Machine Learning

Montaigne 2023-2024

– mpi23@arrtes.net –

Qu'est le Machine Learning (ML) ?

- **Machine Learning** = Apprentissage automatique
- Champ d'étude qui vise à concevoir des algorithmes et des programmes susceptibles de réaliser des tâches sans avoir été explicitement programmés pour le faire.
- Algorithmes fondés sur des modèles mathématiques de traitements de données (**data training**) en vue de faire des **prédictions** ou de prendre des **décisions**.

Qu'est le Machine Learning (ML) ?

Deux phases essentielles.

- **Apprenstissage** (ou entraînement) : conception d'un modèle à partir de données. Préalable à l'utilisation pratique.
- **Exploitation** : mise en œuvre du modèle en vue de produire des résultats. Possibilité d'évolution du système par un apprentissage en relation avec les résultats produits et une évaluation de leur qualité.

Qu'est le Machine Learning (ML) ?

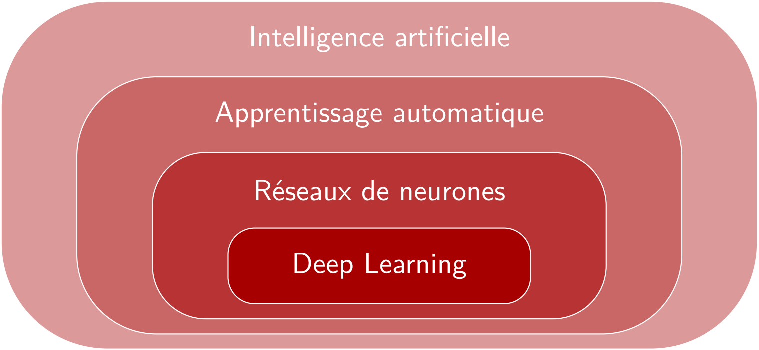
- **Apprentissage supervisé** - À partir de données d'entrée et de sortie, un modèle de classification est proposé en vue d'établir un lien entre l'entrée et la sortie.
- **Apprentissage non supervisé** - À partir de données d'entrée seules, un modèle doit découvrir une structure cachée produisant des données de sortie.
- **Apprentissage par renforcement** - Le modèle peut évoluer au gré de ses interactions avec son environnement (feedback).
- **Autres** - Les fontières entre ces approches ne sont pas toujours nettes. D'autres approches complètent les précédentes : apprentissage par transfert, rédution de la dimension, meta-learning.

Objectifs de ce mini-exposé

Apprenstissage supevisé

- Problèmes de **régression** : régression linéaire et non-linéaire
- Descente de gradient
- Problèmes de **classification** : régression logistique

Qu'est le Machine Learning (ML) ?

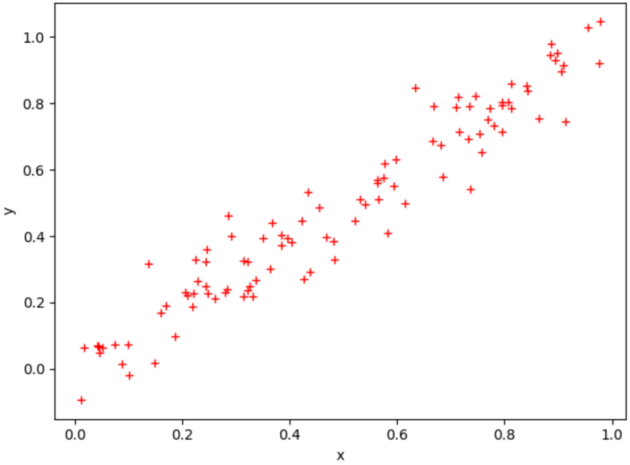


Régression linéaire

Régression linéaire

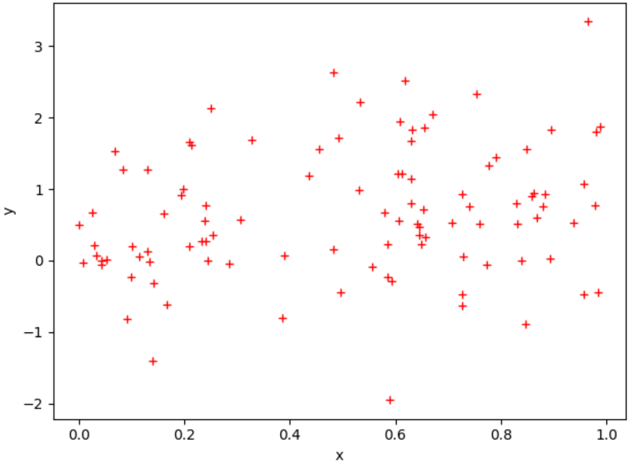
- **Objectif** - Étant donné un jeu de données, **prédire** un résultat **quantitatif**.
- **Données** - Pour illustrer notre propos, $m \in \mathbb{N}^*$ couples de valeurs numériques sont données : $(x_i, y_i)_{0 \leqslant i \leqslant m-1}$.
- **Question** - Existe-t-il une **relation linéaire** entre les données d'entrée x_i et les données de sortie y_i ?

