# Apprentissage supervisé

HADJADJ ABDELHALIM

**Centre universitaire de Mila**

**Master I: Matière Intelligence artificielle : Principes et Applications**

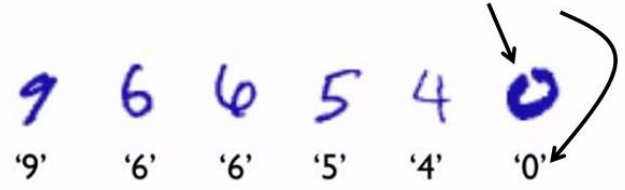
# Apprentissage supervisé

**Principe de l'apprentissage supervisé**

Les algorithmes d'apprentissage supervisé procèdent comme suit:

- On fournit à l'algorithme des données d'entrainement:

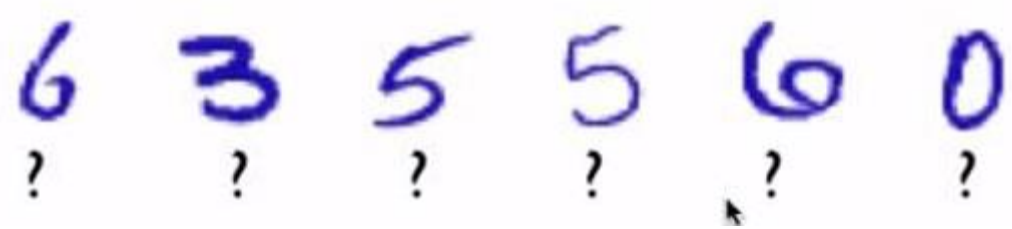
$$\mathcal{D} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(N)}, y^{(N)})\}$$




- On appelle $x(i)$ l'entrée et $y(i)$ la cible du i-iéme exemple.
- Un élément de D est appelé exemple d'apprentissage ou une instance de données.

# Apprentissage supervisé

• L'algorithme retourne un «programme» capable de se généraliser à de nouvelles données:




On note le «programme» généré par l'algorithme d'apprentissage *f(x).*

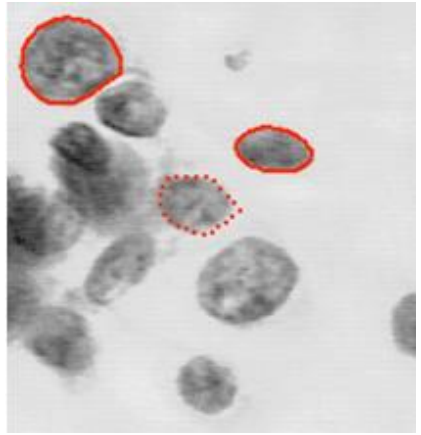
• On appelle *f(x)* un modèle ou une hypothèse.

• On utilise souvent un ensemble de test *Dtest* pour mesurer la performance du modèle *f(x)*.

# Apprentissage supervisé

## Exemple motivation

- Des cellules cancéreuses sont prises de tumeurs de cancer du sein avant la chirurgie et elles sont photographiées.



- Les tumeurs sont excisées.
- Les patients sont suivis pour voir s'il y a récurrence du cancer.

On mesure le temps avant que la récurrence du cancer ou que le patient est déclaré sans la maladie.

# Apprentissage supervisé

## Exemple motivation

- On utilise 30 caractéristiques par tumeur.

- Deux variables sont prédites: Résultat ( R: récurrence, N: non-récurrence). Temps (Jusqu'a récurrence, pour R, et en santé, pour N).

- L'ensemble de données est représenté par une matrice X :

$\mathbf{X}$

| tumor size | texture | perimeter | . . . | outcome | time |
|---|---|---|---|---|---|
| 18.02 | 27.6 | 117.5 | | N | 31 |
| 17.99 | 10.38 | 122.8 | | N | 61 |
| 20.29 | 14.34 | 135.1 | | R | 27 |
| . . . | | | | | |

# Apprentissage supervisé

Les colonnes sont appelées variables d'entrée, attributs ou caractéristiques.

- Le résultat et le temps (que nous essayons de prédire) sont appelés les variables résultats ou les cibles.
- Une ligne du tableau est appelée un exemple d'entrainement ou instance.
- Le tableau en entier est appelé l'ensemble d'entrainement.

# Types de prédiction

Le problème de prédiction des résultats de la maladie est appelé une classification.

| tumor size | texture | perimeter | . . . | outcome | time |
|---|---|---|---|---|---|
| 18.02 | 27.6 | 117.5 | | N | 31 |
| 17.99 | 10.38 | 122.8 | | N | 61 |
| 20.29 | 14.34 | 135.1 | | R | 27 |

- Le problème de prédiction du temps est appelé régression.

# Types de prédiction

- Un exemple d'entrainement à la forme x($i$), $y(i)$ , où:

$$x^{(i)} = \left(x_1{}^{(i)}, x_2{}^{(i)}, \cdots, x_D{}^{(i)}\right)$$

- $D$ est le nombre d'attributs (dans notre cas 30).
- L'ensemble d'entrée D contient $N$ exemples.
- On dénote par X l'espace des variables d'entrées (ex. $\mathbb{R}$D)
- On dénote par Y l'espace des variables de sortie (ex. $\mathbb{R}$)

# Apprentissage supervisé

Ayant un ensemble de données $D \subset X \times Y$, trouver une fonction

$$f: X \rightarrow Y$$

- Telle que $f$ est un bon prédicteur de la valeur de $y$. La fonction $f$ est appelée une hypothèse.
- Les problèmes sont classés par type de domaine de sortie:
  - Si Y= $\mathbb{R}$, on parle alors de régression.
  - Si Y est un ensemble discret fini, on parle de classification.
  - Si Y a 2 éléments, on parle de classification binaire.

# Apprentissage supervisé

**Étapes de résolution d'un problème d'apprentissage supervisé**

- Identifier les variables d'entrée et de sortie.
- Déterminer le codage des variables d'entrée (X) et de sortie (Y).
- Sélectionner la classe d'hypothèses ou les représentations possibles (H).
- Trouver l'hypothèse $f$ optimale pour effectuer la prédiction.

# Principe de la régression

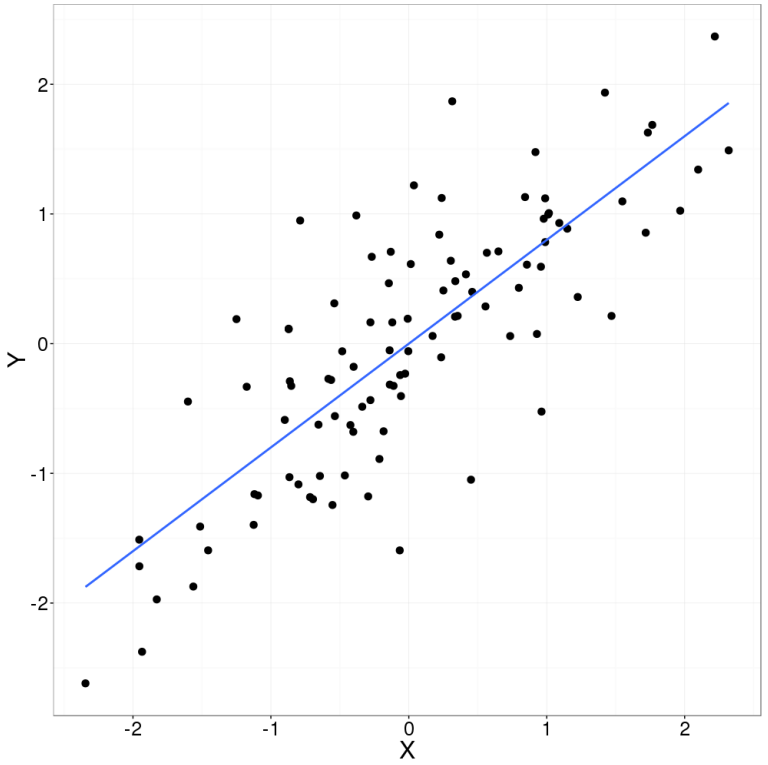
**Définition :** **Méthode statistique pour modéliser la relation entre variables**

**Objectif :** **Prédire une variable dépendante (Y) à partir de variable(s) indépendante(s) (X)**

# Principe de la régression

## Exemple: (régression linéaire)

| $x$ | $y$ |
|---|---|
| 0.86 | 2.49 |
| 0.09 | 0.83 |
| -0.85 | -0.25 |
| 0.87 | 3.10 |
| -0.44 | 0.87 |
| -0.43 | 0.02 |
| -1.10 | -0.12 |
| 0.40 | 1.81 |
| -0.96 | -0.83 |
| 0.17 | 0.43 |



# Principe de la régression

En termes algébriques, le modèle serait défini comme suit :

***y=wx+b,***

- **Y:** est la valeur que nous voulons prédire.
- **W:** est la pente de la droite.
- ***x:*** correspond à  notre valeur d'entrée.
- **B:** est l'ordonnée à l'origine.

