HUB IQ

Inteligencia Artificial & Machine Learning

Applicaciones en movilidad

Dr. Iván S. Razo Zapata

Engineering X

Founded by the Royal Academy of Engineering and Lloyd's Register Foundation

Aprendizaje Supervisado

Clasificación

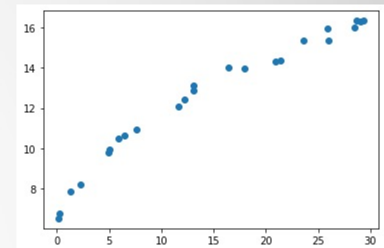
Idea general

- Aprendizaje supervisado
  - Regresión
  - Clasificación

Idea general

Regresión

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \quad Y = \begin{pmatrix} y_1 \\ y_m \end{pmatrix} \qquad Y \in \mathbb{R}$$



$$y = \theta_0 + \theta_1 x_1$$

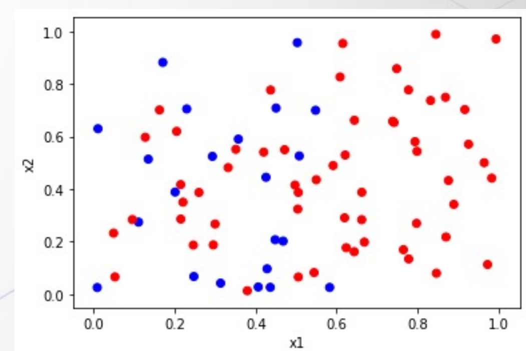
Clasificación

Y es una variable cualitativa

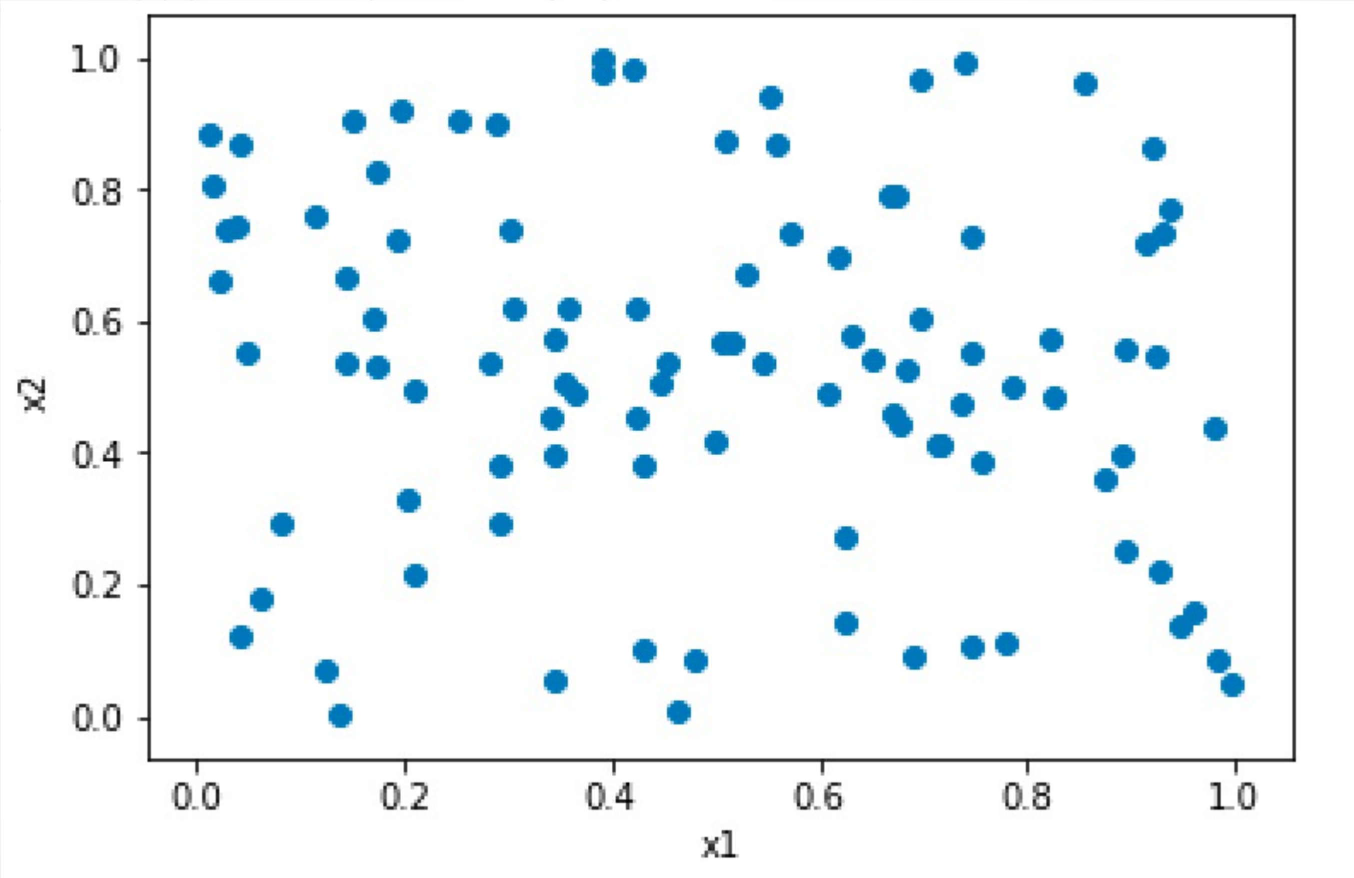
$$Y \in \{Spam, no\ spam\}$$

$$Y \in \{gato, perro, oso\}$$

$$Y \in \{saludable, no\ saludable\}$$

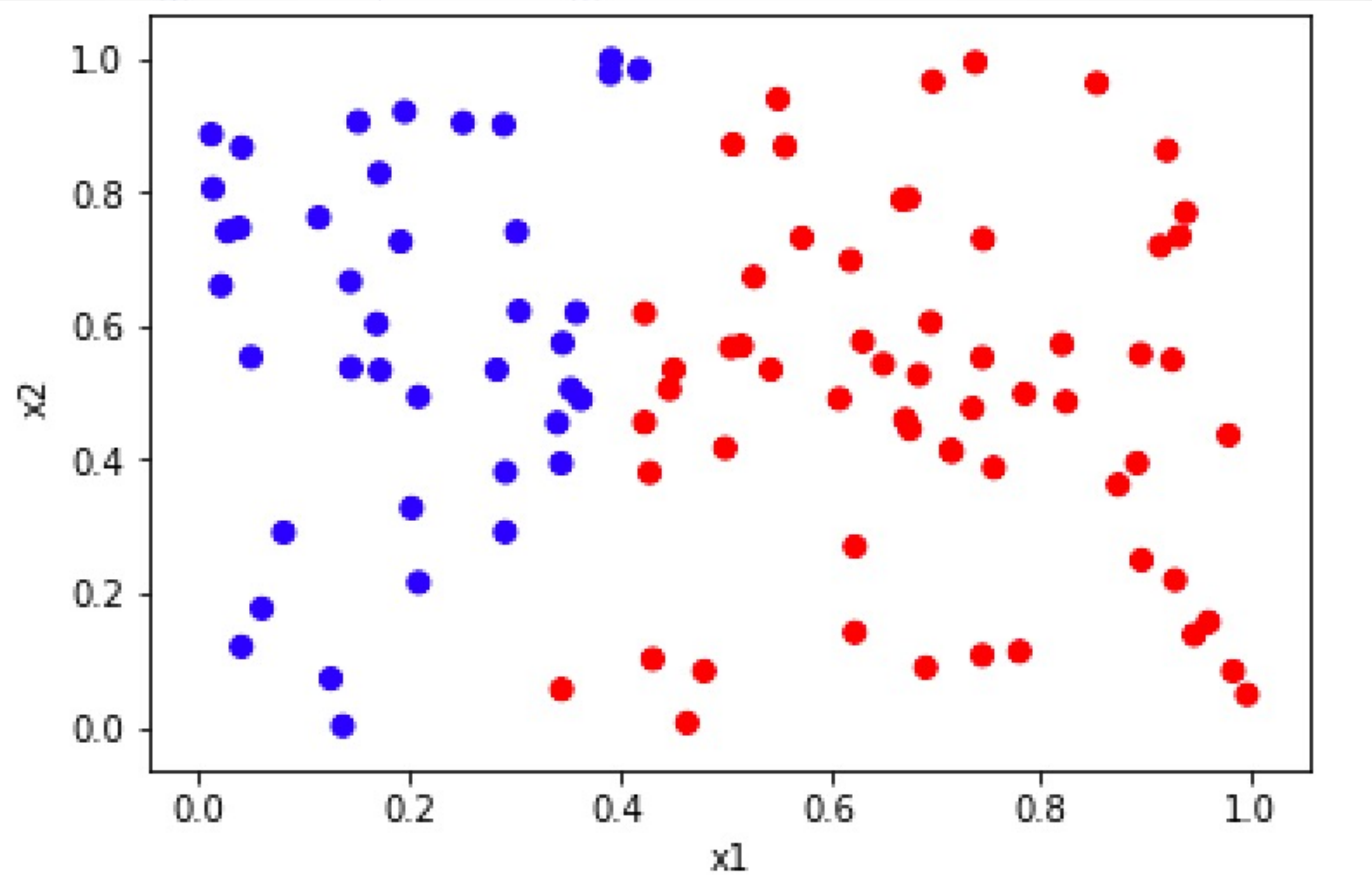


Idea general

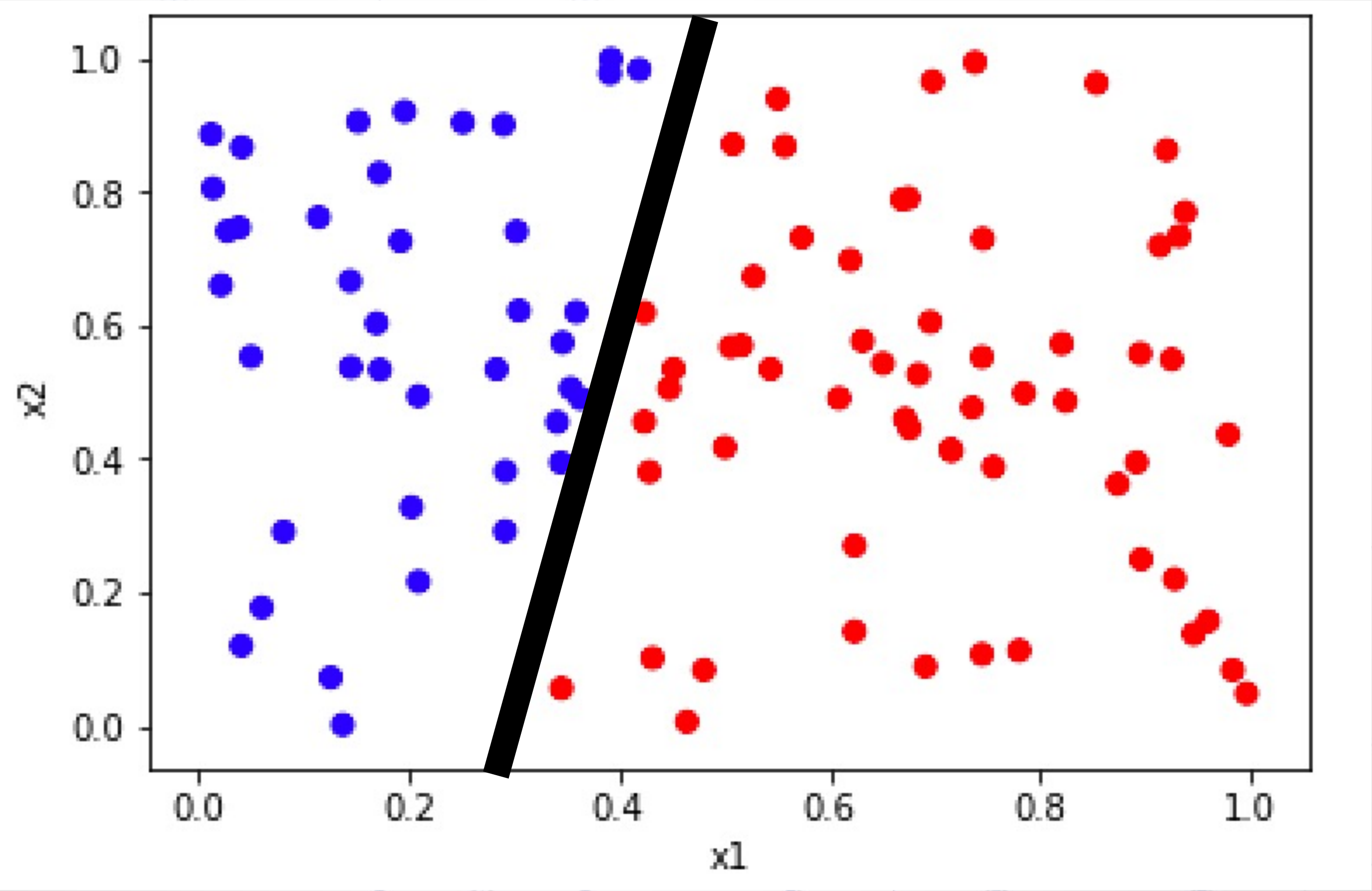


$$O1(x_1, x_2, y_1), O2(x_1, x_2, y_2), \ldots, On(x_1, x_2, y_n)$$

Idea general



Idea general



HUB IQ

EngineeringX
Founded by the Royal Academy of Engineering and Lloyd's Register Foundation

