



FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE



Para **estudiantes** de Educación Básica y Media.
UNIVERSIDAD DE CHILE

Clase V-III: Métodos de evaluación en *Machine Learning*

FM849 Proyecto de Ciencia de Datos: Inteligencia Artificial (IA) y sus aplicaciones

Máximo Flores Valenzuela (mflores@dcc.uchile.cl)

Universidad de Chile • 23 de agosto de 2025

Roadmap de la clase

- 1 Evaluación de métodos de aprendizaje supervisado.
- 2 Evaluación de métodos de aprendizaje no supervisado.



Evaluación en aprendizaje supervisado

Errores estadísticos de tipo I y II

Antes de introducir las métricas que permiten evaluar los errores en modelos de aprendizaje supervisado, necesitamos entender bien qué son los errores de tipo I (falso positivo) y tipo II (falso negativo).

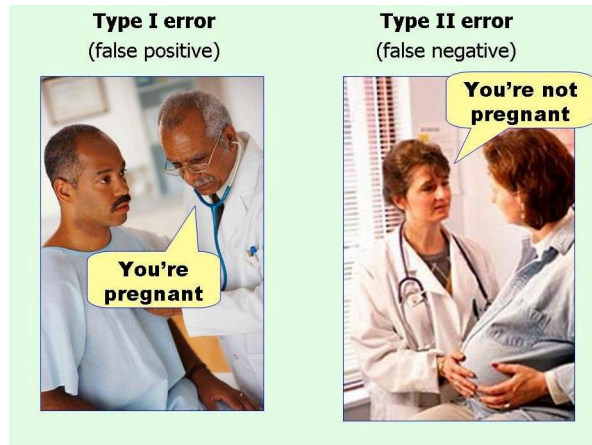


Figura 3: Errores de tipo I y II.

	Predicted YES	Predicted NO
Actual YES	TRUE positive	FALSE negative
Actual NO	FALSE positive	TRUE negative

Figura 4: Tabla de confusión que compara la predicción con el valor real.

Métrica de evaluación: *Recall*

El *recall* mide la proporción de los ejemplos positivos correctamente clasificados:

$$\text{Recall} = \frac{TP}{TP + FN}$$

donde TP son los verdaderos positivos, y FN los falsos negativos. Un falso negativo debía haber sido clasificado como positivo.

- **Utilidad:** Problemas donde queremos minimizar la cantidad de falsos negativos (p. ej., detección de cáncer).

¡Cuidado! Imagínense que tienen un examen que en el 99% de los casos da un resultado positivo. Podríamos considerar el clasificador que dice que todo es positivo, y la métrica Recall va a ser máxima, a pesar de que el clasificador no tenga capacidad de distinción entre clases.

Métrica de evaluación: *Precision*

La métrica *precision* mide la proporción de los ejemplos clasificados como positivos que fueron detectados correctamente:

$$\text{Precision} = \frac{TP}{TP + FP}$$

donde FP son los falsos positivos. Sufre del mismo problema que tiene *recall* con respecto a las clases desbalanceadas.

- **Utilidad:** Problemas donde queremos minimizar los falsos positivos (p. ej., detección de personas para una condena judicial).

Métrica de evaluación: *Accuracy*

La métrica *accuracy* mide la proporción total de clasificaciones que se hicieron correctamente:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total de clasificaciones}}$$

En el total de clasificaciones, se considera la suma de TP (verdaderos positivos), TN (verdaderos negativos), FP (falsos positivos) y FN (falsos negativos). Sufre el mismo problema de *recall* y *precision*.

Métrica de evaluación: *F1-Score*

La métrica *F1-Score* corresponde a la media armónica entre las métricas *precision* y *recall*. Matemáticamente, se define de la siguiente forma:

$$F1\text{-Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

La media armónica le da más peso a los valores menores, entonces los errores pesan más.

Visualización de errores: Matriz de confusión

En la clase pasada sobre visualización de datos vimos que la matriz de confusión es un tipo de *heatmap* que nos permite establecer relaciones entre las confusiones (errores) y aciertos del modelo.

