Regresión Lineal con **Múltiples** Variables de **Salida**

Regresión Lineal - Versiones

- **Regresión Lineal Simple**
  - **1** variable de entrada
  - **1** variable de salida
  - Modelo
    - f(x) = m x +b
- **Regresión Lineal**
  - **m** variables de entrada
  - **1** variable de salida
  - Modelo
    - f(x)=x[1] w[1] +x[2] w[2]+...+x[m] w[m] + b

- **Regresión Lineal Múltiple**
  - **m** variables de entrada
  - **k** variables de salida

Regresión Lineal Múltiple - Motivación

- Predecir M valores
  - a partir de N valores
- Transformar un vector en otro
  - **Vector de flotantes**
    - codificación **universal**
- Puede usarse para
  - **Predecir** la posición de un objeto
    - a partir de una imagen
  - **Generar** un texto
    - A partir de un sonido
  - **Generar** una imagen
    - A partir de un vector de características
  - Etc...

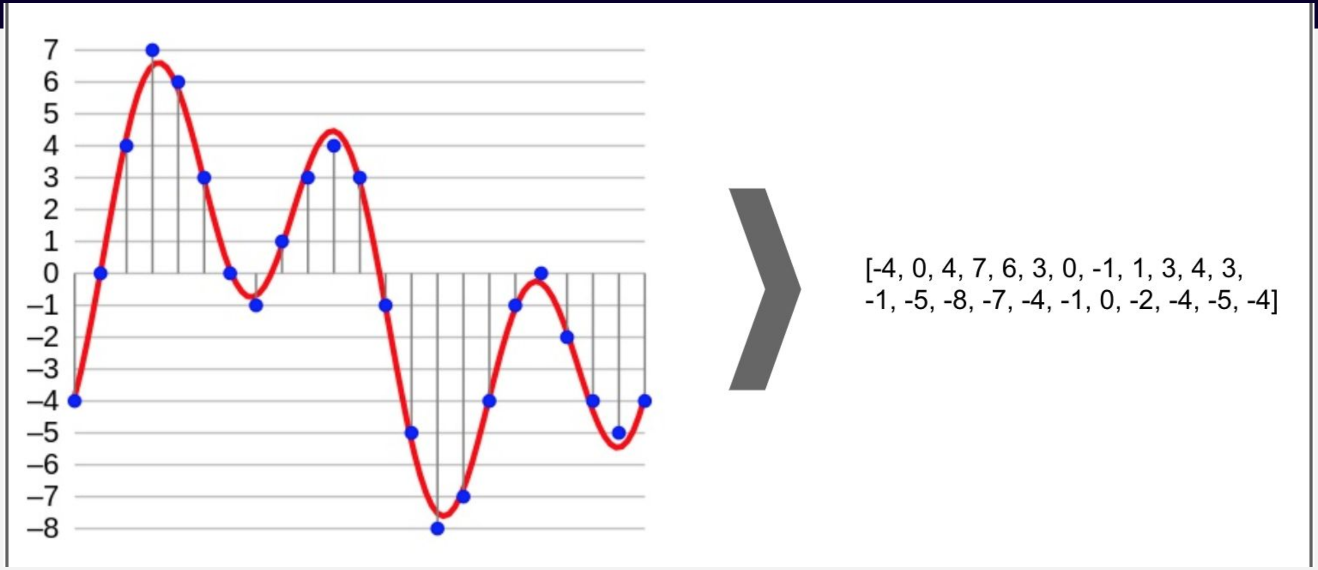
Estudio	Edad	Promedio	N1	N2
2	24	4	7	7.2
5	22	3	4.5	5.2
7	25	4	6.3	6
9	20	7	5.4	4.4
10	19	4	8.2	9.5
11	20	3	7.2	8.1
13,4	21	5	5,5	10
14	20	3	3	4.3

Regresión Lineal Múltiple - Aplicaciones

- **Predecir** la **posición** de un cartel
  - vector de 4 valores: (x,y,alto,ancho)
- **A partir** de una imagen
  - vector de dimensiones: ancho*alto*colores

x	y	alto	ancho
300	200	50	400

Regresión Lineal Múltiple - Aplicaciones



p_1	p_2	p_3	p_4
4	3	2	6

Diccionario

Palabra	Código
hola	1
curso	2
buen	3
que	4
la	5
che	6
el	7

- **Generar** un **texto**
  - vector de P dimensiones
- **A partir** de un **sonido**
  - vector de t dimensiones

que buen curso che

Regresión Lineal Múltiple - Aplicaciones

- **Generar** una **imagen**
- **A partir** de un vector de **características**

Perro	Gato	Exterior
1	0	0

Perro	Gato	Exterior
1	0	1

Perro	Gato	Exterior
0	1	0

Perro	Gato	Exterior
1	1	0

Regresión Lineal Múltiple - Problema

Predecir una variable

Estudio	Edad	Promedio	Nota
2	24	4	7
5	22	3	4.5
7	25	4	6.3
9	20	7	5.4
10	19	4	8.2
11	20	3	7.2
13,4	21	5	5,5
14	20	3	3

