

Introducción a la Inteligencia Artificial
Facultad de Ingeniería
Universidad de Buenos Aires

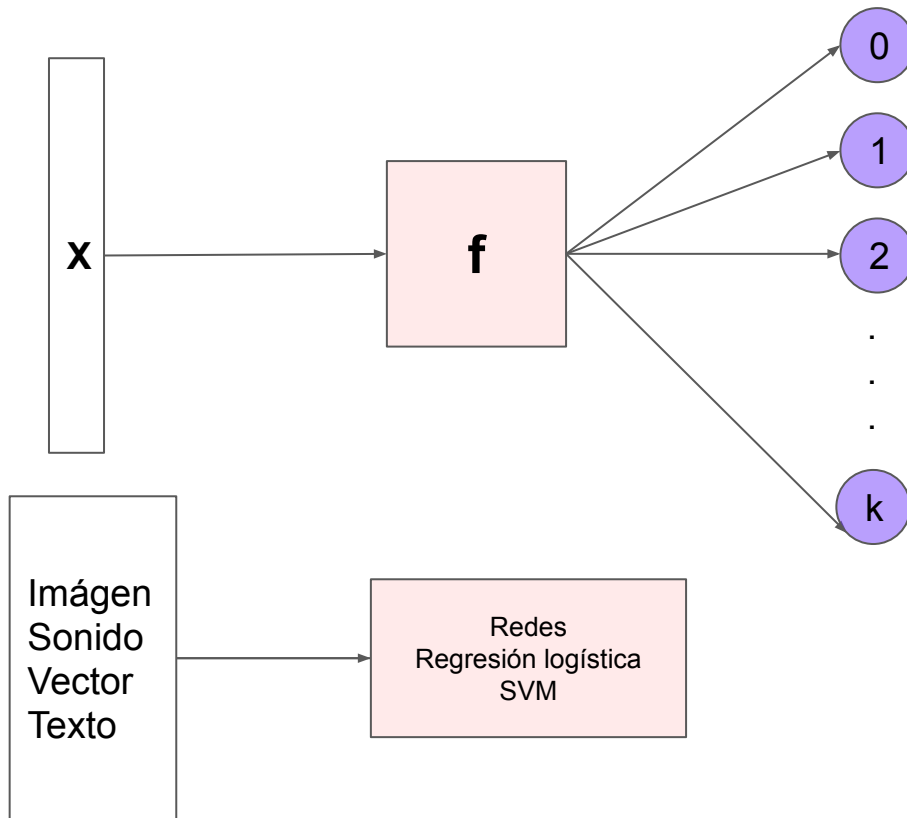


Clase 5

1. Clasificación Binaria
 - a. Motivación
 - b. Regresión Logística - Ejercicio de Aplicación
 - c. Regresión Logística - Teoría
2. Clasificación Multiclase
 - a. Motivación
 - b. Softmax
 - c. Ejercicio de Aplicación
3. Ejercicio integrador

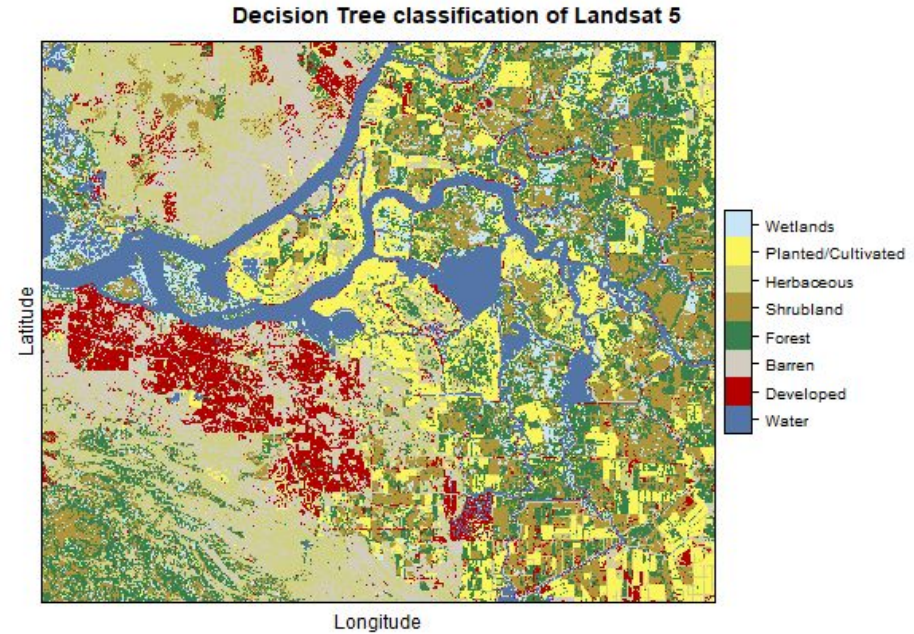
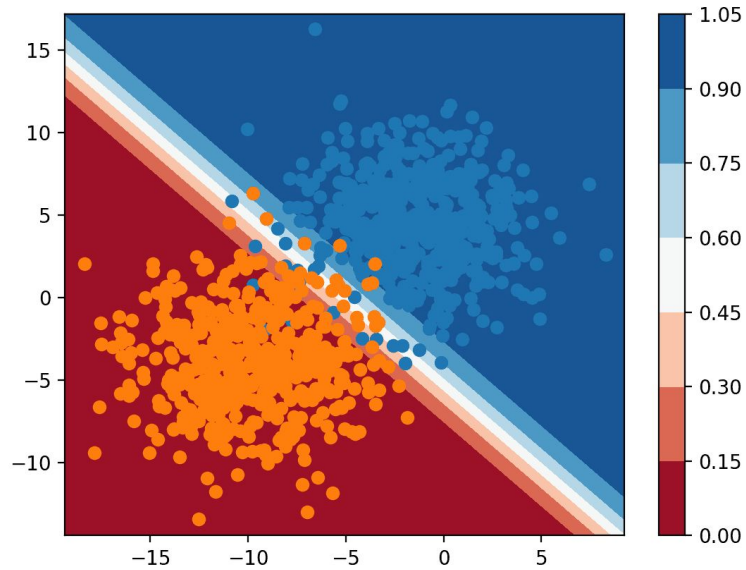