Régression linéaire



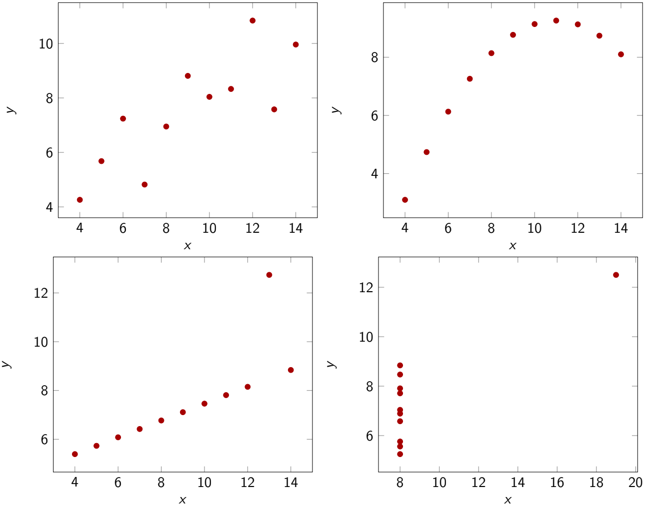
Relation affine ?

Régression linéaire



Relation affine ?

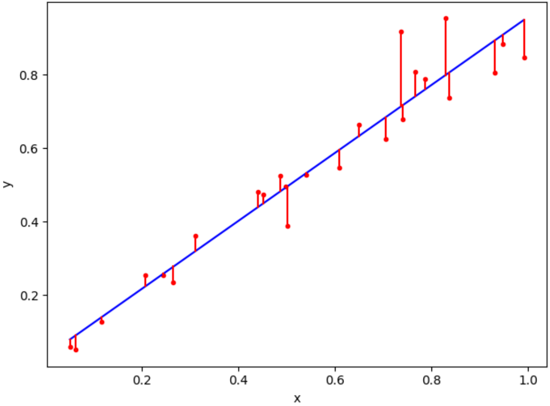
Régression linéaire



Quartet d'Anscombe - $y = 3 + 0{,}5x$ - $r = 0{,}816$

`https://fr.wikipedia.org/wiki/Quartet_d'Anscombe`

Méthode des moindres carrés

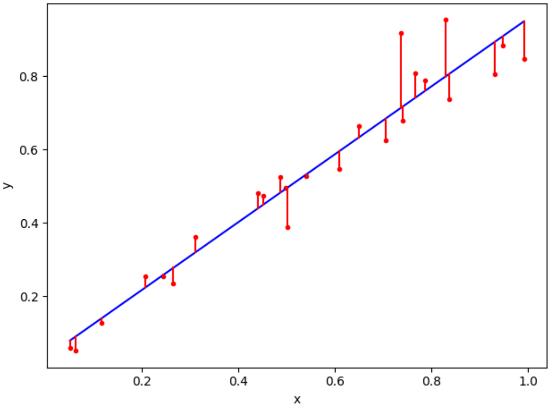


Recherche d'une droite d'équation :

$$y = ax + b$$

$$a = ? \qquad b = ?$$

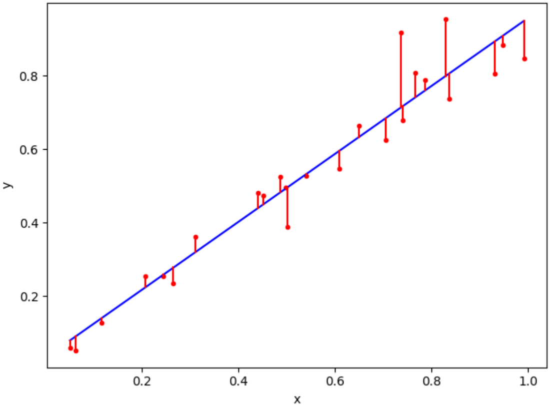
Méthode des moindres carrés



Écarts verticaux entre données (x_i, y_i) et estimations $(x_i, ax_i + b)$:

$$ax_i + b - y_i$$

Méthode des moindres carrés



Fonction de coût

$$J(a, b) = \frac{1}{2m} \sum_{i=0}^{m-1} (ax_i + b - y_i)^2$$

Méthode des moindres carrés

Idée de la **méthode des moindres carrés** : trouver a et b qui minimisent J.

$$\partial_a J(a,b) = 0 \quad \Longrightarrow \quad \sum_{i=0}^{m-1} x_i(ax_i + b - y_i) = 0$$

$$\partial_b J(a,b) = 0 \quad \Longrightarrow \quad \sum_{i=0}^{m-1} (ax_i + b - y_i) = 0$$

$$J(a,b) = \frac{1}{2m} \sum_{i=0}^{m-1} (ax_i + b - y_i)^2$$

Méthode des moindres carrés

Notations

$$\overline{x} = \frac{1}{m} \sum_{i=0}^{m-1} x_i \qquad \overline{y} = \frac{1}{m} \sum_{i=0}^{m-1} y_i$$

$$\overline{x^2} = \frac{1}{m} \sum_{i=0}^{m-1} x_i^2 \qquad \overline{xy} = \frac{1}{m} \sum_{i=0}^{m-1} x_i y_i$$

