Contenido

1. REGRESIÓN LOGÍSTICA

- Aprendizaje supervisado.
- Aprendizaje basado en modelo.
- Se corresponde con un modelo lineal generalizado.
- Realiza predicciones computando una suma ponderada de las características de entrada y sumándole una constante conocida como bias, ***pero se aplica una función logística (sigmoide) al resultado.***
- **Intenta predecir valores discretos,** normalmente:
    - 0 → clase negativa (por ejemplo, *no spam*)
    - 1 → clase positiva (por ejemplo, *spam*)

2. DIFERENCIA ENTRE REGRESIÓN LINEAL Y REGRESIÓN LOGÍSTICA

La principal diferencia con la regresión lineal radica en que, al resultado de la suma ponderada de las características de entrada, se le aplica una **función logística (sigmoide)**. Esta transformación es la que da nombre al algoritmo de **regresión logística**.

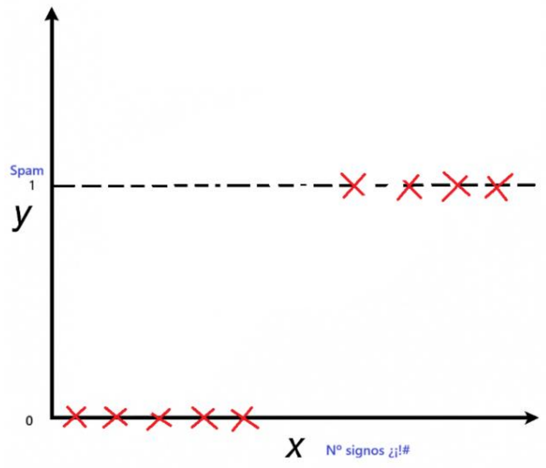
La característica que la distingue de forma más significativa de la regresión lineal es el **tipo de variable que pretende predecir**. Mientras que la regresión lineal se utiliza para predecir **valores continuos** dentro de un rango amplio, la regresión logística está diseñada para predecir **valores discretos**, normalmente acotados entre 0 y 1, lo que la hace especialmente adecuada para problemas de **clasificación binaria**, como determinar si un correo electrónico es *spam* o *no spam*.

3. FUNCIÓN HIPÓTESIS.

En regresión lineal, la función hipótesis se define para predecir **valores continuos** a partir de una combinación lineal de las variables de entrada.

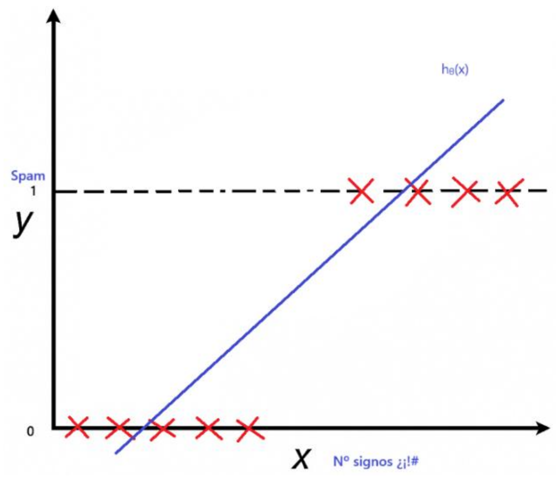
$$h_\theta(x)=\theta_0+\theta_1 x$$

Aplicamos esta función de regresión lineal a nuestros problemas de clasificación logística (valores discretos):



