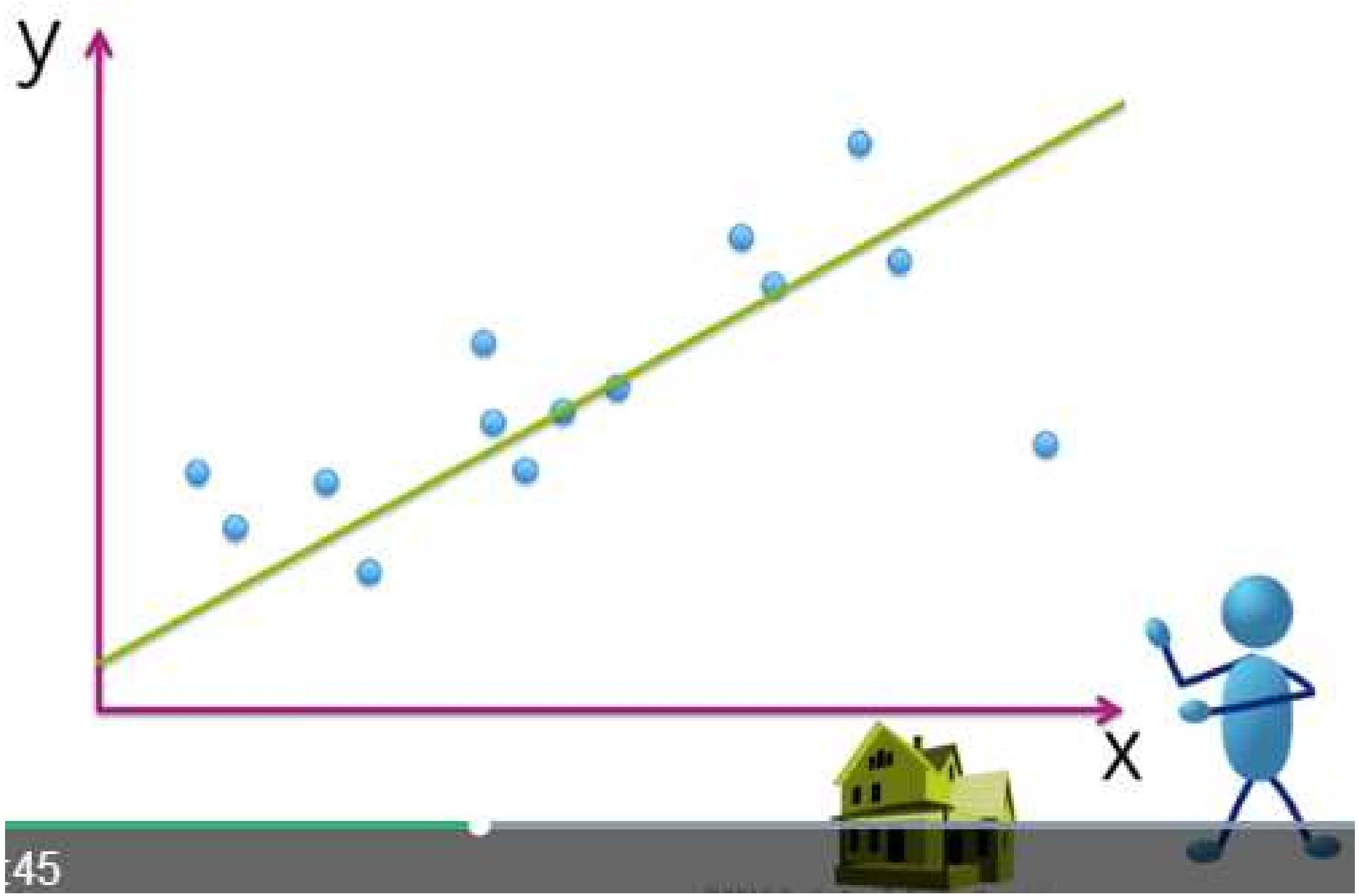
