DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Aprendizaje Automático

1C-2025

Clase 5:

Métricas Regresión
Métricas Clasificación
Evaluación de clasificadores probabilísticos.
Teoría de la decisión

Métricas para regresión (brevísima intro)

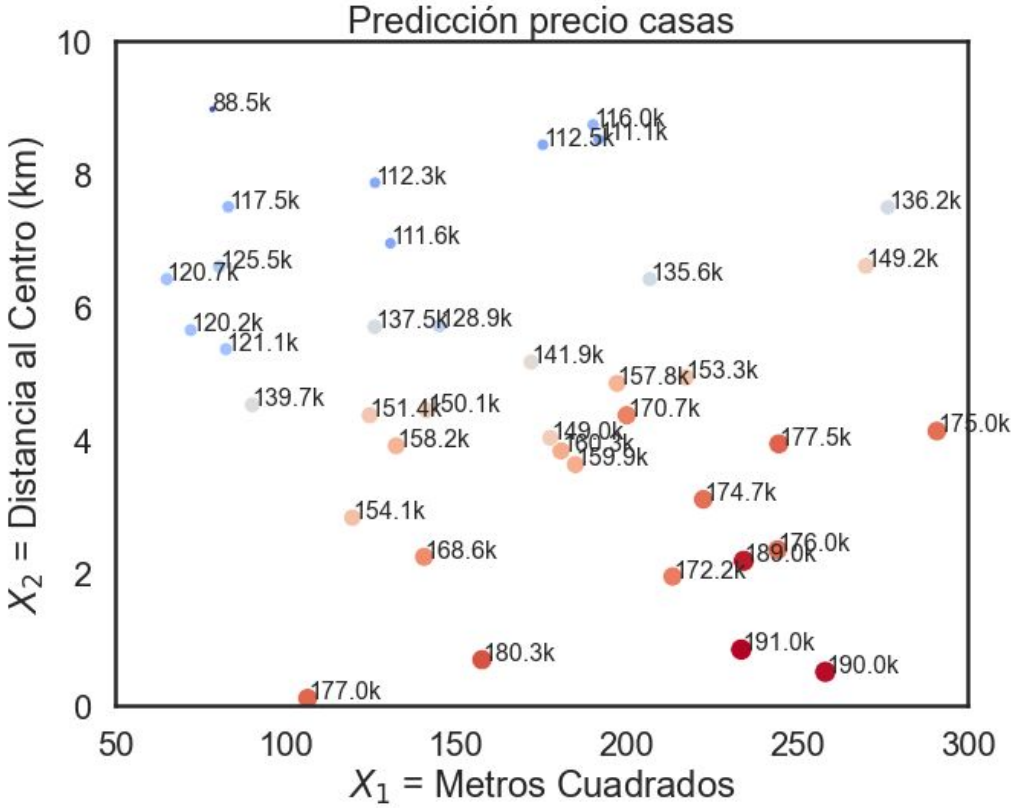
¿Cómo medimos la performance de un regresor?

Dado un conjunto de datos etiquetados, y predicciones para esos datos (en cross validation por ejemplo).

¿Cómo calculamos el error cometido?

¿Qué opinan de?

- ***0*** error si le pega al valor de la casa.
- ***1*** si no le pega.



¿Cómo medimos la performance de un regresor?

Parte I

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y^{(i)} - h(x^{(i)})\right|$$

Proporciona una idea del error promedio **en las unidades del target.**

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - h(x^{(i)})\right)^2$$

O "Error cuadrático medio". **Más sensible a valores atípicos (outliers)**. ¿Por qué?

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - h(x^{(i)})\right)^2}$$

Como MSE, **sensible a outliers,** pero el error **se mantiene en la escala** de los valores de las etiquetas.

¿Cómo medimos la performance de un regresor?

Parte II

$$\text{Correlación (Pearson)} = r = \frac{\sum_{i=1}^{n} \left(y^{(i)} - \bar{y}\right)\left(h(x^{(i)}) - \overline{h(x)}\right)}{\sqrt{\sum_{i=1}^{n} \left(y^{(i)} - \bar{y}\right)^2 \sum_{i=1}^{n} \left(h(x^{(i)}) - \overline{h(x)}\right)^2}}$$

¿**Correlación lineal** como métrica?

- ¿Qué pasa si sumamos una constante a cada predicción de $h(x^{(i)})$?
- ¿Qué pasa si multiplicamos por una constante a cada predicción de $h(x^{(i)})$?
- ¿Cuánto mejor es $r = 0.7$ es que $r = 0.5$? (rta: 2 veces mejor)

No descartarla, a veces se utiliza cuando no nos importa que haya problemas de escala o suma de constantes (ejemplo: señales de cerebro, EEG).

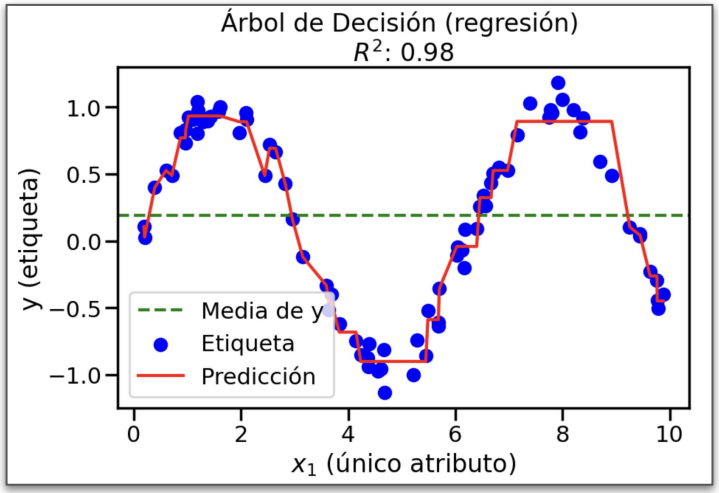
Ojo también si el regresor utilizado no es lineal (ver otros tipos de correlaciones).

¿Cómo medimos la performance de un regresor?

Parte III

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y^{(i)} - h(x^{(i)})\right)^2}{\sum_{i=1}^{n} \left(y^{(i)} - \bar{y}\right)^2}$$

R^2 "Coeficiente de determinación": **¿qué tanto mejor ajusta mi modelo que utilizar la media como predicción?**



$0 \leq R^2 \leq 1$ (¿Cuándo da 0? ¿Cuándo 1?)

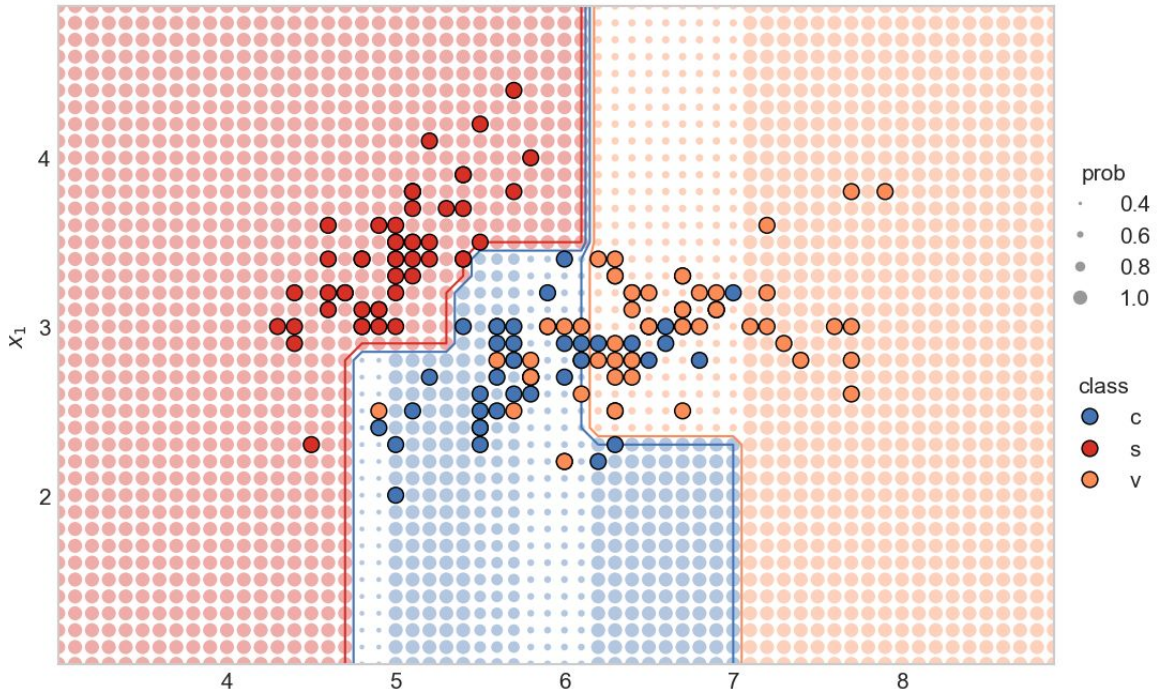
Ventaja 1: $R^2 = 0.7$ es **1.4** veces mejor que $R^2 = 0.5$ (como uno esperaría). A diferencia de $r = 0.7$ vs $r = 0.5$ (en donde es dos veces mejor).

Ventaja 2: a diferencia de **MSE** o **MAE**, R^2 permite comparar modelos en distintos datasets con distintas unidades. Desventaja obvia: se pierden las unidades.

Clasificación

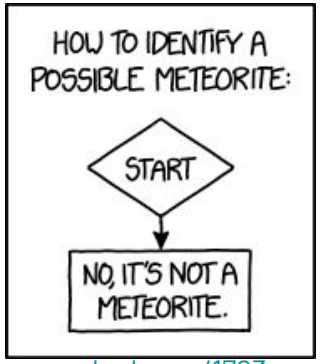
¿Cómo medimos la performance de un clasificador?

1. ¿Qué tan buenas son las **asignaciones** a clases?
2. ¿Qué tan buenas son las **probabilidades** que se asignan?
3. ¿Funciona de igual manera **para cada clase**?
4. ¿Cómo hago un reporte con **un solo valor** que resuma la performance de mi clasificador?



Ejemplo de fronteras de decisión de un árbol de altura máxima 4.
Código para generar estos gráficos:
https://www.tvhahn.com/posts/beautiful-plots-decision-boundary/

Medidas de Performance



xkcd.com/1723

Un modelo tiene una *accuracy* (eficacia) del 99%.

- O sea, de cada 100 instancias, clasifica bien 99.

¿Qué significa esto?

