Phaw AI

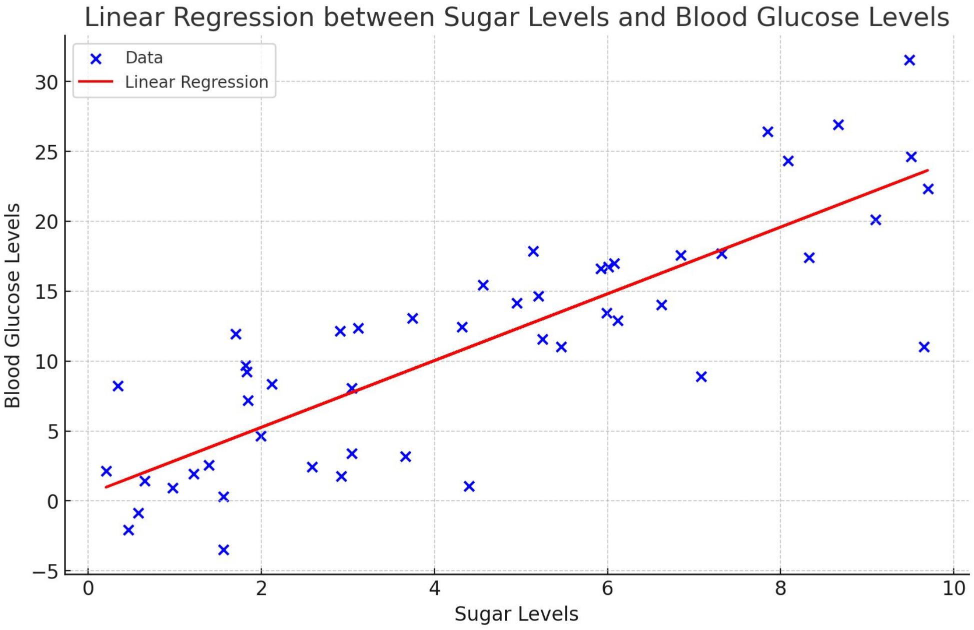
Dr. Cristian López Del Alamo
clopezd@utec.edu.pe
ULASALLE

Phaw AI

Dr. Cristian López Del Alamo
clopezd@utec.edu.pe
ULASALLE

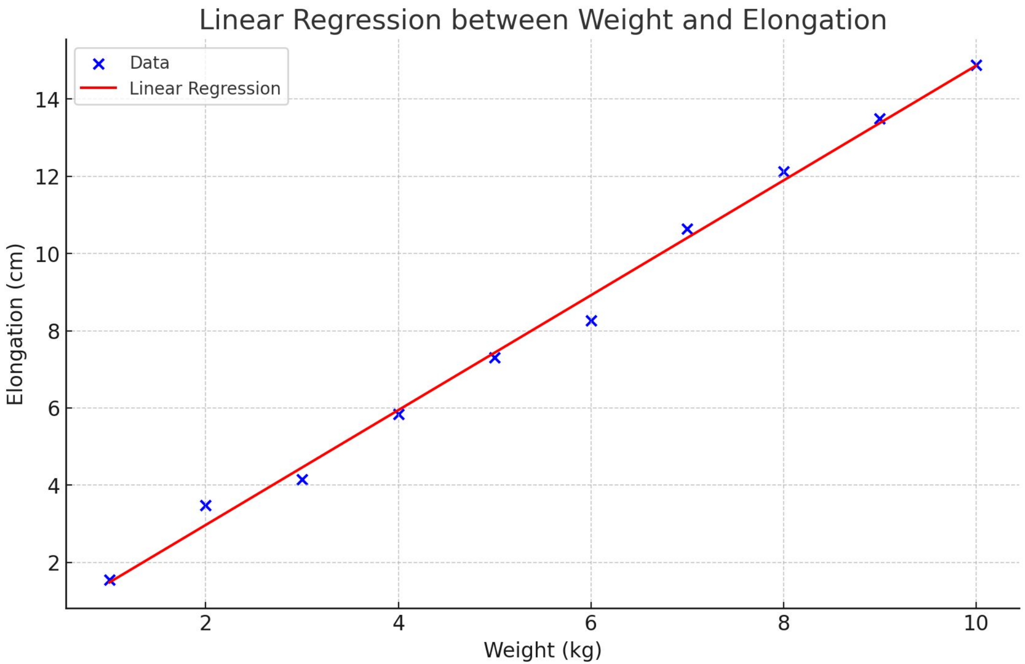
Objective: Understand the idea behind univariate linear regression

Prediction of blood sugar levels



Sugar Levels	Blood Glucose Levels
3.75	13.06
9.51	24.62
7.32	17.72
5.99	13.46
1.56	-3.49

Prediction of the elongation of a spring of static size



Weight (kg)	Elongation (cm)
1	1.55
2	3.48
3	4.15
4	5.84
5	7.30
6	8.27
7	10.65
8	12.13
9	13.50
10	14.88