Predecir múltiples variables

Estudio	Edad	Promedio	N1	N2
2	24	4	7	7.2
5	22	3	4.5	5.2
7	25	4	6.3	6
9	20	7	5.4	4.4
10	19	4	8.2	9.5
11	20	3	7.2	8.1
13,4	21	5	5,5	10
14	20	3	3	4.3

Regresión Lineal Múltiple - Versión Simple

Dos salidas → Dos modelos

Estudio	Edad	Promedio	N1
2	24	4	7
5	22	3	4.5
7	25	4	6.3
9	20	7	5.4
10	19	4	8.2
11	20	3	7.2
13,4	21	5	5,5
14	20	3	3

Modelo para N1

Estudio	Edad	Promedio	N2
2	24	4	7.2
5	22	3	5.2
7	25	4	6
9	20	7	4.4
10	19	4	9.5
11	20	3	8.1
13,4	21	5	10
14	20	3	4.3

Modelo para N2

Regresión Lineal Múltiple - Versión Simple

Modelo para N1

Entrada
X: 3 números
Parámetros
W1: 3 números
B1: 1 número
Salida
N1: 1 número

Modelo para N2

Entrada
X: 3 números
Parámetros
W2: 3 números
B2: 1 número
Salida
N2: 1 número

Fórmula del modelo para N1

```
N1(X) = (X[1] * W1[1] + X[2] * W1[2] + X[3] * W1[3])+B1
      = X . W1 + B1
```

En Python

```
N1 = np.dot(W1,X) + B1
```

X . W es el producto **punto** o **escalar**.

```
def dot(X,W):
  p=0
  for i in range(3):
    p=p+(X[i]*W[i])
  return p
```

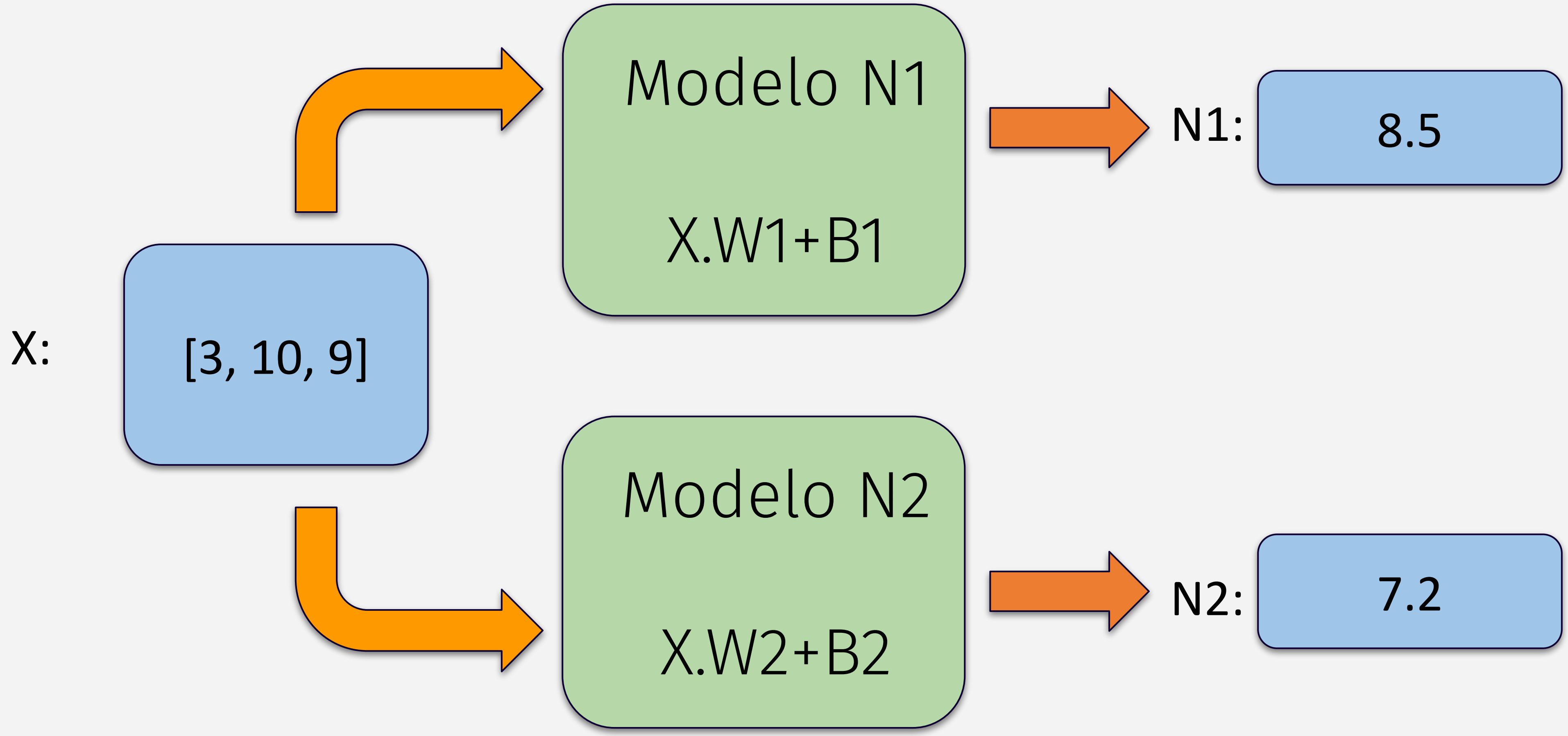
Fórmula del modelo para N2

```
N2(X) = (X[1] * W2[1] + X[2] * W2[2] + X[3] * W2[3])+B2
      = X . W2 + B2
```