- Según la tarea y la distribución de clases en el dominio, 99% puede ser muy bueno o pésimo.
- No dice nada sobre el tipo de aciertos y errores que comete el modelo.
- Ejemplos:
  - Un filtro de spam que marca todos los emails como spam.
  - Un detección de fraude que siempre dice “no fraude”
  - Un Identificación de meteoritos que siempre dice “NO”
- Nos interesa medir mejor, poder también asignar distintos costos a cada tipo de error y también poder tomar en cuenta los “scores” que devuelve un sistema.

Medidas de Performance

Caso binario: Matriz de confusión

- *TP: True Positives* (verdaderos positivos)
- *TN: True Negatives* (verdaderos negativos)
- *FP: False Positives* (falsos positivos)
- *FN: False Negatives* (falsos negativos)

Caso binario: una clase "claramente" **POSITIVA** y una clase "claramente" **NEGATIVA**.

Ejemplo: Detección COVID.

```
COVID? :: Atributos -> {Si, No}
```

Ejemplo resultado (sobre un val set):

Instancia	1	2	3	4	5	6
Clase real	*Sí*	*Sí*	*No*	*Sí*	*Sí*	*No*
Predicha	*Sí*	*No*	*Sí*	*Sí*	*No*	*No*
Resultado	***OK***	*Err*	*Err*	***OK***	*Err*	***OK***
Resultado Detallado	***TP***	***FN***	***FP***	***TP***	***FN***	***TN***

Matriz de Confusión		**Clase Predicha**	
	Total (P+N) 3841	COVID (PP) 2743	NO COVID (PN) 1098
Clase Real	COVID (P) 2795	2739 ***TP***	56 ***FN***
	NO COVID (N) 1046	4 ***FP***	1042 ***TN***

Para los siguientes sistemas,
¿qué tipo de error ***FP*** vs ***FN*** parece más dañino?

1. **Filtro de spam:** que descartará directamente los emails sospechosos.
2. **Detección de fraude:** prepara un listado de casos sospechosos para ser revisados por humanos.

Medidas de Performance

Caso binario: Precision y Recall

- *TP: True Positives* (verdaderos positivos)
- *TN: True Negatives* (verdaderos negativos)
- *FP: False Positives* (falsos positivos)
- *FN: False Negatives* (falsos negativos)

Precision: De las instancias predichas como positivas ¿qué porcentaje lo eran?
También llamado Positive Predictive Value (PPV)

Recall: De las instancias positivas, ¿qué porcentaje fueron predichas como tal?
También llamado Sensitivity, Hit Rate o True Positive Rate **(TPR)**

Matriz de Confusión		**Clase Predicha**	
	Total (P+N) 3841	COVID (PP) 2743	NO COVID (PN) 1098
Clase Real	COVID (P) 2795	2739 ***TP***	56 ***FN***
	NO COVID (N) 1046	4 ***FP***	1042 ***TN***

La precisión y el recall pueden interpretarse como probabilidades condicionales (estimadas)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \mathbb{P}(C = P | \hat{C} = P)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \mathbb{P}(\hat{C} = P | C = P)$$

Para los siguientes sistemas,
¿qué métrica priorizararían? ¿**Precision** o **Recall**?

1. **Filtro de spam:** que descartará directamente los emails sospechosos.
2. **Detección de fraude:** prepara un listado de casos sospechosos para ser revisados por humanos.

Medidas de Performance

Caso binario: F-score

Precision (de los que dije positivo, cuántos lo eran):
$TP \ / \ (TP + FP)$
Recall (de los positivos, cuántos acerté)
$TP \ / \ (TP + FN)$

Vemos que alto **Recall** y alto **Precision** son dos características deseables del sistema. Pero un sistema con **Recall** 100% y **Precision** 0% o viceversa, no es un buen sistema.
¿ $Promedio = (recall + precision)/2$? → Mala idea (googleen por qué).

En general se utiliza: F_β (el "F-score")

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

F_1 es lo que se conoce como **media armónica.**

Efecto: los valores más bajos se penalizan proporcionalmente al inverso de su magnitud: cuanto más bajo sea el valor, mayor será la penalización.

Para los siguientes sistemas,
¿qué métrica usarían entre $F_{0.5}$ y F_2 ?

1. **Filtro de spam:** que descartará directamente los emails sospechosos.
2. **Detección de fraude:** prepara un listado de casos sospechosos para ser revisados por humanos.

Medidas de Performance

Caso binario: F-score

Precision (de los que dije positivo, cuántos lo eran):
$TP \ / \ (TP + FP)$
Recall (de los positivos, cuántos acerté)
$TP \ / \ (TP + FN)$

Vemos que alto **Recall** y alto **Precision** son dos características deseables del sistema. Pero un sistema con **Recall** 100% y **Precision** 0% o viceversa, no es un buen sistema.
¿ $Promedio = (recall + precision)/2$? → Mala idea (googleen por qué).

En general se utiliza: F_β (el "F-score")

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

F_1 es lo que se conoce como **media armónica.**

Efecto: los valores más bajos se penalizan proporcionalmente al inverso de su magnitud: cuanto más bajo sea el valor, mayor será la penalización.

Imaginen el siguiente desbalanceo: **80% clase positiva (COVID), 20% clase negativa (NO-COVID)**

Ahora imaginen un clasificador "dummy"

$h\_dummy(x) = clase\_mayoritaria(Y)$
$F_{1(h\_dummy)} =$ **<calcular>**

Ahora imaginen el mismo ejemplo, pero llamemos clase negativa a COVID, clase positiva a NO-COVID.

$h\_dummy = clase\_mayoritaria(Y)$
$F_{1(h\_dummy)} =$ **<calcular>**

Medidas de Performance

Caso multiclase

Matriz de Confusión (multiclase)		Clase Predicha		
		Perro	Gato	Loro
Clase Real	Perro	$TP_{(C=perro)}$	X_2	
	Gato	X_1	$TP_{(C=gato)}$	
	Loro			$TP_{(C=pato)}$

X_1 = ***FP*** para Perro, ***FN*** para Gato, ***TN*** para Loro.
X_2 = ***FN*** para Perro, ***FP*** para Gato, ***TN*** para Loro.

Calculamos las métricas por clase:

$$\text{Precision}_{(C=c)} = \frac{\text{TP}_{(C=c)}}{\text{TP}_{(C=c)} + \text{FP}_{(C=c)}}$$

$$\text{Recall}_{(C=c)} = \frac{\text{TP}_{(C=c)}}{\text{TP}_{(C=c)} + \text{FN}_{(C=c)}}$$

$$\text{F-1 Score}_{(C=c)} = \frac{2 \times \text{Precision}_{(C=c)} \times \text{Recall}_{(C=c)}}{\text{Precision}_{(C=c)} + \text{Recall}_{(C=c)}}$$

¿Puede obtener un sólo número para la métrica M?

a) Macro average:
$\frac{1}{3} * (M(C=p) + M(C=g) + M(C=l))$

b) "Weighted average":
$freq(p).M(C=p) + freq(g).M(C=g) + freq(l).M(C=l)$

c) "Micro average" (juntamos antes de calcular nada)
TP $= TP(C=p) + TP(C=g) + TP(C=l)$
FP $= FP(C=p) + FP(C=g) + FP(C=l)$
FN $= FN(C=p) + FN(C=g) + FN(C=l)$
TN $= TN(C=p) + TN(C=g) + TN(C=l)$

Notas hasta acá

Algunas críticas

(fuente https://en.wikipedia.org/wiki/F-score)

- **F1** le da **igual importancia al precisión y al recall.**
- **F1 es menos veraz e informativa que** el coeficiente de correlación de Matthews (MCC) en la evaluación de clasificaciones binarias.
- **F1 ignora los verdaderos negativos** y, por lo tanto, es engañosa para clases desbalanceadas.
- **F1 no es simetríca.** Esto significa que puede cambiar su valor cuando se cambian las etiquetas del conjunto de datos, es decir, las muestras "positivas" se nombran "negativas" y viceversa.
- **Complejidad en ajuste de umbral**. Más sobre esto a continuación.

Opiniones personales:

Hay una costumbre muy grande de dejarse llevar por las métricas que "usa la mayoría" (o que vienen usando en los papers de tal rama).

En ciertas áreas de machine learning **(ej: análisis de emociones)** las métricas que se utilizan no tienen sentido (o al menos favorecen sistemas que no aprendieron tan bien como otros).

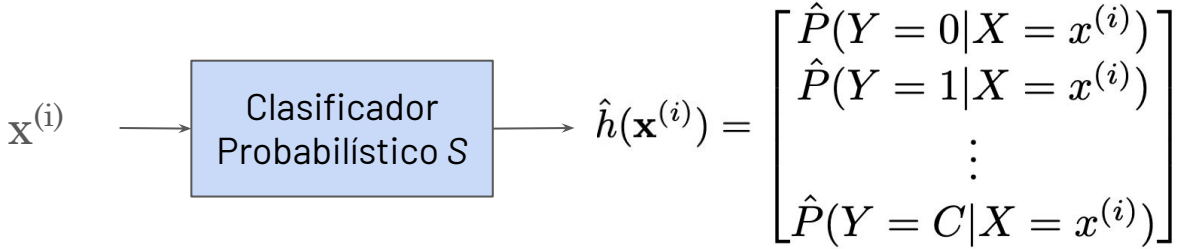
Es importante **ser críticos** en este tema y tomarse una gran parte del tiempo del desarrollo del sistema para elegir una métrica correcta.

Ver por ejemplo: https://www.nature.com/articles/s41592-023-02150-0

Evaluación de clasificadores probabilísticos

Clasificadores probabilísticos

Un **clasificador probabilístico** es un tipo de clasificador que produce puntuaciones que (intentan ser (*)) probabilidades a posteriori para las clases dada la entrada.



La mayoría de los métodos de clasificación pertenecen a esta categoría (**).
Ej, en árboles, observando la frecuencia por clase en la hoja asignada a la instancia $\mathbf{x}$.

Para el caso binario: $\hat{h}(\mathbf{x}^{(i)}) = \hat{P}(Y=1|X=\mathbf{x}^{(i)}) = 1 - \hat{P}(Y=0|X=\mathbf{x}^{(i)})$

(*) "Intentan ser" tiene que ver con el problema de ***calibración de modelos*** (no lo veremos en la materia)

(**) Lo que diremos a continuación también aplica a SVM por ejemplo, que outputea "**scores**" y no probas.