Idea general

	X1	X2	Y (Clase)
Obj1	2	4	1
Obj2	3	2	2
…	5	1	1
Obj_m	6	8	2

Idea general

Regresión

$$\hat{y} = \theta_0 + \theta_1 x_1$$

$$\hat{f}(x_0) = \theta_0 + \theta_1 x_1$$

Clasificación

$$|Y| = K$$

$$\Pr(Y = j \mid X = x_0)$$

$$j \in \{1,2,\ldots,K\}$$

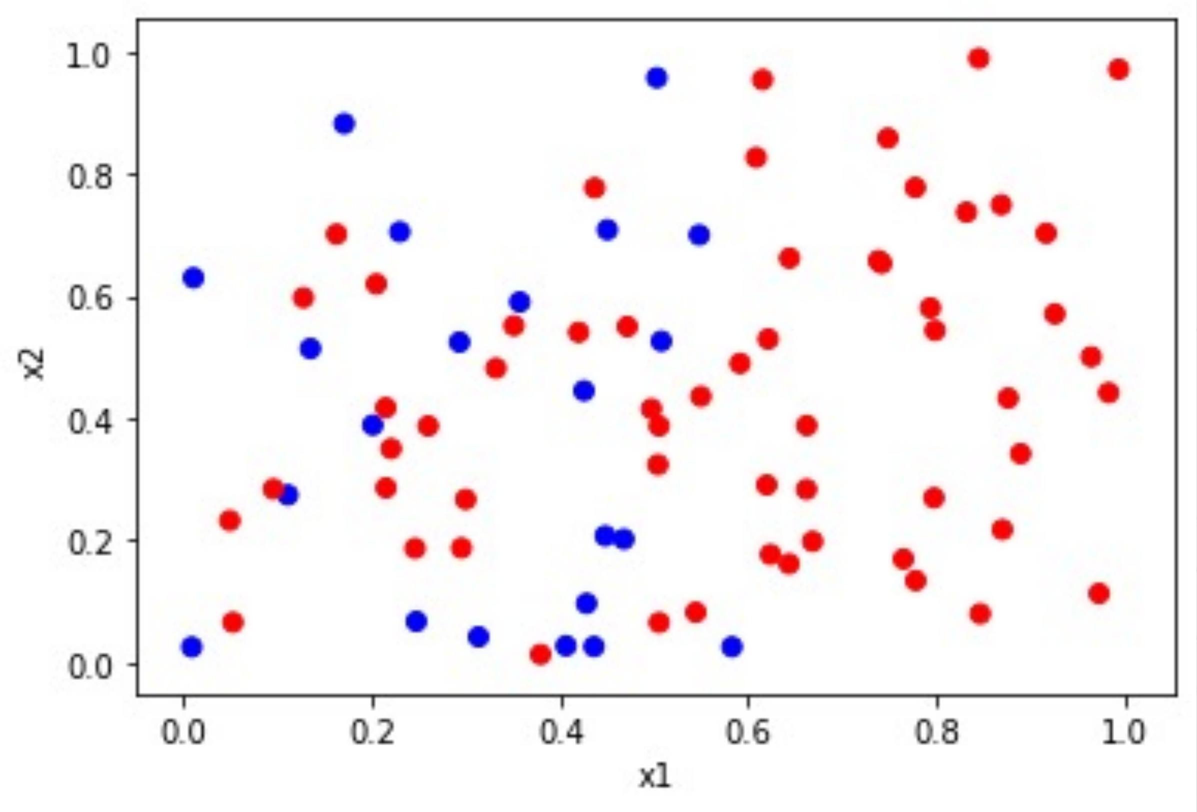
Idea general

Clasificación

$$|Y| = 2$$

$$\Pr(Y = j \mid X = x_0)$$

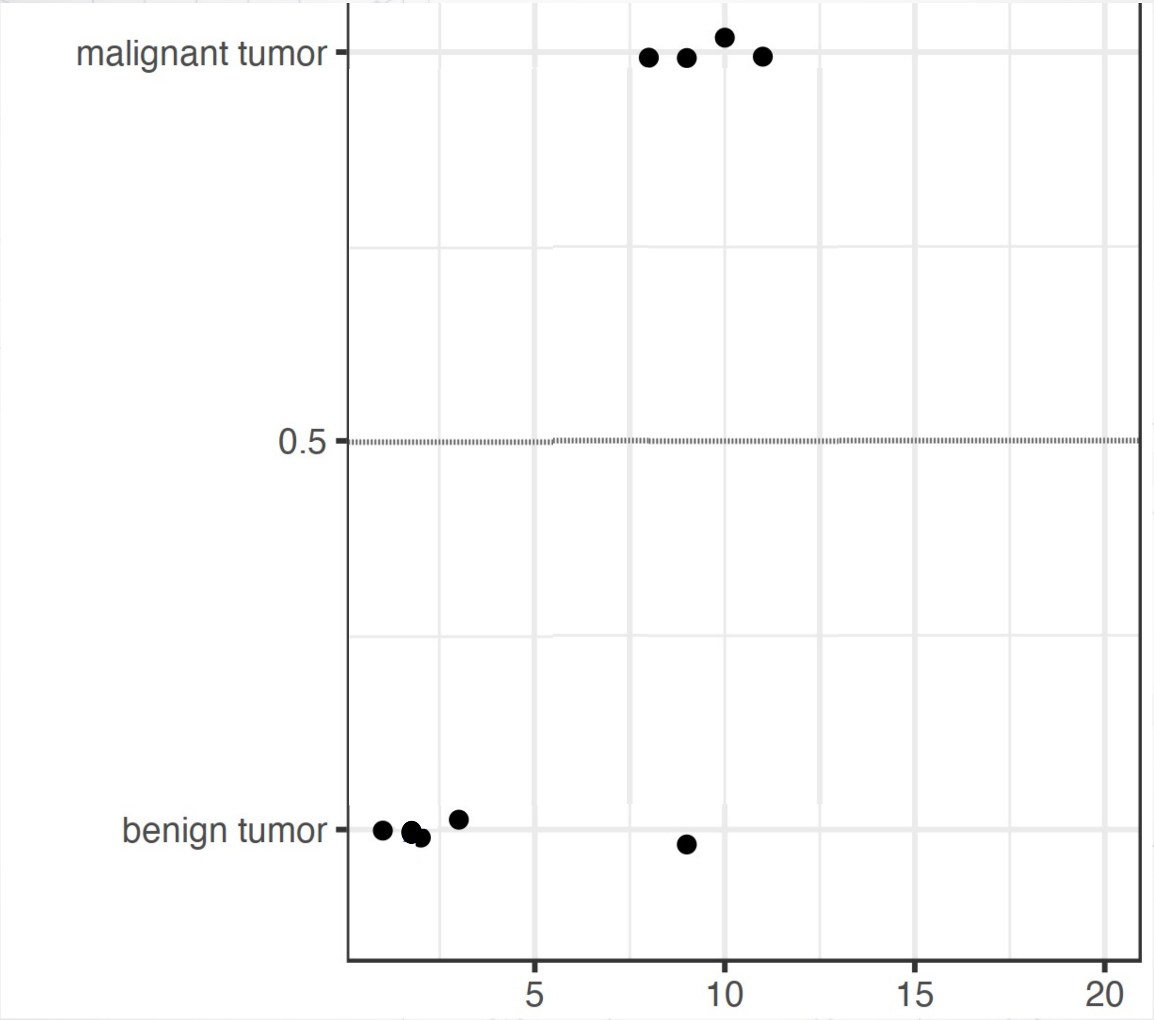
$$j \in \{1,2\}$$



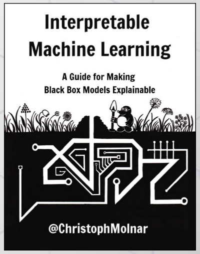
Regresión logística

Idea general

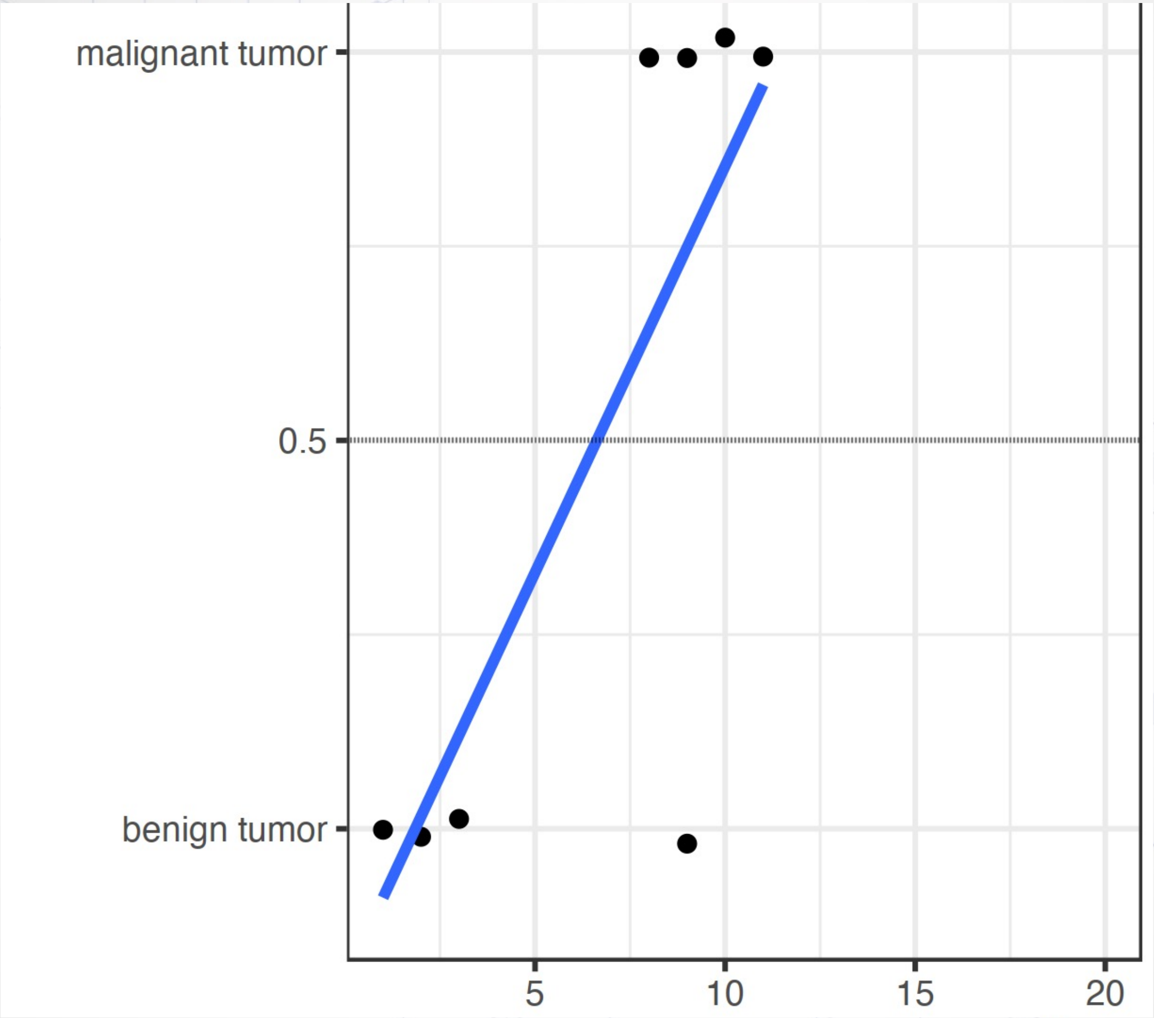