# Principe de la régression

L'hypothèse d'une r égression linéaire s'écrit sous la forme suivante :

$$y = f(x) = w_0 + w_1 x_1 + \cdots$$

où $x = (x_1, x_2, \ldots, x_D)$.

Pour faciliter l'écriture, on introduit un attribut supplémentaire x0=1, ce qui permet de réécrire la formule sous forme de produit scalaire :

$$y = f(x) = \sum_{d=0}^{D} w_d x_d = \tilde{\mathbf{w}}^T \tilde{x}$$

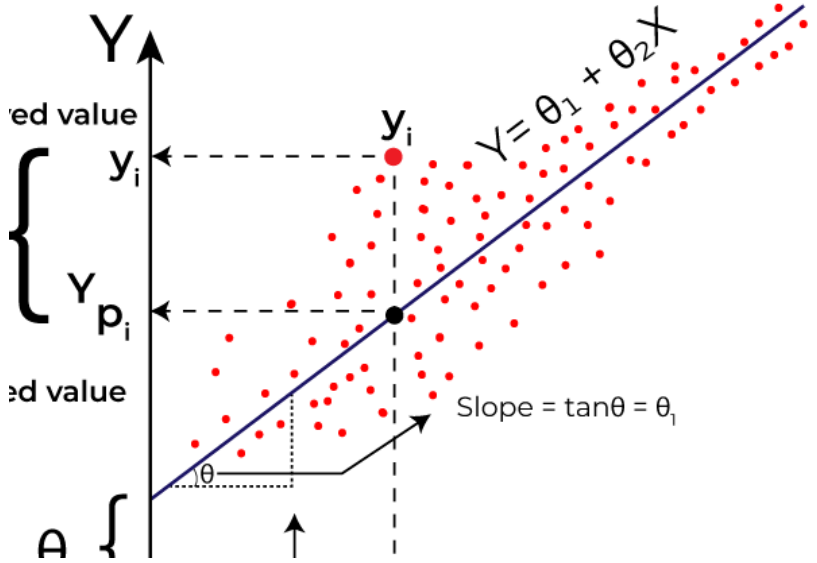
sous notation vectorielle

# Principe de la régression

L'objectif de la régression linéaire est de trouver les paramètres w qui permettent au modèle f(x) de prédire des valeurs y aussi proches que possible des valeurs réelles.

Pour ajuster correctement w, nous devons :

1. **Définir une fonction d'erreur ou de perte** qui mesure l'écart entre les prédictions du modèle et les valeurs réelles.
1. **Minimiser cette fonction d'erreur** afin de trouver les meilleurs paramètres w.



# Erreur des moindres carrés

- Une fonction d'erreur couramment utilisée est **l'erreur quadratique moyenne** (Mean Squared Error - MSE), aussi appelée **erreur des moindres carrés** :
- Essayer de rendre f(x) très proche des valeurs des y dans tous les exemples d'apprentissage dans D,

$$E(\tilde{\mathbf{w}}) = \frac{1}{2}\sum_{i=1}^{N}\left(f\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- OU

- *N est le nombre d'exemples d'entraînement,*
- *yi est la valeur réelle associée à 'entrée xi*
- *f(xi)=wTx est la prédiction du modèle.*

- **L'objectif est de trouver w qui minimise cette erreur.**

# Méthodes d'optimisation des paramètres w

**Dérivation de la Fonction d'Erreur**

$$\frac{\partial E(\tilde{\mathbf{w}})}{\partial w_d} = \frac{\partial}{\partial w_d}\left(\frac{1}{2}\sum_{i=1}^{N}\left(f\left(x^{(i)}\right) - y^{(i)}\right)^2\right)$$

$$= 2\left(\frac{1}{2}\sum_{i=1}^{N}\left(f\left(x^{(i)}\right) - y^{(i)}\right)\frac{\partial}{\partial w_d}\left(f\left(x^{(i)}\right) - y^{(i)}\right)\right)$$

$$= \sum_{i=1}^{N}\left(f\left(x^{(i)}\right) - y^{(i)}\right)\frac{\partial}{\partial w_d}\left(\sum_{d=0}^{D} w_d x_d^{(i)} - y^{(i)}\right)$$

$$= \sum_{i=1}^{N}\left(f\left(x^{(i)}\right) - y^{(i)}\right)x_d^{(i)}$$

# Méthodes d'optimisation des paramètres w

En notation matricielle, l'erreur quadratique moyenne est écrite comme :

$$E(\mathbf{w}) = \frac{1}{2N}(\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y})$$

Sa dérivée par rapport à w est donnée par :

$$\nabla E(\mathbf{w}) = \mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y})$$

En utilisant la **formule de Taylor**, on obtient :

En posant ∇E(w)=0 pour minimiser l'erreur, on obtient $\nabla E(\mathbf{w}) = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y}$

$$2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} = 0$$

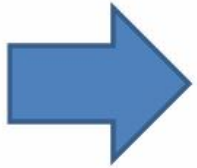
$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

Si la matrice XTXest inversible (ce qui signifie que les colonnes de X sont **linéairement indépendantes**), on peut isoler w :

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

# Exemple

| $x$ | $y$ |
|---|---|
| 0.86 | 2.49 |
| 0.09 | 0.83 |
| -0.85 | -0.25 |
| 0.87 | 3.10 |
| -0.44 | 0.87 |
| -0.43 | 0.02 |
| -1.10 | -0.12 |
| 0.40 | 1.81 |
| -0.96 | -0.83 |
| 0.17 | 0.43 |



$$\mathbf{X} = \begin{bmatrix} 0.86 & 1 \\ 0.09 & 1 \\ -0.85 & 1 \\ 0.87 & 1 \\ -0.44 & 1 \\ -0.43 & 1 \\ -1.10 & 1 \\ 0.40 & 1 \\ -0.96 & 1 \\ 0.17 & 1 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 2.49 \\ 0.83 \\ -0.25 \\ 3.10 \\ 0.87 \\ 0.02 \\ -0.12 \\ 1.81 \\ -0.83 \\ 0.43 \end{bmatrix}$$

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 0.86 & 0.09 & -0.85 & 0.87 & -0.44 & -0.43 & -1.10 & 0.40 & -0.96 & 0.17 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.86 & 1 \\ 0.09 & 1 \\ -0.85 & 1 \\ 0.87 & 1 \\ -0.44 & 1 \\ -0.43 & 1 \\ -1.10 & 1 \\ 0.40 & 1 \\ -0.96 & 1 \\ 0.17 & 1 \end{bmatrix} = \begin{bmatrix} 4.95 & -1.39 \\ -1.39 & 10 \end{bmatrix}$$

