

Laboratorio de datos, clase 10

Más sobre evaluación de modelos y
preparación de features

Prof. Enzo Tagliazucchi

tagliazucchi.enzo@gmail.com

www.cocucolab.org

En la clase pasada vimos que...

- Si un dataset no está balanceado, $\text{accuracy} = 0.5$ no es el nivel de “chance”
- Podemos resolver esto introduciendo la matriz de confusión y cantidades derivadas
- Además, si el dataset no está balanceado el clasificador está sesgado para clasificar bien la clase más representada
- Esto se puede resolver con downsampling, o cambiando los pesos en la función de costo

Labels reales

		sano	enfermo
Labels	sano	verdadero negativo	falso negativo
predichos	enfermo	falso positivo	verdadero positivo

$$\text{Specificity} = \frac{\begin{array}{|c|c|} \hline \text{TN} & \\ \hline & \\ \hline \end{array}}{\begin{array}{|c|c|} \hline \text{TN} & \\ \hline \text{FP} & \\ \hline \end{array}}$$

$$\text{Sensitivity} = \frac{\begin{array}{|c|c|} \hline & \\ \hline & \text{TP} \\ \hline \end{array}}{\begin{array}{|c|c|} \hline & \text{FN} \\ \hline & \text{TP} \\ \hline \end{array}}$$

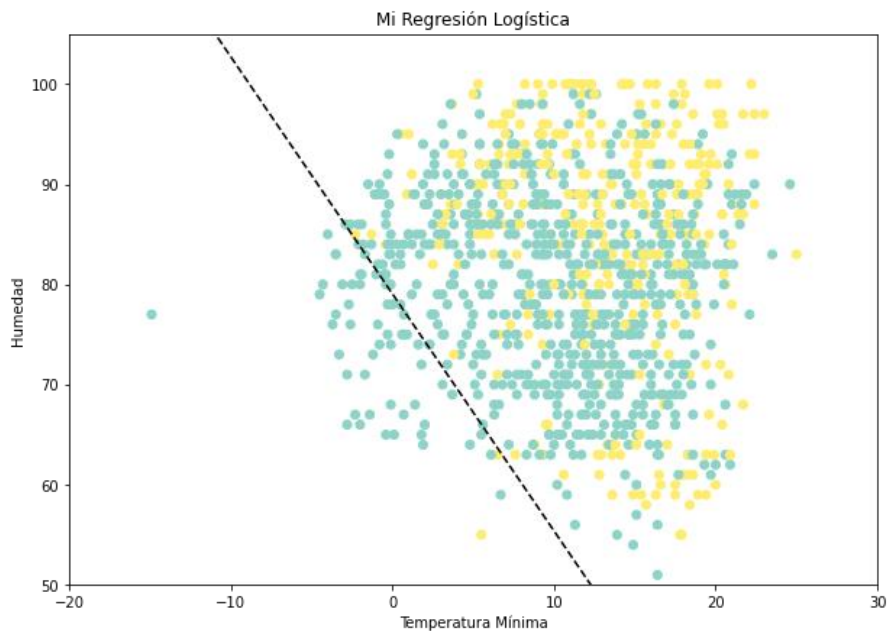
$$\text{Balanced accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

Observación importante - I

La matriz de confusión se construye para una *elección determinada* de los pesos.

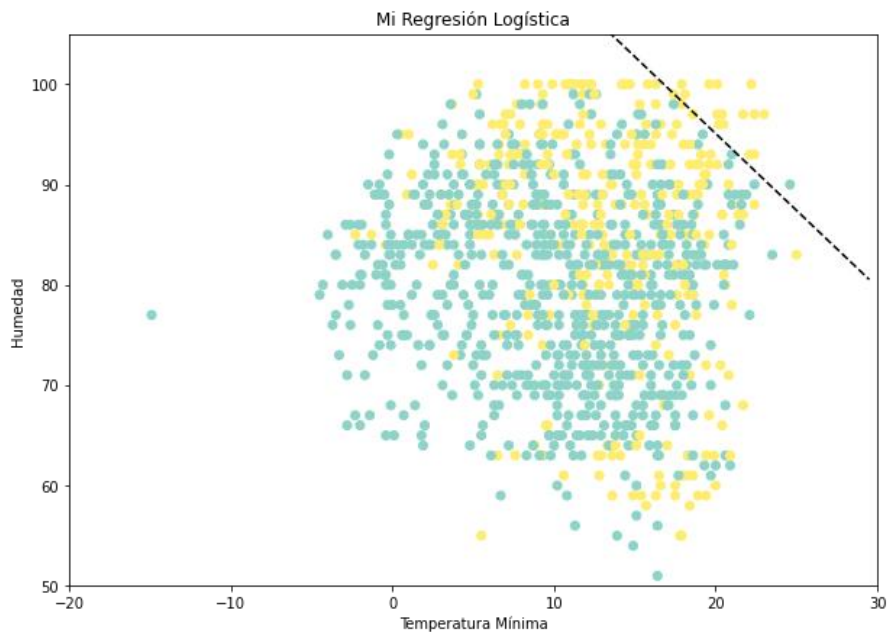
Puede que la performance del modelo sea buena para una elección de los pesos pero no tan buena para otra elección posible.

