Curso base

INTELIGENCIA ARTIFICIAL DESDE CERO

LECCIÓN 17
Generalización con train/test

Cómo separar entrenamiento y prueba para comprobar si el modelo aprende patrones y no solo memoriza ejemplos

Lección	17	**Tema**	Generalización con train/test
Nombre del participante	_____	**Fecha**	_____

Actividad central

Actividad principal: mini-proyecto de regresión con partición `train/test`, cálculo de una métrica sencilla de error en prueba e interpretación de generalización.
Producto esperado: notebook corto con dos ejemplos guiados, separación de entrenamiento y prueba, y una conclusión clara sobre si el modelo parece generalizar o si existe riesgo de memorizar demasiado.

Meta de la lección. Comprender por qué un modelo debe evaluarse con datos distintos a los que usó para aprender, aplicar una partición `train/test` en un notebook sencillo y desarrollar criterio para decidir si una regresión realmente generaliza o si solo memoriza ejemplos.

Idea central

Un modelo puede verse excelente cuando se evalúa con los mismos datos con los que fue entrenado. Sin embargo, eso no prueba que sea útil en la práctica. La pregunta importante es otra: **¿qué ocurre cuando el modelo se enfrenta a datos nuevos?**

En esta lección aprenderemos que separar entrenamiento y prueba permite evaluar con más honestidad si el modelo aprendió un patrón general o si solo se ajustó demasiado a los ejemplos iniciales.

¿Qué vamos a hacer en esta guía?

- Entender, con lenguaje sencillo, para qué sirve dividir los datos en entrenamiento y prueba.
- Reconocer un vocabulario mínimo: `train`, `test`, generalización, memorizar y `random_state`.
- Desarrollar un primer ejemplo guiado con publicidad y ventas.
- Repetir la idea en un segundo ejemplo con área de vivienda y precio.
- Interpretar resultados con apoyo de prompts de IA generativa, sin reemplazar el razonamiento propio.

¿Por qué dividir los datos?

El conjunto de entrenamiento sirve para que el modelo aprenda. El conjunto de prueba se guarda aparte y actúa como un examen final. Si el modelo funciona bien también en la prueba, ganamos confianza en que la relación aprendida podría servir con datos nuevos.

Si no hacemos esta separación, corremos el riesgo de sobreestimar el modelo: podríamos creer que es bueno solo porque se le preguntó sobre ejemplos que ya conocía.

Situación	¿Qué nos enseña la separación?
Ventas	Permite saber si el modelo predice bien días nuevos y no solo los registros históricos usados para entrenarlo.
Precio de vivienda	Ayuda a comprobar si la relación aprendida funciona en inmuebles que el modelo no vio durante el ajuste.
Consumo de energía	Sirve para evaluar si el modelo conserva utilidad cuando llegan registros nuevos.

Vocabulario mínimo para comenzar

Término	Explicación sencilla
`train`	Conjunto de entrenamiento. Son los datos con los que el modelo aprende.
`test`	Conjunto de prueba. Son datos reservados para evaluar al final.
Generalización	Capacidad del modelo para comportarse razonablemente bien con datos nuevos.
Memorizar	Ocurre cuando el modelo se ajusta demasiado a ejemplos conocidos y luego falla fuera de ellos.
`random_state`	Número que permite repetir la misma partición de manera ordenada y comparable.

Recomendación pedagógica antes de empezar

En esta lección no buscamos fórmulas complicadas. Lo importante es que el estudiante pueda responder preguntas como estas: *¿con cuáles datos aprende el modelo?, ¿cuáles datos se guardan para probar?, ¿por qué esa separación vuelve más honesta la evaluación?, ¿qué señal me hace sospechar que el modelo memoriza demasiado?*

Ejemplo guiado 1: predecir ventas a partir de la inversión en publicidad

Vamos a usar un caso sencillo. Imaginemos una tabla donde la entrada es la inversión en publicidad y la salida son las ventas. Nuestro objetivo no será solo entrenar una regresión, sino comprobar si mantiene un error razonable cuando se le pide predecir casos que no vio durante el aprendizaje.

Caso	Publicidad	Ventas	Lectura rápida
1	1	3	Inicio de una tendencia creciente.
2	2	4	Al aumentar la publicidad, aumentan las ventas.
3	3	5	El patrón sigue siendo estable.
4	4	6	La relación parece casi lineal.
5	5	7	Se mantiene la misma lógica.
6 a 10	...	...	Continúa una tendencia creciente similar.

Paso a paso del ejemplo 1

Paso 1. Crear o cargar la tabla de datos. Lo primero es tener la información organizada en un `DataFrame`.

Paso 2. Separar la variable de entrada y la variable objetivo. La entrada será `publicidad`. La salida será `ventas`.

Paso 3. Dividir los datos en entrenamiento y prueba. Aquí ocurre la idea central de esta guía. Una parte se usa para aprender y otra parte se guarda para evaluar.

Código base para partición `train/test`

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error

datos = pd.DataFrame({
    "publicidad": [1,2,3,4,5,6,7,8,9,10],
    "ventas": [3,4,5,6,7,8,9,10,11,12]
})

X = datos[["publicidad"]]
y = datos["ventas"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Entrenar y evaluar

Paso 4. Crear y entrenar el modelo. El modelo solo debe ver los datos de entrenamiento. Esa es una regla importante.

```
modelo = LinearRegression()
modelo.fit(X_train, y_train)
```

Paso 5. Predecir sobre el conjunto de prueba. Ahora usamos los datos que el modelo no vio durante el entrenamiento. Este es el examen real.

```
pred = modelo.predict(X_test)
mae = mean_absolute_error(y_test, pred)

print("Filas train:", len(X_train))
print("Filas test:", len(X_test))
print("MAE en prueba:", round(mae, 2))
```

Interpretación del resultado

Si el error en prueba es pequeño y coherente con la escala del problema, podemos decir que el modelo parece generalizar razonablemente bien. Si en entrenamiento todo luce perfecto, pero en prueba el error se dispara, debemos sospechar que el modelo aprendió demasiado pegado a los ejemplos iniciales.