Clasificación binaria

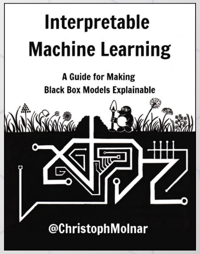


https://christophm.github.io/interpretable-ml-book/index.html



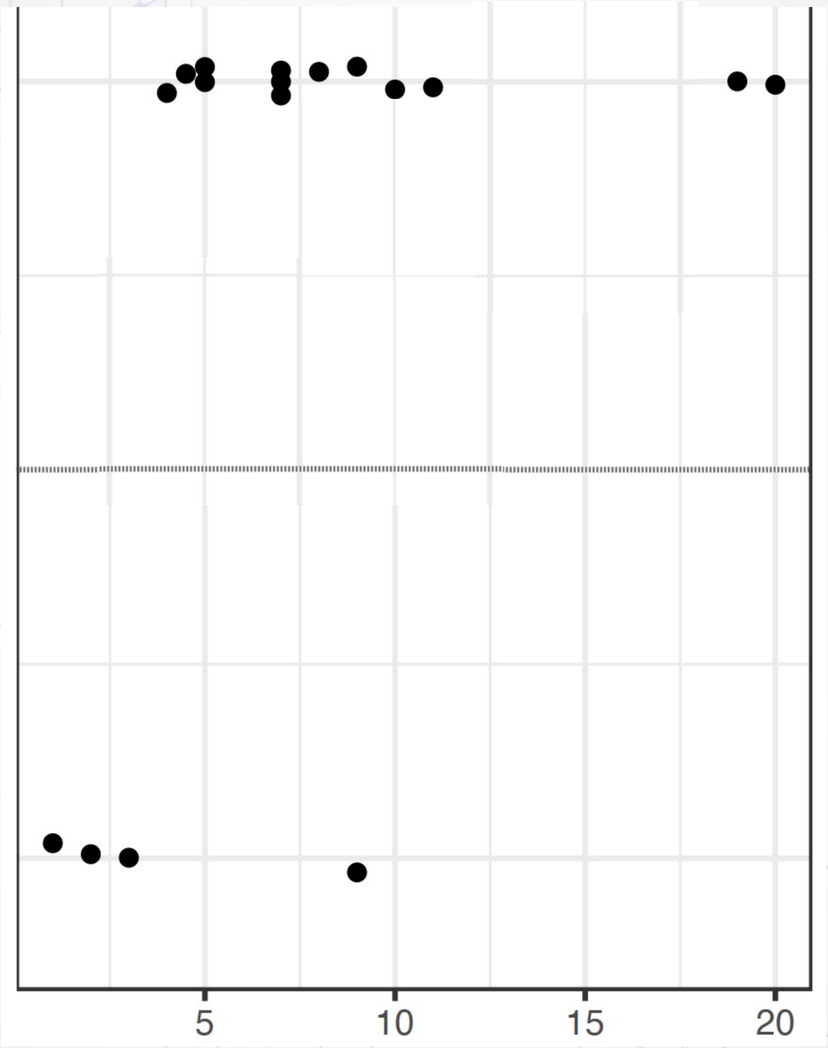
Clasificación binaria

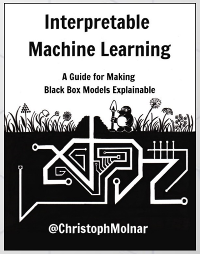




https://christophm.github.io/interpretable-ml-book/index.html

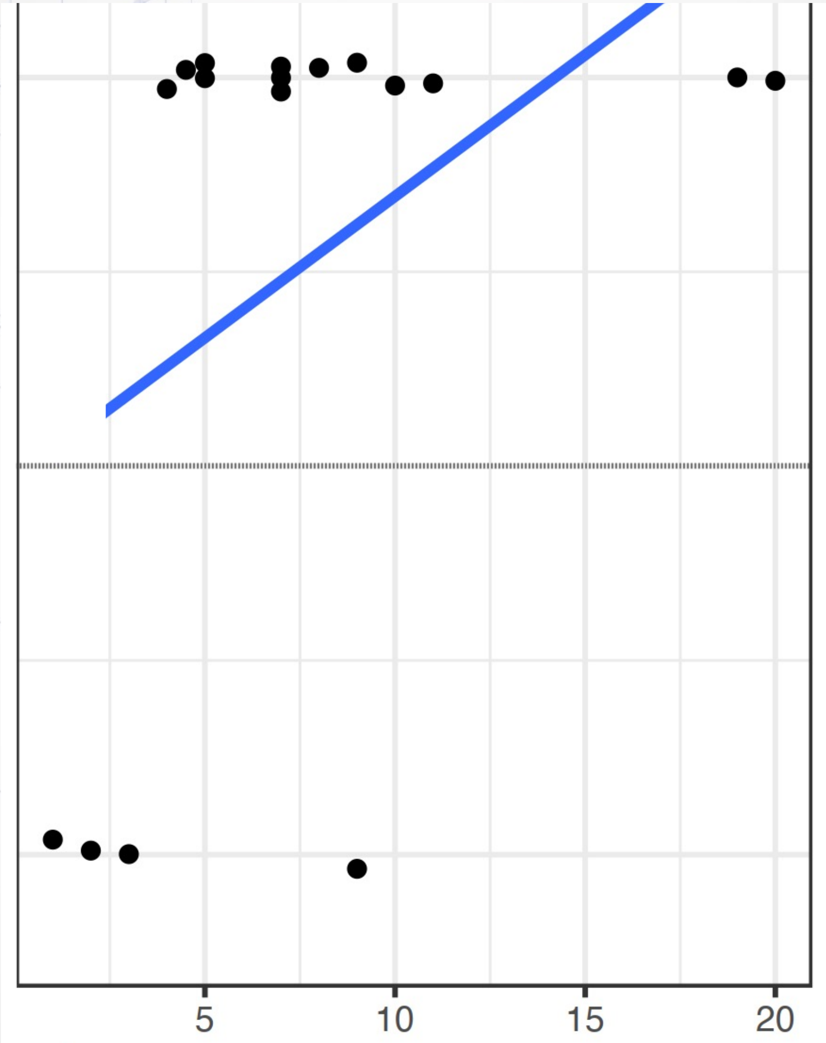
Clasificación binaria



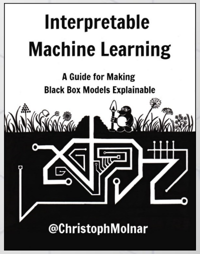


https://christophm.github.io/interpretable-ml-book/index.html

Clasificación binaria



