cluster) sean muy similares entre sí, y muy diferentes a los miembros de otros grupos.



RESUMEN Y ¿QUÉ SIGUE?

- La diferencia entre ML Supervisado (Regresión/Clasificación) y No Supervisado.
- La importancia de la generalización y la división en *train/test*.
- El dilema central: *Overfitting* vs *Underfitting* y el trade-off Sesgo-Varianza.
- Cómo evaluar modelos de clasificación (Matriz de Confusión, Accuracy, Precision, Recall, F1, AUC).
- Técnicas para modelos más robustos: Validación Cruzada y Regularización.
- **En la próxima clase:** Nos sumergiremos en nuestro primer modelo: la Regresión Lineal, desde sus fundamentos hasta su implementación. **¡Ahora, a la práctica!**

TALLER PRÁCTICO

- **Objetivo:** Familiarizarnos con el entorno de trabajo (Jupyter Notebook) y aplicar los conceptos de carga y exploración de datos.
- **Herramientas:** Python, Jupyter, Pandas, NumPy, Scikit-learn, Matplotlib.
- **Actividades:**
 - Cargar un dataset clásico (Iris).
 - Explorar sus características: head(), describe(), info().
 - Realizar una visualización simple para entender la relación entre variables.
 - Dividir los datos en conjuntos de entrenamiento y prueba usando train-test split de scikit-learn.
- **Abran el Jupyter Notebook Clase01-Intro-ML.ipynb y sigan las instrucciones.**