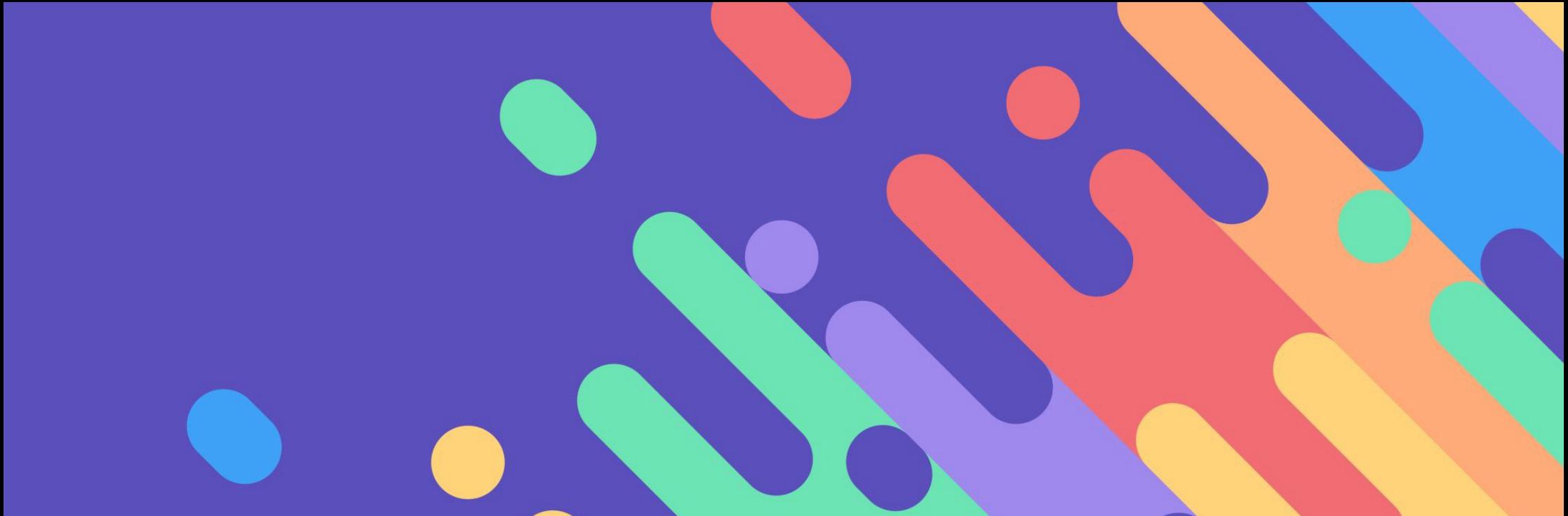


CLASIFICADORES



Aprendizaje
Automático
CEIoT - FIUBA

Dr. Ing. Facundo Adrián
Lucianna



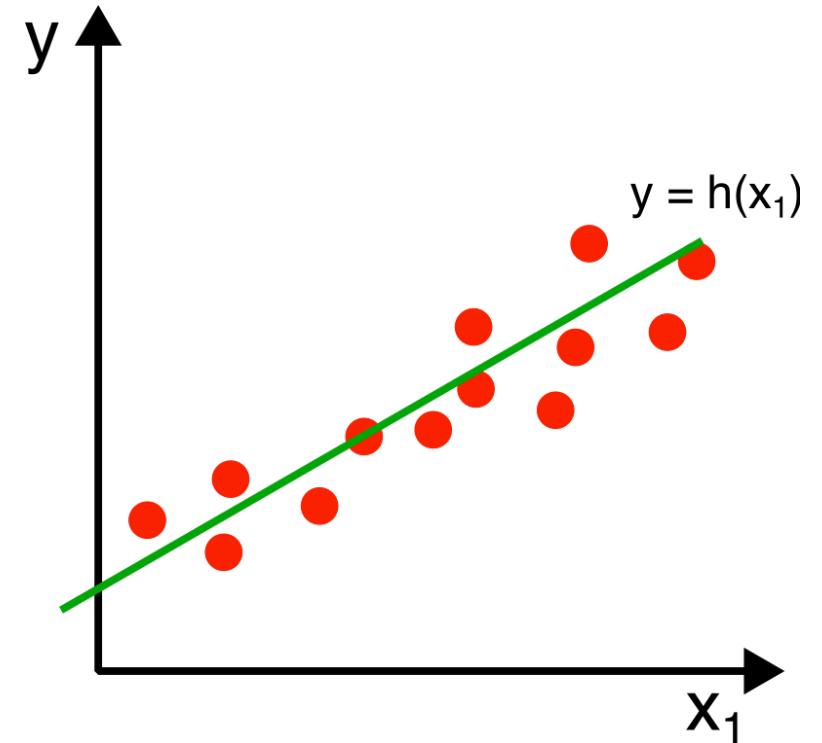
LO QUE VIMOS LA CLASE ANTERIOR...

REGRESIÓN

Si tenemos un problema donde el target y es una *variable numérica*, se llama un **problema de regresión**.

Se centra en estudiar las relaciones entre una variable dependiente de una o más variables independientes.

Es importante notar que, en Aprendizaje Automático, cuando buscamos una $h(X)$ estamos armando un modelo puramente empírico. Es decir, nos basamos 100% en los datos medidos. En contraste con los modelos basados en propiedades fundamentales.



REGRESIÓN LINEAL

El modelo de regresión lineal más simple es el que involucra una combinación lineal de las variables de entradas:

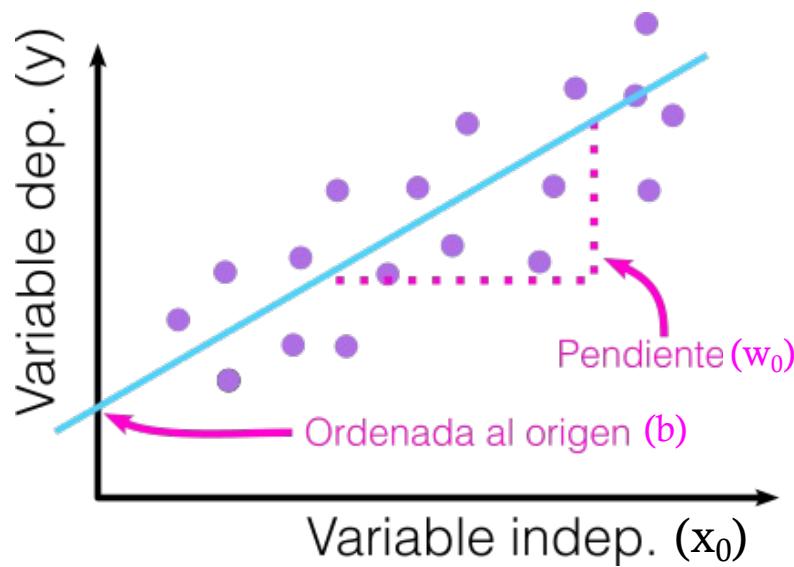
$$\hat{y} = h(X) = b + w_0x_0 + \cdots + w_dx_d$$

- $X = (x_0, x_1, \dots, x_d)$ Son los *features* de nuestras observaciones. Son todas variables numéricas
- b, w_0, \dots, w_d Son los coeficientes del modelo. Son números reales. Cuanto más cerca de cero, la variable dependiente depende menos del *feature* que multiplica.
- \hat{y} Es la predicción del modelo. Es con quien comparamos con el *Label* de la observación

REGRESIÓN LINEAL

La regresión lineal de una sola variable independiente:

$$\hat{y} = h(X) = b + w_0x_0$$



MÉTRICAS DE EVALUACIÓN

Vimos las siguientes métricas:

- El **coeficiente de determinación (R^2)**
- **Error absoluto medio (MAE):** $\frac{1}{N} \sum_{i=0}^{N-1} |y_{[i]} - \hat{y}_{[i]}|$
- **Error cuadrático medio (MSE):** $\frac{1}{N} \sum_{i=0}^{N-1} (y_{[i]} - \hat{y}_{[i]})^2$
- **Error absoluto porcentual medio (MAPE):** $\frac{100\%}{N} \sum_{i=0}^{N-1} \left| \frac{y_{[i]} - \hat{y}_{[i]}}{y_{[i]}} \right|$

REGRESIÓN DE RIDGE Y LASSO

Regresión de Ridge:

$$\sum_{i=0}^{N-1} (y_{[i]} - b - W^T X_{[i]})^2 + \alpha \sum_{j=0}^{d-1} w_j^2$$

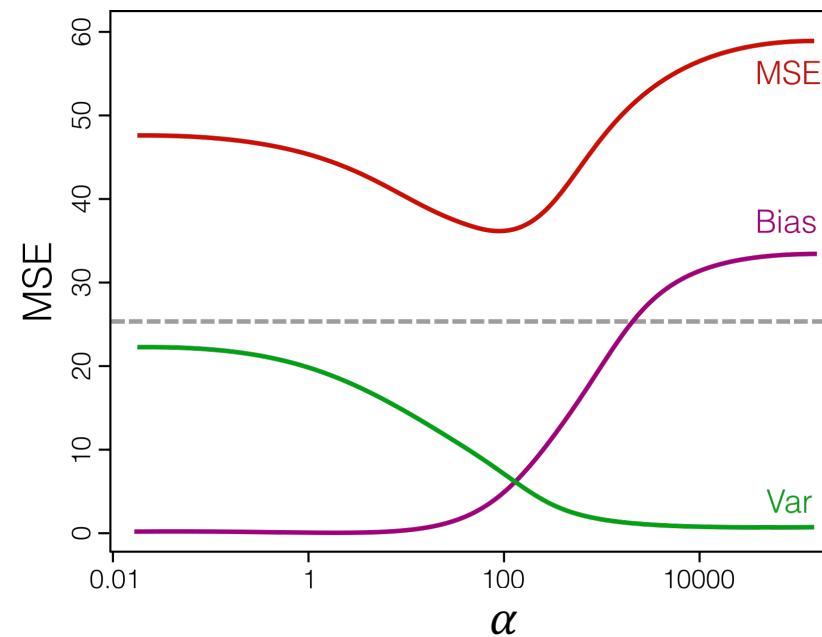
Regresión de Lasso:

$$\sum_{i=0}^{N-1} (y_{[i]} - b - W^T X_{[i]})^2 + \alpha \sum_{j=0}^{d-1} |w_j|$$

REGRESIÓN DE RIDGE

¿Para qué nos sirve?

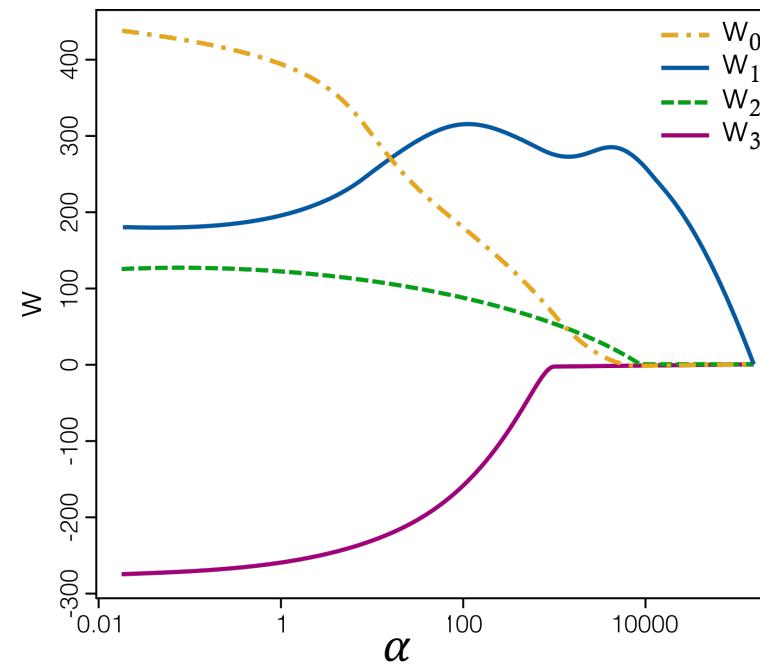
En general, cuando la verdadera relación es lineal, la regresión lineal tiene mucha varianza. Esto principalmente ocurre cuando el **número de observaciones es cercano al número de coeficientes**. En estos casos, la regresión de Ridge funciona mejor.

