

---

---

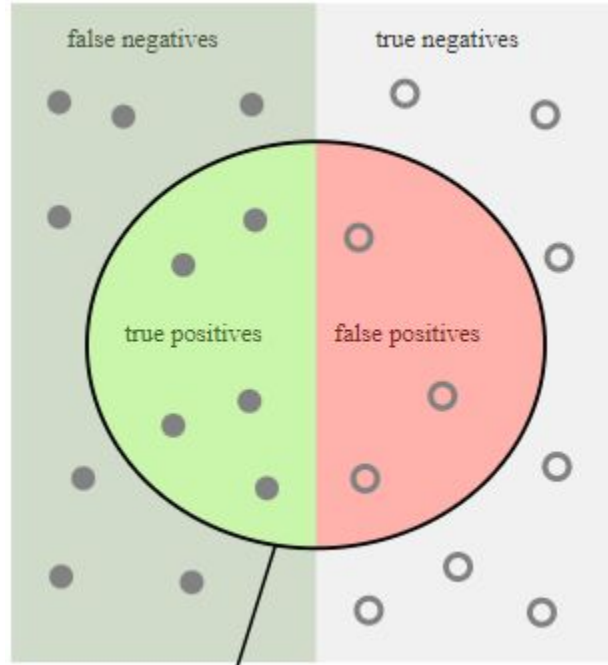
# Validación cruzada

Curso de inteligencia artificial-Intermedio

---

# Objetivo

Entender en qué consiste la validación cruzada y sus aplicaciones en modelos de ML.



---

# Validación cruzada

## Teoría

---

# Introducción

La validación cruzada es una técnica para evaluar el rendimiento de un modelo de machine learning, asegurando que sea generalizable y no esté sobreajustado a los datos de entrenamiento. La validación cruzada ayuda a estimar cómo se comportará un modelo en datos no vistos, dividiendo los datos en múltiples subconjuntos.

# ¿Por qué usar validación cruzada?

- **Generalización:** Asegura que el modelo no solo se ajuste bien a los datos de entrenamiento, sino que también funcione bien en datos nuevos.

# ¿Por qué usar validación cruzada?

- **Generalización:** Asegura que el modelo no solo se ajuste bien a los datos de entrenamiento, sino que también funcione bien en datos nuevos.
- **Prevención de Sobreajuste:** Evita que el modelo se ajuste demasiado a los datos de entrenamiento específicos.

# ¿Por qué usar validación cruzada?

- **Generalización:** Asegura que el modelo no solo se ajuste bien a los datos de entrenamiento, sino que también funcione bien en datos nuevos.
- **Prevención de Sobreajuste:** Evita que el modelo se ajuste demasiado a los datos de entrenamiento específicos.
- **Estimación del Rendimiento del Modelo:** Proporciona una medida más robusta del rendimiento del modelo en comparación con una sola partición de entrenamiento/prueba.

# Tipos de Validación Cruzada

## Validación Cruzada k-Fold:

- **Procedimiento:**
  1. Divide el conjunto de datos en  $k$  subconjuntos (o "folds").



# Tipos de Validación Cruzada

## Validación Cruzada k-Fold:

- **Procedimiento:**
  1. Divide el conjunto de datos en  $k$  subconjuntos (o "folds").
  2. Entrena el modelo  $k$  veces, usando  $k-1$  folds para el entrenamiento y el fold restante para la prueba.

# Tipos de Validación Cruzada

## Validación Cruzada k-Fold:

- **Procedimiento:**
  1. Divide el conjunto de datos en  $k$  subconjuntos (o "folds").
  2. Entrena el modelo  $k$  veces, usando  $k-1$  folds para el entrenamiento y el fold restante para la prueba.
  3. Promedia las **métricas** de rendimiento obtenidas en cada fold para obtener una estimación general.