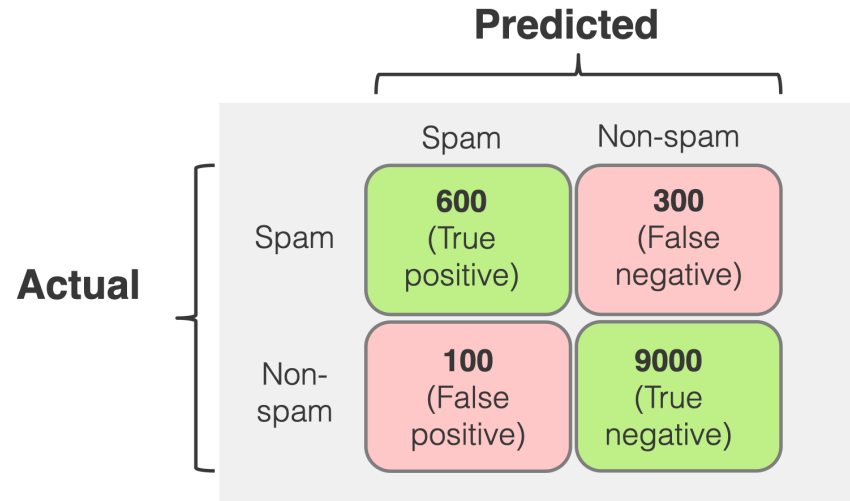


Figura 5: Visualización de una matriz de confusión.

Costo de un modelo

El costo de un modelo se puede calcular estableciendo un ponderador para cada entrada de la matriz de confusión. Si un modelo es más costoso que otro para una tarea específica, entonces es peor.

Por defecto, este ponderador es 1 para cada clase. En el caso binario, sólo tenemos 4 clases: TP, TN, FP, FN. Si queremos, por ejemplo, asignarle mayor peso a cometer un error de tipo 1 (FP), entonces podemos asignarle un ponderador de 10 a dicha clase.

- Generalmente, si nos interesa quitar penalización según cuántos aciertos hay, se le puede poner un ponderador negativo a TP y/o TN. Si no, simplemente se dejan como 0 (neutro aditivo).

Extensión de las métricas al caso multiclase

Cuando hay muchas clases ($C > 2$), hay dos formas de calcular las métricas que acabamos de ver.

- Micropromedio (*microaveraging*): se mezcla todo el sistema en una matriz de confusión binaria (TP, TN, FP, FN) y se calcula la métrica de interés sobre esa matriz.
 - **Problema:** Las clases que tienen más ejemplos tienen más dominancia.
- Macropromedio (*macroaveraging*): se computa la métrica de interés para cada clase, para posteriormente promediarlas todas con el mismo peso.
 - **Problema:** Las clases que tienen menos ejemplos generan sobrerrepresentación.

Ejemplo multiclase

Class 1: Urgent			Class 2: Normal			Class 3: Spam			Pooled		
	true urgent	true not		true normal	true not		true spam	true not		true yes	true no
system urgent	8	11	system normal	60	55	system spam	200	33	system yes	268	99
system not	8	340	system not	40	212	system not	51	83	system no	99	635
precision = $\frac{8}{8+11} = .42$			precision = $\frac{60}{60+55} = .52$			precision = $\frac{200}{200+33} = .86$			microaverage precision = $\frac{268}{268+99} = .73$		
			macroaverage precision = $\frac{.42+.52+.86}{3} = .60$								

Figura 6: Gráfica de *microaveraging* vs. *macroaveraging* para la métrica *precision* en un sistema.

Metodología de evaluación

Para obtener las observaciones que nos permiten evaluar nuestro modelo, necesitamos separar nuestro conjunto de datos en 3 partes:

- 1 Conjunto de entrenamiento: usado para entrenar al modelo. Estos datos **no** se pueden ver en el conjunto de prueba, si no le estaríamos «soplando» las respuestas al evaluarlo. Se puede ver como «el estudio del temario de la PAES».
- 2 Conjunto de validación: usado para mejorar el poder de generalización del modelo. Se puede ver como «los simulacros de la prueba PAES realizados para mejorar el rendimiento».
- 2 Conjunto de prueba: usado al final para evaluar la eficacia del modelo, es decir, qué tan bien realiza su tarea. Se puede ver como «la PAES en sí misma».

Si yo tengo acceso a la PAES al estudiar, obviamente me irá mejor. Si yo estudio cosas muy específicas en los simulacros, puede que mi rendimiento no sea tan bueno en la PAES oficial.

Implementando la evaluación: *K-Fold Cross-Validation*

- Corresponde a un algoritmo que divide el conjunto de datos en K subconjuntos lo más balanceados en elementos posible.
- En cada iteración, cada uno de esos K subconjuntos es parte del conjunto de validación, y los demás son parte del conjunto de entrenamiento.