En Python

```
N2 = np.dot(W2,X) + B2
```

Regresión Lineal Múltiple - Versión Simple

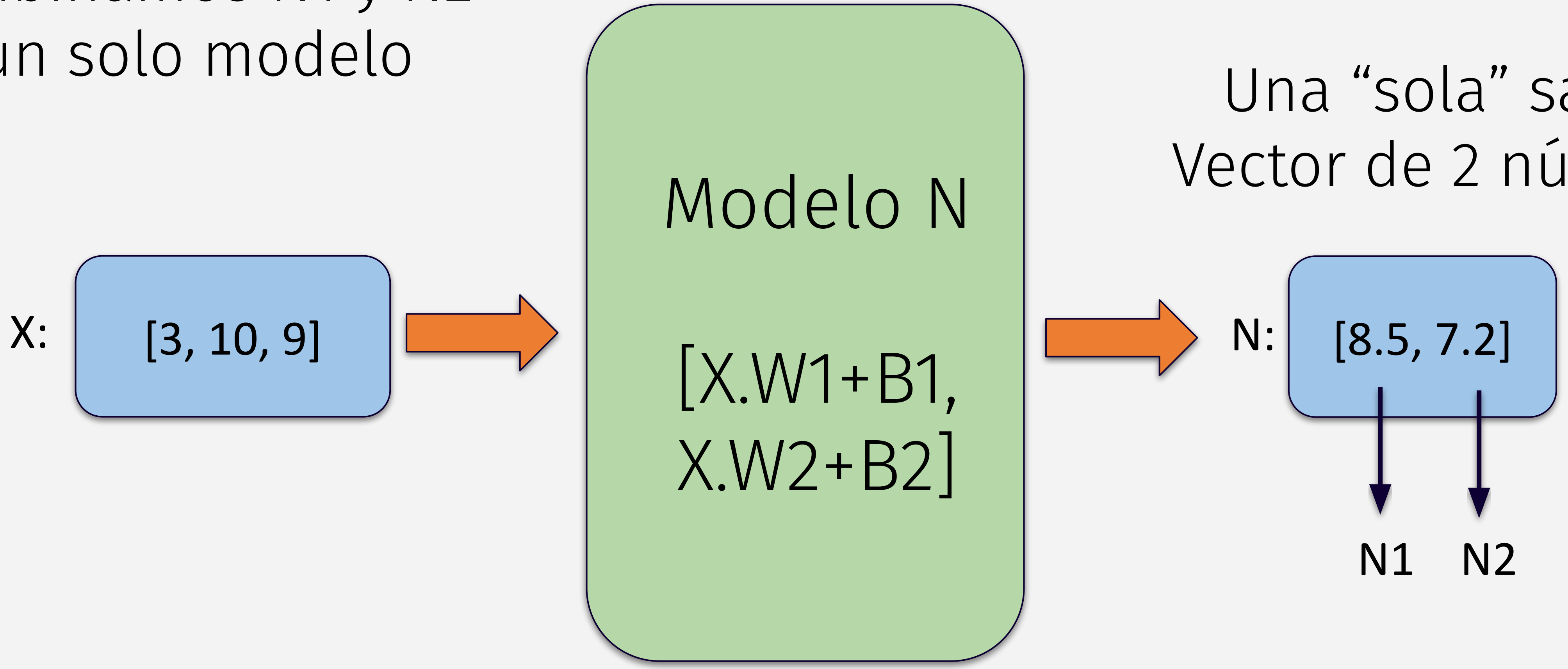


Si tenemos 100 clases?
Hay una forma más eficiente

Regresión Lineal Múltiple - Modelo

Combinamos N1 y N2 en un solo modelo

Una “sola” salida
Vector de 2 números



Regresión Lineal Múltiple - Modelo

Combinamos los parámetros en:

- Matriz W de 3x2
- Vector B de 2x1

X: [3, 10, 9]

Modelo N

X.W+B

donde

W= [W1,W2]

B=[B1,B2]

N: [8.5, 7.2]

