

# Big Data : analyse des données avec Python

Redha Moulla

Bruxelles, 3 - 5 novembre 2025

# Plan de la formation

- Qu'est-ce que l'intelligence artificielle ?
- Introduction au machine learning
- Apprentissage supervisé
- Apprentissage non supervisé
- Introduction au deep learning

# Qu'est-ce que l'intelligence artificielle ?

# Définition littérale de l'intelligence artificielle

## 1. Intelligence

Ensemble des fonctions mentales ayant pour objet la connaissance conceptuelle et relationnelle.

- Larousse

## 2. Artificielle

Qui est produit de l'activité humaine (opposé à la nature).

- Larousse

# Qu'est-ce que l'intelligence ?

La notion d'intelligence recouvre plusieurs facultés cognitives :

- ① **Raisonnement** : La capacité à résoudre des problèmes et à faire des déductions logiques.
- ② **Apprentissage** : L'aptitude à acquérir de nouvelles connaissances et à s'améliorer grâce à l'expérience.
- ③ **Perception** : La compétence pour reconnaître et interpréter les stimuli sensoriels.
- ④ **Compréhension** : L'habileté à saisir le sens et l'importance de divers concepts et situations.
- ⑤ **Mémorisation** : La faculté de stocker et de rappeler des informations.
- ⑥ **Créativité** : Le pouvoir d'inventer ou de produire de nouvelles idées, de l'originalité dans la pensée.

Mais est-ce que l'intelligence est réductible à des facultés mesurables ?

# Citation d'Aristote

*"Si chaque instrument était capable, sur une simple injonction, ou même pressentant ce qu'on va lui demander, d'accomplir le travail qui lui est propre, comme on le raconte des statues de Dédales ou des trépieds d'Héphaïstos, lesquels, dit le poète, : "Se rendaient d'eux-mêmes à l'assemblée des dieux", si, de la même manière, les navettes tissaient d'elles-mêmes, et les plectres pinçaient tout seuls la cithare, alors, ni les chefs d'artisans n'auraient besoin d'ouvriers, ni les maîtres d'esclaves."*

— Aristote

# Conférence de Dartmouth ?

Articles

AI Magazine Volume 27 Number 4 (2006) © AAAI

## A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence

August 31, 1955

John McCarthy, Marvin L. Minsky,  
Nathaniel Rochester,  
and Claude E. Shannon

The 1956 Dartmouth summer research project on artificial intelligence was proposed by John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. The proposal required 17 pages plus a title page. Copies of the reprints are bound in a volume of the *Proceedings of the Conference on Cognition and Computer*, published by Stanford University. The first 5 papers state the proposal and give the general goals and objectives and interests of the four who proposed the study. In the interest of brevity, this article reprints the last 12 pages which contain the bibliographical statements of the proposal.

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use lan-

guage, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

### 1. Automatic Computers

If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. It may be argued that the present state of present computers may be insufficient to simulate many of the higher functions of the mind. This is true, but it is also true that lack of machine capacity, but our inability to write programs taking full advantage of what we have.

### 2. How Can a Computer be Programmed to Use a Language

It may be speculated that a large part of human language consists of sequences of words according to rules of reasoning and rules of conjecture. From this point of view, learning a generalization consists of admitting a new

"We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer."

# L'intelligence artificielle selon John McCarthy

*"It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable."*

— John McCarthy

# Le Test de Turing

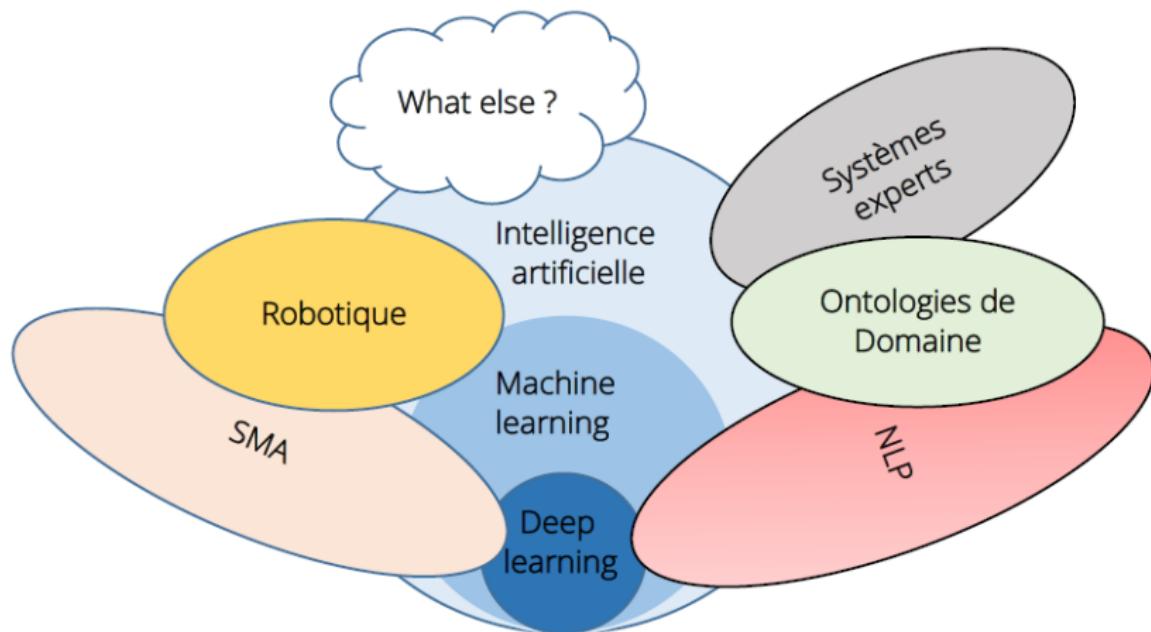
Le Test de Turing, développé par Alan Turing en 1950, est une tentative de mesurer l'intelligence d'une machine, plus précisément de la faculté d'une machine à penser. Cette dernière n'étant pas si évidente à mesurer, le test substitute finalement à la faculté de penser celle de traiter le langage naturel comme un humain.

Les points clés du Test de Turing sont :

- Un interrogateur humain engage une conversation avec un humain et une machine, chacun étant caché de la vue de l'interrogateur.
- Si l'interrogateur ne peut pas déterminer systématiquement quelle est la machine, celle-ci est considérée comme ayant passé le test.
- Le test ne mesure pas la connaissance ou la capacité à être vérifique, mais plutôt la capacité de reproduire le comportement humain.

# Définition pragmatique de l'intelligence artificielle

Il s'agit d'un ensemble de techniques qui permettent à la machine d'accomplir des tâches qui requièrent traditionnellement une intelligence humaine.



# IA forte vs IA faible

La distinction entre IA forte et IA faible se réfère à deux approches conceptuelles différentes dans le domaine de l'intelligence artificielle.

## IA faible :

- Aussi connue sous le nom d'IA "étroite", elle est conçue pour effectuer des tâches spécifiques et ne possède pas de conscience.
- Les systèmes d'IA faible agissent et réagissent uniquement en fonction des instructions programmées et des algorithmes spécifiques.
- Exemples : assistants virtuels, systèmes de recommandation, reconnaissance vocale.

