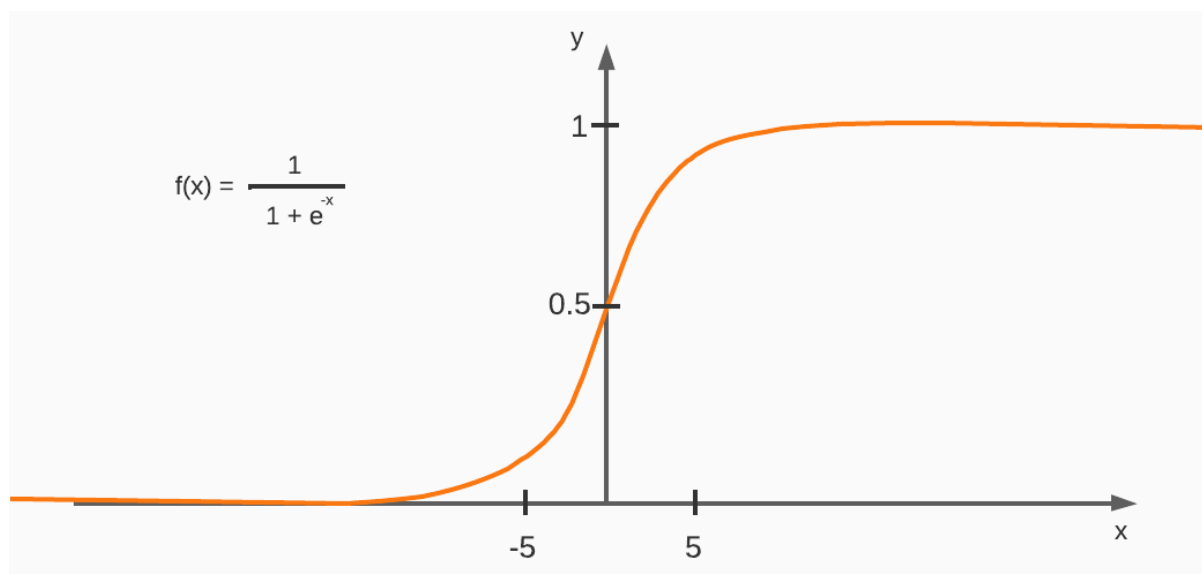


# Regressão Logística

## Função sigmóide

A função sigmóide é uma função matemática utilizada em algoritmos de aprendizado de máquina, especificamente em modelos de classificação binária. Ela transforma valores de entrada em um intervalo entre 0 e 1, que é interpretado como a probabilidade de uma classe.



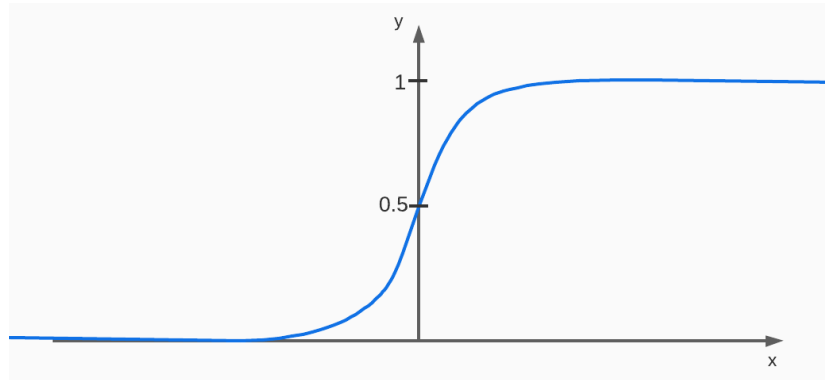
Graficamente, a função sigmóide é representada por uma curva "S", que tem seu ponto máximo no eixo y em 1 e seu ponto mínimo em 0. A função é simétrica em relação ao eixo  $y = 0.5$ .

É possível observar que a função sigmóide tem dois limites: quando  $x$  é muito grande positivamente, a saída se aproxima de 1 e quando  $x$  é muito grande negativamente, a saída se aproxima de 0. Isso significa que a função sigmóide pode ser usada para modelar situações onde existe uma clara distinção entre eventos, como por exemplo prever se um cliente é devedor ou não devedor.

