MINITEL

MACHINE LEARNING

02 SUPERVISED MACHINE LEARNING : REGRESSION AND CLASSIFICATION

THÉO GACHET

Multiple Features

x_j = j^{th} feature

n = number of features

$\vec{x}^{(i)}$ = features of i^{th} training example

$x_j^{(i)}$ = value of feature j in i^{th} training example

Model

$$f_{\vec{w},b}(\vec{x}) = \sum_{i=1}^{n} w_i x_i + b = w_1 x_1 + \ldots + w_n x_n$$

$$\left.\begin{array}{l} \vec{x} = [x_1 \quad x_2 \quad \ldots \quad x_n] \\ \vec{w} = [w_1 \quad w_2 \quad \ldots \quad w_n] \end{array}\right\} \quad f_{\vec{w},b}(\vec{x}) = \vec{w} \underset{\text{dot product}}{\cdot} \vec{x} + b$$

Vectorization

Without vectorization

$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + \dots + w_n x_n$$

```
f = 0
for i in range(n):
    f += w[i] * x[i]
f += b
```

Vectorization

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

```
f = np.dot(w, x) + b
```

np.dot: multiplies in parallel

shorter
faster

Gradient Descent

$$\vec{w} = \vec{w} - \alpha \vec{d} \quad \text{where} \quad \vec{d} = \left[\frac{\partial J_{w_1}}{\partial w_1} \; \frac{\partial J_{w_2}}{\partial w_2} \dots \frac{\partial J_{w_m}}{\partial w_m} \right]$$

Gradient Descent for Multiple Linear Regression

Parameters

$\vec{w} = [w_1 \; \cdots \; w_n]$

b still a number

Model

$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Cost

$J(\vec{w}, b)$

Gradient Descent

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

Gradient Descent

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

$j=1$ $$w_1 = w_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_1^{(i)}$$

$\vdots$

$j=n$ $$w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_n^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)$$

Alternative to gradient descent : Normal Equation Method

Normal equation
- Only for linear regression
- Solve for w, b without iterations

Disadvantages
- Doesn't generalize to other learning algorithms.
- Slow when number of features is large (> 10,000)

What you need to know
- Normal equation method may be used in machine learning libraries that implement linear regression.
- Gradient descent is the recommended method for finding parameters w,b