## IA forte :

- Vise à créer des machines dotées de conscience, de compréhension et d'esprit, similaires à l'intelligence humaine.
- L'IA forte serait capable d'apprendre, de raisonner, de résoudre des problèmes et de prendre des décisions indépendamment.
- À ce jour, l'IA forte reste un objectif à atteindre, qui fait l'objet de recherches intensives.

# IA connexionniste vs IA symbolique

## **Intelligence artificielle symbolique :**

Systèmes basés sur des règles et des symboles pour imiter le raisonnement humain.

- Logique
- Ensemble de règles
- Orientée connaissance

## **Intelligence artificielle connexionniste**

: Modèles inspirés du cerveau humain pour apprendre des tâches à partir de données.

- Probabiliste
- Apprentissage machine
- Orientée données

# Machine learning

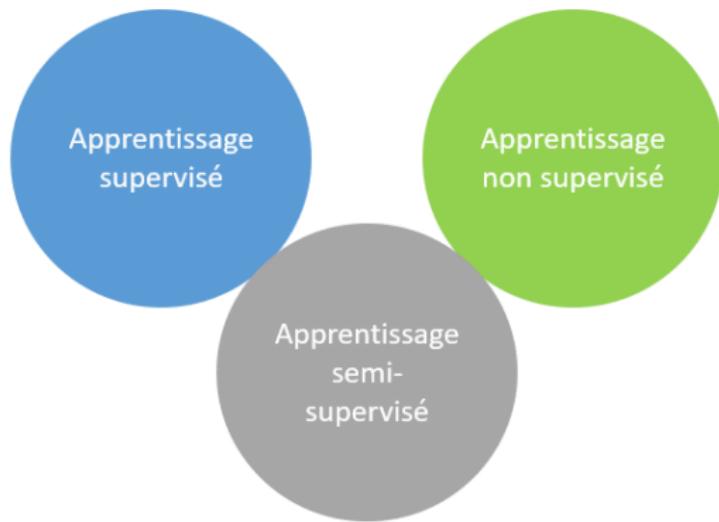
# Définition de l'apprentissage automatique

L'apprentissage automatique est une branche de l'intelligence artificielle qui consiste à doter les machines de la capacité d'apprendre à partir de données sans que celles-ci ne soient explicitement programmées pour exécuter des tâches spécifiques.

Le machine learning englobe plusieurs types d'apprentissage :

- **Supervisé** : Les algorithmes apprennent à partir de données étiquetées pour faire des prédictions ou classifications.
- **Non supervisé** : L'apprentissage est effectué sur des données non étiquetées pour trouver des structures cachées.
- **Semi-supervisé** : Combine des éléments des deux premiers types en utilisant une petite quantité de données étiquetées et une grande quantité de données non étiquetées.
- **Par renforcement** : Les modèles apprennent à prendre des décisions en maximisant une récompense à travers des interactions.

# Typologies d'apprentissage automatique



# L'apprentissage supervisé

**L'apprentissage supervisé** consiste à apprendre un modèle qui associe une étiquette (*label*) à un ensemble de caractéristiques (*features*).

- **Inputs** : un jeu de données *annotées* pour entraîner le modèle.
  - Exemple : des textes (tweets, etc.) avec les *sentiment* associés, positifs ou négatifs.
- **Output** : une étiquette pour un point de donnée inconnu par le modèle.

L'apprentissage supervisé se décline lui-même en deux grandes familles :

- **La classification** : prédire une catégorie ou une classe.
  - Exemple : prédire l'étiquette d'une image (chat, chien, etc.), le sentiment associé à un texte, le centre d'intérêt d'un client à partir de ses commentaires, etc.
- **La régression** : prédire une valeur continue (un nombre réel typiquement).
  - Exemple : prédire le prix d'un appartement, la lifetime value d'un client, etc.

# Classification

## Exemple de classification : credit scoring

Âge	Revenu annuel (k€)	Historique de crédit	Nombre de cartes de crédit	Niveau d'éducation	Propriétaire immobilier (Oui/Non)	Label (y)
30	50	Bon	2	Licence	Oui	Accepté
45	80	Moyen	3	Master	Oui	Accepté
22	20	Mauvais	1	Bac	Non	Refusé
35	60	Bon	4	Licence	Oui	Accepté
40	70	Moyen	2	Bac+2	Non	Refusé

# Régression

## Exemple de régression : prédition des prix des logements

Surface (m <sup>2</sup> )	Nombre de chambres	Distance du centre-ville (km)	Année de construction	Quartier (Score 1-10)	Prix (k€) (y)
80	3	5	2010	8	300
120	4	10	2005	7	450
60	2	2	2020	9	200
150	5	15	1995	6	600
100	3	8	2015	7	400

# L'apprentissage non supervisé

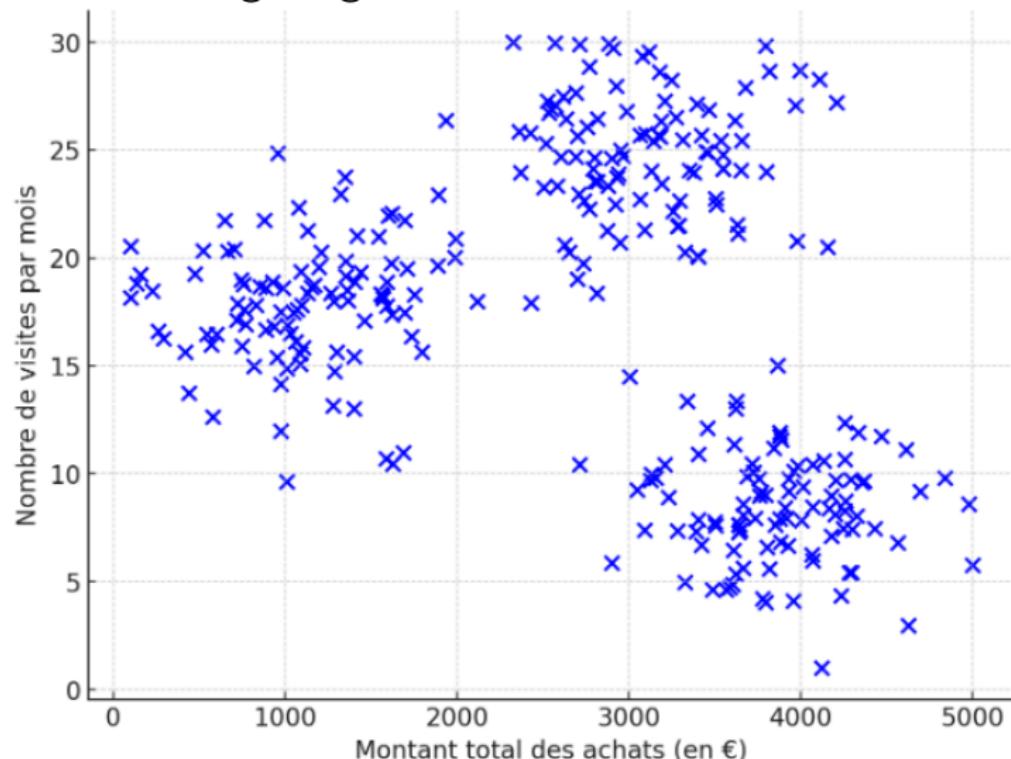
**L'apprentissage non supervisé** se réfère à l'utilisation de modèles d'apprentissage automatique pour identifier des patterns et des structures dans des données qui ne sont pas étiquetées.