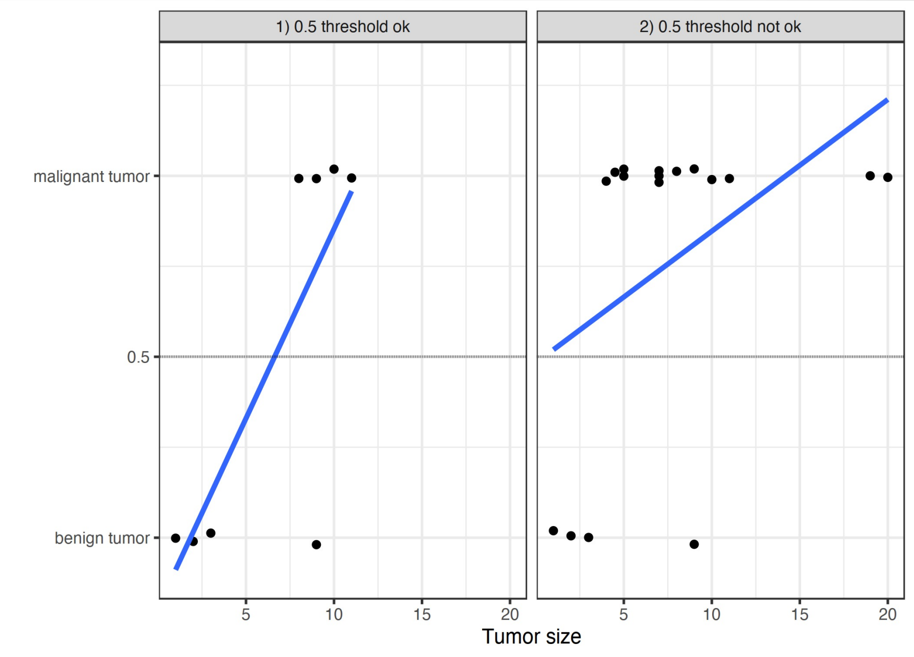
https://christophm.github.io/interpretable-ml-book/index.html



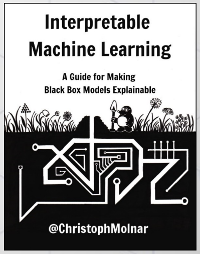
Regresión Lineal – Problema

- La regresión lineal arroja valores por abajo de cero y arriba de uno

- La salida cuantitativa de una regresión lineal no puede ser interpretada como una probabilidad



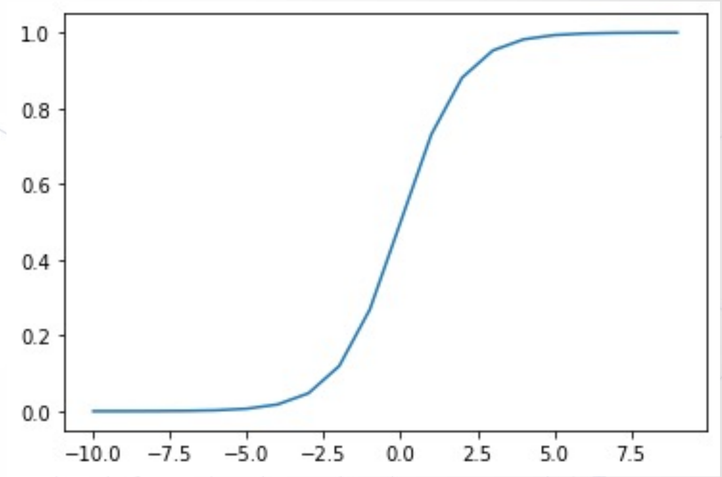
https://christophm.github.io/interpretable-ml-book/index.html



Regresión logística – Solución

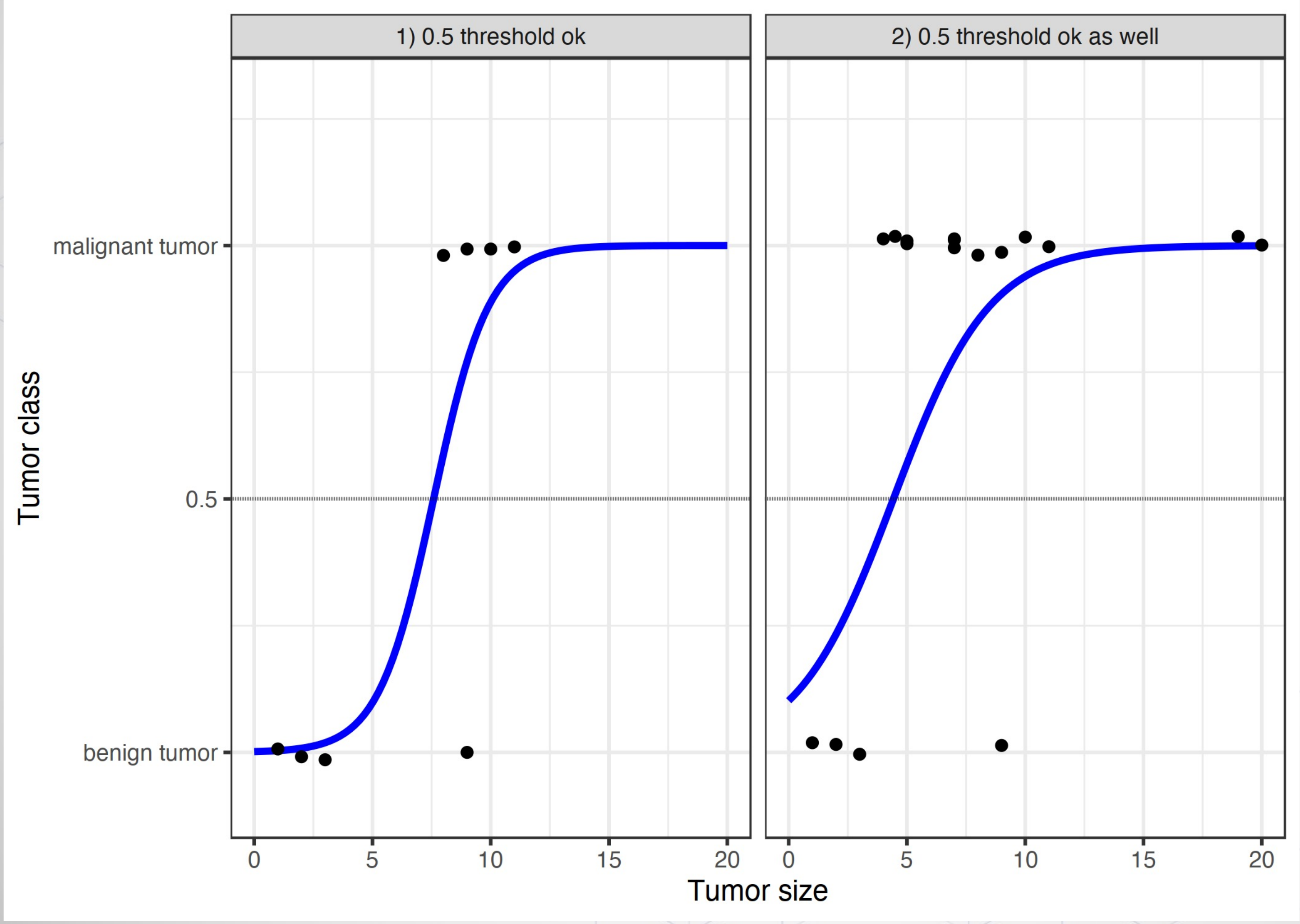
Función logística (sigmoide)

