Machine Learning
& Intelligence Artificielle

Manuel Simoes
manuel.simoes@cpc-analytics.fr

- Régression Logistique -

Classification

Classification *v.s.* Régression

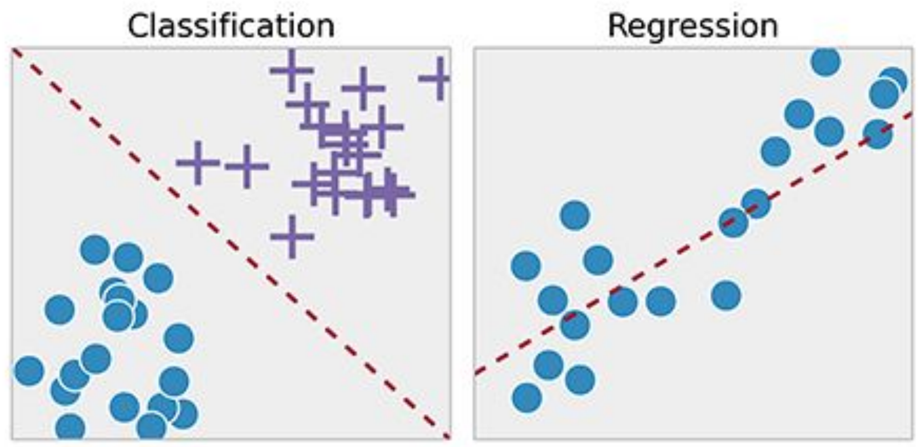


Illustration de la différence entre régression linéaire et classification linéaire

La régression logistique est une classification

Classification binaire

0 : "Classe Négative"
1 : "Classe Positive"

On utilise la classification binaire lorsque l'on veut répondre à une question par Oui (1, True) ou par Non (0, False).

Exemple de classification :

Email (développement): Ce courriel est-il un spam ? (Oui / Non)

Transaction frauduleuse (secteur bancaire): Cette transaction bancaire internet est-elle frauduleuse ?

Tumeur (santé) : Maligne / Bénigne ?

Relation Client (Marteking): Ce client va quitter notre clientèle (ou pas) ?

Classification binaire

Définition d'une frontière entre les deux zones

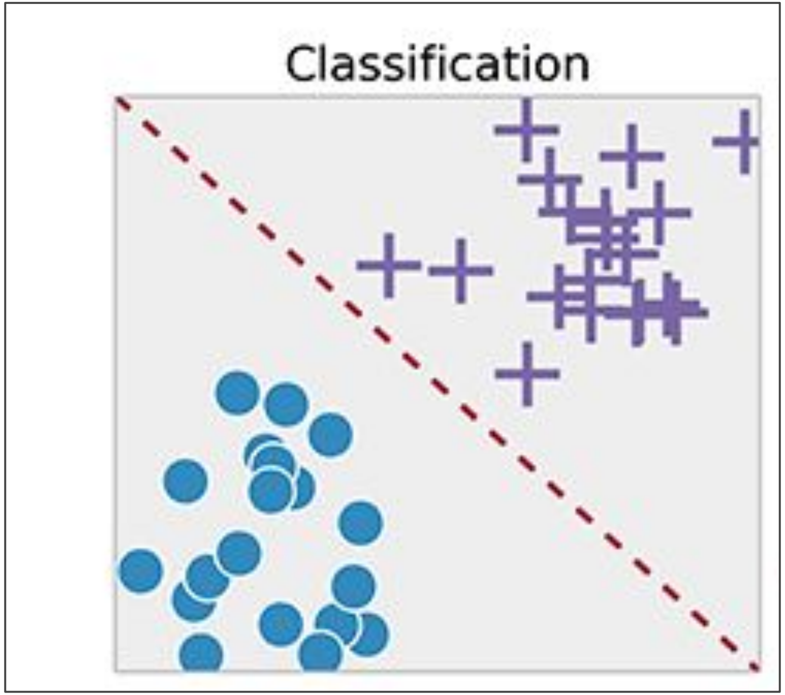
Ici la réflexion va être similaire à la régression linéaire.

- On introduit la notion de Probabilité dans la régression logistique.
    - Spam : Oui ou Non ...

- On garde la notion de modèle mathématique
    - une droite peut séparer 2 populations différentes (fig de gauche).

    - Une fonction polynomiale peut créer des séparations des ensembles dont la frontière est plus complexe.

