



**Ingeniería en Inteligencia Artificial, Machine Learning**

Semestre: 2025-1, 5BM1, Ejercicio de Laboratorio 4:  
Métodos de validación

Fecha de entrega: 14 de Octubre de 2024



## **Ejercicio de Laboratorio 4: Métodos de validación**

**Machine Learning**

**Grupo:** 5BM1

**Profesor:** Andrés Floriano García

**Integrantes:**

Juan Manuel Alvarado Sandoval  
Alexander Iain Crombie Esquinca  
Herrera Saavedra Jorge Luis  
Quiñones Mayorga Rodrigo

# Contents

|   |                       |    |
|---|-----------------------|----|
| 1 | Introducción          | 3  |
| 2 | Conceptos Básicos     | 3  |
| 3 | Métodos de Validación | 3  |
| 4 | Resultados            | 3  |
| 5 | Referencias           | 11 |
| 6 | Conclusión            | 11 |

# 1 Introducción

En esta práctica, exploramos diferentes métodos de validación cruzada para evaluar el rendimiento de modelos de Machine Learning en distintos conjuntos de datos. Los métodos utilizados incluyen Hold-Out, K-Fold Cross-Validation y Leave-One-Out. Estos métodos permiten generar conjuntos de entrenamiento y prueba disjuntos y garantizan que los resultados obtenidos sean representativos del rendimiento real del modelo en nuevos datos.

## 2 Conceptos Básicos

- **Validación cruzada:** Es una técnica de evaluación de modelos donde los datos disponibles se dividen en diferentes conjuntos de entrenamiento y prueba para verificar su capacidad predictiva.
- **Hold-Out:** Separa los datos en un conjunto de entrenamiento y uno de prueba con una proporción fija ( $r$ ).
- **K-Fold Cross-Validation:** Divide los datos en  $K$  subconjuntos, entrenando el modelo en  $K-1$  subconjuntos y probando en el subconjunto restante.
- **Leave-One-Out:** Es un caso especial de K-Fold donde  $K$  es igual al número de ejemplos en el conjunto de datos. Cada ejemplo se usa como conjunto de prueba una vez.

## 3 Métodos de Validación

### 3.1 Hold-Out

Este método consiste en separar el conjunto de datos en dos partes disjuntas, una para entrenamiento y otra para prueba, con una proporción  $r$  fijada por el usuario. Por ejemplo, si  $r = 0.7$ , el 70% de los datos se utilizan para entrenar el modelo y el 30% restante se usa para probar su rendimiento.

## 4 Resultados

### 4.1 plant dataset

### 4.2 Car evaluation dataset

9]:

|   | name                  | state | state2 |
|---|-----------------------|-------|--------|
| 0 | abelia                | fl    | nc     |
| 1 | abelia x grandiflora  | fl    | nc     |
| 2 | abelmoschus moschatus | hi    | pr     |
| 3 | abies alba            | nc    | NaN    |
| 4 | abies bracteata       | ca    | NaN    |

Figure 1: Dataset de plant

```
[20]: from sklearn.model_selection import train_test_split, KFold, LeaveOneOut

# Parámetro: proporción de prueba
r = 0.2

# División Hold-Out
train_set, test_set = train_test_split(df, test_size=r, random_state=42, shuffle=True)

print(f"Tamaño del conjunto de entrenamiento: {train_set.shape}")
print(f"Tamaño del conjunto de prueba: {test_set.shape}")

Tamaño del conjunto de entrenamiento: (13151, 3)
Tamaño del conjunto de prueba: (3288, 3)
```

Figure 2: Hold out aplicado a plant

Fold 1:

- Tamaño del conjunto de entrenamiento: (13151, 3)
- Tamaño del conjunto de prueba: (3288, 3)

Fold 2:

- Tamaño del conjunto de entrenamiento: (13151, 3)
- Tamaño del conjunto de prueba: (3288, 3)

Fold 3:

- Tamaño del conjunto de entrenamiento: (13151, 3)
- Tamaño del conjunto de prueba: (3288, 3)

Fold 4:

- Tamaño del conjunto de entrenamiento: (13151, 3)
- Tamaño del conjunto de prueba: (3288, 3)

