



Curso de especialización inteligencia artificial y big data

Tema 5: Evaluación y selección de modelos

Profesor: Sebastián Rubio Valero

Septiembre 2025



Sistema aprendizaje automático

1 Objetivos de la sesión

- Comprender la necesidad de la validación cruzada.
- Diferenciar los tipos de validación cruzada.
- Analizar el impacto del sesgo y la varianza.
- Implementar ejemplos en Scikit-learn.
- Reconocer implicaciones computacionales en entornos distribuidos.

2 Motivación y contexto

El objetivo en aprendizaje automático es construir modelos que generalicen bien. Un modelo puede sufrir:

- **Sobreajuste (overfitting)**: aprende ruido o detalles específicos.
- **Subajuste (underfitting)**: no captura los patrones esenciales.

Tradicionalmente se usan tres subconjuntos: entrenamiento, validación y prueba. Sin embargo, con pocos datos o distribuciones inestables, esta división no es suficiente. La validación cruzada (*cross-validation*) ofrece una forma más robusta de estimar el error de generalización. Existen diferentes técnicas, unas orientadas a los modelos supervisados y otras a los no supervisados (con objetivos diferentes). Los vamos a utilizar, la motivación es doble:

- Durante la selección del modelo. Para comparar el rendimiento de diferentes algoritmos
- Durante el ajuste de hiperparámetros (parámetros que no se aprenden directamente de los datos)

En un primer análisis, medimos la precisión (accuracy) con los datos de entrenamiento, y luego el rendimiento con los datos de test. Si el rendimiento obtenido en el entrenamiento es mucho mayor que el rendimiento en los datos de test (75% vs 40%), entonces es muy probable que el modelo este sobreajustado. Por otro lado si el rendimiento es bajo (40% vs 45%) tanto en el entrenamiento como en las pruebas es muy probable que el modelo esté subajustado. Lo ideal es que ambas sean altas y con valores muy cercanos p.e 93% en el entrenamiento y el 90% en las pruebas. Es decir las curvas de aprendizaje y prueba convergen a un valor de error bajo y estable. O bien la brecha entre estas dos curvas es pequeño (Esto ya lo estudiamos al principio) Los pasos a seguir serían:

- Divide tus datos: Es tu mejor amigo. Guarda el conjunto de prueba y no lo toques hasta el final.
- Entrena tu modelo: Usa solo el conjunto de entrenamiento.
- Evalúa y Grafica: Calcula la métrica elegida (accuracy, F1-score, pérdida, etc.) tanto para el entrenamiento como para la validación (o prueba). Gráfica las curvas de aprendizaje. Esto es crucial.
- Analiza las Curvas:
 - ¿Las dos curvas están juntas pero en un error alto? -¿ Underfitting.
 - ¿Hay una gran brecha entre ellas? -¿ Overfitting.
 - ¿Están juntas y en un error bajo? -¿ ¡Felicidades!

2.1 Soluciones al overfitting

- Recolectar más datos: La solución más efectiva, pero no siempre posible.
- Regularización (L1 - Lasso, L2 - Ridge): Añade una penalización a los pesos del modelo para evitar que se vuelvan demasiado grandes y complejos
- Reducir la complejidad del modelo: Usar menos capas en una red neuronal, menos profundidad en un árbol de decisión, etc.
- Dropout (para Redes Neuronales): "Apagar" neuronas aleatoriamente durante el entrenamiento para evitar dependencias co-adaptadas.
- Early Stopping (Parada Temprana): Detener el entrenamiento cuando el rendimiento en validación deje de mejorar.

2.2 Soluciones al underfitting

- Aumentar la complejidad del modelo: Añadir más capas o neuronas (en redes neuronales), aumentar la profundidad de los árboles, usar modelos más potentes.
- Entrenar por más tiempo (más épocas): A veces el modelo simplemente no ha convergido.
- Reducir la regularización: Disminuir los parámetros de penalización.
- Mejorar las características (Feature Engineering): Crear características más informativas o usar más características relevantes.

3 ¿Qué es la validación cruzada?