Clasificación



$$f : R^D \rightarrow \{0, 1, 2, \dots, k\}$$

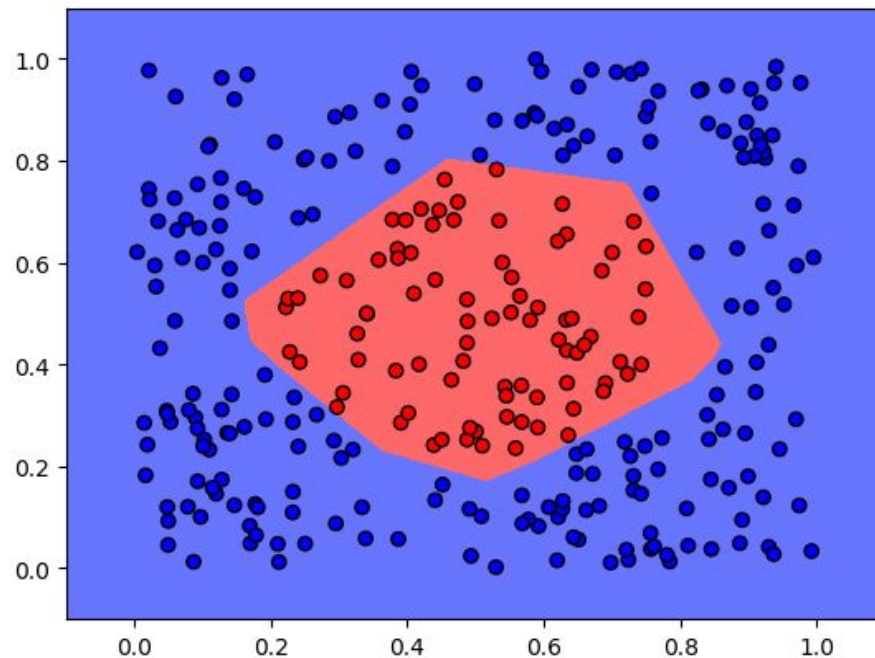
Clasificación



Clasificación binaria

Clasificación Binaria - Ejemplos

- Detección de fraudes
- Diagnóstico médico
- Detección de spam
- Sentiment Analysis
- Detección de objetos
- Outliers



Clasificación Binaria - Diagnóstico médico

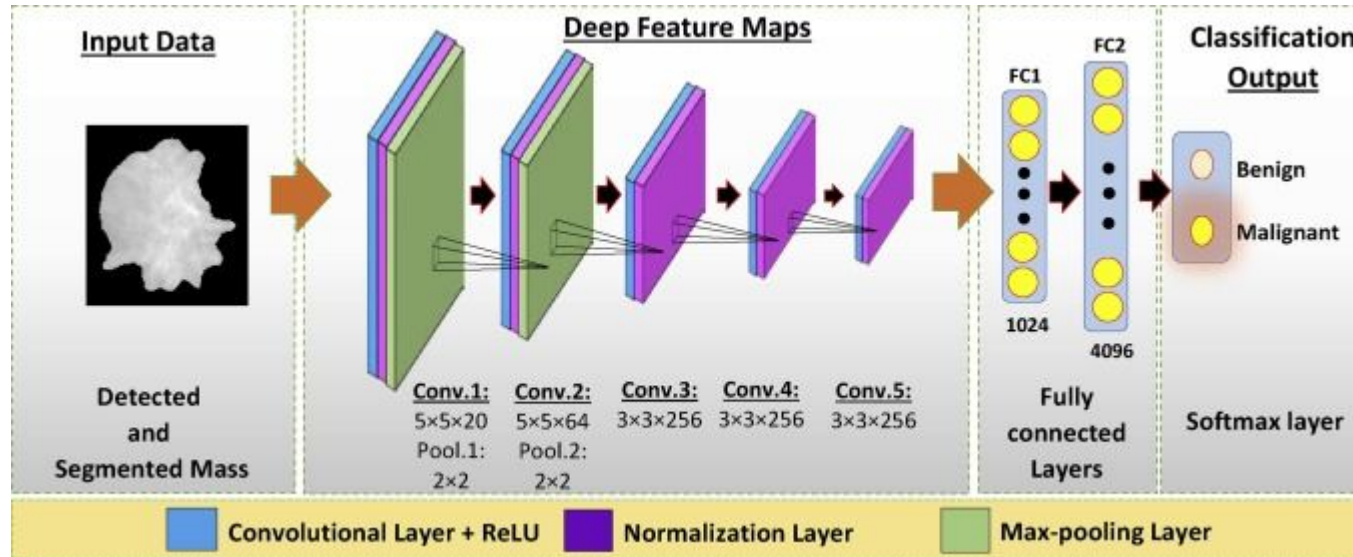


Imagen de: "A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification"