Dado um conjunto de dados de clientes de uma instituição financeira, cada entrada tem informações como renda, idade, etc. e uma saída que indica se o cliente é devedor (1) ou não devedor (0). Um modelo de classificação binária usando a função sigmóide pode ser treinado com esses dados para prever se um novo cliente será devedor ou não devedor, a partir de suas informações. A saída da função sigmóide seria a probabilidade de o cliente ser devedor, ou seja, um valor entre 0 e 1.

# Regressão Logística

A regressão logística é um tipo de algoritmo de aprendizado supervisionado utilizado para resolver problemas de classificação binária. Ele tenta prever a probabilidade de ocorrência de uma determinada classe (geralmente representada como 1 ou 0) dado um conjunto de recursos de entrada. O resultado final é mapeado para uma probabilidade através da função logística, e então é usado para prever a classe.



Na prática, a regressão logística é implementada usando a equação logística, que é dada por:

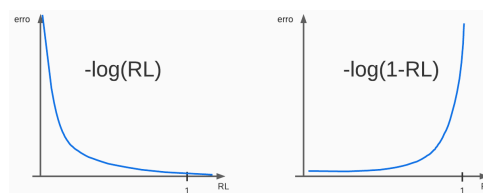
$$RL = \frac{1}{1 + e^{-(\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \dots + \alpha_n x_n + \beta_0)}}, \text{ onde } x_1, x_2, x_3, \dots, x_n \text{ são as variáveis do modelo e}$$

as constantes  $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$  são seus respectivos pesos.

A função de custo utilizada na regressão logística é dada por:

$$custo = - \sum_{i=1}^n ((y_i \times \log(RL_i)) + ((1 - y_i) \times \log(1 - RL_i))) , \text{ onde } n \text{ é a quantidade de}$$

amostras,  $y$  é o dado real (0 ou 1),  $RL_i$  é a probabilidade prevista pela equação logística, e as equações  $-\log(RL_i)$  e  $-\log(1 - RL_i)$  são definidas pelos gráficos:



A ideia por trás desta função de custo é medir o quão longe a probabilidade prevista está da verdadeira probabilidade. Quando a probabilidade prevista é igual à verdadeira probabilidade, a função de custo é igual a zero. Quando a probabilidade prevista é diferente da verdadeira probabilidade, a função de custo é maior do que zero.

Depois de calcular o custo, o próximo passo é atualizar os coeficientes  $\alpha_1, \alpha_2, \dots, \alpha_n$ . Esses coeficientes são atualizados usando um algoritmo de otimização, como o gradiente descendente. O gradiente descendente é um algoritmo de otimização que busca encontrar os valores dos parâmetros que minimizam a função de custo.

A ideia básica do gradiente descendente é calcular o gradiente (ou derivada parcial) da função de custo em relação a cada parâmetro e atualizar esses parâmetros na direção oposta do gradiente, com um passo chamado taxa de aprendizado. Isso é repetido várias vezes até que a função de custo atinja um mínimo global.

A atualização dos parâmetros pode ser feita usando a seguinte equação:

$\alpha_i = \alpha_i + \lambda \times \frac{d \text{ custo}}{d \alpha_i}$ , onde  $\alpha_i$  é o parâmetro atual,  $\lambda$  é a taxa de aprendizado, e  $\frac{d \text{ custo}}{d \alpha_i}$  é o gradiente da função de custo em relação ao parâmetro  $\alpha_i$ .

A taxa de aprendizado é um hiperparâmetro que controla o tamanho do passo a ser dado na direção oposta do gradiente. Se a taxa de aprendizado for muito pequena, o algoritmo poderá precisar de muitas iterações para convergir, mas se for muito grande, pode causar oscilações ou o algoritmo pode não convergir.

É importante notar que o gradiente descendente é um algoritmo de otimização iterativo e que pode precisar de muitas iterações para encontrar os valores ótimos dos parâmetros. Além disso, é importante escolher uma boa função de custo e uma boa taxa de aprendizado para garantir a convergência do algoritmo e a capacidade de generalização do modelo.

### **Mas o que acontece se estivermos analisando um modelo com mais de uma classe?**

Quando um problema de classificação tem mais classes, a regressão logística precisa ser estendida para lidar com múltiplas classes. Isso pode ser feito por meio da forma *One-vs-All* (também conhecido como *One-vs-Rest*). Essa abordagem consiste em treinar um modelo de regressão logística para cada classe, onde cada modelo é treinado para prever se um exemplo pertence à classe em questão ou não. Esses modelos são treinados de forma independente e, em seguida, a classe com a maior probabilidade é selecionada.