¿Cómo podemos decir algo sobre la performance para todas las posibles elecciones?



```
El score del modelo es de: 0.3681
Matriz de confusión del modelo es:
[[ 53  2]
 [664 335]]
Sensibilidad del modelo es de: 0.9941
Especificidad del modelo es de: 0.0739
BA del modelo es de: 0.534
```

pero BA mayor
para el caso
balanceado!



```
El score del modelo es de: 0.6926
Matriz de confusión del modelo es:
[[716 323]
 [ 1 14]]
Sensibilidad del modelo es de: 0.0415
Especificidad del modelo es de: 0.9986
BA del modelo es de: 0.5201
```

Observación importante - II

Recordemos que la interpretación de la regresión logística es probabilística:

$$P(y|x, \beta) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot x)}}$$

Probabilidad de que $y=1$
dados los features \mathbf{x} , y los
parámetros β

Decimos que $y = 1$
si pasa que $P(y|x, \beta) > 0.5$
($y = 1$ en caso contrario)

Pero podemos introducir un umbral T arbitrario.

Decimos que $y = 1$ si pasa que $P(y|x, \beta) > T$

($y = 1$ en caso contrario)

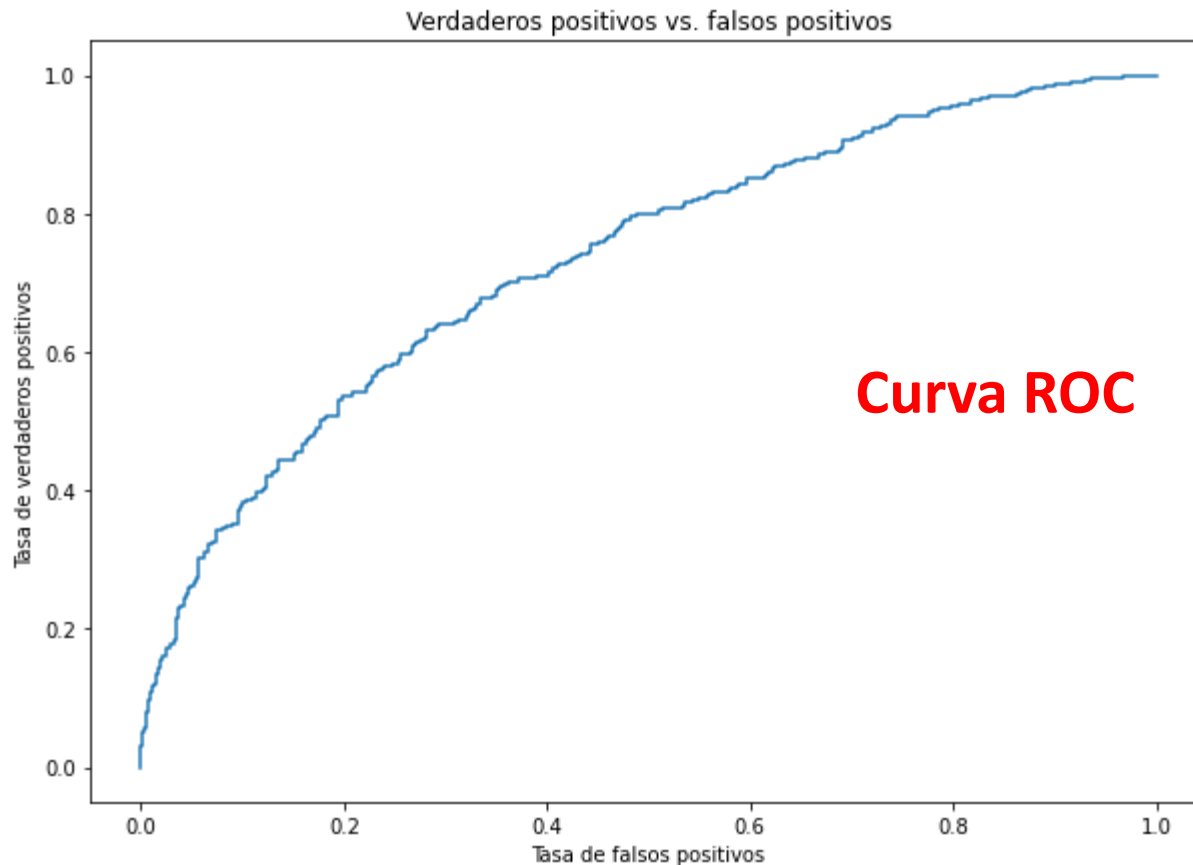
A mayor T , más alto el requerimiento para ser asignado a la clase 1. Esto determina a su vez el error tipo 1 y tipo 2.