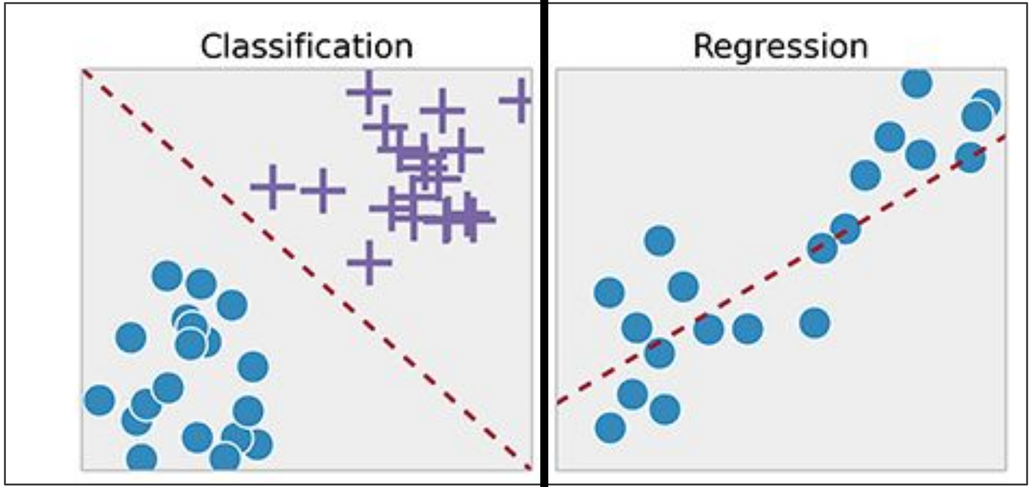
- On intègre une fonction coût qu'il faut minimiser pour ajuster les paramètres du modèle mathématique.



Classification binaire : Probabilité

Les paramètres θ_1 et θ_2 ont été calculé pour séparer deux populations.

Ici, les paramètres θ_1 et θ_2 ont été calculé pour s'ajuster aux données.



Ici la droite sépare les deux classes et doit représenter une probabilité d'appartenir à un des groupes.

Les valeurs de la droite représentent les prédictions (quelques soient leur valeurs)

Classification binaire : Probabilité

0 : “Classe Négative”

1 : “Classe Positive”

P($y=1$ **|** x **)** est la probabilité d’être dans la classe Positive

P($y=0$ **|** x **)** est la probabilité d’être dans la classe Négative

Pour avoir une probabilité, la fonction **P** doit suivre les conditions suivantes :

- $P(y=1 \mid x)$ est compris entre 0 et 1 (inclus)
- $P(y=0 \mid x) = 1 - P(y=1 \mid x)$

Notation : ***x*** sont les données d'entrées (valeur numérique, colonnes, images, texte...)

Interprétation de la sortie h

$h_\theta(x)$ estime la probabilité d'avoir ***y = 1*** avec l'input ***x***

Exemple: Si $$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$

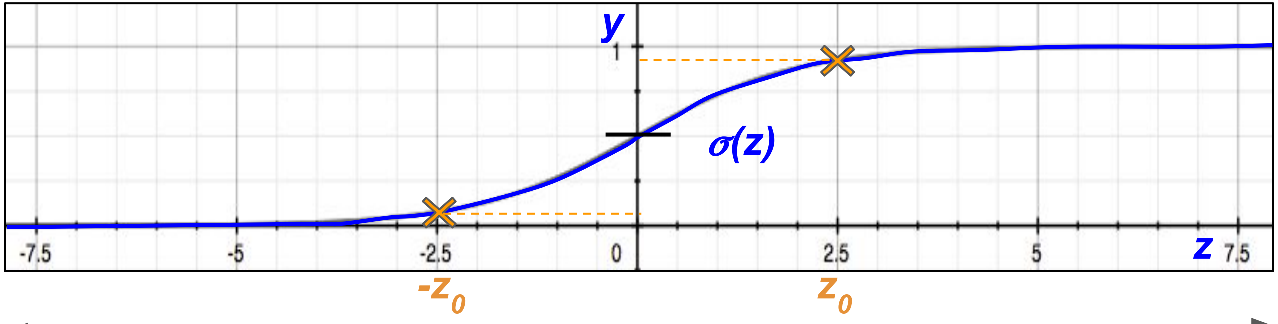
$h_\theta(x) = 0.7$ –> *Le patient à 70 % de chance d'avoir une tumeur maligne*

La probabilité d'avoir ***y = 0*** avec l'input ***x*** est donnée par

$$P(y = 0|x;\theta) + P(y = 1|x;\theta) = 1$$
$$P(y = 0|x;\theta) = 1 - P(y = 1|x;\theta)$$

Régression Logistique : hypothèse

La fonction Sigmoïde (ou fonction logistique)



$z \in \Re$

$$\boxed{\sigma(z) = \frac{1}{1+e^{-z}}}$$

$\sigma(z) \in \{0, 1\}$

Plus ***z*** tend vers les valeurs négatives décroissante et plus la valeur de la sigmoïde tend vers ***0***.