Clasificación Binaria - Spam detection

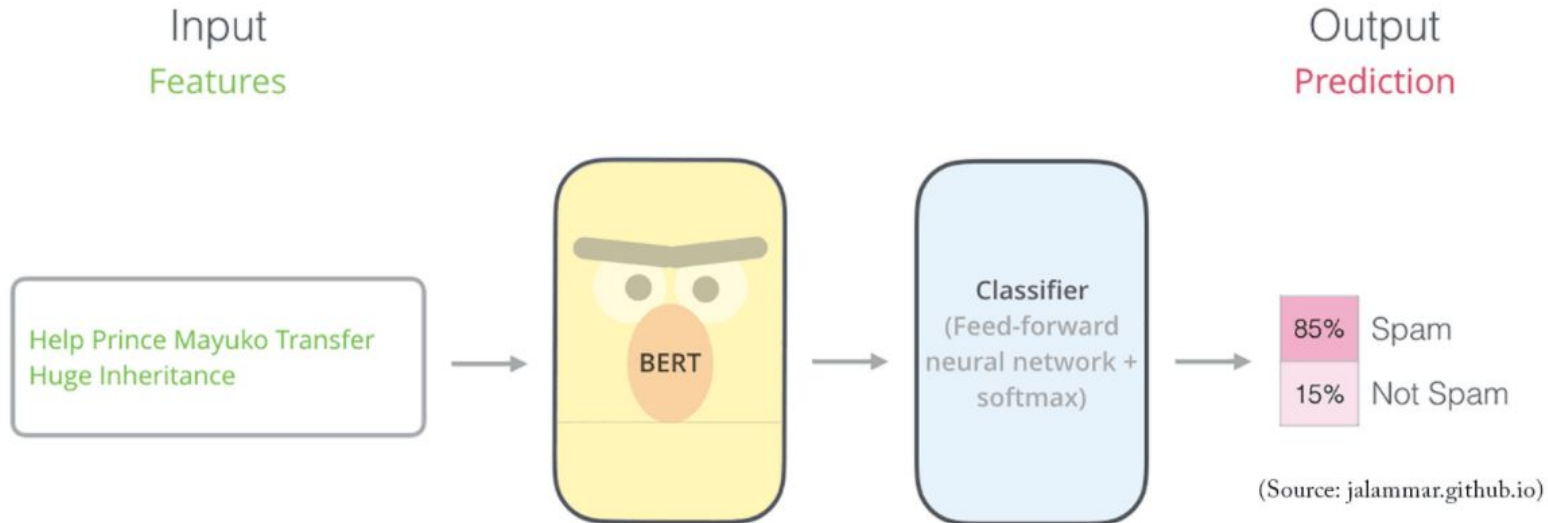
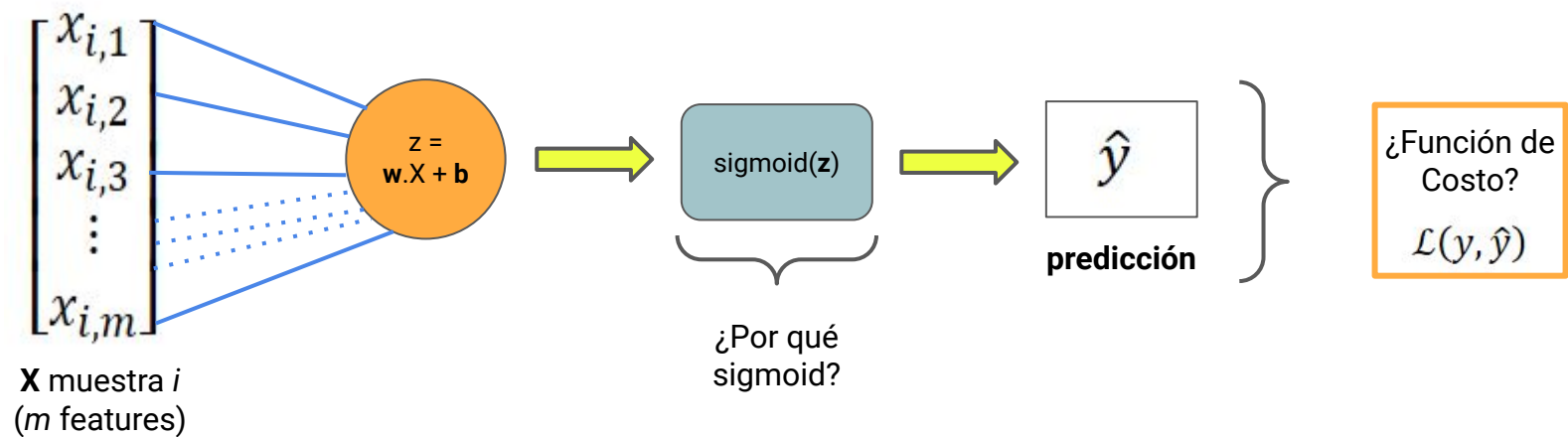