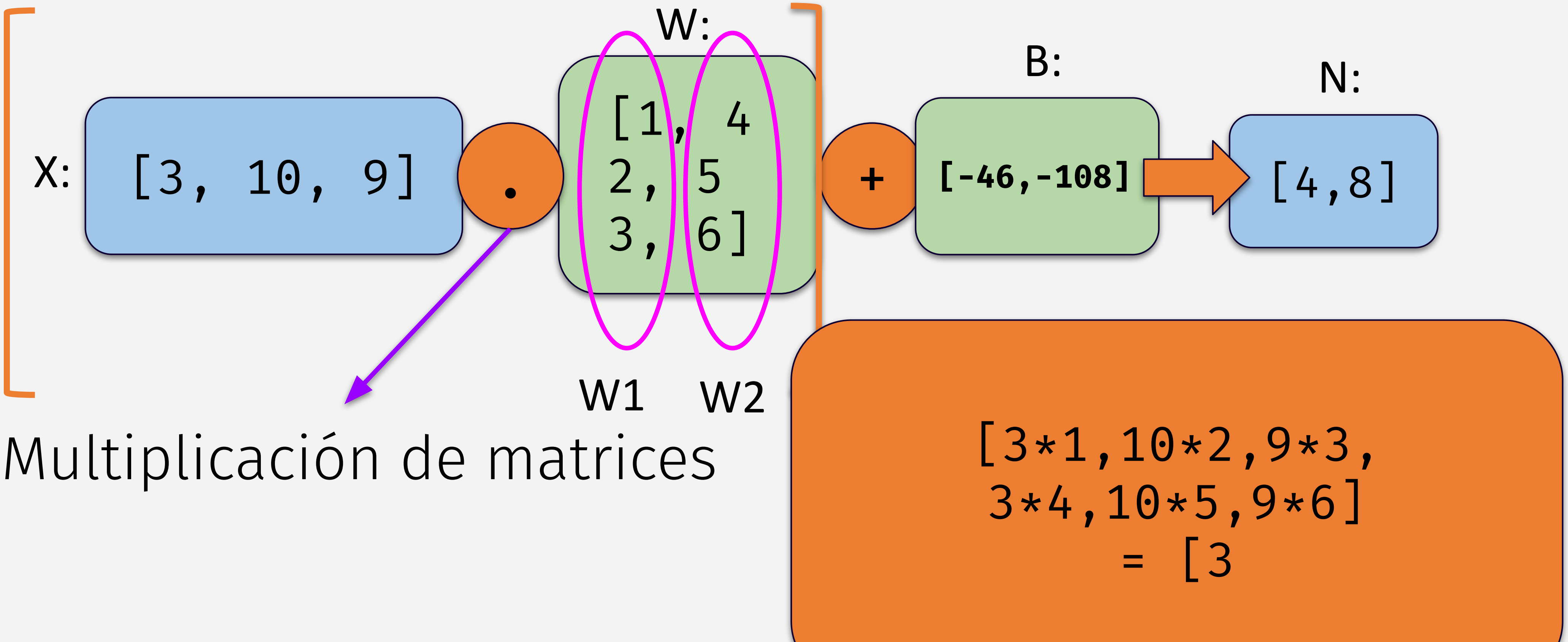
Producto punto con matrices

Suma vectorial

Regresión Lineal Múltiple - Cálculo

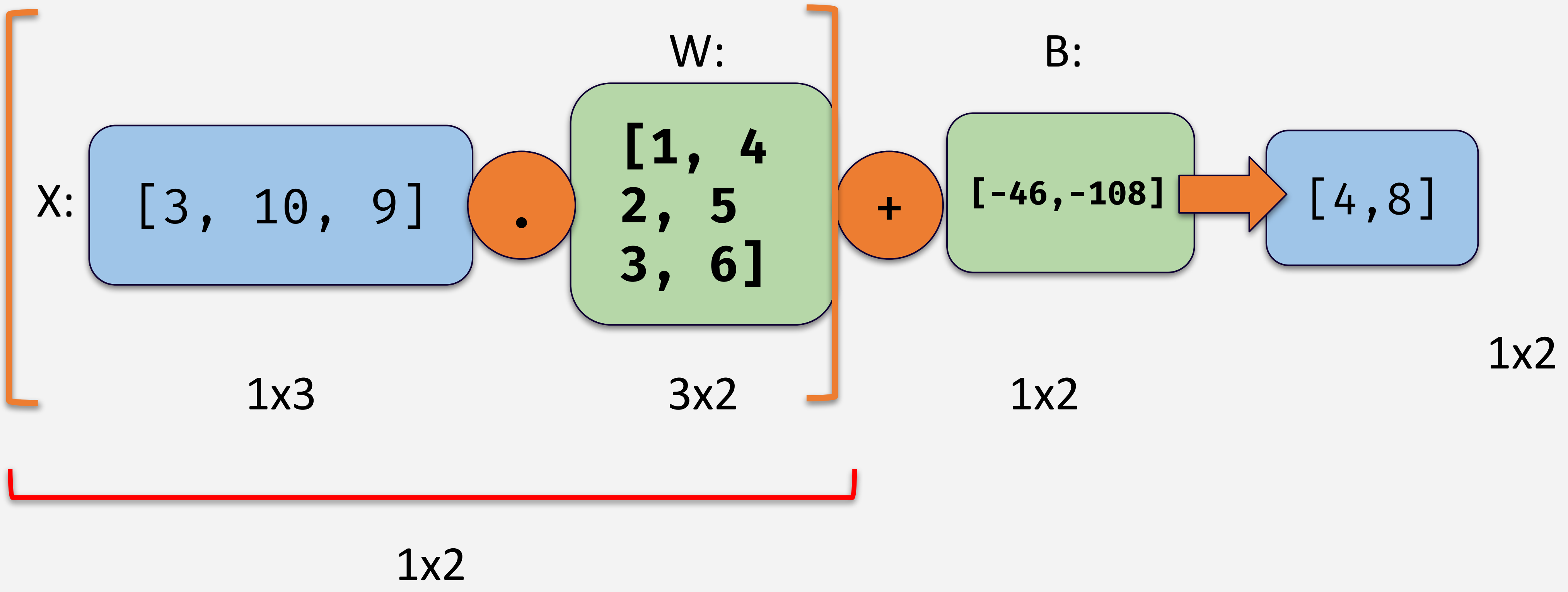
- W1 [1, 2, 3]’
- B1 -46

- W2 [4, 5, 6]’
- B2 -108



Regresión Lineal Múltiple - Tamaños

Tamaños de vectores y matrices

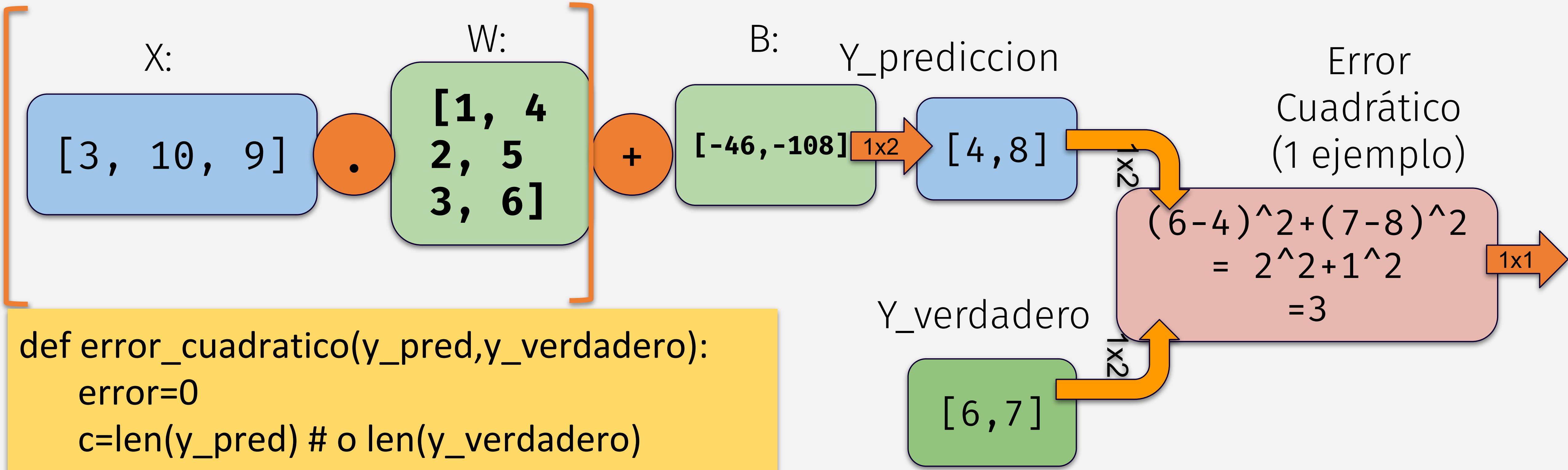


Recursos para repasar operaciones vectoriales

- Producto punto o escalar entre dos vectores: https://www.youtube.com/watch?v=N5f7pYTNcFM