Plus ***z*** tend vers les positifs croissant, et plus la valeur de la sigmoïde tend vers ***1***.

La fonction sigmoid donne une probabilité :

On a $\sigma(z_0) = 1 - \sigma(-z_0)$

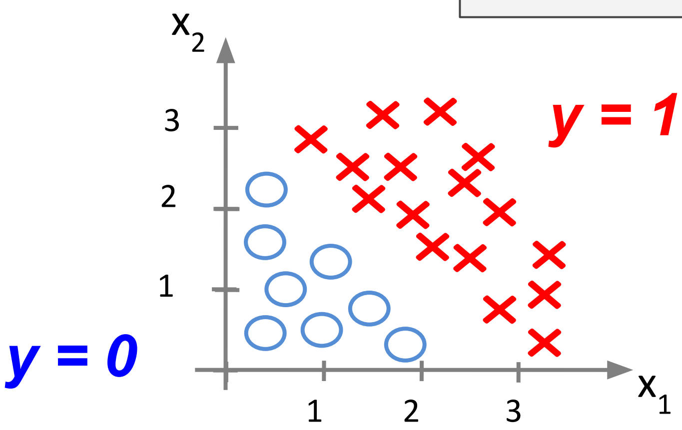
$\sigma(z) \in \{0, 1\}$

La probabilité d'avoir y=0 avec la donnée x

$$P(y=0 \mid x) = 1 - P(y=1 \mid x)$$

Régression Logistique : linear Decision Boundaries

Nous “cherchons” une droite qui sépare les données rouges des bleues.

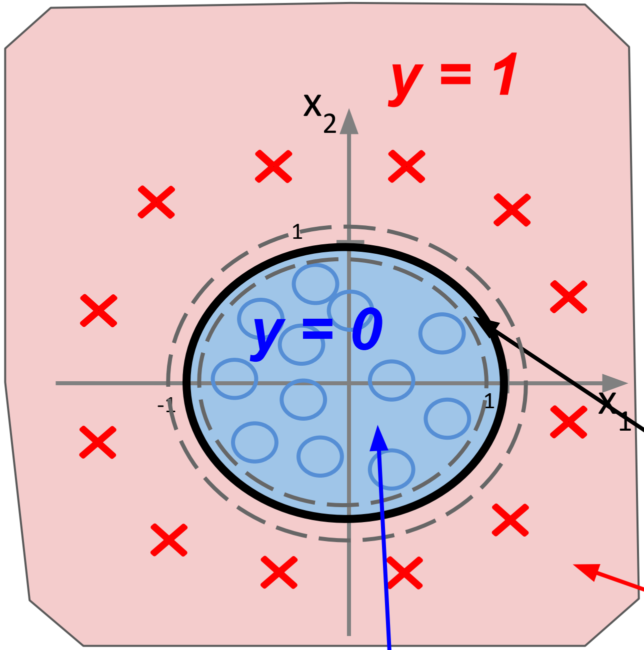


$$h_\theta(x) = g(\theta^T x)$$

$$h_\theta(x) = g(\underset{-3}{\theta_0} + \underset{1}{\theta_1} x_1 + \underset{1}{\theta_2} x_2)$$

Imaginons que nous avons déjà paramétré les coefficients θ^i (coloré en violet)

Régression Logistique : Non-linear Decision Boundaries



Cette fois nous prenons un polynôme (multivarié) du 2ème degré.

$$h_\theta(x) = g(\theta^T x)$$

$$h_\theta(x) = g(\underset{-1}{\theta_0} + \underset{0}{\theta_1} x_1 + \underset{0}{\theta_2} x_2 + \underset{1}{\theta_3} x_1^2 + \underset{1}{\theta_4} x_2^2)$$

En vert la valeur des coefficients des paramètres THETA (coloré en vert)

Si $x_1^2 + x_2^2 - 1 = 0$ la probabilité d'être égale à *1* est donné par
$h(x) = \sigma(x_1^2 + x_2^2 - 1) = 0.5$ *(50 %)*
-> la ligne de séparation des aires -> Decision Boundary