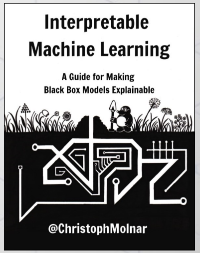
$$logistic(x) = \frac{1}{1 + e^{-x}}$$



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

Regresión logística – Solución (cont)





Regresión logística – Solución (cont)

$$\hat{y}(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1)}}$$

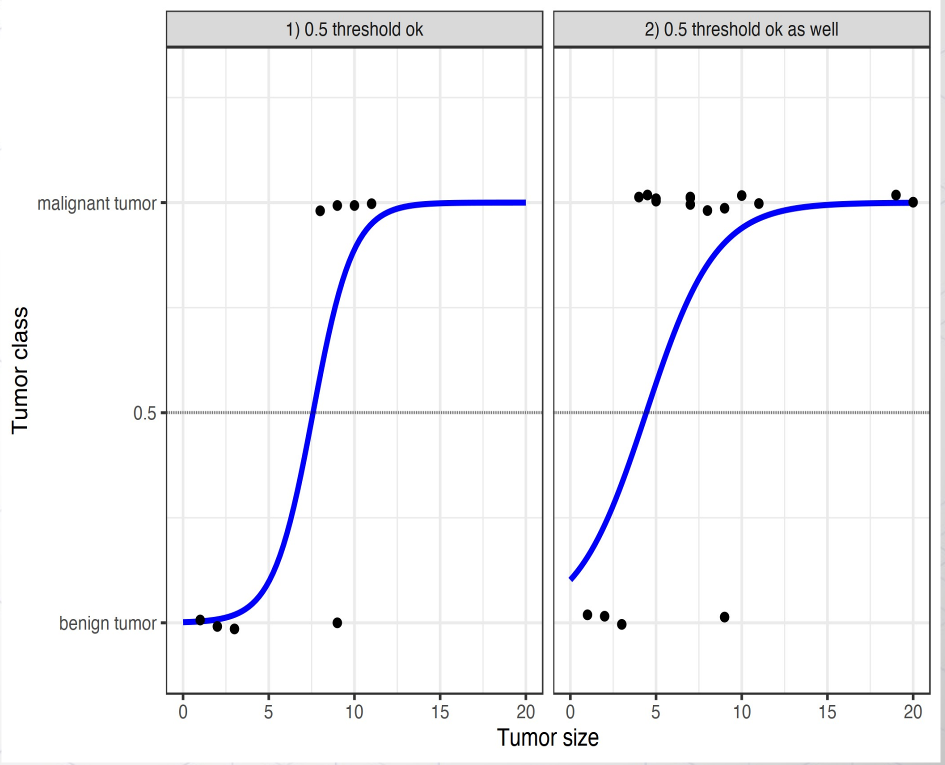
Regresión logística – Solución (cont)

- Probabilidad $$\Pr(Y = j \mid X = x_0)$$

$$\Pr(Y = 1 \mid X = x_0; \hat{f}(x_0) = logistic(\theta_0 + \theta_1 x))$$

$$\hat{f}(x_0) \geq 0.5 \rightarrow \text{Clase } 1$$

$$\hat{f}(x_0) < 0.5 \rightarrow \text{Clase } 0$$



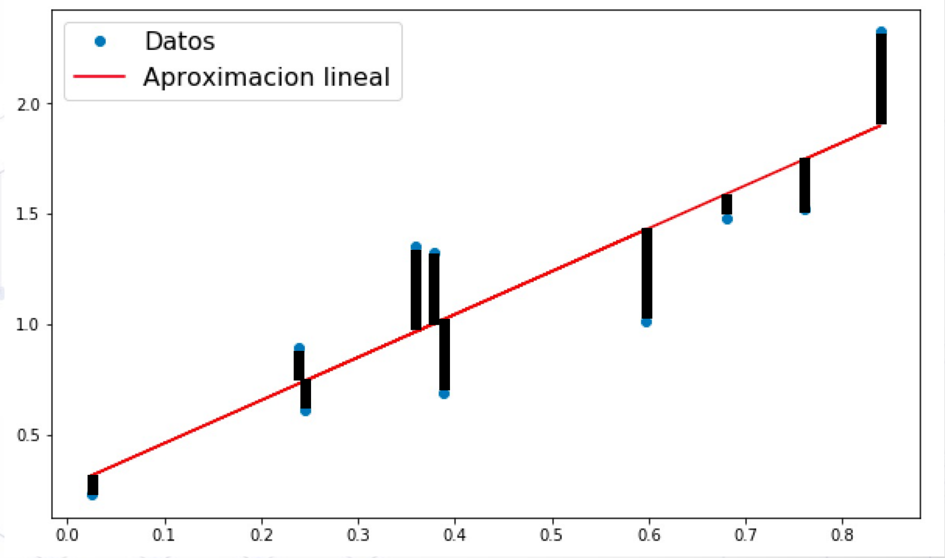
Regresión logística – Ajuste de parámetros

- Descenso de gradiente
- Máxima verosimilitud

Regresión lineal – Función de error

- Regresión

$$E(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_{\theta}(x) - y)^2$$



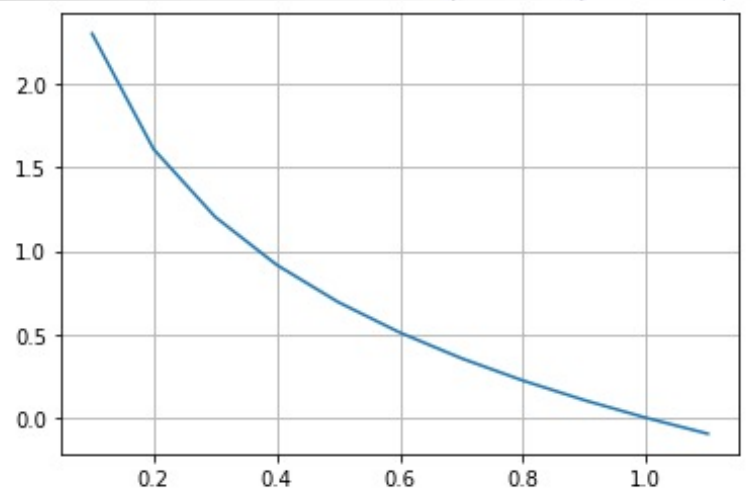
Regresión logística – Función de error

- Caso Y = 1
- Si se predice algo muy cercano a 1.0 de probabilidad, no se debe penalizar tanto
- Si se predice algo cercano a 0.0 de probabilidad, se debe penalizar mucho

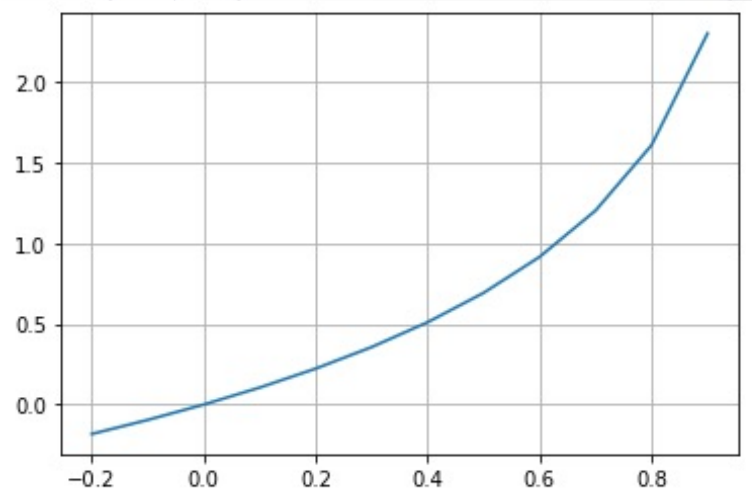
- Caso Y = 0
- Si se predice algo muy cercano a 0.0 de probabilidad no se debe penalizar tanto
- Si se predice algo cercano a 1.0 de probabilidad, se debe penalizar mucho

Regresión logística – Función de error

- Caso Y = 1
- Si se predice algo muy cercano a 1.0 de probabilidad, no se debe penalizar tanto
- Si se predice algo cercano a 0.0 de probabilidad, se debe penalizar mucho

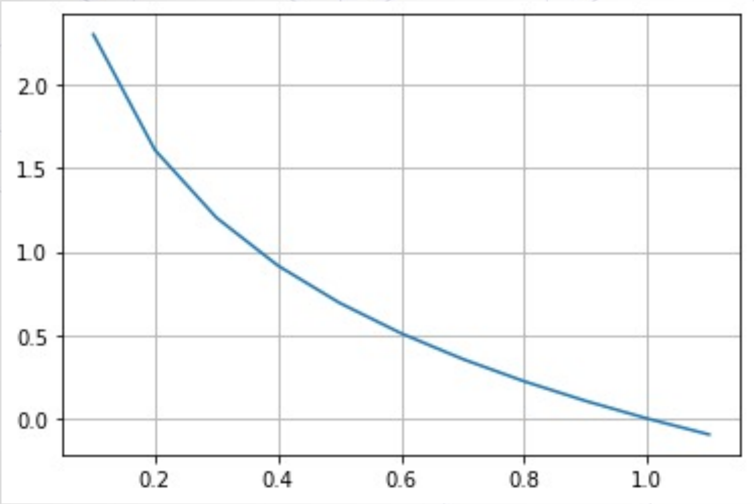


- Caso Y = 0
- Si se predice algo muy cercano a 0.0 de probabilidad, no se debe penalizar tanto
- Si se predice algo cercano a 1.0 de probabilidad, se debe penalizar mucho