- Multiplicación de un vector por un escalar: https://es.khanacademy.org/math/algebra-home/alg-vectors/alg-scalar-multiplication/v/understanding-multiplying-vectors-by-scalars
- Multiplicación de una matriz por un escalar: https://www.youtube.com/watch?v=-ArUqjhQIBM
- Producto de una matriz y un vector https://www.youtube.com/watch?v=2Gdy1xRnqjk
- Producto de una matriz y otra matriz https://www.youtube.com/watch?v=Tjrm3HsqBXE https://es.khanacademy.org/math/linear-algebra/matrix-transformations/composition-of-transformations/v/linear-algebra-matrix-product-examples

Regresión **Lineal** Múltiple - Función de Error $\mathbf{E_i}$

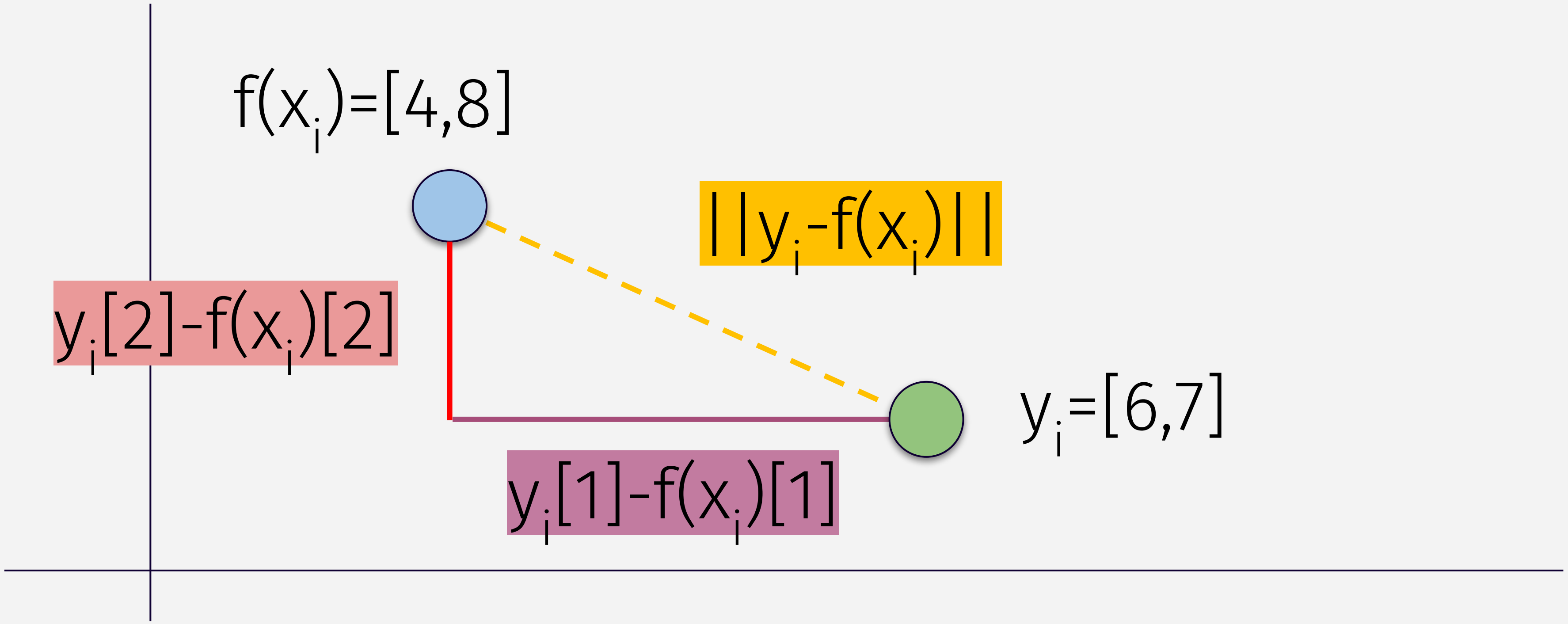
$$E_i(\mathbf{W,B}) = (y_i[1]-f(x_i)[1])^2 + (y_i[2]-f(x_i)[2])^2 = ||y_i-f(x_i)||^2$$




```
def error_cuadratico(y_pred,y_verdadero):
    error=0
    c=len(y_pred) # o len(y_verdadero)
    for i in range(c):
        error+= (y_pred[i]-y_verdadero[i])**2
    return error
```

