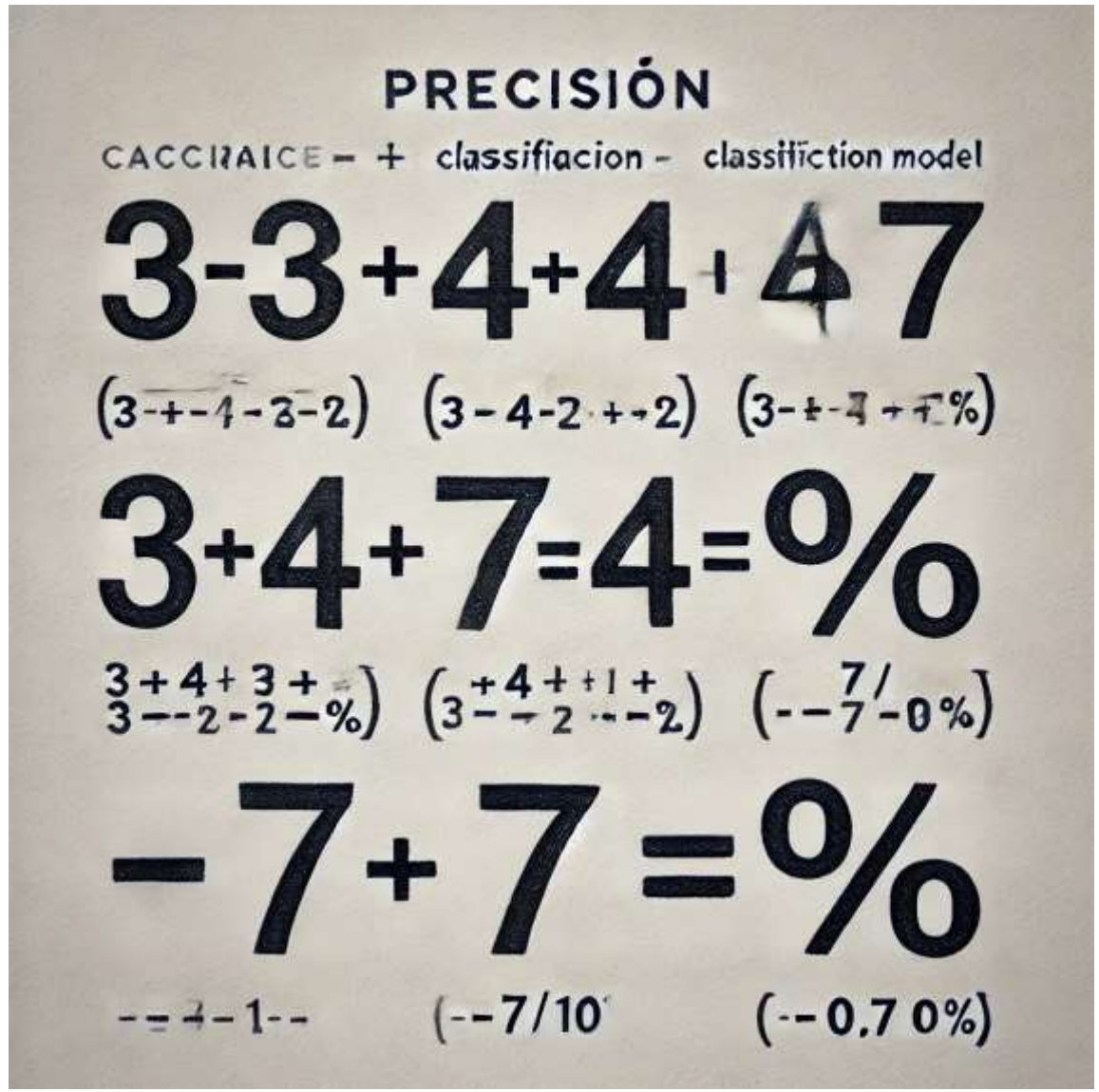
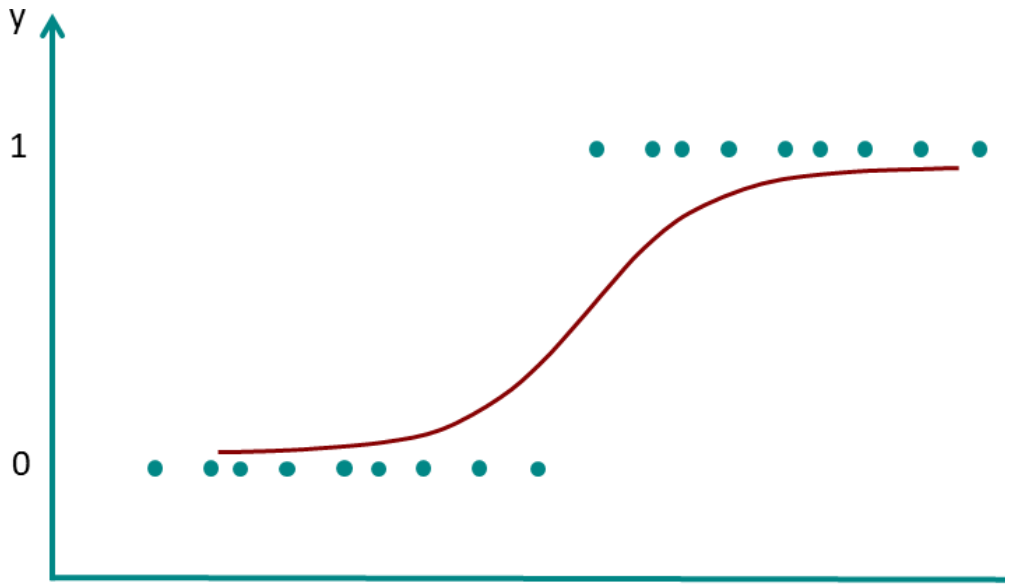
Regresión logística



Regresión logística

La regresión logística es una técnica estadística de machine learning para clasificar los registros de un conjunto de datos, basandose en los valores de los campos de entrada**. En regresión logística, usaremos una o mas variables independientes para predecir un resultado, al cual llamaremos variable dependiente**. La regresión logística es análoga a la regresión lineal pero intenta predecir un campo objetivo categórico o discreto en lugar de uno numérico.