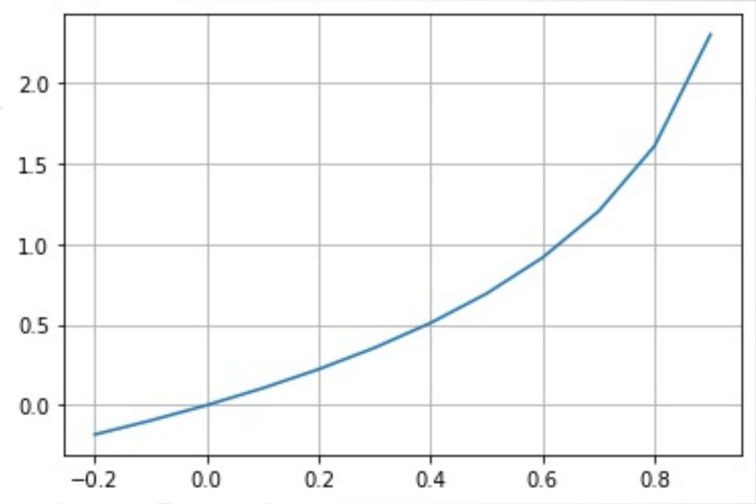


Regresión logística – Función de error

- Caso Y = 1



- Caso Y = 0



$$Costo(\theta) = \begin{cases} -\log\left(\hat{y}_\theta(x)\right)\, si\ y = 1 \\ -\log\left(1 - \hat{y}_\theta(x)\right)\, si\ y = 0 \end{cases}$$

Regresión logística – Función de error

- Versión combinada

$$Costo(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{y}_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log(1 - \hat{y}_\theta\left(x^{(i)}\right))\right]$$

$$Costo(\theta) = \begin{cases} -\log\left(\hat{y}_\theta(x)\right)\ si\ y = 1 \\ -\log\left(1 - \hat{y}_\theta(x)\right)\ si\ y = 0 \end{cases}$$

Regresión logística – Modificando parámetros

$$\theta\nabla$$

$$\frac{\partial Costo(\theta)}{\partial\theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{y}_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log(1 - \hat{y}_\theta\left(x^{(i)}\right))\right]$$

Regresión logística – Modificando parámetros

$\theta \nabla$

$$\frac{dy}{du}\log(u) = \frac{u'}{u}$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{y}_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)})\log(1 - \hat{y}_\theta\left(x^{(i)}\right))\right]$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{1}{\hat{y}_\theta(x^{(i)})}\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j} + \left(1 - y^{(i)}\right)\frac{-1}{1 - \hat{y}_\theta(x^{(i)})}\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j}\right]$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{1}{\hat{y}_\theta(x^{(i)})} + \left(1 - y^{(i)}\right)\frac{-1}{1 - \hat{y}_\theta(x^{(i)})}\right]\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j}$$

HUB IQ

EngineeringX
Founded by the Royal Academy of Engineering and Lloyd's Register Foundation

Regresión logística – Modificando parámetros (cont.)

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{1}{\hat{y}_\theta(x^{(i)})} + \left(1-y^{(i)}\right)\frac{-1}{1-\hat{y}_\theta(x^{(i)})}\right]\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j}$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}\left(1-\hat{y}_\theta\left(x^{(i)}\right)\right)-\left(\left(1-y^{(i)}\right)\hat{y}_\theta\left(x^{(i)}\right)\right)}{\hat{y}_\theta(x^{(i)})\left(1-\hat{y}_\theta(x^{(i)})\right)}\right]\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j}$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{\left(y^{(i)}-y^{(i)}\hat{y}_\theta\left(x^{(i)}\right)\right)-\left(\left(\hat{y}_\theta\left(x^{(i)}\right)-y^{(i)}\hat{y}_\theta\left(x^{(i)}\right)\right)\right)}{\hat{y}_\theta(x^{(i)})\left(1-\hat{y}_\theta(x^{(i)})\right)}\right]\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j}$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}-y^{(i)}\hat{y}_\theta\left(x^{(i)}\right)-\hat{y}_\theta\left(x^{(i)}\right)+y^{(i)}\hat{y}_\theta\left(x^{(i)}\right)}{\hat{y}_\theta(x^{(i)})\left(1-\hat{y}_\theta(x^{(i)})\right)}\right]\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j}$$

Regresión logística – Modificando parámetros (cont.)

Derivada sigmoide / logística

$$y = \frac{1}{1 + e^{-x}}$$

$$y' = y(1 - y)$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)} - \hat{y}_\theta\left(x^{(i)}\right)}{\hat{y}_\theta\left(x^{(i)}\right)\left(1 - \hat{y}_\theta\left(x^{(i)}\right)\right)}\right]\frac{\partial \hat{y}_\theta\left(x^{(i)}\right)}{\partial \theta_j}$$

$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)} - \hat{y}_\theta\left(x^{(i)}\right)}{\hat{y}_\theta\left(x^{(i)}\right)\left(1 - \hat{y}_\theta\left(x^{(i)}\right)\right)}\right]\hat{y}_\theta\left(x^{(i)}\right)\left(1 - \hat{y}_\theta\left(x^{(i)}\right)\right)\frac{\partial \theta x^{(i)}}{\partial \theta_j}$$

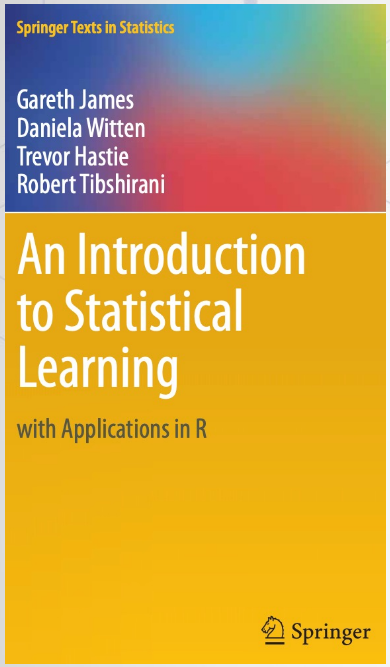
$$\frac{\partial Costo(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}_\theta\left(x^{(i)}\right)\right)\frac{\partial \theta x^{(i)}}{\partial \theta_j}$$

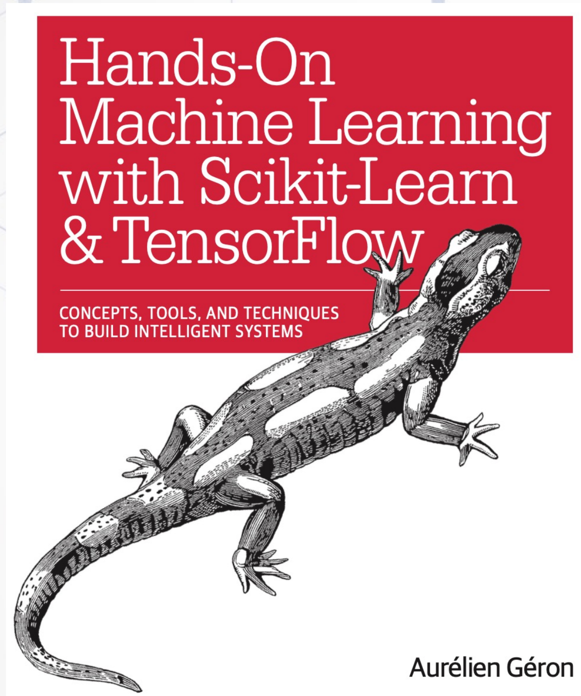
$$\frac{\partial Costo(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

HUB IQ

EngineeringX
Founded by the Royal Academy of Engineering and Lloyd's Register Foundation

Referencias extra





Referencias extra

Linear Regression

In Chapter 1, we looked at a simple regression model of life satisfaction: *life_satisfaction* $= \theta_0 + \theta_1 \times$ *GDP_per_capita*.

This model is just a linear function of the input feature `GDP_per_capita`. θ_0 and θ_1 are the model's parameters.

More generally, a linear model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the *bias term* (also called the *intercept term*), as shown in Equation 4-1.

Equation 4-1. Linear Regression model prediction

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- $\hat{y}$ is the predicted value.
- n is the number of features.
- x_i is the ith feature value.
- θ_j is the jth model parameter (including the bias term θ_0 and the feature weights $\theta_1, \theta_2, \cdots, \theta_n$).

Equation 4-2. Linear Regression model prediction (vectorized form)

$$\hat{y} = h_\theta(\mathbf{x}) = \theta^T \cdot \mathbf{x}$$

Equation 4-3. MSE cost function for a Linear Regression model

$$\mathrm{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

Equation 4-5. Partial derivatives of the cost function

$$\frac{\partial}{\partial \theta_j} \mathrm{MSE}(\theta) = \frac{2}{m} \sum_{i=1}^{m} \left(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)}\right) x_j^{(i)}$$

Referencias extra

Logistic Regression

As we discussed in Chapter 1, some regression algorithms can be used for classification as well (and vice versa). *Logistic Regression* (also called *Logit Regression*) is commonly used to estimate the probability that an instance belongs to a particular class (e.g., what is the probability that this email is spam?). If the estimated probability is greater than 50%, then the model predicts that the instance belongs to that class (called the positive class, labeled "1"), or else it predicts that it does not (i.e., it belongs to the negative class, labeled "0"). This makes it a binary classifier.

Estimating Probabilities

So how does it work? Just like a Linear Regression model, a Logistic Regression model computes a weighted sum of the input features (plus a bias term), but instead of outputting the result directly like the Linear Regression model does, it outputs the *logistic* of this result (see Equation 4-13).