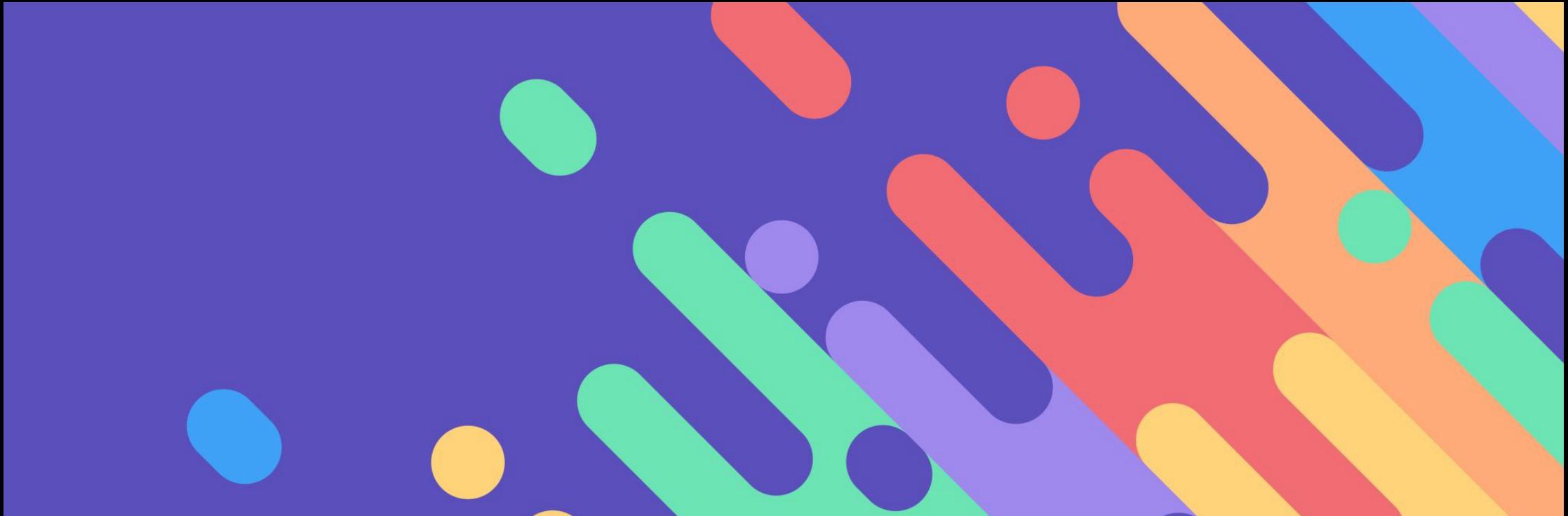


REGRESIÓN DE LASSO

¿Para qué nos sirve?

Esta regresión cuando α crece, algunos coeficientes se hacen exactamente cero. Por lo que Lasso realiza una selección de atributos.



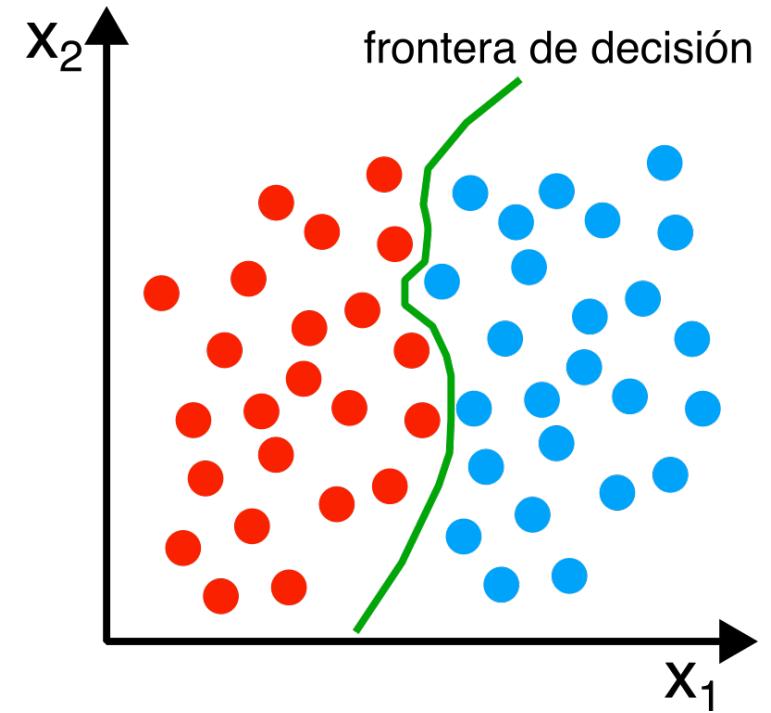


CLASIFICACIÓN

CLASIFICACIÓN

Es más común encontrarnos con problema de clasificación que de regresión:

- Una persona llega a una guardia con un set de síntomas atribuidos a una de tres condiciones médicas.
- Un servicio de banca online debe determinar si una transacción en el sitio es fraudulenta o no, usando como base la dirección IP, historia de transacciones, etc.
- En base a la secuencia de ADN de un número de pacientes con y sin una enfermedad dada, un genetista debe determinar que mutaciones de ADN genera un efecto nocivo relacionado a la enfermedad o no.



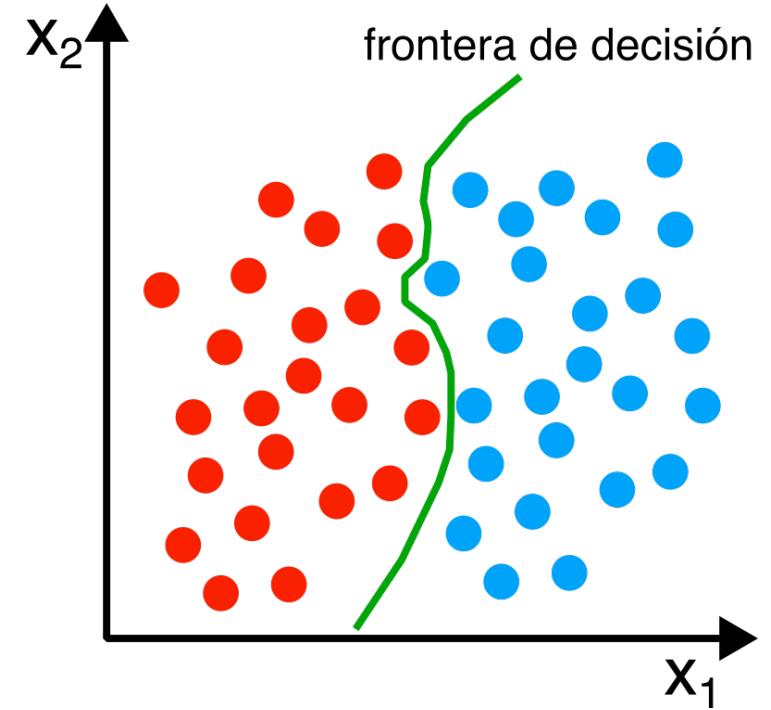
CLASIFICACIÓN

Regresión y clasificación son problemas muy similares entre sí. En ambos buscamos predecir una variable, la diferencia radica en que **regresión** predice una variable **numérica** y **clasificación** una **categórica**.

¿Por qué no usar regresión para predecir respuestas cualitativas?

Si usamos el ejemplo de los pacientes que llegan a la guardia, supongamos que hay tres diagnósticos:

- ACV
- Sobredosis
- Ataques epilépticos



CLASIFICACIÓN

Realizamos la siguiente codificación

- **ACV**: 1
- **Sobredosis**: 2
- **Ataques epilépticos**: 3

Aplicamos un modelo de regresión lineal para predecir en base a los predicadores del paciente.

El problema con esto es que la codificación implica un orden en los resultados, poniendo a **sobredosis** entre **ACV** y **ataques epilépticos**, y además que la distancia entre **ACV** y **sobredosis** es la misma que **sobredosis** y **ataques epilépticos**.

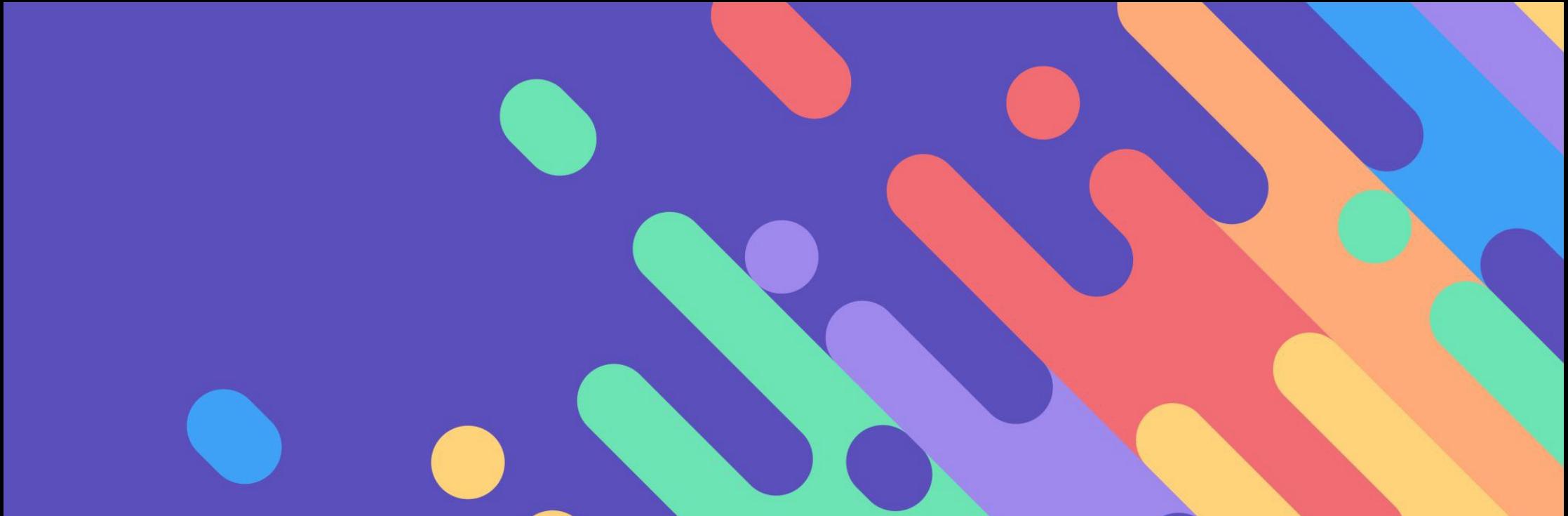
CLASIFICACIÓN

Pero tranquilamente podríamos haber elegido:

- **Ataques epilépticos**: 1
- **ACV**: 2
- **Sobredosis**: 3

Esto nos da una relación totalmente diferente.

Cada una de estas codificaciones produciría modelos lineales diferentes que, en última instancia, conducirían a diferentes conjuntos de predicciones sobre observaciones de prueba.

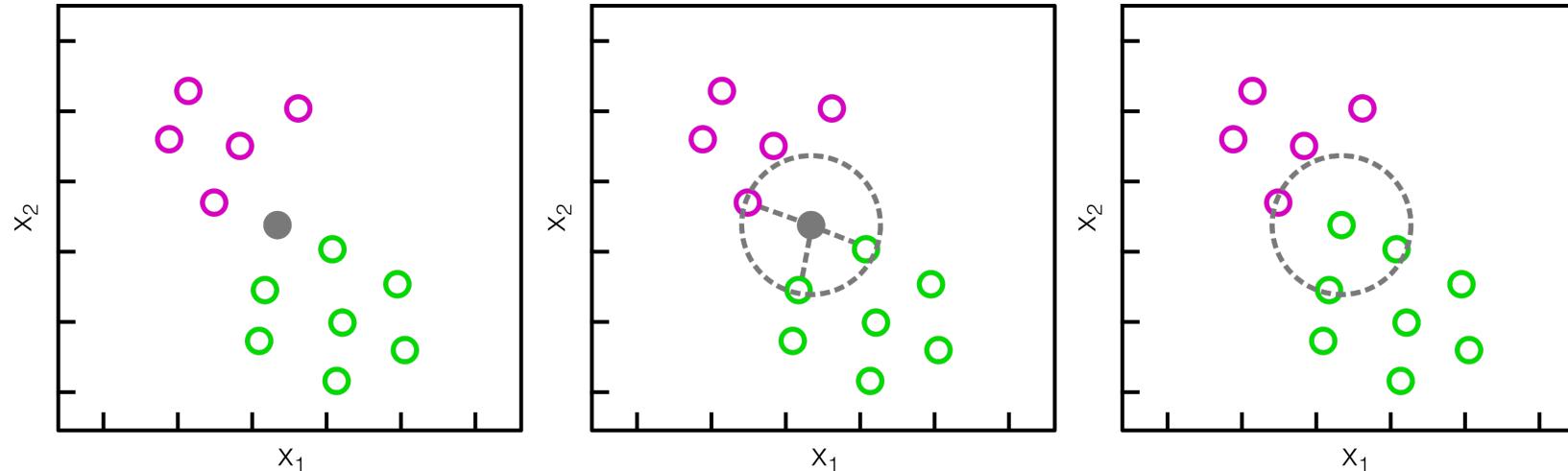


CLASIFICADOR KNN

CLASIFICADOR KNN

El clasificador de k vecinos más cercanos (KNN o k-NN), es un algoritmo que utiliza la proximidad de sus vecinos para hacer clasificaciones sobre la agrupación de un punto.