# k-fold validation



# k-fold validation

**Ventajas:** Utiliza todos los datos para entrenamiento y prueba, proporcionando una evaluación más robusta.

**Desventajas:** Puede ser computacionalmente costoso si  $k$  es grande.

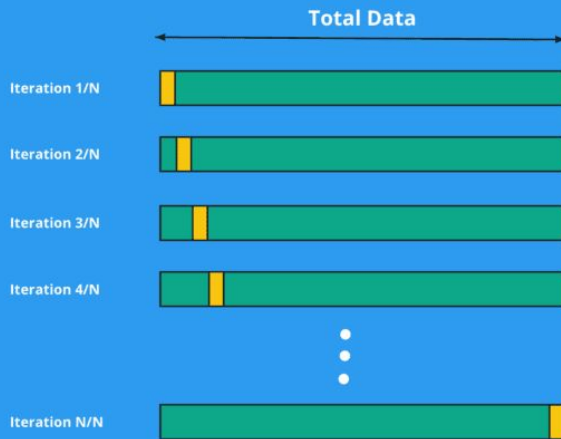
# Validación Cruzada Leave-One-Out (LOO)

## **Procedimiento:**

Cada instancia del conjunto de datos se usa como el conjunto de prueba una vez, mientras que el resto sirve como entrenamiento y se promedian las métricas usadas.

# Validación Cruzada Leave-One-Out (LOO)

## LOOCV: Leave One Out Cross Validation



# Validación Cruzada Leave-One-Out (LOO)

- **Ventajas:** Proporciona una evaluación casi sin sesgo, especialmente útil para conjuntos de datos pequeños.

# Validación Cruzada Leave-One-Out (LOO)

- **Ventajas:** Proporciona una evaluación casi sin sesgo, especialmente útil para conjuntos de datos pequeños.
- **Desventajas:** Muy costosa computacionalmente para conjuntos de datos grandes.