- Predecir la probabilidad de que una persona tenga un ataque al corazón en un periodo especificado de tiempo, basado en nuestro conocimiento de la edad de la persona, sexo, e indice de masa corporal.
- Predecir la probabilidad de mortalidad en un paciente herido, o predecir si un paciente tiene una enfermedad, como la diabetes, basado en las características observadas de ese paciente, como el peso, la altura, la presión sanguínea, y el resultado de varios test de sangre, etc.
- En un contexto de marketing, podemos usarlo para predecir la probabilidad de un cliente de estar pagando o cancelando una suscripción.
- También podemos usar regresión logística para predecir la probabilidad de fallo de un proceso, sistema o producto.
- Podemos precedir la probabilidad del propietario de dejar de pagar la hipoteca.

¿Cuáles son las aplicaciones de la regresión logística?

La regresión logística tiene varias aplicaciones del mundo real en muchos sectores diferentes.

Fabricación

Las empresas de fabricación utilizan el análisis de regresión logística para estimar la probabilidad de fallo de las piezas en la maquinaria. Luego, planifican los programas de mantenimiento en función de esta estimación para minimizar los fallos futuros.

Sanidad

Los investigadores médicos planifican la atención y el tratamiento preventivos mediante la predicción de la probabilidad de enfermedad en los pacientes. Utilizan modelos de regresión logística para comparar el impacto de los antecedentes familiares o los genes en las enfermedades.

Finanzas

Las empresas financieras tienen que analizar las transacciones financieras en busca de fraudes y evaluar las solicitudes de préstamos y seguros en busca de riesgos. Estos problemas son adecuados para un modelo de regresión logística porque tienen resultados discretos, como alto riesgo o bajo riesgo y fraudulento o no fraudulento.

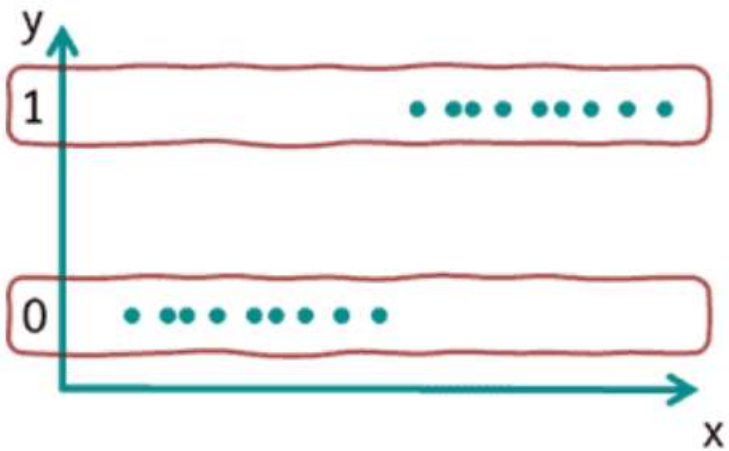
Marketing

Las herramientas de publicidad en línea utilizan el modelo de regresión logística para predecir si los usuarios harán clic en un anuncio. Como resultado, los especialistas en marketing pueden analizar las respuestas de los usuarios a diferentes palabras e imágenes y crear anuncios de alto rendimiento con los que los clientes interactuarán.

Regresión logística

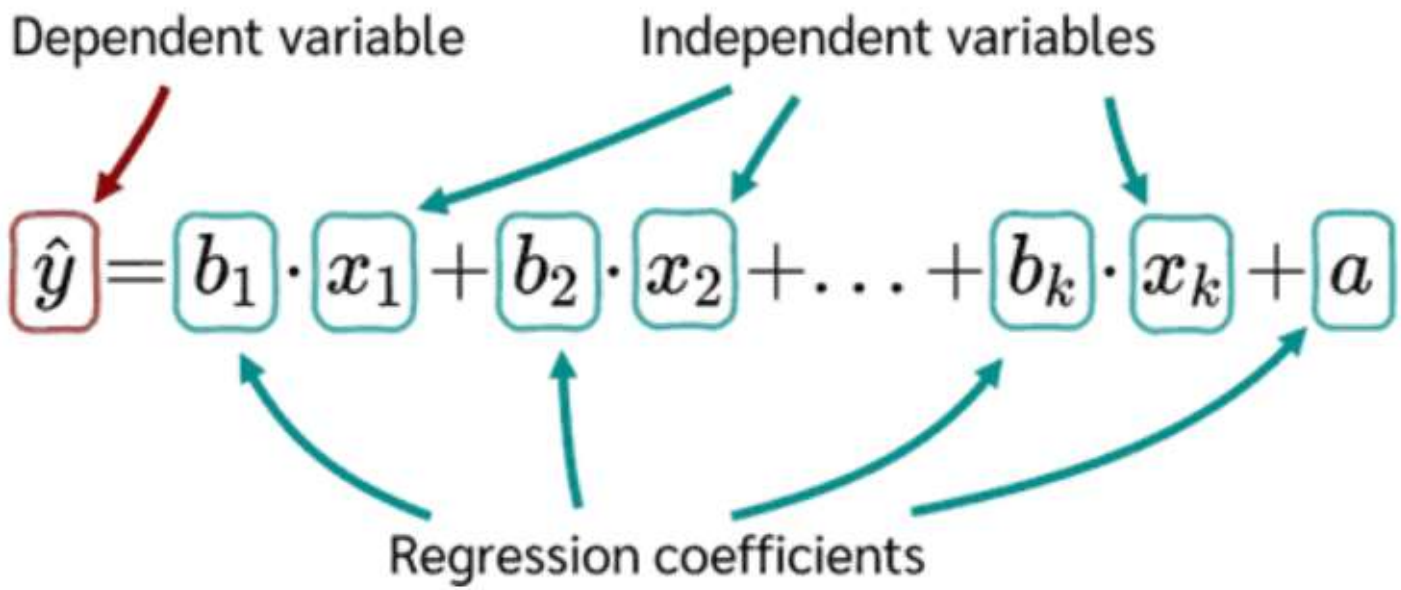
La regresión logística es un **caso especial del análisis de regresión** y se utiliza cuando la variable **dependiente tiene una escala nominal**. Es el caso, por ejemplo, de la variable decisión de compra con los dos valores *compra un producto* y *no compra un producto*.