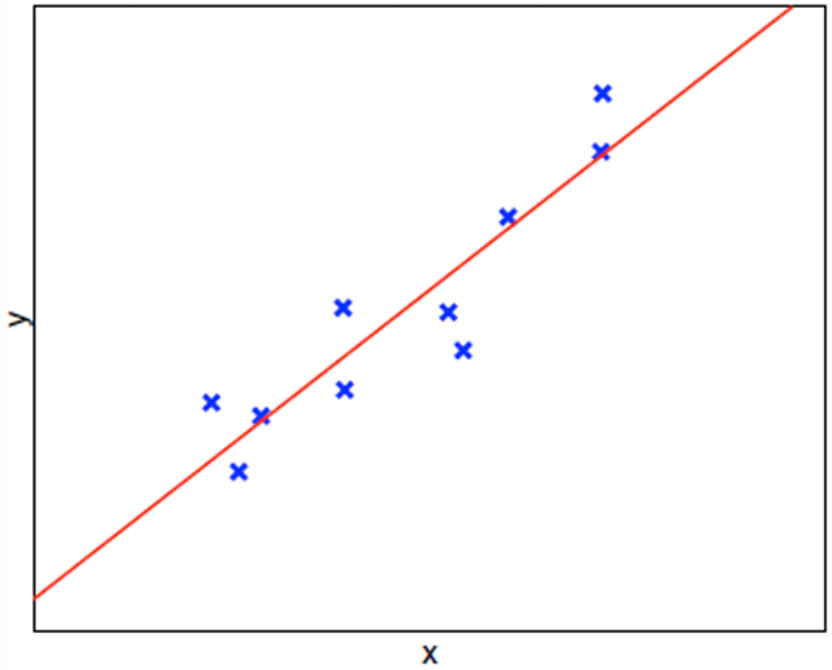
# Exemple

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 0.86 & 0.09 & -0.85 & 0.87 & -0.44 & -0.43 & -1.10 & 0.40 & -0.96 & 0.17 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2.49 \\ 0.83 \\ -0.25 \\ 3.10 \\ 0.87 \\ 0.02 \\ -0.12 \\ 1.81 \\ -0.83 \\ 0.43 \end{bmatrix} = \begin{bmatrix} 6.49 \\ 8.34 \end{bmatrix}$$

$$\widetilde{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} = \begin{bmatrix} 4.95 & -1.39 \\ -1.39 & 10 \end{bmatrix}^{-1} \begin{bmatrix} 6.49 \\ 8.34 \end{bmatrix} = \begin{bmatrix} 1.60 \\ 1.05 \end{bmatrix}$$

# Exemple

La meilleure ligne est égale alors à: y=1.6$x$ +1.05

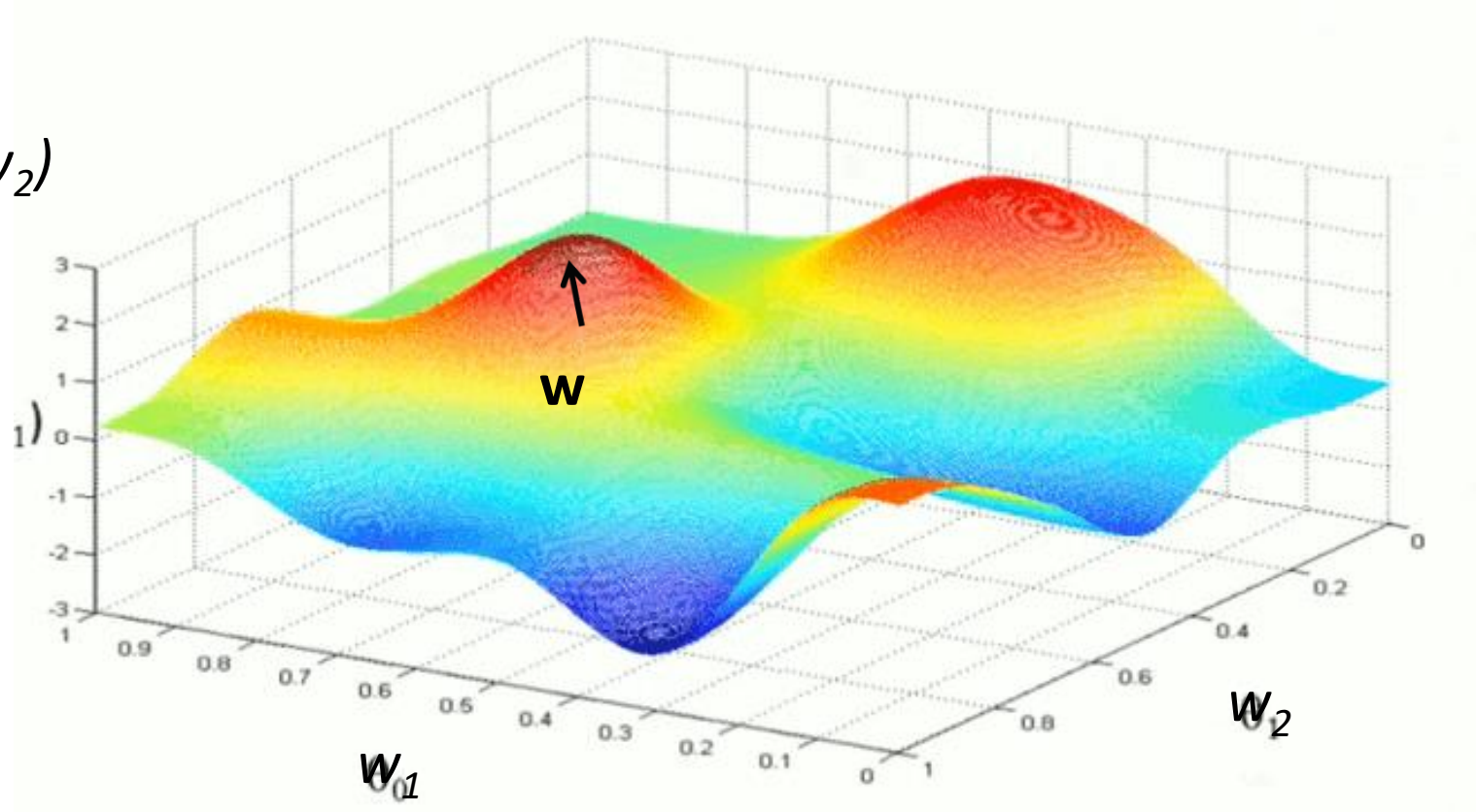


# La descente de gradient

- La descente de gradient est une méthode d'optimisation utilisée pour minimiser une fonction, généralement appelée fonction objectif ou fonction de coût .

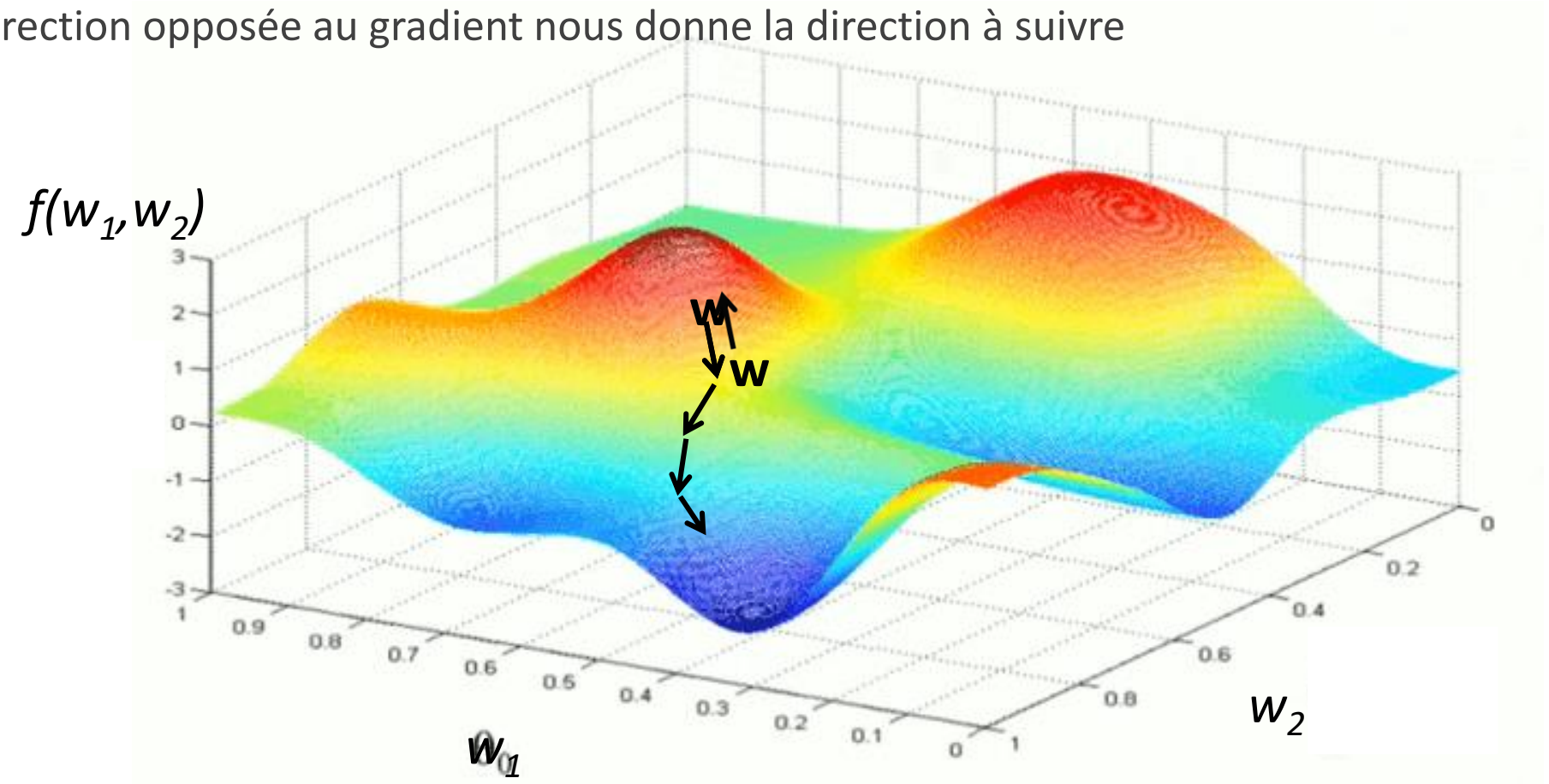
# exemple:La descente de gradient

Le gradient donne la direction (vecteur) ayant le taux d'accroissement de la fonction le plus élevé