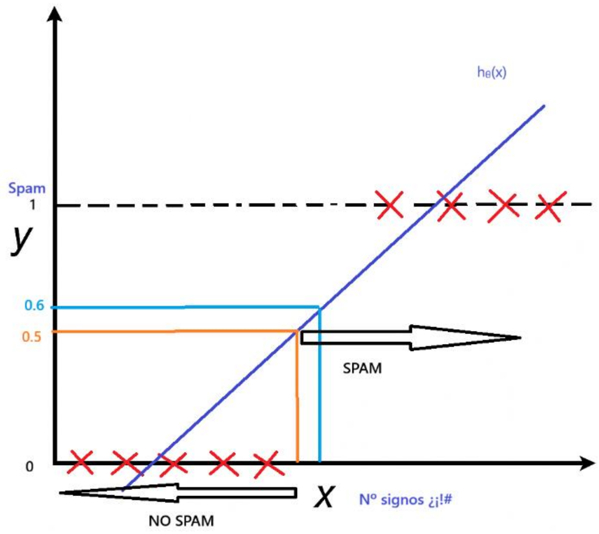
Si intentamos aplicar esta misma función de regresión lineal a un problema de **clasificación** (donde las salidas son valores discretos), lo primero que hacemos es ajustar el modelo a nuestro conjunto de datos: recorremos los puntos de entrenamiento, definimos la función hipótesis, aplicamos una función de error y utilizamos un algoritmo de optimización para minimizar dicho error y obtener los valores óptimos de los parámetros θ_0y θ_1.

Una vez encontrados estos valores óptimos, disponemos de un modelo basado en la función de hipótesis de la regresión lineal con el que podemos realizar predicciones.



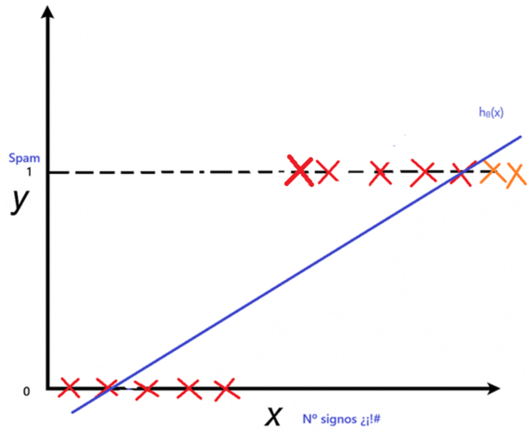
Por ejemplo, si realizamos una predicción para un determinado número de signos de interrogación y exclamación en un correo electrónico, el modelo podría devolver un valor aproximado de **0.6**. Este resultado es coherente, ya que la regresión lineal produce valores continuos. Sin embargo, este tipo de salida no es adecuada para un problema de clasificación, ya que el resultado debería estar **acotado a 0 o 1**.

Podríamos pensar en resolver este problema introduciendo un **límite de decisión (threshold)**:en este caso está señalado por las flechas en la gráfica. Si el valor predicho es mayor o igual que 0.5, asignamos la clase 1; en caso contrario, asignamos la clase 0. No obstante, aunque esta solución pueda parecer suficiente, existen aspectos importantes que no se están teniendo en cuenta.

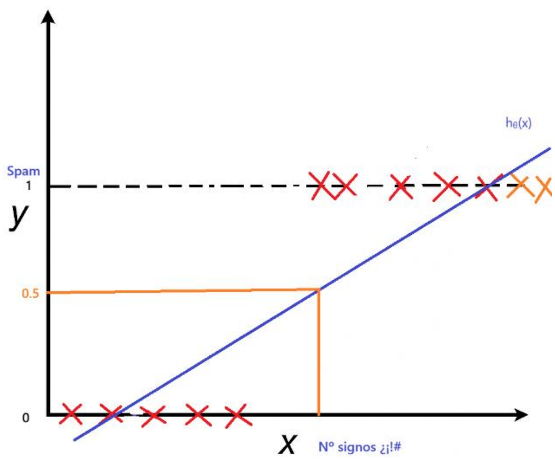


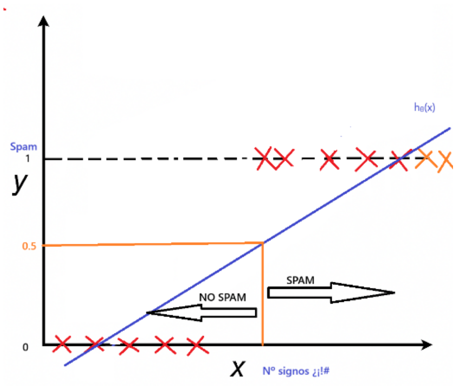
Pero el problema viene si por ejemplo, si en el conjunto de datos aparecen algunos correos adicionales situados más a la derecha del resto de ejemplos ya que estos valores pueden alterar significativamente la recta de regresión, desplazando el límite de decisión y provocando **clasificaciones incorrectas**