La idea se basa de la **suposición de que se pueden encontrar puntos similares cerca uno del otro en base a votación de pluralidad** (se elige la clase en función de la moda de la clase de sus vecinos). Este modelo no obtiene una salida de probabilidad, solo nos dice de qué clase es.



CLASIFICACIÓN

El clasificador de proximidad más cercano.

La idea se basa en base a votos de sus vecinos). Este

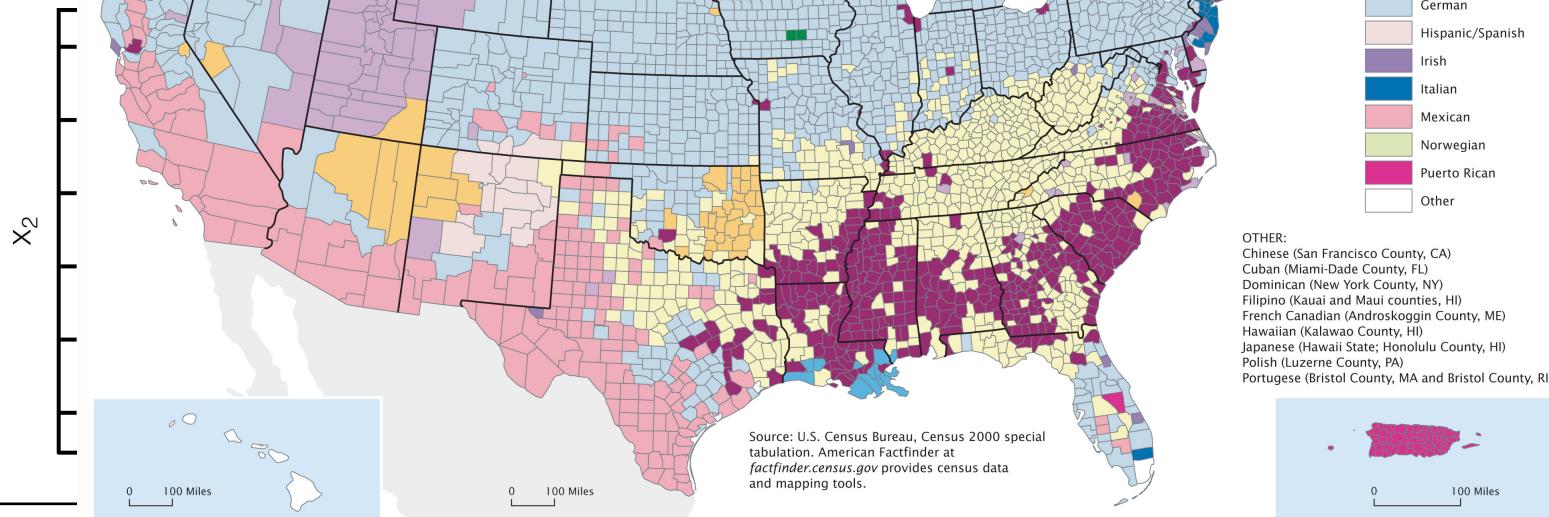
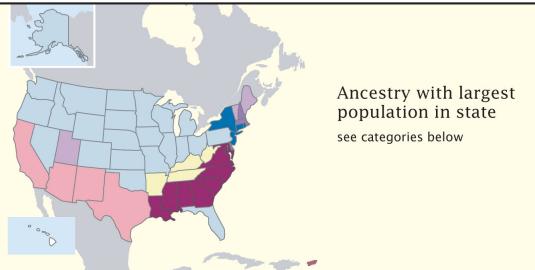


Figure 3.
Largest Ancestry: 2000

(Data based on sample. For information on confidentiality protection, sampling error, nonsampling error, and definitions, see www.census.gov/prod/cen2000/doc/sf3.pdf)



iliza la
unto.

rca uno del otro
use de sus
e clase es.

CLASIFICADOR KNN

Como vimos, este algoritmo se fija en la distancia entre observaciones.

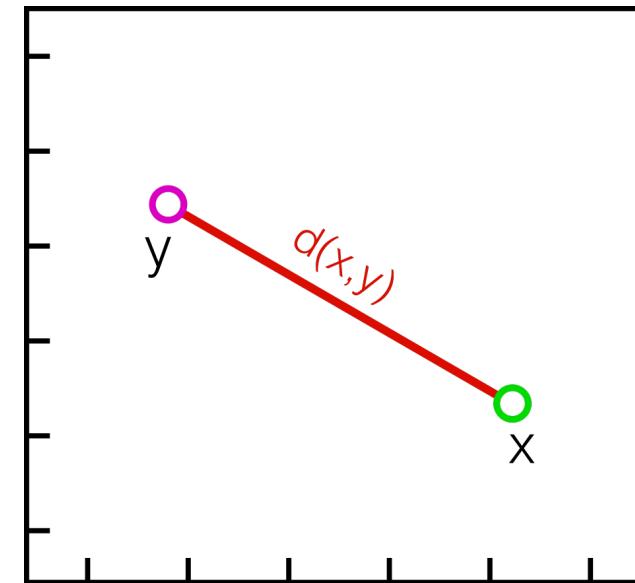
¿Ahora la pregunta es como medimos la distancia?

CLASIFICADOR KNN

Distancia euclídea (modulo 2)

Es la más conocida, es la mínima distancia (una recta) entre dos puntos en un espacio euclidiano. Es adecuada para datos numéricos continuos.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

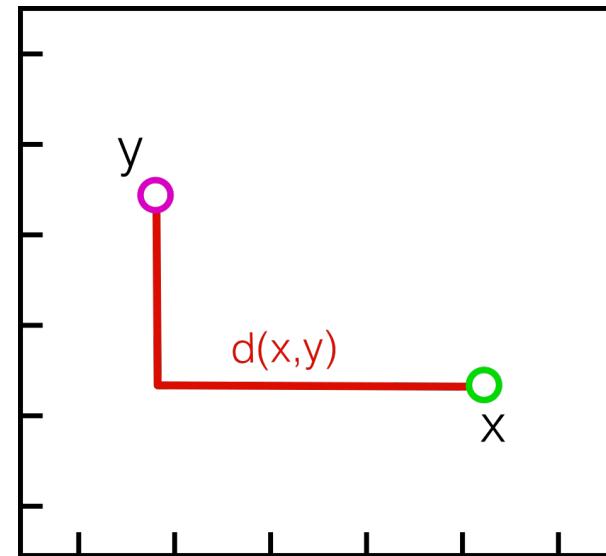


CLASIFICADOR KNN

Distancia de Manhattan (modulo 1)

Es la medida del valor absoluto entre dos puntos. Se conoce también como distancia taxi o de cuadra de ciudad, ya que mide distancias como en una ciudad. Es adecuada para datos que pueden tener correlaciones no lineales y no sigue la suposición de varianzas iguales en todas las dimensiones.

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

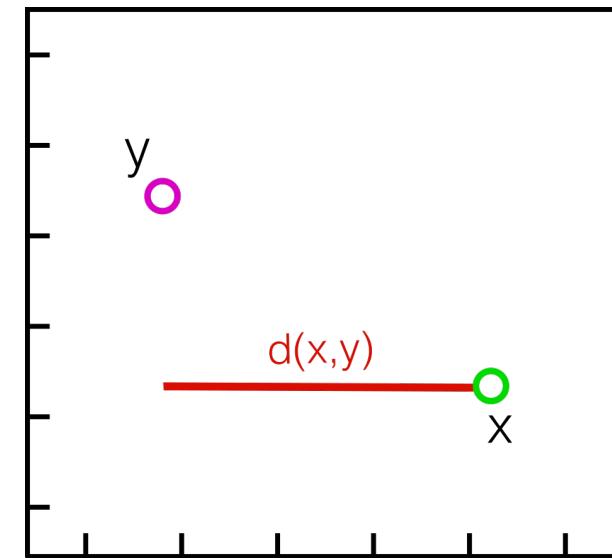


CLASIFICADOR KNN

Distancia de Chebyshev (modulo infinito)

Se calcula como la diferencia máxima entre las coordenadas de dos puntos. Es adecuada cuando las dimensiones son independientes y la distancia máxima es relevante.

$$d(x, y) = \max(|x_i - y_i|)$$

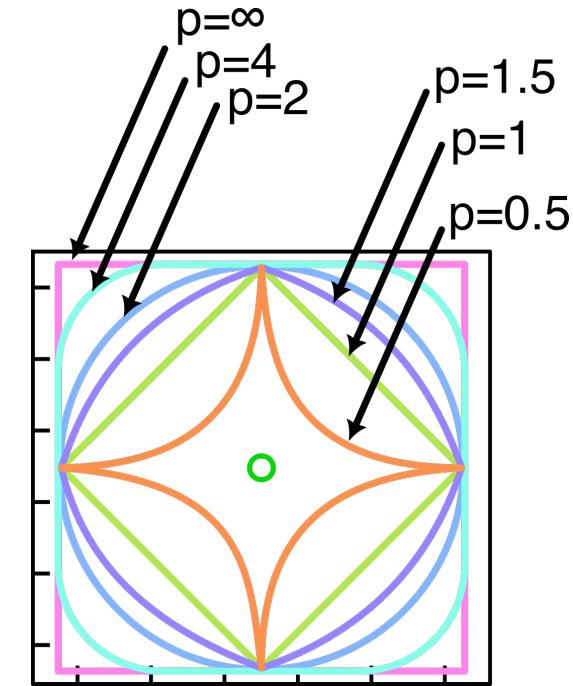


CLASIFICADOR KNN

Distancia de Minkowski

Es una medida generalizada que incluye las anteriores. Posee un parámetro, p , es la que permite variar el tipo de distancia.

$$d_p(x, y) = \left(\sum_{i=1}^n (x_i - y_i)^p \right)^{1/p}$$



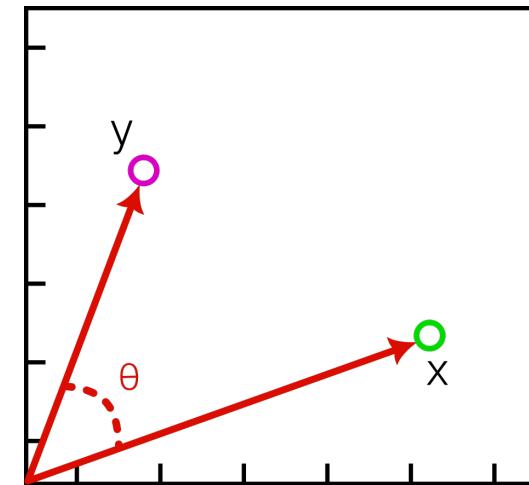
CLASIFICADOR KNN

Distancia Coseno

La similitud coseno mide la similitud entre dos vectores como el coseno del ángulo entre ellos, y la distancia es 1 menos la similitud coseno. Es adecuada para datos donde la magnitud de los vectores es irrelevante, pero si su orientación.

$$d_c(x, y) = 1 - S_c(x, y)$$

$$S_c(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \|y\|}$$



CLASIFICADOR KNN

Distancia de Jaccard

Se utiliza comúnmente en conjuntos o datos binarios. Mide la similitud entre dos conjuntos como el tamaño de su intersección dividido por el tamaño de su unión.

$$J(x, y) = \frac{|x \cap y|}{|x \cup y|}$$

$$d_J(x, y) = 1 - J(x, y)$$

CLASIFICADOR KNN

Distancia de Hamming

Se usa típicamente con vectores booleanos, en donde se mide la cantidad de elementos del vector que son diferentes entre sí.

$$\begin{array}{l} x \quad \boxed{1} \boxed{0} \boxed{\color{red}{1}} \boxed{0} \boxed{0} \boxed{\color{red}{0}} \boxed{0} \boxed{\color{red}{1}} \boxed{1} \\ y \quad \boxed{1} \boxed{0} \boxed{\color{red}{0}} \boxed{0} \boxed{1} \boxed{\color{red}{0}} \boxed{0} \boxed{\color{red}{1}} \end{array} \quad \left. \right\} d_H=3$$

CLASIFICADOR KNN

Dada la métrica de distancia, debemos definir el valor de k, que es quien define con cuantos vecinos se usará para determinar la clasificación de un punto.