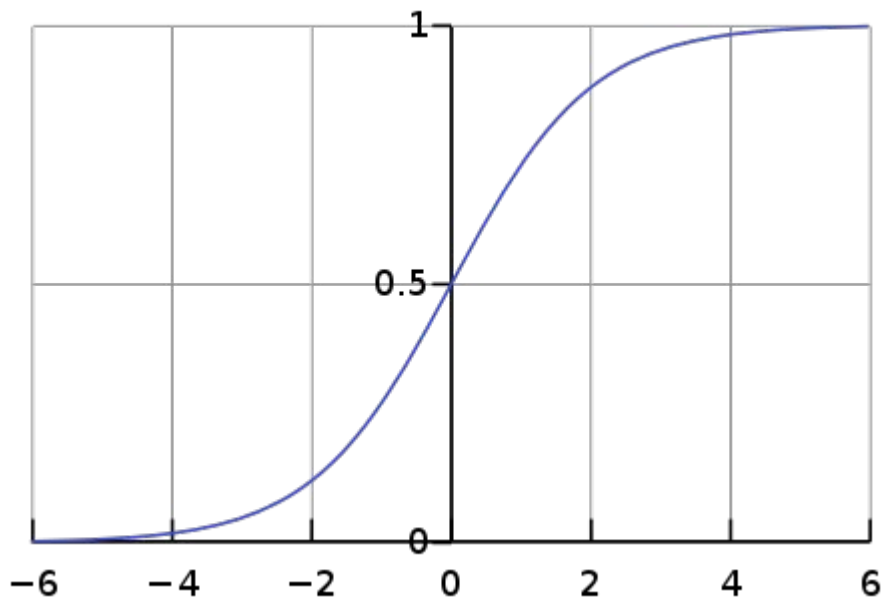
Imagen de: "Tutorial: Fine tuning BERT for Sentiment Analysis"

Regresión Logística

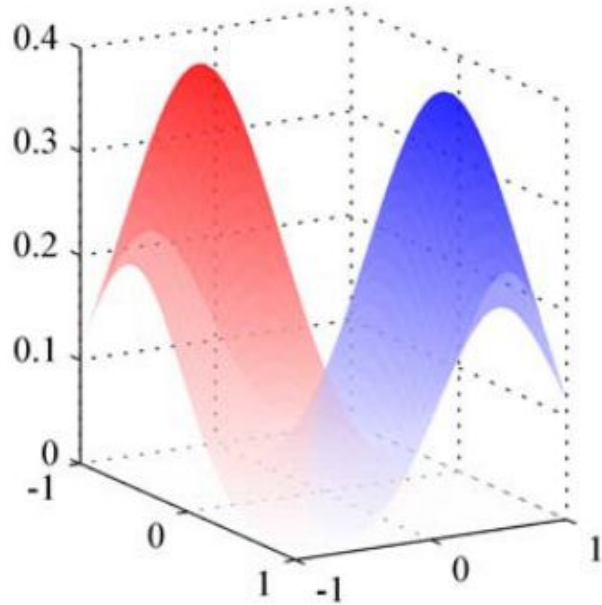


Logistic function

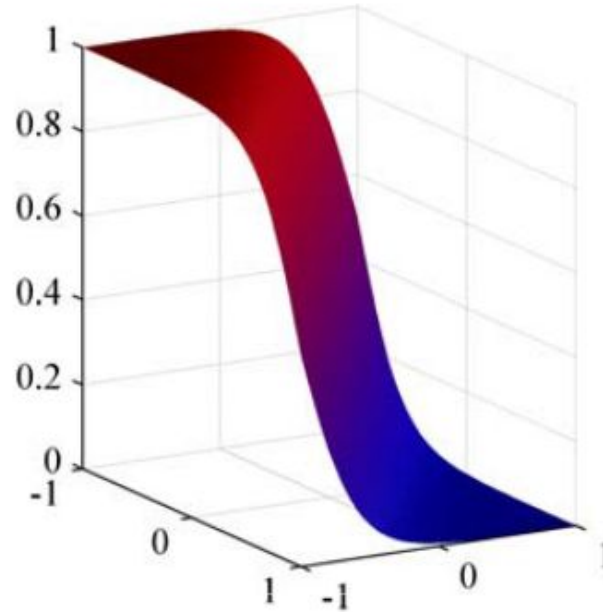
$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x).$$



