

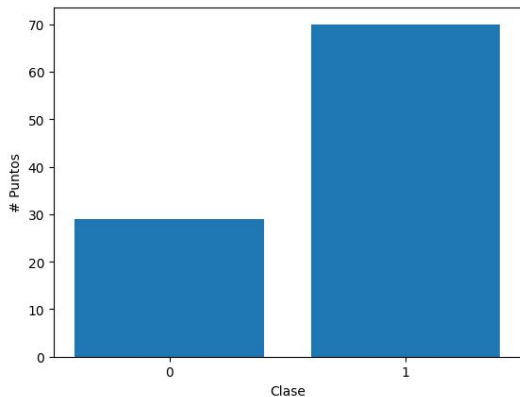
IAA-2023

Clase 10: Métricas de Clasificación



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Repaso: Clasificación Desbalanceada



Clasifico a todos como clase 1:

Exactitud = 70%

Nuevas Métricas:

$$\text{precision} = \frac{TP}{TP + FP} = \frac{\text{fish}}{\text{fish} + \text{shell}}$$

$$\text{recall} = \frac{TP}{TP + FN} = \frac{\text{fish}}{\text{fish} + \text{fish}}$$

- F1-score: $\frac{1}{F_1} = \frac{1}{2} \left(\frac{1}{\text{precision}} + \frac{1}{\text{exhaustividad}} \right)$

$$F_1 = \frac{2 \text{ precision} \cdot \text{exhaustividad}}{\text{precision} + \text{exhaustividad}}$$

Predicciones

	0	1
Target Real	$\begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$	

Repaso: Clasificación Desbalanceada

Estrategias para lidiar con un dataset desbalanceado:

- Usar métricas adecuadas (nada de exactitud)
- Compensar el sesgo natural a la clase mayoritaria:
 - Dar mayor peso en la loss function a la clase minoritaria (`class_weights`)
 - Sobre-sampear la clase minoritaria (`sklearn.utils.resample`)
- Ajustar el umbral para maximizar la métrica de interés
(más sobre esto la clase próxima)

Repaso: Clasificación MultiClase

- **Uno-vs-el resto**

Si tengo K clases, entreno K clasificadores: Cada uno clasifica si un sample es de la clase K o no. Luego clasifico con la clase de mayor probabilidad (y puedo normalizar las probabilidades para obtener una probabilidad multiclase)

Tanto la exhaustividad como la precisión se definen una para cada clase.

		Predicciones				
		0	1	2	...	K
Target Real	0	$T0$	$F1$	$F2$	\dots	FK
	1	$F0$	$T1$	$F2$	\dots	FK
	2	$F0$	$F1$	$T2$	\dots	FK
	...	\vdots	\vdots	\vdots	\ddots	\vdots
	K	$F0$	$F1$	$F2$	\dots	TK

Métricas de Clasificación Binaria

Matriz de confusión:

$$\begin{array}{c} \text{Target} \\ \text{Real} \end{array} \begin{array}{cc} \text{Predicciones} \\ 0 & 1 \end{array} \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$$

$$precision = \frac{TP}{TP + FP}$$

$$exhaustividad = TPR = \frac{TP}{TP + FN}$$

$$TNR = \frac{FP}{FP + TN}$$

Métricas de Clasificación Binaria

Matriz de confusión:

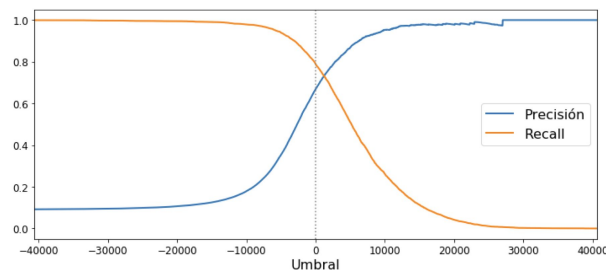
$$\begin{array}{c} \text{Target Real} \\ \begin{array}{c} 0 \\ 1 \end{array} \end{array} \begin{array}{c} \text{Predicciones} \\ \begin{array}{c} 0 \\ 1 \end{array} \end{array} \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$$