Por ejemplo, si $k=1$, la observación se asignará a la misma clase de su vecino más cercano.

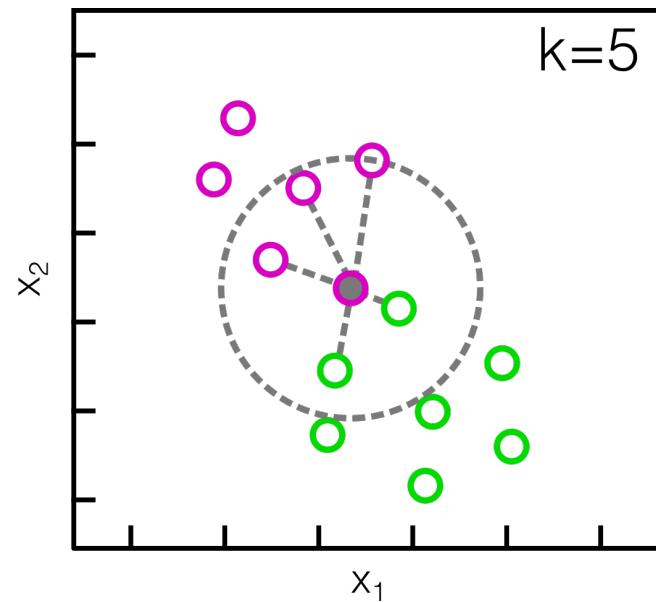
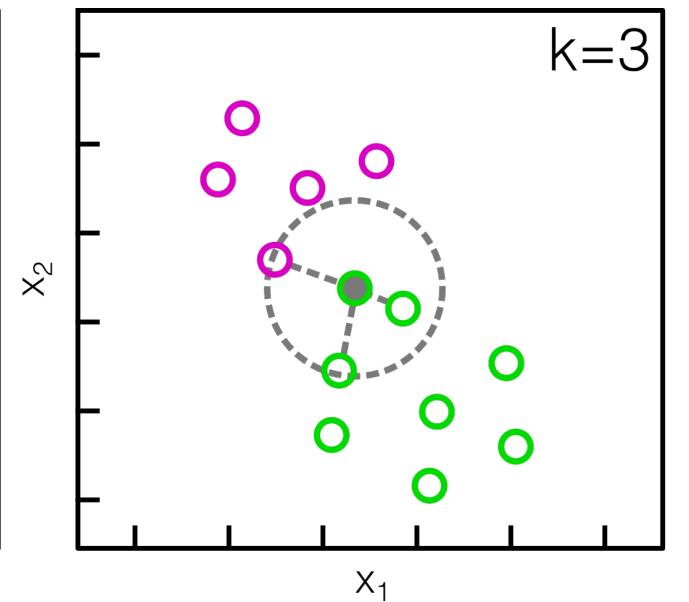
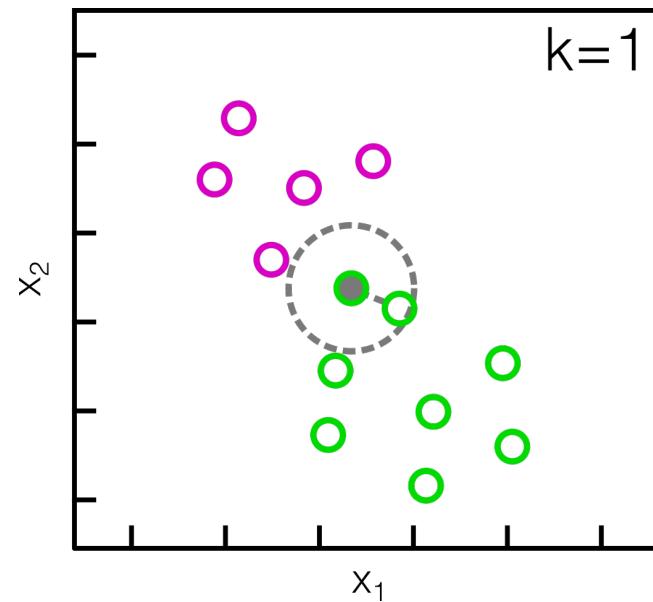
Definir k, el cual es un híper-parámetro junto al tipo de distancia elegida, **es un acto de equilibrio**.

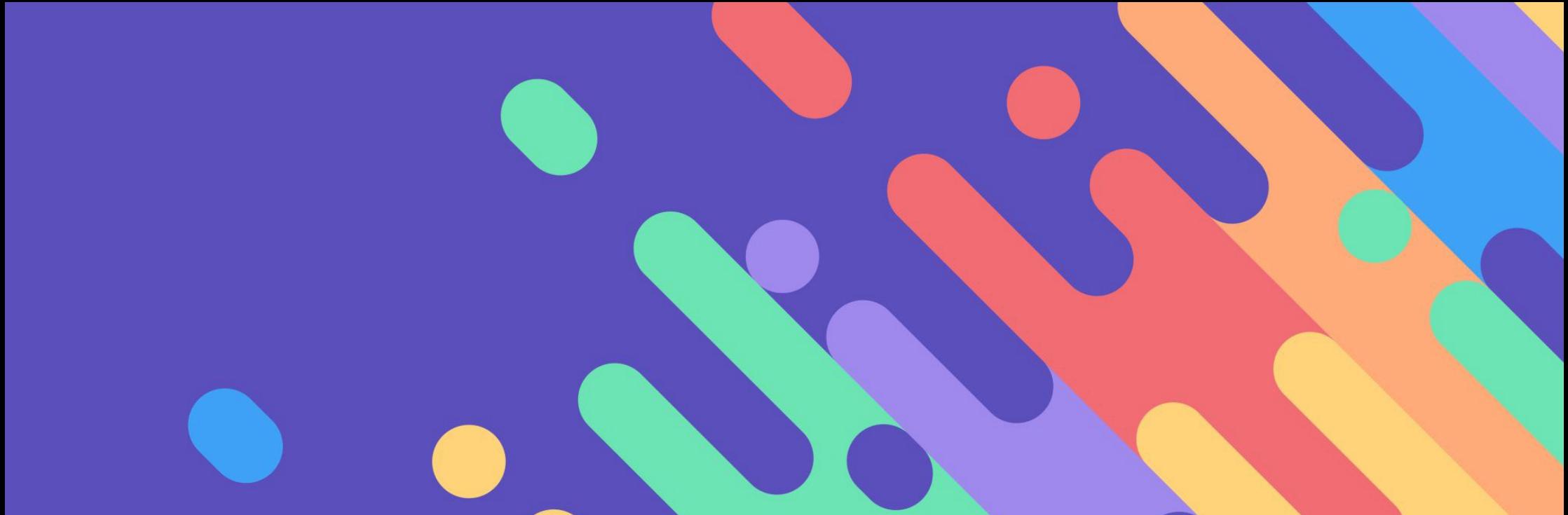
Valores bajos de k pueden tener una **varianza alta**, pero un **sesgo bajo**, y valores altos de k un **sesgo alto y poca varianza**.

En general, se recomienda tener un número impar para k para evitar empates en la clasificación.

Este algoritmo no tiene “entrenamiento” ya que debe guardar todo el dataset para evaluar a nuevos valores a que clase pertenece. Si el dataset de entrenamiento es muy grande, puede tener dificultades para almacenarse o ejecutarse.

CLASIFICADOR KNN

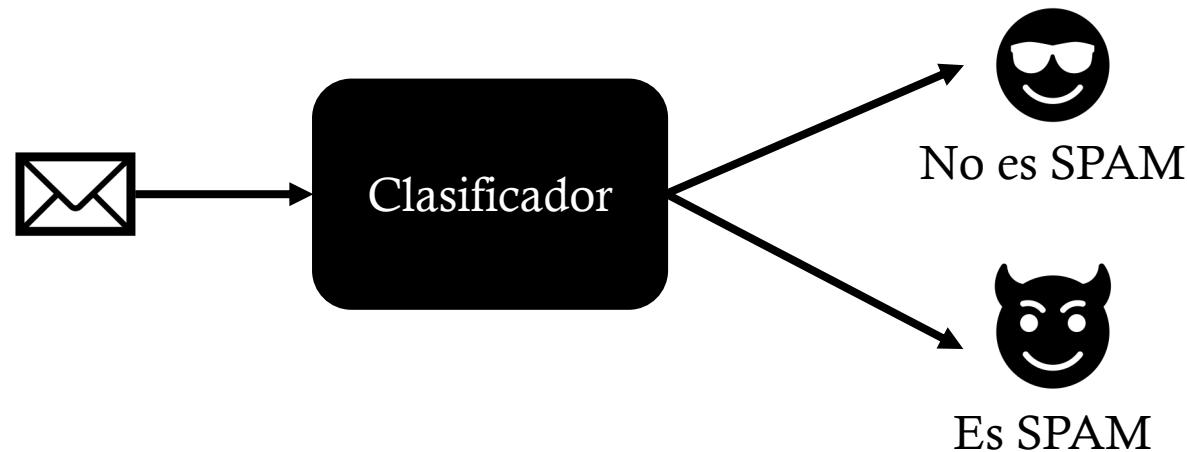




MÉTRICAS DE EVALUACIÓN

MÉTRICAS DE CLASIFICACIÓN

Supongamos que tenemos un modelo de clasificación encargado de medir si un correo es SPAM o no:



¿Como medimos la calidad de este clasificador? ¿Como sé que funciona bien?

MÉTRICAS DE CLASIFICACIÓN

¿Como medimos la calidad de este clasificador? ¿Como sé que funciona bien?

Uno piensa intuitivamente en tasa de aciertos. Pero los **SPAM** son mucho menos que los **no SPAM**, supongamos que tenemos una relación 1 a 1000.

Entonces un modelo que clasifica a todo como **no SPAM**, va a tener una tasa de acierto de:

99.9%

Entonces, ¿seguimos pensando que la tasa de acierto es una buena métrica?

MÉTRICAS DE CLASIFICACIÓN

Entonces, es importante para saber si el clasificador binario es bueno o malo, entender cómo se puede equivocar.

Supongamos que, si el clasificador dice que es SPAM, entonces la salida es positiva, y si no es negativo, con eso podemos tener los siguientes casos en comparación con el verdadero valor

		Valor verdadero	
		Verdadero positivo (TP)	Falso positivo (FP)
Salida del clasificador	Verdadero		
	Falso		

Esta estructura se llama **matriz de confusión**

MÉTRICAS DE CLASIFICACIÓN

Matriz de confusión

- **Verdadero positivo:** Es aquellas observaciones que clasificamos como 1 y que realmente eran 1.
- **Verdadero negativo:** Es aquellas observaciones que clasificamos como 0 y que realmente eran 0.
- **Falso positivo:** Es aquellas observaciones que clasificamos como 1 y que realmente eran 0. Este tipo de error se llaman de tipo I.
- **Falso negativo:** Es aquellas observaciones que clasificamos como 0 y que realmente eran 1. Este tipo de error se llaman de tipo II.

MÉTRICAS DE CLASIFICACIÓN

Matriz de confusión

El desempeño específico de clase también es importante en medicina y biología, donde los términos **sensibilidad** y **especificidad** caracterizan el desempeño de una prueba de detección:

- **Sensibilidad (tasa de verdaderos positivos):** Representa la capacidad del clasificador para detectar todos los casos positivos existentes en los datos.

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

- **Especificidad (Tasa de verdaderos negativos):** Indica la capacidad del clasificador para identificar correctamente los casos negativos.