El resultado de la evaluación es un promedio de las métricas obtenidas en cada iteración.

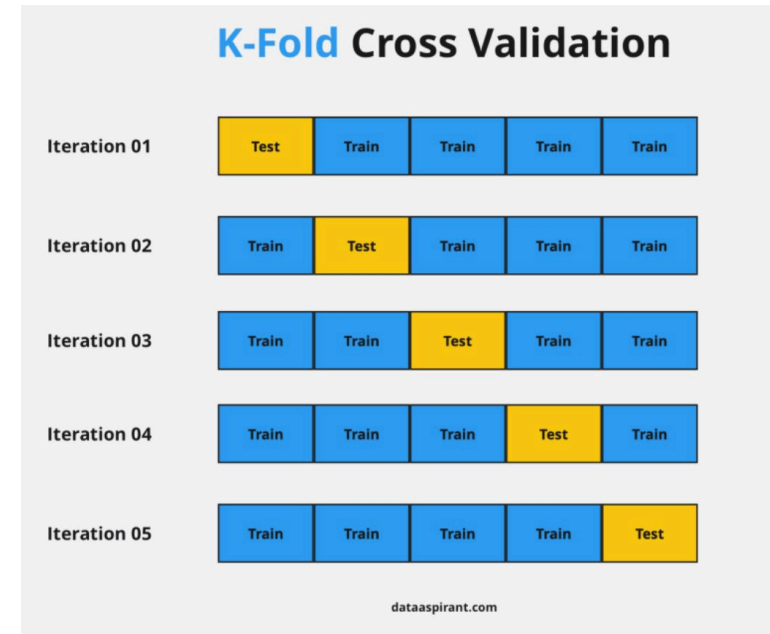


Figura 7: Visualización de *K-Fold Cross-Validation*.

Variantes de *K-Fold Cross-Validation*

Algunas de las formas alternativas para implementar este algoritmo consideran:

- *Stratified K-Fold*: garantiza que la proporción de clases en cada *fold* sea similar a la que existía en el conjunto original de datos.
- *Repeated K-Fold*: se repite varias veces el proceso de *K-Fold* para intentar desprenderse del factor azaroso.
- *Leave-One-Out* (no recomendado, salvo casos muy específicos): es el caso de *K-Fold* donde K coincide con el número total de datos.

Desventaja principal

La validación cruzada requiere K iteraciones, lo que puede ser muy costoso computacionalmente, especialmente si el número de datos es muy grande.

En la práctica, una alternativa menos costosa computacionalmente simplemente divide el conjunto de datos en las 3 partes vistas al inicio, con una proporción 70/10/20 o 70/15/15 para *train*, *validation* y *test* respectivamente.

- La partición escogida influye en los resultados de las métricas.
- Un *training set* muy pequeño induce sesgos, al igual que cuando este conjunto no tiene suficientes ejemplos de las clases.
- Si el *testing set* es muy pequeño, las métricas no son tan confiables (por eso, *Leave-One-Out* es mala idea para conjuntos de datos grandes).

Evaluación en aprendizaje no supervisado

Motivación

A pesar de que los *clusters* son más complejos de evaluar que los modelos de clasificación, es necesario saber su eficacia en la realización de su tarea.

- ¿Qué pasa si las agrupaciones encontradas son aleatorias?
- ¿Y si mis datos en realidad no tenían un patrón?

La complejidad radica en que ahora no tenemos una interpretación directa, a diferencia del aprendizaje supervisado (la etiqueta asignada **está bien** o **no**).

Tipos de métricas

La división de las métricas que se usan para evaluar la generación de *clusters* se dividen principalmente en 3 estrategias:

- No supervisadas: miden la calidad sin usar información externa (estadísticas como SSE, cohesión, separación).
- Supervisadas: incorporan el conocimiento experto externo para determinar si los *clusters* se ajustan a dicha estructura ajena a mis datos.
- Relativas: compara resultados entre distintas instancias de aplicación de un método. Por ejemplo, aplicar *K*-means dos veces y comparar su SSE.

Métricas no supervisadas

Se basan meramente en los datos; la estadística que se puede extraer desde ellos.

- Suelen basarse en índices de cohesión, relacionados con la distancia intracluster, y separación, relacionados con la distancia intercluster.
- De manera general, la validez de un conjunto de K *clusters* se puede expresar como una validez ponderada de cada *cluster* C_i , $i \in [K]$:

$$\text{validez total} = \sum_{i=1}^K w_i \cdot \text{validez}(C_i)$$

Cuando no nos interesa darle más relevancia a algún *cluster*, entonces $w_i = 1$ para todo $i \in [K]$.