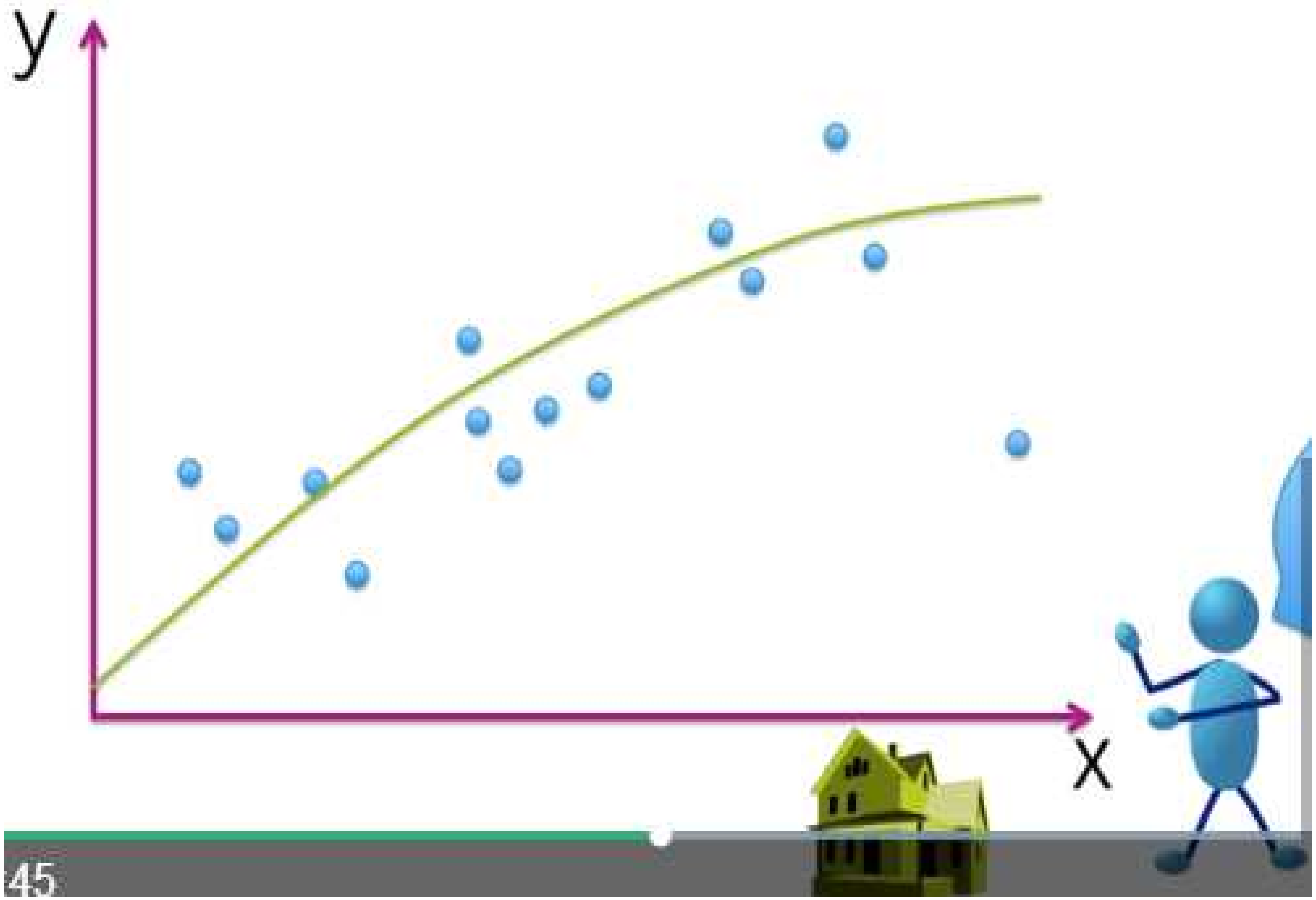
Chapitre II : Analyse Prédictive :