Imaginar que nos sale la siguiente función hipótesis:

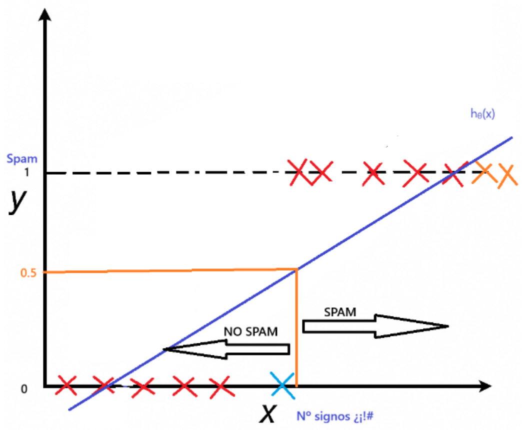


De la misma forma que hacíamos anteriormente ponemos el límite en 0.5 para determinar si el valor será de 0 o 1.



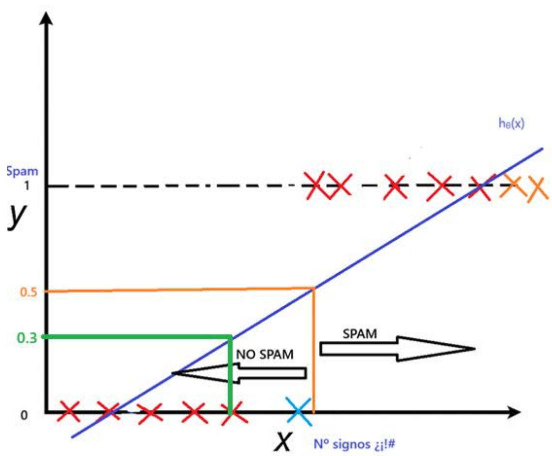


Al haberse hecho más plana a consecuencia de estos dos ejemplos, si nosotros tuviéramos un nuevo correo que cayese por aquí (color azul):



En este caso, el modelo indicaría que, al situarse a la izquierda del límite de decisión, el correo no es *spam*. Sin embargo, si atendemos al número de signos de interrogación y exclamación, dicho correo debería clasificarse como *spam*, ya que sigue claramente la tendencia general de los datos.

Esto implica que el límite de decisión debería estar desplazado más a la izquierda, aproximadamente en torno a 0.3. Al no cumplirse esta condición, el modelo realizaría **predicciones incorrectas**, lo que pone de manifiesto que la función de regresión lineal no es adecuada para problemas de regresión logística o clasificación.



4. CONSTRUCCIÓN DEL MODELO

Como hemos visto, la regresión lineal no es adecuada para problemas de clasificación. Por ello, la nueva función hipótesis que necesitamos debe garantizar que las predicciones del modelo estén **acotadas entre 0 y 1**, tal y como requiere un problema de **clasificación binaria**.

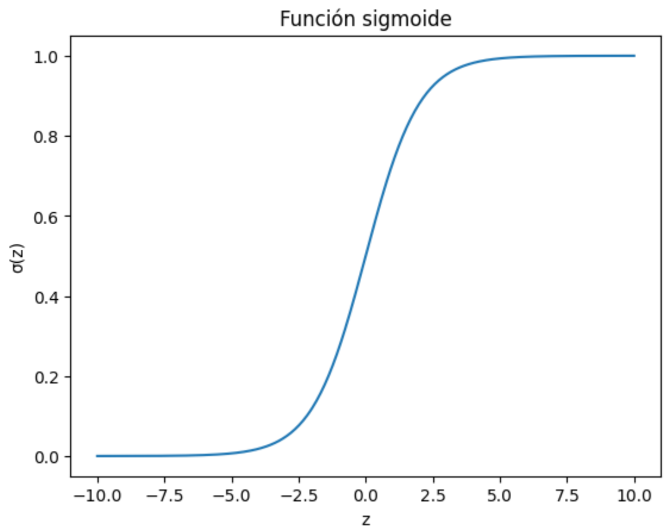
Este objetivo se consigue aplicando a la combinación lineal de las variables de entrada una **función de activación**, conocida como **función sigmoide**, que transforma cualquier valor real en un valor comprendido dentro de ese intervalo.

