Evaluación de modelos de aprendizaje automático

José Hernández Orallo (jorallo@dsic.upv.es)

Universitat Politècnica de València, Valencia (www.upv.es)

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Índice

- Evaluación: Tareas y Métricas
- Sobreajuste, evaluación partida y validación cruzada
- Contingencia, costes y desbalance
- Clasificadores suaves
- Evaluación sensible al coste

(Análisis ROC)

Evaluación: dependiente de la tarea

- La evaluación depende del trabajo:
  - Aprendizaje supervisado: El problema está presentado con ejemplos de entrada y la solución deseada, la meta es aprender la regla general que convierte los datos de entrada en la solución correcta. Clasificación: La salida es categórica
    - Regresión: El resultado es numérico
  - Aprendizaje no supervisado: No se asigna ninguna etiqueta al algoritmo de aprendizaje, dejándolo solo para encontrar la estructura de su entrada de datos.
    - Clustering, reglas de asociación…
  - Aprendizaje por refuerzo: Un programa informático interactúa con ambiente controlado en el cual se debe alcanzar una meta dada: Conducir un vehículo o videojuegos.

Métricas para Clasificación

- Dado un conjunto *S* de *n* instancias, definimos el error de clasificación:

$$error_S(h) = \frac{1}{n} \sum_{x \in S} \partial(f(x), h(x))$$

Clase predicha (h(x))	Clase actual (f(x))	Error
Comprar	Comprar	No
No Comprar	Comprar	Yes
Comprar	No Comprar	Yes
Comprar	Comprar	No
No Comprar	No Comprar	No
No Comprar	Comprar	Yes
No Comprar	No Comprar	No
Comprar	Comprar	No
Comprar	Comprar	No
No Comprar	No comprar	No

Donde $\delta(a,b)=0$ si a=b y de otra manera 1.

Errores / Total

Error = 3/10 = 0.3

Métricas para Clasificación

- Medidas comunes en IR: *Precision and Recall*
- Se definen en términos de un conjunto de documentos recuperados y un conjunto de documentos relevantes.
  - Precision: El porcentaje de documentos que son relevantes para la consulta. (TP / Positivos predichos)
  - Recall: Porcentaje de documentos que se devuelven por el estudio.

    (TP / Positivos reales)
- Ambas medidas suelen combinarse en uno (significado harmónico):

$$\text{medida} - \text{F} = 2\frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

Métricas para Regresión

- Dado un conjunto *S* de *n* instancias,
  - Error Medio Absoluto:

$$MAE_S(h) = \frac{1}{n}\sum_{x \in S} \left| f(x) - h(x) \right|$$

  - Error Cuadrático Medio:

$$MSE_S(h) = \frac{1}{n}\sum_{x \in S} (f(x) - h(x))^2$$

  - Raíz Cuadrada del Error Cuadrático Medio:

$$RMSE_S(h) = \sqrt{\frac{1}{n}\sum_{x \in S} (f(x) - h(x))^2}$$

    - MSE es más sensible en los valores extremos
    - RMSE y MAE son la misma magnitud de los valores actuales

Métricas para Regresión

- Ejemplo:

Valor predicho (h(x))	Valor Actual(f(x))	**Error**	**Error^2**
100 mill. €	102 mill. €	**2**	**4**
102 mill. €	110 mill. €	**8**	**64**
105 mill. €	95 mill. €	**10**	**100**
95 mill. €	75 mill. €	**20**	**400**
101 mill. €	103 mill. €	**2**	**4**
105 mill. €	110 mill. €	**5**	**25**
105 mill. €	98 mill. €	**7**	**49**
40 mill. €	32 mill. €	**8**	**64**
220 mill. €	215 mill. €	**5**	**25**
100 mill. €	103 mill. €	**3**	**9**