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

MÉTRICAS DE CLASIFICACIÓN

Matriz de confusión

Volviendo a nuestro clasificador que dice que todo no es SPAM, tendríamos:

$$\text{Sensibilidad} = 0 \quad \text{Especificidad} = 1$$

La exactitud es la métrica que vimos, la tasa de aciertos:

$$\text{Exactitud} = \frac{TP + TN}{P + N} = 0.999$$

Pero cuando tenemos desbalance de clases conviene calcular la exactitud balanceada:

$$\text{Exactitud balanceada} = \frac{TPR + TNR}{2} = 0.5$$

MÉTRICAS DE CLASIFICACIÓN

Matriz de confusión

Volviendo a nuestro clasificador que dice que todo no es SPAM, tendríamos:

$$\text{Sensibilidad} = 0 \quad \text{Especificidad} = 1$$

La exactitud es la métrica que vimos, la tasa de aciertos:

$$\text{Exactitud} = \frac{TP + TN}{P + N} = 0.999$$

Pero cuando tenemos desbalance de clases conviene calcular la exactitud balanceada:

$$\text{Exactitud balanceada} = \frac{TPR + TNR}{2} = \boxed{0.5}$$

Nos dice que el clasificador está adivinando

MÉTRICAS DE CLASIFICACIÓN

Precisión y recuperación

Otras dos métricas muy importantes son **precisión** y **recuperación**, y estas juegan un rol importante cuando la clase positiva tiene más importancia que la negativa:

- **Precisión**: Se refiere a la proporción de casos positivos identificados correctamente por el clasificador con respecto a todos los casos que el clasificador etiquetó como positivos.

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

- **Recuperación**: Mide la proporción de casos positivos que el clasificador identificó correctamente con respecto a todos los casos positivos reales en los datos. En otras palabras, la recuperación indica la capacidad del clasificador para *recuperar* los casos positivos.

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

MÉTRICAS DE CLASIFICACIÓN

Precisión y recuperación

En general existe un trade-off entre estas dos métricas, si queremos mejorar una, lo vamos a hacer en pos de empeorar la otra.

Veamos ejemplos donde una métrica es más importante que la otra:

- **Precisión:** En nuestro clasificador de SPAM es mejor esta métrica, ya que queremos que nuestro clasificador cuando diga que es SPAM, realmente este seguro, ya que no queremos que el usuario pierda correos electrónicos importantes.
- **Recuperación:** Un clasificador de imágenes para detectar cáncer, la recuperación es más importante. Es fundamental que el modelo capture la mayor cantidad posible de casos de cáncer para garantizar que los pacientes no se pierdan un diagnóstico temprano y, por lo tanto, un tratamiento oportuno. Incluso si esto significa algunos falsos positivos.

MÉTRICAS DE CLASIFICACIÓN

Precisión y recuperación

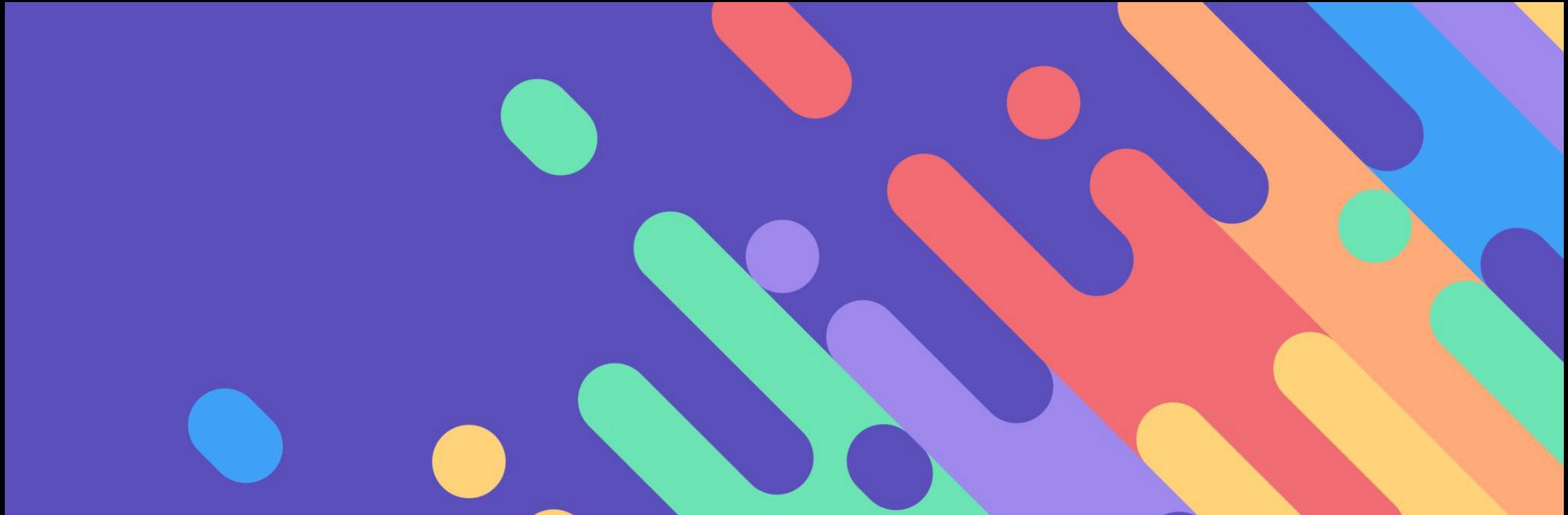
Hay veces que nos importa tener un balance de ambos casos, y para ello podemos usar el puntaje F_1 :

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

Y si queremos darle más importancia uno que a otro, podemos usar:

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

Si $0 < \beta < 1$, precisión es más importante, $\beta > 1$, recuperación es más pesado.



REGRESIÓN LOGÍSTICA

REGRESIÓN LOGÍSTICA

Supongamos que nos dicen que juega el Bayern Múnich versus Barcelona
¿Qué probabilidad decimos que tiene de ganar el Bayern Múnich?



¿Y si juega contra Sacachispas (Primera B Metropolitana - Argentina)?



REGRESIÓN LOGÍSTICA

Decir dado dos equipos de futbol, cuál va a ganar es un clasificador binario.

Pero, si queremos armar una página de apuestas, por ahí no queremos decir quién va a ganar, *sino dar la probabilidad de quien va a ganar*.

Entonces buscamos modelar no es el label, sino la probabilidad de que pertenezca a una clase en particular.

REGRESIÓN LOGÍSTICA

La **probabilidad** es una medida de la certidumbre de que ocurra un evento. Su valor es un número entre **0 y 1**, donde un evento imposible corresponde a cero y uno seguro corresponde a uno.

Es importante notar que, si un evento ya fue observado, la probabilidad puede ser solamente 0 o 1.



2



1

28/11/2000

$$P(\text{Gane Boca}) = 1$$

$$P(\text{Gane Real Madrid}) = 0$$

REGRESIÓN LOGÍSTICA

La **probabilidad** es una medida de la certidumbre de que ocurra un evento. Su valor es un número entre **0 y 1**, donde un evento imposible corresponde a cero y uno seguro corresponde a uno.

Pero si el evento no ocurrió, y queremos predecir, **podemos modelar la probabilidad**



12/04/2102



$$P(\text{Gane Boca}) = 0.22$$

$$P(\text{Gane Real Madrid}) = 0.78$$

REGRESIÓN LOGÍSTICA

Entonces una forma que se nos ocurre de modelar la probabilidad es usando una regresión lineal:

$$P(\text{Ganar Boca} = 1 | \mathbf{X}) = b + w_0 x_0 + \cdots + w_d x_d$$

En el caso de clasificación de dos clases:

$$\begin{aligned} P(y = 0 | \mathbf{X}) \\ P(y = 1 | \mathbf{X}) \end{aligned}$$

Pero, además, en el caso de dos clases: $P(y = 1 | \mathbf{X}) = 1 - P(y = 0 | \mathbf{X})$

Por lo que podemos simplificar la notación...

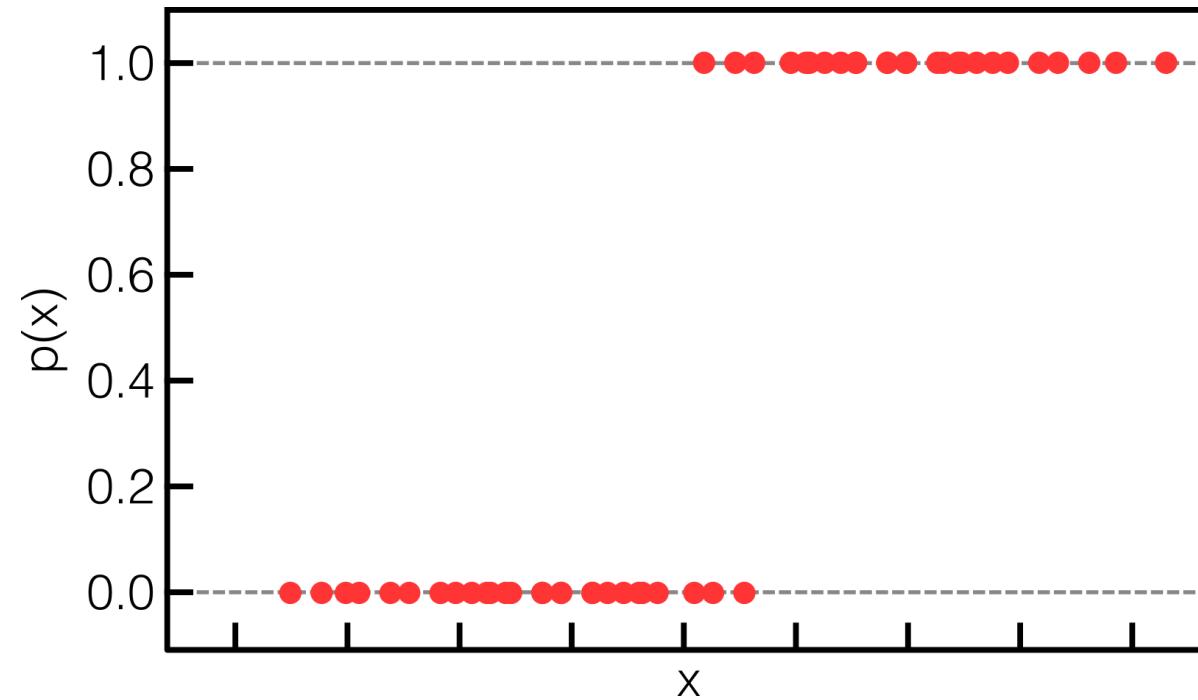
$$P(y = 1 | \mathbf{X}) = p(\mathbf{X})$$

Las probabilidades son valores que van entre 1 y 0.

Además, la hagamos más simple, el caso de un solo atributo: $p(x) = b + w_0 x_0$

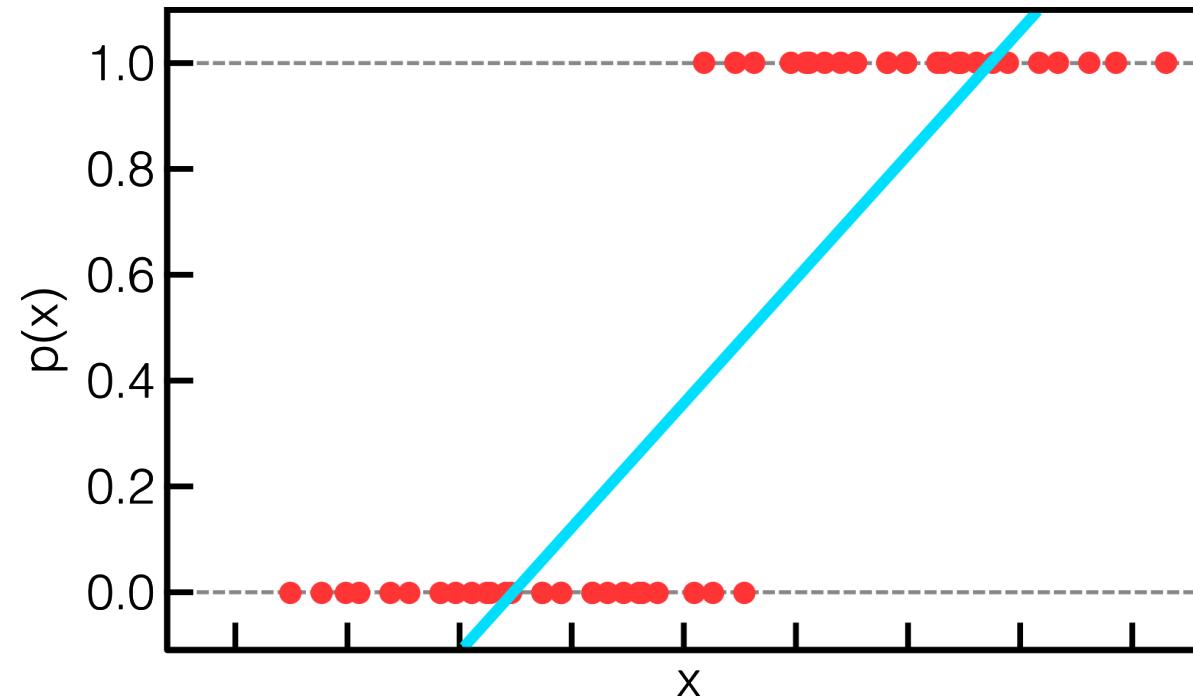
REGRESIÓN LOGÍSTICA

En un dataset, dado que son observaciones ya tenemos la probabilidad a la que pertenece.