$$0 \leq h_\Theta(x) \leq 1$$

$$h_\Theta(x) = g(\theta_0 + \theta_1 x)$$

$$g(z) = \frac{1}{1+e^{-z}} \quad \text{Función sigmoide}$$

La función sigmoide presenta una forma característica en *S*. Su valor está siempre comprendido entre 0 y 1, toma el valor 0.5 cuando la entrada es 0, tiende a 0 para valores negativos grandes y tiende a 1 para valores positivos grandes. Esta propiedad permite interpretar su salida como una probabilidad.



Por tanto, la nueva función hipótesis se obtiene al **aplicar la función sigmoide a la combinación lineal propia de la regresión lineal**. De este modo, el modelo transforma la salida continua de la regresión lineal en un valor comprendido entre 0 y 1.

A modo de ejemplo, para un problema con una sola característica de entrada, la función hipótesis se expresa de la siguiente forma:

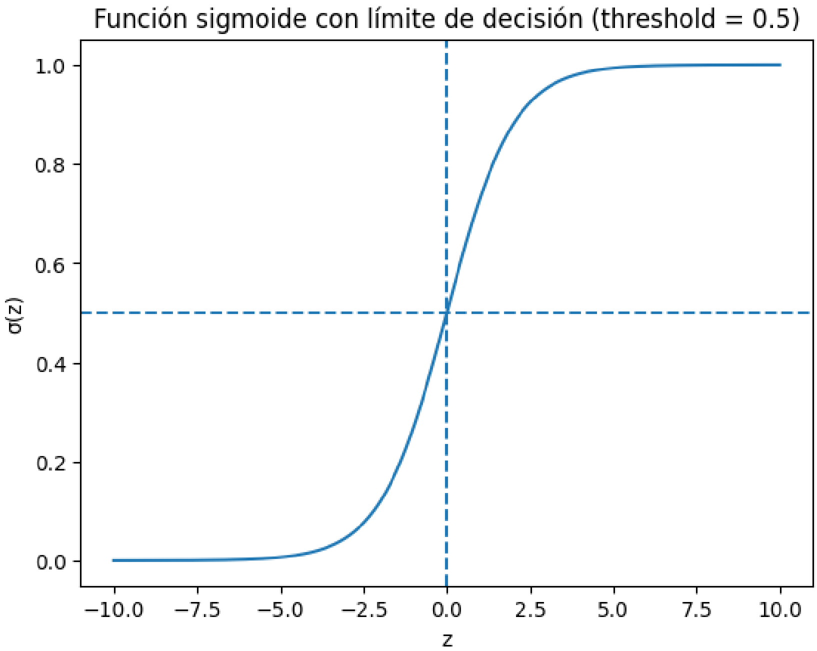
$$\boldsymbol{h_\Theta(x) = \frac{1}{1+e^{-(\theta 0+\theta 1 x 1)}}}$$

5. INTERPRETACIÓN DE LA FUNCIÓN HIPÓTESIS

La función hipótesis de la regresión logística devuelve la **probabilidad de que la variable de salida y sea igual a 1** para una entrada concreta. Es decir, indica la probabilidad de que un determinado ejemplo pertenezca a la **clase positiva** (por ejemplo, que un correo sea *spam*).

Como resultado, el modelo produce un **valor continuo comprendido entre 0 y 1**, lo que evita los problemas derivados de valores fuera de rango o de datos anómalos que podrían distorsionar la función hipótesis.