Prompts de IA generativa para acompañar el ejemplo 1

- Explícame con palabras sencillas por qué un modelo no debe evaluarse con los mismos datos con los que se entrenó.
- Tengo un MAE en prueba de cierto valor. Ayúdame a redactar una interpretación prudente y clara para un principiante.
- Revisa este código con `train_test_split` y dime si está correcto. Si hay errores, corrígelos paso a paso.
- Explícame la diferencia entre aprender un patrón y memorizar ejemplos usando el contexto de publicidad y ventas.

Comentario pedagógico

Antes de mirar el resultado del notebook, conviene que el estudiante haga una predicción verbal. Por ejemplo: *si la publicidad aumenta, probablemente las ventas también aumenten.* Esta anticipación ayuda a que el código tenga sentido y evita que el aprendizaje se reduzca a copiar y pegar instrucciones.

Ejemplo guiado 2: predecir precio de vivienda según área con separación `train/test`

Ahora repetiremos la lógica en otro contexto. El objetivo es que el estudiante descubra que `train/test` no depende del tema del problema: es una práctica general de evaluación honesta.

Área (m^2)	Precio (millones)	Lectura didáctica
40	120	Vivienda pequeña con precio menor.
55	155	A mayor área, mayor precio estimado.
70	190	La relación sigue creciendo.
85	230	El patrón conserva coherencia.
100	265	La tendencia continúa en ascenso.

Paso a paso del ejemplo 2

Paso 1. Crear la tabla. Usaremos una tabla pequeña para concentrarnos en el proceso.

Paso 2. Separar X e y. `X` será el área y `y` será el precio.

```
datos_vivienda = pd.DataFrame({
    "area_m2": [40, 55, 70, 85, 100, 115, 130, 145],
    "precio_millones": [120, 155, 190, 230, 265, 300, 340, 380]
})

X2 = datos_vivienda[["area_m2"]]
y2 = datos_vivienda["precio_millones"]
```

Paso 3. Dividir entrenamiento y prueba. De nuevo usamos `train_test_split` para reservar algunos casos para el final.

```
X2_train, X2_test, y2_train, y2_test = train_test_split(
    X2, y2, test_size=0.25, random_state=42
)
```

Paso 4. Entrenar y evaluar.

```
modelo2 = LinearRegression()
modelo2.fit(X2_train, y2_train)

pred2 = modelo2.predict(X2_test)
mae2 = mean_absolute_error(y2_test, pred2)

print("MAE en prueba:", round(mae2, 2))
```

Comparar ambos ejemplos

En el primer ejemplo cambiaba la publicidad y observábamos ventas. En el segundo cambia el área y observamos precio. Aunque los contextos son distintos, la idea de generalización es la misma: entrenar con una parte, probar con otra y leer el error con honestidad.

Prompts de IA generativa para acompañar el ejemplo 2

- Compara en lenguaje sencillo estos dos ejemplos de `train/test` y explica qué tienen en común.
- Explícame qué significa que un error en prueba sea parecido al error en entrenamiento.
- Dame una explicación corta sobre el riesgo de sobreajuste para un estudiante principiante.
- Ayúdame a redactar una conclusión final sobre si este modelo parece generalizar.

Buenas prácticas para usar `train/test`

Recomendación	¿Por qué importa?
Separar antes de evaluar	Evita medir el desempeño con los mismos datos usados para aprender.
Fijar `random_state`	Permite repetir la partición y comparar resultados con orden.
No mezclar `train` y `test`	Si el modelo ve información de prueba durante el ajuste, la evaluación pierde credibilidad.
Leer el error en contexto	Una métrica aislada no basta; hay que juzgar su utilidad real.
Explicar antes de ejecutar	Obliga a pensar qué se espera del modelo y qué significan los resultados.

Mini ruta de trabajo sugerida para el participante

- Observe la tabla y describa verbalmente la tendencia general.
- Identifique con claridad cuál es la entrada y cuál es la salida.
- Haga la partición `train/test` y confirme cuántas filas quedaron en cada conjunto.
- Entrene el modelo solo con `train`.
- Evalúe con `test` y escriba una conclusión prudente sobre generalización.

Producto esperado

Un notebook corto y ordenado que muestre dos ejemplos básicos de regresión con partición `train/test`, una métrica sencilla de error en prueba y una conclusión redactada en lenguaje claro sobre si el modelo parece generalizar o si todavía existe riesgo de memorizar demasiado.

Lista de verificación

Revisión	Sí / No
☐	¿Se identificaron correctamente la variable de entrada y la variable objetivo?
☐	¿Se separaron los datos en entrenamiento y prueba?
☐	¿Se entrenó el modelo solo con el conjunto de entrenamiento?
☐	¿Se evaluó el modelo con datos de prueba?
☐	¿Se redactó una conclusión sobre generalización?
☐	¿Se usó al menos un prompt de IA generativa para comprender o mejorar el trabajo?

Cierre

Separar entrenamiento y prueba es una práctica sencilla, pero cambia por completo la calidad del análisis. Nos obliga a evaluar con honestidad y a distinguir entre un modelo que impresiona en pantalla y uno que realmente puede servir en casos nuevos. Cuando un estudiante comprende esta idea, ya está entrando al aprendizaje automático con una base mucho más sólida y crítica.