Con la regresión logística, ahora es posible explicar la variable dependiente o estimar la probabilidad de ocurrencia de las categorías de la variable.



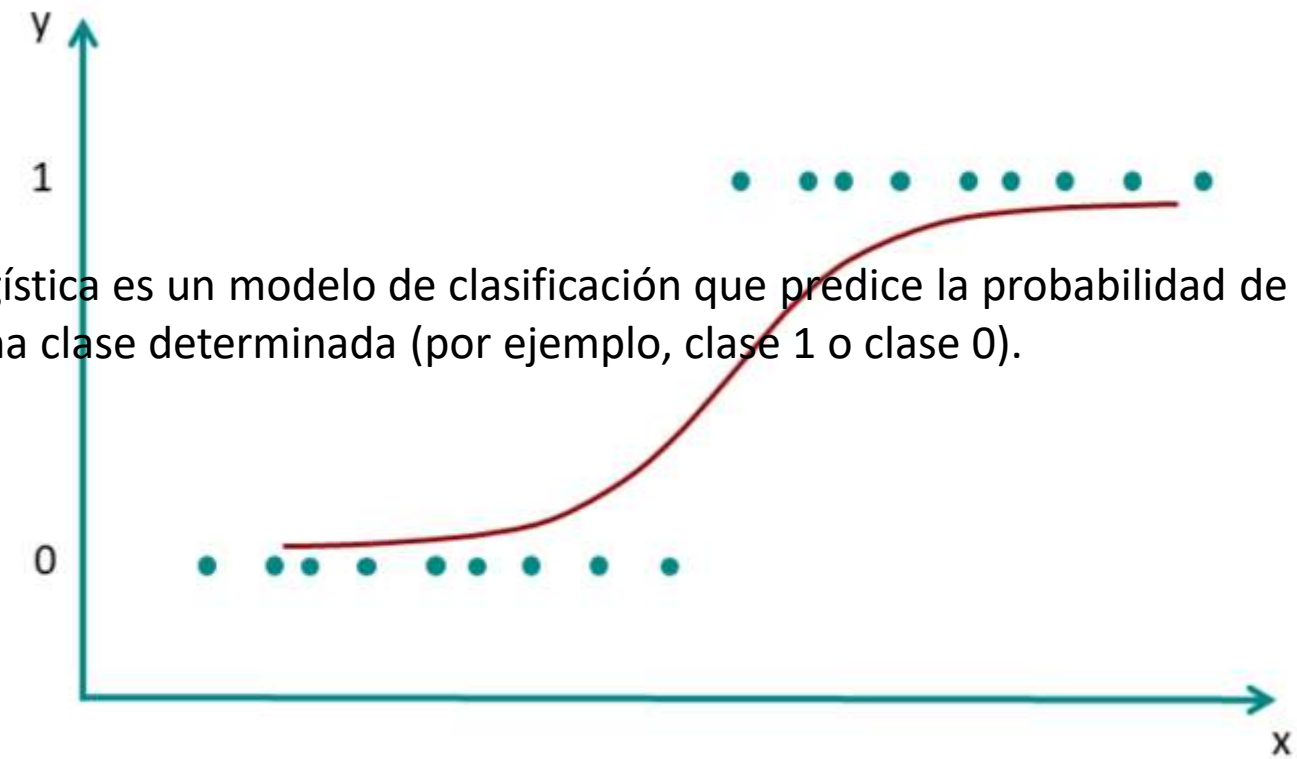
Calcular la regresión logística

Para construir un modelo de regresión logística, se parte de la ecuación de regresión lineal.



$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

$$f(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}$$



La regresión logística es un modelo de clasificación que predice la probabilidad de que una observación pertenezca a una clase determinada (por ejemplo, clase 1 o clase 0).

La **función sigmoidea** es una función matemática utilizada ampliamente en regresión logística, redes neuronales y otros modelos de aprendizaje automático. Es especialmente conocida por su capacidad para transformar cualquier valor de entrada en un valor de salida que se encuentra en el rango de 0 a 1, lo que la hace ideal para modelar probabilidades.

Fórmula de la Función Sigmoidea

La función sigmoidea estándar se define como:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$P(y = 1 \mid x) = \sigma(w \cdot x + b) = \frac{1}{1 + e^{-(w \cdot x + b)}}$$

Regresión logística Simple

Regresión logística múltiple

Donde:

- $w \cdot x$ es el producto punto entre el vector de pesos w y el vector de características x.
- b es el sesgo o intercepto del modelo.