# exemple: La descente de gradient

La direction opposée au gradient nous donne la direction à suivre



L'idée principale derrière la descente de gradient repose sur deux concepts fondamentaux :

1. **Gradient** : Le gradient est un vecteur qui indique la direction de la plus forte augmentation de la fonction. En mathématiques, il est représenté par $\nabla J(\theta)$, où $J(\theta)$ est la fonction de coût.

2. **Mise à jour des paramètres** : Les paramètres du modèle sont mis à jour itérativement dans la direction opposée au gradient, ce qui permet de descendre vers le minimum de la fonction

$$\vartheta := \vartheta - \alpha \cdot \nabla J(\vartheta)$$

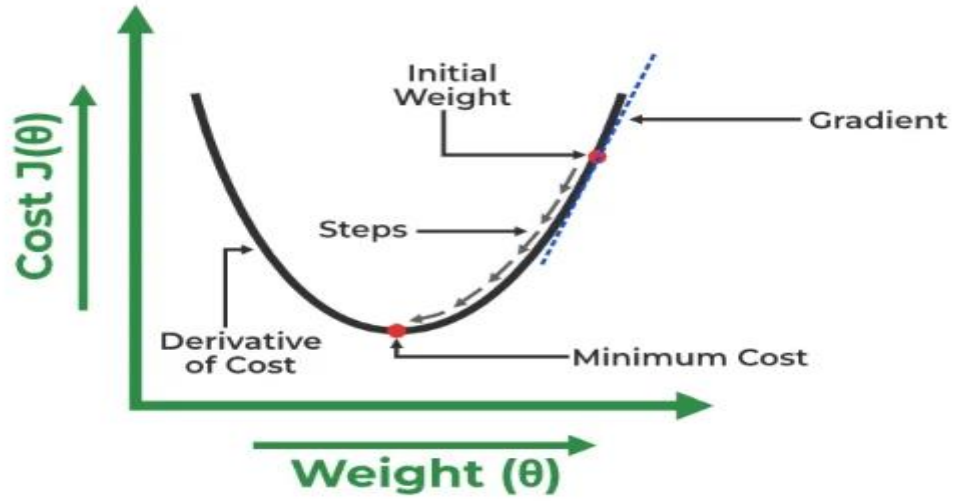
- $\vartheta$ représente les paramètres du modèle,
- $\alpha$ est le taux d'apprentissage (learning rate), qui contrôle la taille des pas lors de la descente,
- $\nabla J(\vartheta)$ est le gradient de la fonction de coût par rapport aux paramètres.

$$\theta_1 = \theta_1 - \alpha\,(J'_{\theta_1})$$
$$= \theta_1 - \alpha \left( \frac{2}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) \right)$$
$$\theta_2 = \theta_2 - \alpha\,(J'_{\theta_2})$$
$$= \theta_2 - \alpha \left( \frac{2}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) \cdot x_i \right)$$

# Descente de gradient

Le taux d'apprentissage est crucial pour la convergence de l'algorithme :

- Si $\alpha$ est trop petit, la convergence sera très lente.
- Si $\alpha$ est trop grand, l'algorithme pourrait diverger ou osciller autour du minimum.




Gradient Descent

$$\theta_1 = \theta_1 - \alpha\,(J'_{\theta_1})$$
$$= \theta_1 - \alpha\left(\frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\right)$$
$$\theta_2 = \theta_2 - \alpha\,(J'_{\theta_2})$$
$$= \theta_2 - \alpha\left(\frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\cdot x_i\right)$$

# Modèles avec plusieurs caractéristiques

- Bien que l'exemple de cette section n'utilise qu'une seule caractéristique, un modèle plus sophistiqué peut reposer sur plusieurs caractéristiques, chacune étant associée à une pondération distincte (w1,w2,…,wn).
- Par exemple, un modèle basé sur n caractéristiques peut être représenté sous la forme générale suivante :
- Dans le cas d'un modèle de prédiction de la consommation de carburant, les caractéristiques utilisées peuvent inclure :

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

- Cylindrée (x1), Accélération (x2), Nombre de cylindres (x3), Cheval-vapeur (x4)

L'équation du modèle prend alors la forme :

$$\text{Consommation} = w_0 + w_1(\text{Cylindrée}) + w_2(\text{Accélération}) + w_3(\text{Nombre de cylindres}) + w_4(\text{Cheval-vapeur})$$

# Modèlesavec plusieurs caractéristiques

Nous pouvons reformuler ce modèle sous forme matricielle en définissant une matrice de conception X qui contient les différentes puissances des observations x(i):

$$X = \begin{bmatrix} 1 & x^{(1)} & (x^{(1)})^2 & \dots & (x^{(1)})^m \\ 1 & x^{(2)} & (x^{(2)})^2 & \dots & (x^{(2)})^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{(N)} & (x^{(N)})^2 & \dots & (x^{(N)})^m \end{bmatrix}$$

où chaque ligne correspond a une observation, et chaque colonne représente une puissance de x. L'équation du problème devient alors : Xw≈y

# Modèlesavec plusieurs caractéristiques

Pour notre exemple, une régression quadratique (m=2) aura la forme:

$$\mathbf{X} = \begin{bmatrix} 0.75 & 0.86 & 1 \\ 0.01 & 0.09 & 1 \\ 0.73 & -0.85 & 1 \\ 0.76 & 0.87 & 1 \\ 0.19 & -0.44 & 1 \\ 0.18 & -0.43 & 1 \\ 1.22 & -1.10 & 1 \\ 0.16 & 0.40 & 1 \\ 0.93 & -0.96 & 1 \\ 0.03 & 0.17 & 1 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 2.49 \\ 0.83 \\ -0.25 \\ 3.10 \\ 0.87 \\ 0.02 \\ -0.12 \\ 1.81 \\ -0.83 \\ 0.43 \end{bmatrix}$$

# Modèlesavec plusieurs caractéristiques

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 0.75 & 0.01 & 0.73 & 0.76 & 0.19 & 0.18 & 1.22 & 0.16 & 0.93 & 0.03 \\ 0.86 & 0.09 & -0.85 & 0.87 & -0.44 & -0.43 & -1.10 & 0.40 & -0.96 & 0.17 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.75 & 0.86 & 1 \\ 0.01 & 0.09 & 1 \\ 0.73 & -0.85 & 1 \\ 0.76 & 0.87 & 1 \\ 0.19 & -0.44 & 1 \\ 0.18 & -0.43 & 1 \\ 1.22 & -1.10 & 1 \\ 0.16 & 0.40 & 1 \\ 0.93 & -0.96 & 1 \\ 0.03 & 0.17 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 4.11 & -1.64 & 4.95 \\ -1.64 & 4.95 & -1.39 \\ 4.95 & -1.39 & 10 \end{bmatrix}$$