Clasificadores probabilísticos

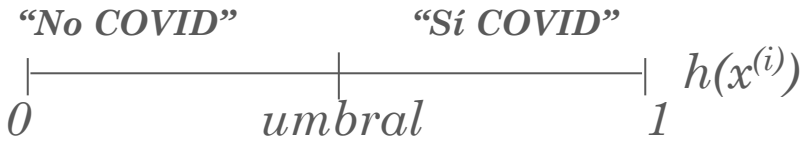
Umbrales de decisión

Volviendo al ejemplo de COVID -> {Sí, No}

Instancia	1	2	3	4	5	6
Clase real	*Sí*	*Sí*	*No*	*Sí*	*Sí*	*No*
Predicha	*Sí*	*No*	*Sí*	*Sí*	*No*	*No*
Resultado	***OK***	*Err*	*Err*	***OK***	*Err*	***OK***
Resultado Detallado	***TP***	***FN***	***FP***	***TP***	***FN***	***TN***

Podemos pensar que esta tabla (y consecuentemente la **matriz de confusión**) se generó de la siguiente manera:

a) Utilizamos el modelo **M** que prediga cada instancia.
b) M devolvió **puntajes** (probas) entre 0 y 1.
c) Decidimos etiquetar según el **umbral** ***0.5***.
$\textbf{\textit{Predicción}}(x^{(i)}) =$ *"Sí"* si $h(x^{(i)}) > \mathbf{0.5}$ *"No"* si no.



Qué ocurre si cambiamos el umbral:

$\textbf{\textit{Predicción}}(x^{(i)}) =$ *"Sí"* si $h(x^{(i)}) > \mathbf{0.3}$ *"No"* si no.

Instancia	1	2	3	4	5	6
Clase real	*Sí*	*Sí*	*No*	*Sí*	*Sí*	*No*
Predicha	??	??	??	??	??	??
Resultado	??	??	??	??	??	??
Resultado Detallado	??	??	??	??	??	??

Clasificadores probabilísticos

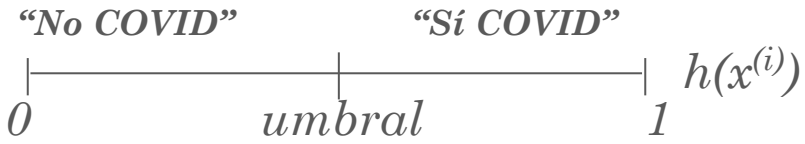
Umbrales de decisión

Volviendo al ejemplo de COVID -> {Sí, No}

Instancia	1	2	3	4	5	6
Clase real	*Sí*	*Sí*	*No*	*Sí*	*Sí*	*No*
Predicha	*Sí*	*No*	*Sí*	*Sí*	*No*	*No*
Resultado	***OK***	*Err*	*Err*	***OK***	*Err*	***OK***
Resultado Detallado	***TP***	***FN***	***FP***	***TP***	***FN***	***TN***

Podemos pensar que esta tabla (y consecuentemente la **matriz de confusión**) se generó de la siguiente manera:

a) Utilizamos el modelo **M** que prediga cada instancia.
b) M devolvió **puntajes** (probas) entre 0 y 1.
c) Decidimos etiquetar según el **umbral** ***0.5***.
$\textbf{\textit{Predicción}}(x^{(i)})$ = *"Sí"* si $h(x^{(i)}) > \mathbf{0.5}$ *"No"* si no.



Qué ocurre si cambiamos el umbral:

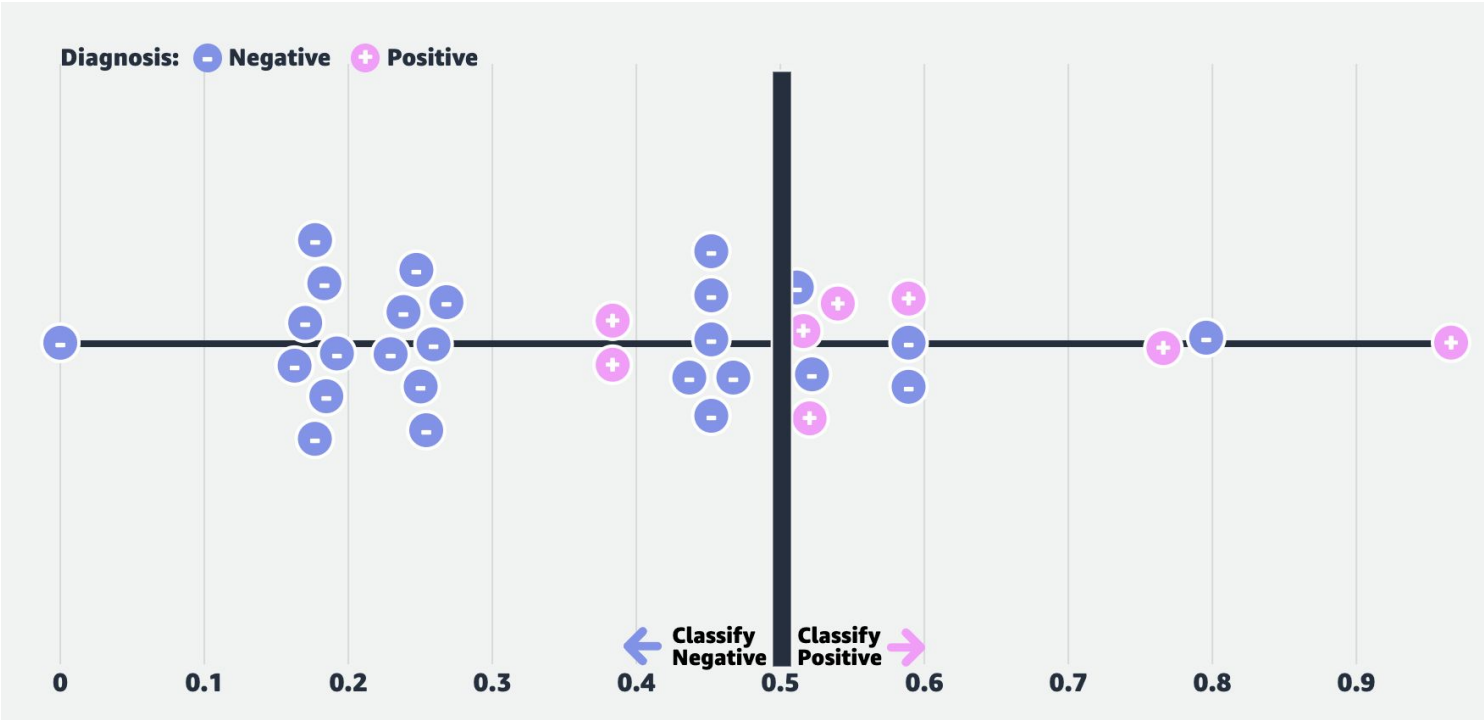
$\textbf{\textit{Predicción}}(x^{(i)})$ = *"Sí"* si $h(x^{(i)}) > \mathbf{0.3}$ *"No"* si no.

Instancia	1	2	3	4	5	6
Clase real	*Sí*	*Sí*	*No*	*Sí*	*Sí*	*No*
Predicha	*Sí*	***Sí***	*Sí*	*Sí*	*No*	***Sí***
Resultado	***OK***	***OK***	*Err*	***OK***	*Err*	*Err*
Resultado Detallado	***TP***	***TP***	***FP***	***TP***	***FN***	***FP***

¡Habrá una tabla de confusión (potencialmente) distinta por cada umbral!

Clasificadores probabilísticos

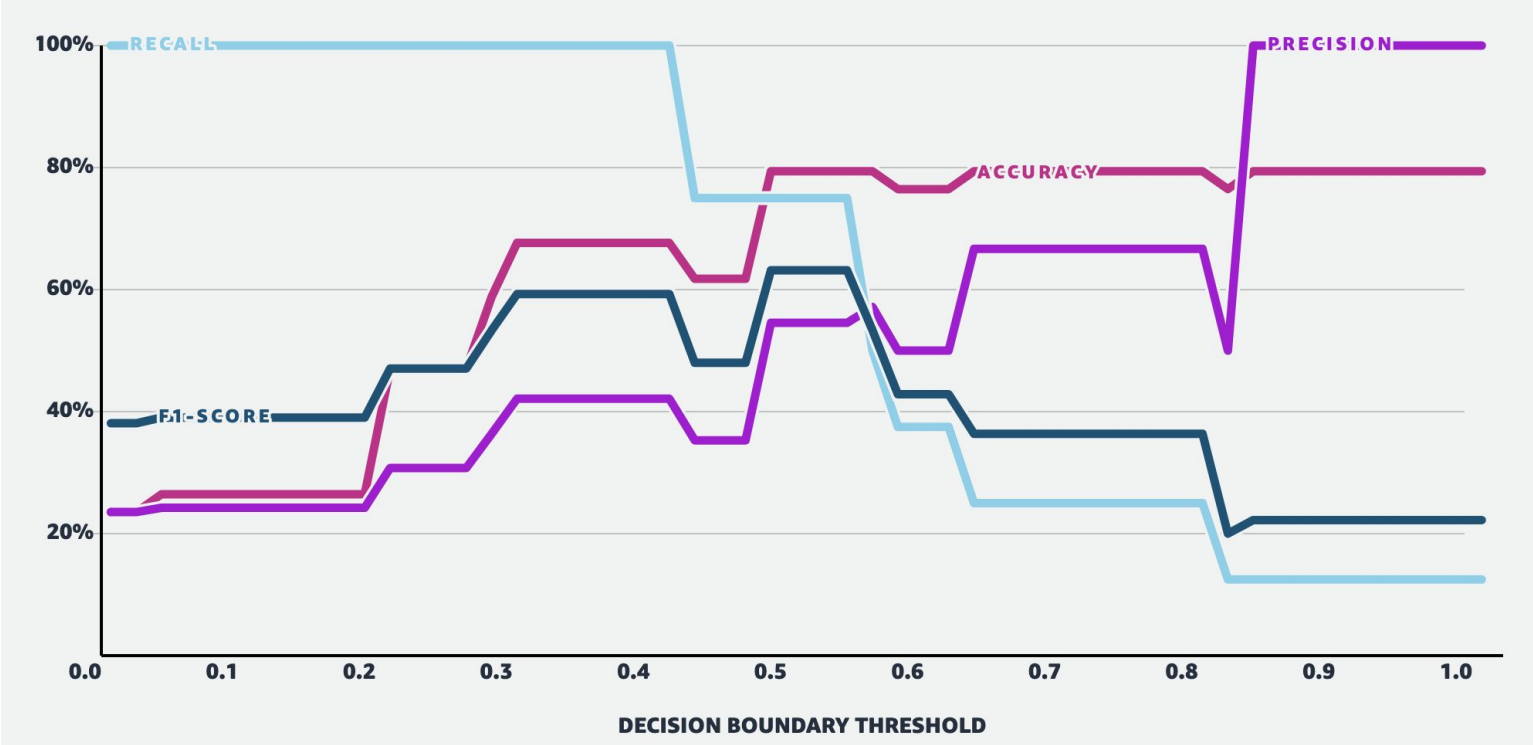
Umbrales de decisión



Fuente: https://mlu-explain.github.io/precision-recall/

Clasificadores probabilísticos

Umbrales de decisión



Fuente: https://mlu-explain.github.io/precision-recall/

Clasificadores probabilísticos

Curvas ROC

Para comparar entre dos clasificadores. No suele interesar tanto la etiqueta final, sino la asignación de puntajes a cada instancia y poder medir **qué tan bien las ordena.** Luego es cuestión de **definir un buen umbral** (que refleje el caso de uso del sistema)

Podemos entonces procrastinar esa decisión.