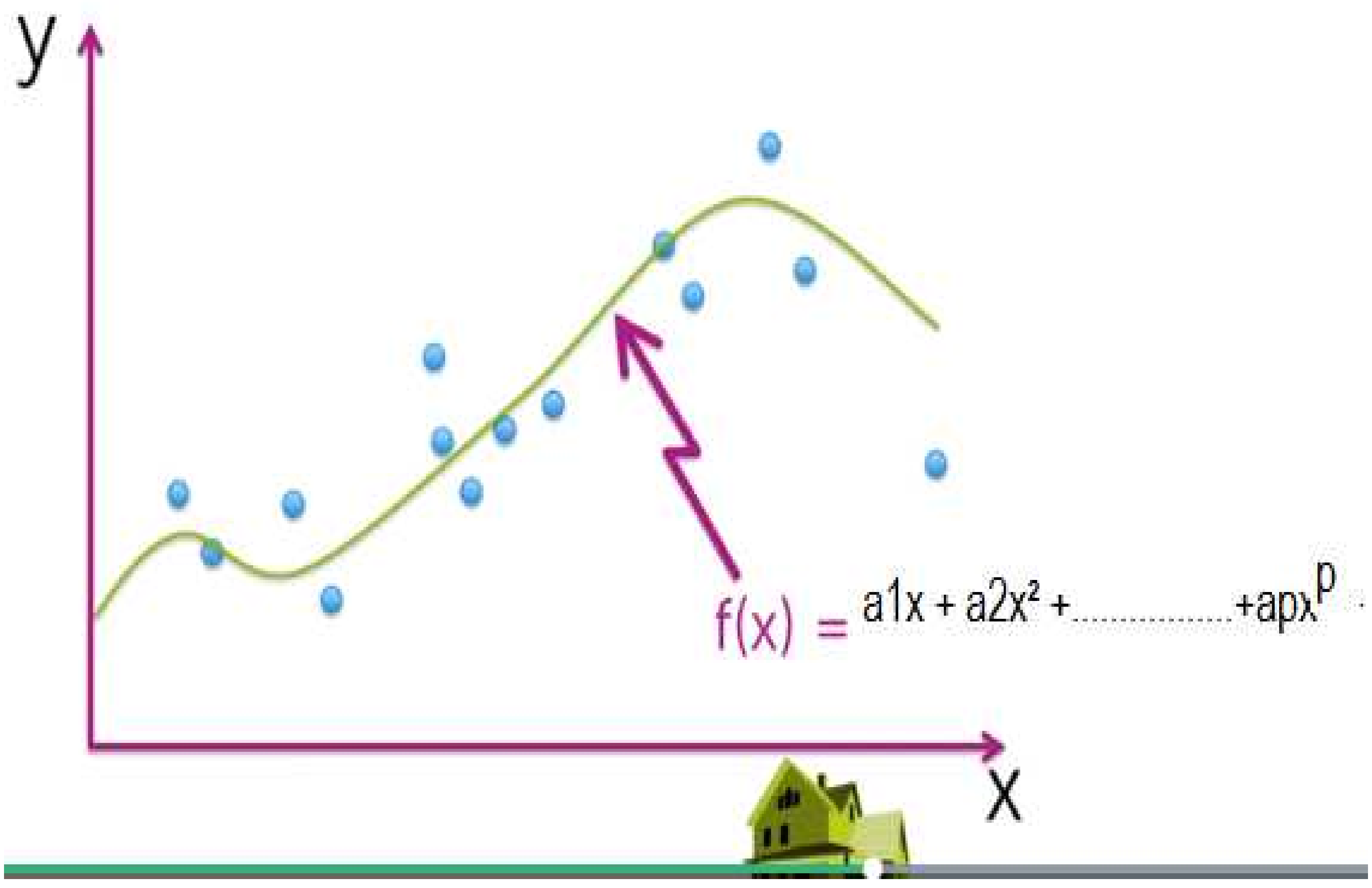
II- 2- Régression Non Linéaire: Régression Polynomiale