Système à résoudre

$$a\overline{x^2} + b\overline{x} - \overline{xy} = 0 \qquad a\overline{x} + b - \overline{y} = 0$$

Solution

$$a = \frac{\overline{xy} - \overline{x}\ \overline{y}}{\overline{x^2} - \overline{x}^2} \qquad b = \overline{y} - a\overline{x}$$

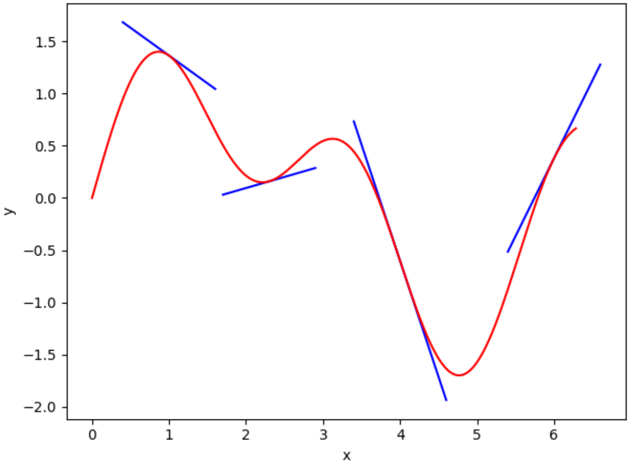
Méthode des moindres carrés

Code sur machine

- Générer un jeu de données pseudo-aléatoires
- Calculer a et b.
- Tracer le nuage de points et la droite d'ajustement
- Prédire des résultats.

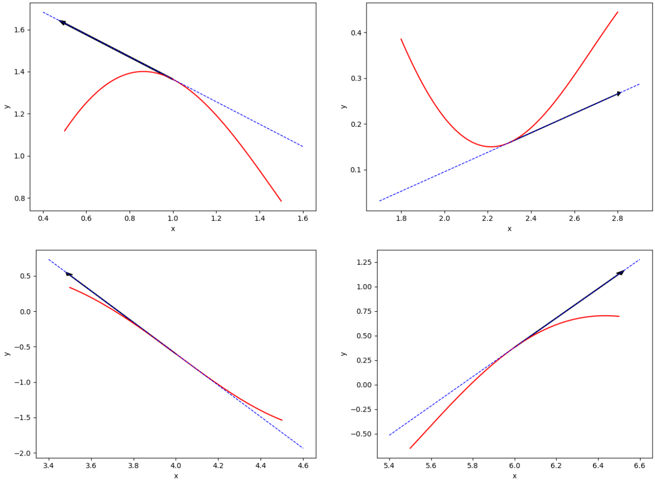
Descente de gradient

Gradient



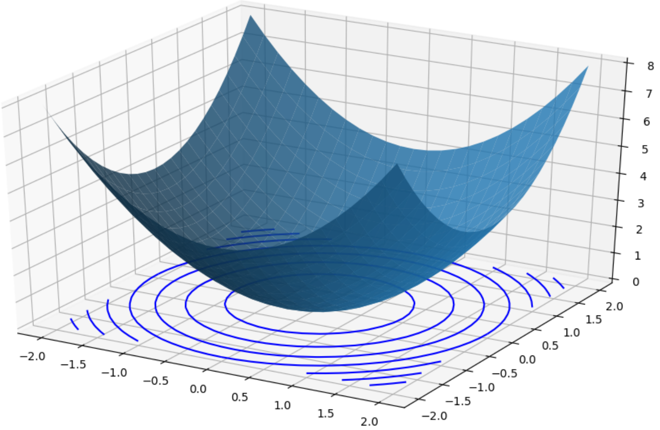
Tangentes à une courbe en 4 de ses points