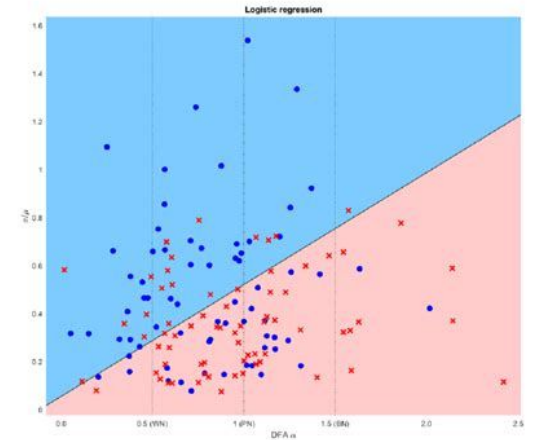
Regresión Logística



Class-conditional - $P(x|C_n)$

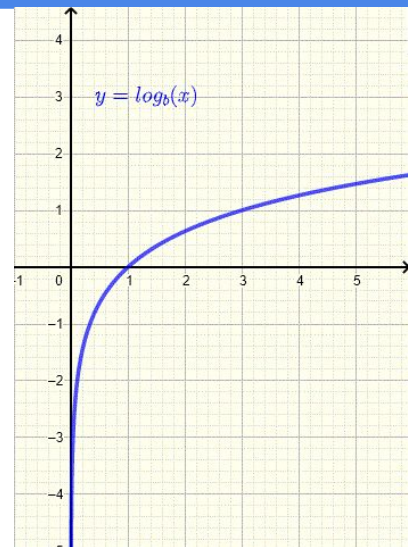
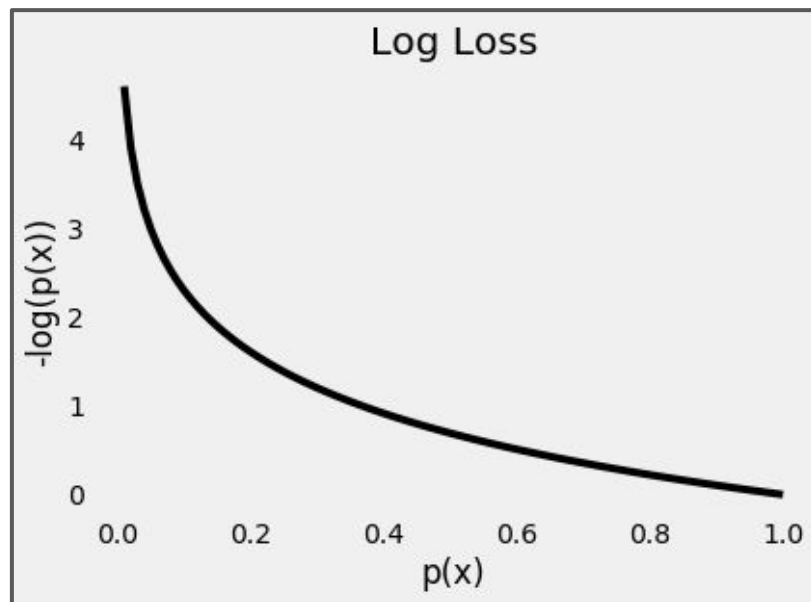