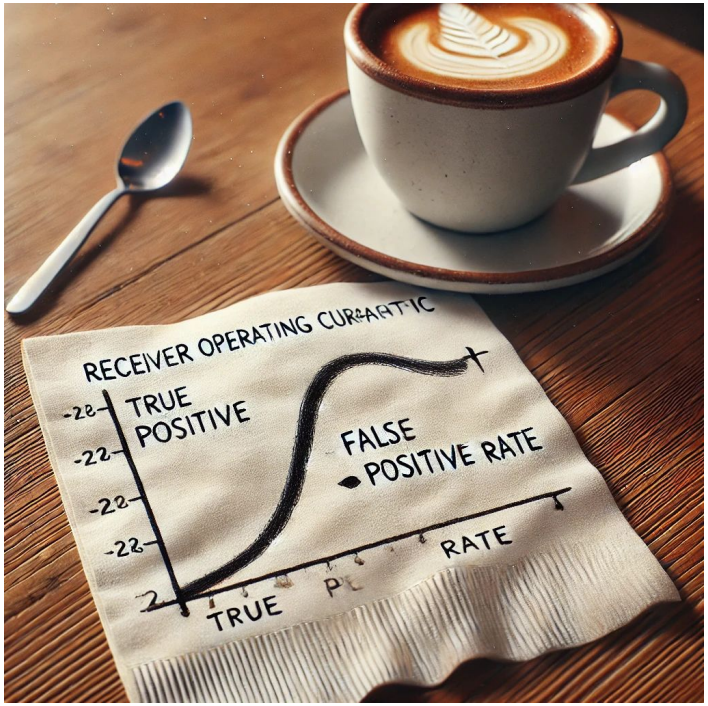
Idea curvas ROC:
¿Qué obtengo al recorrer todos los umbrales?

- *"Sí"* si $h(x^{(i)}) > \mathbf{0.01}$ *"No"* si no.
- *"Sí"* si $h(x^{(i)}) > \mathbf{0.02}$ *"No"* si no.

...

- *"Sí"* si $h(x^{(i)}) > \mathbf{0.99}$ *"No"* si no.

(en el ejemplo suponemos puntajes entre 0 y 1,
aunque funciona de igual para cualquier rango de puntajes)



"A ROC curve drawn on a napkin next to a coffee"

Clasificadores probabilísticos

Curvas ROC

Para comparar entre dos clasificadores. No suele interesar tanto la etiqueta final, sino la asignación de puntajes a cada instancia y poder medir **qué tan bien las ordena.** Luego es cuestión de **definir un buen umbral** (que refleje el caso de uso del sistema)

Podemos entonces procrastinar esa decisión.

Idea curvas ROC:
¿Qué obtengo al recorrer todos los umbrales?

- "Sí" si $h(x^{(i)}) > \mathbf{0.01}$ "No" si no.
- "Sí" si $h(x^{(i)}) > \mathbf{0.02}$ "No" si no.

...

- "Sí" si $h(x^{(i)}) > \mathbf{0.99}$ "No" si no.

(en el ejemplo suponemos puntajes entre 0 y 1,
aunque funciona de igual para cualquier rango de puntajes)

Método:

Ordenar las instancias según su predicción por el modelo. Considerar como umbral cada una de estos scores.

Ej, dadas los siguiente puntajes dados por el modelo M:

$h(x^{(1)}) = 0.2$
$h(x^{(2)}) = 0.1$
$h(x^{(3)}) = 0.5$
$h(x^{(4)}) = 0.9$
$h(x^{(5)}) = 0.8$

Umbrales a considerar:
μ = 0.1, 0.2, 0.5, 0.8, 0.9

Obtenemos así **5 matrices de confusión**

Clasificadores probabilísticos

Curvas ROC

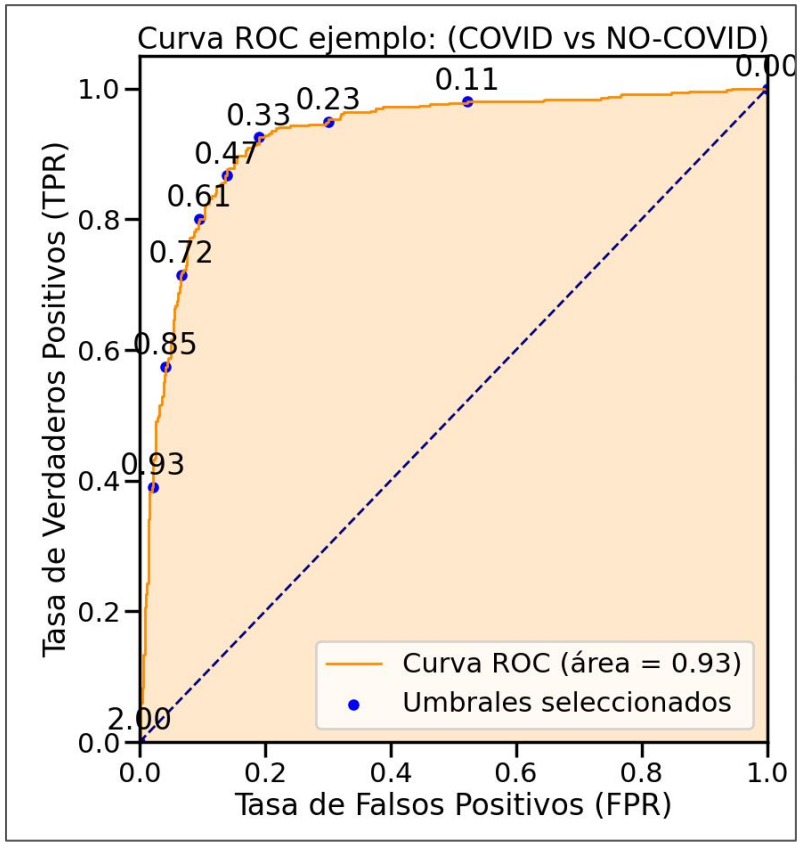
Para cada Matriz de Confusión computamos:

1. **TPR** "True Positive Rate"
   Tasa de verdaderos positivos
   = *TP* / (*TP* + *FN*)
   (ya la vimos, "Recall")
   (mundo medicina: Sensibilidad)

2. **FPR** "False Positive Rate"
   Tasa de falsos positivos
   = *FP* / (*FP* + *TN*)
   (Probabilidad de una falsa alarma)
   (mundo medicina: 1 - Especificidad)

"ROC" = "Receiver Operating Characteristic"
"AUC" = "Area Under the Curve"

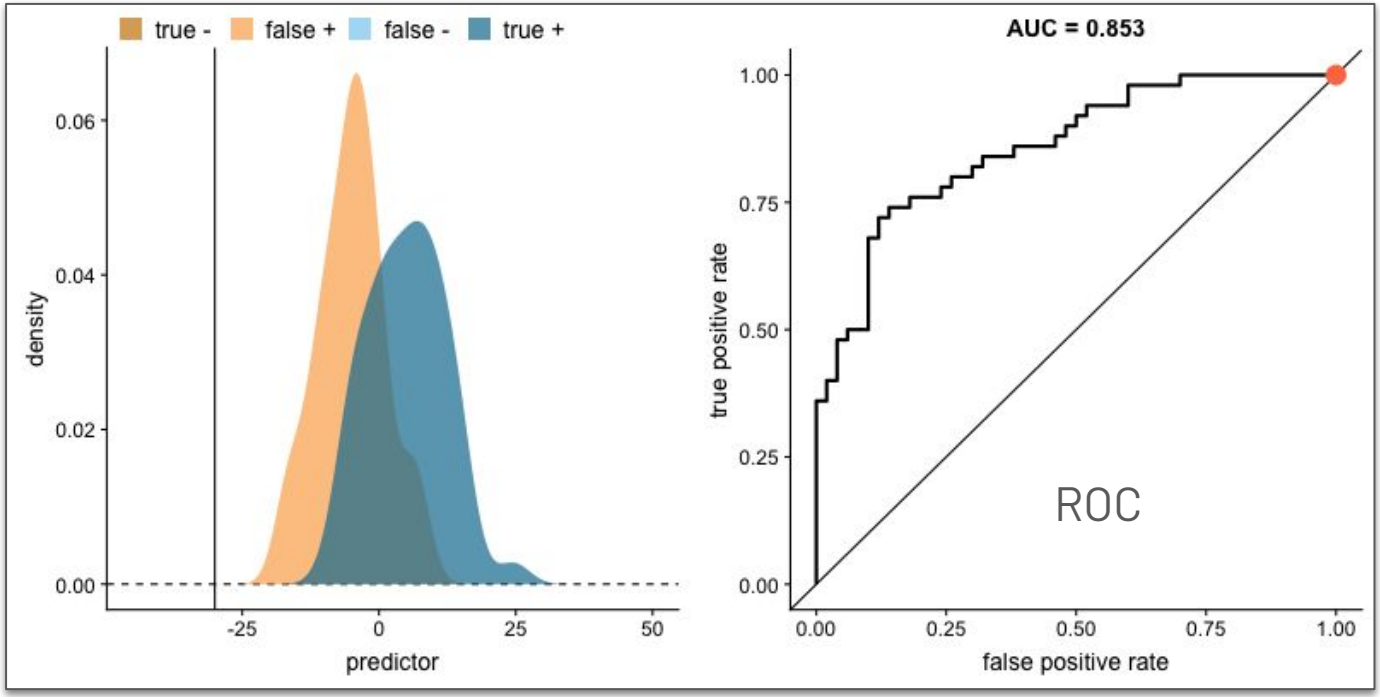
Métrica "AUC-ROC"



Clasificadores probabilísticos

Curvas ROC

Para discutir: **¿Qué muestra este gráfico? ¿Cómo se genera el de la izquierda?**



Fuente: https://paulvanderlaken.com/2019/08/16/roc-auc-precision-and-recall-visually-explained/