Gradient



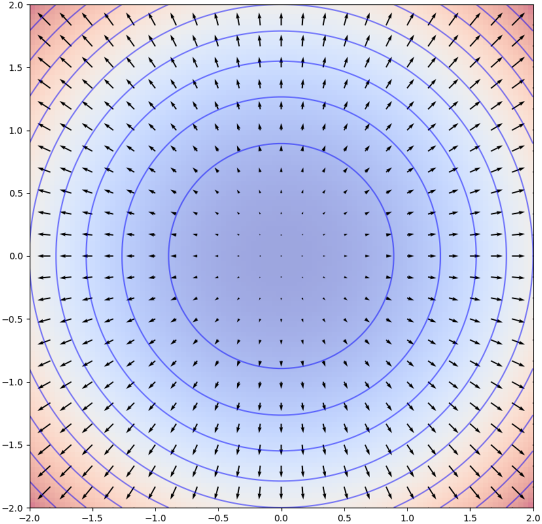
Sens croissants des pentes

Gradient



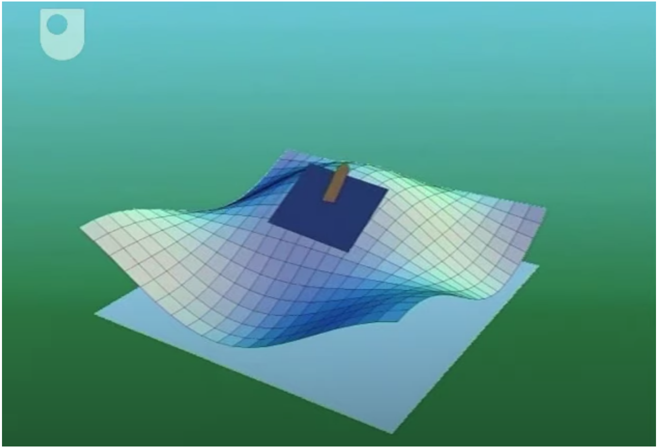
Surface $z = x^2 + y^2$ et lignes de niveau dans le plan $z = 0$

Gradien



Gradient : champ de vecteurs

Gradient



Open University (`https://youtu.be/ynzRyIL2atU`)

Gradient

Fonction de m variables

$$F : \mathbb{R}^m \to \mathbb{R} \qquad \nabla F = \begin{pmatrix} \partial_0 F \\ \partial_1 F \\ \vdots \\ \partial_{m-1} f \end{pmatrix}$$

Exemple (coordonnées cartésiennes)

$$F : \begin{cases} \mathbb{R}^2 \to \mathbb{R} \\ (x, y) \to (x^2 - 3y)^2 \end{cases} \qquad \nabla F(x_0, y_0) = \begin{pmatrix} \partial_0 F(x_0, y_0) = 4x_0(x_0^2 - 3y_0) \\ \partial_1 F(x_0, y_0) = -6(x_0^2 - 3y_0) \end{pmatrix}$$

Plus grande pente croissante en (x_0, y_0) dirigée suivant le vecteur :

$$(4x_0(x_0^2 - 3y_0), -6(x_0^2 - 3y_0))$$

Descente de gradient

- **Objectif :** minimiser une fonction.
- **Idée générale :** partir d'un point d'une courbe ou d'une surface et suivre la plus grande pente pour atteindre un minimum.

Descente de gradient

Algorithme du gradient ou descente de gradient.

- Dans $\mathbb{R}^m$, soit une surface définie par un ensemble de points $(X, F(X))$ avec $X = (x_0, \ldots, x_{m-1})$ et $F : \mathbb{R}^m \to \mathbb{R}$ donnée.
- Choisir un point initial $X_0 = (x_{0,0}, x_{1,0}, \ldots, x_{m-1,0})$ sur la surface définie par le point $(X_0, F(X_0))$.
- Calculer une suite d'itérés $X_1, X_2, \ldots$ jusqu'à ce qu'une condition d'arrêt soit vérifiée.

Descente de gradient

- Choisir un seuil $\varepsilon > 0$ pour la **condition d'arrêt** et un paramètre (**taux d'apprentissage**) $\alpha > 0$.
- Calculer la direction de la **plus grande pente descendante**

$$-\nabla F(X_k)$$

- Choisir un **point initial** $X_0 = (x_{0,0}, x_1, 0, \dots, x_{m-1,0})$.
- Tester la condition d'arrêt

$$\| \nabla F(X_k) \| > \varepsilon \qquad \text{ou} \qquad \| \nabla F(X_k) \| \leqslant \varepsilon$$

- Si $\| \nabla F(X_k) \| > \varepsilon$, calculer l'itéré :

$$X_{k+1} = X_k - \alpha \nabla F(X_k)$$

Descente de gradient

- Si la fonction F est **strictement convexe**, la **convergence** de l'algorithme est assurée.
- Dans le cas général, la convergence n'est pas assurée.
- S'il y a **convergence**, elle peut être **lente** et nécessiter **beaucoup d'itérations**.
- La **valeur de** α influe sur le comportement de l'algorithme.
- Le choix d'un pas α constant n'est pas toujours pertinent. Il existe des méthodes à **pas adaptatif**.

Code sur machine

- Illustrer la méthode de descente de gradient sur une fonction $F : \mathbb{R} \to \mathbb{R}$.
- Observer le comportement de la suite des itérés en fonction du pas α.
- Déterminer le minimum de F dans un intervalle donné.

Descente de gradient et régression linéaire

Code sur machine

- Appliquer l'algorithme de descente de gradient à la détermination des paramètres a et b.
- Illustrer graphiquement les résultats obtenus.
- Prédire des résultats.