MAE= 70/10 =7

MSE= 744/10 = 74,4

RMSE= sqrt(744/10) = 8.63

Métricas para Regresión

- A veces los valores del error relativo son más apropiados:
  - 10% para un error de 50 cuando se ha predicho 500

- Cuánto mejora el esquema simplemente prediciendo el promedio:
  - Error cuadrático medio relativo

$$RSE_S(h) = \frac{\sum_{x \in S} \left(f(x) - h(x)\right)^2}{\sum_{x \in S} \left(\bar{f} - f(x)\right)^2}$$

  - Error absoluto medio relativo

$$RAE_S(h) = \frac{\sum_{x \in S} \left|f(x) - h(x)\right|}{\sum_{x \in S} \left|\bar{f} - f(x)\right|}$$

- La correlación de Pearson/Spearman también es útil

Métricas para aprendizaje no supervisado

- Reglas de asociación: encontrar relaciones interesantes entre las variables de un base de datos
- Métricas comunes:
  - Soporte: Estima la popularidad de una regla
  - Confianza: Estima la exactitud de una regla

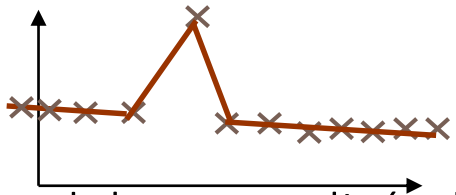
- Las reglas son ordenadas respecto a medidas que combinan ambos valores.

Métricas para aprendizaje no supervisado

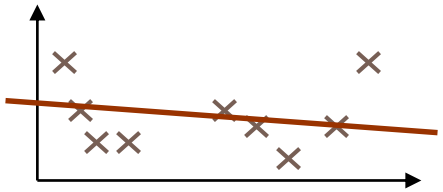
- Agrupamiento *(Clustering):* agrupar un conjunto de objetos de tal forma que los de un mismo grupo (*cluster*) son más similares entre ellos que entre los de otros *clusters*.
  - Tarea difícil de evaluar
- Algunas medidas de evaluación basadas en distancia:
  - Distancia entre los bordes de los *clusters*
  - Distancia entre centros *(centroids)* de los *clusters*
  - Radio y densidad de los *clusters*

¿Sobreajuste?

- ¿Que medida de evaluación vamos a utilizar para estimar las anteriores métricas?
  - Si utilizamos todos los datos para entrenar y evaluar los modelos, entonces tendremos un modelo sobre representado:
    - Sobreajuste*(Over-fitting):*



  - Si intentamos compensar el modelo generalizándolo *(por ejemplo, podar un árbol de datos)* obtendremos:
    - Subajuste*(Under-fitting):*



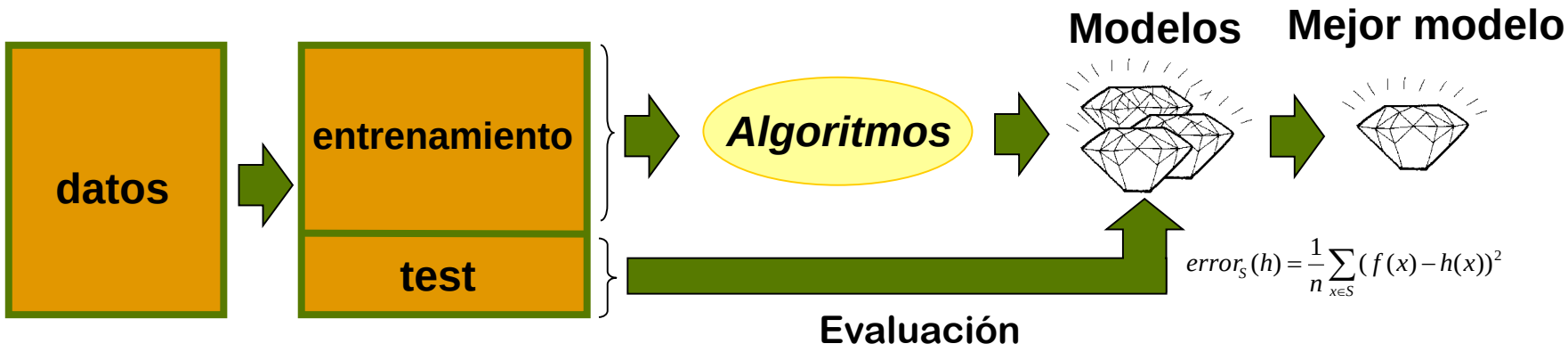
  - ¿Como podemos compensarlo?

