Introducción a la Ciencia de Datos

Maestría en Probabilidad y Estadística

Dr. Marco Antonio Aquino López

Centro de Investigación en Matemáticas

Agosto–Diciembre 2025

Motivación

- Ya hemos visto varios clasificadores: Bayes, LDA/QDA, Fisher, k-NN.
- Todos ellos producen una **regla de decisión** a partir de los datos de entrenamiento.
- Pregunta fundamental: ¿qué tan bien funcionará esa regla en **nuevos datos**?
- Aquí aparece la necesidad de estudiar los **métodos de validación**.

Error aparente

- **Error aparente (training error):** mide qué tan bien el clasificador reproduce las etiquetas en los mismos datos usados para entrenarlo.

$$\hat{L}_{\text{train}} = \frac{1}{n} \sum_{i=1}^{n} 1\{g_n(X_i) \neq Y_i\}.$$

  - Tiende a ser demasiado optimista.
  - Puede ser incluso 0 para métodos muy flexibles (ej. 1-NN).

Error real

- **Error real (riesgo de generalización):** mide el desempeño esperado en una nueva observación (X, Y) independiente.

$$L(g_n) = \Pr\left(g_n(X) \neq Y\right).$$

  - Refleja la capacidad de predecir en datos no vistos.
  - Es la cantidad que realmente nos interesa.

- En general:

$$\hat{L}_{\text{train}} \leq L(g_n).$$

- **Meta de la validación:** obtener un buen estimador de $L(g_n)$ usando sólo los datos disponibles.

Sobreajuste y necesidad de validación

- Un clasificador muy flexible (ej. 1-NN) puede lograr error de entrenamiento casi cero.
- Pero esto no garantiza buen desempeño en datos nuevos: **sobreajuste**.
- Un clasificador demasiado rígido puede tener alto sesgo: **subajuste**.
- Necesitamos métodos de validación para encontrar el balance correcto y comparar clasificadores de manera justa.

ver animación

Validación simple (hold-out)

- Idea: separar los datos en dos subconjuntos disjuntos:
  - **Entrenamiento**: para ajustar el clasificador g_n.
  - **Validación**: para estimar su error de generalización.
- Procedimiento:
  1. Dividir la muestra en entrenamiento ($\mathcal{D}_{\text{train}}$) y validación ($\mathcal{D}_{\text{val}}$).
  2. Entrenar el clasificador con $\mathcal{D}_{\text{train}}$.
  3. Evaluar el error en $\mathcal{D}_{\text{val}}$:

$$\hat{L}_{\text{val}} = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(X_i, Y_i) \in \mathcal{D}_{\text{val}}} 1\{g_{\text{train}}(X_i) \neq Y_i\}.$$

- $\hat{L}_{\text{val}}$ es un estimador de $L(g_n)$.

Hold-out: ventajas y desventajas

Ventajas:

- Muy simple de implementar.
- Computacionalmente eficiente (se entrena solo una vez).

Desventajas:

- Estimación del error depende fuertemente de la partición elegida.
- Alta varianza: distintas divisiones pueden dar estimaciones muy diferentes.
- Reduce el tamaño de la muestra usada para entrenar.

Ejemplo ilustrativo del hold-out

- Supongamos que separamos 70% de los datos para entrenamiento y 30% para validación.
- Dos particiones diferentes pueden dar estimaciones muy distintas de $\hat{L}_{\text{val}}$.
- En muestras pequeñas, esta inestabilidad se acentúa.

Idea clave

El método hold-out es útil como introducción, pero poco confiable en la práctica. Esto motiva el uso de **validación cruzada**, que promedia sobre múltiples particiones.

Validación cruzada (k-fold)

- Idea: usar **todas las observaciones** para entrenamiento y validación, en diferentes particiones.
- Procedimiento:
  1. Dividir la muestra en k bloques (folds) aproximadamente del mismo tamaño.
  2. Para cada fold $j = 1, \ldots, k$:
     - ★ Entrenar el clasificador $g^{(-j)}$ con los $k-1$ folds restantes.
     - ★ Evaluar en el fold j (datos no usados en entrenamiento).
  3. Promediar los errores:

$$\hat{L}_{CV} = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{|D_j|} \sum_{(X_i, Y_i) \in D_j} 1\{g^{(-j)}(X_i) \neq Y_i\}.$$

- Así cada observación se usa una vez para validación y $k-1$ veces para entrenamiento.