Classification binaire

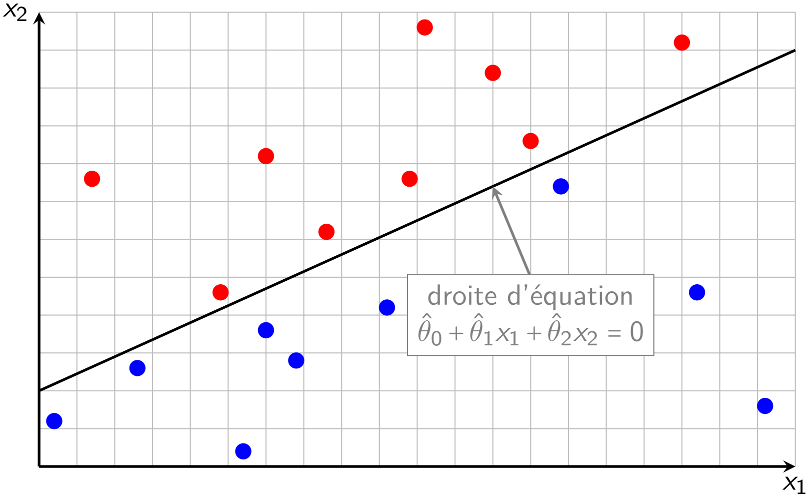
Classification binaire

- Dans cet exposé, toutes les situations abordées traitent de **points dans le plan**.
- Ces points sont désignés par des **couples** $(x_{1,i}, x_{2,i})_{1 \leqslant i \leqslant m}$, m étant un entier naturel non nul.
- Ces points sont **localisés dans deux régions du plan**.

Classification binaire

Peut-on déterminer une frontière entre ces deux régions ?

Classification binaire



Classification binaire

Peut-on déterminer une frontière entre ces deux régions ?

Répondre à cette question, c'est trouver un triplet $\left(\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2\right)$ qui permette la définition d'une **séparatrice** (droite dans l'exemple).

Comment trouver un tel triplet $\left(\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2\right)$?

Observations

- Conservons l'exemple illustratif précédent pour lequel un triplet $\left(\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2\right)$ définit une séparatrice par l'équation :

$$\hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_2 = 0$$

- Soit (x_1, x_2) un point quelconque du plan.
- Introduisons les notations :

$$X = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \qquad \hat{\Theta} = \begin{pmatrix} \hat{\theta}_0 \\ \hat{\theta}_1 \\ \hat{\theta}_2 \end{pmatrix}$$

de sorte que :

$$X^T \cdot \hat{\Theta} = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_2$$

Observations

- Si $X^T \cdot \hat{\Theta} = 0$ alors (x_1, x_2) est un **point sur la droite**.
- Si $X^T \cdot \hat{\Theta} > 0$ alors (x_1, x_2) est d'**un côté de la droite**.
- Si $X^T \cdot \hat{\Theta} < 0$ alors (x_1, x_2) est de **l'autre côté de la droite**.

Introduisons un paramètre y pouvant prendre les valeurs 0 ou 1 et adoptons la **convention** suivante.

- $y = 0$ si $X^T \cdot \hat{\Theta} < 0$.
- $y = 1$ si $X^T \cdot \hat{\Theta} \geqslant 0$.

Classification binaire

- Les données initiales $(x_{1,i}, x_{2,i})_{1 \leqslant i \leqslant m}$ du problème peuvent ainsi être classées en deux catégories.
- Posons :

$$\forall\, i \in [\![1, m]\!] \qquad X_i = \begin{pmatrix} 1 \\ x_{1,i} \\ x_{2,i} \end{pmatrix}$$

- Si $X_i^T \cdot \hat{\Theta} < 0$, on peut lui associer la valeur $y_i = 0$.
- Si $X^T \cdot \hat{\Theta} \geqslant 0$, on peut lui associer la valeur $y_i = 1$.

Classification binaire

On peut à présent reformuler le problème.

Trouver un triplet $(\theta_0, \theta_1, \theta_2)$ tel que pour tous les indices $i \in [\![1, m]\!]$ pour lesquels :

- $y_i = 0$, on ait $X_i^T \cdot \Theta < 0$;
- $y_i = 1$, on ait $X_i^T \cdot \Theta \geqslant 0$.

Idée : construire un **algorithme itératif**.

$$\Theta_0 \to \Theta_1 \to \Theta_2 \to \cdots \to \Theta_{\text{fin}} \approx \hat{\Theta}$$

Vers une fonction de coût

Étant donnée une valeur Θ_k, l'algorithme doit permettre la construction d'une **fonction de coût** qui **pénalise les situations défavorables**.