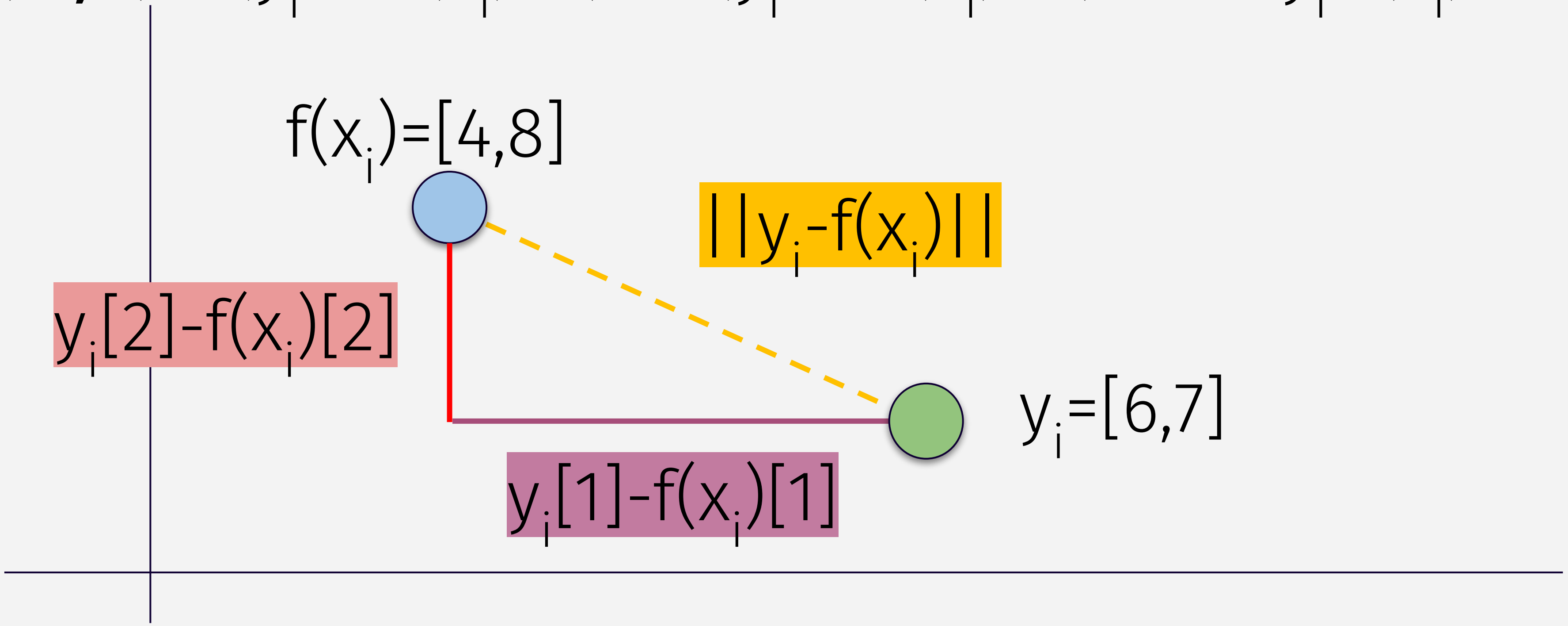
Regresión **Lineal** Múltiple - Función de Error $\mathbf{E_i}$