(por ejemplo, si clase 1 es “enfermo”, quiero tener un T no muy alto para asegurarme de errar en la dirección de falsos positivos)

Para cada T computo,

- Tasa de verdaderos positivos (sensibilidad) = $\frac{TP}{TP+FN}$
- Tasa de falsos positivos = $\frac{FP}{FP+TN}$

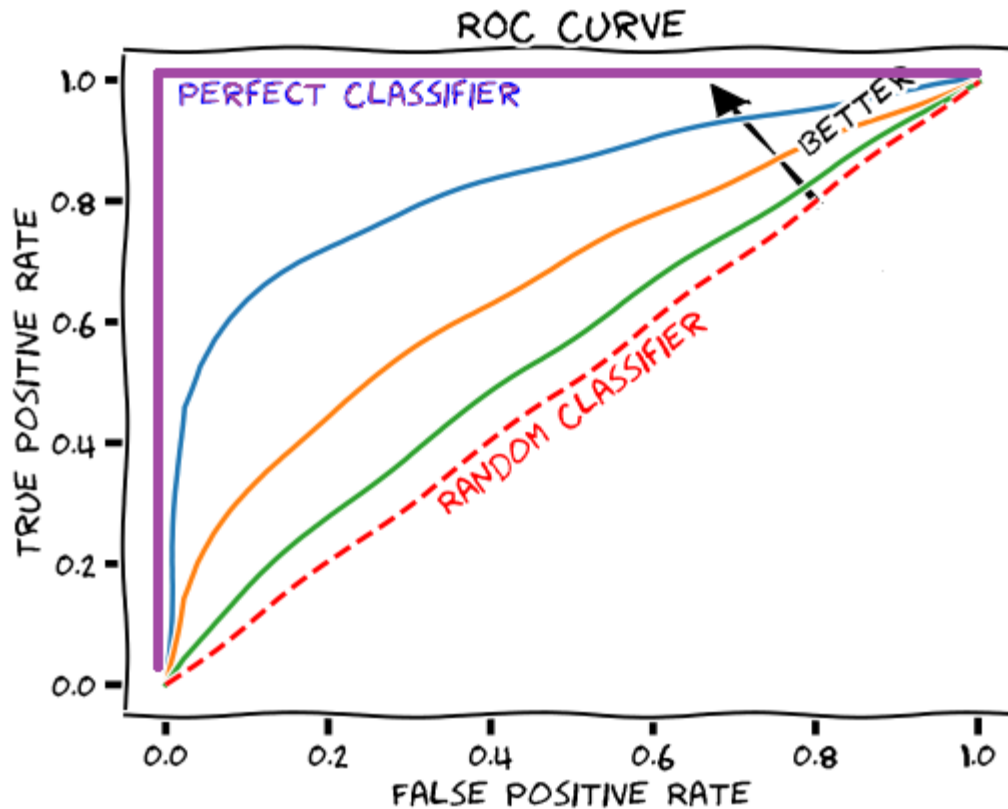
donde TP = verdaderos positivos, FN = falsos negativos, FP = falsos positivos y TN = verdaderos negativos.