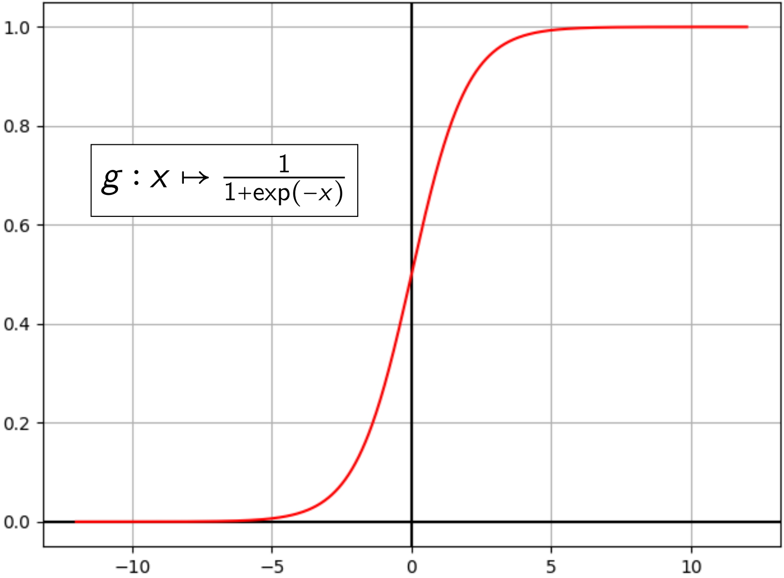
- $y_i = 0$ avec $X_i^T \cdot \Theta_k \geqslant 0$.
- $y_i = 1$, on ait $X_i^T \cdot \Theta_k < 0$.

Vers une fonction de coût

Une fonction particulière peut aider dans cette recherche : la **fonction logistique** (sigmoïde).

$$g : \begin{cases} \mathbb{R} \to [0, 1] \\ x \mapsto \dfrac{1}{1 + \exp(-x)} \end{cases}$$

Vers une fonction de coût



Vers une fonction de coût

Une fonction particulière peut aider dans cette recherche : la **fonction logistique** (sigmoïde).

$$g : \begin{cases} \mathbb{R} \to [0, 1] \\ x \mapsto \dfrac{1}{1 + \exp(-x)} \end{cases}$$

On remarque :

- si $X_i^T \cdot \Theta_k < 0$ alors $g\left(X_i^T \cdot \Theta_k\right) < 1/2$;
- si $X_i^T \cdot \Theta_k \geqslant 0$ alors $g\left(X_i^T \cdot \Theta_k\right) \geqslant 1/2$.

Vers une fonction de coût

Observons le **comportement de** $\Theta_k \mapsto -\ln\left[g\left(X_i^T \cdot \Theta_k\right)\right]$.

- Si $X_i^T \cdot \Theta_k \to -\infty$ alors :
  - $g\left(X_i^T \cdot \Theta_k\right) \to 0$
  - $-\ln\left[g\left(X_i^T \cdot \Theta_k\right)\right] \to +\infty$.
- Si $X_i^T \cdot \Theta_k \to +\infty$ alors :
  - $g\left(X_i^T \cdot \Theta_k\right) \to 1$
  - $-\ln\left[g\left(X_i^T \cdot \Theta_k\right)\right] \to 0$.

Ainsi, si $y = 1$, la fonction :

$$\Theta_k \mapsto -\ln\left[g\left(X_i^T \cdot \Theta_k\right)\right]$$

est une **candidate à la fonction de coût**.

Vers une fonction de coût

Observons le **comportement de** $\Theta_k \mapsto -\ln\left[1 - g\left(X_i^T \cdot \Theta_k\right)\right]$.

- Si $X_i^T \cdot \Theta_k \to -\infty$ alors :
  - $1 - g\left(X_i^T \cdot \Theta_k\right) \to 1$
  - $-\ln\left[1 - g\left(X_i^T \cdot \Theta_k\right)\right] \to 0$.
- Si $X_i^T \cdot \Theta_k \to +\infty$ alors :
  - $1 - g\left(X_i^T \cdot \Theta_k\right) \to 0$
  - $-\ln\left[1 - g\left(X_i^T \cdot \Theta_k\right)\right] \to +\infty$.

Ainsi, si $y = 0$, la fonction :

$$\Theta_k \mapsto -\ln\left[1 - g\left(X_i^T \cdot \Theta_k\right)\right]$$

est une **candidate à la fonction de coût**.

Vers une fonction de coût

Ces résultats peuvent être rassemblés sous la forme suivante.

$$J(\Theta_k)_i = -y_i \times \ln\left[g\left(X_i^T \cdot \Theta_k\right)\right] - (1 - y_i) \times \ln\left[1 - g\left(X_i^T \cdot \Theta_k\right)\right]$$

En prenant en compte tous les points, avec un facteur $1/m$, on définit la **fonction de coût** J :

$$J(\Theta_k) = -\frac{1}{m}\sum_{i=1}^{m}\left(y_i \times \ln\left[g\left(X_i^T \cdot \Theta_k\right)\right] + (1 - y_i) \times \ln\left[1 - g\left(X_i^T \cdot \Theta_k\right)\right]\right)$$

Vers une fonction de coût

- La fonction J **pénalise les cas défavorables**.
- La fonction J est **convexe**.
- Sa **justification** ainsi que celle de la fonction logistique est possible dans un cadre **probabiliste**.
- On peut utiliser l'**algorithme de descente de gradient pour approcher** $\hat{\Theta}$.