Model

$$L = xw + b$$

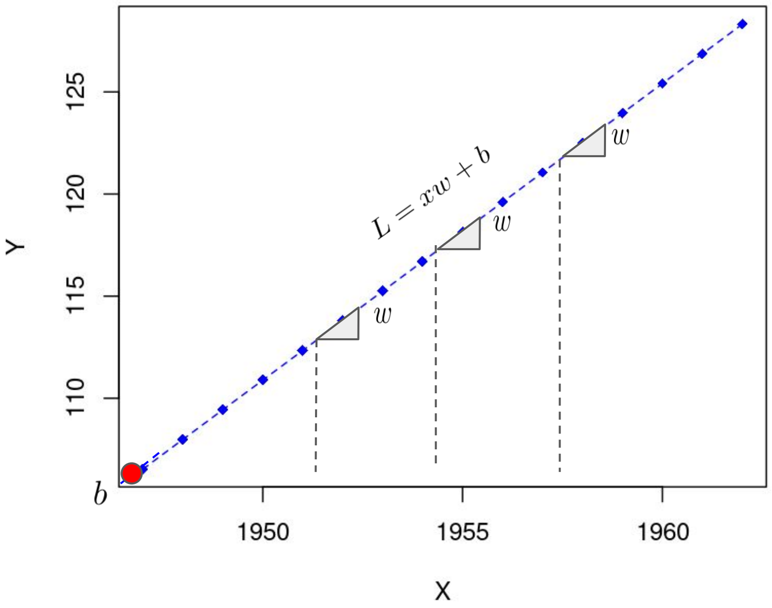
Parameters

Model

$$L = xw + b$$

bias

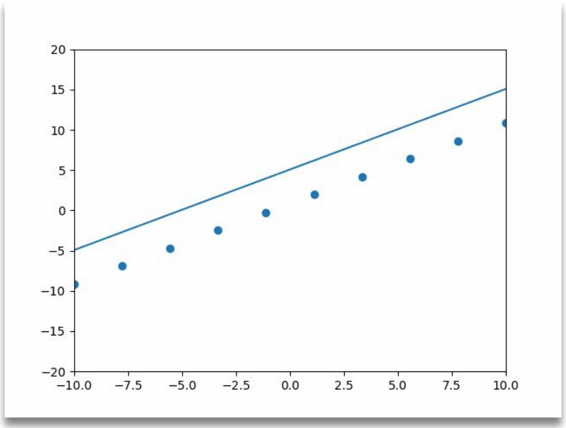
Slope

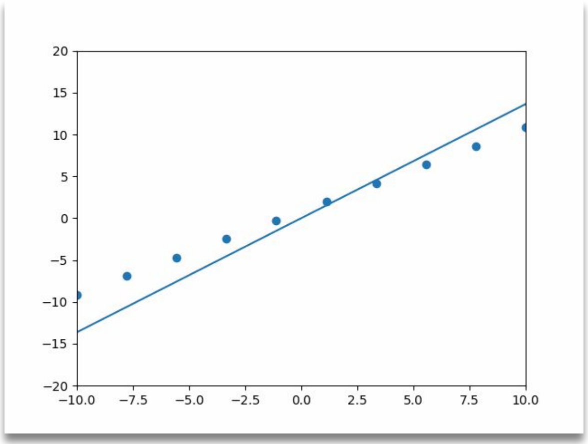


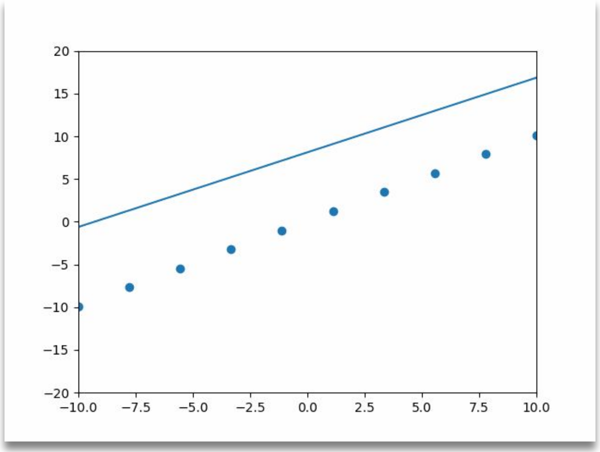
$$L = xw + \boxed{b}$$

$$L = x\boxed{w} + b$$

$$L = x\boxed{w} + \boxed{b}$$

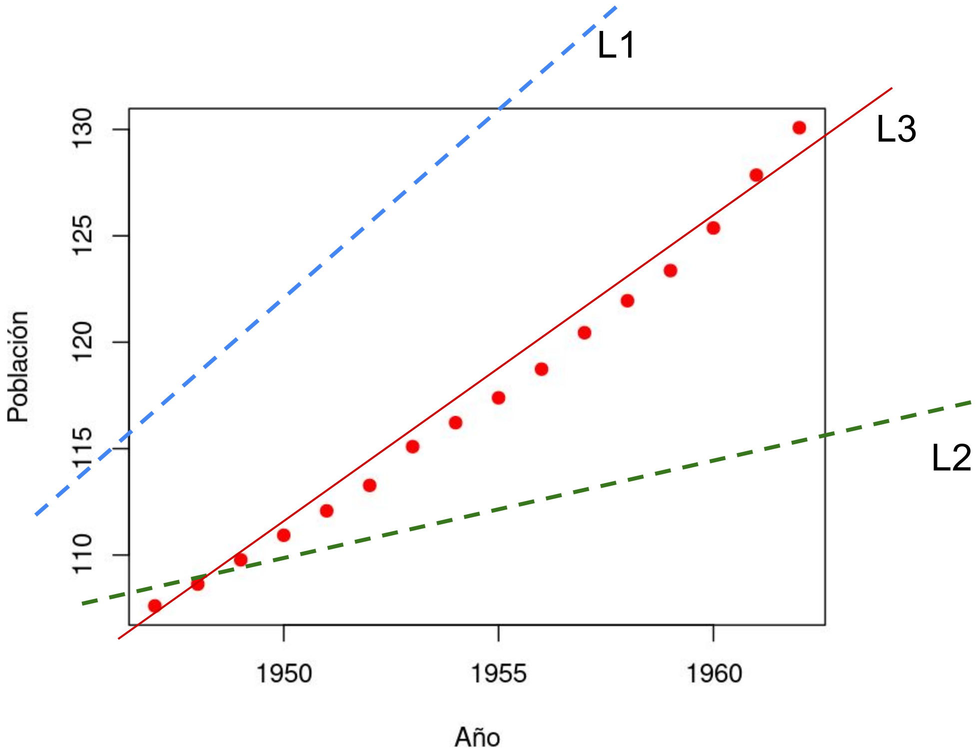




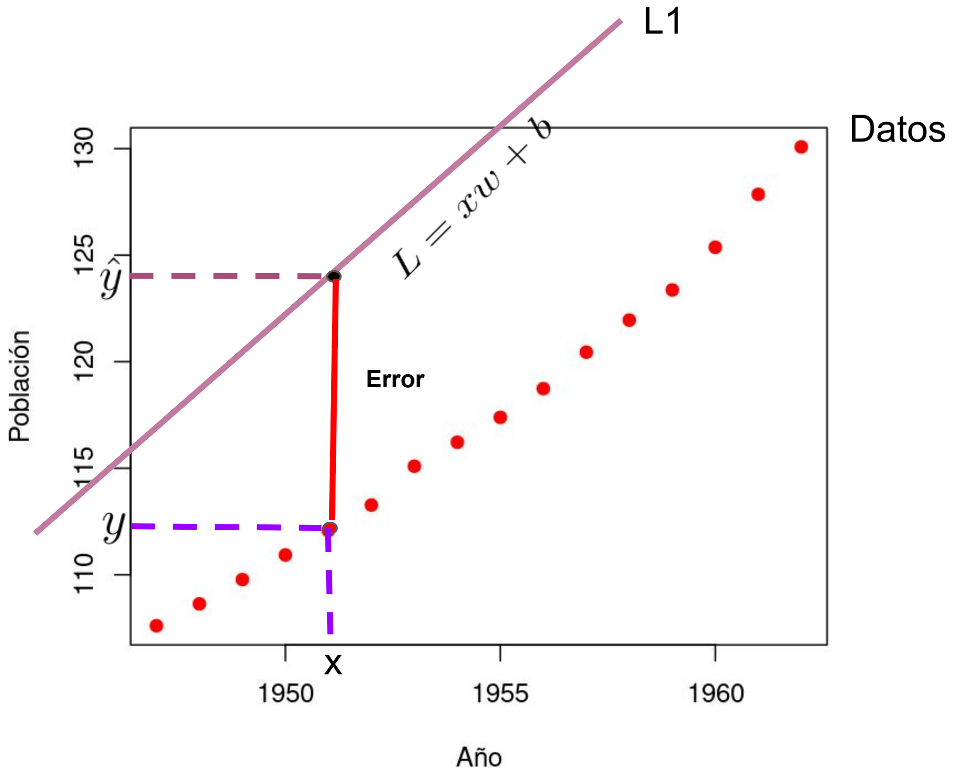


UTEC UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

How do I find the correct values of w and b?



How do I find the correct values of w and b?



How do I find the correct values of w and b?