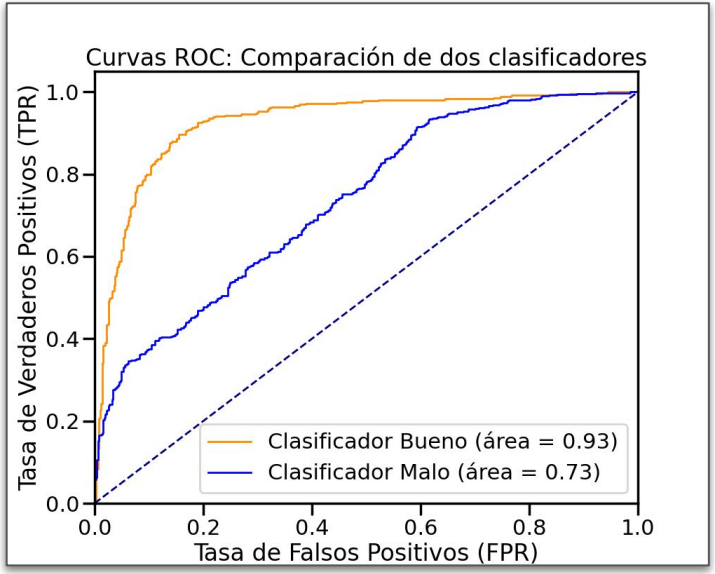
Para un ejemplo interactivo: https://mlu-explain.github.io/roc-auc/

Clasificadores probabilísticos

Curvas ROC - Interpretación AUC

*Si seleccionáramos al azar una instancia **positiva** y otra **negativa**, el **AUC ROC** nos dice la **probabilidad de que el modelo le asigne un puntaje más alto a la positiva**.*

- *Un modelo que siempre predice que una muestra negativa es más probable que tenga una etiqueta positiva que una muestra positiva tendrá un **AUC de 0** (o sea si ordena al reves).*
- *Si las probabilidades predichas son aleatorias, será **0.5.***
- *Si el modelo siempre predice que una muestra positiva es más probable que tenga una etiqueta positiva que una muestra negativa, entonces tendrá un **AUC de 1.***

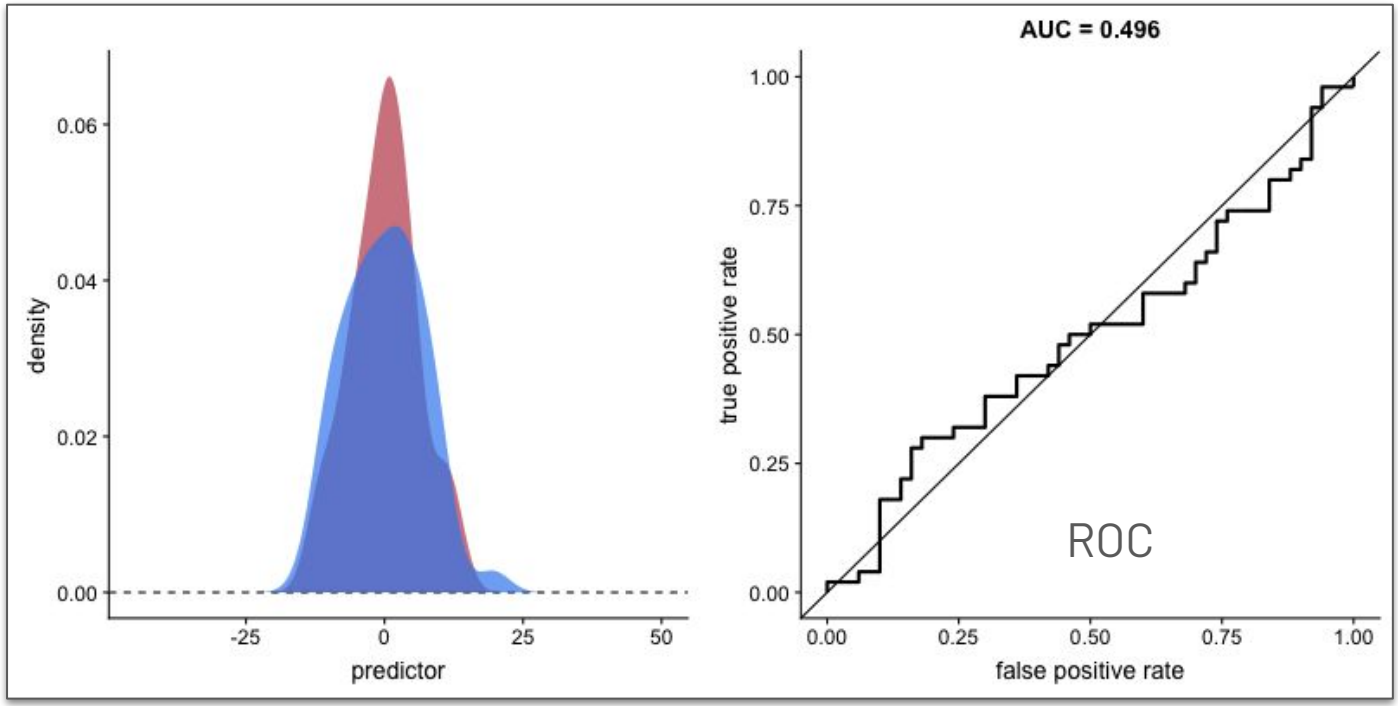


*Esta interpretación proporciona un método sencillo **para estimar el AUC:** contar la proporción de pares positivo-negativo clasificados correctamente. Se ha demostrado que este método para estimar el AUC es **equivalente** a una prueba estadística no paramétrica popular: la prueba de **Wilcoxon-Mann-Whitney***

[Mason, S. J., & Graham, N. E. (2002). Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation.]

Curvas ROC

Para discutir: **¿Qué muestra este gráfico? ¿Cuál es el modelo que se está probando?**

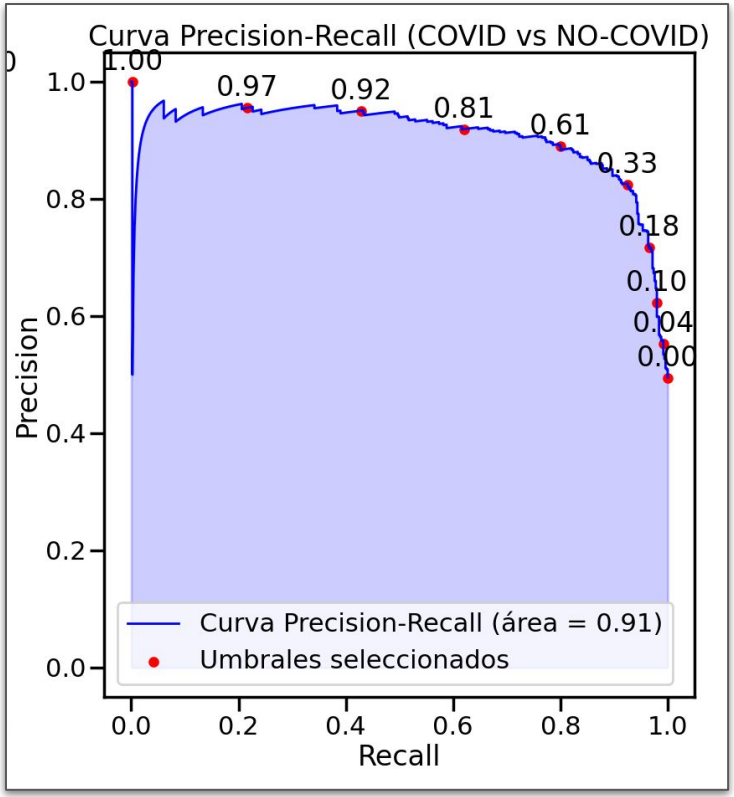


Fuente: https://paulvanderlaken.com/2019/08/16/roc-auc-precision-and-recall-visually-explained/
Para un ejemplo interactivo: https://mlu-explain.github.io/roc-auc/