Justificación de la validación cruzada

- Cada fold ofrece una estimación insesgada del riesgo, condicionado al clasificador entrenado sin esos datos.
- El promedio $\hat{L}_{CV}$ estabiliza la estimación al **reducir la varianza** respecto al hold-out.
- Uso eficiente de los datos:
  - Todos los puntos sirven para entrenar en algún fold.
  - Todos los puntos sirven para validar en algún fold.

Ventajas y desventajas de k-fold CV

Ventajas:

- Estimación más estable que el hold-out.
- Usa toda la muestra tanto para entrenar como para validar.
- Recomendado para comparar clasificadores.

Desventajas:

- Requiere entrenar el modelo k veces (más costo computacional).
- Folds diferentes pueden inducir variabilidad si n es pequeño.
- Asume independencia entre observaciones (cuidado en series de tiempo o datos correlacionados).

Variantes de validación cruzada

- **Leave-One-Out (LOO):**
  - $k = n$, cada observación se deja fuera una vez.
  - Casi insesgado, pero alta varianza y costo computacional elevado.
- **Repeated CV:** repetir k-fold con particiones aleatorias distintas para mayor estabilidad.
- **Stratified CV:** mantener la proporción de clases en cada fold (importante con clases desbalanceadas).
- **Group/Blocked CV:** asegurar que observaciones correlacionadas (ej. mismo sujeto) caigan en el mismo fold.

Leave-One-Out Cross-Validation (LOO-CV)

- Caso extremo de k-fold donde $k = n$.
- Procedimiento:
  1. Para cada observación $i = 1, \ldots, n$:
     - ★ Entrenar el clasificador $g^{(-i)}$ con todos los datos excepto (X_i, Y_i).
     - ★ Evaluar el error en (X_i, Y_i).
  2. Promediar todos los errores:

$$\hat{L}_{LOO} = \frac{1}{n} \sum_{i=1}^{n} 1\{g^{(-i)}(X_i) \neq Y_i\}.$$

- Cada observación se utiliza exactamente una vez como validación.

Propiedades de LOO-CV

- **Uso eficiente:** casi todos los datos se emplean en el entrenamiento en cada iteración.
- **Insesgado:** estimación del error muy cercana al riesgo verdadero.
- **Alta varianza:** depende mucho de una sola observación dejada fuera.
- **Costo computacional elevado:** requiere entrenar el modelo n veces.
- Especialmente útil en muestras pequeñas, pero menos recomendable en problemas grandes.

Validación cruzada repetida

- Extensión de k-fold donde se realizan varias particiones aleatorias distintas.
- Procedimiento:
  1. Elegir un número de folds k y un número de repeticiones R.
  2. Para cada repetición $r = 1, \ldots, R$:
     - ★ Dividir los datos en k folds distintos.
     - ★ Ejecutar la validación cruzada k-fold habitual.
  3. Promediar los resultados de las R repeticiones:

$$\hat{L}_{RCV} = \frac{1}{R} \sum_{r=1}^{R} \hat{L}_{CV}^{(r)}.$$

Propiedades de la CV repetida

- Reduce la **variabilidad** que depende de una sola partición.
- Produce una estimación del error más estable y confiable.
- Útil cuando n es pequeño o cuando hay alta variabilidad entre folds.
- Desventaja: mayor costo computacional (se entrena $k \times R$ veces).
- Recomendado en la práctica con $R = 5$ o 10, y $k = 5$ o 10.

Validación cruzada estratificada

- Variante de k-fold donde cada fold mantiene la **misma proporción de clases** que el conjunto original.
- Procedimiento:
  1. Dividir los datos en k folds $D_1, \ldots, D_k$ de modo que

$$\frac{|D_j \cap \{Y = 1\}|}{|D_j|} \approx \frac{|\{Y = 1\}|}{n}, \quad \frac{|D_j \cap \{Y = 0\}|}{|D_j|} \approx \frac{|\{Y = 0\}|}{n}.$$

  2. Para cada fold j, entrenar en los $k - 1$ restantes y evaluar en D_j.
  3. Promediar los errores:

$$\hat{L}_{CV}^{\text{strat}} = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{|D_j|} \sum_{(X_i, Y_i) \in D_j} 1\{g^{(-j)}(X_i) \neq Y_i\}.$$

Propiedades de la CV estratificada

- Reduce la varianza de la estimación del error cuando las clases están desbalanceadas.
- Evita folds sin observaciones de alguna clase.
- Produce comparaciones más justas entre clasificadores en problemas de desbalance.
- En notación:

$$\mathbb{E}[\hat{L}_{CV}^{\text{strat}} \mid g] \approx L(g),$$

  pero con menor variabilidad que la CV no estratificada.
- Es la práctica estándar en clasificación supervisada.