Estrategias de mejora

- Si un *cluster* tiene cohesión muy baja, podemos separarlo. Así mismo, si dos *clusters* tienen una separación muy baja, podemos unirlos.
- También podemos evaluar cuánto contribuye un punto en particular a la cohesión y separación de su *cluster*.
- Una métrica que combina la cohesión con la separación es el coeficiente de Silhouette.

Coeficiente de Silhouette

Para un punto individual i :

- 1 Se calcula a_i como la distancia promedio de i a los puntos de su *cluster* (llamémosle C):

$$a_i = \frac{1}{|C| - 1} \cdot \sum_{j \in C, j \neq i} d(i, j)$$

- 2 Se calcula b_i como la mínima distancia promedio de i a puntos de otro *cluster*:

$$b_i = \min_{C' \in \mathcal{C}, C' \neq C} \frac{1}{|C'|} \cdot \sum_{j \in C'} d(i, j)$$

- 3 El coeficiente de Silhouette es $s_i = (b_i - a_i) / \max\{a_i, b_i\} \in [-1, 1]$. Mientras más cerca esté de 1, mejor.

Esto se puede generalizar para calcular el coeficiente de Silhouette de un *cluster*, o incluso del conjunto total de *clusters*.

Métricas supervisadas

- La información experta externa generalmente sí tiene etiquetas, porque el grado de conocimiento adquirido por las personas en distintos campos lo permite.
- Si tenemos ejemplos etiquetados, finalmente, podríamos ver si es consistente con lo que representan nuestros *clusters*.
- De ser así, ¿cuál es el fin del aprendizaje no supervisado?
 - ▶ A veces, los patrones capturados por los *clusters* sí pueden revelar comportamientos no triviales, por lo que es un buen complemento.
 - ▶ También, permite saber si las técnicas de *clustering* permiten generar grupos dado un problema que quiero resolver.
 - ▶ No siempre tenemos a mano varias etiquetas, podemos tener pocas, y usarlas sólo para validar...

Métrica supervisada: entropía

Corresponde al grado en que un *cluster* en específico contiene ejemplos de una sólo clase.

- 1 Para cada *cluster*, calculamos la probabilidad de que un elemento i del *cluster* pertenezca a la clase j : $p_{ij} = m_{ij}/m_i$, donde m_i es el número de elementos en el *cluster* i , y m_{ij} la cantidad de elementos de la clase j en el *cluster* i .
- 2 Calculamos la entropía del *cluster* i , considerando que hay L etiquetas en total:

$$e_i = - \sum_{j=1}^L p_{ij} \log_2 p_{ij}$$

- 3 Calculamos finalmente la entropía total del sistema, considerando que tenemos K *clusters* y m datos en total. Esta métrica es óptima cuando es baja:

$$e = \sum_{i=1}^K \frac{m_i}{m} \cdot e_i$$

Métrica supervisada: pureza

- Corresponde al nivel en que un *cluster* contiene elementos de una sólo clase, usando la clase que predomine.
- Se calcula como la probabilidad máxima de una de las clases. Para un *cluster* i :

$$\text{purity}(i) = \max_j p_{ij}$$

- Al igual que la entropía, la pureza total del sistema es un promedio ponderado, donde el ponderador corresponde a la proporción de elementos de cada *cluster* con respecto al total. Esta métrica es óptima cuando es alta:

$$\text{purity} = \sum_{i=1}^K \frac{m_i}{m} \cdot \text{purity}(i)$$

Significancia de las medidas supervisadas

No es fácil interpretar estos puntajes, a pesar de que nos puedan guiar para entender la calidad de los *clusters* generados.

- Una pureza de 0 indica una agrupación deficiente, mientras que una pureza de 1 sugiere una buena agrupación.
- Lo contrario sucede para la entropía y la SSE, donde valores bajos indican una mejor agrupación.

Validación con expertos e iteración

- La validación con expertos implica evaluar los *clusters* para determinar si producen el resultado esperado.
- Ellos nos pueden decir, dadas las descripciones que nosotros tengamos de los *clusters*, si las relaciones que se establecen en los grupos tienen sentido y por qué.
- Es importante iterar si se encuentran inconsistencias. Existen muchos motivos por los cuales el resultado pudo haber sido deficiente (p. ej., *garbage-in*, *garbage-out*, o escoger un método inadecuado).

¿Preguntas?