Si $x_1^2 + x_2^2 - 1 > 0$ la probabilité d'être égale à *1* est donné par
$h(x) = \sigma(x_1^2 + x_2^2 - 1) > 0.5$
-> l'aire rouge -> y = 1

Si $x_1^2 + x_2^2 - 1 < 0$ la probabilité d'être égale à 1 est donné par
$h(x) = \sigma(x_1^2 + x_2^2 - 1) < 0.5$
-> l'aire bleu -> y = 0

Plus on s'éloigne de la ligne de décision ("Decision Boundary") et plus la probabilité tend "rapidement" vers 0 ou 1 (voir la courbe de la sigmoid).

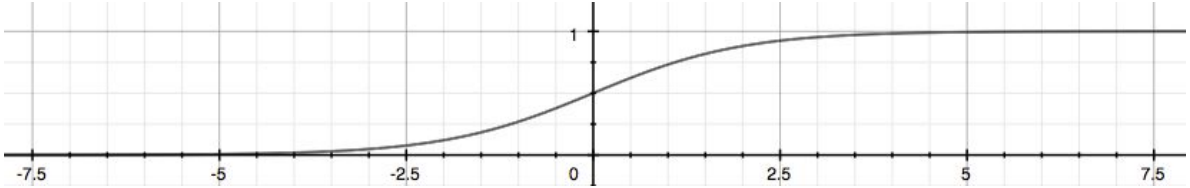
$$y = 1 \ \text{si} \ x_1^2 + x_2^2 \geq 1$$
$$y = 0 \ \text{si} \ x_1^2 + x_2^2 < 1$$

Modèle de Régression Logistique

Données d'entraînement (m) $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

Nombre de feature (n) $x \in \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix}$

Une sortie



$y \in \{0, 1\}$

Modèle $$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Régression Logistique : La fonction coût

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \Big(\underbrace{h_\theta(x^{(i)})}_{\hat{y}^{i}} - \underbrace{y^{(i)}}_{y^{i}} \Big)^2$$

Erreur du modèle pour chaque valeur prédite : c'est la différence entre
la prédiction ŷ, et
la valeur réel y.

Cette forme d'écriture, de la fonction coût, n'est pas intéressante pour les régressions logistiques car elle crée des fonctions non convexes avec de nombreux minima locaux dans lequel l'algorithme du "Gradient descent" peut se perdre.

Nous préférons définir la fonction coût par morceaux en fonction de y :

- dans le cas où Y = 0
- et dans le cas ou Y = 1

La forme générale de la fonction coût peut s'écrire

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}\big(\underbrace{h_\theta(x^{(i)})}_{\hat{y}^{i}}, \underbrace{y^{(i)}}_{y^{i}}\big)$$

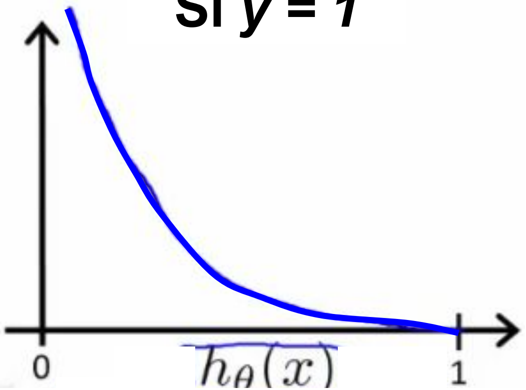
Régression Logistique : La fonction coût

Cas où Y= *1*

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Étude de la fonction coût

Si *y* = *1*



Le **log(** *h(x)* **)** tends vers 0

*quand h(x) tend vers **1***

-> Réduction du coût de l'erreur lorsqu'on tend vers Y = 1

-> Pas de pénalité pour la bonne solution

par contre

Le **log(** *h(x)* **)** tends vers ∞

*quand h(x) tend vers **0***

-> Augmentation du coût de l'erreur lorsqu'on tend vers Y = 0

-> Pénalité forte pour la mauvaise solution

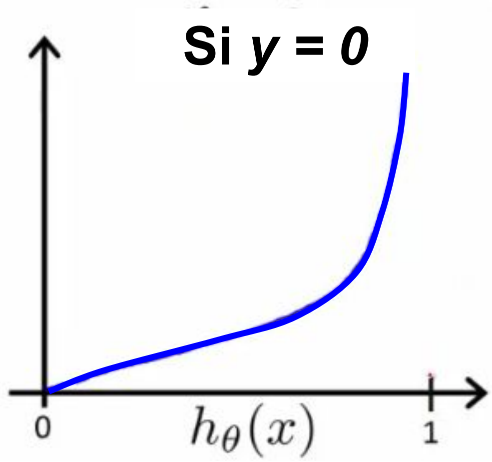
Si ***y* = *1*** et que l'on prédit ***ŷ* = *0*,**
une pénalité « *infini* » est introduite dans la fonction coût.