Función Logit:

A veces, la ecuación se expresa en términos de la función logit, que es el logaritmo de las probabilidades (odds):

Probabilidad que sea blanco, bonito

$$\text{logit}(P) = \ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Probabilidad que sea negro,feo

Aquí:

LOGISTICA es por que es una función LOG para volverla una lineal

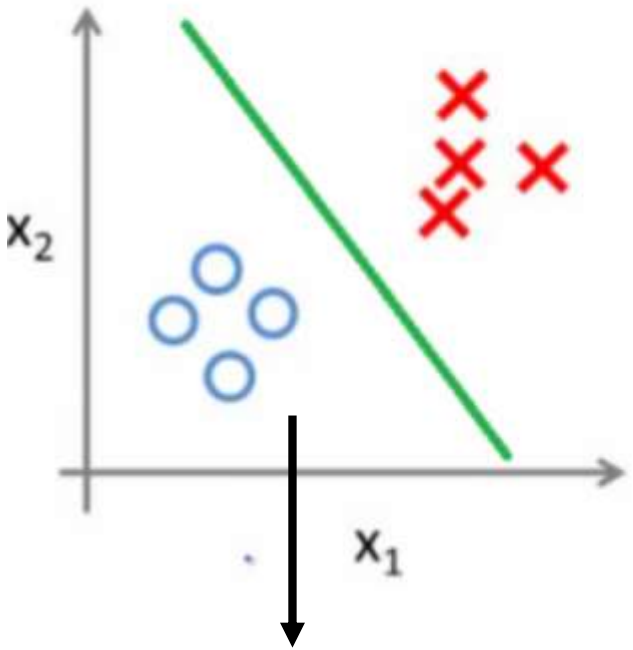
- $\ln$ es el logaritmo natural.
- P es la probabilidad de que Y sea 1.
- $1 - P$ es la probabilidad de que Y sea 0.

El número de ecuaciones que una regresión logística genera , es igual al número de clases que yo quiero predecir.

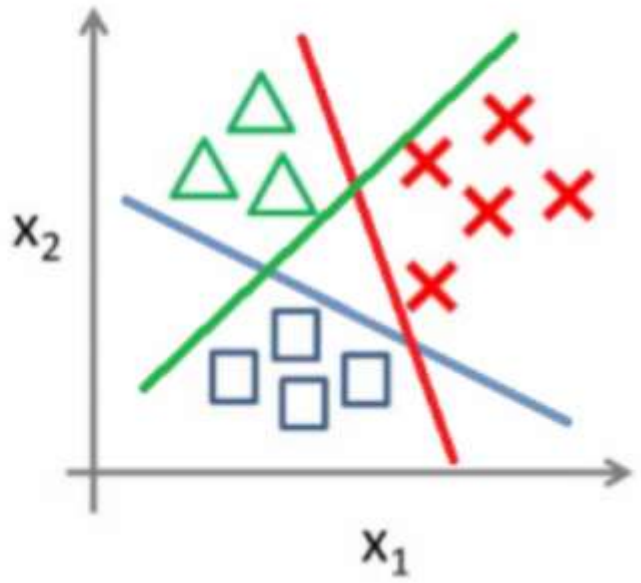
Problema es binario una sola ecuación
Si un cliente es tacaño , exagerado , buena paga , paga para cada cliente voy a tener una ecuación # ecuaciones= clases

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}}$$

Binary classification:



Multi-class classification:



Una clasificación binaria, voy a utilizar una sola regresión logística, una sola ecuación.

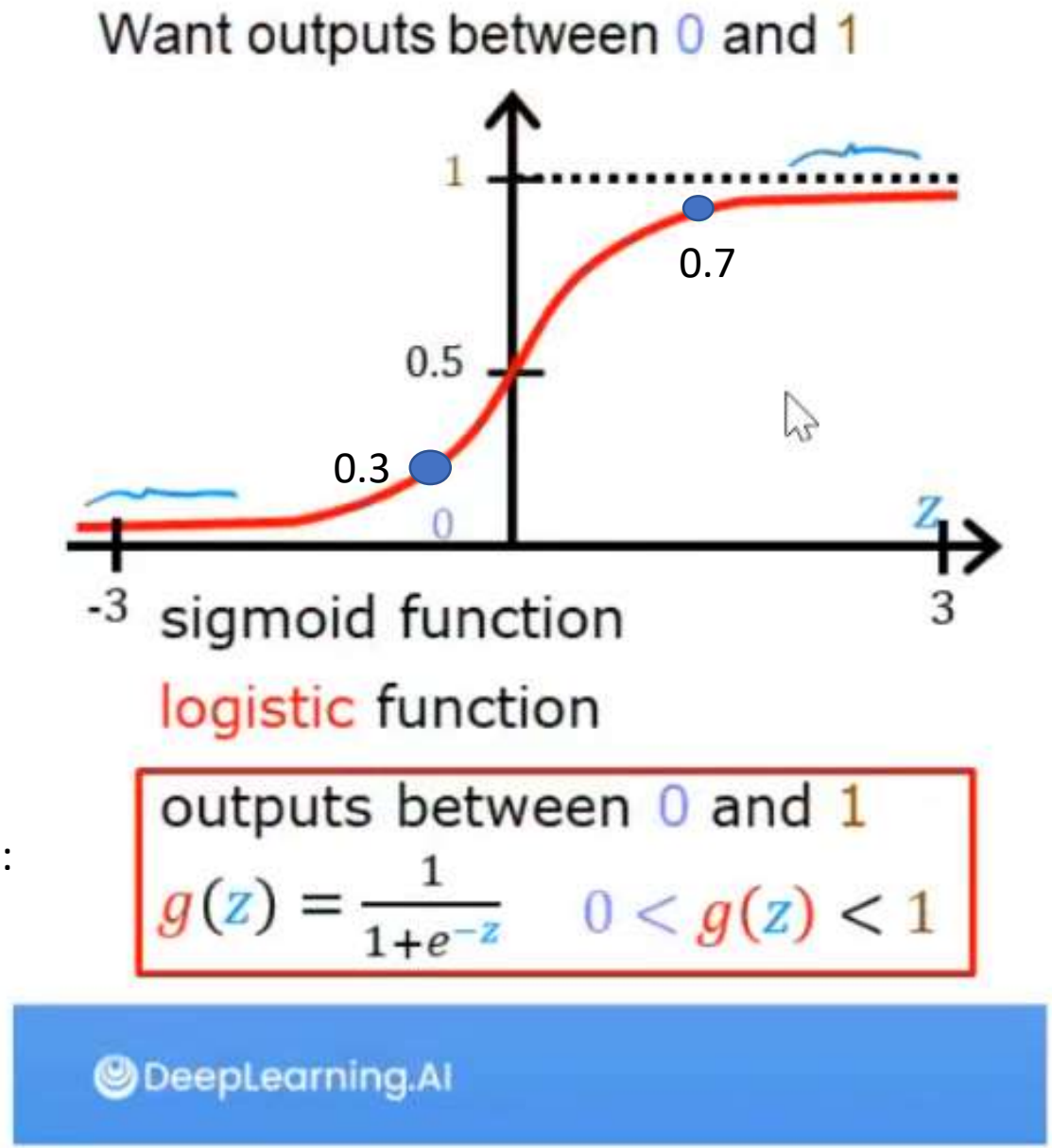
$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