No obstante, dado que el objetivo final es realizar una **clasificación discreta**, el modelo debe asignar una clase concreta. Para ello se utiliza un **límite de decisión (threshold)**: si la probabilidad es mayor o igual que 0.5, se asigna la clase $y = 1$; en caso contrario, se asigna la clase $y = 0$.



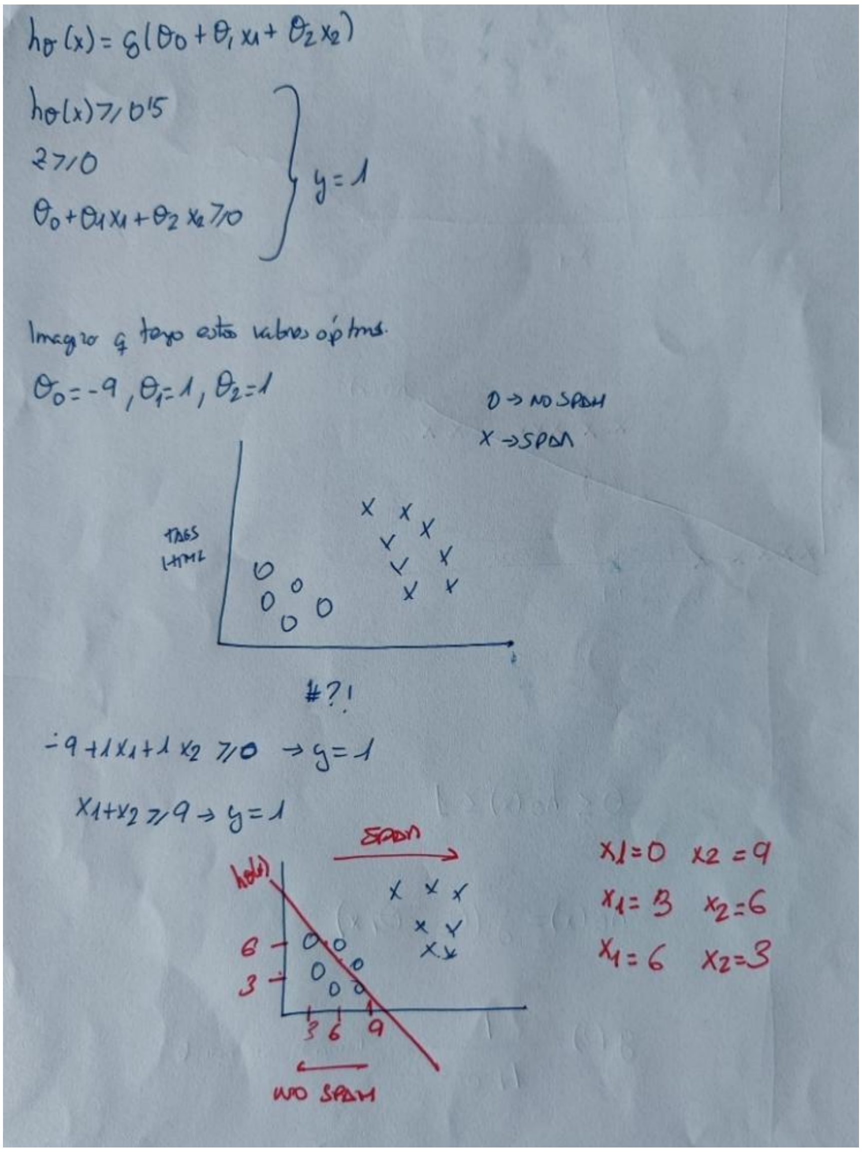
En esta gráfica representa la función sigmoide junto con el **límite de decisión (threshold)**. La línea horizontal discontinua marca el valor **0.5**, que es el umbral habitual en regresión logística.

El punto donde la sigmoide corta este valor corresponde a **z = 0**, que define la frontera de decisión del modelo.

- Valores por encima del threshold se clasifican como **clase 1**.
- Valores por debajo del threshold se clasifican como **clase 0**.