Principales typologies de l'apprentissage non supervisé :

- **Clustering** : Regroupement de points de données similaires ensemble.  
Exemple : segmentation de marché, regroupement social.
- **Détection d'anomalies** : Détecter des observations dont les caractéristiques sont inhabituelles par rapport à la majorité.

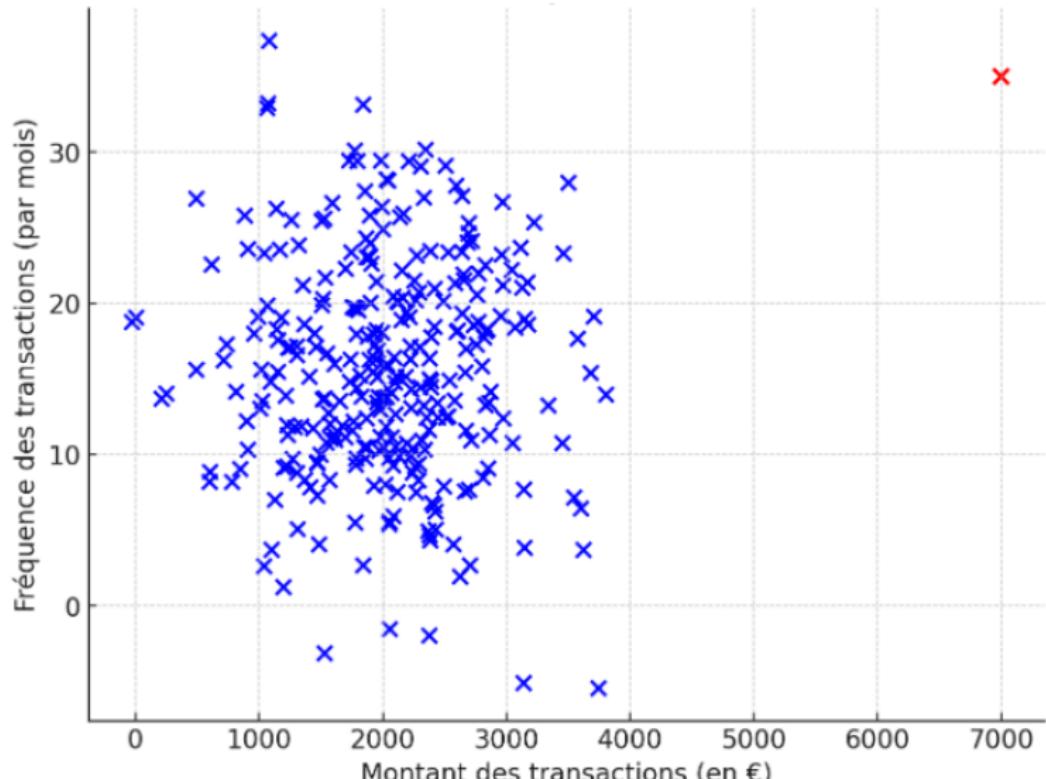
# Clustering

## Exemple de clustering : segmentation clients



# Détection d'anomalies

## Exemple de détection d'anomalies : fraude bancaire



# L'apprentissage semi-supervisé

**L'apprentissage semi-supervisé** combine des éléments des approches supervisées et non supervisées. Il utilise un petit ensemble de données étiquetées et un plus grand ensemble de données non étiquetées pour former des modèles.

Cette méthode est particulièrement utile quand :

- Les données étiquetées nécessitent des ressources coûteuses pour les obtenir, mais les données non étiquetées sont abondantes.
- L'ajout d'un peu d'information étiquetée peut améliorer significativement la performance de modèles entraînés avec des données non étiquetées.

Les applications typiques incluent :

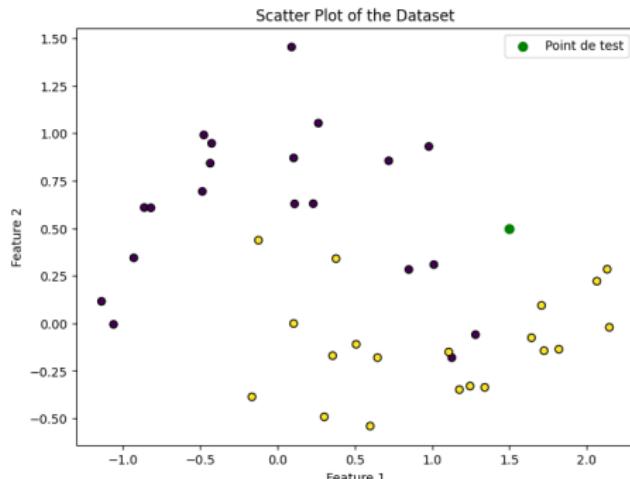
- Développement de systèmes de recommandation plus performants.
- Traitement de langage naturel et analyse de sentiment lorsque les annotations complètes ne sont pas disponibles.

L'apprentissage tente d'exploiter "le meilleur des deux mondes" de l'étiquetage et de la découverte de structure.

# Apprentissage supervisé

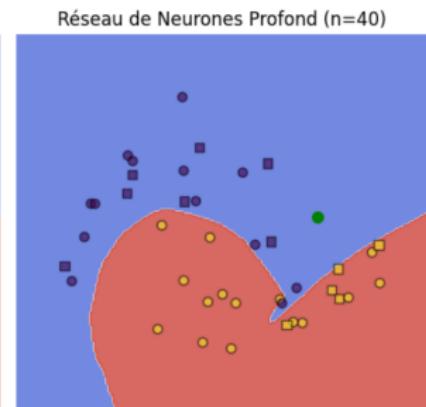
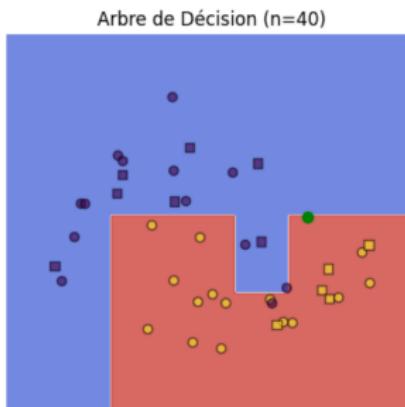
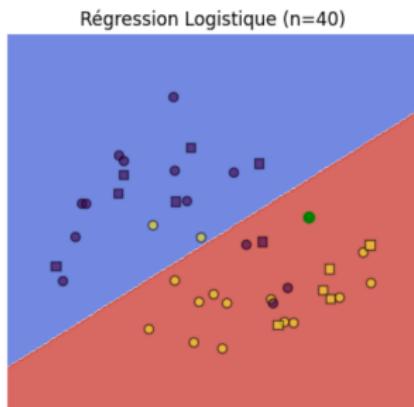
# L'apprentissage supervisé comme un problème d'induction

- **Définition** : L'apprentissage supervisé consiste à apprendre une fonction  $f$  qui mappe les entrées  $X$  aux sorties  $y$ , à partir d'un ensemble d'exemples d'entraînement  $(X, y)$ .
- **Induction** : Le modèle induit une règle générale à partir de données particulières, dans le but de généraliser à de nouvelles instances.
- **Problème de généralisation** : Comment garantir que le modèle apprend une règle qui s'applique à de nouvelles données ?