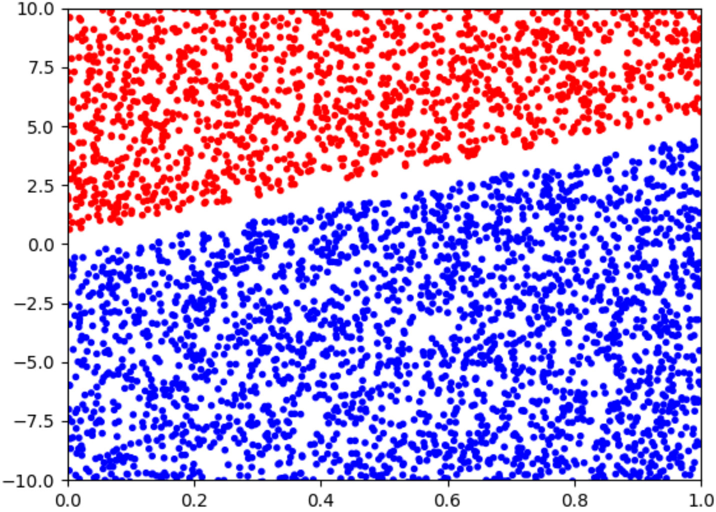
Modèle linéaire 2D

Régression logistique - Modèle linéaire 2D

On considère un **jeu de m données** $(x_{1,i}, x_{2,i}, y_i)_{1 \leqslant i \leqslant m}$.

- $(x_{1,i}, x_{2,i})_{1 \leqslant i \leqslant m}$ désignent **des points du plan**.
- Pour tout $1 \leqslant i \leqslant m$, $y_i = 0$ ou 1 précise la **classe** à laquelle chaque point appartient.

Régression logistique - Modèle linéaire 2D



Régression logistique - Modèle linéaire 2D

$$X = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,m} \\ x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,m} \end{pmatrix} \in \mathcal{M}_{3,m}(\mathbb{R})$$

$$\Theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix} \in \mathcal{M}_{3,1}(\mathbb{R})$$

$$X^T\Theta = \begin{pmatrix} \theta_0 + \theta_1 x_{1,1} + \theta_2 x_{2,1} \\ \theta_0 + \theta_1 x_{1,2} + \theta_2 x_{2,2} \\ \theta_0 + \theta_1 x_{1,3} + \theta_2 x_{2,3} \\ \vdots \\ \theta_0 + \theta_1 x_{1,m} + \theta_2 x_{2,m} \end{pmatrix} \in \mathcal{M}_{m,1}(\mathbb{R})$$

Régression logistique - Modèle linéaire 2D

$$G(X,\Theta) = \begin{pmatrix} g(\theta_0 + \theta_1 x_{1,1} + \theta_2 x_{2,1}) \\ g(\theta_0 + \theta_1 x_{1,2} + \theta_2 x_{2,2}) \\ g(\theta_0 + \theta_1 x_{1,3} + \theta_2 x_{2,3}) \\ \vdots \\ g(\theta_0 + \theta_1 x_{1,m} + \theta_2 x_{2,m}) \end{pmatrix} \in \mathcal{M}_{m,1}(\mathbb{R})$$

$$\mathbb{I}_m = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathcal{M}_{m,1}(\mathbb{R})$$

$$L : \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_m \end{pmatrix} \in \mathcal{M}_{m,1}(\mathbb{R}) \mapsto \begin{pmatrix} \ln U_1 \\ \ln U_2 \\ \vdots \\ \ln U_m \end{pmatrix} \in \mathcal{M}_{m,1}(\mathbb{R})$$

Régression logistique - Modèle linéaire 2D

- **Fonction de coût** (écriture vectorielle)

$$J(\Theta) = -\frac{1}{m}\left[Y^T \cdot L(G(X,\Theta)) + (\mathbb{I}_m - Y)^T \cdot L\left(\mathbb{I}_m - G(X,\Theta)\right)\right]$$

- **Gradient** de la fonction de coût

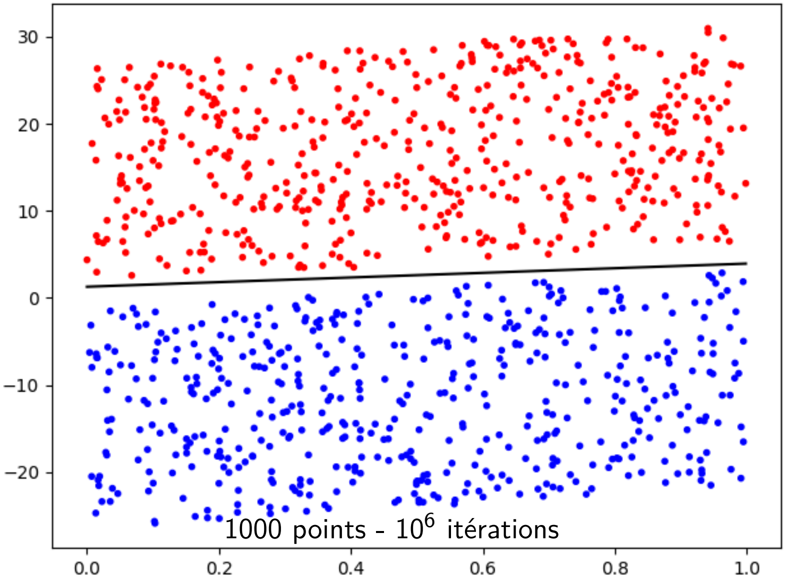
$$\nabla J(\Theta) = \frac{1}{m} X^T \cdot (G(X,\Theta) - Y) \in \mathcal{M}_{3,m}(\mathbb{R})$$