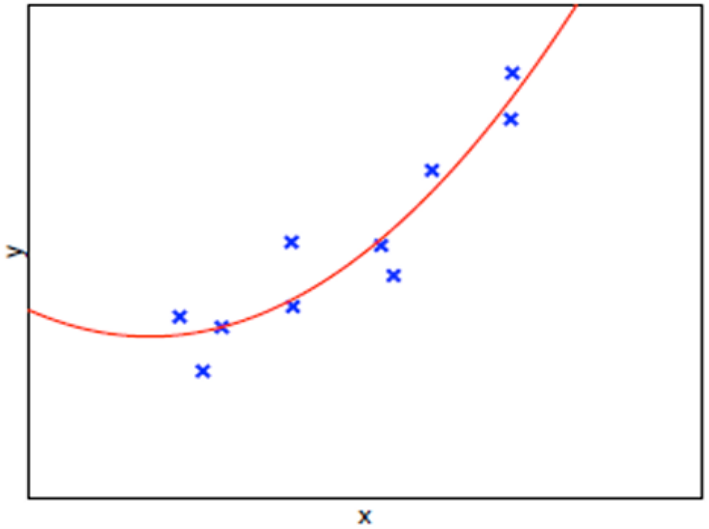
$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 0.75 & 0.01 & 0.73 & 0.76 & 0.19 & 0.18 & 1.22 & 0.16 & 0.93 & 0.03 \\ 0.86 & 0.09 & -0.85 & 0.87 & -0.44 & -0.43 & -1.10 & 0.40 & -0.96 & 0.17 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2.49 \\ 0.83 \\ -0.25 \\ 3.10 \\ 0.87 \\ 0.02 \\ -0.12 \\ 1.81 \\ -0.83 \\ 0.43 \end{bmatrix}$$

$$= \begin{bmatrix} 3.60 \\ 6.49 \\ 8.34 \end{bmatrix}$$

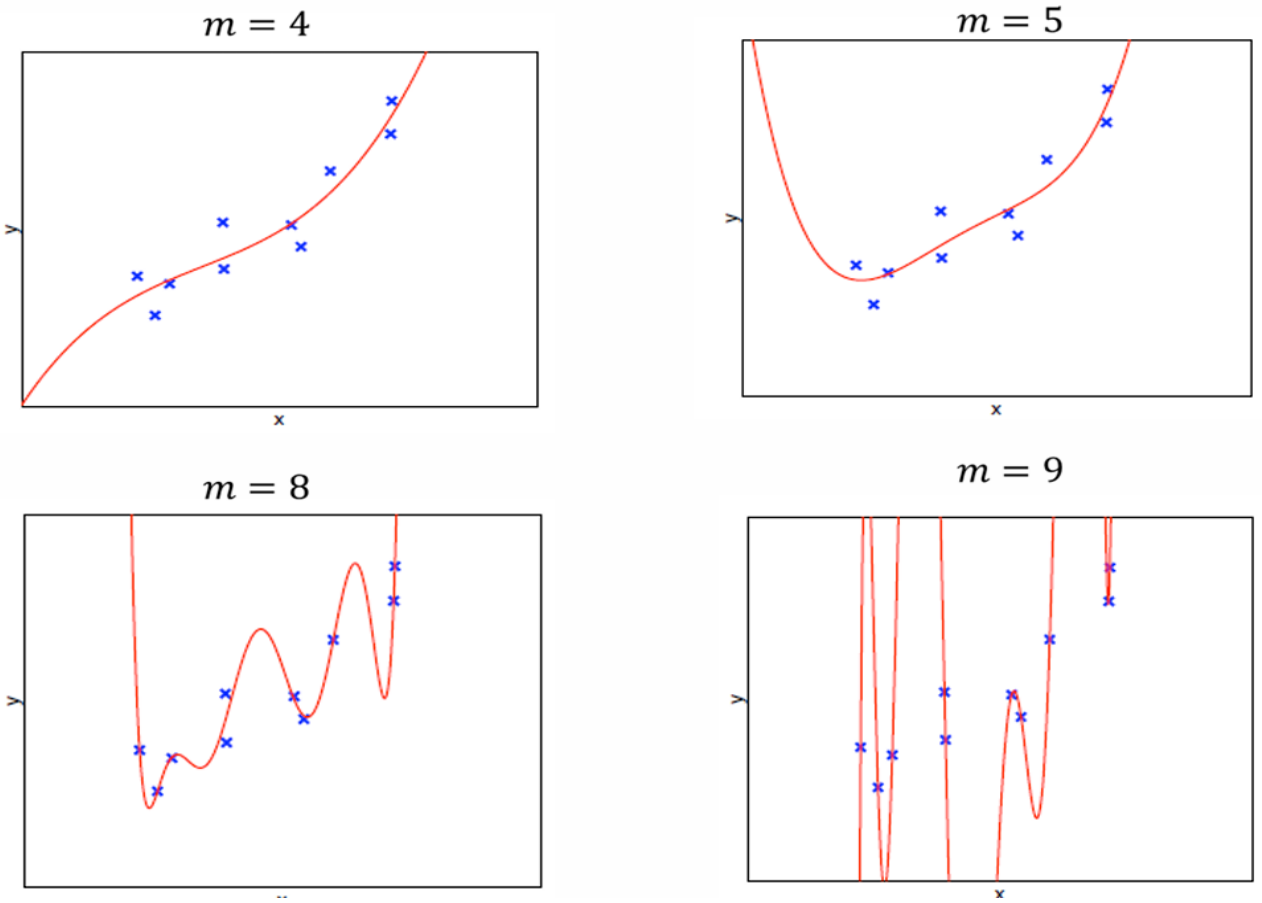
# Modèlesavec plusieurs caractéristiques

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} = \begin{bmatrix} 4.11 & -1.64 & 4.95 \\ -1.64 & 4.95 & -1.39 \\ 4.95 & -1.39 & 10 \end{bmatrix}^{-1} \begin{bmatrix} 3.60 \\ 6.49 \\ 8.34 \end{bmatrix} = \begin{bmatrix} 0.68 \\ 1.74 \\ 0.73 \end{bmatrix}$$

$$f(x) = 0.73 + 0.68x^2 + 1.74x$$



# Modèlesavec plusieurs caractéristiques



# Surapprentissage (Overfitting)

- Le surapprentissage est un défi majeur dans les techniques d’apprentissage automatique.
- Un modèle peut apprendre trop spécifiquement les données d'entraînement, obtenant de très bonnes performances sur ces données, mais échouant à bien généraliser sur de nouvelles données.
- Nous explorerons différentes méthodes tout au long du cours pour limiter ce phénomène et améliorer la capacité de généralisation des modèles.

# Théorie de la décision pour la classification

Dans un problème de classification, nous avons plusieurs classes C1,C2,…,CK et notre but est d'attribuer une entrée x à l'une de ces classes.

- **Fonction discriminante :** Elle prend une entrée x et lui attribue la classe qui lui correspond parmi K classes.
- **But :** Prédire la valeur cible y à partir de l'entrée x.

**Exemple concret :**
Imaginons une image de rayon-X . Notre tâche est de prédire si l'image montre la présence d'une maladie (classe C1) ou son absence (classe C2).

# La Théorie des Probabilités et la Décision

- La **théorie des probabilités** nous aide à traiter l'incertitude associée à un phénomène aléatoire, comme la classification d'un objet dans une certaine classe. Elle nous permet de quantifier cette incertitude et de prendre des décisions sur cette base.
- **Rôle de la probabilité :**

Elle quantifie les chances qu'un exemple x appartienne à une classe donnée $C_k$.

- Lors de la classification d'une image, la question est : **À quelle classe appartient x ?**

# La Théorie des Probabilités et la Décision

**Calcul de la Probabilité : Théorème de Bayes**

La probabilité d'une classe $C_k$ pour une donnée x est donnée par la formule de **Bayes** :

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

- $p(x|C_k)$ probabilité a posteriori, qui nous donne la probabilité que x appartienne à la classe $C_k$, après avoir vu x.
- $P(C_k)$ est la probabilité a priori d'observer la classe $C_k$.
- $p(x)$ est la probabilité de l'observation x.

# La Théorie des Probabilités et la Décision

## Diviser l'Espace des Données

- La règle minimisant l'erreur de classification consiste à diviser l'espace des données *D* en différentes **régions de décision**. Chaque région correspond à une classe donnée, et un exemple x sera classé dans la région à laquelle il appartient.
- Chaque région $R_k$ correspond à une classe $C_k$.
- Par exemple, une région $R_1$ correspond à la classe $C_1$, et ainsi de suite.

# Minimiser l'Erreur de Classification

- **Erreur de classification :**
  L'erreur se produit lorsqu'une donnée qui appartient à $C_1$ est assignée à $C_2$, ou vice versa.