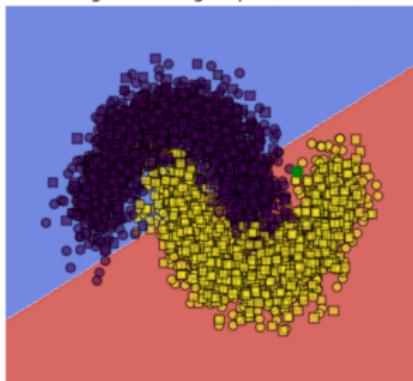
# Une indétermination intrinsèque pour le choix du modèle

Il y a une infinité de manières d'induire un modèle à partir d'un échantillon de données d'entraînement.

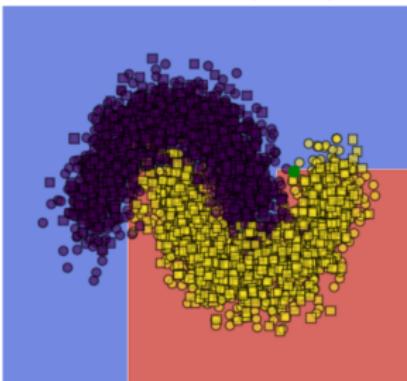


# Sous-apprentissage et surapprentissage sur les données d'entraînement

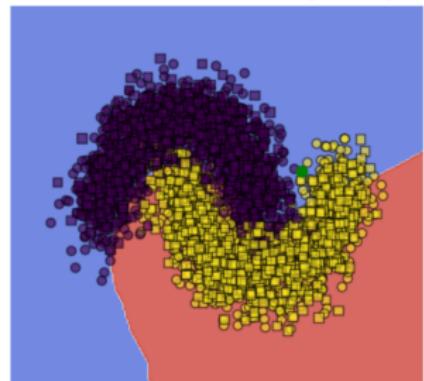
Régression Logistique (n=3000)



Arbre de Décision (n=3000)



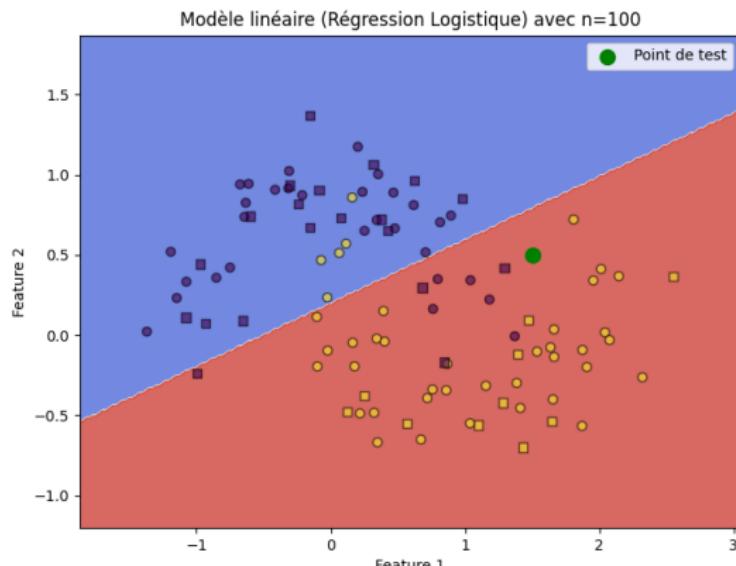
Réseau de Neurones Profond (n=3000)



# Sous-apprentissage

## Definition

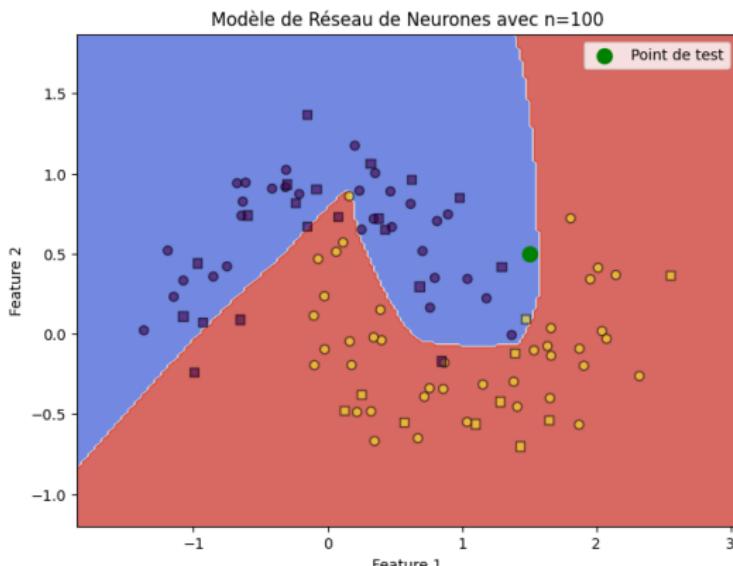
On dit qu'un modèle de machine learning est en régime de sous-apprentissage (underfitting) lorsqu'il n'arrive pas à capturer la complexité (l'information) présente dans le jeu de données d'entraînement.



# Sur-apprentissage

## Definition

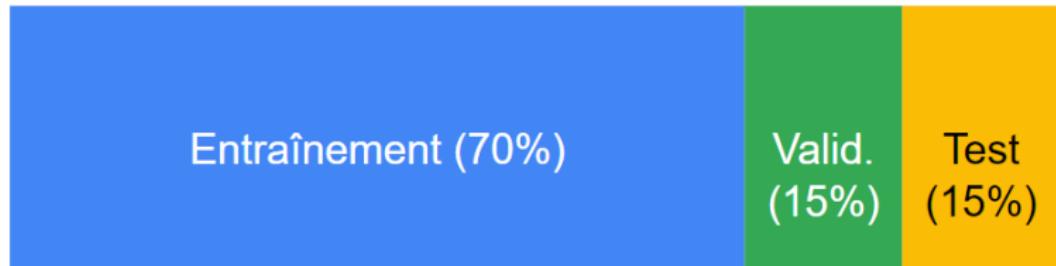
On dit qu'un modèle de machine learning est en régime de sur-apprentissage (overfitting) lorsqu'il n'arrive pas à généraliser à des données non encore observées, i.e. lorsqu'il est trop adapté aux données d'entraînement.



# Sélection de modèle

Pour sélectionner le modèle le plus pertinent par rapport à une métrique donnée, on applique la méthodologie suivante :

- On partitionne le jeu de données disponible en trois parties : un jeu d'entraînement, un jeu de validation et un jeu de test.
- On entraîne  $M$  modèles sur le jeu d'entraînement.
- On évalue les performances respectives des  $M$  modèles sur le jeu de validation et on sélectionne le meilleur.
- Le modèle sélectionné est ensuite évalué sur le jeu de test. Idéalement, le jeu de test est ainsi utilisé une seule fois.