Área bajo la curva ROC (AUC) como un criterio de performance independiente de T.

AUC cercano a 1: performance casi perfecta

AUC cercano a 0.5: performance a nivel chance



Problemas con train-test split

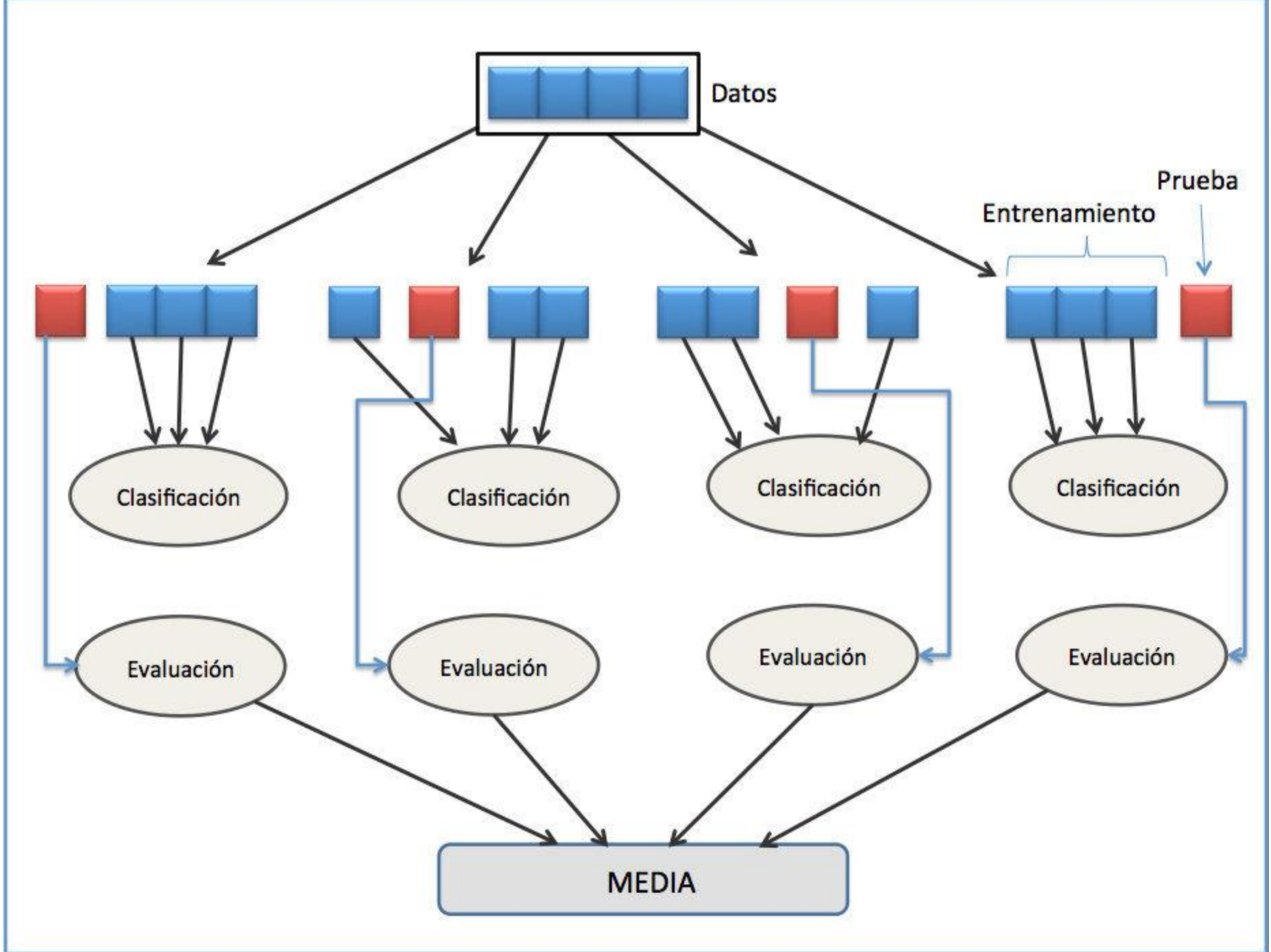
La performance estimada depende de si tuve suerte y quedaron datos muy informativos en el conjunto de entrenamiento