Dividiendo entre Entrenamiento y Testeo

- Solución común(especialmente en aprendizaje supervisado):

REGLA DE ORO: ¡¡Nunca usar el mismo ejemplo para entrenar el modelo y evaluarlo!!

  - Separar entre datos de entrenamiento y de testeo



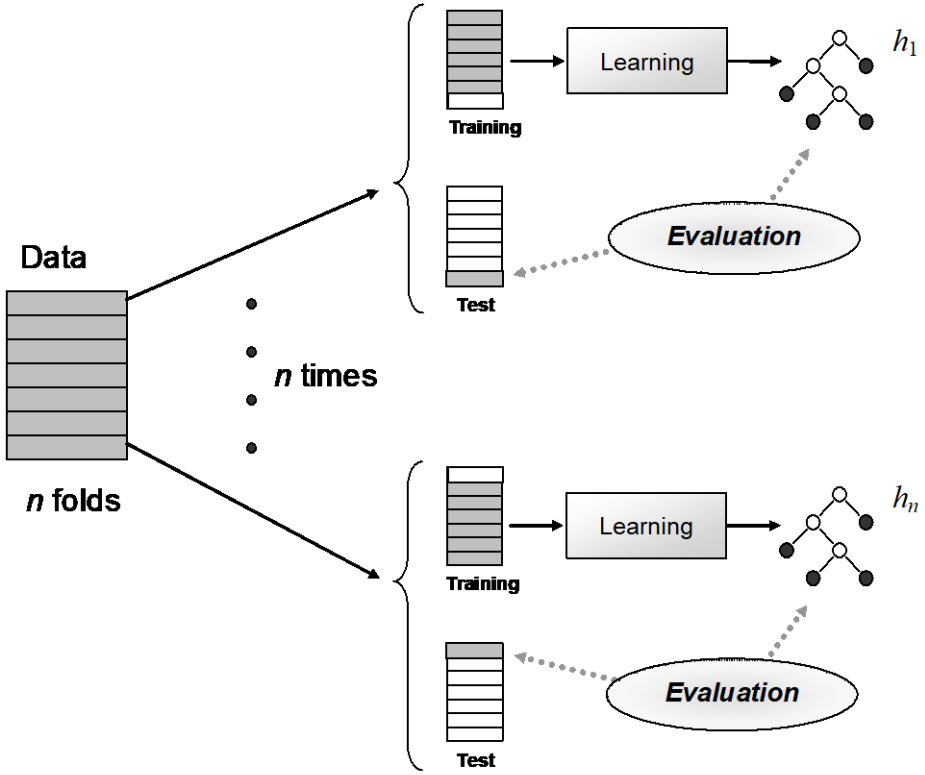
¿Qué pasa si no hay mucha información disponible?

Aprovechando al máximo los datos

- Demasiados datos de entrenamiento: evaluación pobre
- Demasiados datos de testeo: entrenamiento pobre
- ¿Podemos tener más datos de entrenamiento y testeo sin romper la regla de oro?
  - ¡Repite el experimento!
    - *Bootstrap*: llevamos a cabo n muestras (con repeticiones) y testeo con el resto.
    - Validación cruzada: Los datos se separan en *n* divisiones del mismo tamaño.
      - *Hold-out*: caso especial con tantas divisiones como ejemplos.

Validación cruzada

- ¡Podemos entrenar y probar con todos los datos!