Curvas Precision-Recall

Misma idea, pero medimos recall vs precision

De manera análoga, definimos las curvas **Precision-Recall (PR Curves)**



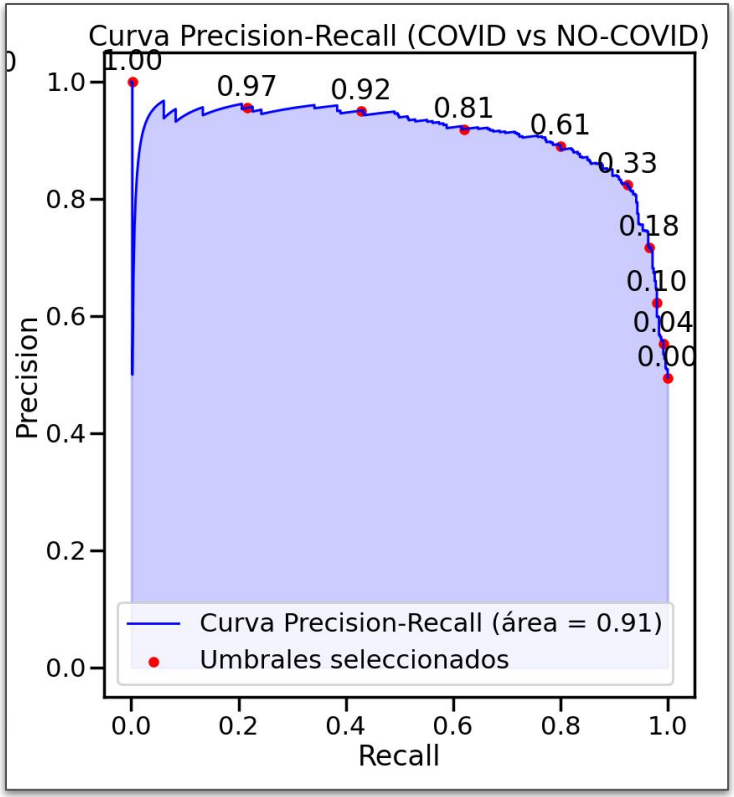
Curvas Precision-Recall

De manera análoga, definimos las curvas **Precision-Recall (PR Curves)**

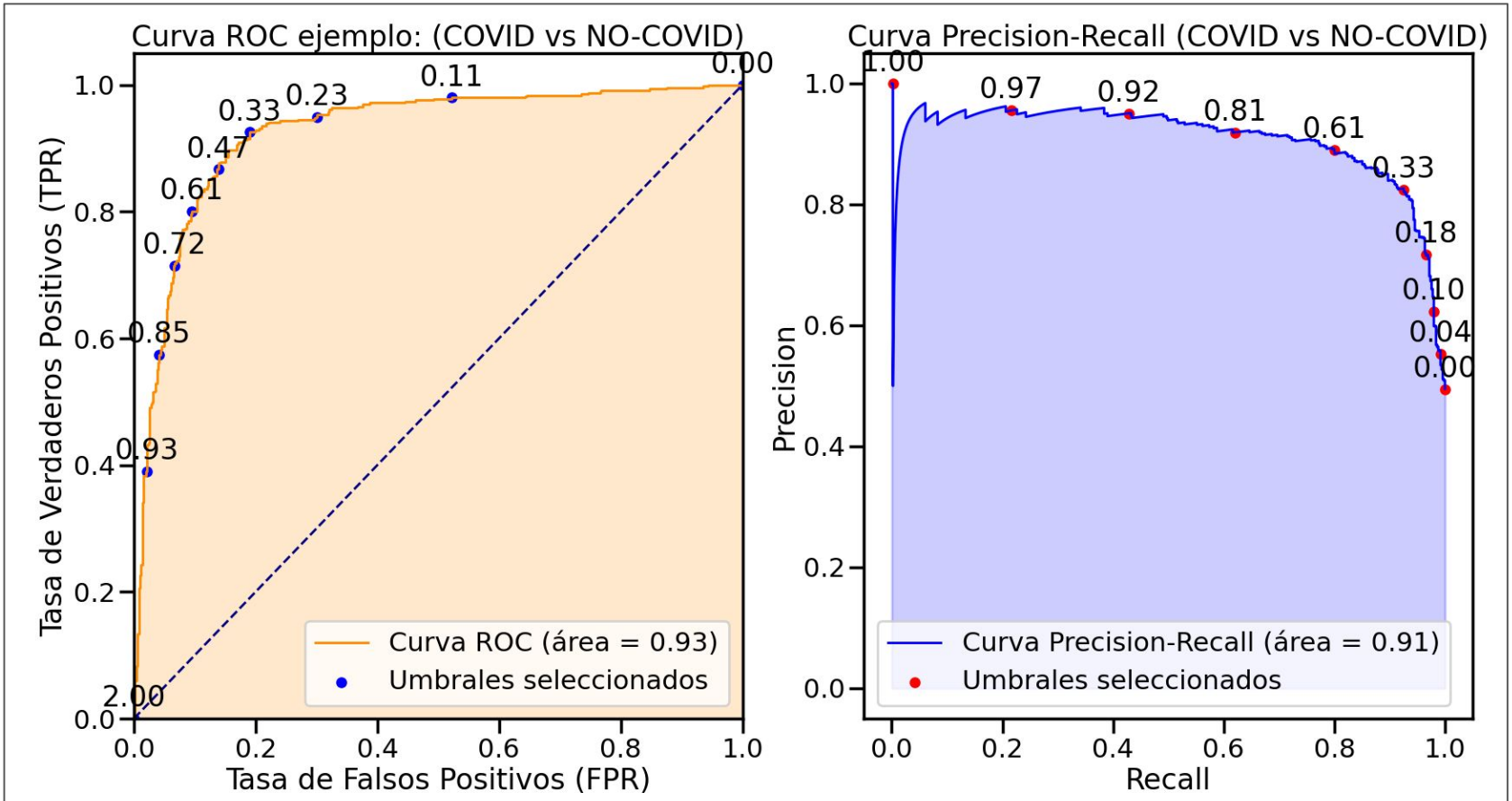
Algorithm: Construir Curva de Precisión-Recall

Require: Scores predichos: *predicciones*, Etiquetas verdaderas: *etiquetas_reales*

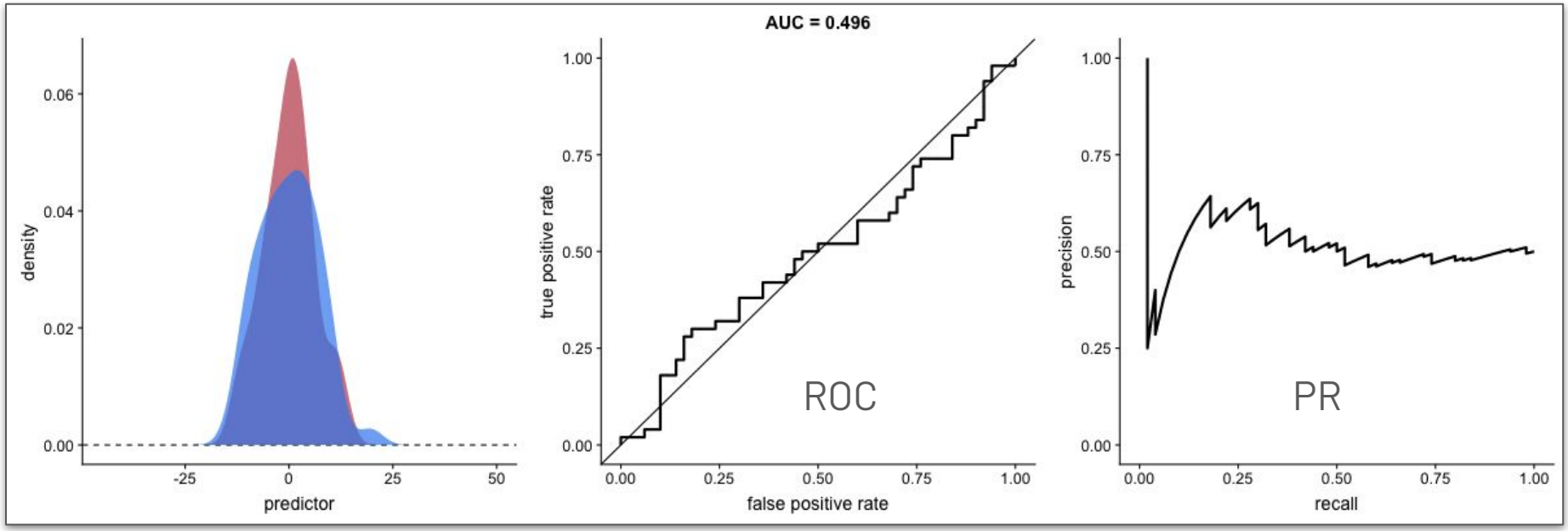
umbrales ← valores únicos de *predicciones* ordenados de forma descendente
Inicializar *precision* ← [ ], *recall* ← [ ], *umbrales_resultantes* ← [ ]
for *umbral* en *umbrales* **do**
    $TP \leftarrow$ Conteo de *predicciones* $\geq$ *umbral* y *etiquetas_reales* $= 1$
    $FP \leftarrow$ Conteo de *predicciones* $\geq$ *umbral* y *etiquetas_reales* $= 0$
    $FN \leftarrow$ Conteo de *predicciones* $<$ *umbral* y *etiquetas_reales* $= 1$
    $TN \leftarrow$ Conteo de *predicciones* $<$ *umbral* y *etiquetas_reales* $= 0$
    if $(TP + FP) > 0$ **then**
        precision_actual $\leftarrow \frac{TP}{TP+FP}$
    else
        precision_actual ← 1 {Si no hay positivos predichos, la precisión se considera perfecta}
    end if
    if $(TP + FN) > 0$ **then**
        recall_actual $\leftarrow \frac{TP}{TP+FN}$
    else
        recall_actual ← 0 {Si no hay positivos reales, el recall es cero}
    end if
    Añadir *precision_actual* a *precision*
    Añadir *recall_actual* a *recall*
    Añadir *umbral* a *umbrales_resultantes*
end for
Retornar *precision*, *recall*, *umbrales_resultantes*



Curvas Precision-Recall



Comparaciones

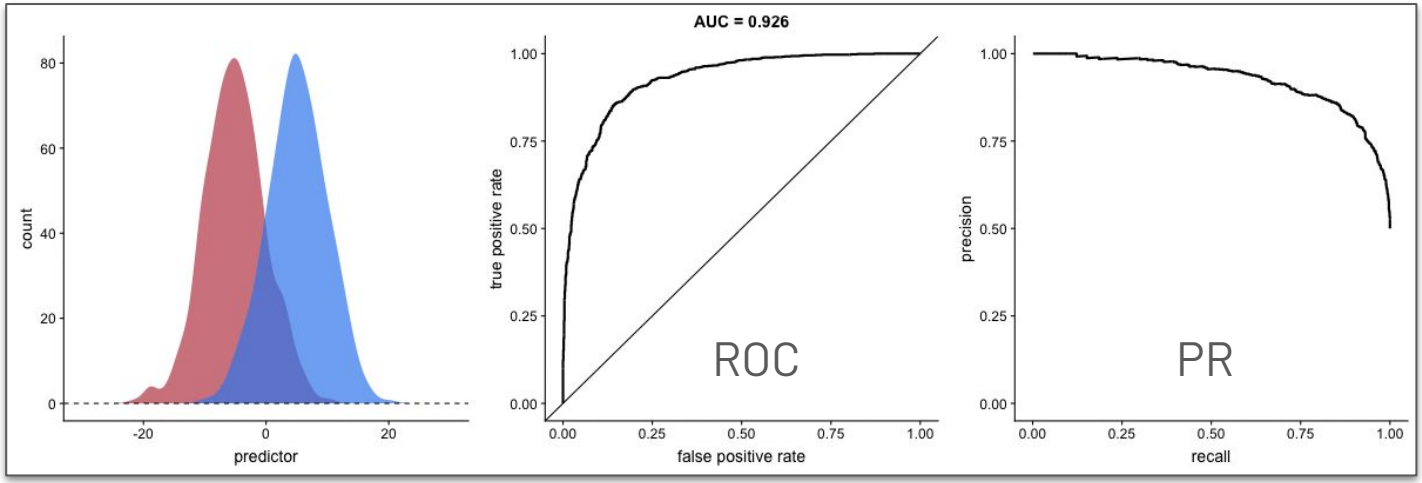


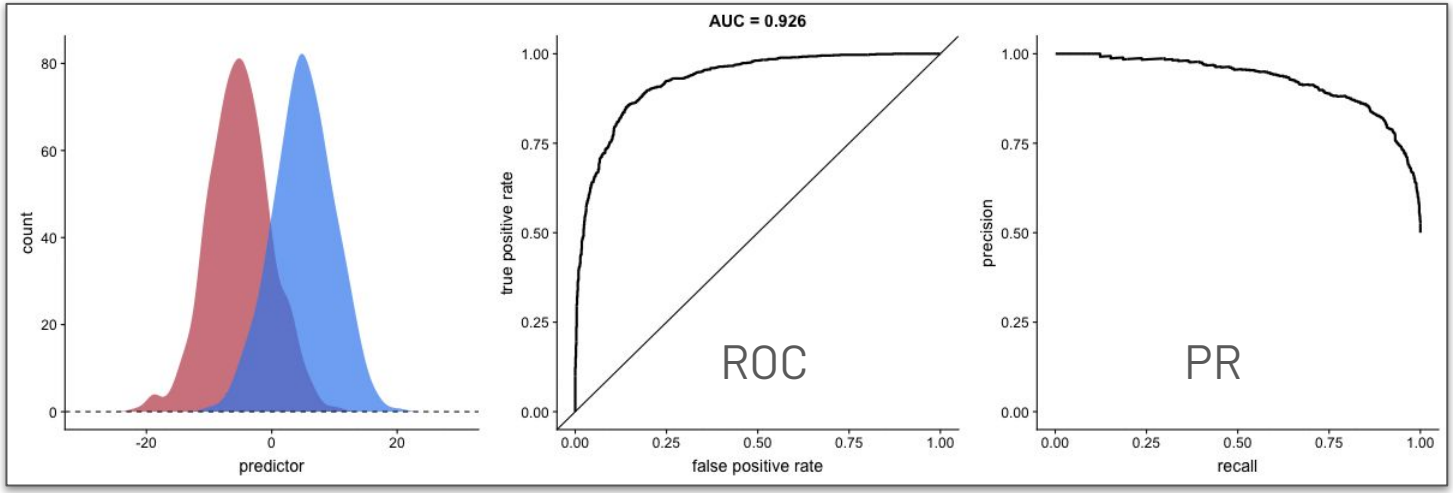
Fuente: https://paulvanderlaken.com/2019/08/16/roc-auc-precision-and-recall-visually-explained/

Desbalance

De 1000 a 50 ejemplos.