Supongamos que hago un split 70%-30% de mis datos. Entonces tengo una predicción no sobreajustada únicamente sobre el 30% de mis datos (el otro 70% es un conjunto donde potencialmente sobreajusté el modelo)

Validación cruzada estratificada

Una forma de atenuar estos problemas es usar validación cruzada estratificada con K folds:

1. Se dividen los datos en K subconjuntos de forma tal que la prevalencia de ambas clases esté balanceada en cada subconjunto.
2. Se elige uno de esos subconjuntos y se lo designa set de evaluación. Todos los demás son sets de entrenamiento. Se entrena entonces al modelo usando estos datos para evaluación y entrenamiento, y se genera una predicción para cada uno de los datos en el conjunto que fue elegido como evaluación.
3. Se repite el proceso usando cada uno de los K subconjuntos para evaluación exactamente una vez.



Datos

Entrenamiento

Prueba

Clasificación

Clasificación

Clasificación

Clasificación

Evaluación

Evaluación

Evaluación

Evaluación

MEDIA

¿Qué features incluir?

No es obvio que sumar más y más *features* sea siempre positivo.

Select K-best: obtengo una medida de qué tan diferente es el feature entre las dos clases (por defecto, ANOVA F-score) y luego me quedo con los K features que tienen máximo score.


¿Cómo incluir datos categóricos?

One-hot encoding: crear un feature binario por cada valor posible del dato categórico, que vale 1 si ese sample tiene ese valor posible, y 0 sino.

Human-Readable

Machine-Readable

Pet	Cat	Dog	Turtle	Fish
Cat	1	0	0	0
Dog	0	1	0	0
Turtle	0	0	1	0
Fish	0	0	0	1
Cat	1	0	0	0



¿Qué tan seguro estoy de que mi clasificador es mejor que el azar?