- Tomamos todas las combinaciones posibles con *n*–1 para entrenar y el resto para testear.
- El error (o cualquier otra métrica) se calcula n veces y después se calcula la media.
- El modelo final se entrena con todos los datos.

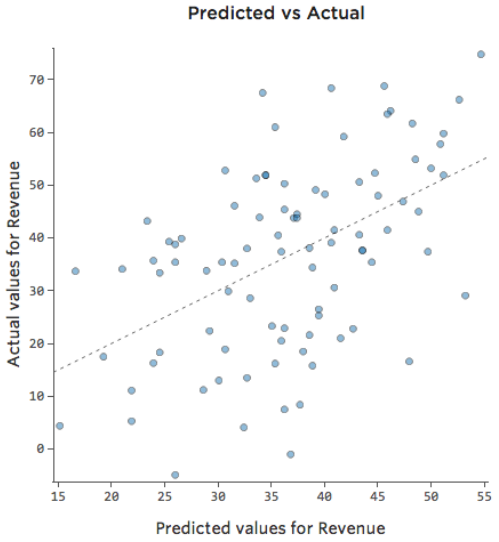
¿Dónde falla mi modelo?

- Ver la contingencia
  - Clasificación: una matriz

Actual

Pred. / *c*	Comprar	No Comprar
Comprar	4	1
No Comprar	2	3

  - Regresión: un plot

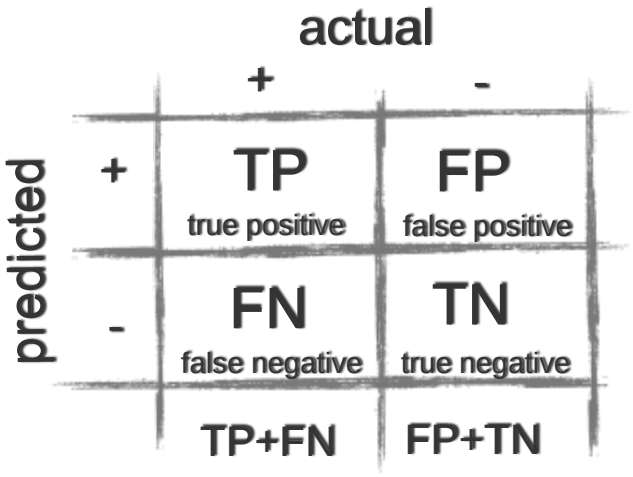


Clasificación Binaria: Matriz de confusión

- Matriz de confusión para dos clases:

Actual

Pred. *c*	Comprar	No comprar
Comprar	4	1
No comprar	2	3

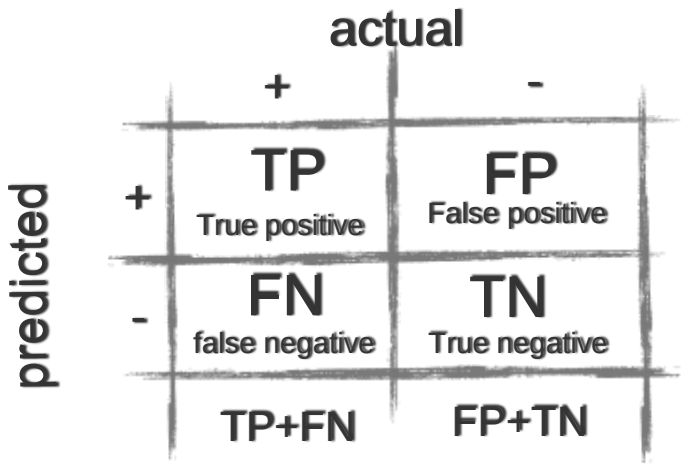


$$Accuracy = \frac{TP+TN}{N}$$

$$\text{Error}=1-Accuracy = \frac{FP+FN}{N}$$

Clasificación Binaria: Matriz de confusión

- Matriz de confusión para dos clases:

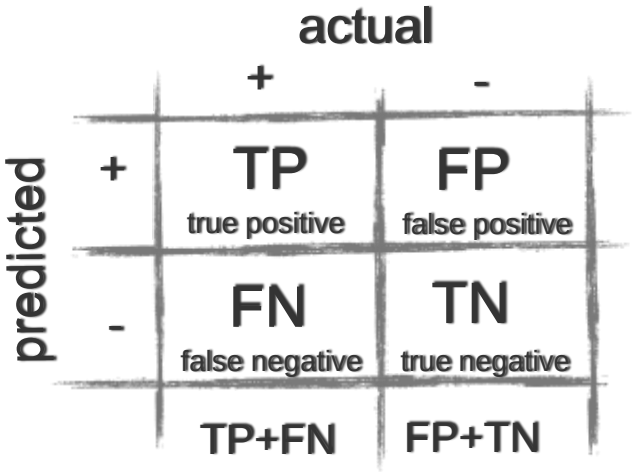


$$\text{TPRate, Sensibilidad} = \frac{TP}{TP+FN}$$

$$\text{TNRate, Especifidad} = \frac{TN}{FP+TN}$$

Clasificación Binaria: Matriz de confusión

- Matriz de confusión para dos clases:



$$\textit{TPRate, Sensitivity, Recall} = \frac{TP}{TP+FN}$$

Valor positivo predecible (PPV), *Precision* $= \frac{TP}{TP+FP}$

$$\textit{F-measure} = 2\frac{Precision \times Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN}$$

Problemas Multiclase

- Las tablas de confusión (contingencia) pueden ser de múltiples clases

ERROR		*actual*		
		low	medium	high
predicted	low	20	0	13
	medium	5	15	4
	high	4	7	60

- Las medidas basadas en arrays de 2 clases se computan
  - 1 vs todos (promedio de N medidas parciales)
  - 1 vs 1 (promedio N*(N-1)/2 medidas parciales)
    - ¿Peso promedio?

Datos no balanceados

- En algunos casos podemos encontrar diferencias significativas entre la proporción de clases
  - Un Clasificador Bayesiano Ingenuo que siempre son clases mayoritarias predictivas (ignorando las clases minoritarias) obtiene un buen rendimiento
    - En un problema binario (+/-) con un 1% de instancias negativas, el modelo ingenuo “siempre positivo” tiene una precisión del 99%.

- Macroprecisión: Precisión media por clase

$$macroacc(h) = \frac{\frac{Hits_{\text{class 1}}}{total_{\text{class 1}}} + \frac{hits_{\text{class 2}}}{total_{\text{class 2}}} + \ldots + \frac{hits_{\text{class m}}}{total_{\text{class m}}}}{m}$$

  - El Clasificador Bayesiano Ingenuo tiene una macroprecisión=0,5

Clasificadores Suaves

- Clasificadores Duros y Suaves:
  - Un clasificador “duro” o “*crisp*” predice una clase entre un conjunto de clases posibles
  - Un clasificador (probabilístico) “suave” o “*scoring*” predice una clase, pero acompaña cada predicción con una estimación de la fiabilidad de cada predicción.
    - La mayoría de métodos de aprendizaje pueden ser adaptados para generar este valor de fiabilidad.

Clasificadores suave: estimador de probabilidad

- Un tipo especial de clasificador ”suave” es un estimador probabilístico de clase.
  - En vez de predecir “a”, “b” o “c”, da una estimación probabilística para “a”, “b” o “c”, por ejemplo, “p_a”, “p_b” y “p_c”.
    - Ejemplo:
      - Clasificador 1: $p_a = 0.2$, $p_b = 0.5$ and $p_c = 0.3$.
      - Clasificador 2: $p_a = 0.3$, $p_b = 0.4$ and $p_c = 0.3$.
  - Ambos predicen “b”, pero el clasificador 1 es más confiable.