- **Calcul de l'erreur :**
  L'erreur totale est la somme des probabilités des erreurs commises dans chaque région de décision :

$$\text{Erreur} = p(\mathbf{x} \in R_1, C_2) + p(\mathbf{x} \in R_2, C_1)$$
$$= \int_{R_1} p(x, C_2)\, dx + \int_{R_2} p(x, C_1)\, dx$$

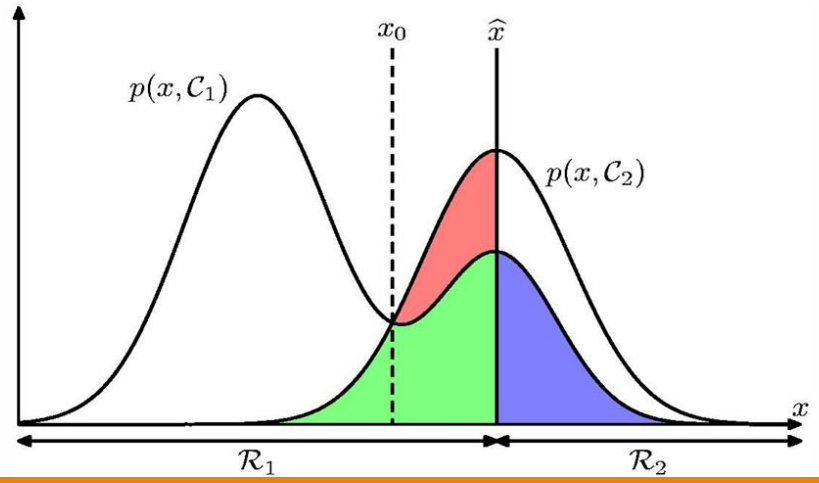
- Cette somme représente la probabilité d'attribuer une mauvaise classe à **x**.

# Règle de Décision pour Minimiser l'Erreur

- Afin de minimiser l'erreur de classification, nous devons assigner chaque exemple x à la classe $C_k$ ayant la probabilité a posteriori **la plus élevée** $p(C_k|\mathbf{x})$.

- **Simplification de l'algorithme :**

Puisque p(x) est un facteur constant entre toutes les classes, le critère de minimisation revient à sélectionner la classe $C_k$ qui maximise $p(C_k|\mathbf{x})$.



# Modèles linéaires pour la classification

# Introduction

- Pour la classification, on possède K classes {C1, C2, …, CK}. Dans la plupart des scenarios, les classes sont disjointes.
- L'espace d'entré est alors divisé en régions séparées par des frontières (ou surface) de décision.
- Dans cette partie du cours, nous considérons les modèles linéaire pour la classification, c.-à-d., la frontière de décision est une fonction linéaire de la variable d'entrée $x$.

# Modèle Linéaire de Classification

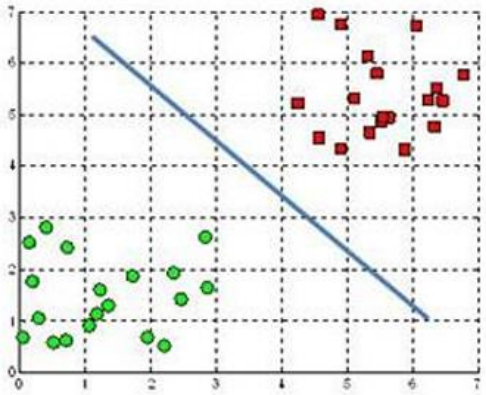
- Dans un problème de classification, l'objectif est de prendre une **entrée x** et de lui attribuer une classe parmi un ensemble de **K classes existantes**. Cette tâche est effectuée à l'aide d'une fonction discriminante qui attribue la classe appropriée à x.
- La frontière linéaire est définie dans

l'espace à $(D - 1)$ dimensions dans

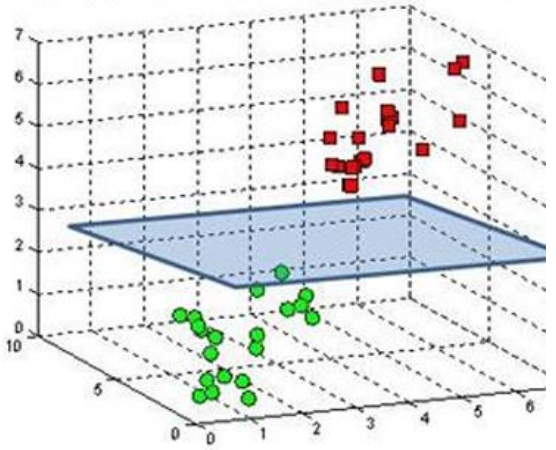
l'espace de données à $D$ dimensions.

- Pour D = 2, la frontière sera une droite.
- Pour D = 3, la frontière sera un plan.
- Pour D > 3, la frontière sera appelée hyperplan.

A hyperplane in $\mathbb{R}^2$ is a line



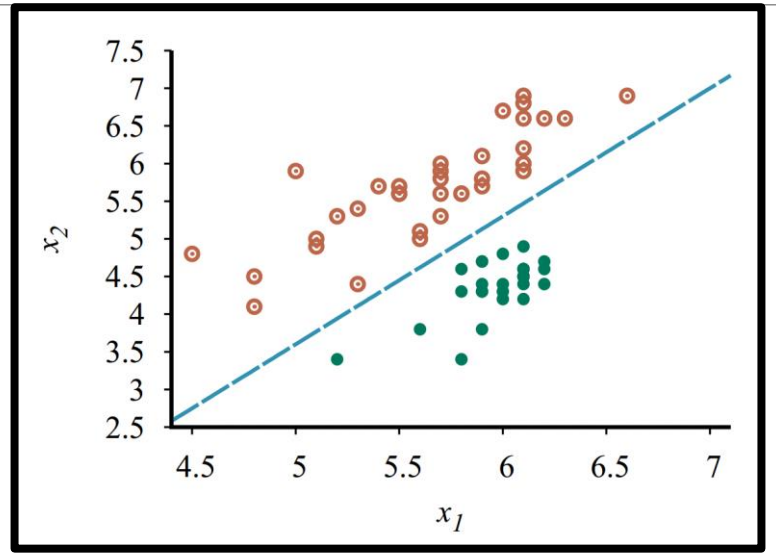
A hyperplane in $\mathbb{R}^3$ is a plane



A hyperplane in $\mathbb{R}^n$ is an $n$-1 dimensional subspace

# Exemple

| Sample $x_i$ | $x_{i1}$ | $x_{i2}$ | $y_i$ |
|---|---|---|---|
| $x_1$ | 4.5 | 4.7 | 1 |
| $x_2$ | 4.7 | 4.1 | 1 |
| $x_3$ | 4.7 | 4.9 | 1 |
| $x_4$ | 5 | 5.9 | 1 |
| $x_5$ | 5.1 | 4.9 | 1 |
| $x_6$ | 5.1 | 5 | 1 |
| $x_7$ | 5.2 | 3.5 | 0 |
| .. | .. | .. | .. |




*Équation de la droite, à peu près:* $-2.6 - 0.2x_1 + x_2 = 0$

*En général:* $w_0 + w_1 x_1 + w_2 x_2 = 0$

*Si* $w_0 + w_1 x_1 + w_2 x_2 >= 0$ alors $y = 1$ sinon $y = 0$

# Modèle Linéaire de Classification

**Régression vs. Classification**

- Dans **la régression**, la variable cible y prend des valeurs réelles continues.
- Dans **la classification**, la variable cible y représente des classes discrètes. Pour un problème de classification binaire, y peut prendre les valeurs 1 ou 0 :
  - y=1 pour la classe C1
  - y=0 pour la classe C2

# Frontière de décision linéaire

**Codage pour K>2**

Pour des problèmes de classification avec plus de deux classes (K>2), un **codage vectoriel** est utilisé.

- Soit $y=(y_1,y_2,\ldots,y_K)$, où :
  - Si $x \in C_k$, alors $y_k=1$
  - Pour toutes les autres classes $j \neq k$ , $y_j=0$

Cela permet de représenter de manière binaire la classe à laquelle appartient x.