Diferentes coeficientes diferentes pesos

Para generar esta Función:



Me garantiza que mi modelo va a generar entre 0 y 1

```
def sigmoid(z):
    """
    Compute the sigmoid of z

    Parameters
    ----------
    z : array_like
        A scalar or numpy array of any size.

    Returns
    -------
     g : array_like
         sigmoid(z)
    """
    z = np.clip( z, -500, 500 )           # protect against overflow
    g = 1.0/(1.0+np.exp(-z))

    return g
```

Z es el parámetro de la función y se espera que sea un array de NumPy o un escalar (un solo número).

La función devuelve g, que es un array del mismo tamaño que z, con los valores transformados por la función sigmoide.

Una distribución de probabilidad, o sea, una estructura tipo vector donde él me va a decir para cada observación. Cuál es la probabilidad entre 0 y 1

1. **Matriz de Confusión**

- **Descripción**: Es una tabla que muestra las predicciones correctas e incorrectas realizadas por un modelo de clasificación, dividiendo los resultados en cuatro categorías:

- **Descripción**: La matriz de confusión proporciona una visión detallada de las predicciones correctas e incorrectas del modelo, desglosadas por clase.
- **Interpretación**: Permite identificar qué tipos de errores está cometiendo el modelo (FP o FN).

Vamos a ilustrar cada una de las métricas con un ejemplo práctico basado en un modelo de clasificación binaria para predecir si un correo electrónico es spam (1) o no spam (0). Supongamos que el modelo ha hecho las siguientes predicciones y tenemos las etiquetas reales:

Suposiciones:

- **Predicciones del modelo**: [1, 0, 1, 1, 0, 0, 1, 0, 1, 0]
- **Etiquetas reales (verdaderas)**: [1, 0, 1, 0, 0, 0, 1, 1, 0, 0]

A partir de estas predicciones, construimos la **matriz de confusión** y calculamos las métricas correspondientes.

1. Matriz de Confusión

Predicciones del modelo: [1, 0, 1, 1, 0, 0, 1, 0, 1, 0]
Etiquetas reales (verdaderas): [1, 0, 1, 0, 0, 0, 1, 1, 0, 0]

La matriz de confusión se vería as

	Predicho No Spam (0)	Predicho Spam (1)
Real No Spam (0)	TN = 4	FP = 1
Real Spam (1)	FN = 2	TP = 3

- **Descripción**: La matriz de confusión proporciona una visión detallada de las predicciones correctas e incorrectas del modelo, desglosadas por clase.

spam.

- **FP (False Positives)**: 1 correo que no era spam fue incorrectamente predicho como spam.
- **FN (False Negatives)**: 2 correos que eran spam fueron incorrectamente predichos como no spam.

2. **Precisión (Accuracy)**

La precisión es la proporción de predicciones correctas (tanto positivas como negativas) respecto al total de predicciones.

Fórmula:

$$\text{Precisión (Accuracy)} = \frac{TP + TN}{TP + TN + FP + FN}$$

Interpretación:

El 70% de las predicciones realizadas por el modelo fueron correctas.

3. **Precisión (Precision)**

La precisión mide la proporción de verdaderos positivos sobre todas las predicciones positivas.

Fórmula:

$$\text{Precisión (Precision)} = \frac{TP}{TP + FP}$$

Interpretación:

El 75% de los correos que el modelo predijo como spam fueron realmente spam. Esta métrica es importante cuando los falsos positivos son costosos.

4. **Recall (Sensibilidad o Tasa de Verdaderos Positivos)**

El recall mide la proporción de verdaderos positivos identificados correctamente sobre todos los casos que son realmente positivos.

Fórmula:

$$\text{Recall (Sensibilidad)} = \frac{TP}{TP + FN}$$

Interpretación:
El modelo identificó correctamente el 60% de los correos que eran spam. Esta métrica es crucial cuando los falsos negativos son inaceptables.