Régression logistique - Modèle linéaire 2D

Algorithme de descente de gradient

- Choisir un seuil $\varepsilon > 0$ et un taux d'apprentissage $\alpha > 0$.
- Choisir un vecteur initial Θ.
- Tant que $\parallel \nabla J(\Theta) \parallel > \varepsilon$:
  - calculer $\nabla J(\Theta)$;
  - $\Theta \leftarrow \Theta - \alpha \nabla J(\Theta)$

Régression logistique - Modèle linéaire 2D



Régression logistique - Modèle linéaire 2D

Code sur machine

- Générer un jeu de données pseudo-aléatoires.
- Construire les matrices initiales.
- Calculer le vecteur Θ.
- Afficher le nuage de points.
- Tracer la droite séparatriece des domaines.
- Faire des prédictions.

Modèle quadratique 2D

Régression logistique - Modèle quadratique 2D

Équation d'un cercle de centre $(x_{1,x}, x_{2,c})$, de rayon R :

$$(x_1 - x_{1,c})^2 + (x_2 - x_{2,c})^2 = R^2$$

Après avoir développé :

$$\underbrace{x_{1,c}^2 + x_{2,c}^2 - R^2}_{\theta_0} \underbrace{-2x_{1,c}}_{\theta_1} x_1 \underbrace{-2x_{2,c}}_{\theta_2} x_2 \underbrace{+1}_{\theta_3} x_1^2 \underbrace{+0}_{\theta_4} x_1 x_2 \underbrace{+1}_{\theta_5} x_2^2 = 0$$

Régression logistique - Modèle quadratique 2D

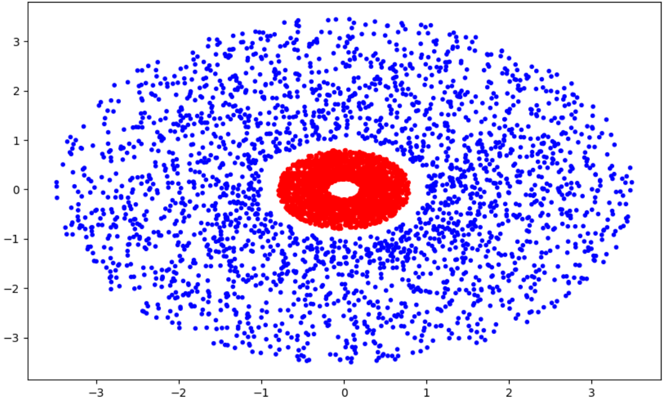
On peut rechercher un sextuplet $(\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4, \hat{\theta}_5)$ tel que la frontière soit d'équation :

$$\hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_2 + \hat{\theta}_3 x_1^2 + \hat{\theta}_4 x_1 x_2 + \hat{\theta}_5 x_2^2 = 0$$

Ou encore :

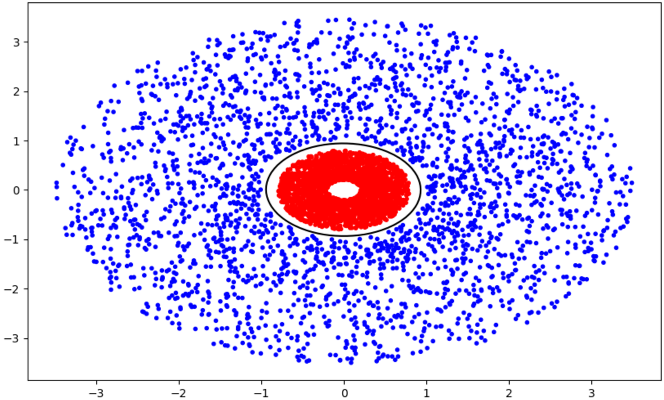
$$X^T \cdot \Theta = 0 \qquad \text{avec} \qquad X = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix} \qquad \hat{\Theta} = \begin{pmatrix} \hat{\theta}_0 \\ \hat{\theta}_1 \\ \hat{\theta}_2 \\ \hat{\theta}_3 \\ \hat{\theta}_4 \\ \hat{\theta}_5 \end{pmatrix}$$

Régression logistique - Modèle quadratique 2D



nuages de points

Régression logistique - Modèle quadratique 2D



5000 points - 10^5 itérations

Régression logistique - Modèle linéaire 2D

Code sur machine

- Générer un jeu de données pseudo-aléatoires.
- Construire les matrices initiales.
- Calculer le vecteur Θ.
- Afficher le nuage de points.
- Tracer la courbe séparatrice des domaines.
- Faire des prédictions.