## Exemple : sélection de modèle

Modèle	Précision Entraînement	Précision Validation
Régression Logistique	0.90	0.88
Arbre de Décision	0.95	0.92
Réseau de Neurones Profond	1	0.90

Table: Comparaison des précisions des modèles sur les jeux d'entraînement et de validation

# Métriques de performance : régression

On dispose d'un certain nombre de métriques pour évaluer les performances des modèles de machine learning. Celles-ci peuvent être divisées en deux catégories.

## Régression

- L'erreur quadratique moyenne (MSE) : elle est définie comme la moyenne des carrés des écarts entre les prédictions et les valeurs observées.

$$MSE = \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2$$

- La racine carrée de l'erreur quadratique moyenne (RMSE) :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2}$$

## Métriques de performance : classification 1/2

**Accuracy** : L'accuracy est la métrique de base qui permet d'évaluer les performances d'un modèle de classification. Elle est définie comme :

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

**Matrice de confusion** : La matrice de confusion est une représentation permettant d'offrir plus de finesse par rapport à l'accuracy, notamment quand le jeu de données est déséquilibré (présence de classes majoritaires). Elle compare les prédictions du modèle avec les valeurs réelles et est structurée comme suit :

		Valeur Prédite	
		Positif	Négatif
Valeur Réelle	Positif	Vrai Positif (VP)	Faux Négatif (FN)
	Négatif	Faux Positif (FP)	Vrai Négatif (VN)

## Métriques de performance : classification 2/2

A partir de la matrice de confusion, on peut dériver d'autres métriques :

- Précision : elle est définie comme la proportion des prédictions correctes parmi toutes les prédictions positives :

$$\text{Précision} = \frac{VP}{VP + FP}$$

- Rappel (recall) : il représente la proportion des vrais positifs correctement prédits par le modèle.

$$\text{Rappel} = \frac{VP}{VP + FN}$$

- Score F1 (F1-score) : Le score F1 est défini comme la moyenne harmonique de la précision et du rappel.

$$\text{Score F1} = 2 \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

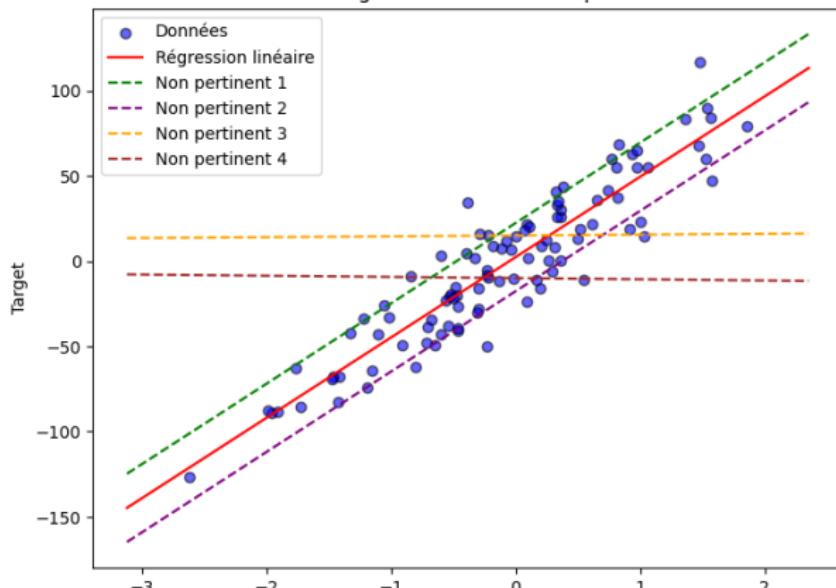
# Modèles classiques de machine learning

# Régression linéaire simple

Soit un ensemble de  $n$  observations  $x_1, x_2, \dots, x_n$  avec les labels correspondants  $y_1, y_2, \dots, y_n$ , on cherche le modèle linéaire qui ajuste le mieux ces données.

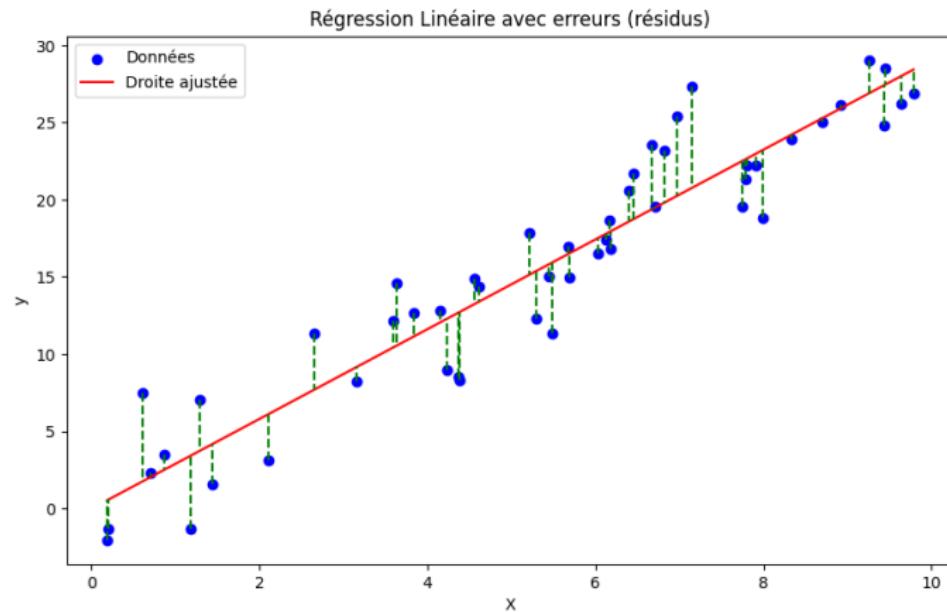
$$\hat{y} = \beta_0 + \beta_1 x$$

Illustration de la régression linéaire avec plusieurs modèles



# Minimisation du risque empirique 1/2

L'erreur de prédiction pour la  $i$  ième observation est :  $e_i = y_i - \hat{y}_i$ . où  $\hat{y}_i = \beta_0 + \beta_1 x_i$ .



## Minimisation du risque empirique 2/2

Déterminer un modèle de régression linéaire simple revient à minimiser les erreurs de prédiction (risque empirique) :

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Autrement dit, chercher les coefficients  $\beta_j$  qui minimisent le risque empirique :

$$\arg \min_{\beta_0, \beta_1} \left( \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \right)$$

Les expressions des  $\beta$  sont obtenues en résolvant les équations normales :

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

où  $\bar{x}$  et  $\bar{y}$  sont les moyennes des  $x_i$  et  $y_i$ , respectivement.