Evaluando clasificadores probabilísticos

- Clasificadores probabilísticos: Los clasificadores que son capaces de predecir una distribución probabilística por encima de un conjunto de clases
  - Proporcionar clasificación con un cierto grado de certeza:
    - Combinando clasificadores
    - Contextos sensibles al coste
- Error cuadrático medio(MSE o *Brier Score*)

$$MSE = \frac{1}{n}\sum_{i\in S}\sum_{j\in C}[f(i,j) - p(i,j)]^2$$

f(i,j)=1 si la instancia i es de la clase j, 0 de otro modo.
p(i,j) devuelve la estimación de i en la clase j

- *Log Loss*

$$Logloss = -\frac{1}{n}\sum_{i\in S}\sum_{j\in C}(f(i,j) * \log_2 p(i,j))$$

Evaluando clasificadores probabilísticos

- El MSE o el *Brier Score* pueden ser descompuestos en dos factores:
  - BS=CAL+REF
    - Calibración: Mide la calidad de las puntuaciones del clasificador con respecto a las probabilidades asignadas a cada clase.
    - Refinamiento: es una agregación de determinación e incertidumbre. Está relacionado con el área por debajo de la curva ROC.
- Métodos de calibración:
  - Intenta transformar las puntuaciones del clasificador en las probabilidades asignadas a cada clase
    - *Platt scaling, Isotonic Regression, PAVcal…*

Clasificadores suaves: *"rankers"*

- "*Rankers*":
  - Siempre que tengamos un estimador de probabilidad para un problema de dos clases :
    - $p_a = x$, then $p_b = 1 - x$.
  - Llamemos a una clase 0 (neg) y a la otra 1 (pos).
  - Un *ranker* es un clasificador suave que da un valor (puntuación) monótonamente relacionado con la probabilidad de clase 1.
- Podemos clasificar instancias de acuerdo con la probabilidad estimada
  - CRM: Estás interesado en el % de potenciales clientes
    - Ejemplos:
      - Clasificar al cliente según la probabilidad de que pueda comprar un producto.
      - Clasificar los mensajes de spam de menor a mayor probabilidad.

Evaluando *Rankers*

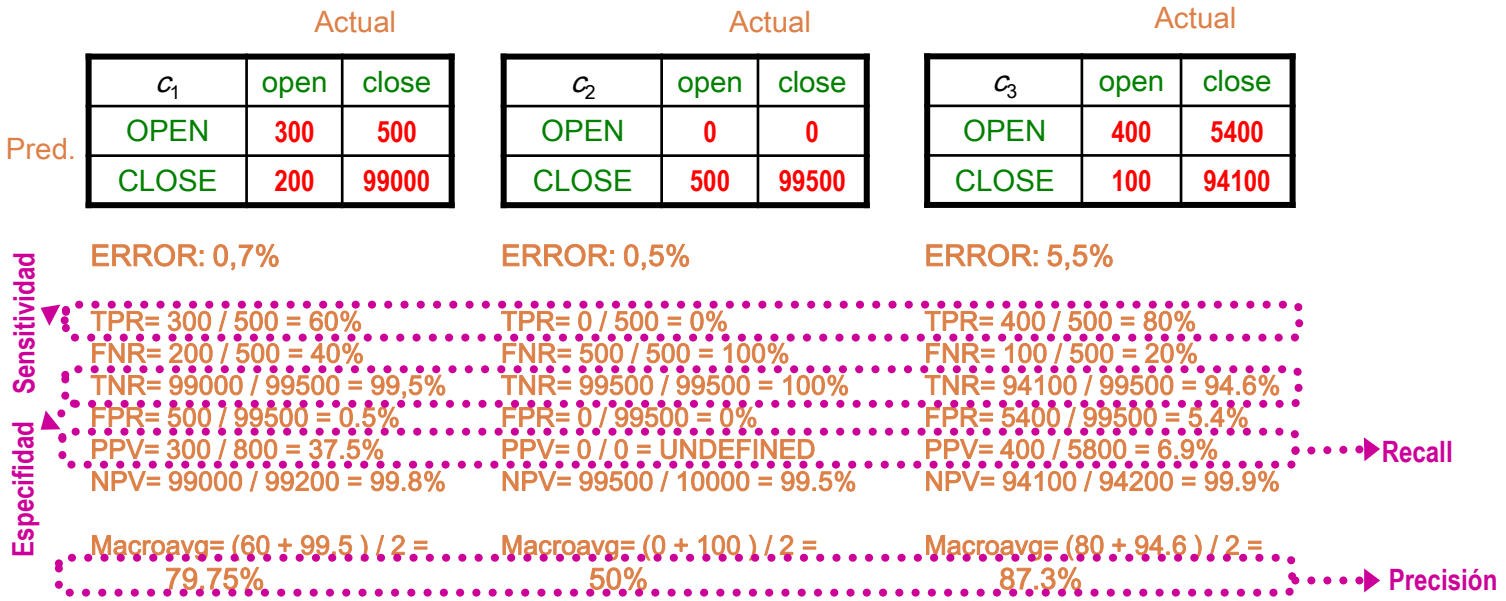
- Medidas para los clasificadores
  - AUC: Área bajo la curva ROC
    - Equivalente a la estadística Wilcoxon-Mann-Whitney, definida como:
      - "Dado un ejemplo positivo y un ejemplo negativo, la probabilidad de que el modelo clasifique el ejemplo positivo por encima del ejemplo negativo".
  - Otras: La distancia entre el *ranking* estimado y el *ranking* perfecto (si hubiese una noción de orden entre los valores).