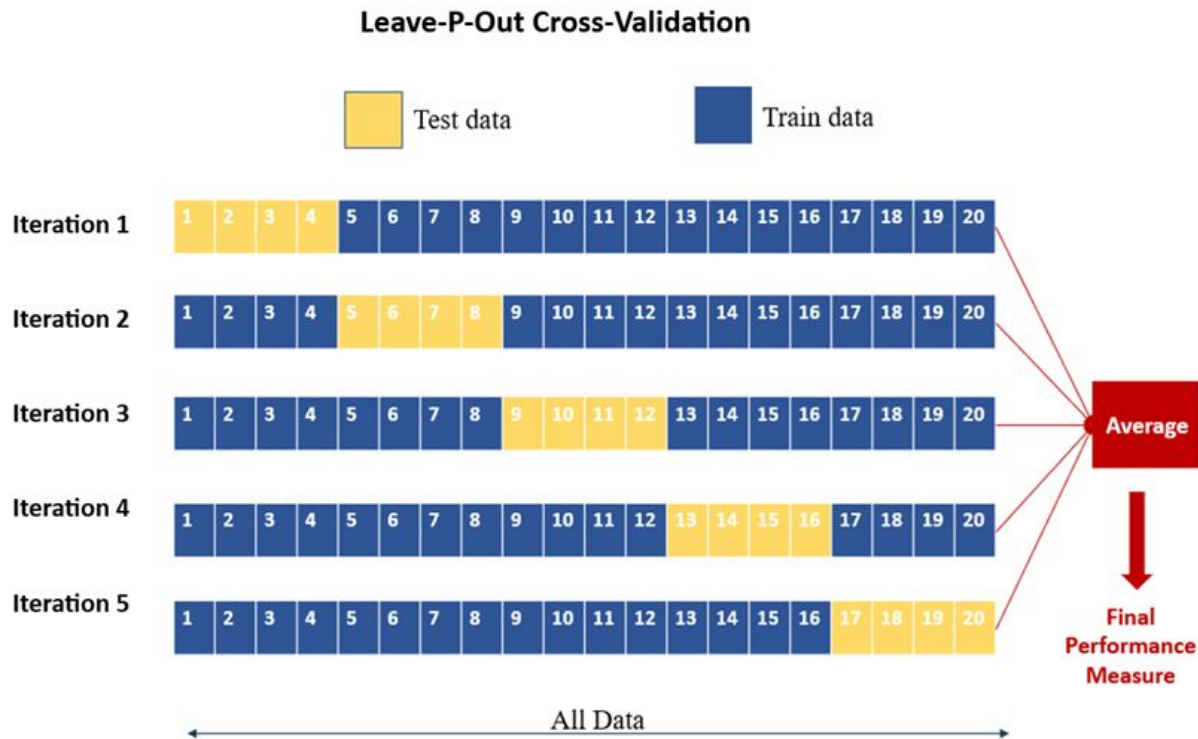


# Validación Cruzada Leave-p-Out

## **Procedimiento:**

Similar a LOO, pero se excluyen  $p$  instancias en cada iteración.

# Validación Cruzada Leave-p-Out



# Validación Cruzada Leave-p-Out

.

**Ventajas:** Flexibilidad para ajustar el tamaño del conjunto de prueba.

# Validación Cruzada Leave-p-Out

.

**Ventajas:** Flexibilidad para ajustar el tamaño del conjunto de prueba.

**Desventajas:** La complejidad computacional aumenta con el valor de  $p$ .

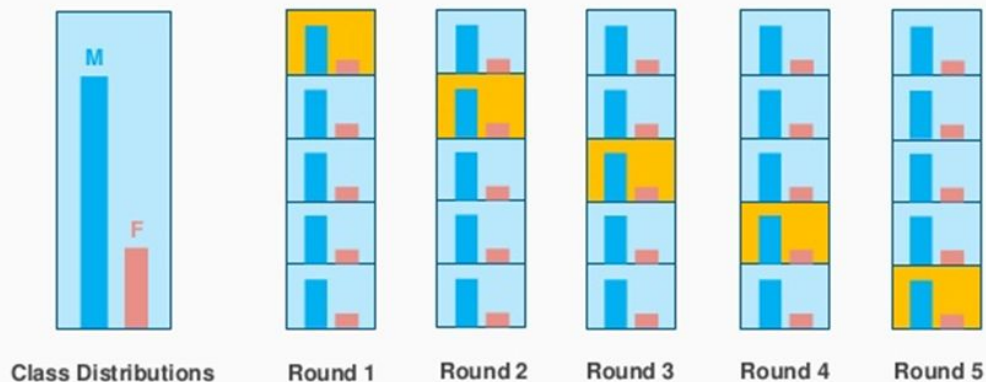
# Stratified k-fold Cross Validation

## Procedimiento:

1. Divide los datos en folds manteniendo la proporción de clases en cada fold, útil para problemas de clasificación con datos desequilibrados.

# Stratified k-fold Cross Validation

Stratified K-fold Cross Validation ( $K = 5$ )



*Keep the distribution of classes in each fold*

Training Data

Validation Data

# Stratified k-fold Cross Validation

**Ventajas:** Garantiza que cada fold sea representativo de la clase de distribución.

**Desventajas:** Puede ser menos eficiente si el número de clases es grande.

# Stratified k-fold Cross Validation

**Ventajas:** Garantiza que cada fold sea representativo de la clase de distribución.

**Desventajas:** Puede ser menos eficiente si el número de clases es grande.



# Comparación métodos de validación cruzada

## 1. Validación Cruzada k-Fold

### Uso:

- **Tamaño del Conjunto de Datos:** Adecuada para conjuntos de datos de tamaño medio a grande.
- **Aplicaciones Generales:** Es un buen punto de partida y proporciona una buena estimación del rendimiento del modelo.
- **Computación:** Más eficiente que Leave-One-Out (LOO) ya que el modelo se entrena solo k veces.