Posterior - $P(C_n|x)$



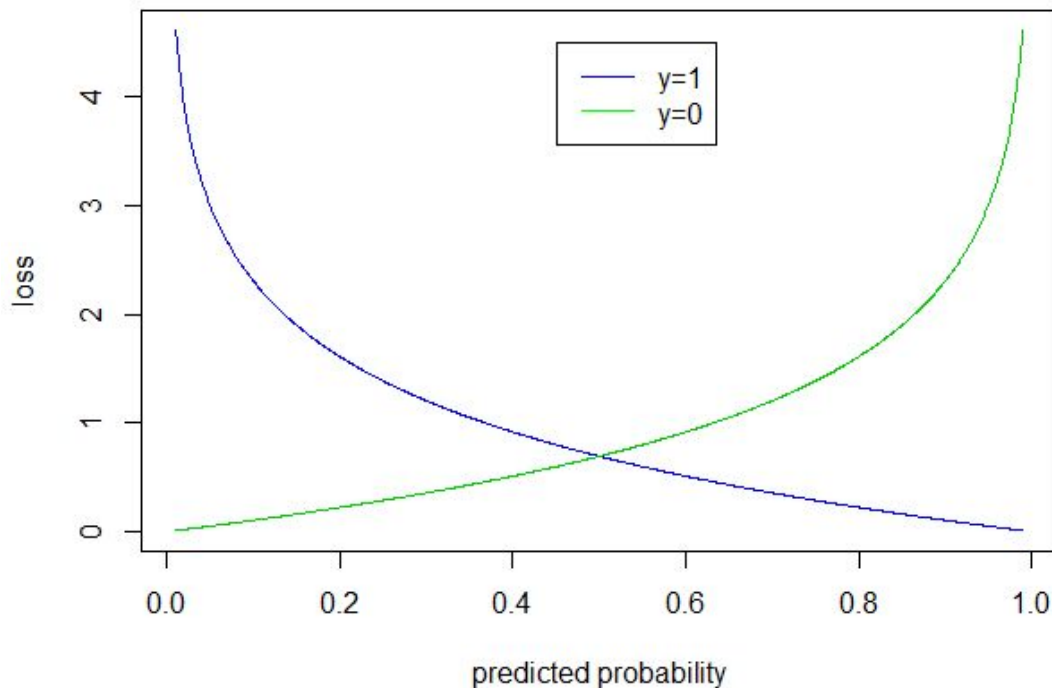
Función de costo - Binary cross entropy

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

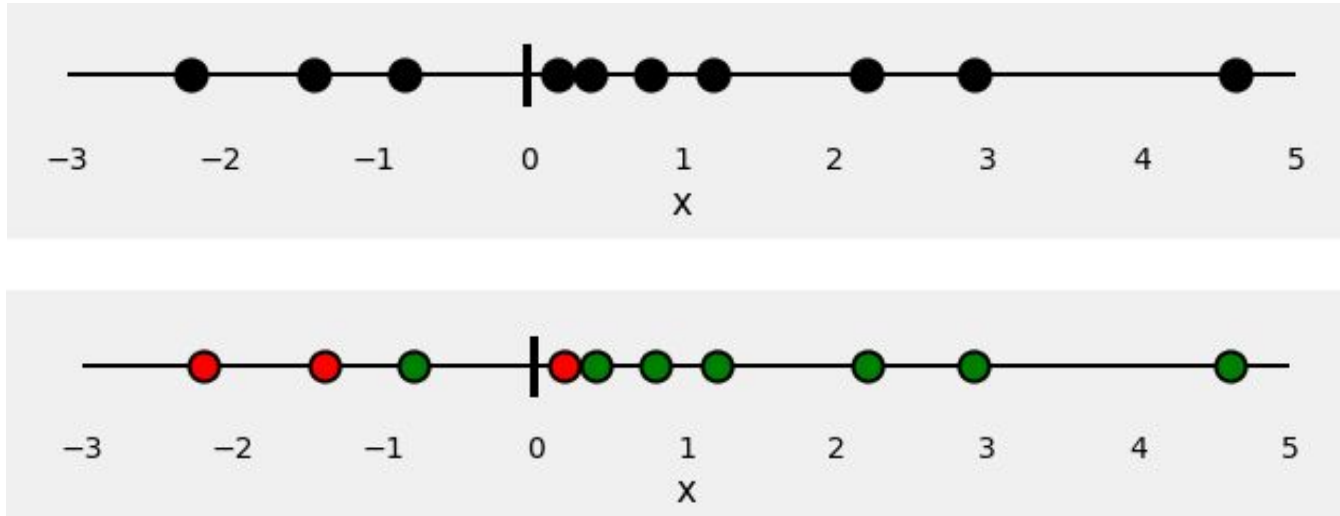


Función de costo - Binary cross entropy

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