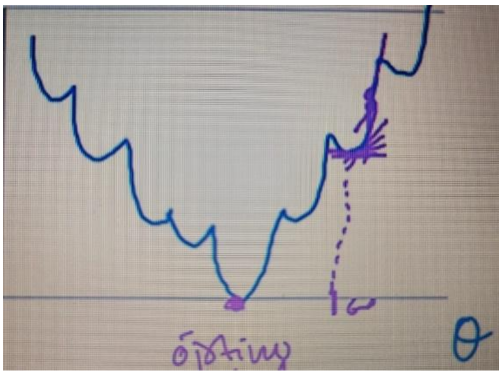
El modelo no decide directamente 0 o 1; primero calcula una probabilidad y después el threshold convierte esa probabilidad en una clase.

Ejemplo:



6. FUNCIÓN DE COSTE

La función de coste utilizada en la regresión lineal no es adecuada para la regresión logística, ya que al combinarse con la función sigmoide da lugar a una función de error **no convexa**, lo que provoca la aparición de **mínimos locales** y dificulta el proceso de optimización.



- Un **mínimo local** es un punto de una función en el que el valor de la función es menor que el de los puntos cercanos, pero **no necesariamente es el menor valor de toda la función**.
- Una **función convexa** tiene una forma similar a un cuenco abierto hacia arriba. Su propiedad fundamental es la siguiente: **Toda función convexa tiene un único mínimo, y ese mínimo es global.**

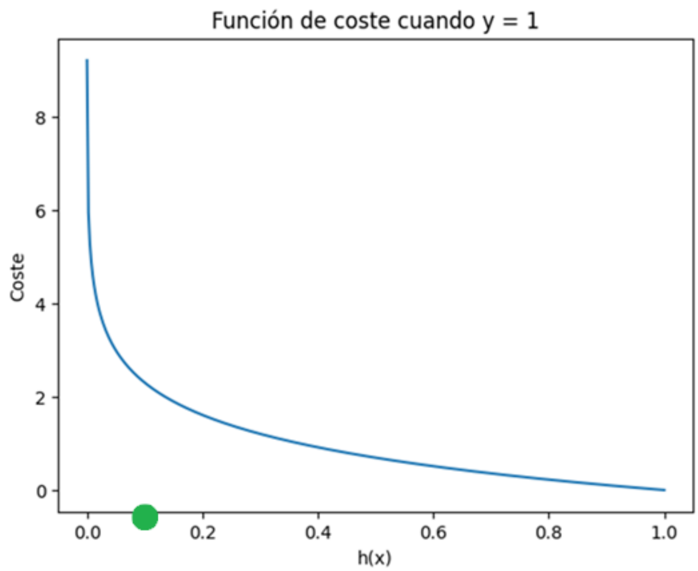
Por este motivo, es necesario definir una **nueva función de coste específica** para la regresión logística.

De forma intuitiva, la función de coste debe cumplir las siguientes condiciones:

- El **coste debe ser alto** cuando el modelo realiza una mala predicción.
- El **coste debe ser bajo** cuando el modelo realiza una buena predicción.

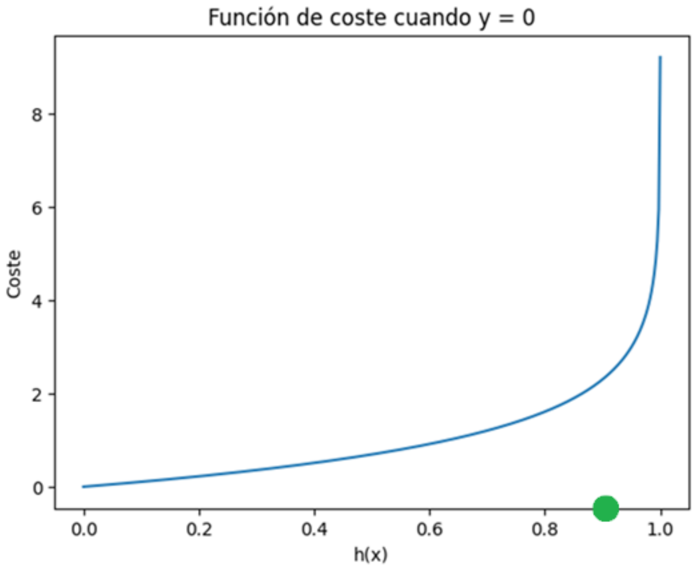
Para entenderlo mejor, pensemos en el proceso de entrenamiento del modelo. Partimos de un conjunto de datos de entrenamiento y de unos valores iniciales (normalmente aleatorios) para los parámetros del modelo θ_0y θ_1. A partir de ellos, el modelo realiza predicciones y se compara cada predicción con la etiqueta real del conjunto de datos. En función del error cometido, los parámetros se van ajustando para mejorar el rendimiento del modelo.