Hipótesis

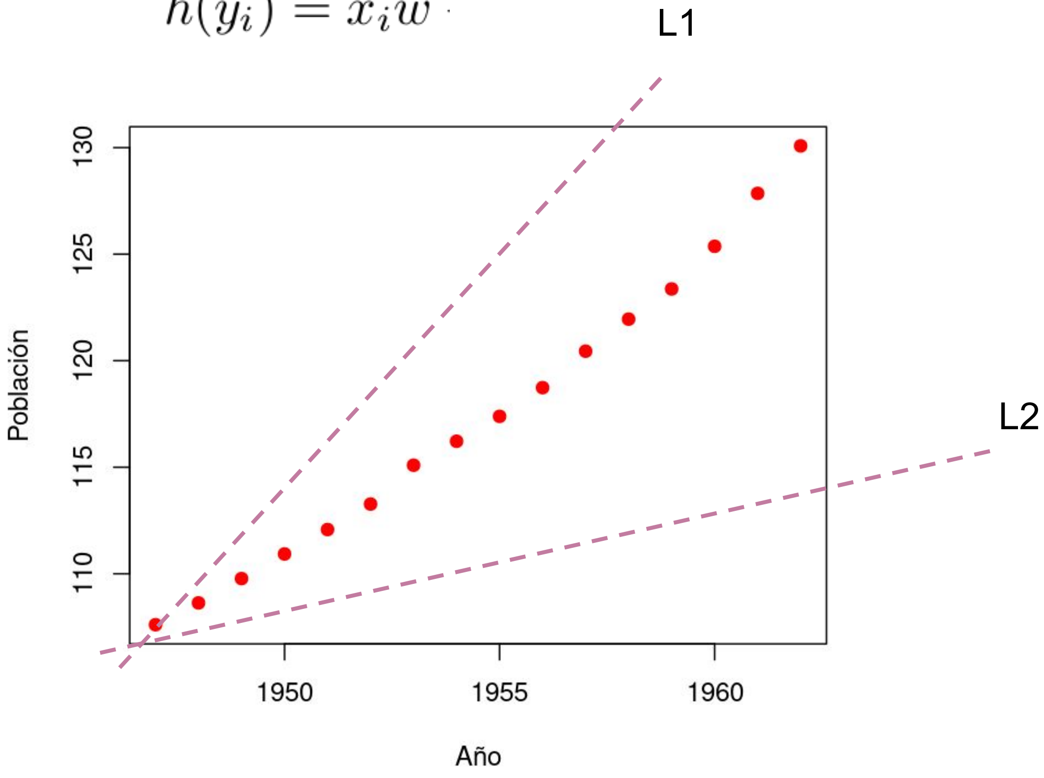
$$h(y_i) = x_i w + b$$

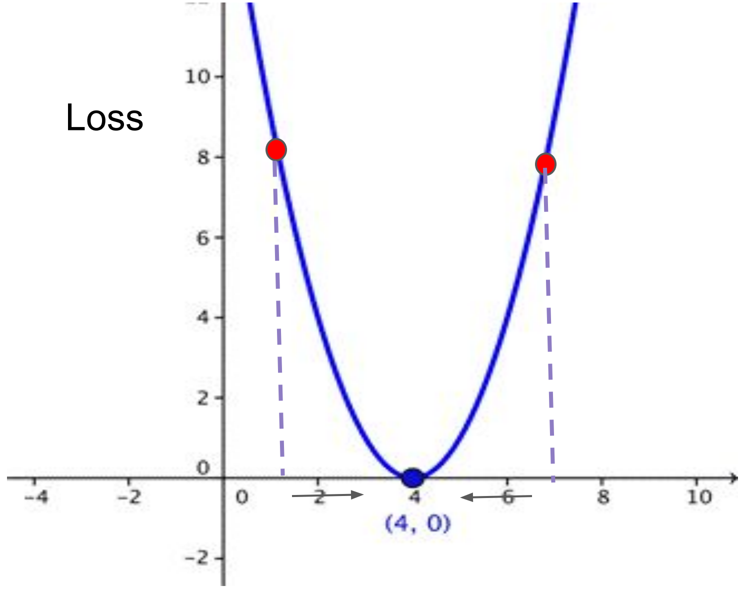
Loss Function

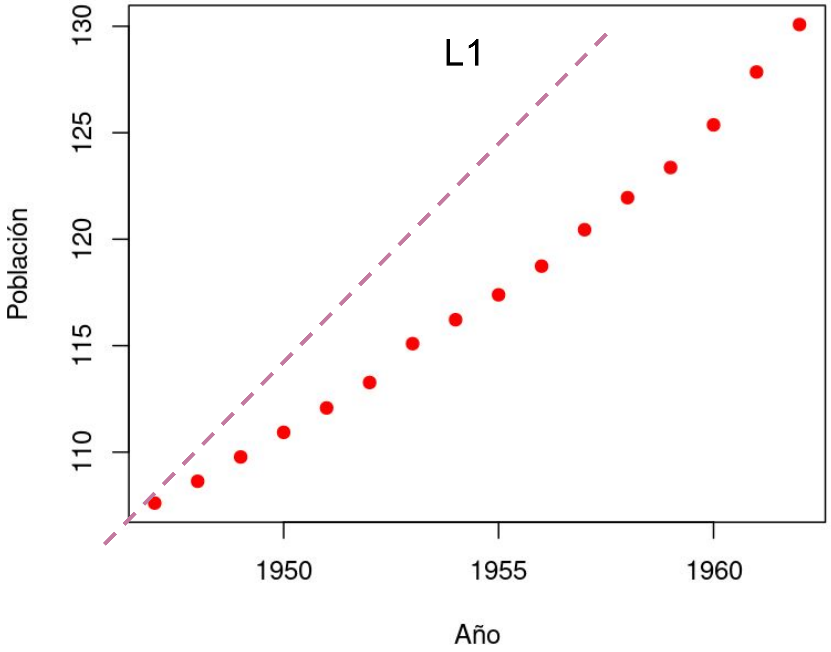
$$\mathcal{L} = \frac{1}{2n} \sum_{i=0}^{n} (y_i - \hat{y}_i)^2$$

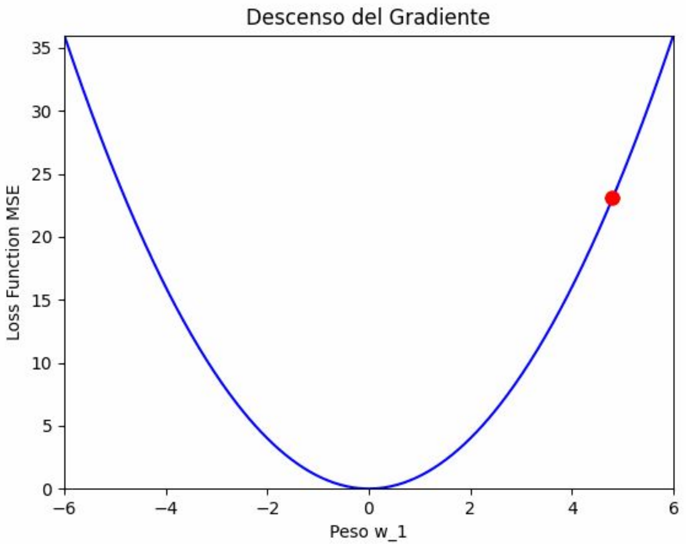
How do I find the correct values of w and b?