$$Y = ax + b$$

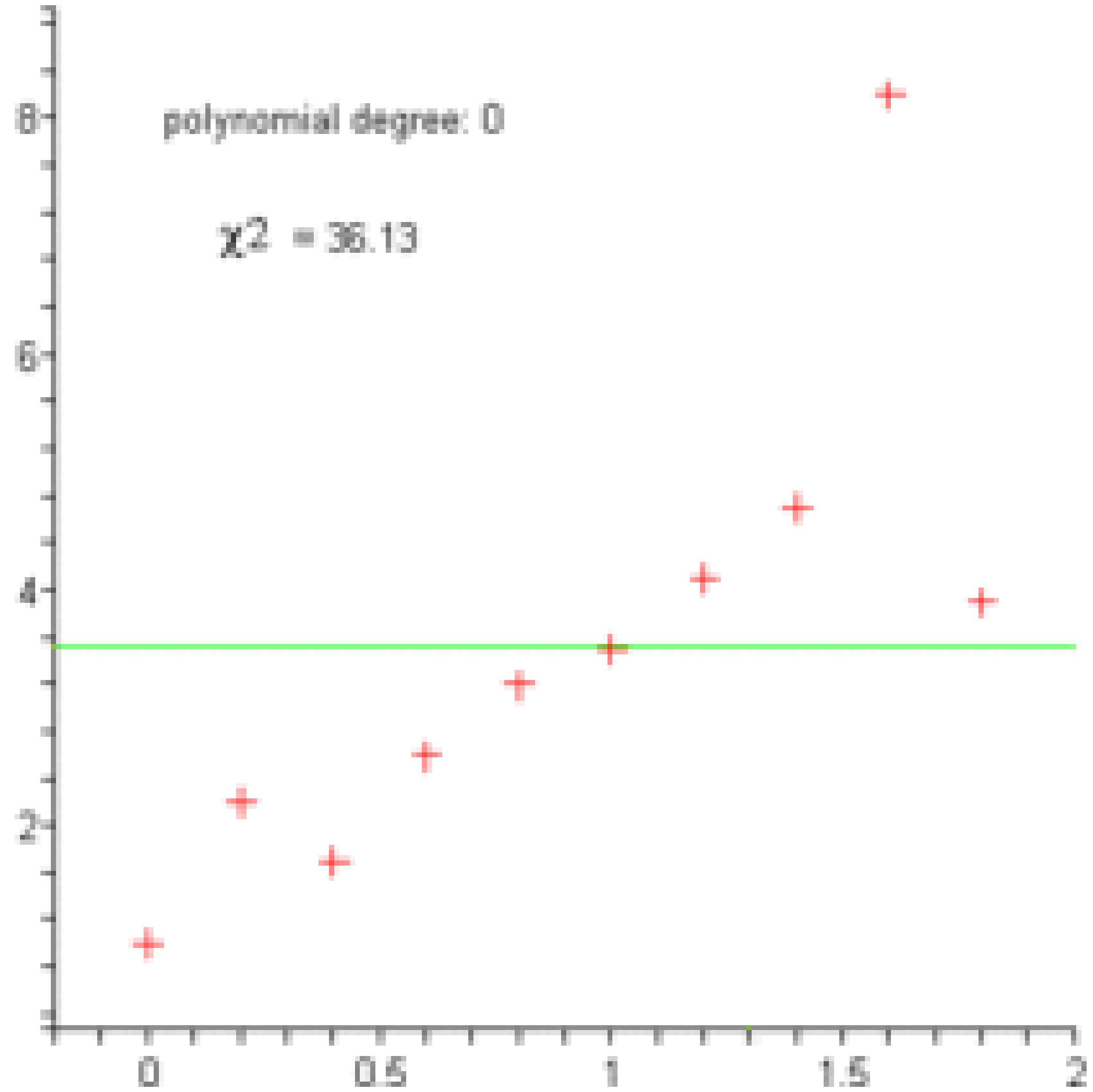


$$y=a_1x + a_2x^2 + b$$



$$yi = a_0 + a_1 x_i + a_2 x_i^{2} + \quad + a_{p-1} x_i^{p-1} + a_p x_i^{p} + ei$$

contoso


polynomial degree: 0
χ^2 = 36.13
8
6
4
2
0
0.5
1
1.5
2

La Régression Polynomiale:

La régression polynomiale est un modèle non linéaire simple ou nous avons deux variables Y et X :

- Y est la variable expliquée et
- X est la variable explicative , seulement ici X est donnée sous forme d'une Fonction plus complexe de la seule entrée X.

But de la régression Polynomiale:

Le but de la régression polynomiale est d'ajuster une série de points expérimentaux, par un polynôme.

Régression Polynomiale:

La base de la démarche est identique à la régression linéaire :

- Pour un jeu de données (x_i, y_i)i=1…n,
-
-R: $\left(\sum_{i=1}^{n} ei^2\right)$ est une fonction des paramètres $(a_j)_{,j=1..D}$

- Appliquer MMCO→minimiser(R)

Régression Polynomiale:

La MMCO est encore utilisée, c'est-à-dire minimiser la somme des carrés des écarts entre valeurs expérimentales et valeurs calculées par le polynôme.

Régression Polynomiale:

Si R est minimum, alors :

$$\frac{\partial R}{\partial aj} = 0 \qquad \text{pour tout } j$$

FSi

si ces dérivées existent alors :

Cela fournit un système de +EURS équations, en général non linéaires, qu'il n'est pas possible de résoudre de manière analytique.

On utilise des Algorithmes Itératifs pour résoudre ce système.