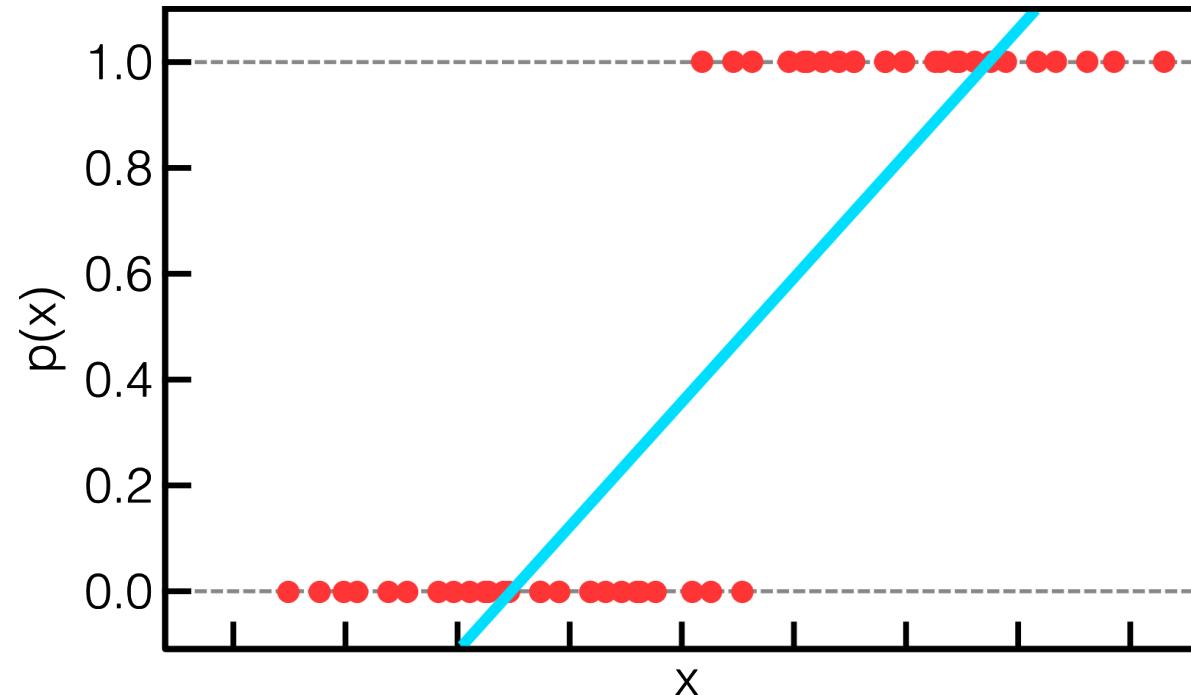
REGRESIÓN LOGÍSTICA

Podemos usar una regresión lineal para estimar la probabilidad $p(x) = b + w_0x$



REGRESIÓN LOGÍSTICA

En la gráfica se observa el problema de predecir usando **regresión lineal**. Dada la naturaleza de la función, hay valores en donde se obtienen $p(x) < 0$, o $p(x) > 1$. Esto va a ocurrir con cualquier regresión que de valores por fuera a 0 y 1.



REGRESIÓN LOGÍSTICA

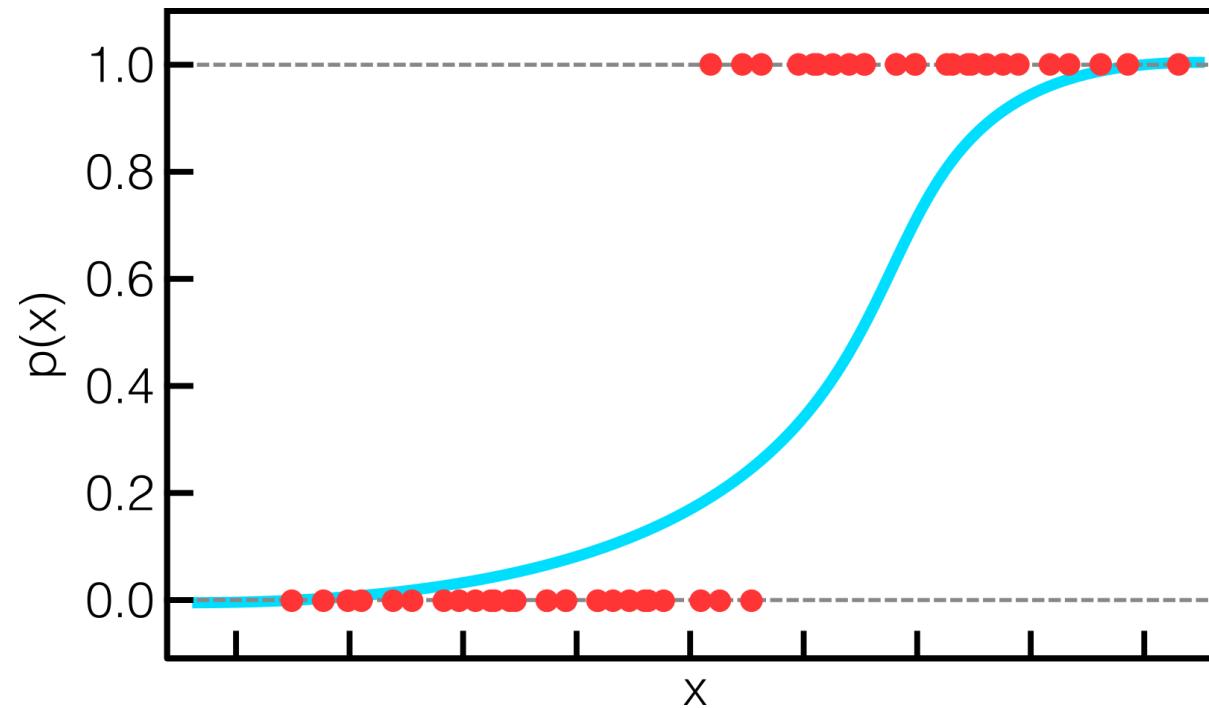
Para evitar esto, podemos modelar a la probabilidad usando una función que nos asegure que siempre tendremos valores entre 0 y 1.

En regresión logística, esto lo resolvemos usando una función sigmoide:

$$p(x) = \frac{e^{b+w_0x}}{1 + e^{b+w_0x}} = \frac{1}{1 + e^{-(b+w_0x)}}$$

REGRESIÓN LOGÍSTICA

Lo que visualmente se observa:

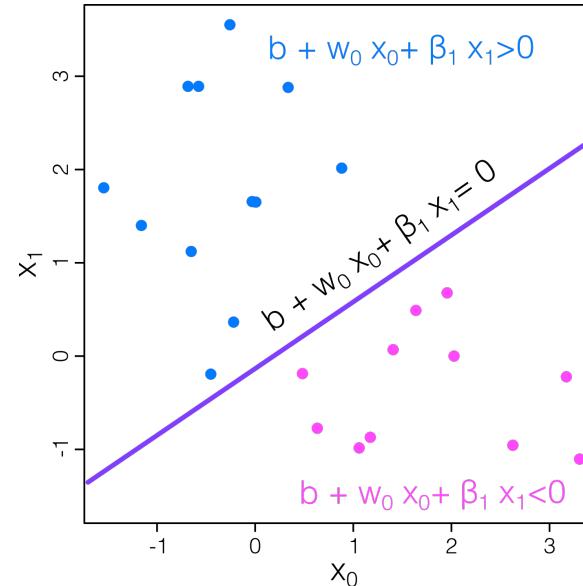


REGRESIÓN LOGÍSTICA

Esta regresión siempre va a formar una curva con forma sigmoidea. E independientemente del valor de x, siempre estará contenido entre 0 y 1.

Dado que el **corazón** de la regresión logística es una función lineal, si observamos la frontera de clasificación para un caso con dos atributos de entrada:

$$p(x) = \frac{1}{1 + e^{-(b+w_0x)}}$$



Que es lo que se conoce como un clasificador lineal

REGRESIÓN LOGÍSTICA - AJUSTE

Para buscar los coeficientes (b y w_0), es decir entrenar, lo hacemos realizándolo por **máxima verosimilitud**.

La intuición básica detrás de la máxima verosimilitud es que buscamos estimaciones para b y w_0 tales que la probabilidad prevista $p(x_i)$ de todos los valores del dataset, utilizando $p(x) = \frac{e^{b+w_0x}}{1+e^{b+w_0x}}$ corresponda lo más cerca posible al estado observado.

En otras palabras, tratamos de encontrar b y w_0 tales que al encontrar estas estimaciones se obtenga un número cercano a uno para la clase positiva, y lo más cercano a 0 para la clase negativa



CURVA ROC

CURVA ROC

Cuando vimos varias métricas de clasificadores binarios, siempre supusimos que nuestro clasificador nos da la salida 1 si es la clase positiva, 0 si es negativa, pero ahora tenemos un clasificador que nos da una probabilidad de que tan probable es que sea de la clase positiva.

De forma intuitiva, podemos definir que si la regresión logística nos devuelve un valor a mayor a 0.5, definimos como clase positiva, sino la negativa. De ahí podemos calcular **exactitud**, **precisión**, etc.

CURVA ROC

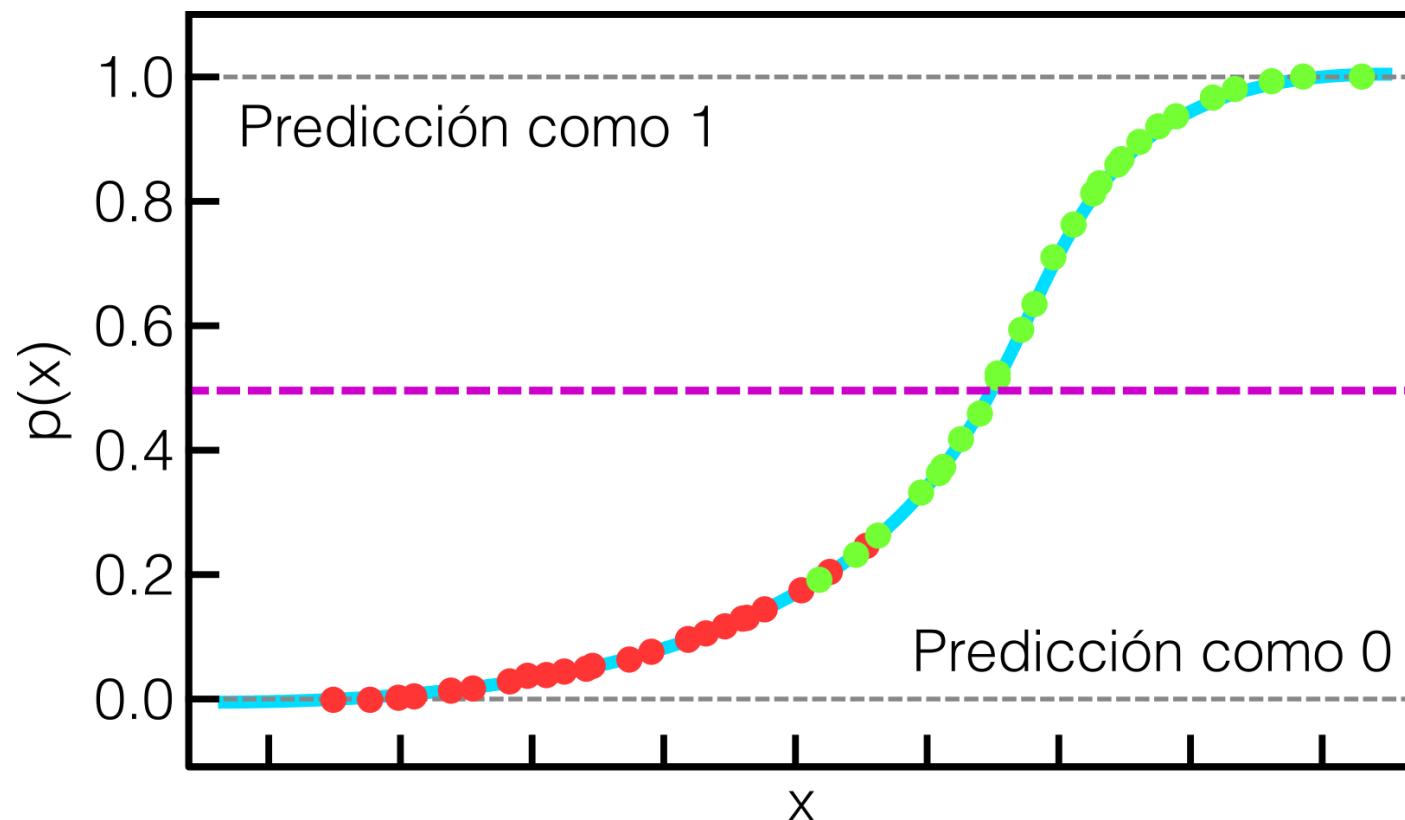
¿Pero por qué este valor?