Pregunta:
¿Queremos que sea o no sensible al desbalance?





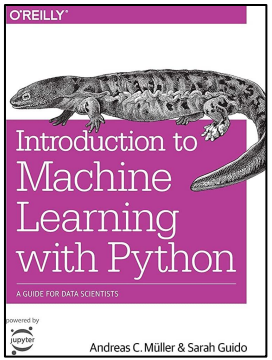
Fuente: https://paulvanderlaken.com/2019/08/16/roc-auc-precision-and-recall-visually-explained/

Tarea

- Leer el (el resto del capítulo) **capítulo 5 "Model Evaluation and Improvement"** de Müller, A. C., & Guido, S. (2016). **Introduction to machine learning with Python: a guide for data scientists.**


Andreas C. Müller


Sarah Guido



Opcional:

- Sección 1.5 **"Decision Theory"** del Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.
- Ojala, M., & Garriga, G. C. (2010). Permutation tests for studying classifier performance. Journal of machine learning research, 11(6). https://www.jmlr.org/papers/volume11/ojala10a/ojala10a.pdf
- **Ferrer, Luciana (CONICET - LIAA)**. "No Need for Ad-hoc Substitutes: The Expected Cost is a Principled All-purpose Classification Metric. Transactions on Machine Learning Research. "https://openreview.net/forum?id=5PPbvCExZs


Christopher Bishop


Markus Ojala


Gemma Garriga


Luciana Ferrer

Significancia estadística (fuera de programa)

Permutation Tests

Con mucha frecuencia, el **número de instancias** disponibles con etiquetas **no es suficiente** para obtener resultados robustos.

Es común, sobre todo en datos de aplicaciones médicas o biológicas, tener una gran dimensionalidad (miles de atributos) y un pequeño número de instancias (pacientes). **(N << D).**

Tengo un clasificador que me dió 55% accuracy.. una pregunta importante es:

¿Debemos creer en la performance obtenida?

TABLE 1 Participants' demographic and neuropsychological informa

	ADD ($n = 21$)	PD ($n = 18$)	Controls ($n = 16$)
Demographic data			
Sex (F:M)	13:8	10:8	13:3
Age	77.24 (6.47)	76.50 (6.40)	75.94 (4.35)
Years of education	11.24 (3.78)	9.39 (5.11)	12.94 (4.28)
Neuropsychological data			
MoCA	13.90 (4.34)	20.33 (4.68)	25.07 (3.43)
IFS battery	11.07 (4.48)	17.08 (4.86)	18.90 (4.26)

Abbreviations: ADD, Alzheimer's disease dementia; PD, Parkinson's disease;

Automated text-level semantic markers of Alzheimer's disease

Camila Sanz, Facundo Carrillo, Andrea Slachevsky, Gonzalo Forno, Maria Luisa Gorno Tempini, Roque Villagra, Agustín Ibáñez, Enzo Tagliazucchi, Adolfo M. García

https://alz-journals.onlinelibrary.wiley.com/doi/full/10.1002/dad2.12276

[Ojala and Garriga. Permutation Tests for Studying Classifier Performance. The Journal of Machine Learning Research (2010) vol. 11]

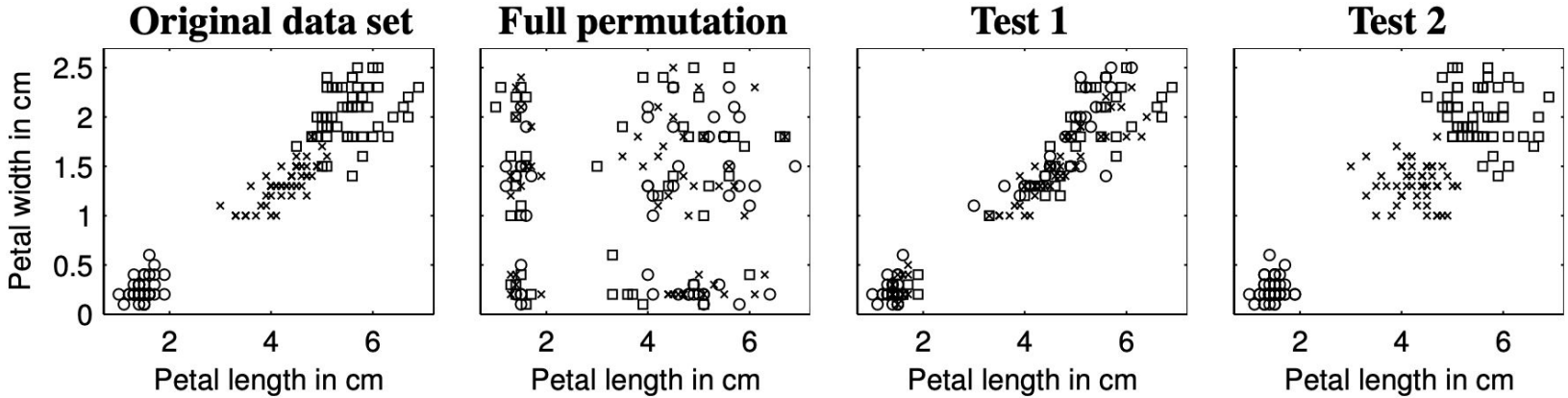
Permutation Tests

"Test 1": ¿Nuestro clasificador encontró una conexión real entre los datos y su clase?;

hipótesis nula: los atributos y las etiquetas son independientes

distribución nula: se estima permutando las etiquetas en los datos.

"Test 2": ¿explota la dependencia entre los atributos en la clasificación? (no lo veremos, ver paper)



[Ojala and Garriga. Permutation Tests for Studying Classifier Performance. The Journal of Machine Learning Research (2010) vol. 11]

Permutation Tests

[Ojala and Garriga. Permutation Tests for Studying Classifier Performance. The Journal of Machine Learning Research (2010) vol. 11]

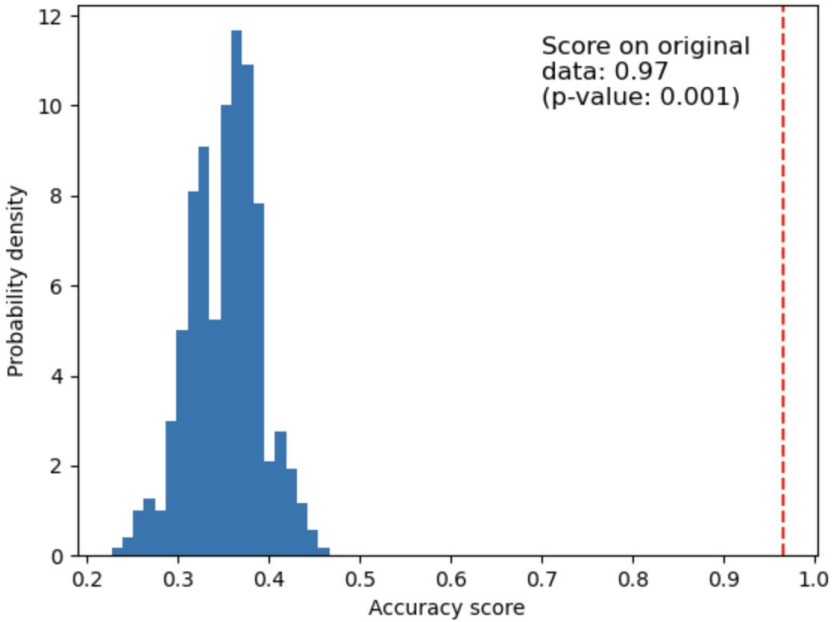
Algorithm: Prueba de Permutación usando Validación Cruzada

Require: clf {Un clasificador sin entrenar}
Require: X, y {Matriz de instancias y etiquetas verdaderas}
Require: $n\_permutaciones$ {Número de permutaciones}
Require: n_folds {Número de folds para k-fold-cross-val}

1: $true\_accuracy \leftarrow \text{CrossVal}(clf, X, y, n\_folds)$
2: Inicializar $permuted\_accuracies \leftarrow []$
3: **for** $i = 1$ hasta $n\_permutaciones$ **do**
4: $\quad y\_permuted \leftarrow \text{Shuffle}(y)$
5: $\quad perm\_accuracy \leftarrow \text{CrossVal}(clf, X, y\_permuted, n\_folds)$
6: $\quad$ Agregar $perm\_accuracy$ a $permuted\_accuracies$
7: **end for**
8: $p\_value \leftarrow \frac{\text{Count}(permuted\_accuracies \geq true\_accuracy)}{n\_permutaciones}$
9: **return** $true\_accuracy, permuted\_accuracies, p\_value$

permutation_test_score # (scikit-learn)

```
sklearn.model_selection.permutation_test_score(estimator, X, y, *,
groups=None, cv=None, n_permutations=100, n_jobs=None, random_state=0,
verbose=0, scoring=None, fit_params=None)
```