Régression Logistique : La fonction coût

Cas Y= *0*

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Étude de la fonction coût



Le **log(** *1 - h(x)* **)** tends vers ∞

quand h(x) tend vers 1 –> 1 - h(x) tend vers 0

- > Augmentation du coût de l'erreur lorsqu'on tend vers Y = 1 (la mauvaise solution)

par contre

Le **log(** *1 - h(x)* **)** tends vers 0

quand h(x) tend vers 0 –> 1 - h(x) tend vers 0 quand h(x) tend vers ***0***

- > Augmentation du coût de l'erreur lorsqu'on tend vers Y = 0 (la mauvaise solution)

Si ***y* = *0*** et que l'on prédit ***h(x)* = *1***, la fonction coût tend vers une pénalité « infini »

→ Compatible avec le comportement d'une fonction erreur

Régression Logistique : La fonction coût

Cas Y= *0*

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

On réécrit la fonction coût en une seule expression

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Écriture Matricielle

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1-y)^T \log(1-h)\right)$$

Régression Logistique : Optimisation

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1-h_\theta(x^{(i)}))]$$

On veux $\min_\theta J(\theta)$

Itérer

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$ (Mise à jour simultanément θ_j)

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$ Mise a jour simultanément (tous les thetas)

Classification multi-classe

Ciblage d’émail: Travail, Amis, Famille, Hobby...

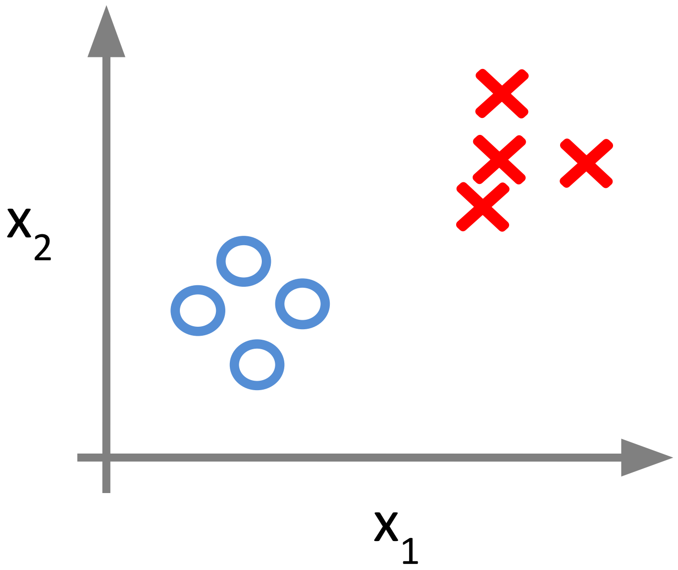
Diagramme médical: Pas malade, Bactérie, Virus...

Méteo : Ensoleillé, Nuageux, Rain, Snow...