Fold 5:

- Tamaño del conjunto de entrenamiento: (13152, 3)
- Tamaño del conjunto de prueba: (3287, 3)

Figure 3: Impresión de proporciones de K folds en plant dataset

---

```
Iteración 1:
- Conjunto de entrenamiento: (16438, 3)
- Conjunto de prueba: (1, 3)

Iteración 2:
- Conjunto de entrenamiento: (16438, 3)
- Conjunto de prueba: (1, 3)

Iteración 3:
- Conjunto de entrenamiento: (16438, 3)
- Conjunto de prueba: (1, 3)

Iteración 4:
- Conjunto de entrenamiento: (16438, 3)
- Conjunto de prueba: (1, 3)

Iteración 5:
- Conjunto de entrenamiento: (16438, 3)
- Conjunto de prueba: (1, 3)

...
Total de iteraciones: 5
```

Figure 4: Leave one out en plant, imprimiendo 5 iteraciones

---

### Hold-Out Validation:

- Conjunto de entrenamiento: (1382, 6)
- Conjunto de prueba: (346, 6)

Figure 5: Hold out en dataset car evaluation

---

## K-Fold Cross-Validation:

### Fold 1:

- Conjunto de entrenamiento: (1382, 6)
- Conjunto de prueba: (346, 6)

### Fold 2:

- Conjunto de entrenamiento: (1382, 6)
- Conjunto de prueba: (346, 6)

### Fold 3:

- Conjunto de entrenamiento: (1382, 6)
- Conjunto de prueba: (346, 6)

### Fold 4:

- Conjunto de entrenamiento: (1383, 6)
- Conjunto de prueba: (345, 6)

### Fold 5:

- Conjunto de entrenamiento: (1383, 6)
- Conjunto de prueba: (345, 6)

Figure 6: proporciones de K-folds en dataset car evaluation



```

loo = LeaveOneOut()

# Iterar sobre Leave-One-Out
print("Leave-One-Out Cross-Validation:")
contador = 1
for train_index, test_index in loo.split(X):
    train_set = X.iloc[train_index]
    test_set = X.iloc[test_index]

    print(f"Iteración {contador}:")
    print(f" - Conjunto de entrenamiento: {train_set.shape}")
    print(f" - Conjunto de prueba: {test_set.shape}\n")

    contador += 1
    if contador > 5: # Solo mostrar las primeras 5 iteraciones
        print("...")
        break

print(f"Total de iteraciones: {contador - 1}")

```

Figure 7: impresion de las primeras 5 iteraciones de Leave one out

Leave-One-Out Cross-Validation:

Iteración 1:

- Conjunto de entrenamiento: (1727, 6)
- Conjunto de prueba: (1, 6)

Iteración 2:

- Conjunto de entrenamiento: (1727, 6)
- Conjunto de prueba: (1, 6)

Iteración 3:

- Conjunto de entrenamiento: (1727, 6)
- Conjunto de prueba: (1, 6)

Iteración 4:

- Conjunto de entrenamiento: (1727, 6)
- Conjunto de prueba: (1, 6)

Iteración 5:

- Conjunto de entrenamiento: (1727, 6)
- Conjunto de prueba: (1, 6)

...

Total de iteraciones: 5

Figure 8: Tamaños de los conjuntos en leave one out

## 5 Conclusión

Los diferentes métodos de validación cruzada ofrecen formas variadas de evaluar el rendimiento de los modelos. Hold-Out es sencillo de implementar, pero puede no ser representativo en datasets pequeños. K-Fold es más robusto, mientras que Leave-One-Out ofrece una evaluación precisa a costa de un mayor costo computacional. En general, la elección del método adecuado depende del tamaño del conjunto de datos y de los recursos disponibles.

## 6 Referencias

- UCI Machine Learning Repository. (n.d.). *Plant Data Set*. Recuperado de <https://archive.ics.uci.edu/ml/datasets/Plant>
- UCI Machine Learning Repository. (n.d.). *Car Evaluation Data Set*. Recuperado de <https://archive.ics.uci.edu/ml/datasets/car+evaluation>