# Exemple de Codage avec un Vecteur

Prenons un exemple où **K=5**, pour un problème de classification avec cinq classes différentes. Les données et leur codage respectif seraient les suivants :

| Donnée ($x$) | Classe | Codage ($y$) |
|---|---|---|
| | $C_1$ | $(1, 0, 0, 0, 0)$ |
| | $C_2$ | $(0, 1, 0, 0, 0)$ |
| | $C_3$ | $(0, 0, 1, 0, 0)$ |
| | $C_4$ | $(0, 0, 0, 1, 0)$ |
| | $C_5$ | $(0, 0, 0, 0, 1)$ |

# Frontière de décision linéaire

- Dans **la classification**, l'objectif est de prédire une étiquette de classe, c'est-à-dire de déterminer à quelle **classe** appartient l'exemple donné **x**. Au lieu d'une prédiction continue, comme en régression, on cherche à prédire les **probabilités a posteriori** des classes, c'est-à-dire des valeurs comprises entre 0 et 1.

- Pour adapter le modèle linéaire de régression à la classification, on peut généraliser la fonction de prédiction en **conservant la forme linéaire**, mais en appliquant une **fonction non-linéaire** pour restreindre la sortie dans l'intervalle [0,1]. Cela donne :
  - $g(x)=g(w^Tx+w_0)$ où g est une fonction **non-linéaire**, telle que la fonction sigmoïde, qui **transforme la sortie linéaire en une probabilité**.

# Frontière de décision linéaire

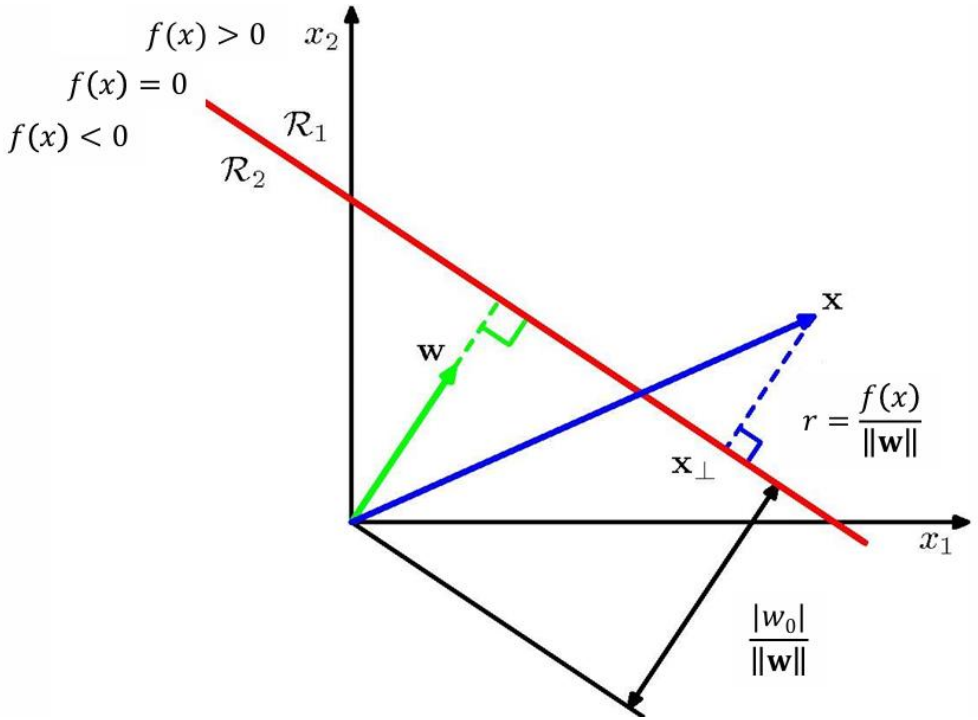
- **Fonction d'activation** permet de transformer la sortie d'une combinaison linéaire des variables d'entrée en une probabilité ou en une valeur discrète. Son inverse est appelé **fonction de lien** en statistique.
- Les **surfaces de décision(**frontières**)** sont des lignes ou des hyperplans définis par l'équation $w^T x+w_0$=constante.
- Les modèles qui reposent sur cette forme d'équation sont appelés **modèles linéaires généralisés**. Ces modèles peuvent inclure des fonctions d'activation non linéaires pour la classification, comme la fonction sigmoïde ou la tangente hyperbolique.

# *Cas de classification binaire (K = 2)*

- ✓ Dans une **classification binaire**, la fonction discriminante est une fonction linéaire $w^Tx+w_0$, qui permet de séparer les deux classes C1 et C2.
- ✓ La **frontière de décision** est définie par les points où f(x)=0, et elle sépare les deux classes.
- ✓ Le vecteur w est **orthogonal** à cette frontière de décision, ce qui signifie que la direction du vecteur w détermine la position de la frontière.

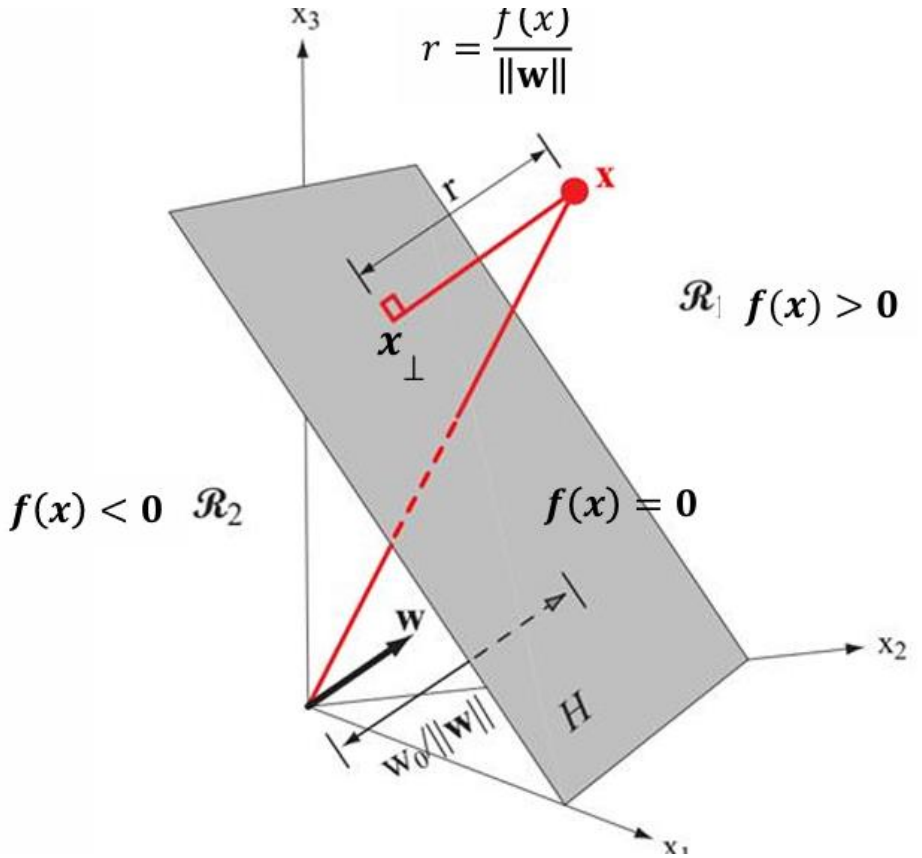
# *Cas de classification binaire (K = 2)*

Soit un problème de classification à 2 classes (classification binaire). Dans le cas de d = 2, on a la géométrie suivante:



# *Cas de classification binaire (K = 2)*

Dans le cas de d = 3, on a la géométrie suivante:

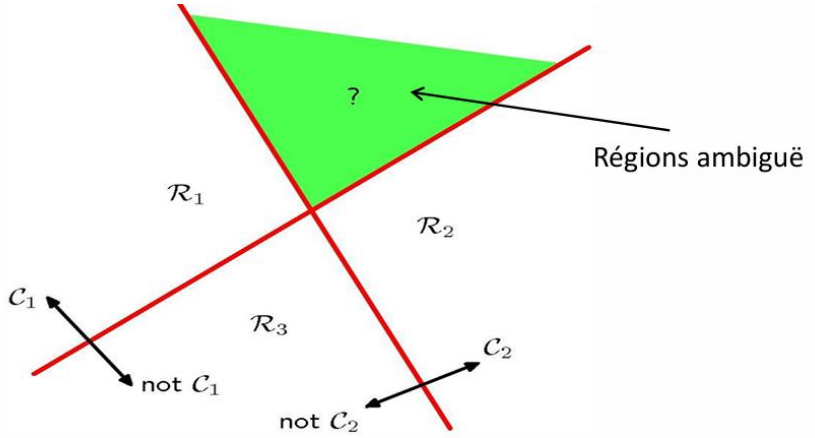


# Cas de classification multiple (K > 2)

- Une des façons de construire un classificateur à K classes est de combiner plusieurs classificateurs binaires.