CURVA ROC

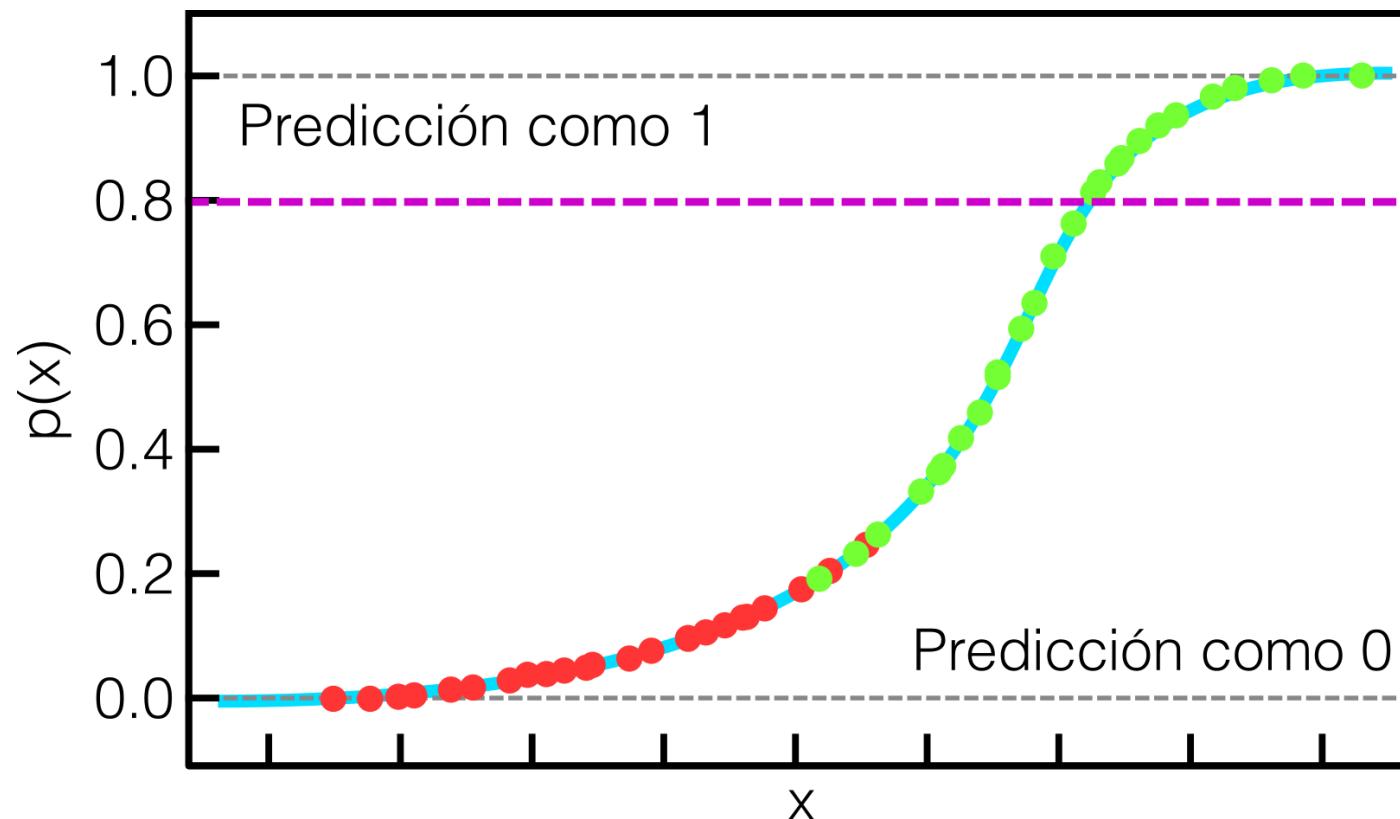
¿Pero por qué este valor?

Nos basamos en la idea de que el modelo nos da un valor de probabilidad. Pero nada impide de que el umbral pueda ser definido en diferentes valores, sobre todo si las clases están desbalanceadas.