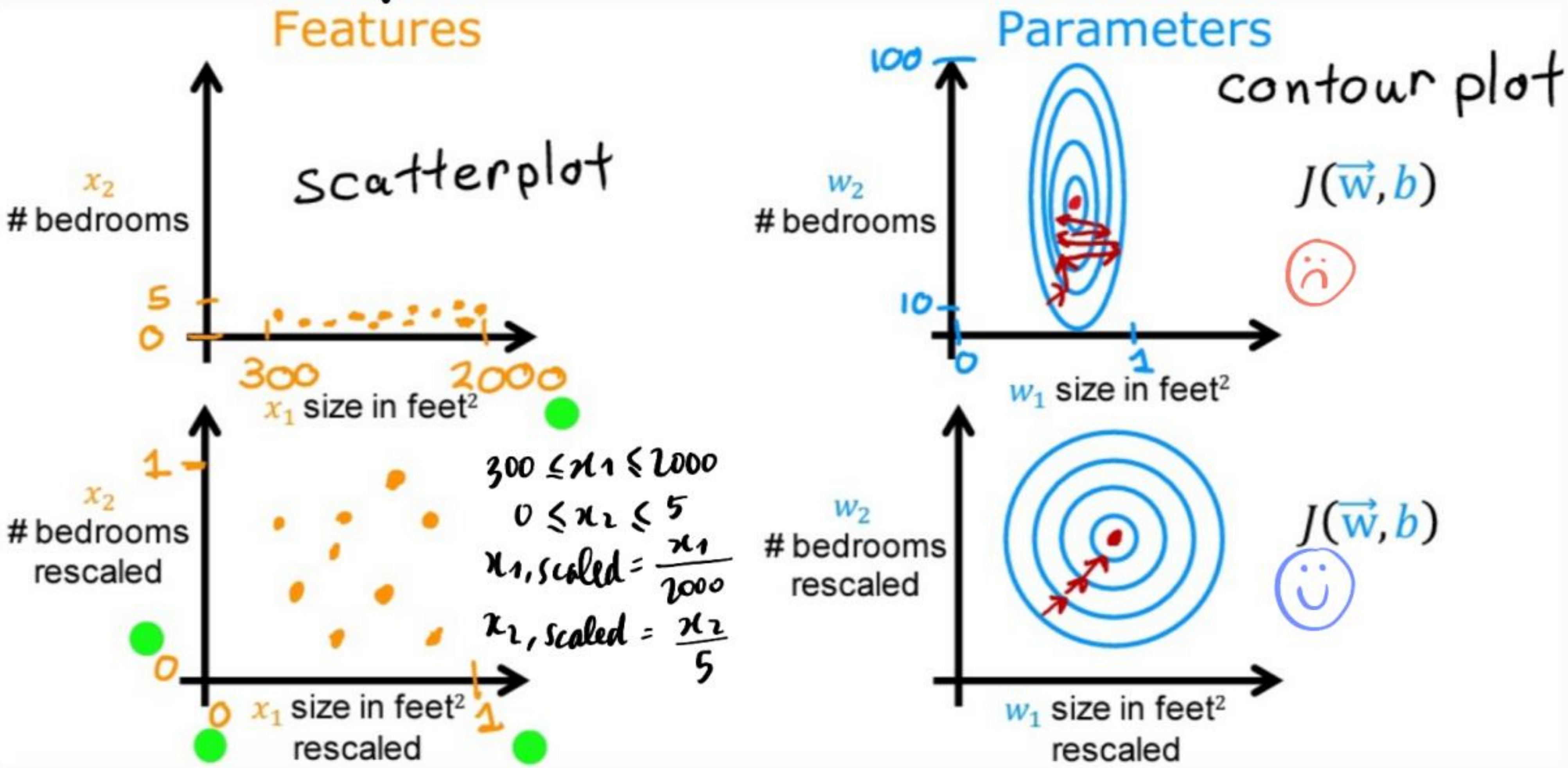
QUESTIONS ?

SUR UN CONCEPT ? UNE IDÉE ?
SUR UN DÉTAIL DU CODE ?
(ENVIE D'UNE PAUSE ?)

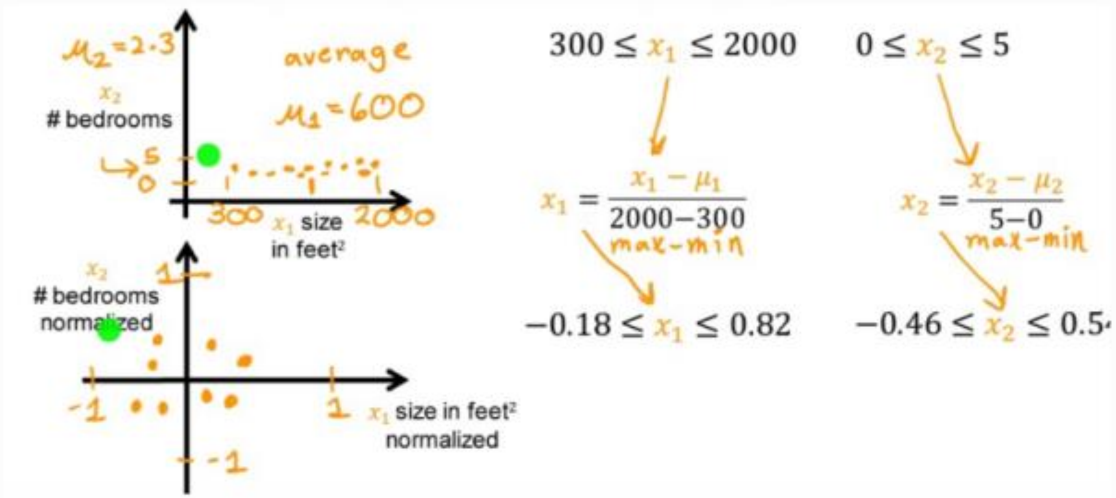
N'HÉSITEZ PAS !