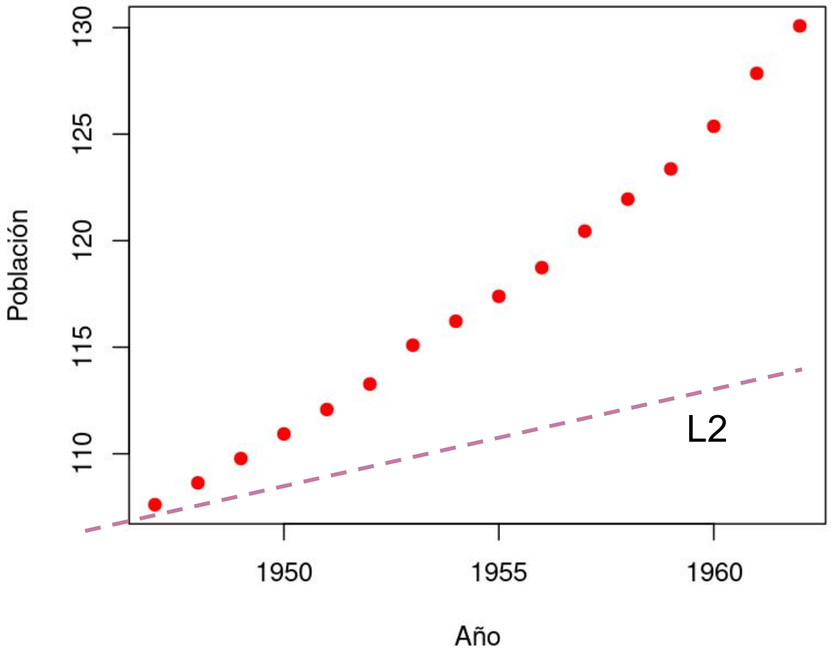
$$h(y_i) = x_i w$$

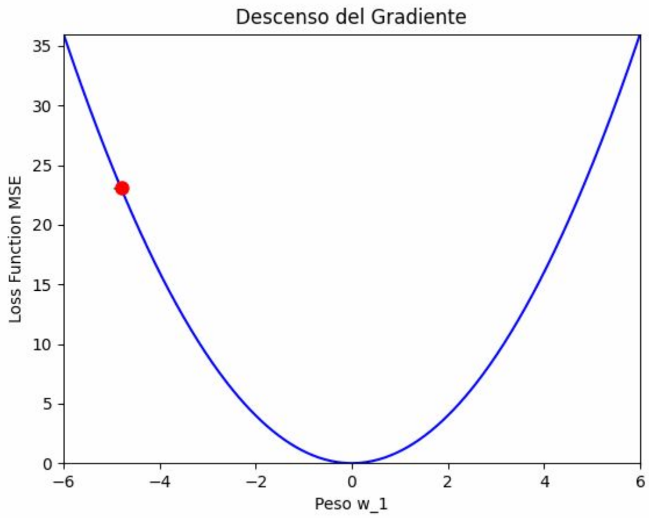


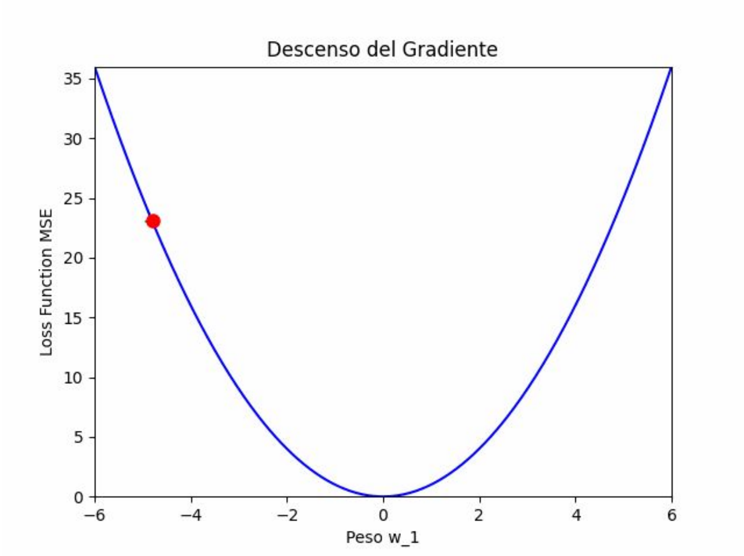



L1
130
125
120
115
110
Población
1950
1955
1960
Año

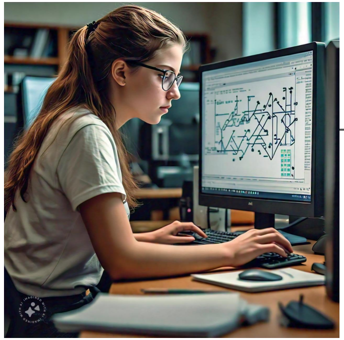

Descenso del Gradiente
35
30
25
20
15
10
5
0
Loss Function MSE
−6
−4
−2
0
2
4
6
Peso w_1


Población
Año
130
125
120
115
110
1950
1955
1960
L2


Descenso del Gradiente
Loss Function MSE
Peso w_1
35
30
25
20
15
10
5
0
−6
−4
−2
0
2
4
6