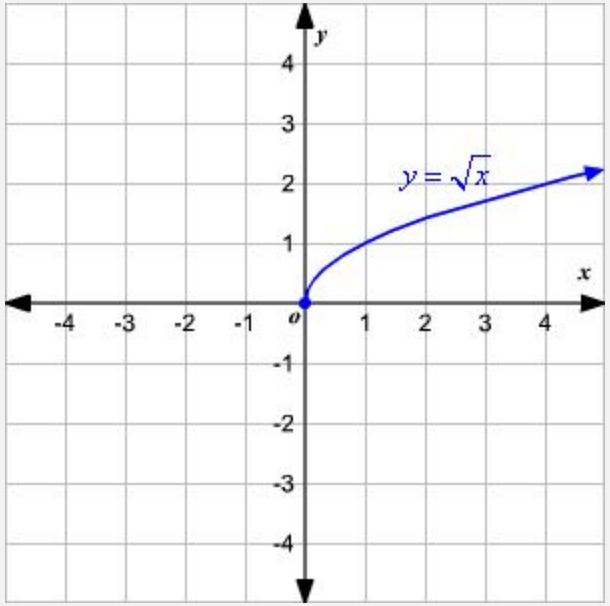
- $E_i(\mathbf{W},\mathbf{B}) = (y_i[1]-f(x_i)[1])^2 + (y_i[2]-f(x_i)[2])^2 = ||y_i-f(x_i)||^2$
  - Distancia euclídea al **cuadrado**
    - Entre $f(x_i)$ e y_i



Regresión **Lineal** Múltiple - Función de Error $\mathbf{E_i}$

$$E_i(\mathbf{W},\mathbf{B}) = (y_i[1]-f(x_i)[1])^2 + (y_i[2]-f(x_i)[2])^2 = ||y_i-f(x_i)||^2$$





- E_i = Distancia Euclídea al cuadrado = $||y_i-f(x_i)||^2$
- ¿Por qué no $||y_i-f(x_i)|| = \mathrm{sqrt}[\ (y_i[1]-f(x_i)[1])^2 + (y_i[2]-f(x_i)[2])^2\ ]$?
  - sqrt es función monótona creciente
    - Misma solución

Regresión **Lineal** Múltiple - Función de Error **E**

X = (Estudio, Edad, Promedio)
Y_verdadero = (N1,N2)
Y_prediccion = (P1,P2)

$$E(\mathbf{W},\mathbf{B}) = 1/n\ \Sigma_i^n\ E_i(\mathbf{W},\mathbf{B})$$

Estudio	Edad	Promedio	N1	N2	P1	P2	E
2	24	4	7	7.2	16	44	1435.24
5	22	3	4.5	5.2	12	40	1267.29
7	25	4	6.3	6	23	69	4247.89
9	20	7	5.4	4.4	24	70	4649.32
10	19	4	8.2	9.5	14	51	1755.89
11	20	3	7.2	8.1	14	54	2153.05
13.4	21	5	5.5	10	24.4	80.6	5341.57
14	20	3	3	4.3	17	66	4002.89

Promedio(E) = 3106.64

```
def error_cuadratico_medio(x,y,w,b):
    n = x.shape[0]  # o y.shape[0]
    error=0
    for i in range(n):
        P = x[i,:]*w+b #prediccion
        N = y[i,:]     #verdadero
        error += error_cuadratico(P,N)
    return error/n
```

```
def error_cuadratico(y_pred,y_verdadero):
    error=0
    c=len(y_pred) # o len(y_verdadero)
    for i in range(c):
        error+=(y_pred[i]-y_verdadero[i])**2
    return error
```

Regresión Lineal Múltiple - Resumen

Resumen

- Predecir N valores
- Transformar un vector en otro
  - Vector de entrada **x** de dimension **N**
  - Vector de salida **y** de dimension **M**
  - Matriz de pesos **W**, de dimension **NxM**
  - Vector de bias **B**, de dimensión **M**
- Descenso de gradiente
  - Sumar errores cuadráticos de cada salida.
  - Error sigue siendo convexo (solución única).
- ¿Es más potente que Regresión Lineal Simple?
  - No, sigue siendo un modelo lineal.

Estudio	Edad	Promedio	N1	N2
2	24	4	7	7.2
5	22	3	4.5	5.2
7	25	4	6.3	6
9	20	7	5.4	4.4
10	19	4	8.2	9.5
11	20	3	7.2	8.1
13.4	21	5	5.5	10
14	20	3	3	4.3