$$precision = \frac{TP}{TP + FP}$$

$$exhaustividad = TPR = \frac{TP}{TP + FN}$$

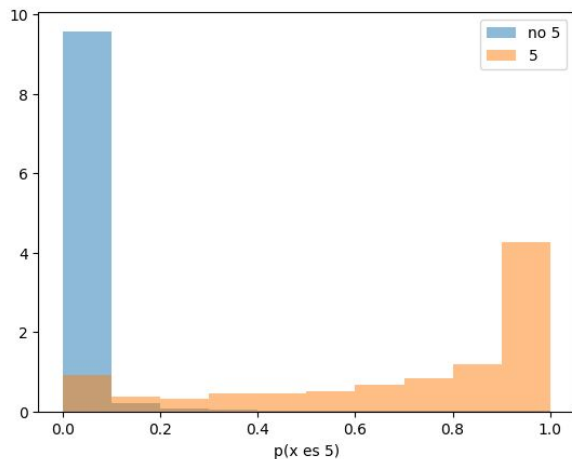
$$TNR = \frac{FP}{FP + TN}$$

Depende del umbral de decisión



Dependencia con el threshold

Los números en la matriz de confusión varían dependiendo del threshold que se use:

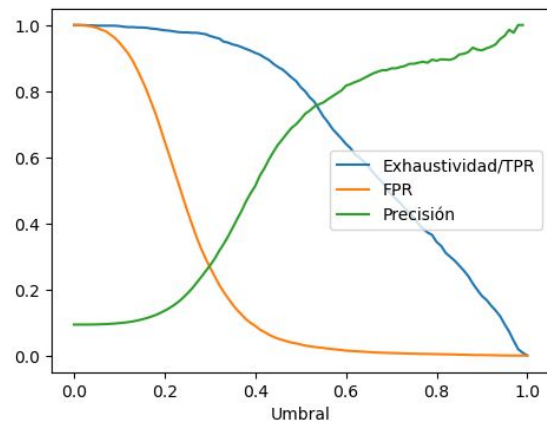


Uno incluso podría elegir el threshold para maximizar una métrica particular (como un hiper-parámetro más).

Dependencia con el threshold

Algo interesante para analizar y comparar modelos, es calcular las métricas para *todos los umbrales posibles* (es decir, $N-1$ umbrales cuanto mucho).

Luego, podemos ver como varían cada uno con el umbral:

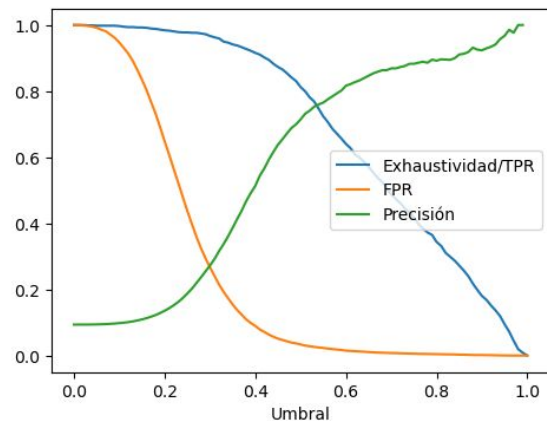
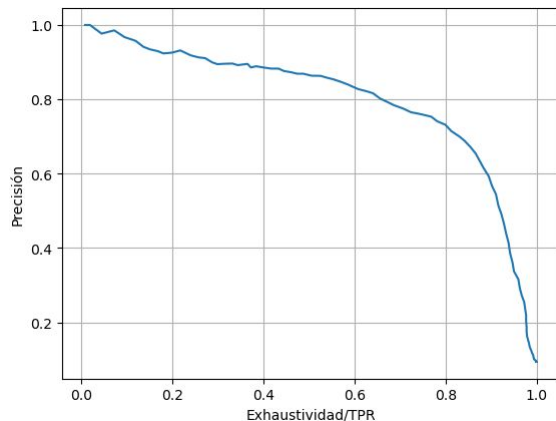


Dependencia con el threshold

Algo interesante para analizar y comparar modelos, es calcular las métricas para *todos los umbrales posibles* (es decir, N-1 umbrales cuanto mucho).