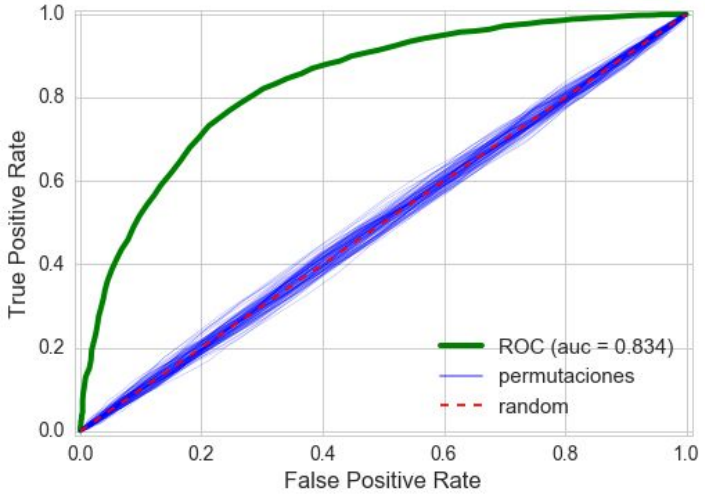
Permutation Tests

[Ojala and Garriga. Permutation Tests for Studying Classifier Performance. The Journal of Machine Learning Research (2010) vol. 11]

Algorithm: Prueba de Permutación usando Validación Cruzada

Require: clf {Un clasificador sin entrenar}
Require: X, y {Matriz de instancias y etiquetas verdaderas}
Require: $n\_permutaciones$ {Número de permutaciones}
Require: n_folds {Número de folds para k-fold-cross-val}

1: $true\_accuracy \leftarrow \text{CrossVal}(clf, X, y, n\_folds)$
2: Inicializar $permuted\_accuracies \leftarrow []$
3: **for** $i = 1$ hasta $n\_permutaciones$ **do**
4: $\quad y\_permuted \leftarrow \text{Shuffle}(y)$
5: $\quad perm\_accuracy \leftarrow \text{CrossVal}(clf, X, y\_permuted, n\_folds)$
6: $\quad$ Agregar $perm\_accuracy$ a $permuted\_accuracies$
7: **end for**
8: $p\_value \leftarrow \frac{\text{Count}(permuted\_accuracies \geq true\_accuracy)}{n\_permutaciones}$
9: **return** $true\_accuracy, permuted\_accuracies, p\_value$



Curva AUC-ROC + test de permutaciones

permutation_test_score

```
sklearn.model_selection.permutation_test_score(estimator, X, y, *,
groups=None, cv=None, n_permutations=100, n_jobs=None, random_state=0,
verbose=0, scoring=None, fit_params=None)
```

Teoría de la Decisión
(fuera de programa)

Teoría de la Decisión

Teoría de la **probabilidad** + **Teoría** de la **utilidad**

$P(Y=c \mid X=x)$, representa el estado de naturaleza desconocido (por ejemplo, si el paciente tiene cáncer de pulmón, cáncer de mama o no tiene cáncer). Ahora discutimos: ¿**qué acción** debemos tomar (por ejemplo, cirugía o no cirugía)?

Matriz de Costos

		Decisión		
		Operar	**Más tests**	**A casa.**
Clase real	**Tumor**	0	10	1000
	No tumor	500	20	0

Dados estos costos, ¿Cómo elijo los umbrales óptimos para tomar cada decisión?

- En algunas aplicaciones, el **impacto de los diferentes tipos de error es muy diferente entre sí** y las métricas vistas pueden no reflejar correctamente lo que es relevante en esos escenarios.
- En otros casos**, no se sabe de antemano cómo se utilizará el sistema**.
- A veces **cambia el caso de uso del sistema,** no queremos tener que reentrenar modelos.
- En cada uno de estos escenarios es necesario diseñar una métrica diferente para el caso de uso específico.

Teoría de la Decisión

Teoría de la **probabilidad** + **Teoría** de la **utilidad**

Idea... ¿Podremos usar este modo de pensamiento para tener la clase “no sé”?

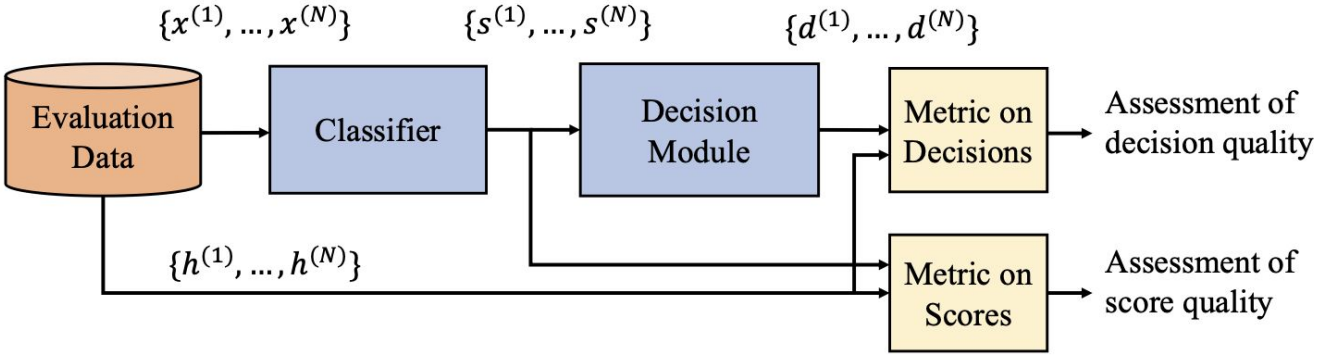
		Decisión		
		Clase 1	**Clase 2**	**No sé**
Clase real	**Clase 1**	…	…	…
	Clase 2	…	…	…

[Bishop 1.5.3] “The Reject Option”

Teoría de la Decisión

Teoría de la **probabilidad** + **Teoría** de la **utilidad**

Luciana Ferrer
(LIAA, Conicet)



Published in Transactions on Machine Learning Research (02/2025)

No Need for Ad-hoc Substitutes: The Expected Cost is a Principled All-purpose Classification Metric

Luciana Ferrer *lferrer@dc.uba.ar*
Instituto de Ciencias de la Computación
CONICET - Universidad de Buenos Aires, Argentina

Reviewed on OpenReview: *https://openreview.net/forum?id=5PPbvCExZs*

https://openreview.net/forum?id=5PPbvCExZs

Evaluating Posterior Probabilities: Decision Theory, Proper Scoring Rules, and Calibration

Luciana Ferrer
LFERRER@DC.UBA.AR
Instituto de Ciencias de la Computación
Universidad de Buenos Aires - CONICET, Argentina

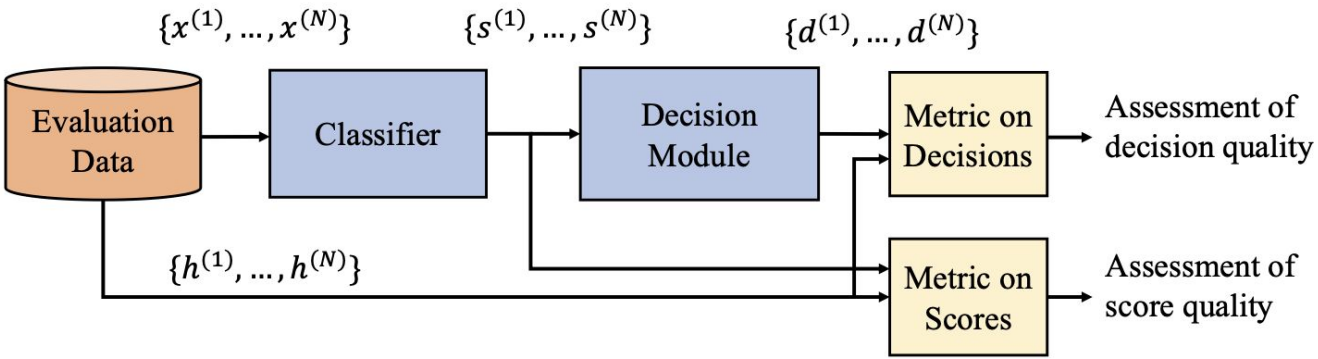
Daniel Ramos
DANIEL.RAMOS@UAM.ES
AUDIAS Lab. - Audio, Data Intelligence and Speech
Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain

https://arxiv.org/pdf/2408.02841

Teoría de la Decisión

Teoría de la **probabilidad** + **Teoría** de la **utilidad**

Luciana Ferrer
(LIAA, Conicet)



Surprisingly, the general form for the EC is rarely used in the machine learning literature. Instead, alternative ad-hoc metrics like the F-beta score and the Matthews correlation coefficient (MCC) are used for many applications. In this work, we argue that the EC is superior to these alternative metrics, being more general, interpretable, and adaptable to any application scenario. We provide both theoretically-motivated discussions as well as examples to illustrate the behavior of the different metrics.

https://openreview.net/forum?id=5PPbvCExZs

Teoría de la Decisión

Teoría de la **probabilidad** + **Teoría** de la **utilidad**

Nueva mética: **Costo Esperado (EC)**

Se define como una generalización de la probabilidad de error para casos en los que los **errores tienen consecuencias con distinto nivel de gravedad**.

Del campo de la "Teoría de la utilidad". (mundo economía) $Utilidad = (-1) * Costo$

puede ser utilizada para:

(A) Obtener la acción óptima.
(B) Como una métrica de desempeño que se puede calcular a partir de decisiones ya tomadas.

Matriz de Costos

		Decisión		
		Decisión 1	**...**	**Decisión M**
Clase real	**Clase 1**	$C_{1,1}$	...	$C_{1,M}$
	...	...	...	...
	Clase K	$C_{K,1}$	...	$C_{K,M}$

Costo Esperado Empírico

$$\hat{EC} = \sum_{c=1}^{K}\sum_{d=1}^{M} C_{c,d}\frac{n_{cd}}{n} = \sum_{c=1}^{K}\sum_{d=1}^{M} C_{c,d}P_c R_{cd}$$

donde

- $C_{c,d}$ el costo de tomar la decisión d para la clase c.
- $P_c = n_c/n$ es la probabilidad a priori empírica de la clase c (n_c es el número de instancias de la clase c)
- $R_{cd} = n_{cd}/n_c$ es la fracción de instancias de clase c para las cuales el sistema tomó la decisión d.

Decisor de Bayes

- Dado el costo de cada decisión
- Y dadas las probabilidades

Qué decisión tomar?

Un clasificador calibrado garantiza que, por ejemplo, si el modelo predice una probabilidad de 0,7 para una determinada clase, entonces aproximadamente el 70% de esas instancias pertenecen realmente a esa clase.

Si supieramos que clasificador está **calibrado**, se puede demostrar que la **mejor decisión (la que minimiza el costo esperado)** está dada por:

$$d^{\star}(q^{(i)}) = \arg\min_{d \in D} \sum_{c=1}^{K} C_{c,d} * q_c^{(i)}$$

- $q^{(i)} = h(\mathbf{x}^{(i)})$ (el vector de probabilidades asignada por el modelo a la i-esima instancia)
- $C_{c,d}$ el costo de tomar la decisión d para la clase c.