Vamos a hacerlo para el conjunto de datos que hemos puesto la etiqueta con el valor de 1, cogemos nuestra x (por ejemplo los signos de interrogación y exclamación) e inicializamos de manera aleatoria Θ_0 y Θ_1 y realizamos una predicción. Imaginaros que esa predicción es 0.1, está mal porque se está aproximando muchísimo a cero cuando realmente debería aproximarse lo máximo posible a uno. Con lo cual el valor de Θ_0 y Θ_1 es un valor muy malo por lo que no está bien ajustado y no está realizando buenas predicciones y el coste de nuestra función de coste es muy alto. Representado gráficamente debería ser:



Sin embargo, a medida que se va aproximando a cero se va reduciendo el coste hasta que llega un punto en el que es prácticamente uno y el coste es prácticamente cero. Con lo cual esta función que represento en la gráfica podría ser una buena función de coste.

Lo mismo pasa para el caso de y=0 por ejemplo para el caso del ejemplo que hemos predecido un 0.9 el valor va a ser alto.



Pero a medida que nos acercamos a cero el coste es prácticamente cero por lo que es una buena función de coste.

Por lo que la función de coste va a tener dos partes, una para cuando y=1 y otra para cuando y=0. Básicamente la función se corresponde con lo que hemos visto y su función sería:

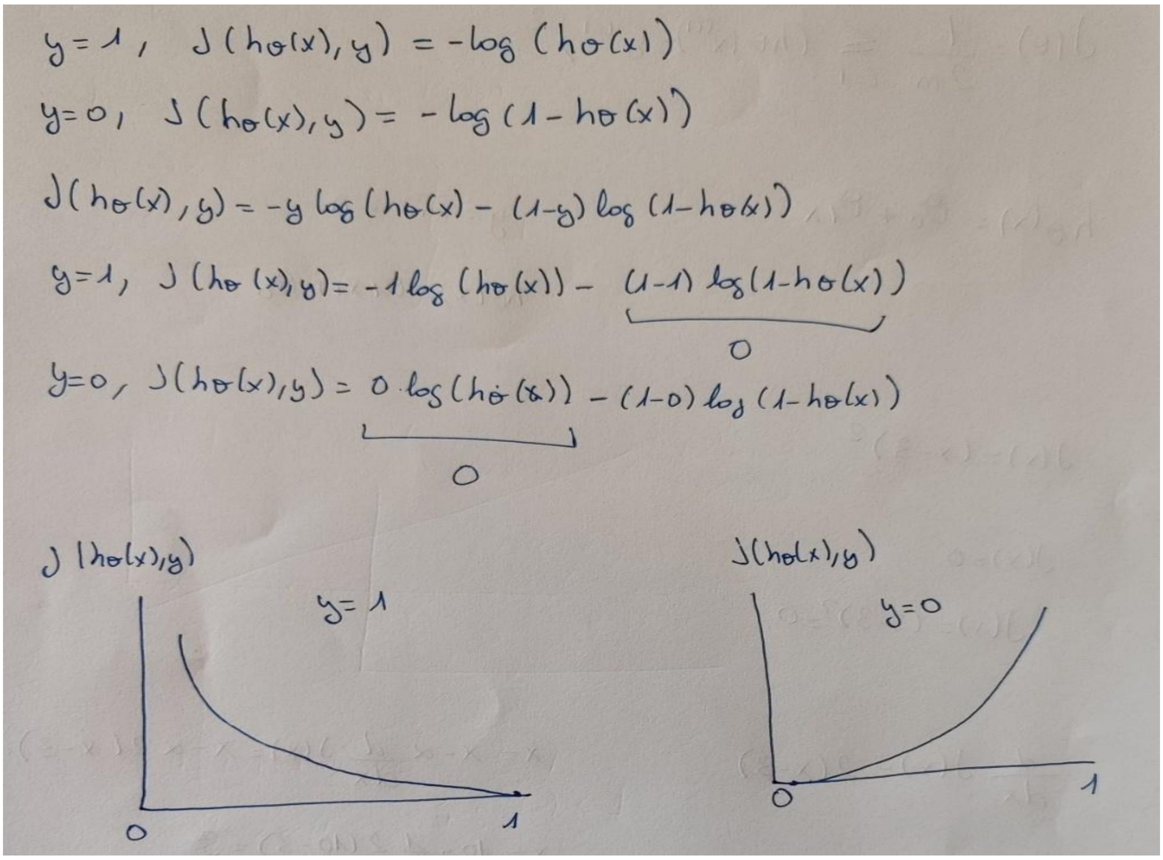
Para y=1, $J(h_\Theta(x),y) = -\log(h_\Theta(x))$