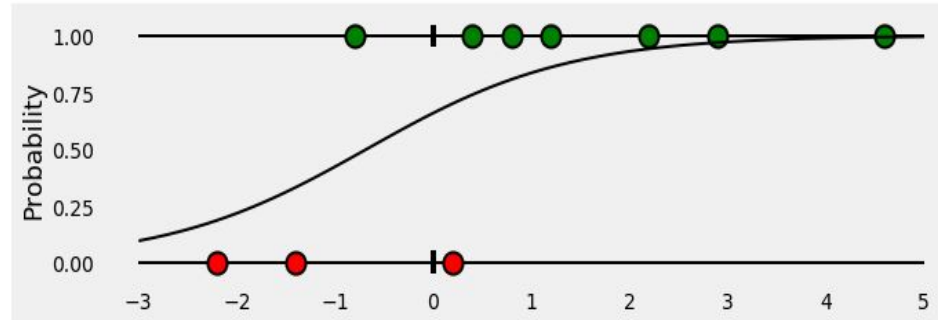
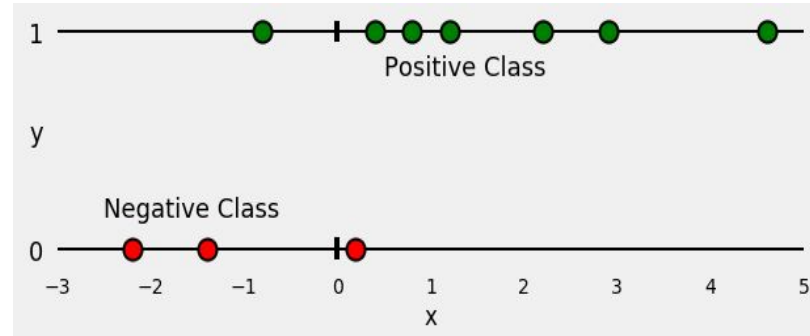


Regresión Logística

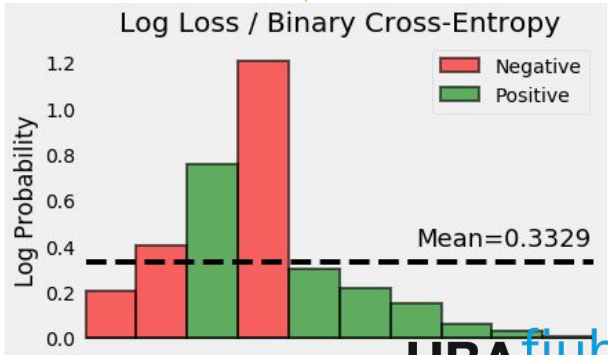
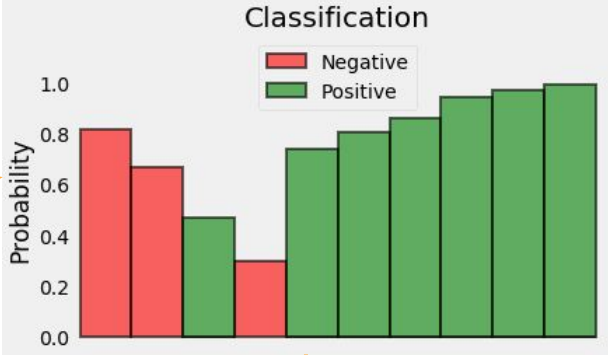
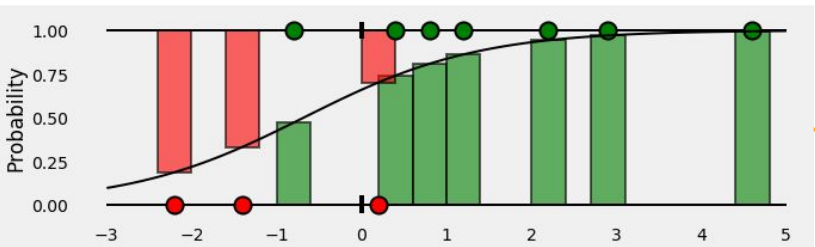
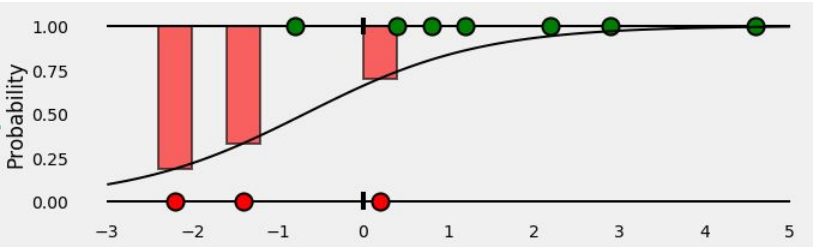
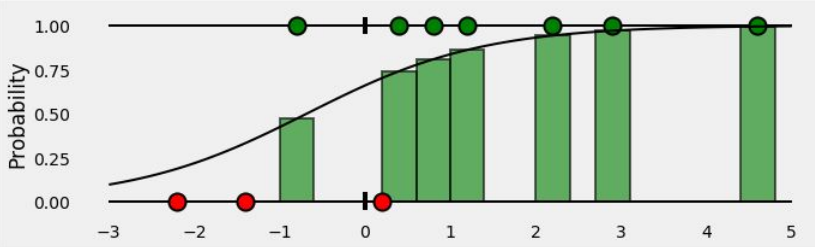