## Exemple : prédire le prix d'un logement en fonction de la surface 1/2

Soit un dataset de 100 points où  $x$  représente la surface (en m<sup>2</sup>) et  $y$  le prix des logements (en €). Les 4 premières lignes sont :

$x$ (Surface m <sup>2</sup> )	$y$ (Prix en €)
40	120 000
65	200 000
80	250 000
55	160 000
:	:

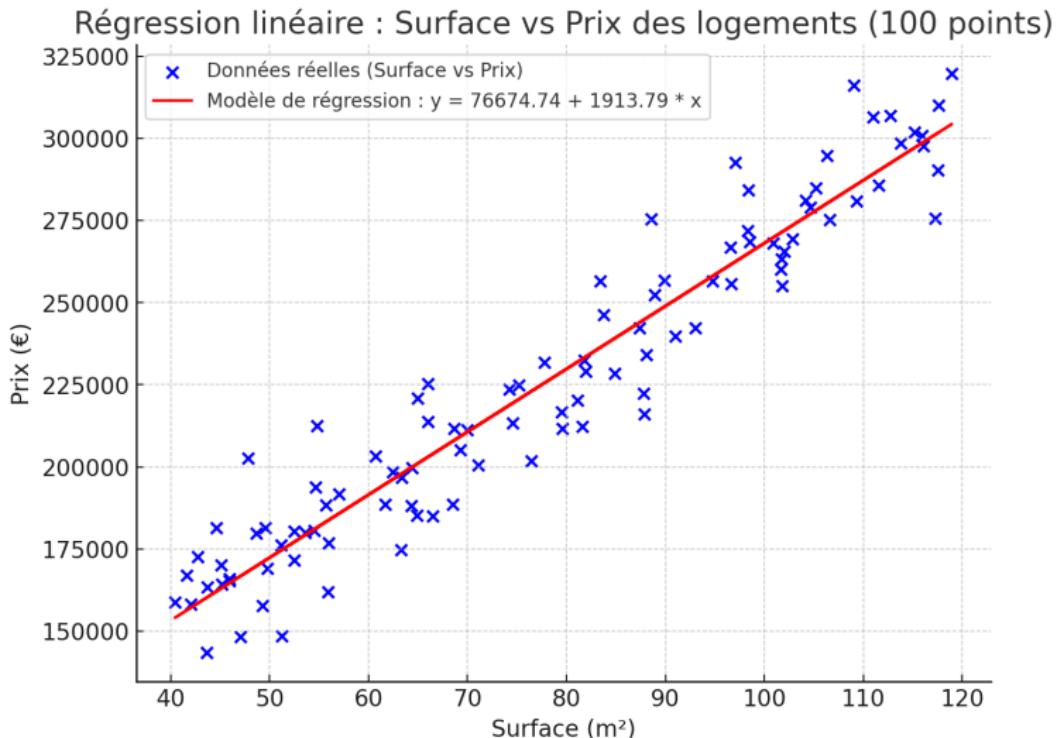
Les coefficients  $\beta_0$  et  $\beta_1$  sont calculés à partir des formules suivantes :

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{1234567}{56789} = 2173.15$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} = 200000 - 2173.15 \times 60 = 69410.5$$

L'équation de la droite de régression obtenue est donc :

## Exemple : prédire le prix d'un logement en fonction de la surface 2/2



# Régression linéaire multiple

On considère  $n$  observations  $X^1, X^2, \dots, X^n$  où chaque observation  $X^i$  est désormais un vecteur ayant  $p$  composantes ( $p$  variables explicatives).

$$X^i = \begin{pmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_p^i \end{pmatrix}$$

La régression linéaire s'écrit alors :

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

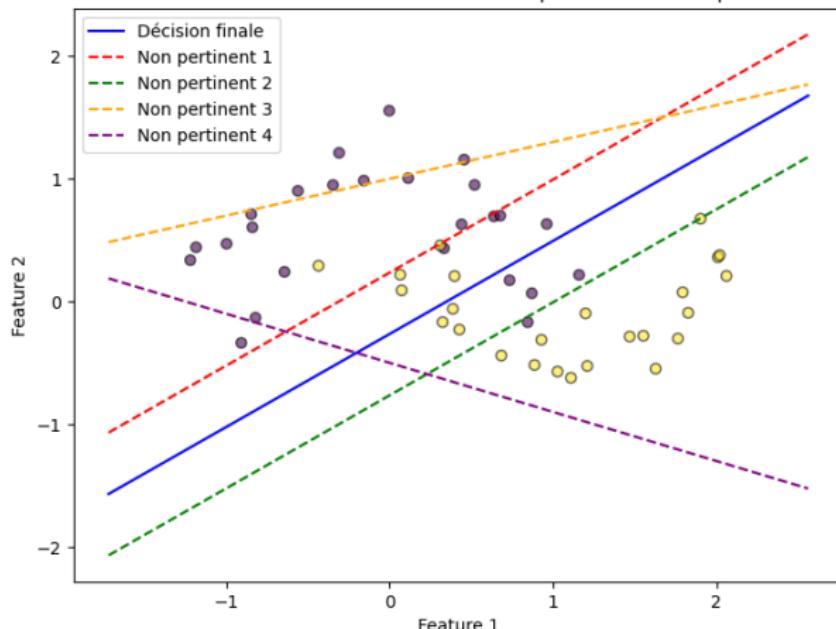
Les coefficients  $\beta_0, \beta_1, \dots, \beta_p$  sont déterminés par la méthode des moindres carrés :

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \sum_{i=1}^n \left( y^i - (\beta_0 + \sum_{j=1}^p \beta_j x_j^i) \right)^2$$

# Principe de la régression logistique

- **Définition :** La régression logistique est un algorithme de classification linéaire utilisé pour prédire la probabilité qu'une observation appartienne à une classe donnée.

Illustration de la surface de décision avec plusieurs droites possibles



# Régression logistique : introduction

La régression logistique est une technique d'analyse statistique utilisée pour modéliser la probabilité d'une variable dépendante binaire. C'est un cas particulier de modèle linéaire généralisé qui est utilisé pour des problèmes de classification.

Principes de la régression logistique :

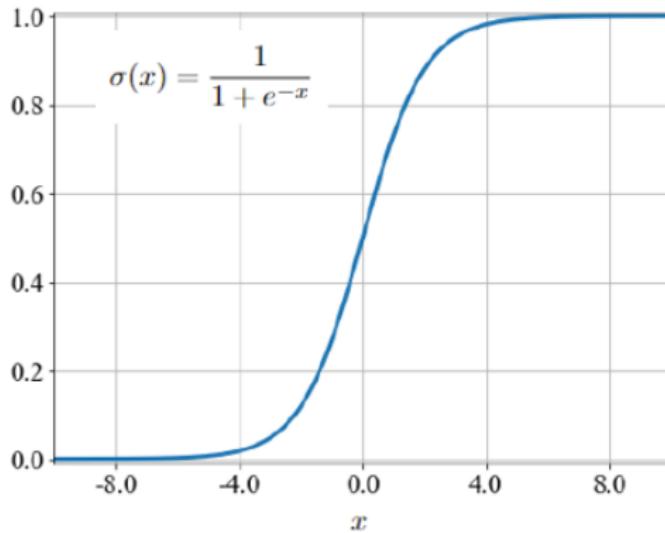
- **Variable dépendante** : On cherche la probabilité que la variable dépendante ( $y$ ) appartienne à une classe (0 ou 1, vrai ou faux, succès ou échec). Autrement dit, on cherche à modéliser  $P(y = 1)$  en fonction des variables dépendantes (explicatives)  $x$ .
- **Odds ratio** : Plus concrètement, on cherche à exprimer la côte anglaise (odd ratio) en fonction des variables dépendantes ( $x$ ).