$$w = w - \alpha \frac{\partial Loss}{\partial w}$$

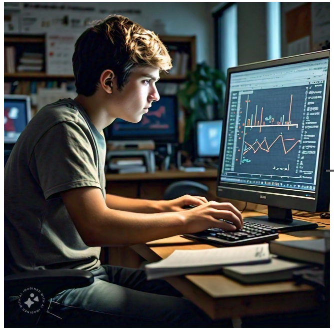


Training time

$$model = x_i w + b$$

$$loss = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{2n}$$

$$w = w - \alpha \frac{\partial Loss}{\partial w}$$

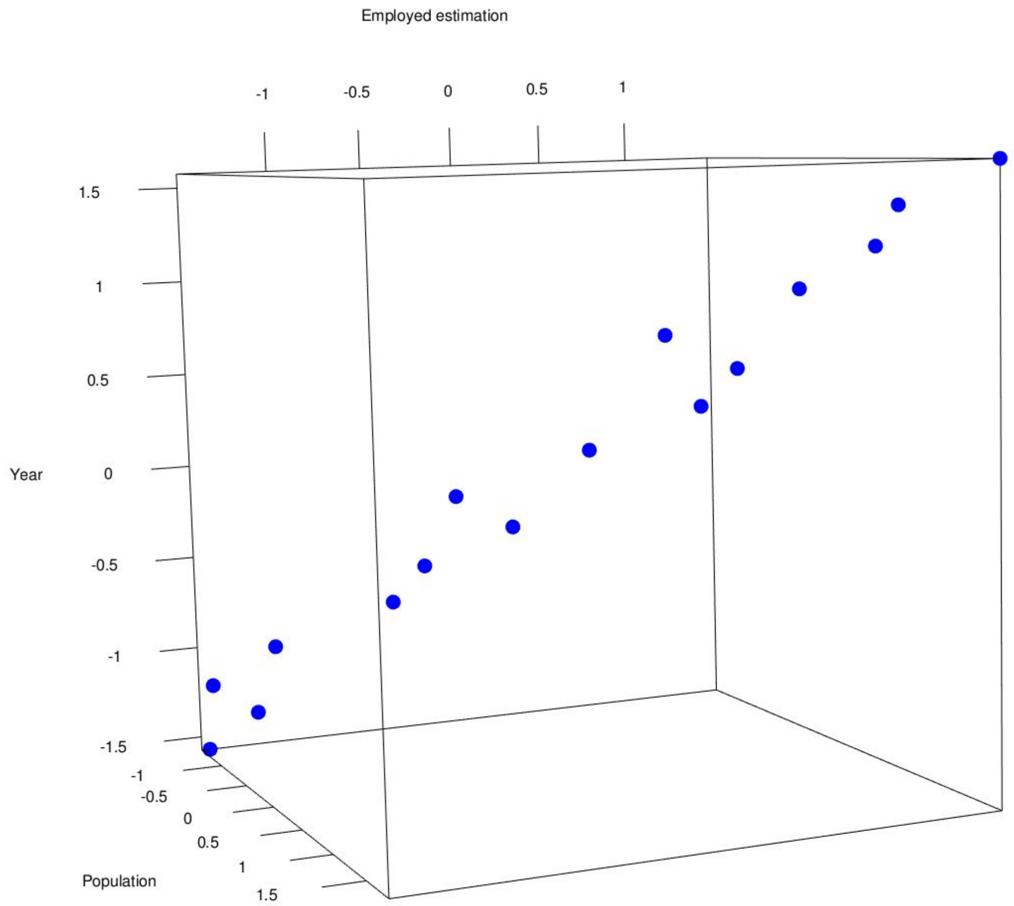


Inference time

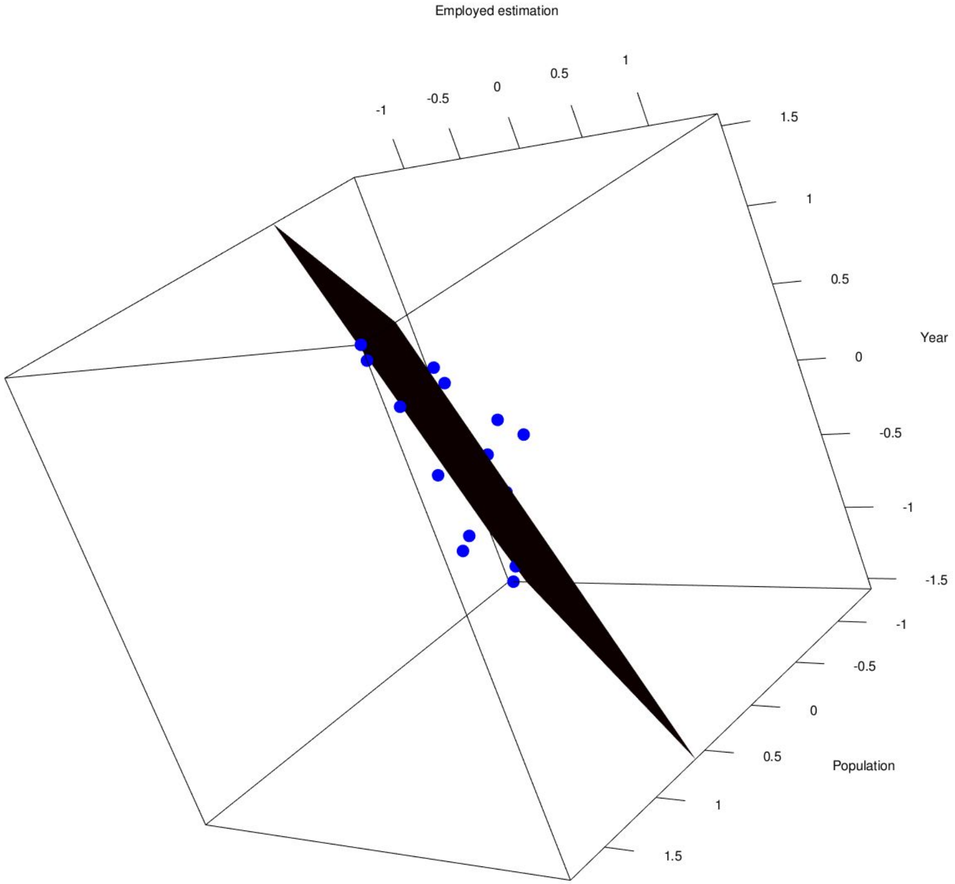
$$model = x_i w^* + b^*$$

Objective: Understand the idea behind multivariable linear regression

How do we approach this set of points to **predict employability** based on **population** and **year**?