Motivación del Bootstrap

- La validación cruzada requiere dividir los datos en folds de manera explícita.
- El **Bootstrap** es un método alternativo basado en **remuestreo con reemplazo**.
- Idea central:
  - Generar múltiples muestras bootstrap de tamaño n a partir de la muestra original.
  - Ajustar el clasificador en cada muestra bootstrap.
  - Evaluar en las observaciones que **no fueron seleccionadas** (out-of-bag).
- Permite estimar el error de generalización sin necesidad de dividir manualmente en folds.

Estimador Bootstrap out-of-bag (OOB)

- Sea B el número de remuestreos bootstrap.
- En cada remuestreo $b = 1, \ldots, B$:
  - Generar muestra bootstrap $\mathcal{D}^{*(b)}$ de tamaño n.
  - Entrenar clasificador $g^{*(b)}$ con $\mathcal{D}^{*(b)}$.
  - Evaluar en las observaciones fuera de la muestra: $\mathcal{D}^{\text{OOB}(b)}$.
- El estimador del error OOB es:

$$\hat{L}_{\text{OOB}} = \frac{1}{B} \sum_{b=1}^{B} \frac{1}{|\mathcal{D}^{\text{OOB}(b)}|} \sum_{(X_i, Y_i) \in \mathcal{D}^{\text{OOB}(b)}} 1\{g^{*(b)}(X_i) \neq Y_i\}.$$

Propiedades del estimador OOB

- Cada observación tiene probabilidad $\left(1 - \frac{1}{n}\right)^n \approx e^{-1} \approx 0.368$ de quedar fuera en un remuestreo.
- Así, en promedio, un $\sim 36.8\%$ de los datos sirven como validación en cada réplica.
- $\hat{L}_{\text{OOB}}$ aprovecha todos los datos para entrenar y validar en múltiples repeticiones.
- Ventajas:
  - No requiere particionar manualmente.
  - Produce un estimador razonable del error de generalización.
- Limitación: puede estar sesgado, especialmente en clasificadores muy flexibles.

Motivación del Bootstrap .632

- El estimador OOB puede estar sesgado:
  - Subestima el error para modelos simples.
  - Sobreestima el error para modelos muy flexibles (sobreajuste).
- Idea del método .632:
  - Combinar el error aparente (entrenamiento) y el error OOB.
  - Ponderación basada en el hecho de que $\sim 63.2\%$ de los datos aparecen en cada bootstrap.

Definición del estimador .632

- Sea $\hat{L}_{\text{train}}$ el error aparente en los datos de entrenamiento.
- Sea $\hat{L}_{\text{OOB}}$ el error estimado por observaciones out-of-bag.
- El estimador .632 se define como:

$$\hat{L}_{.632} = 0.368\,\hat{L}_{\text{train}} + 0.632\,\hat{L}_{\text{OOB}}.$$

- Justificación:
  - En cada réplica, $\sim 63.2\%$ de los datos aparecen en el entrenamiento.
  - El resto ($\sim 36.8\%$) sirve como validación.

Bootstrap .632+

- En problemas con fuerte sobreajuste, $\hat{L}_{.632}$ aún es optimista.
- Solución: el método .632+ introduce un peso adaptativo w:

$$\hat{L}_{.632+} = (1 - w)\hat{L}_{\text{train}} + w\hat{L}_{\text{OOB}}.$$

- Donde

$$w = \frac{0.632}{1 - 0.368R}, \qquad R = \frac{\hat{L}_{\text{OOB}} - \hat{L}_{\text{train}}}{L_{\text{max}} - \hat{L}_{\text{train}}}.$$

- R es la **razón de sobreajuste relativo**, y L_{max} corresponde al error de un clasificador trivial.