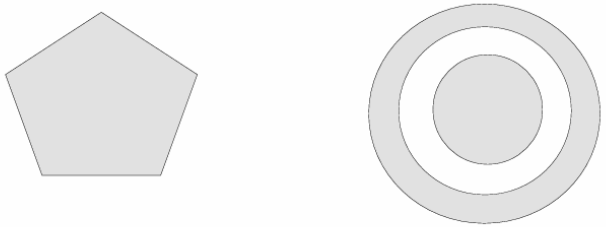
• Soit (K-1) classificateurs binaires chacun séparant une des classes $Ck$, k = 1…, K-1 (classification un-versus-tous).

Une autre alternative est de construire K(K-1)/2 classificateurs binaires, un pour chaque deux classes (classification un versus-un).



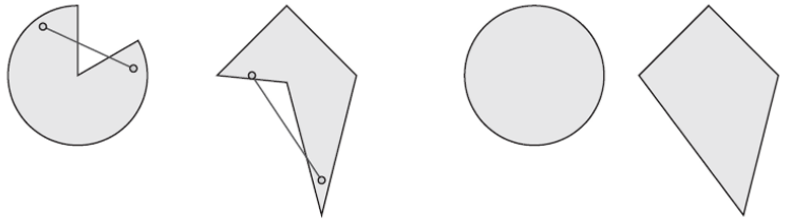
# Exemples de régions connexes et convexes

**Connexe** | **Non connexe**



E est **connexe** ⇒ E n'est pas la réunion de deux ouverts non vides disjoints.

**Connexe mais non convexe** | **Convexe**



# Classification par les Moindres Carrés

- Comme on a fait pour la régression, on augmente d'abord la matrice de données $\mathbf{X}$ d'une colonne égale à **1**.
- On définit pour chaque donnée $x^{(i)}$ un vecteur colonne $y^{(i)} = \left(y_1{}^{(i)}, y_2{}^{(i)}, \cdots, y_K{}^{(i)}\right)^{\mathrm{T}}$ et une matrice $\mathbf{Y}$ dont la $i$-ième ligne contient $y^{(i)^T}$.
- Soit la matrice $\widetilde{\mathbf{W}}$ dont la $k$-ième colonne contient $\widetilde{\mathbf{w}}_k$.
- **L'erreur des moindres carrées** est alors donnée par:

$$E(\widetilde{\mathbf{W}}) = \frac{1}{2} Tr\left\{\left(\mathbf{X}\widetilde{\mathbf{W}} - \mathbf{Y}\right)^T \left(\mathbf{X}\widetilde{\mathbf{W}} - \mathbf{Y}\right)\right\}$$

# Classification par les Moindres Carrés

- En calculant la dérivée de $E$ par rapport à $\mathbf{W}$, on obtient:

$$\widetilde{\mathbf{W}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{X}'\mathbf{Y}$$

- On appelle $\mathbf{X}'$ **le pseudo-inverse** de la matrice $\mathbf{X}$.

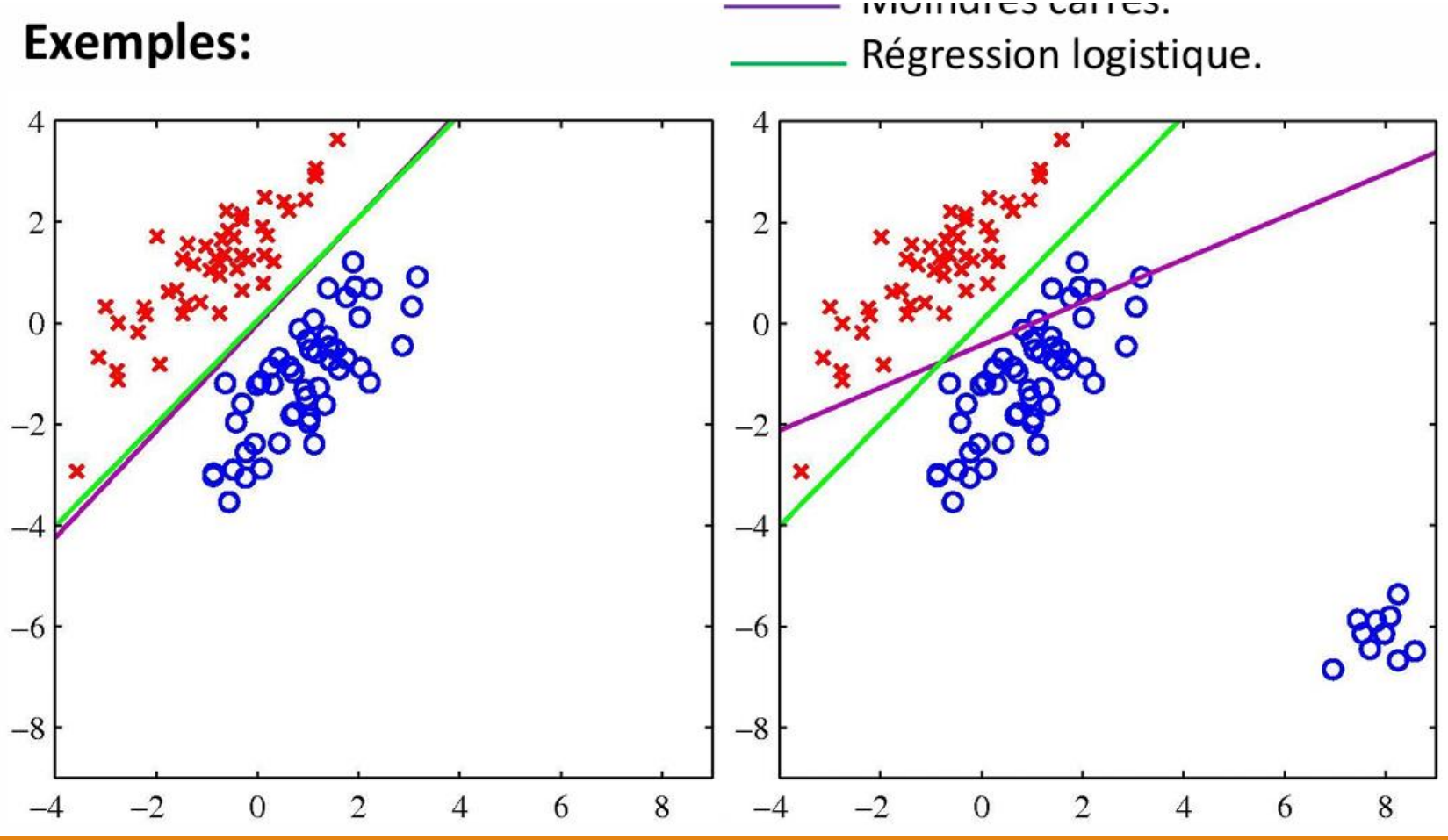
- La fonction discriminante $\mathbf{f}$ est alors donnée par:

$$\mathbf{f}(\mathbf{x}) = \widetilde{\mathbf{W}}^T\mathbf{x} = \mathbf{Y}^T(\mathbf{X}')^T\mathbf{x}$$

- Notons que les éléments de $\mathbf{f}$ auront la somme égale à 1. Mais les entrées de $\mathbf{f}$ ne sont pas pour autant des probabilités.

# Classification par les Moindres Carrés

**Exemples:**



# Classification par les Moindres Carrés

Bien que la méthode des moindres carrés fournisse une solution exacte pour la classification, elle présente certaines limitations :

- Elle fonctionne bien lorsque les classes sont séparables et que les données suivent des lois gaussiennes.
- Elle n'est pas robuste aux données aberrantes.
- Les éléments de **f** ont une somme égale à 1, mais ils ne sont pas nécessairement des probabilités.

# Référence


- The StatQuest Illustrated Guide to Machine Learning!!!, By Josh Starmer, Ph.D
- Studies in Computational Intelligence Volume 498 Series Editor J. Kacprzyk, Warsaw, Poland
- Machine Learning A Probabilistic Perspective Kevin P. Murphy
- Christopher M. Bishop Pattern Recognition and Machine Learning
- Cours Aissa Boulmerka, Associate Professor, The National School of Artificial Intelligence, Algeria