Equation 4-13. Logistic Regression model estimated probability (vectorized form)

$$\hat{p} = h_{\theta}(\mathbf{x}) = \sigma\left(\theta^T \cdot \mathbf{x}\right)$$

The logistic—also called the *logit*, noted $\sigma(\cdot)$—is a *sigmoid function* (i.e., *S*-shaped) that outputs a number between 0 and 1. It is defined as shown in Equation 4-14 and Figure 4-21.

Equation 4-14. Logistic function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

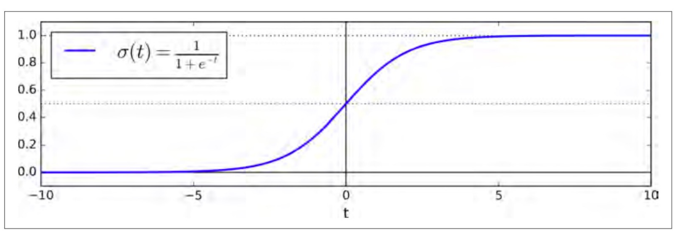


Figure 4-21. Logistic function

Equation 4-15. Logistic Regression model prediction

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5, \\ 1 & \text{if } \hat{p} \geq 0.5. \end{cases}$$

Equation 4-16. Cost function of a single training instance

$$c(\theta) = \begin{cases} -\log(\hat{p}) & \text{if } y = 1, \\ -\log(1 - \hat{p}) & \text{if } y = 0. \end{cases}$$

Equation 4-17. Logistic Regression cost function (log loss)

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{p}^{(i)}\right)\right]$$

Equation 4-18. Logistic cost function partial derivatives

$$\frac{\partial}{\partial \theta_j} \mathrm{J}(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\sigma\left(\theta^T \cdot \mathbf{x}^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

HUB IQ

EngineeringX
Founded by the Royal Academy of Engineering and Lloyd's Register Foundation

Algoritmo

$\theta \leftarrow$ *valores aleatorios, p.ej.* $\in [0,1]$

Para cada j (característica)

i datos/muestra

Repetir {

$$\theta_j \leftarrow \theta_j - \alpha \times \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}_\theta \left(x^{(i)} \right) - y^{(i)} \right) x^{(i)}$$

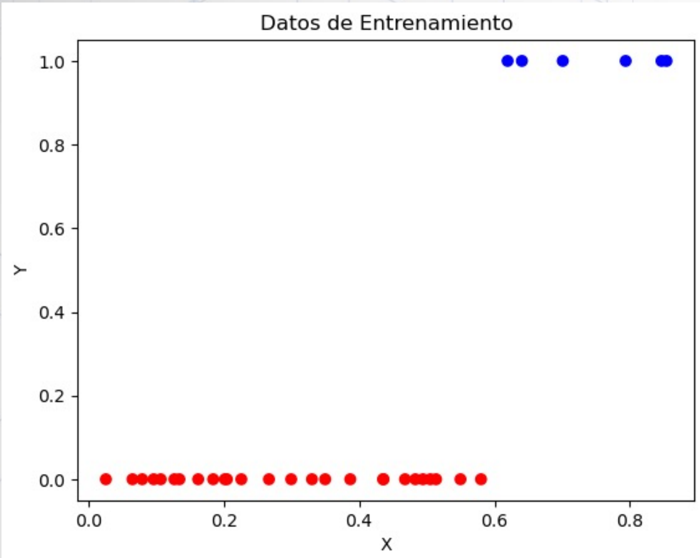
}

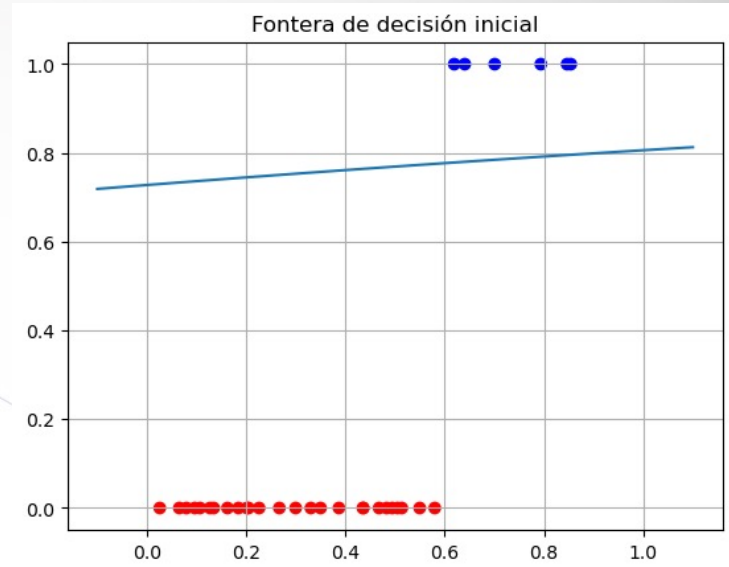
$$\hat{y}_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1)}}$$

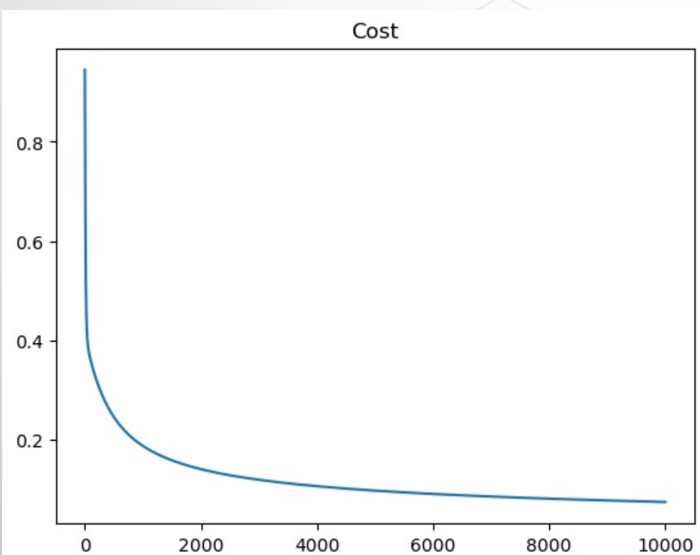
Regresión Logística

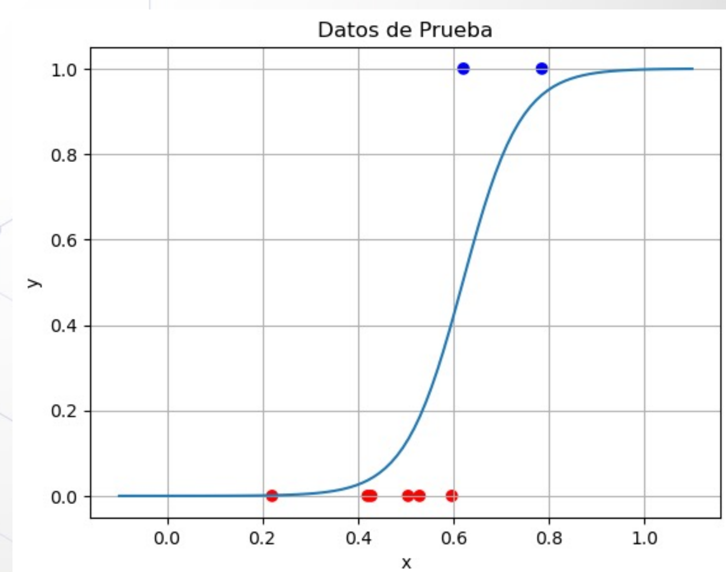
Implementación en Python

1D


Datos de Entrenamiento
Y
X
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8


Fontera de decisión inicial
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0


Cost
0.8
0.6
0.4
0.2
0 2000 4000 6000 8000 10000


Datos de Prueba
y
x
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0

Algoritmo – 2D

$$\hat{y}_\theta(x) = \frac{1}{1 + e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2)}}$$

3D

$$\hat{y}_\theta(x) = \frac{1}{1 + e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_3)}}$$