Parmi ces algorithmes :

-Algorithme de Gauss-Newton ;

-Algorithme de Levenberg-Marquardt ;

-(Algorithme du gradient)

R**égression Polynomiale:**

Formulation du problème :

-y variable expliquée = f(x)

-x_i, x_i^2,x_i^p sont appelés features (caractéristiques) ils représentent une fonction (complexe) de l'entrée X on va les appeler **h(x)**.

On reprend le model:

$$yi= a_0+a_1x_i+ a_2x_i^2+....+a_{p-1}x_i^{p-1}+a_px_i^p+e_i$$

Régression Polynomiale:

Feature 1=x^0=1
Feature 2=x^1
Feature 3=x^2
……
Feature p+1=x^P

Coefficient 1 = a_0
Coefficeint2= a_1….
Coefficient 3 = a_2

coefficient p+1= a_p

A chaque feature est associé un paramètre ou coefficient.

Régression Polynomiale:

<u>En général</u>:

$yi = a_0 h_0 (xi) + a_1 h_1(xi) + \ldots. + a_{p-1} h_{p-1}(xi) + a_p h_p(xi) + ei..(1)$

feature 1 $= h_0(xi) = x^0 = 1$

feature 2 $= h_1(xi) = x^1$

feature 3 $= h_2(xi) = x^2$ (ou bien sin(2pix/12)…)

feature p+1 $= h_p(xi) = x^p$

Régression Polynomiale:
La forme Matricielle est

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots\cdots \\ y_n \end{pmatrix} = \begin{pmatrix} h_0(x_1) & h_1(x_1) & h_2(x_1) & h_D(x_1) \\ h_0(x_2) & h_1(x_2) & h_2(x_2) & h_D(x_2) \\ h_0(x_3) & h_1(x_3) & h_2(x_3) & h_D(x_3) \\ & & & \\ h_0(x_n) & h_1(x_n) & h_2(x_n) & h_D(x_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \\ a_D \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \\ e_n \end{pmatrix}$$

$$\Rightarrow \quad Y = HA + \epsilon \quad \Rightarrow \quad \epsilon = Y - HA$$

Régression Polynomiale:

En plus======Nous avons :

$$R = \sum_{i=1}^{n} ei^{2}$$

ET Nous avons également :

$$: \ \epsilon t\epsilon = \sum_{i=1}^{n} ei^{2}$$

On Remplace R $= (Y - Ha)t(Y - Ha)$

<u>On donne</u> :

Grad (R)=Grad(-2H^t(Y – Ha)

D'une façon générale, et pour n'importe quelle fonction G(a) de coefficients inconnus (a)

l'algorithme est le suivant:

Début

t=1

Initialiser les coefficients aléatoirement par exemple ;

Initialiser ε:tolérance ;

Initialiser : η le 'pas' ;

While (not converged)

$a(t+1) \leftarrow a(t) - \eta \, grad(G(a))$

$t \leftarrow t+1$

Fin

Régression Polynomiale:

Sur_Apprentissage et Régularisation :

Dans la pratique, la régression polynomiale est souvent effectuée à l'aide d'une méthode d'apprentissage régularisée telle que la régression de Ridge.

Voir exemple de régression polynomiale utilisant scikit-learn en TP.

la régression de Ridge:
Avec cette régression, les coefficients W et b sont appris en minimisant la somme des écarts aux carré en plus d'un pénalité des valeurs élevées de ces coefficients.

$$RSS_{RIDGE}(\boldsymbol{w}, b) = \sum_{\{i=1\}}^{N} (\boldsymbol{y}_i - (\boldsymbol{w} \cdot \boldsymbol{x}_i + b))^2 \boxed{+ \alpha \sum_{\{j=1\}}^{p} w_j^2}$$

L'ajout de cette pénalité s'appelle régularisation, elle réduit la complexité du modèle et évite donc le surapprentissage.
L'influence de la régularisation est controlée par un paramètre α .
Plus α est grand plus le modèle est simple.

Régression Polynomiale:

-**Exemple**,
les prix des logements varient en fonction quadratique de la surface et du montant des taxes payées d'une habitation.
-Un modèle linéaire simple ne pourrait pas capturer cette relation non linéaire, mais en ajoutant des caractéristiques non linéaires telles que des polynômes au modèle de régression linéaire, nous pouvons capturer cette non-linéarité