Luego, podemos ver como varían cada uno con el umbral:

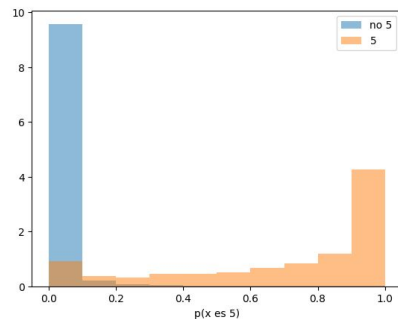
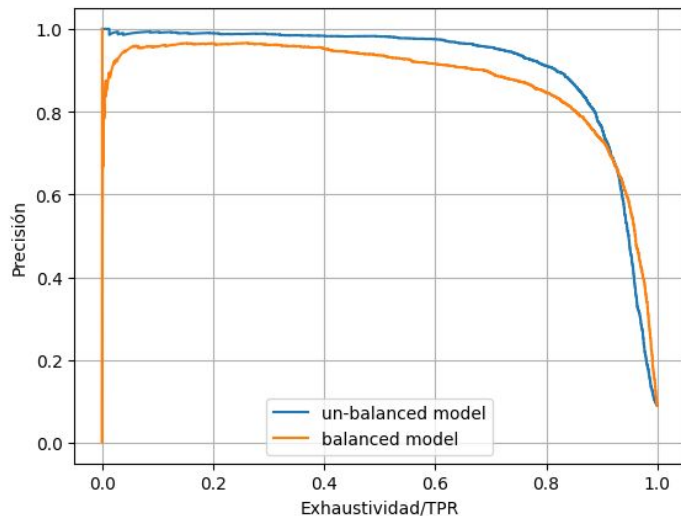
O incluso una métrica en función de la otra:



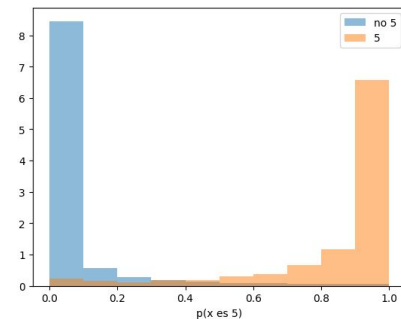
Comparación de modelos

Distintos modelos / hiperparámetros,
darán lugar a distintas distribuciones de probabilidad:

Lo que se traduce en distintas curvas



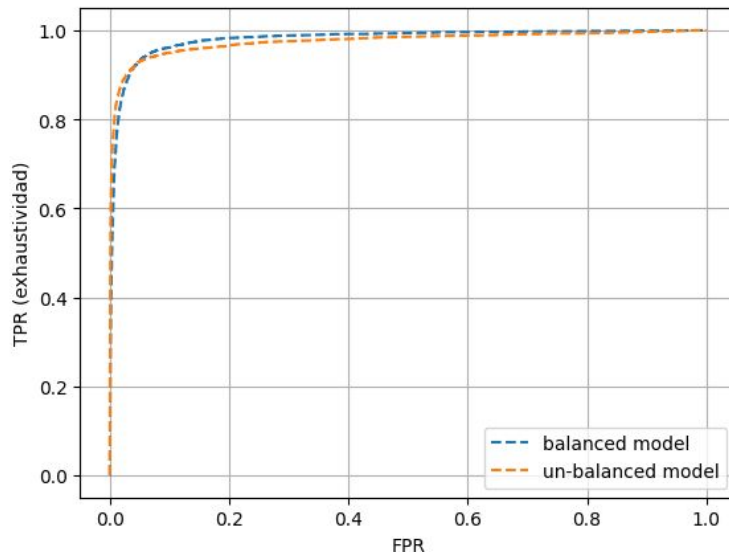
un-balanced



balanced

ROC, Precision-Recall, AUC

ROC: *Receiving Operating Characteristic* = TPR vs FPR (aka recall)



Para comparar modelos, independientemente del umbral, se suele usar como métrica el **area bajo la curva** (AUC) de esta curva.

Útil cuando nos interesa **ordenarlos** por probabilidad.