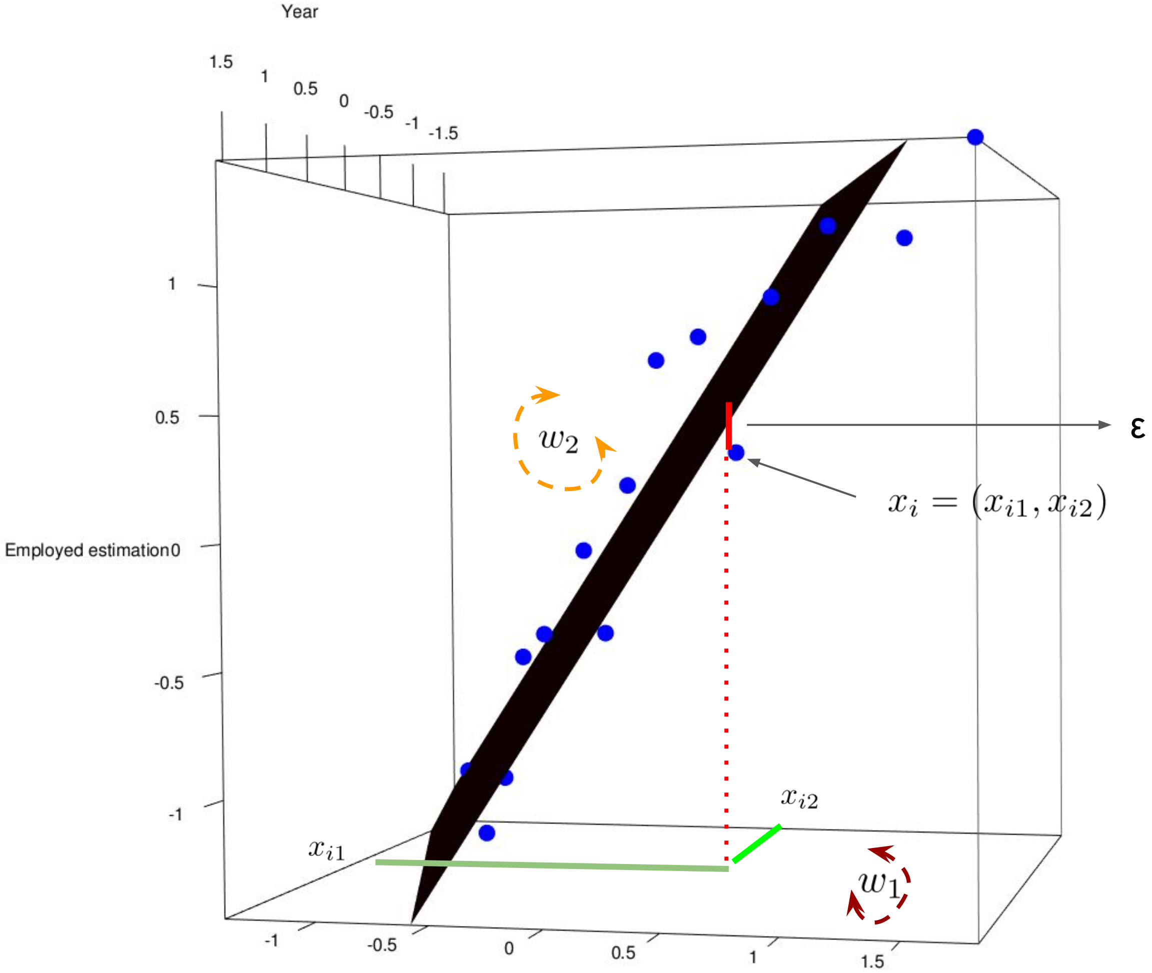


How can we **create** the **plane** that **best fits** this **set of points**?



Plane Equation

$$ax_1 + bx_2 + c = 0 \longrightarrow x_1 w_1 + x_2 w_2 + b = 0$$


Year
1.5
1
0.5
0
-0.5
-1
-1.5
1
0.5
Employed estimation0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
w_2
ε
$x_i = (x_{i1}, x_{i2})$
x_{i2}
x_{i1}
w_1

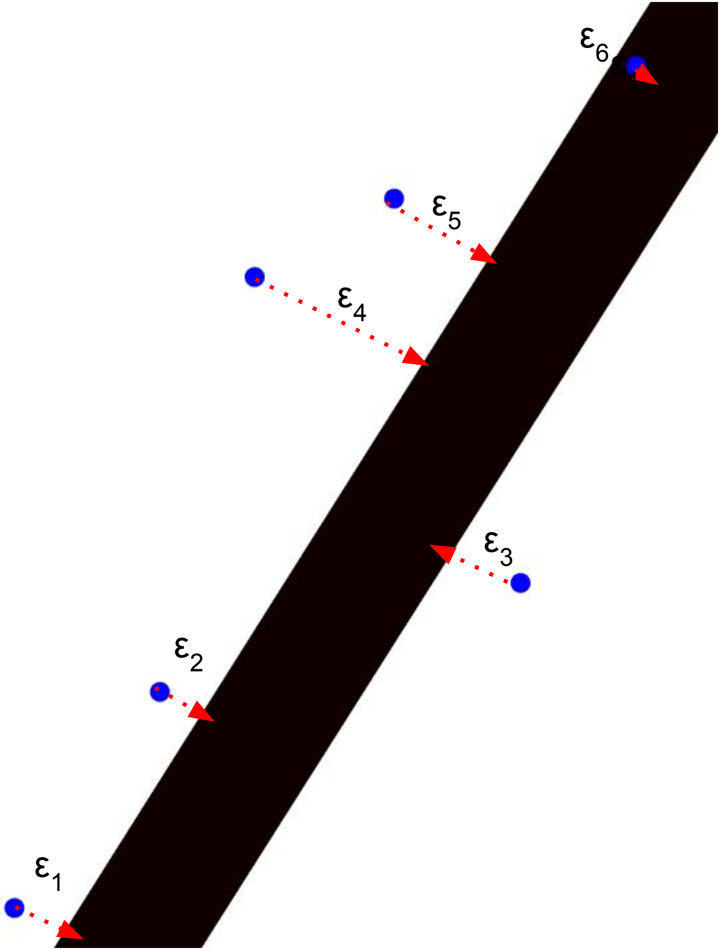
Hypothesis for **univariate** linear regression

$$h(x_i) = x_i w + b$$

Hypothesis for **multivariate** linear regression

$$h(x_i) = x_{i1} w_1 + x_{i2} w_2 + b$$

$$h(x_i) = x_{i1}w_1 + x_{i2}w_2 + b$$



$$\mathcal{L} = \frac{\varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_6}{6}$$

$$\mathcal{L} = \frac{\sum_{i=1}^{n} \varepsilon_i}{n}$$

What would be missing?

Hypothesis

$$h(x_i) = x_{i1}w_1 + x_{i2}w_2 + b$$

Loss Function

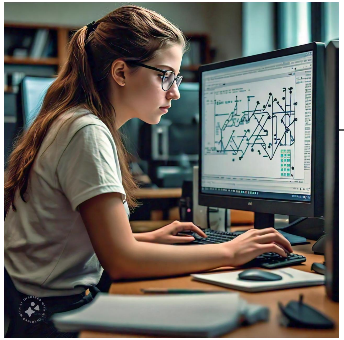
$$\mathcal{L} = \frac{\sum_{i=1}^{n}(y_i - h(x_i))^2}{n}$$

Derivatives

$$\frac{\partial Loss}{\partial b} \qquad \frac{\partial Loss}{\partial w_1} \qquad \frac{\partial Loss}{\partial w_2}$$

Change parameters

$$w_i = w_i - \alpha\frac{\partial loss}{\partial w_i}$$

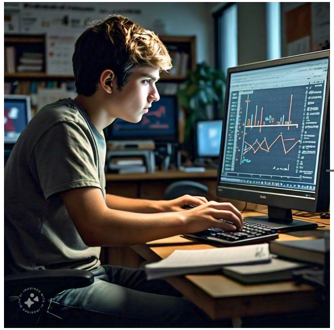


Training time

$$model = x_{i1}w_1 + x_{i1}w_2 + b$$

$$loss = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{2n}$$

$$w = w - \alpha\frac{\partial Loss}{\partial w}$$

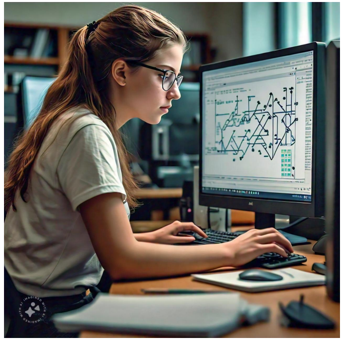


Inference time

$$model = x_{i1}w_1^* + x_{i1}w_2 + b^*$$

Video

Generalizing to a k-dimensional space



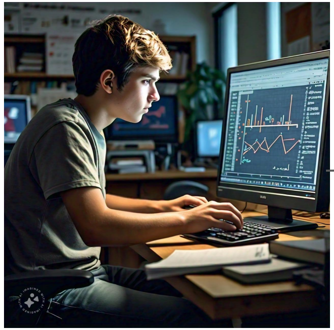
Training time

$$model = x_{i1}w_1 + x_{i1}w_2 + ... + x_{ik}w_k + b$$

$$loss = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{2n}$$

$$w = w - \alpha\frac{\partial Loss}{\partial w}$$

$$\frac{\partial Loss}{\partial w_i} = \frac{1}{n}\sum_{i=0}^{n}(y_i - h(x_i))^2$$