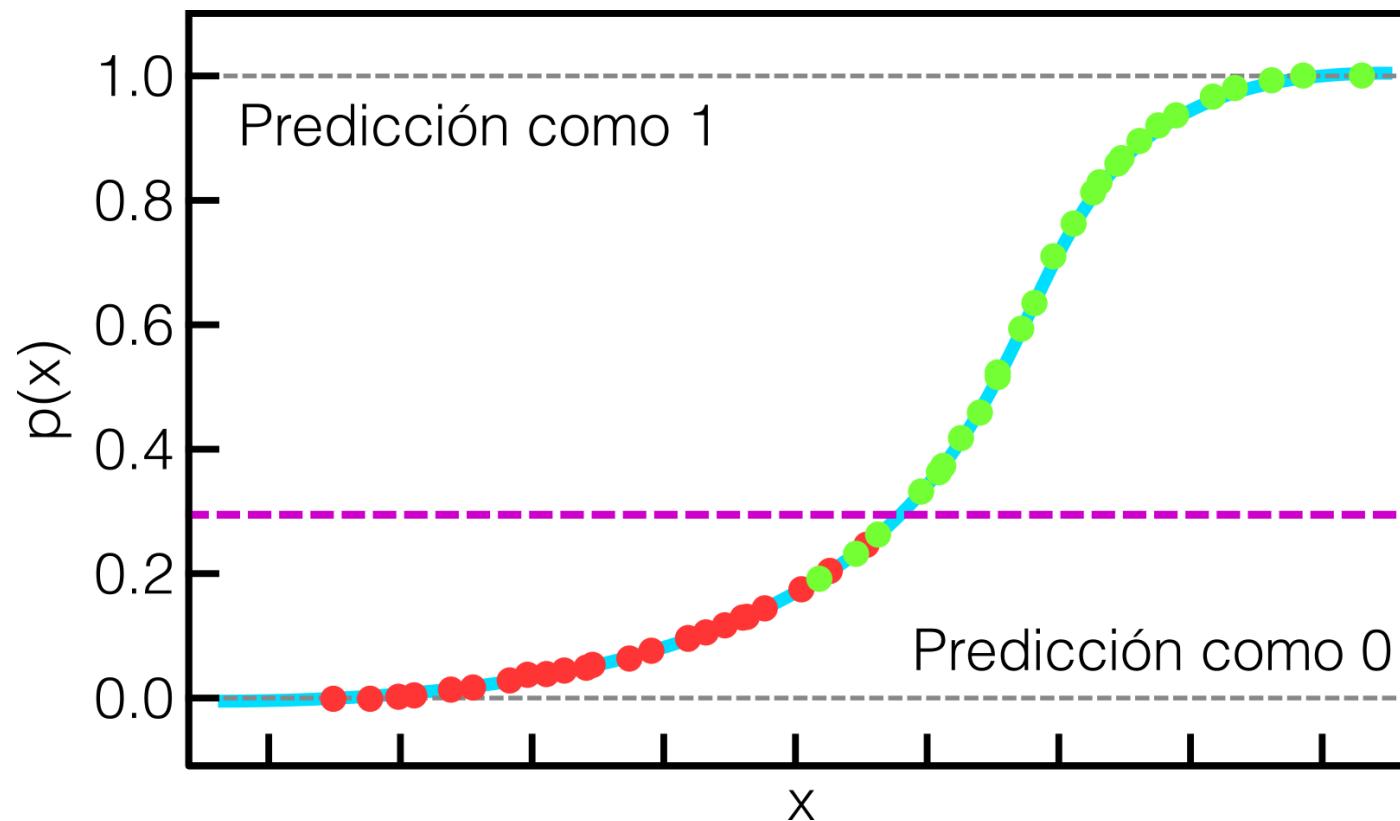
CURVA ROC



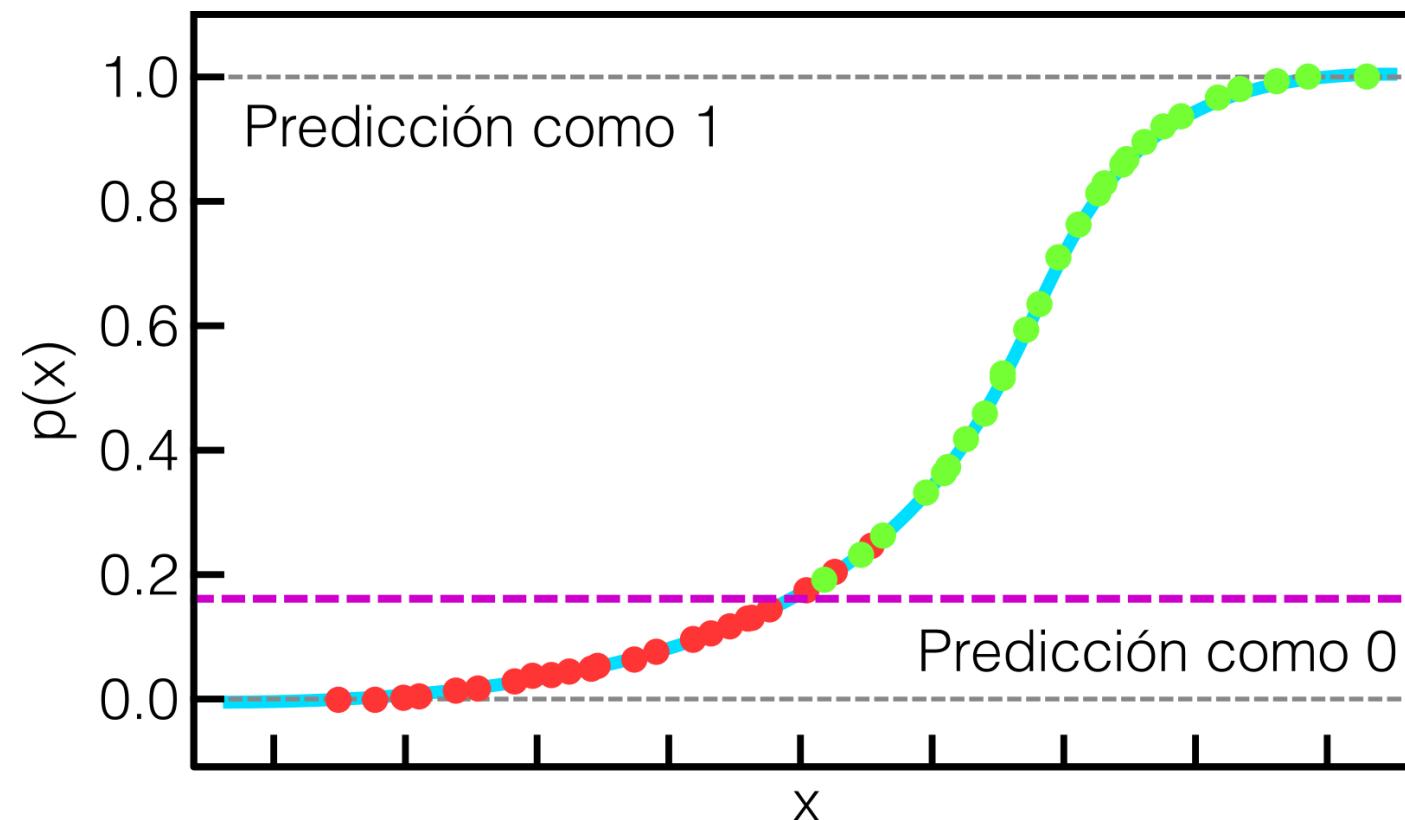
CURVA ROC



CURVA ROC



CURVA ROC

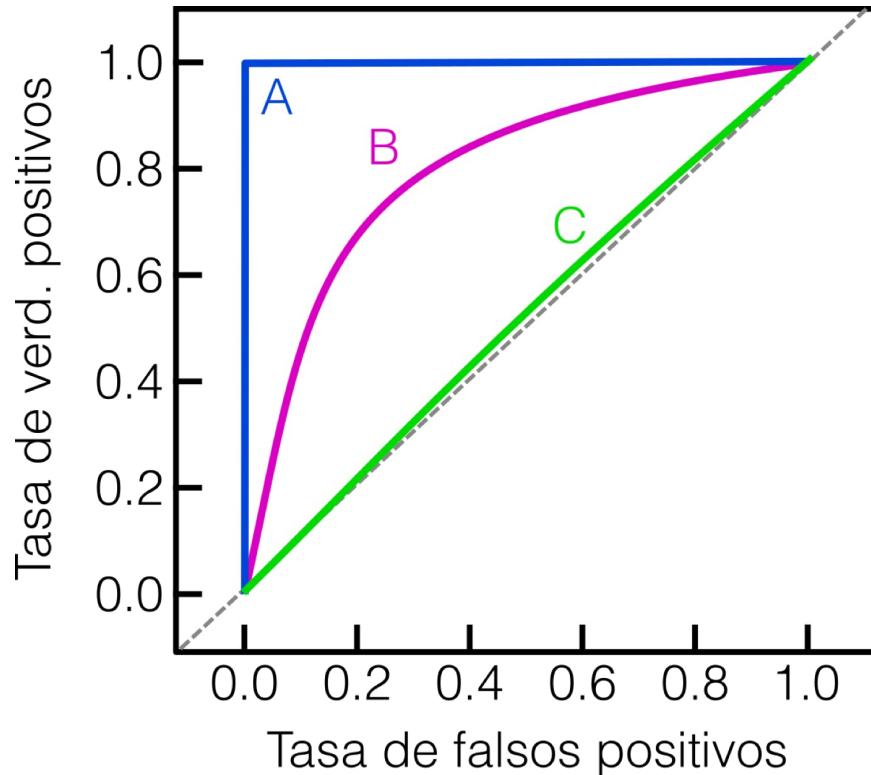


CURVA ROC

La curva ROC nos permite ver para todo valor de umbral, los dos tipos de errores. En el eje de las abscisas se utiliza la **tasa de falsos positivos** (o 1-especificidad) y en la ordenada **la tasa de verdadero positivos** (sensibilidad).

La curva se obtiene midiendo la sensibilidad y la especificad para todos los valores de umbrales de 0 a 1.

CURVA ROC



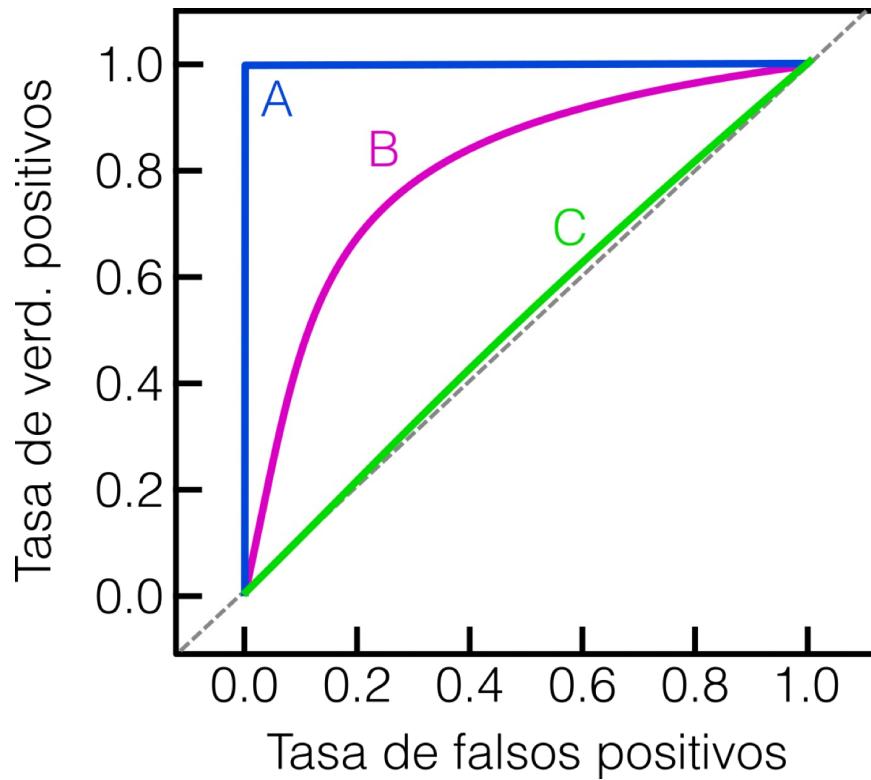
Siempre se arranca de umbral 1, donde la TPR es 0 y TFP es 0 y termina en 0 donde TVP es 1 y TFP es 1.

- **A** es la curva de un clasificador perfecto
- **B** es la curva de un clasificador estándar.
- **C** es la curva de un clasificador que adivina (el peor caso).

La curva ROC permite encontrar el valor umbral que mejor resultado dé.

Además, permite comparar clasificadores sin preocuparme del valor umbral elegido.

CURVA ROC

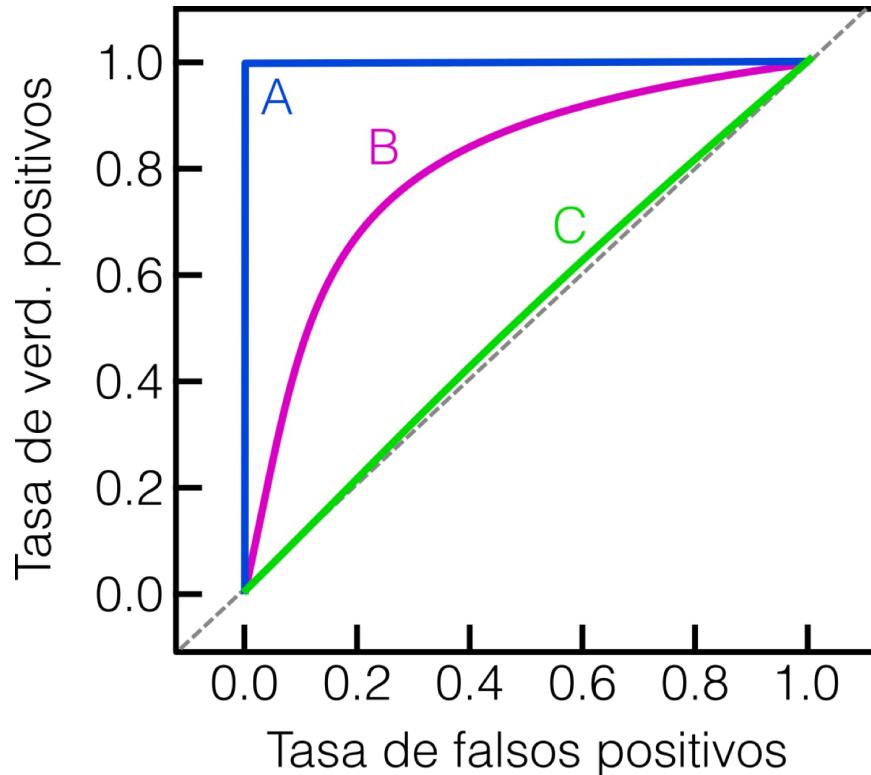


Si quiero bajar a una métrica a esta curva, podemos calcular el área bajo la curva (AUC).

- **A** tendrá un $AUC = 1$
- **B** tendrá un $0.5 < AUC < 1$
- **C** tendrá un $AUC = 0.5$

Si un clasificador tiene AUC menor a 0.5, ¿qué significa?

CURVA ROC

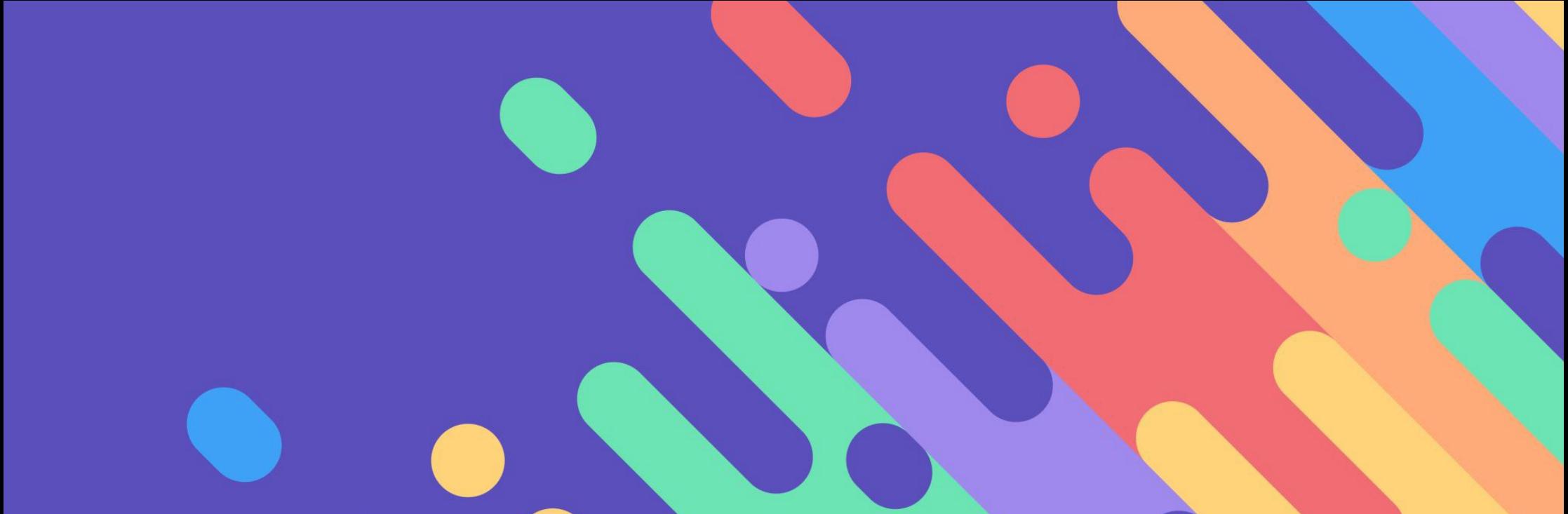


Si quiero bajar a una métrica a esta curva, podemos calcular el área bajo la curva (AUC).

- **A** tendrá un $\text{AUC} = 1$
- **B** tendrá un $0.5 < \text{AUC} < 1$
- **C** tendrá un $\text{AUC} = 0.5$

Si un clasificador tiene AUC menor a 0.5, ¿qué significa?

Significa que las clases están **invertidas**. Con solo cambiar las positivas por negativas, se soluciona.



REGRESIÓN LOGÍSTICA MULTI-CLASE

REGRESIÓN LOGÍSTICA MULTI-CLASE

Hasta ahora hemos visto clasificadores binarios, es decir, pueden predecir dos clases. Pero es posible extender a la regresión logística para que pueda predecir 3 o más clases (K).

Por ejemplo, si queremos clasificar correctamente un partido de futbol es entre 3 clases: **Gana Boca**, **Empatan** y **Gana Real Madrid**.

Creamos tres regresiones logísticas individuales, y para una observación particular tenemos:

$$[0.2, 0.55, 0.73]$$

Vemos que si sumamos a los tres nos dan mayor a uno ($0.2 + 0.55 + 0.73 = 1.48$), y por lo tanto perdemos lo que buscamos, que se mantenga una probabilidad

REGRESIÓN LOGÍSTICA MULTI-CLASE

Si normalizamos los tres valores con respecto a la suma, recuperamos esta habilidad:

$$\left[\frac{0.2}{1.48}, \frac{0.55}{1.48}, \frac{0.73}{1.48} \right]$$

$$[0.13, 0.38, 0.49]$$

Y nuestro clasificador combinado nos dice que, para esta observación, la observación es más probable es que gane Real Madrid. Cuando tenemos multi-clase, se elige la salida más grande.

Obsérvese además que esta salida tiene una forma de **one-hot encoding**.

$$[0, 0, 1]$$

REGRESIÓN LOGÍSTICA MULTI-CLASE

Este proceso es el que llamamos regresión logística multi-clase:

$$P(y = k|X) = \frac{e^{b_k + \mathbf{W}_k^T \mathbf{X}}}{\sum_k e^{b_{(k)} + \mathbf{W}_{(k)}^T \mathbf{X}}}$$

Se puede chequear que esta fórmula vuelve a la formula original si tenemos 2 clases, y se hace:

- $b = b_1 - b_0$
- $\mathbf{W} = \mathbf{W}_1 - \mathbf{W}_0$