5. **F1-Score**

El F1-Score es la media armónica de la precisión y el recall. Es útil cuando necesitamos un equilibrio entre ambas métricas.

Fórmula:

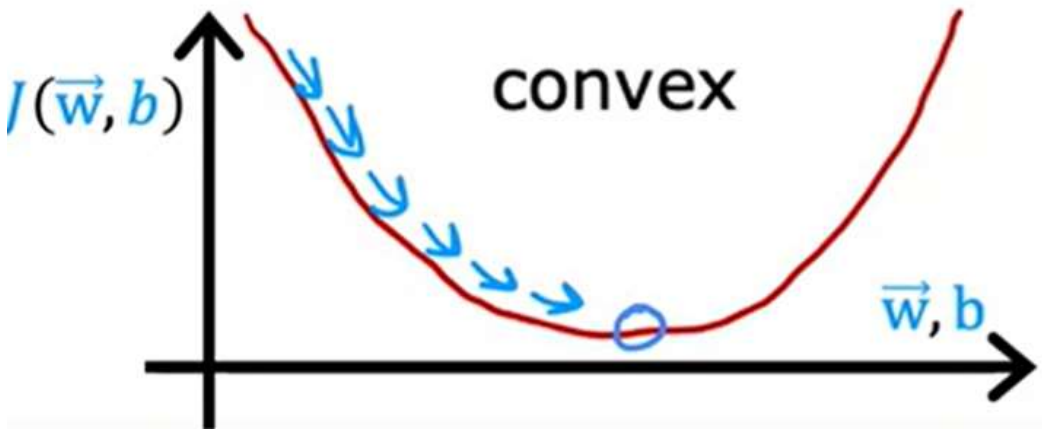
$$\text{F1-Score} = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

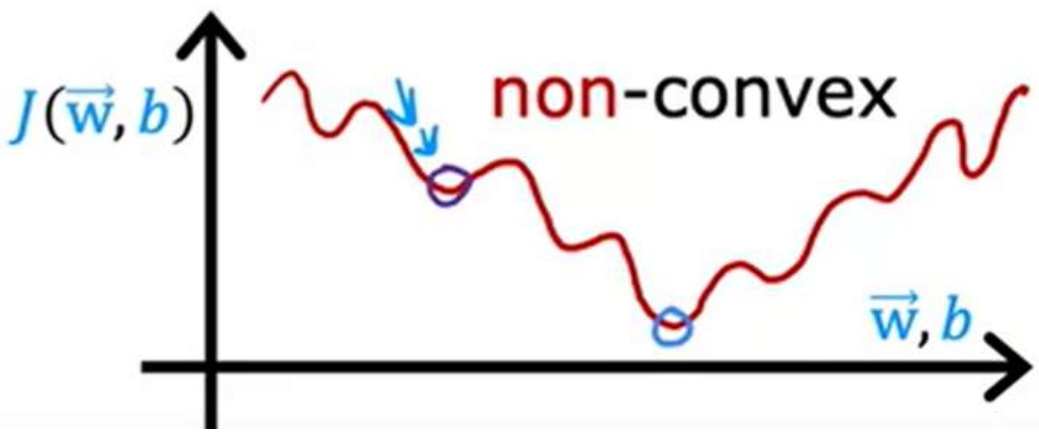
Interpretación:
El F1-Score es del 67%, lo que refleja un equilibrio entre la precisión y el recall. Es útil en escenarios donde hay un desbalance entre las clases y se desea un compromiso entre la capacidad del modelo para predecir correctamente los positivos y evitar falsos positivos.

Squared error cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)^2$$

loss $L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right)$





- La **pérdida cuadrática** (mencionada en tu ejemplo) es más común en **regresión lineal**, donde se mide la diferencia entre el valor real y la predicción para problemas de regresión (predicción de valores continuos).
- En **regresión logística**, usamos la **función de costo logarítmica** porque estamos interesados en clasificar entre dos clases y la función logarítmica maneja mejor las probabilidades que produce la función sigmoide utilizada en este tipo de regresión.

Aprox método Datos reales

$$J(w,b) = \frac{1}{2m} \sum_{i=0}^{m-1} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

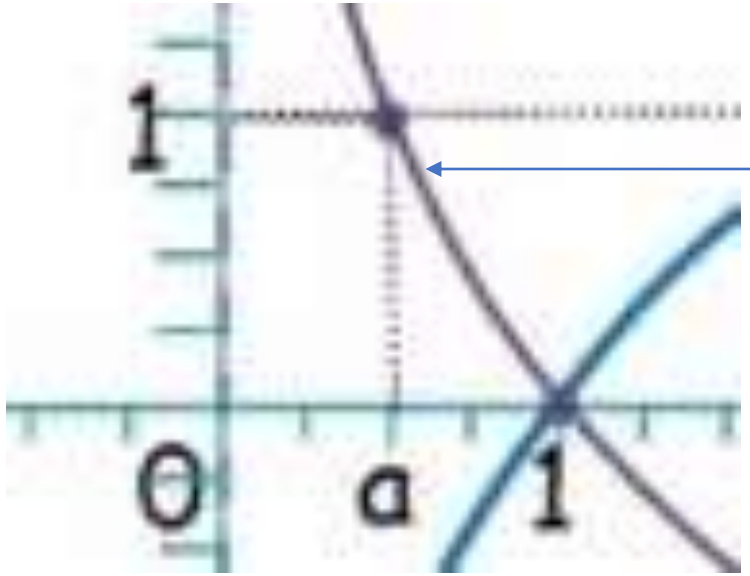
Para cada muestra i, la función de costo mide la discrepancia entre la predicción $f_{w,b}(\vec{x}^{(i)})$ y la etiqueta verdadera $y^{(i)}$:

$$L(f_{w,b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{w,b}(\vec{x}^{(i)})) & \text{si } y^{(i)} = 1 \\ -\log(1 - f_{w,b}(\vec{x}^{(i)})) & \text{si } y^{(i)} = 0 \end{cases}$$

