Cross Validation

(Validación Cruzada)

Analítica de Datos, Universidad de San Andrés

Si encuentran algún error en el documento o hay alguna duda, mandenmé un mail a rodriguezf@udesa.edu.ar y lo revisamos.

1. Introducción

La validación cruzada es una técnica de evaluación de modelos que nos permite estimar qué tan bien un modelo de aprendizaje automático se generalizará a datos nuevos e independientes. Es especialmente útil cuando:

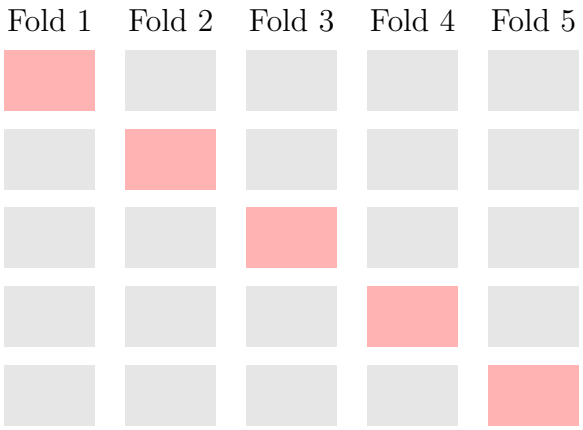
- La cantidad de datos disponibles es limitada
- Queremos evitar el sobreajuste (overfitting)
- Necesitamos una estimación más robusta del rendimiento del modelo

2. K-Fold Cross Validation

La técnica más común de validación cruzada es k-fold CV, donde:

1. Los datos se dividen en k partes iguales (folds)
2. Se realizan k iteraciones donde:
    - Se usa una parte como conjunto de validación
    - Se usan las k-1 partes restantes como conjunto de entrenamiento
3. Se promedian los resultados de las k iteraciones

A continuación se muestra una representación visual de cómo funciona la validación cruzada con 5 folds. En cada iteración, un fold diferente (marcado en rojo) se utiliza como conjunto de validación, mientras que los demás folds (en gris) se utilizan para entrenar el modelo.



3. Ejemplo Práctico

Consideremos un conjunto de datos con 100 observaciones y k=5:

Iteración	Error de Entrenamiento	Error de Validación
1	0.15	0.18
2	0.14	0.19
3	0.16	0.17
4	0.15	0.20
5	0.14	0.18
Promedio	0.148	0.184

El error promedio de validación (0.184) es una estimación más robusta del error de generalización que si hubiéramos usado una única división train-test.

- El error de entrenamiento es consistentemente menor que el error de validación, lo que indica un leve overfitting del modelo.
- La variación en los errores de validación (entre 0.17 y 0.20) sugiere que el modelo es relativamente estable.
- La diferencia promedio entre el error de entrenamiento y validación (0.036) nos da una idea de cuánto podría estar el modelo sobreajustándose a los datos.

4. Consideraciones Importantes

4.1. Elección del número de folds (k)

La elección del número de folds (k) es crucial y presenta diferentes trade-offs:

- **k=5:**
  - Mayor sesgo pero menor varianza
  - Más rápido computacionalmente
  - Útil cuando el conjunto de datos es grande
- **k=10:**
  - Menor sesgo pero mayor varianza
  - Más preciso en la estimación del error
  - Recomendado para conjuntos de datos más pequeños

4.2. Ordenamiento y mezcla de datos

- El orden de los datos puede afectar significativamente los resultados de la validación cruzada, hay que mezclar aleatoriamente los datos antes de realizar las divisiones
- Para garantizar la reproducibilidad de los resultados, utilizar un valor fijo de *random_state*

4.3. Preprocesamiento de datos

- Realizar preprocesamiento **dentro de cada fold** para evitar data leakage (que el conjunto de entrenamiento tenga información del conjunto de validación)
- Incluir: escalado, normalización, codificación categórica, manejo de faltantes y selección de features

5. Implementación en Python

```python
from sklearn.model_selection import KFold, cross_val_score
from sklearn.linear_model import LogisticRegression

# Crear el modelo
model = LogisticRegression()

# Configurar 5-fold CV
kfold = KFold(n_splits=5, shuffle=True, random_state=42)

# Calcular scores
scores = cross_val_score(model, X, y, cv=kfold)

# Imprimir resultados
print(f"Scores por fold: {scores}")
print(f"Score promedio: {scores.mean():.3f}")
print(f"Desviacion estandar: {scores.std():.3f}")
```