# Comparación métodos de validación cruzada

## **Cuándo Usarla:**

- Cuando tienes suficiente capacidad computacional y un conjunto de datos grande.
- Como una técnica estándar para evaluar el rendimiento del modelo.

# Comparación métodos de validación cruzada

## 2. Validación Cruzada Leave-One-Out (LOO)

Uso:

- **Tamaño del Conjunto de Datos:** Adecuada para conjuntos de datos pequeños.
- **Aplicaciones Específicas:** Útil cuando cada instancia es valiosa y el conjunto de datos es pequeño.

# Comparación métodos de validación cruzada

## **Cuándo Usarla:**

- Cuando el conjunto de datos es pequeño y se desea una evaluación precisa.
- Cuando la computación no es una limitación.

# Comparación métodos de validación cruzada

## 3. Validación Cruzada Leave-p-Out

### Uso:

- **Tamaño del Conjunto de Datos:** Puede ser útil tanto para conjuntos de datos pequeños como grandes, pero la eficiencia depende de  $p$ .
- **Aplicaciones Específicas:** Flexible en el tamaño del conjunto de prueba.

# Comparación métodos de validación cruzada

## Desventajas:

- La complejidad computacional aumenta con el tamaño de  $p$ .

# Comparación métodos de validación cruzada

## 4. Validación Cruzada Estratificada

### Uso:

- **Tamaño del Conjunto de Datos:** Adecuada para problemas con conjuntos de datos desbalanceados.
- **Aplicaciones Específicas:** Importante en problemas de clasificación con clases desbalanceadas.

# Comparación métodos de validación cruzada

## **Cuándo Usarla:**

- Cuando trabajas con conjuntos de datos desbalanceados.
- Para obtener una evaluación que refleje mejor la distribución de clases.



# Resumen

- **Conjuntos de Datos Grandes y Recursos Computacionales Adecuados:** Usa k-Fold para un equilibrio entre eficiencia y precisión.
- **Conjuntos de Datos Pequeños:** Considera Leave-One-Out para una evaluación exhaustiva, si la computación lo permite.
- **Problemas de Clasificación con Clases Desbalanceadas:** Usa validación cruzada estratificada para asegurar que cada fold tenga una representación adecuada de todas las clases.

---

# Implementación en Python

---

# Nuevas librerías

```
from sklearn.model_selection
import cross_val_score, KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
from sklearn.metrics import make_scorer, precision_score,
recall_score, f1_score
```

## Importaciones:

- `cross_val_score` para realizar validación cruzada.
- `KFold` para definir la técnica de validación cruzada k-Fold.
- `RandomForestClassifier` como el modelo de clasificación.
- `load_iris` para cargar un conjunto de datos de ejemplo.
- `make_scorer`, `precision_score`, `recall_score`, y `f1_score` para definir y calcular métricas de evaluación.

# Código k-fold cross validation

```
kf = KFold(n_splits=k, shuffle=True, random_state=42)
```

## Definición de k-Fold:

- `KFold(n_splits=k, shuffle=True, random_state=42)` define el número de folds y asegura que los datos se barajen antes de dividirse.

# Código k-fold cross validation

```
accuracy_scores = cross_val_score(model, X, y, cv=kf,
scoring='accuracy')
precision_scores = cross_val_score(model, X, y, cv=kf,
scoring=make_scorer(precision_score, average='weighted'))
recall_scores = cross_val_score(model, X, y, cv=kf,
scoring=make_scorer(recall_score, average='weighted'))
f1_scores = cross_val_score(model, X, y, cv=kf,
scoring=make_scorer(f1_score, average='weighted'))
```

## Validación Cruzada:

- `cross_val_score(model, X, y, cv=kf, scoring='accuracy')` realiza la validación cruzada y calcula la exactitud promedio.
- `make_scorer` permite definir otras métricas (precisión, exhaustividad, y F1 Score) usando `cross_val_score`.