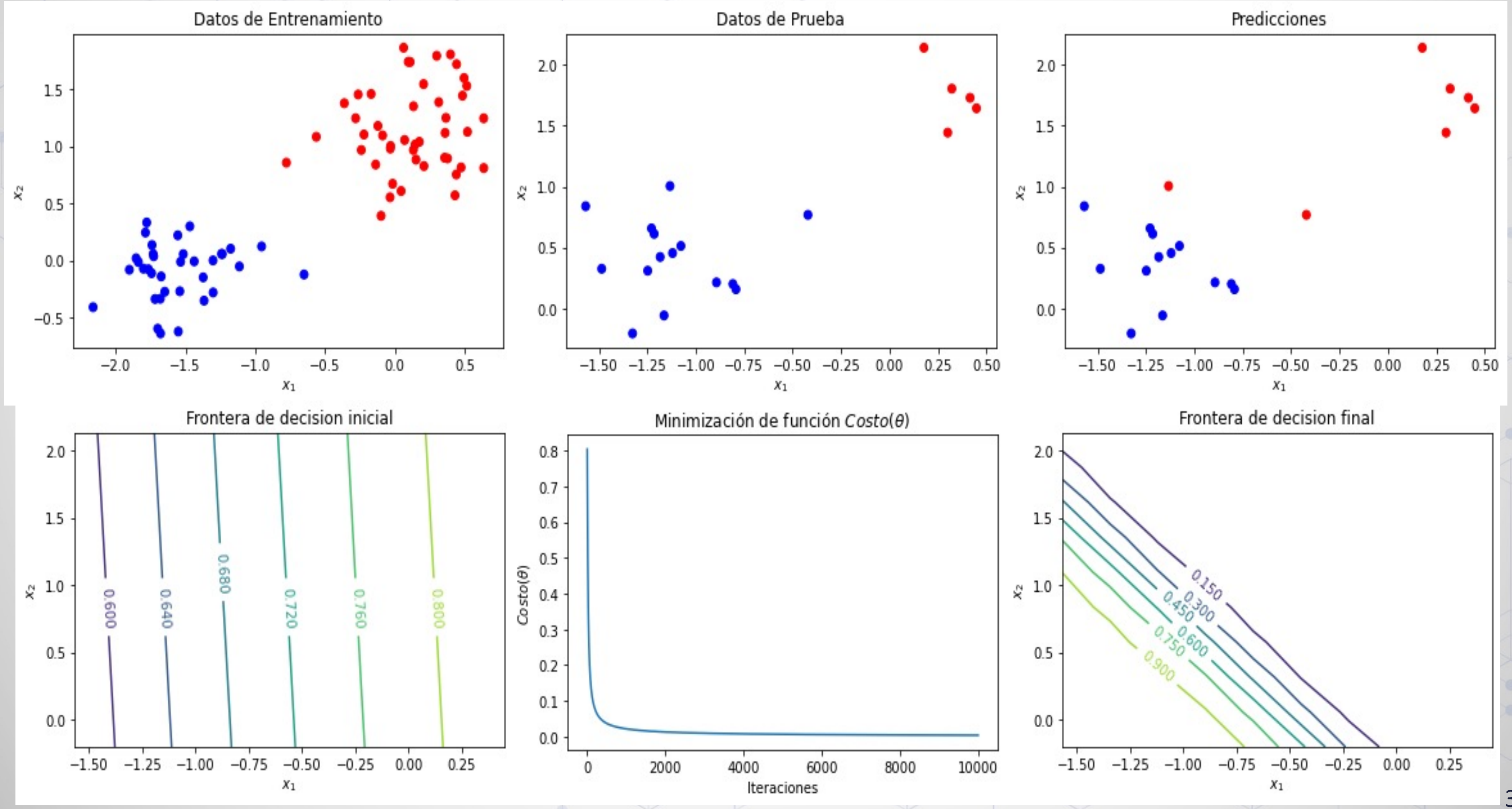
$$\hat{y}_\theta(x) = \frac{1}{1 + e^{-(\theta_0+\theta_1 x_1^2+\theta_2 x_1^2)}}$$

nD

$$\hat{y}_\theta(x) = \frac{1}{1 + e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\cdots+\theta_n x_n)}}$$

Algoritmo – 2D

$$\hat{y}_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2)}}$$



Validar con Scikit learn

```python
from sklearn.linear_model import LogisticRegression
modeloRL = LogisticRegression(random_state=0,max_iter=10000).fit(X_train, y_train)
print(modeloRL.predict(X_test))
print(modeloRL.predict_proba(X_test))
```

sklearn.linear_model.LogisticRegression

class `sklearn.linear_model.`**`LogisticRegression`**(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)

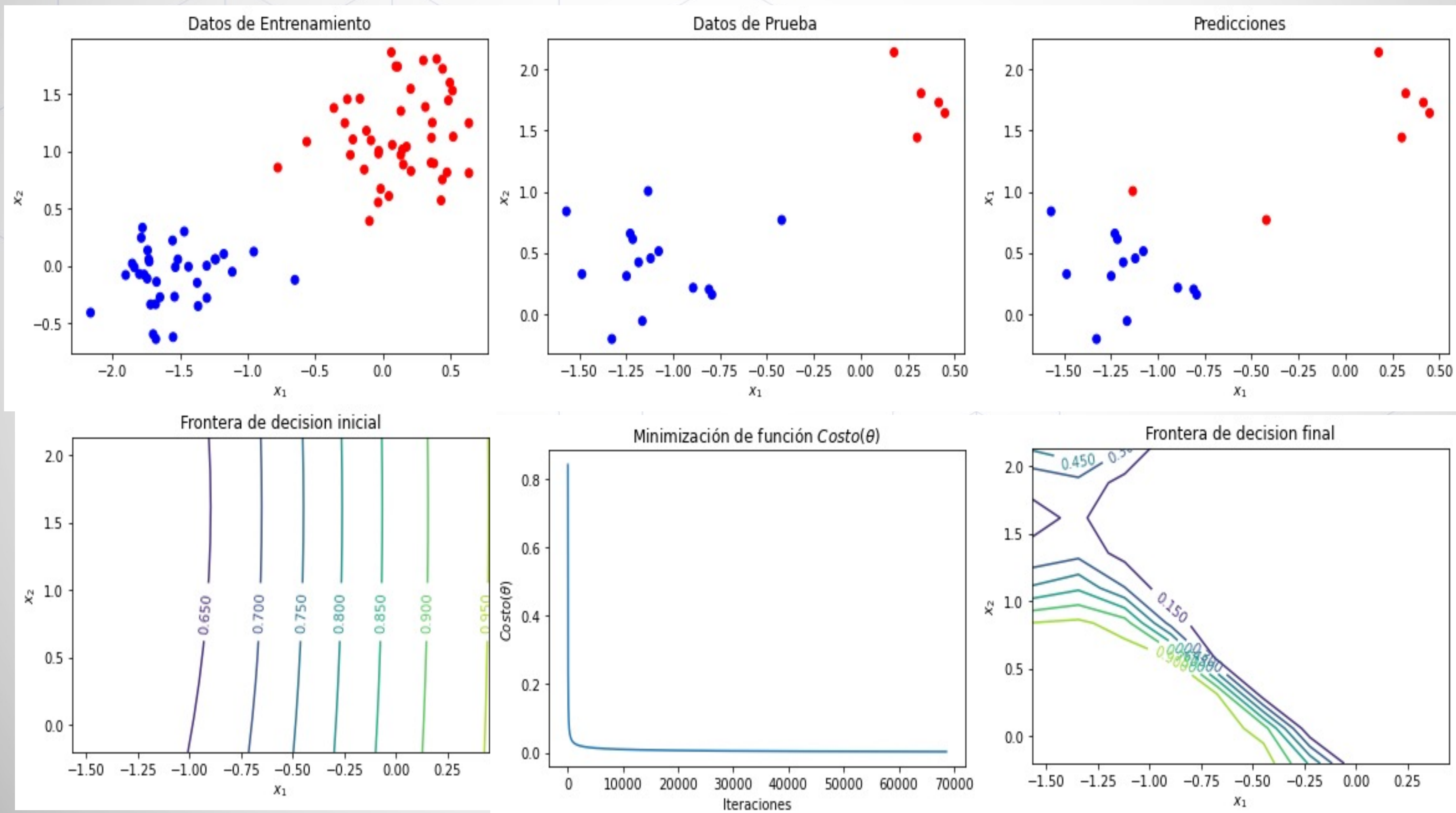
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. **Note that regularization is applied by default**. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.

Read more in the User Guide.

Algoritmo – 2D

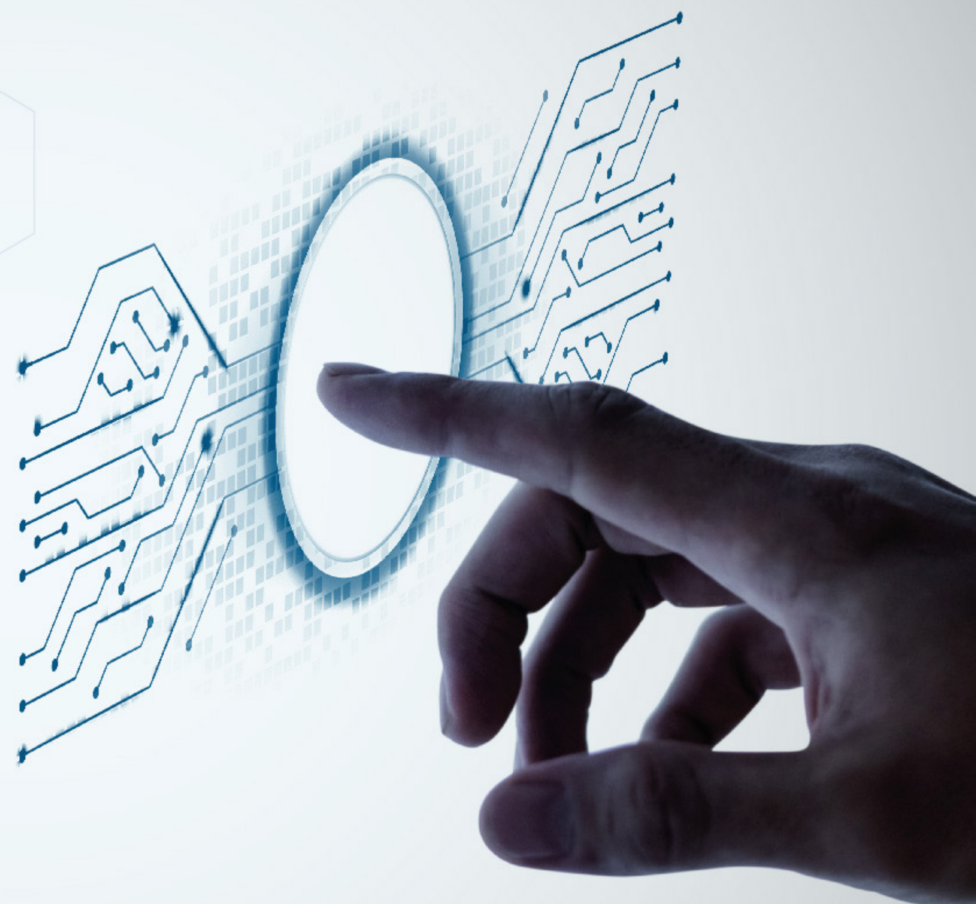
$$\hat{y}_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1^2 + \theta_2 x_1^2)}}$$



EngineeringX

Founded by the Royal Academy of Engineering
and Lloyd's Register Foundation

GRACIAS



https://hubiq.mx/

HUBIQRO HUBIQ HUBIQRO