IL N'Y A PAS DE QUESTION BÊTE, SI VOUS AVEZ UN DOUTE, D'AUTRES ONT SÛREMENT LE MÊME

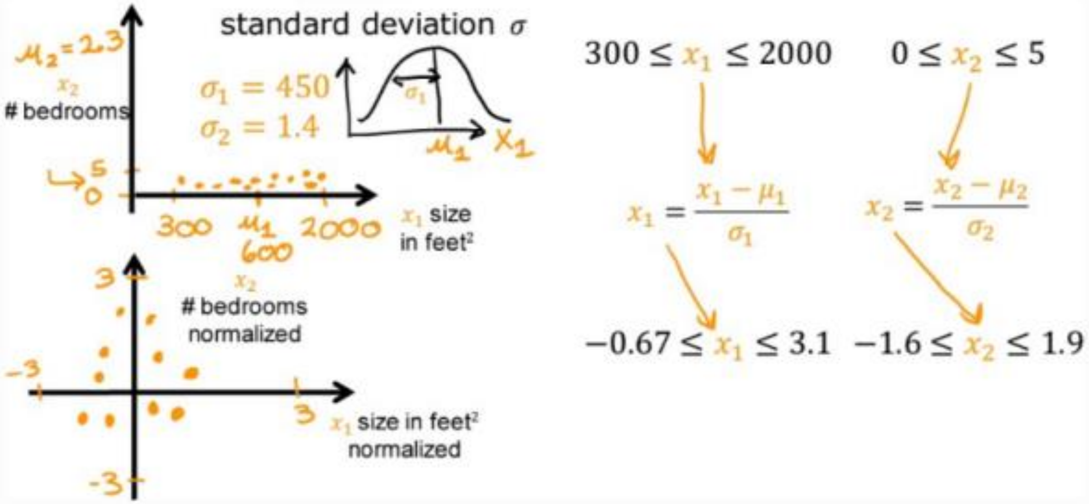
Feature Scaling



Mean Normalization



Z-score Normalization

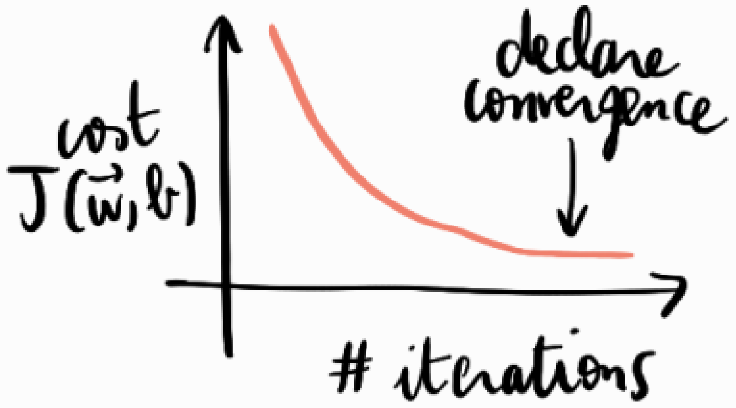


Feature Engineering Using intuition to design new features by using original features.

Example: if x_1 = frontage, x_2 = depth, then we can create $x_3 = x_1 \times x_2$ = area of the house, which is more relevant for price prediction $\Rightarrow f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3$

Gradient Descent Convergence

$J(\vec{w}, b)$ should decrease after each iteration



Automatic Convergence Test

let $\varepsilon = 10^{-3}$. If $J(\vec{w}, b)$ decreases by $\leq \varepsilon$ in 1 iteration, declare convergence
(found parameters $\vec{w}$, b to get close to global minimum)

QUESTIONS ?

SUR UN CONCEPT ? UNE IDÉE ?
SUR UN DÉTAIL DU CODE ?
(ENVIE D'UNE PAUSE ?)

N'HÉSITEZ PAS !

IL N'Y A PAS DE QUESTION BÊTE, SI VOUS AVEZ UN DOUTE, D'AUTRES ONT SÛREMENT LE MÊME

MINITEL

MACHINE LEARNING

02 SUPERVISED MACHINE LEARNING : REGRESSION AND CLASSIFICATION

THÉO GACHET