Corro mi clasificador y obtengo $AUC=0.62$

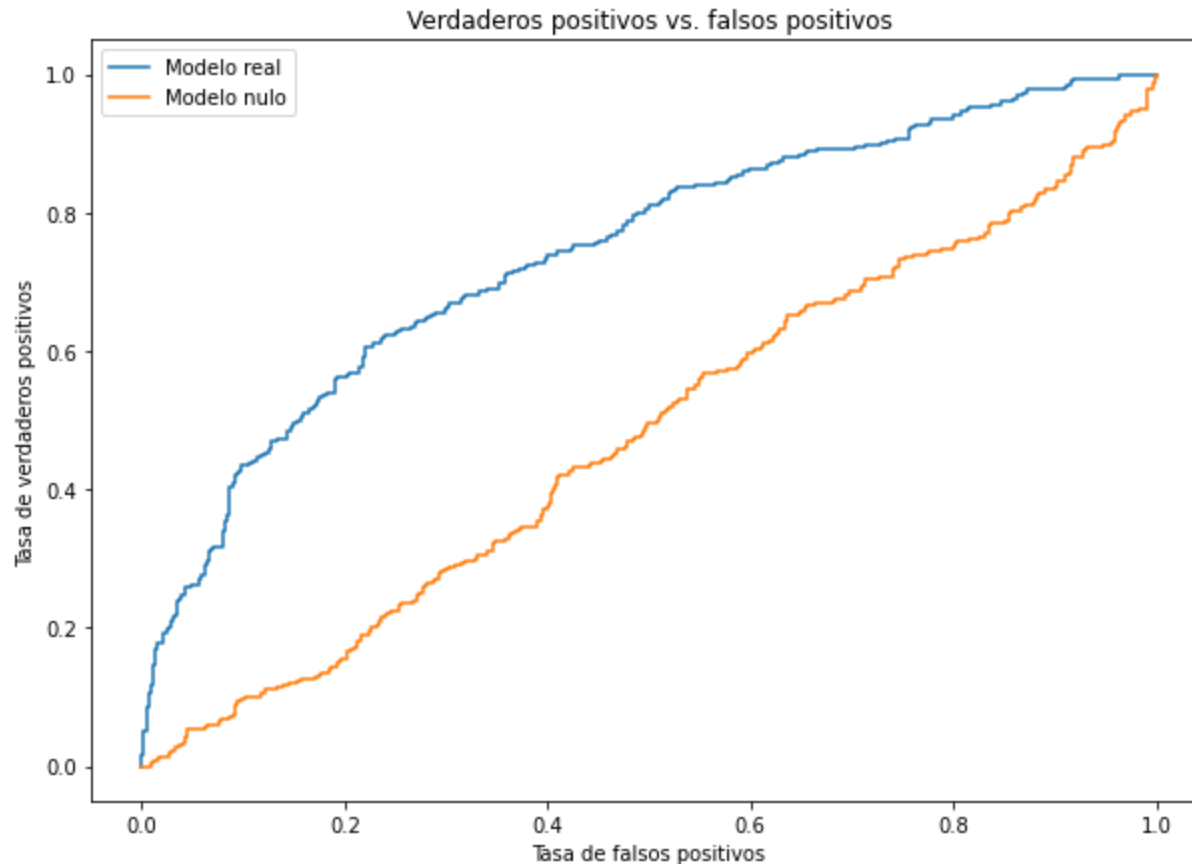
Lo vuelvo a correr y obtengo $AUC=0.59$

Claramente hay fluctuaciones por el armado de los folds, etc., en el score AUC que obtengo.

¿Cómo decido si realmente $AUC>0.5$ o simplemente tuve suerte?

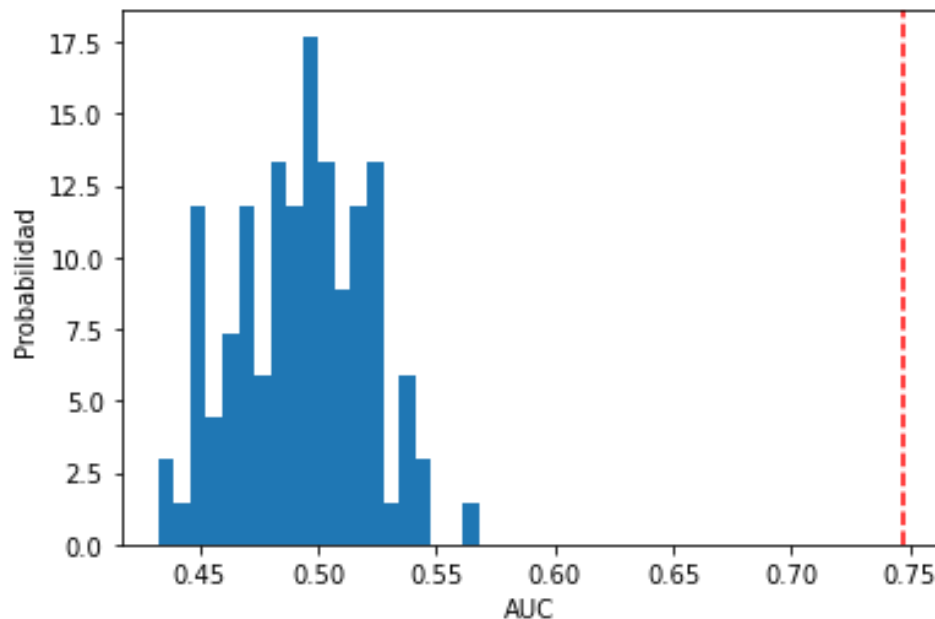
¿Qué tan seguro estoy de que mi clasificador es mejor que el azar?

Construyo un modelo nulo randomizando las etiquetas de las clases:

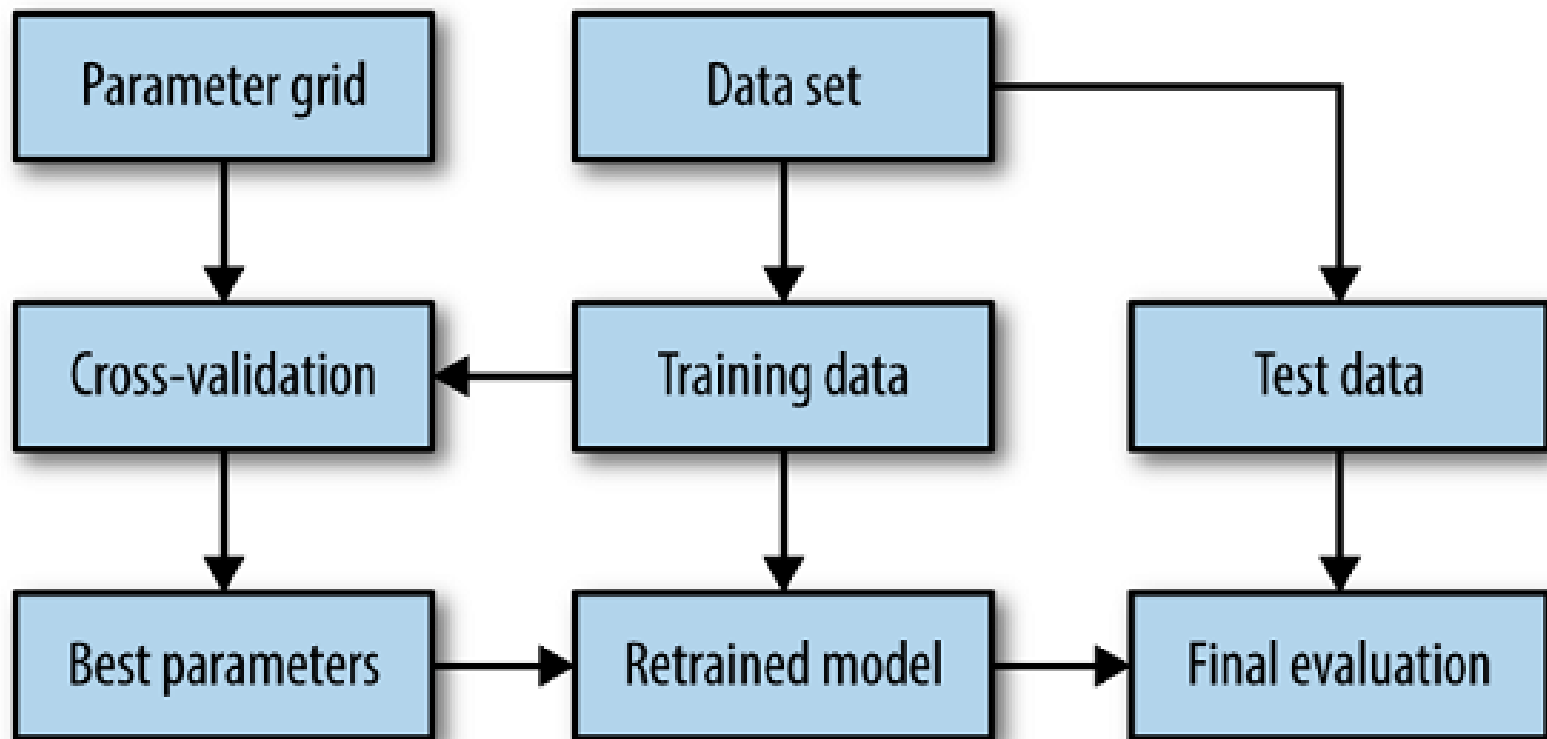


¿Qué tan seguro estoy de que mi clasificador es mejor que el azar?

Repito muchas veces obteniendo los valores de AUC con el modelo nulo shuffleado. Luego, cuanto la cantidad de veces que ese AUC es mejor que el del modelo sin shufflear y divido por la cantidad de veces que corrí el modelo nulo. Esto me da un p-valor que se interpreta como la probabilidad de que el AUC de mi clasificador sea compatible con el nivel chance.



El esquema general para seguir en un problema de clasificación



Segundo ejercicio obligatorio

1. Reciben un dataset más la información relevante, que en realidad viene de hacer un split 70%-30%. Nosotros nos quedamos con el 30%, ustedes tienen todos el mismo 70%.
2. El ejercicio es entrenar el mejor modelo de regresión logística que puedan. Recomendamos usar todo lo que vimos: agregar features nuevos no-lineales, usar K-best feature selection, usar regularización, optimizar hiperparámetros con validación cruzada y evaluar en un conjunto que no haya sido usado para la optimización, etc.
3. Llegado el día de entrega, les mandamos el 30% restante sin los labels. Ustedes nos devuelven un vector binario que son las probabilidades que devuelven su regresión logística.
4. Nosotros computamos el AUC del modelo de cada uno y armamos un ranking

Publicamos el ranking. Todos comparten sus notebooks (si quieren). Los tres mejores son invitados a contar brevemente cómo hicieron y ganan fabulosos premios.

Obs: docentes pueden participar pero no ganar