Propiedades del Bootstrap .632 y .632+

- **.632**:
  - Corrige el sesgo del OOB combinando error de entrenamiento y validación.
  - Funciona bien en escenarios moderados.
- **.632+**:
  - Ajusta dinámicamente el peso para manejar sobreajuste extremo.
  - Produce estimaciones más robustas en clasificadores muy flexibles.
- En la práctica, ambos métodos son útiles cuando el tamaño de muestra es pequeño y la validación cruzada tiene alta varianza.

Propiedades del Bootstrap .632 y .632+

- **.632**:
  - Corrige el sesgo del OOB combinando error de entrenamiento y validación.
  - Funciona bien en escenarios moderados.
- **.632+**:
  - Ajusta dinámicamente el peso para manejar sobreajuste extremo.
  - Produce estimaciones más robustas en clasificadores muy flexibles.
- En la práctica, ambos métodos son útiles cuando el tamaño de muestra es pequeño y la validación cruzada tiene alta varianza.

Cross-Validation vs. Bootstrap: comparación

Validación cruzada (CV):

- Divide explícitamente la muestra en folds.
- Estimación estable al promediar múltiples particiones.
- Muy usada para comparar clasificadores y elegir hiperparámetros.
- Requiere entrenar el modelo k veces (o más en Nested CV).

Bootstrap:

- Basado en remuestreo con reemplazo.
- Usa observaciones out-of-bag para validar.
- Estimadores corregidos: .632 y .632+.
- Puede ser más eficiente en muestras pequeñas.

CV vs. Bootstrap: ventajas y limitaciones

Cross-Validation:

- + Simple de entender e implementar.
- + Estándar en la práctica moderna.
- – Varianza alta si n es pequeño.
- – Costo computacional elevado en Nested CV.

Bootstrap:

- + Permite aprovechar remuestreo en n pequeño.
- + Ofrece múltiples estimadores (OOB, .632, .632+).
- – Puede ser optimista o inestable en problemas de alta dimensión.
- – Menos usado hoy en día que la CV para tuning de hiperparámetros.

Conclusión

Ambos métodos son válidos:

- CV es la elección por defecto en la práctica.
- Bootstrap resulta útil en muestras pequeñas o para complementar la estimación.

Evaluación de clasificadores

- Un clasificador no se evalúa solo por su error promedio.
- Dependiendo del problema, distintas métricas son relevantes:
  - Sensibilidad (detección de positivos).
  - Especificidad (detección de negativos).
  - Precisión, F1, etc.
- La **matriz de confusión** es la base para definir la mayoría de las métricas.

Matriz de confusión (binaria)

	Predicho $= 1$	Predicho $= 0$
Real $= 1$	TP	FN
Real $= 0$	FP	TN

- TP: verdaderos positivos.
- TN: verdaderos negativos.
- FP: falsos positivos.
- FN: falsos negativos.

Métricas derivadas

- **Exactitud (Accuracy):**

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}.$$

- **Precisión:**

$$\text{Precisión} = \frac{TP}{TP + FP}.$$

- **Sensibilidad (Recall/TPR):**

$$\text{Sensibilidad} = \frac{TP}{TP + FN}.$$

- **Especificidad (TNR):**

$$\text{Especificidad} = \frac{TN}{TN + FP}.$$

F1-score y medidas balanceadas

- **F1-score:** media armónica de precisión y sensibilidad.

$$F1 = 2 \cdot \frac{\text{Precisión} \cdot \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}.$$

- **Balanced accuracy:** promedio de sensibilidad y especificidad.

$$\text{BA} = \frac{1}{2}(\text{Sensibilidad} + \text{Especificidad}).$$

- Útiles cuando hay **desbalance de clases**.

Conclusiones

- La **validación** es esencial para estimar el error de generalización y evitar sobreajuste.
- El **hold-out** es el método más simple, pero con alta varianza.
- La **validación cruzada** (k-fold y variantes) es el estándar en la práctica.
- El **bootstrap** ofrece alternativas útiles en muestras pequeñas y corrige sesgos con los estimadores .632 y .632+.
- La **matriz de confusión** es la base para definir métricas de desempeño: exactitud, precisión, sensibilidad, especificidad, F1, entre otras.
- No existe una métrica universal: la elección depende del problema y de los costos asociados a falsos positivos y falsos negativos.

Referencias

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. 2nd ed. Springer.
- Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36(2), 111–147.
- Efron, B., & Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.
- Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1), 37–63.