$$\ln \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

## Régression logistique : fonction sigmoïde

Après quelques simplifications, on peut écrire la probabilité  $p(x)$  (la probabilité pour que  $y$  soit un succès par exemple) :

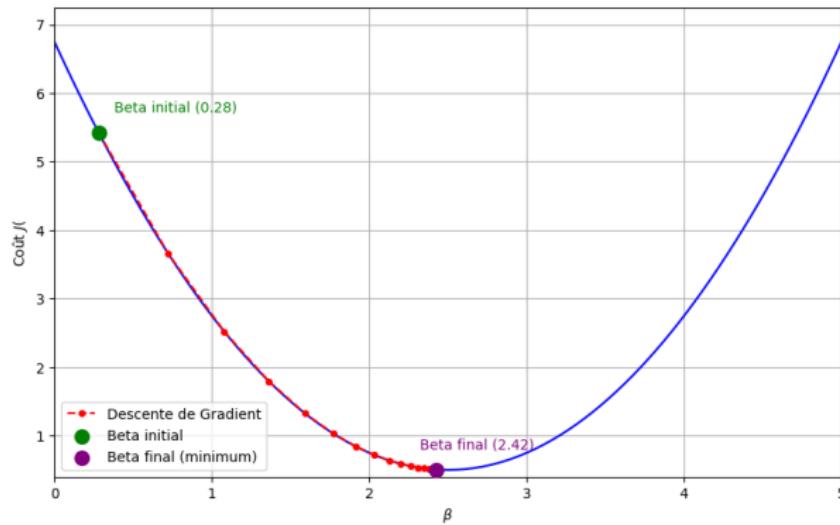
$$p(\mathbf{x}) = \frac{1}{1 + e^{-(\beta^T \mathbf{x})}}$$



# Calcul des coefficients de la régression logistique

Les coefficients de la régression logistique peuvent être calculés en minimisant le risque empirique par rapport à une fonction de coût sous forme d'entropie croisée :

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \left( - \sum_{i=1}^n y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i) \right)$$



# Régression polynomiale

La régression linéaire peut prendre en compte les dépendances non linéaires entre les variables explicatives  $x_1, x_2, \dots, x_p$  et la variable expliquée  $y$ . Lorsque cette dépendance prend la forme d'un polynôme de degré  $d$ , la régression linéaire s'écrit alors :

$$\hat{y} = \beta_{00} + \sum_{j=1}^p \beta_{ij} x_j + \sum_{j=1}^p \beta_{ij} x_j^2 + \dots + \sum_{j=1}^p \beta_{ij} x_j^d$$

Les coefficients  $\beta_{ij}$  peuvent être de la même manière, avec la méthode des moindres carrés.

**Remarque :** On peut utiliser n'importe quelle fonction non linéaire pour transformer les variables explicatives ( $\cos$ ,  $\ln$ , etc.). Le modèle reste tout de même linéaire (linéarité par rapport aux coefficients).

## Choix de modèle

Il y a plusieurs manières d'évaluer la pertinence d'un modèle de régression linéaire. Le coefficient coefficient de détermination  $R^2$  mesure l'ajustement du modèle. Il est donné par :

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

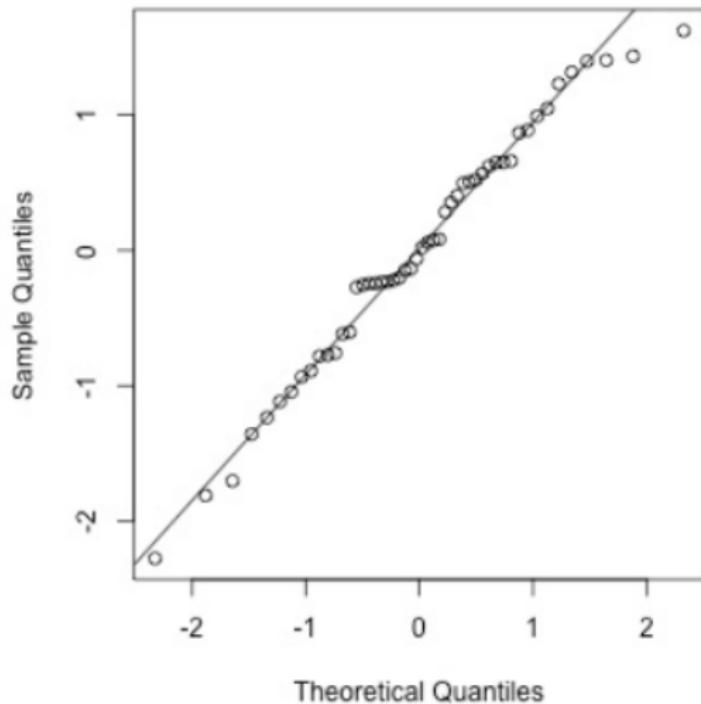
Pour une régression linéaire multiple, on préférera cependant le coefficient de détermination ajusté  $R_a^2$ .

$$R_a^2 = 1 - \frac{n - 1}{n - k - 1} * (1 - R^2)$$

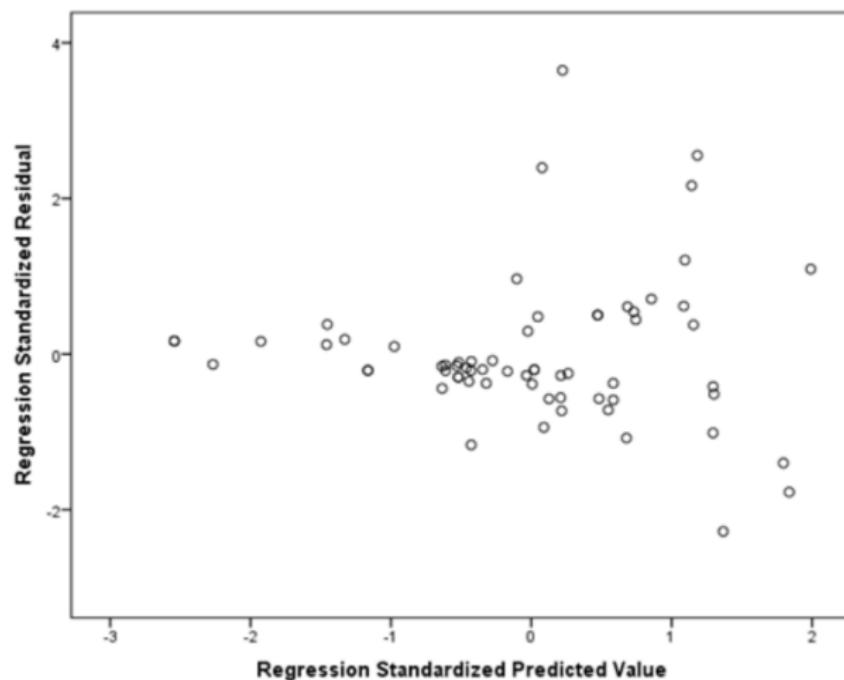
où  $n$  est le nombre d'abosrvations et  $k$  le nombre de variables.

Des critères comme le AIC et le BIC sont également utilisés pour sélectionner un modèle.