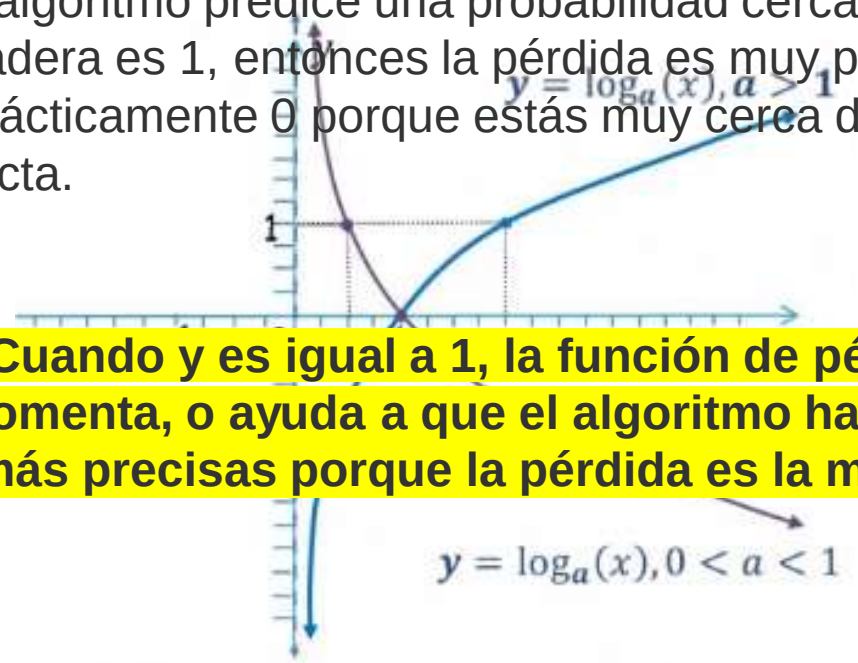
FUNCION DE PERDIDA

Para cada muestra i, la función de costo mide la discrepancia entre la predicción $f_{w,b}(\vec{x}^{(i)})$ y la etiqueta verdadera $y^{(i)}$:

$$L(f_{w,b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{w,b}(\vec{x}^{(i)})) & \text{si } y^{(i)} = 1 \\ -\log(1 - f_{w,b}(\vec{x}^{(i)})) & \text{si } y^{(i)} = 0 \end{cases}$$



Si el algoritmo predice una probabilidad cercana a 1 y la etiqueta verdadera es 1, entonces la pérdida es muy pequeña.
Es prácticamente 0 porque estás muy cerca de la respuesta correcta.



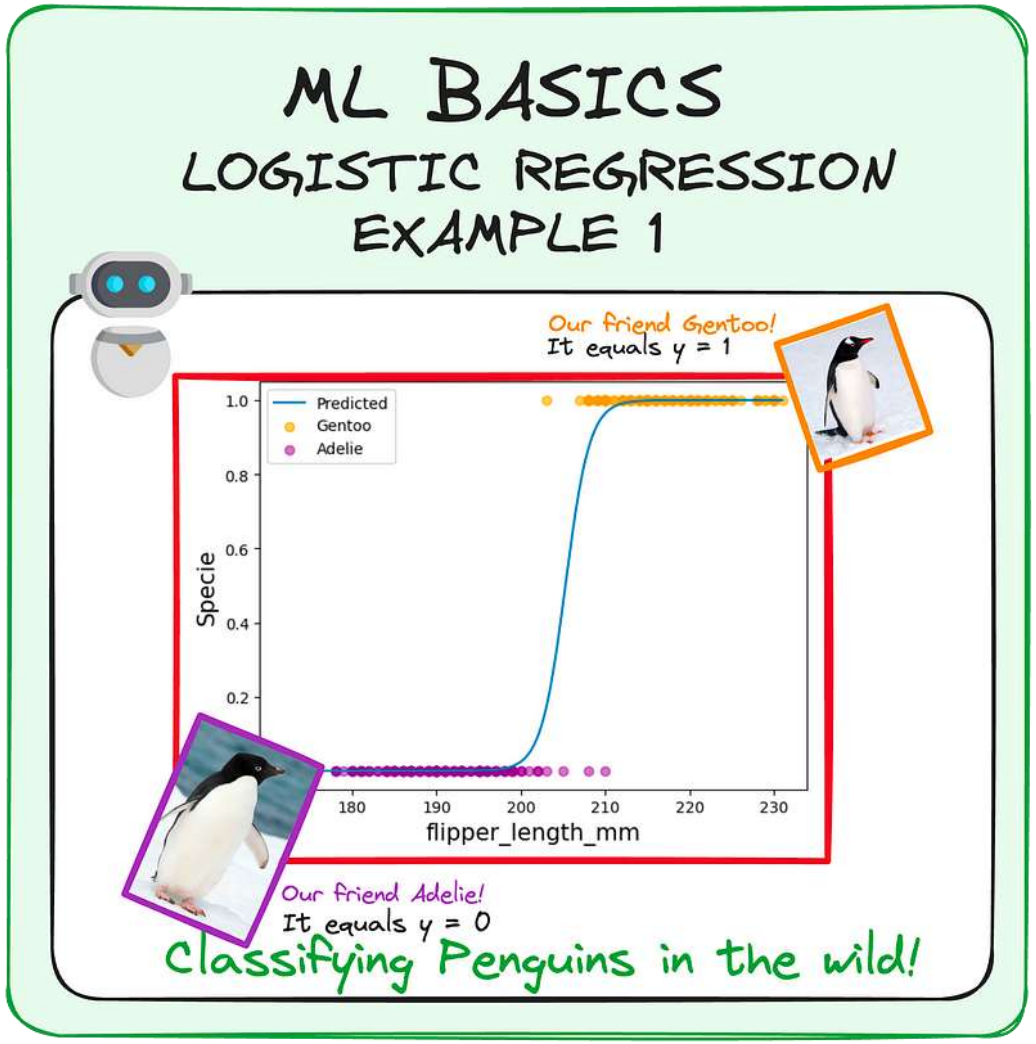
Cuando y es igual a 1, la función de pérdida incentiva o fomenta, o ayuda a que el algoritmo haga predicciones más precisas porque la pérdida es la más baja