Inference time

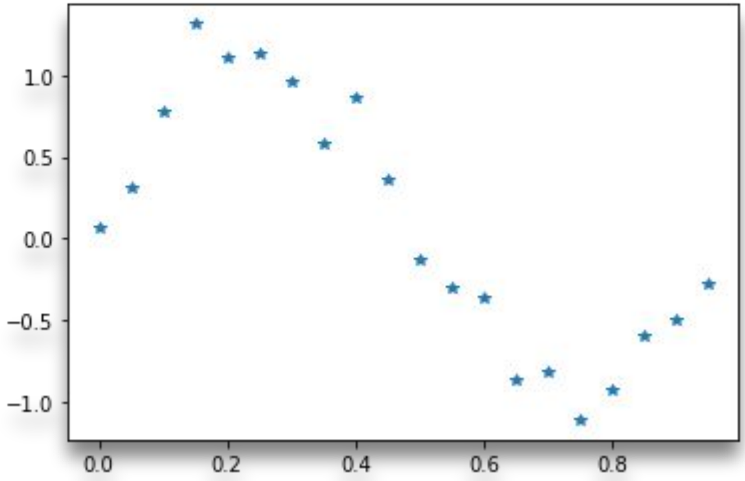
$$model = x_{i1}w_1 + x_{i1}w_2 + ... + x_{ik}w_k + b$$

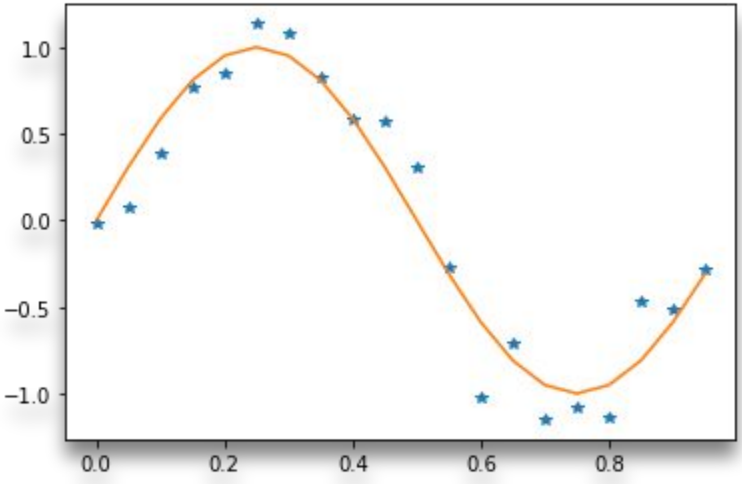
```python
def train(x, y, umbral, alfa):
    w = [np.random.rand() for i in range(1:k)]
    b = np.random.rand()
    L = Error(x, y, w, b)
    loss = []
    while (L > umbral):
        db, dw = derivada(x, y, w, b)
        b, w = update(w, b, alfa, db, dw)
        L = Error(x, y, w, b)
        print(L)
        loss.append(L)
    return b, w

```

Linear Regression

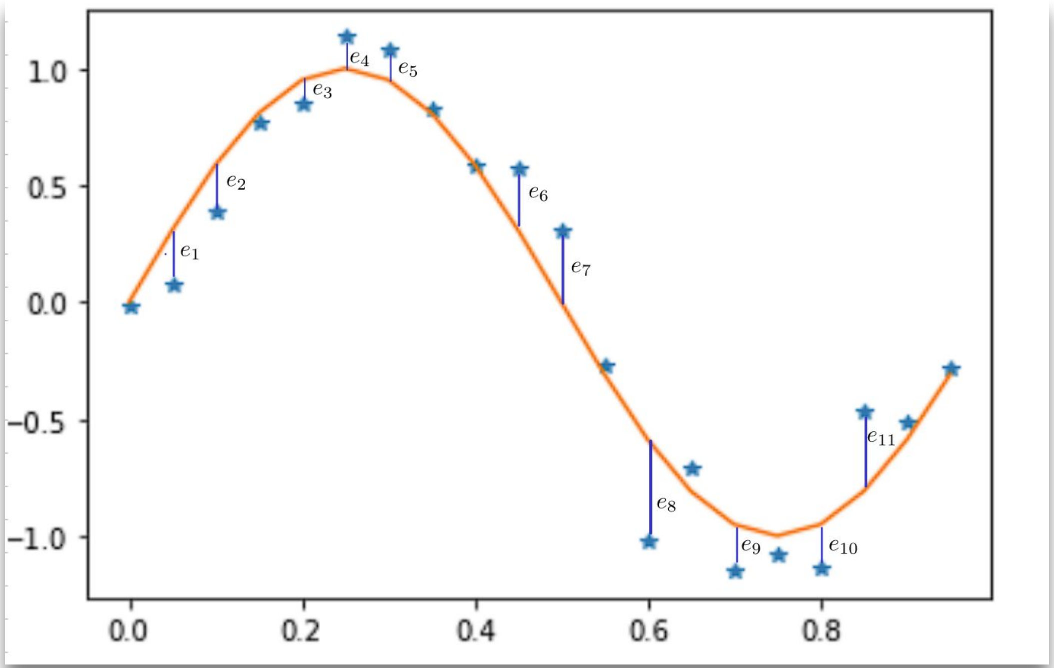
Objective: Understand the idea behind nonlinear


1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8


1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8

Loss Function

Loss



Epochs

Loss Function

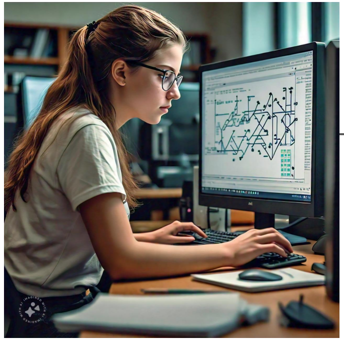
$$\mathcal{L} = \frac{\sum_{i=0}^{n}(y_i - h(x_i))^2}{2n}$$

What would be missing?