La validación cruzada (cross-validation) es una técnica estadística utilizada para evaluar el rendimiento y la capacidad de generalización de los modelos de aprendizaje automático. Pertenecce a la fase de procesamiento de modelado y evaluación de modelos predictivos

4 Fundamento teórico

La validación cruzada divide aleatoriamente el conjunto de datos en K subconjuntos de igual tamaño (*folds*) y repite el entrenamiento K veces, el valor de k va . El error medio estimado es:

$$E_{CV} = \frac{1}{K} \sum_{i=1}^K E_i$$

donde E_i es el error obtenido al validar sobre el i -ésimo fold.

Este estimador tiene menor varianza que una única partición train/test, aunque aumenta ligeramente el sesgo.

5 Tipos de validación cruzada

- **K-Fold:** Divide los datos en K subconjuntos del mismo tamaño.
- **Stratified K-Fold:** Mantiene la proporción de clases.
- **Leave-One-Out (LOOCV):** Cada observación es un fold.
- **Shuffle Split:** Repite particiones aleatorias.
- **Time Series Split:** Respeta el orden temporal.
- **Nested Cross-Validation:** Combina validación interna y externa.

En modelos de regresión lineal, es fundamental evaluar el rendimiento mediante métricas cuantitativas. A continuación, se detallan las principales métricas, sus fórmulas, ventajas y desventajas.

6 Métricas Principales

Las métricas que a continuación se aplican en según cada caso

Métrica	Usar Cuando...	Evitar Cuando...
MSE	Penalizar errores grandes.	Hay muchos <i>outliers</i> .
RMSE	Error en unidades originales.	<i>Outliers</i> problemáticos.
MAE	Robustez frente a <i>outliers</i> .	Necesitas diferenciabilidad.
R²	Medir varianza explicada.	Hay sobreajuste.
R² Ajustado	Comparar modelos con más variables.	Modelos simples.
MAPE	Errores en porcentaje.	Valores cercanos a cero.

Table 1: Guía rápida para selección de métricas.

7 Relación con sesgo y varianza

El error total puede expresarse como:

$$E[(y - \hat{f}(x))^2] = \text{Sesgo}^2 + \text{Varianza} + \text{Ruido irreducible}$$

La validación cruzada busca equilibrar sesgo y varianza proporcionando una estimación estable del error esperado.

Técnicas de Validación: Las diferentes técnicas nos indica las diferentes formas en que debemos tomar los datos de entrenamiento y validación

Métricas de Evaluación: Diferentes formas de calcular el rendimiento tras realizar las pruebas de entrenamiento y validación

8 Tipos de validación cruzada

Existe una relación entre las técnicas de validación y las métricas

8.1 Validación cruzada básica

Es una división en k pliegues (folds). El conjunto de datos de entrenamiento (o el conjunto original si no se ha separado un conjunto de prueba) se divide aleatoriamente en k subconjuntos del mismo tamaño. Normalmente se escoge un k entre 5 y 10. El proceso de entrenamiento y evaluación se repite k veces (los pliegues). En cada iteración $k-1$ pliegues se utilizan para entrenar el modelo, el pliegue restante se utiliza para validar y calcular la métrica de rendimiento (ej. precision, ECM). Una vez completadas las k iteraciones se calcula el promedio de las métricas de rendimiento obtenido en cada iteración, este promedio es la estimación final, y se usa para comparar con otros modelos

Código python

```
from sklearn.model_selection import cross_val_score, KFold
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris

X, y = load_iris(return_X_y=True)
model = LogisticRegression(max_iter=1000)
cv = KFold(n_splits=5, shuffle=True, random_state=42)

scores = cross_val_score(model, X, y, cv=cv)
print("Accuracy medio:", scores.mean())
```

- n_splits (Número de Divisiones)? '¿Qué representa? Este es el parámetro más importante. Representa el número de divisiones en las que se divide el conjunto de datos para la validación cruzada.
- En cada una de las n_splits iteraciones, un grupo diferente se usa como conjunto de prueba (test set), y los $n-1$ grupos restantes se combinan para formar el conjunto de entrenamiento (train set). Al final, cada observación es parte de un conjunto de prueba y un conjunto de entrenamiento.
-

Ejemplo: Con $n_splits = 5$, tendrás 5 iteraciones. En cada iteración, se divide el conjunto de datos en 5 partes.

8.2 Nested Cross-Validation

La validación cruzada anidada (CV) se utiliza a menudo para entrenar un modelo en el que también hay que optimizar los hiperparámetros. La CV anidada estima el error de generalización del modelo subyacente y su búsqueda de (hiper)parámetros. La elección de los parámetros que maximizan la CV no anidada sesga el modelo hacia el conjunto de datos, lo que produce una puntuación demasiado optimista.