1: Verde, 0: Rojo

Regresión Logística

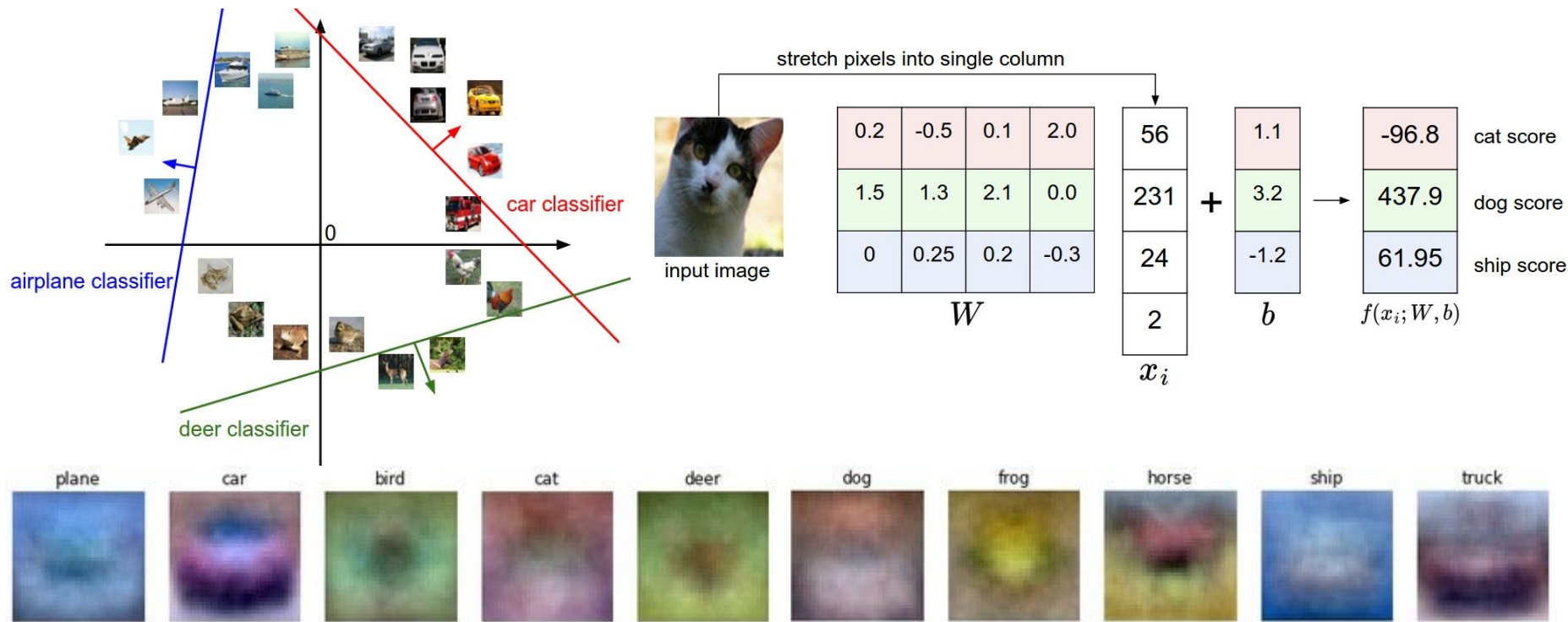


Regresión Logística

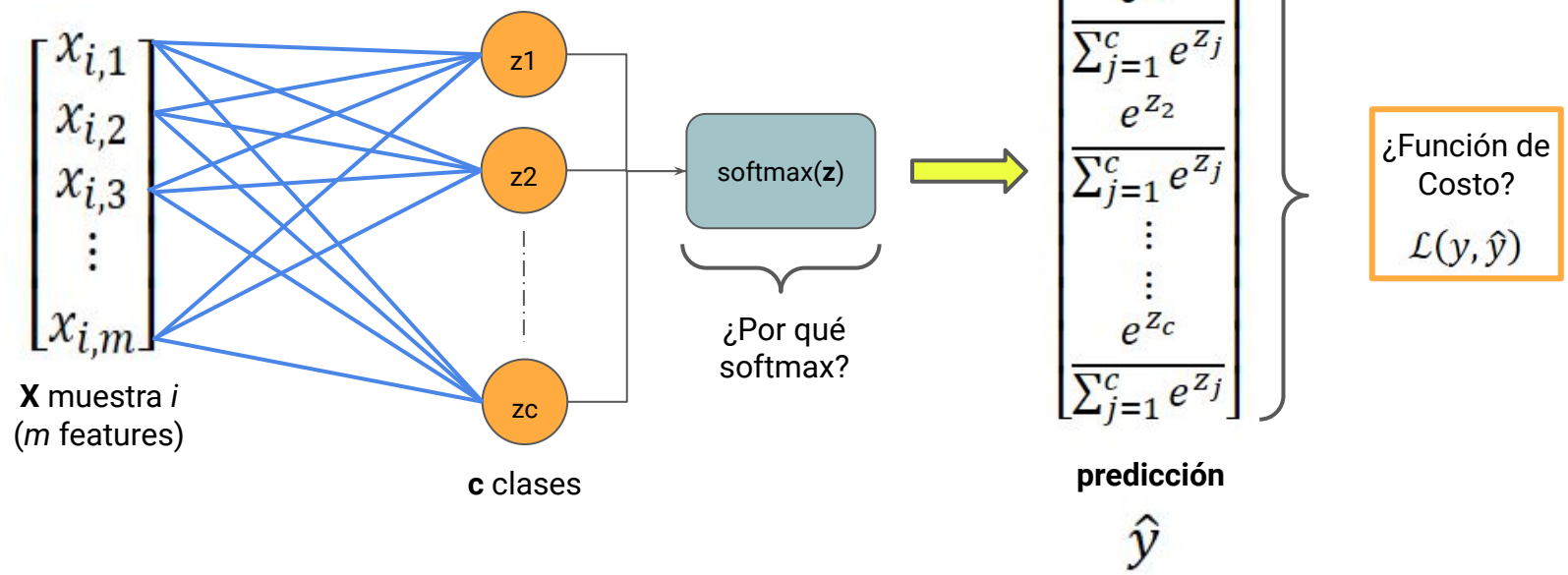


Clasificación multiclase

Clasificación Multiclase - Motivación



Softmax



Softmax

$$P(y_i | x_i; W) = \frac{e^{f_{y_i}}}{\sum_j e^{f_j}}$$

$$\frac{e^{f_{y_i}}}{\sum_j e^{f_j}} = \frac{C e^{f_{y_i}}}{C \sum_j e^{f_j}} = \frac{e^{f_{y_i} + \log C}}{\sum_j e^{f_j + \log C}}$$

$q(x)$

$$H(p, q) = - \sum_x p(x) \log q(x)$$

1

[Softmax Forma Gráfica](#)

2

[Softmax Visualización 3D](#)

Softmax

Derivación Softmax

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}}$$

$$\frac{\partial p_i}{\partial z_k} = \frac{\partial \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}}}{\partial z_k}$$

$$\frac{\partial p_i}{\partial z_k} = p_i(\delta_{ik} - p_k) \quad \delta_{ik} = \begin{cases} 1, i = k \\ 0, i \neq k \end{cases}$$

Derivación Cross-Entropy

$$\begin{aligned} L &= - \sum_i y_i \log(p_i) \\ \frac{\partial L}{\partial z_i} &= - \sum_j y_j \frac{\partial \log(p_j)}{\partial z_i} \\ &= - \sum_j y_j \frac{\partial \log(p_j)}{\partial p_j} \times \frac{\partial p_j}{\partial z_i} \\ &= - \sum_j y_j \frac{1}{p_j} \times \frac{\partial p_j}{\partial z_i} \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial z_i} &= -y_i(1 - p_i) - \sum_{j \neq i} y_j \frac{1}{p_j} (-p_j \cdot p_i) \\ &= -y_i(1 - p_i) + \sum_{j \neq i} y_j \cdot p_i \\ &= p_i \left(y_i + \sum_{j \neq i} y_j \right) - y_i \end{aligned}$$

$$\frac{\partial L}{\partial z_i} = p_i - y_i$$

$$\frac{\partial z_i}{\partial W} = x_i$$

$$\frac{\partial L}{\partial W} = \sum_{i=1}^N (p_i - y_i) x_i$$

Usar gradiente descendente para actualizar W !!!

Ejercicio integrador

1. Implementar el algoritmo de regresión logística en NumPy.
2. Aplicar el modelo a un dataset de elección.
3. Comparar los resultados con Scikit-Learn .
4. Comparar los resultados agregando regularización.

Bibliografía

- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | <https://www.deeplearningbook.org/>
- Mathematics for Machine Learning | Deisenroth, Faisal, Ong
- Artificial Intelligence, A Modern Approach | Stuart J. Russell, Peter Norvig
- Understanding binary cross-entropy: a visual explanation | Daniel Godoy
- Visual Information Theory | [Link](#)
- <https://cs231n.github.io/>
- Classification and Loss Evaluation-Softmax and Cross Entropy Loss | Paras Dahal