Evaluación sensible al coste

- En clasificación, el modelo con la mayor *precision* no es necesariamente el mejor modelo.
  - Algunos errores (por ejemplo, falsos negativos) pueden ser más costosos que otros.
    - Esto está comúnmente (pero no siempre) asociado a conjuntos de datos desequilibrados.
    - Una matriz de costes es una forma simple de resolverlo.
- En regresión, el modelo con el menor error no es necesariamente el mejor modelo.
  - Algunos errores (por ejemplo, sobrepredicciones) pueden ser más costosos que otros (subpredicciones).
    - Una función de costes es una forma simple de resolverlo.

Evaluación sensible al coste

- Clasificación. Ejemplo: 100,000 instancias (solo 500 positivas)
  - Alto desequilibrio (π_0=Pos/(Pos+Neg)=0.005).



¿Cuál es el mejor clasificador?

Evaluación sensible al coste

- No todos los errores son iguales.
  - Ejemplo: Cerrar una válvula de una central nuclear cuando debería estar abierta puede producir una explosión, mientras que tenerla abierta cuando debería estar cerrada puede hacer que pare la central.
  - Matriz de coste:

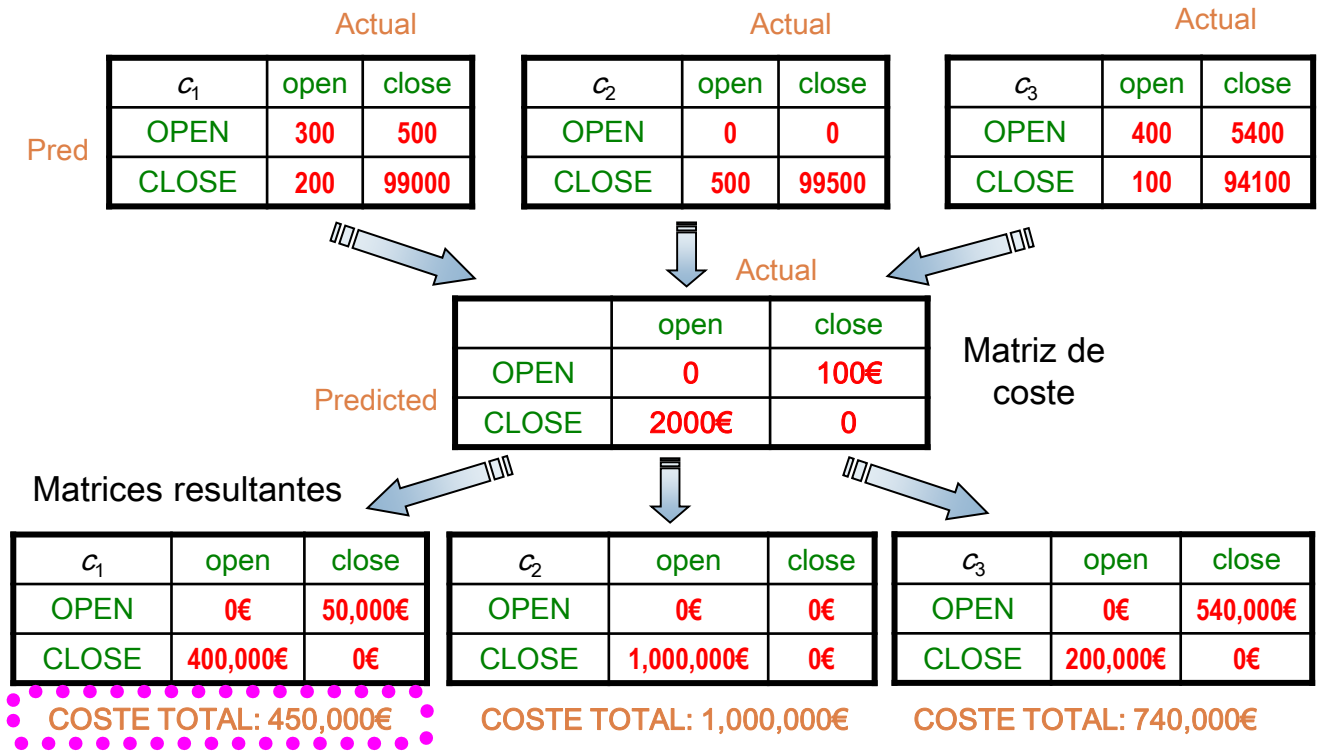