Para y=0, $J(h_\Theta(x),y) = -\log(1-h_\Theta(x))$

Ambas expresiones se pueden unificar en una sola función:

$$J(h_\Theta(x),y) = -y\log(h_\Theta(x)) - (1-y)\log(1-h_\Theta(x))$$

Esta función que es una de las funciones más importantes que usaremos también para redes neuronales, se simplifica así dándole valores :



Ventajas de esta función de coste

Esta función presenta varias ventajas importantes:

- Es **convexa**, por lo que garantiza la existencia de un **mínimo global**.
- Penaliza fuertemente las predicciones incorrectas.
- Facilita el uso de algoritmos de optimización como el **descenso por gradiente**.
- Es la base de funciones de coste utilizadas en **redes neuronales** y otros modelos de clasificación.

7. LÍMITE DE DECISIÓN (THRESHOLD)

Un **límite de decisión (threshold)** es el **valor a partir del cual una probabilidad se transforma en una clase concreta** en un problema de clasificación.

En el contexto de la **regresión logística**, el modelo no devuelve directamente una clase (0 o 1), sino una **probabilidad** comprendida entre 0 y 1. El límite de decisión indica **cómo convertir esa probabilidad en una decisión final**.

El **límite de decisión** es el valor que se utiliza para decidir a qué clase pertenece un ejemplo:

- Si la probabilidad predicha es **mayor o igual** que el threshold → se asigna la **clase 1**.
- Si la probabilidad predicha es **menor** que el threshold → se asigna la **clase 0**.

Habitualmente, el valor más utilizado es: $\mathbf{threshold = 0.5}$

Ejemplo práctico

Supongamos que un modelo de regresión logística devuelve los siguientes valores:

- $h\Theta(x) = 0.7$→ como $0.7 \geq 0.5$, el ejemplo se clasifica como **clase 1**.
- $h\Theta(x) = 0.3$→ como $0.3 < 0.5$, el ejemplo se clasifica como **clase 0**.

Por qué se usa un threshold?

Porque el modelo produce **probabilidades**, pero el problema requiere una **decisión discreta**. El threshold actúa como una **frontera de decisión** entre clases.

¿Siempre es 0.5?

No necesariamente. El valor del límite de decisión puede ajustarse según el problema:

- **Threshold más bajo (por ejemplo, 0.3)** → más ejemplos clasificados como clase 1 → se reduce el número de falsos negativos
- **Threshold más alto (por ejemplo, 0.7)** → menos ejemplos clasificados como clase 1 → se reduce el número de falsos positivos

Por ejemplo, en detección de *spam* o diagnóstico médico, el threshold puede modificarse según el tipo de error que se quiera evitar.