Código python

```
from sklearn.model_selection import GridSearchCV, cross_val_score
from sklearn.svm import SVC

param_grid = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']}
inner_cv = KFold(n_splits=3, shuffle=True, random_state=42)
outer_cv = KFold(n_splits=5, shuffle=True, random_state=42)

model = GridSearchCV(SVC(), param_grid=param_grid, cv=inner_cv)
nested_scores = cross_val_score(model, X, y, cv=outer_cv)
print("Nested-CV-accuracy:", nested_scores.mean())
```

9 Validación cruzada en entornos distribuidos (Spark MLlib)

```
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.ml.evaluation import BinaryClassificationEvaluator

lr = LogisticRegression()
paramGrid = ParamGridBuilder().addGrid(lr.regParam, [0.1, 0.01]).build()
evaluator = BinaryClassificationEvaluator()

cv = CrossValidator(estimator=lr, estimatorParamMaps=paramGrid,
                    evaluator=evaluator, numFolds=5)
```

10 Interpretación de resultados

- La **media** de los resultados representa el rendimiento esperado.
- La **desviación estándar** indica estabilidad.
- La **nested cross-validation** evita fuga de datos entre ajuste y evaluación.

Ejemplo: un modelo con *accuracy* medio de 0.91 y desviación estándar de 0.02 sugiere un rendimiento estable.

11 Validación Cruzada en Aprendizaje No Supervisado

En el aprendizaje no supervisado (como clustering o reducción de dimensionalidad), el objetivo no es la predicción de una etiqueta conocida, sino descubrir patrones o estructuras en los datos. Esto complica la aplicación directa de la validación cruzada tradicional, pero aún así es una herramienta valiosa. Sus aplicaciones incluyen:

- Estimación de Hiperparámetros: La validación cruzada se puede utilizar para seleccionar el número óptimo de parámetros o componentes.
 - Clustering (Agrupamiento): Se puede utilizar para determinar el número óptimo de clusters (k) en algoritmos como K -Means. En lugar de métricas de error de predicción, se emplean métricas de validación interna como el coeficiente de la silueta (Silhouette Score), la cohesión o la separación. El modelo que maximice la métrica promedio en los folds se considera el mejor
 - Reducción de Dimensionalidad: En técnicas como el Análisis de Componentes Principales (PCA), la validación cruzada se puede usar para determinar el número de componentes principales a retener, basándose a menudo en el error de reconstrucción o el likelihood de un modelo subyacente de factor latente
- Evaluación de Estabilidad: En lugar de medir la precisión predictiva, se puede usar para evaluar la estabilidad de las estructuras descubiertas.

Estabilidad del Clustering: Al aplicar el algoritmo de clustering a diferentes subconjuntos de datos (los folds de entrenamiento) y comparar la similitud de las agrupaciones resultantes en el fold de prueba, podemos evaluar cuán robustas son las estructuras encontradas. Una mayor consistencia entre los resultados de los diferentes folds indica un clustering más estable y fiable.

12 Consideraciones especiales

12.1 Series temporales

Para los datos secuenciales donde el tiempo es un factor (pe.el precio de las acciones), se utiliza la validación cruzada basada en el tiempo. En este caso la división no es aleatoria, si no que los datos de entrenamiento deben ser siempre anteriores a los datos de prueba para evitar "fuga de datos" y simular un escenario no real

12.2 Datos desbalanceados

Si las clases en un problema de clasificación están muy desbalanceadas, se utiliza la validación cruzada k-fold estratificada. Esta variación garantiza que la proporción de clases se mantenga aproximadamente igual en cada pliegue de entrenamiento y validación.

12.3 Modelos de agrupamiento

La validación cruzada no se aplica directamente a modelos de aprendizaje no supervisado.

13 Ejercicios prácticos propuestos

1. Aplicar validación cruzada a distintos modelos (SVM, Árbol de decisión, Regresión logística).
2. Comparar K-Fold y Stratified K-Fold en un dataset desbalanceado.
3. Evaluar el impacto del número de folds en el tiempo de ejecución.

14 Bibliografía y lecturas recomendadas

e