		Actual	
		open	close
Predicted	OPEN	0	100€
	CLOSE	2000€	0

  - El mejor clasificador es el de menor coste.

Evaluación sensible al coste

- De cálculo fácil(producto de Hadamard):

Matrices de confusión



Evaluación sensible al coste

- ¿Qué afecta al coste final?
  - Coste por unidad = costeFN*π_0*FNR + costeFP*(1-π_0)*FPR
  - Si dividimos por costeFN*π_0 tenemos M:
  - M = 1*FNR + costeFP*(1-π_0)/(costeFN*π0)*FPR = 1*FNR + *slope**FPR

$$\frac{FPcost}{FNcost} = \frac{100}{2000} = \frac{1}{20} \qquad \frac{Neg}{Pos} = \frac{99500}{500} = 199 \qquad slope = \frac{1}{20} \times 199 = 9.95$$

Clasif. 1: FNR= 40%, FPR= 0.5%
M1= 1 x 0.40 + **9.95** x 0.005 = 0.45
Coste por unidad =
M1*(FNCost*π_0)=4.5

Clasif. 2: FNR= 100%, FPR= 0%
M2= 1 x 1 + **9.95** x 0 = 1
Coste por unidad =
M2*(FNCost*π_0)=10

Clasif. 3: FNR= 20%, FPR= 5.4%
M3= 1 x 0.20 + **9.95** x 0.054 = 0.74
Coste por unidad =
M3*(FNCost*π_0)=7.4

Para dos clases, el valor "*slope*" (pendiente, con FNR y FPR) es suficiente para decir qué clasificador es mejor. A esto se le llama **condición operativa, contexto o sesgo.**