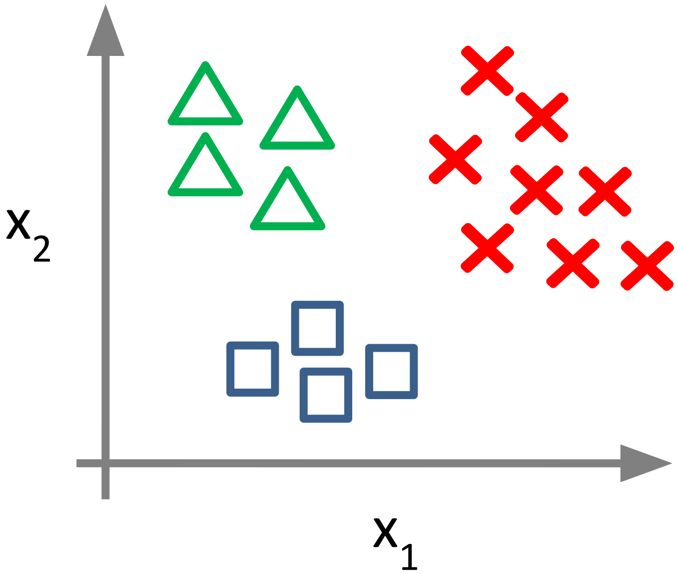
Classification multi-class

Classification Binaire

Binary classification



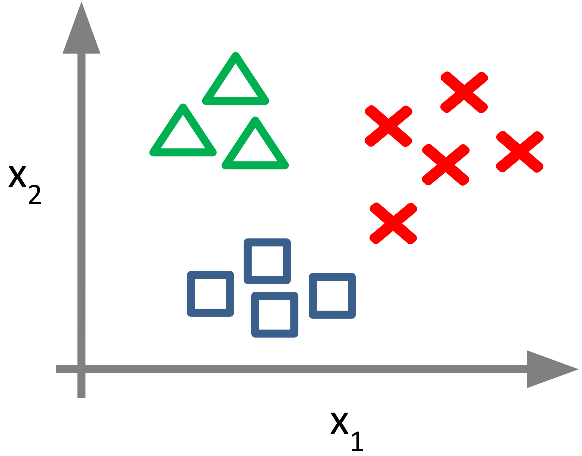
Classification multi-class

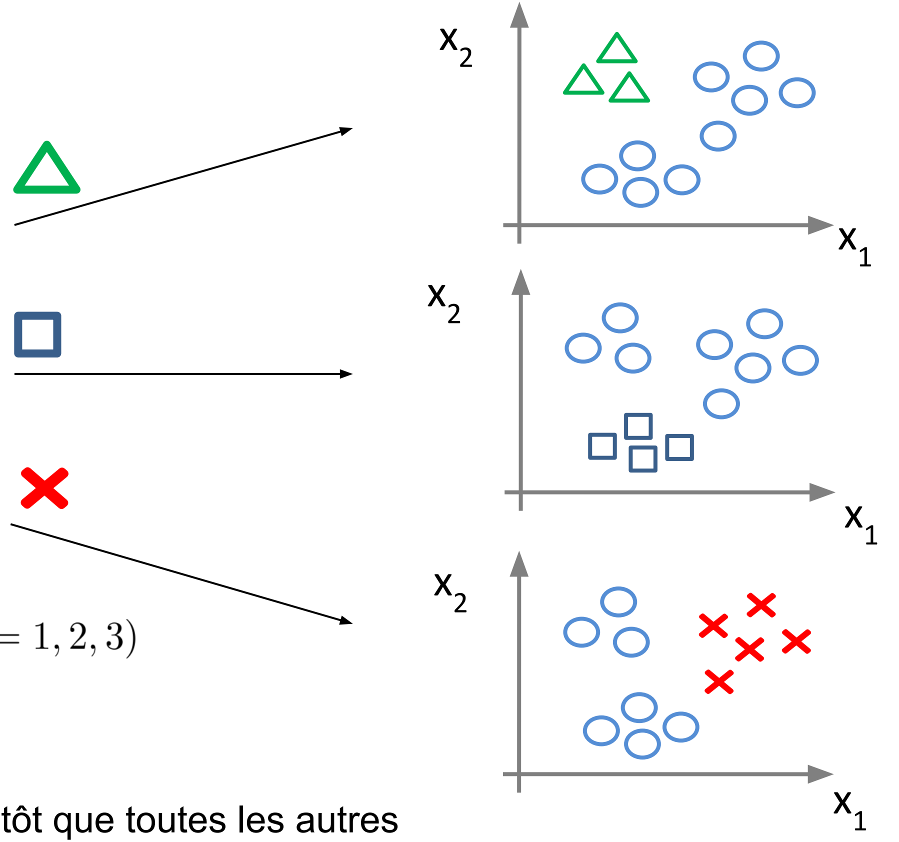


Classification multi-classe : One *versus* All

One-*versus*-all

Dans notre cas cela correspond a 3 one-versus-rest.





Class 1:
Class 2:
Class 3:

$$h_\theta^{(i)}(x) = P(y = i | x; \theta) \qquad (i = 1, 2, 3)$$

On calcul les probabilités d'être de la Class1 plutôt que toutes les autres classes. On fait de même pour les Class2 et Class3.

Classification multi-classe : One versus All

Entraîner un classifieur « logistic regression » $h_\theta^{(i)}(x)$
pour chaque classe ***i*** pour prédire la probabilité que

Pour un input ***x*** prendre la classe avec la prédiction la plus forte soit :

$$\max_i h_\theta^{(i)}(x)$$