3. **Combinar los dos casos en una sola expresión**

Para simplificar, combinamos los dos casos anteriores en una sola expresión:

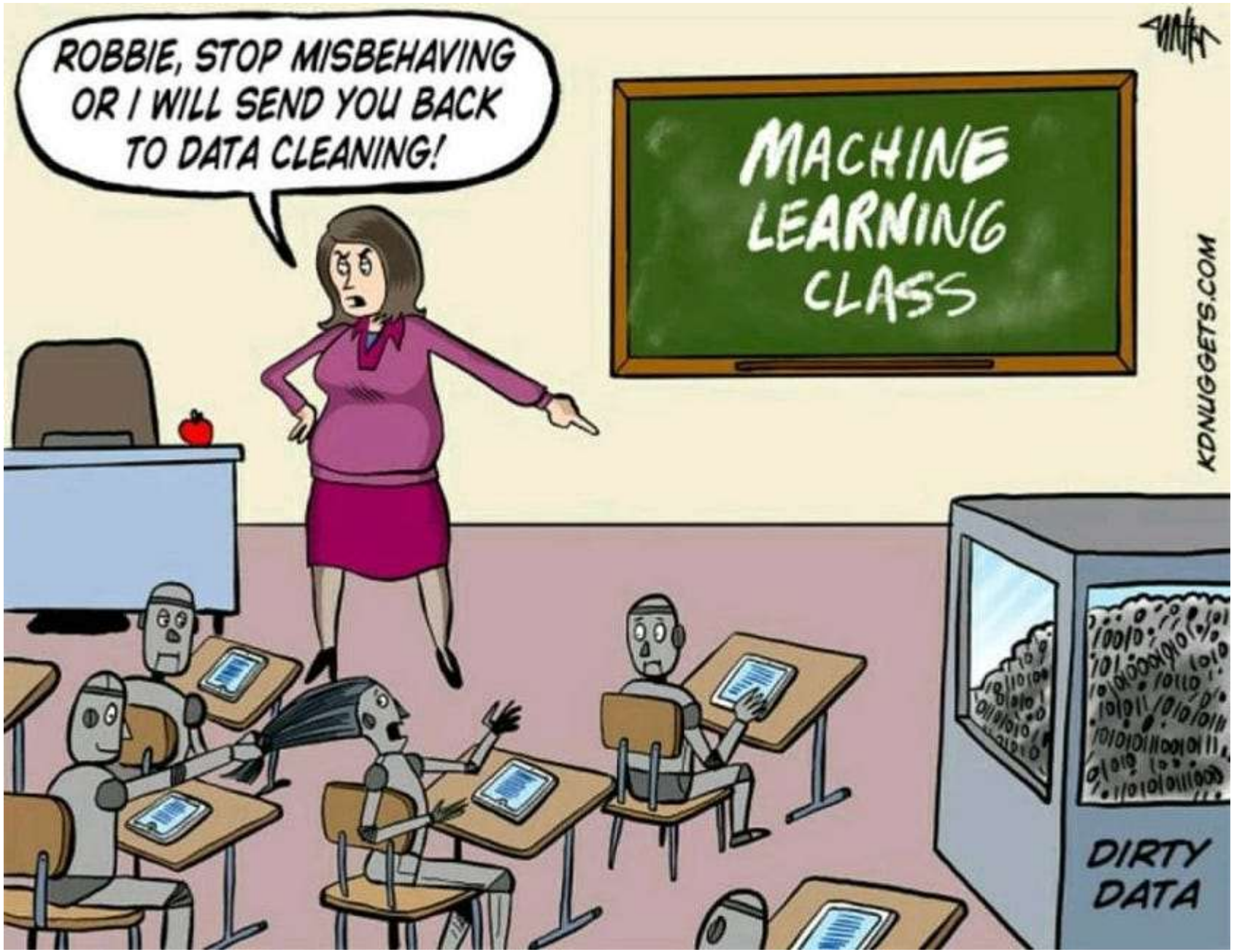
$$L(f_{w,b}(\vec{x}^{(i)}), y^{(i)}) = -\left[y^{(i)} \log(f_{w,b}(\vec{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f_{w,b}(\vec{x}^{(i)}))\right]$$

Esta expresión combina tanto el caso en que $y^{(i)} = 1$ como el caso en que $y^{(i)} = 0$.



Ejemplo en clase medico

Ejemplo de **regresión** logística


ROBBIE, STOP MISBEHAVING OR I WILL SEND YOU BACK TO DATA CLEANING!
MACHINE LEARNING CLASS
KDNUGGETS.COM
DIRTY DATA