Hypothesis : $h(x_i) = b + x_i w_1 + x_i^2 w_2 + x_i^3 w_2 + \ldots + + x_i^p w_2$

Loss Function : $\mathcal{L} = \frac{\sum_{i=1}^{n}(y_i - h(x_i))^2}{n}$

Change parameters → Find derivatives of the error with respect to the parameters



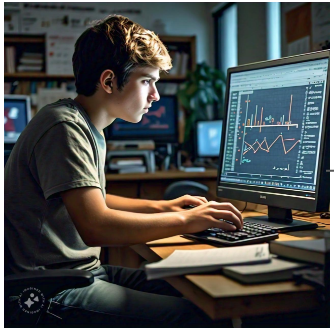
Training time

$$modelo = x_i w_1 + x_i^2 w_2 + \ldots + x_i^k w_k + b$$

$$loss = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{2n}$$

$$w = w - \alpha \frac{\partial Loss}{\partial w}$$

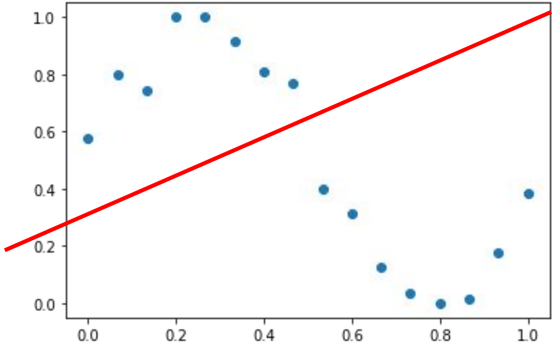
$$\frac{\partial Loss}{\partial w_i} = \frac{1}{n} \sum_{i=0}^{n} (y_i - h(x_i))^2$$



Inference time

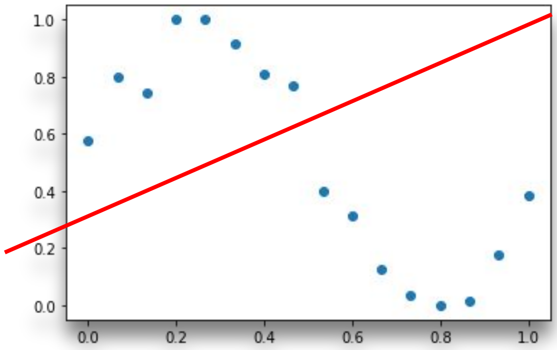
$$model = x_{i1} w_1 + x_{i1} w_2 + \ldots + x_{ik} w_k + b$$

Underfitting



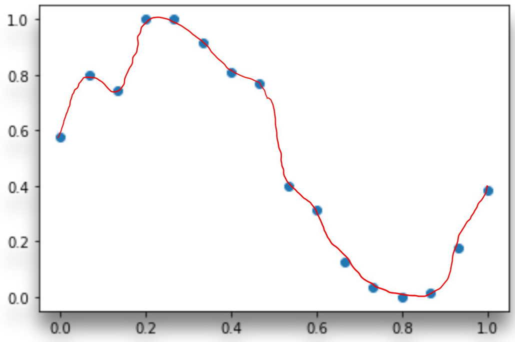
$$h(x_i) = x_i w + b$$

Underfitting



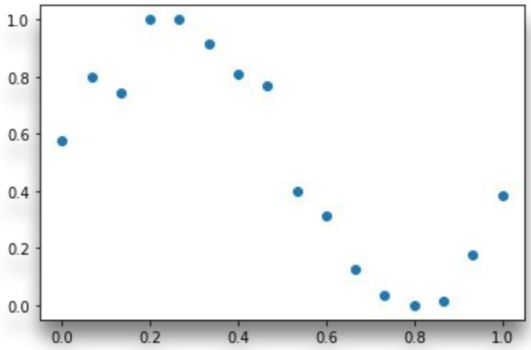
$$h(x_i) = x_i w + b$$

Overfitting



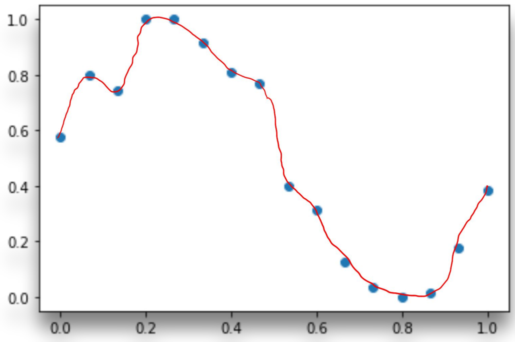
$$h(x_i) = x_i^0 w_0 + x_1^1 w_1 + \ldots + x_i^{20} w_{20}$$

Underfitting



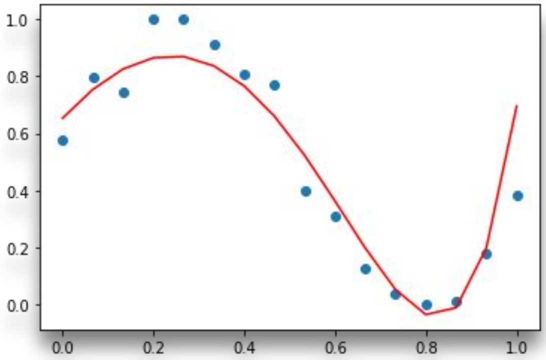
$$h(x_i) = x_i w + b$$

Overfitting



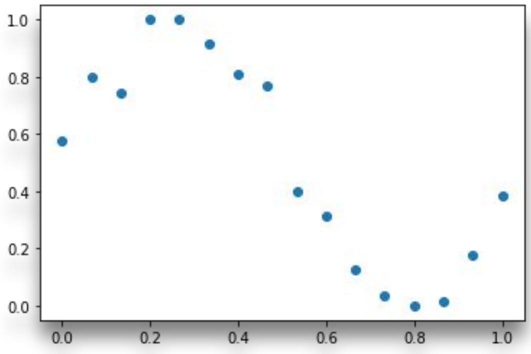
$$h(x_i) = x_i^0 w_0 + x_1^1 w_1 + \ldots + x_i^{20} w_{20}$$

good



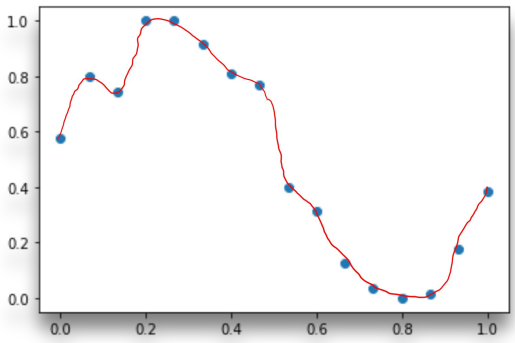
$$h(x_i) = x_i^0 w_0 + x_1^1 w_1 + \ldots + x_i^3 w_3$$

Underfitting	Overfitting	good



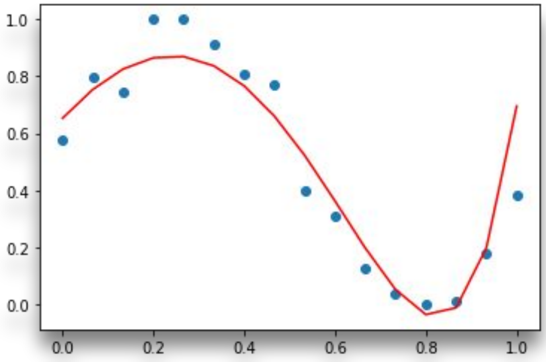
$$h(x_i) = x_i w + b$$

- Simple Model
- Low Capacity Model



$$h(x_i) = x_i^0 w_0 + x_1^1 w_1 + \ldots + x_i^{20} w_{20}$$

- Highly Complex Model
- High Capacity Model



$$h(x_i) = x_i^0 w_0 + x_1^1 w_1 + \ldots + x_i^3 w_3$$

- Data-Fitting Model
- Model with Adequate Capacity

[Linear No Regression]