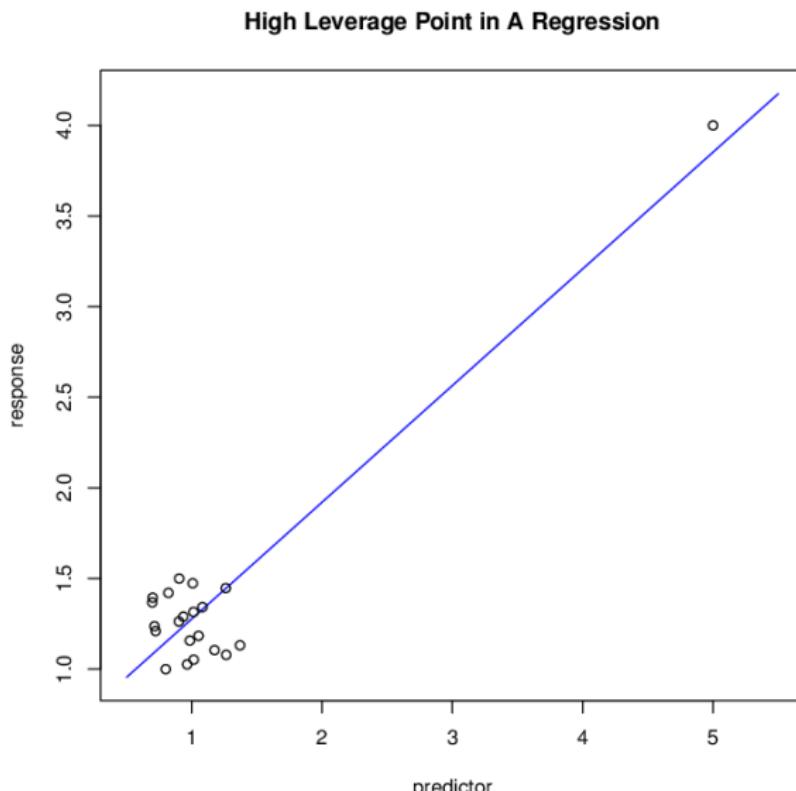
# Diagnostic de la régression linéaire : normalité des résidus



## Diagnostic de la régression linéaire : Homoscédasticité



# Diagnostic de la régression linéaire : effet levier



# Régression Ridge

La régularisation Ridge, dite également régularisation  $L_2$ , permet de remédier au problème de surapprentissage en imposant une contrainte sur les coefficients  $\beta$  lors de la minimisation du risque empirique.

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \left( \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \right)$$

sous la contrainte :

$$\sum_{j=1}^p \beta_j^2 \leq Cte$$

Le problème peut être écrit d'une manière plus compacte :

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \left( \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right)$$

où le paramètre  $\lambda$  contrôle la force la régularisation. Plus  $\lambda$  est grand, plus le modèle est régularisé.

# Formulation vectorielle de la régression Ridge

La somme des carrés des résidus associée à la régression Ridge s'écrit de la manière suivante :

$$RSS = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}$$

Les coefficient *bêta* peuvent alors être calculés en minimisant la *RSS*.

$$\underset{\beta_0, \beta_1, \dots, \beta_p}{\arg \min} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}$$

On peut alors montrer que :

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

où  $\mathbf{I}$  est la matrice identité de dimension  $p \times p$ .

# Formulation vectorielle de la régression Ridge

La matrice de design  $X$  de dimensions  $n \times n$  peut être décomposée en valeurs singulières (SVD) de la manière suivante :

$$X = UDV^T$$

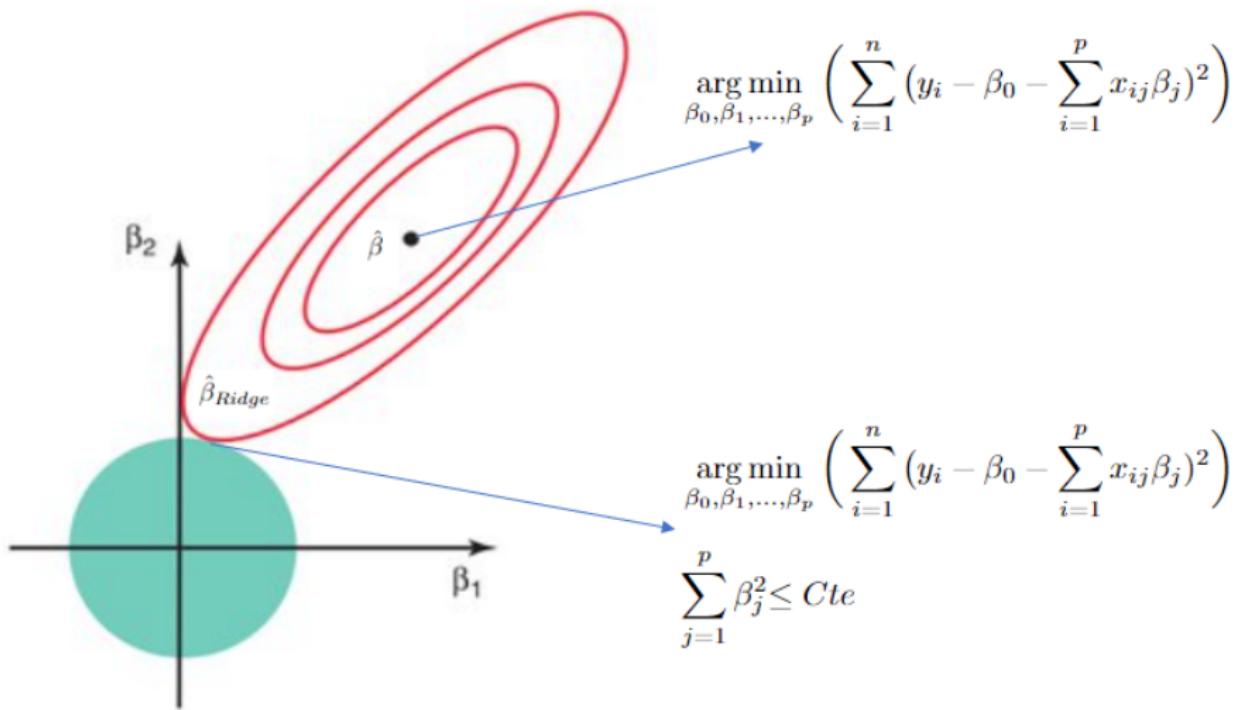
où  $U$  et  $V$  sont des matrices orthogonales de dimensions respectives  $n \times p$  et  $p \times p$ .

On peut montrer que :

$$\begin{aligned} X\beta &= X(X^T X + \lambda I)^{-1} X^T y \\ &= \sum_{j=1}^p u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T \vec{y} \end{aligned} \tag{1}$$

On peut noter que le paramètre de régularisation  $\lambda$  tend à réduire l'influence des variables explicatives associées à une faible valeur singulière.

# Interprétation géométrique de régression Ridge



# Régression Lasso

La régularisation Lasso, dite également régularisation  $L_1$ , force les coefficients associés à des variables explicatives ayant une moindre importance vers zéro. C'est ainsi une technique de réduction de la dimensionnalité du problème pour avoir un modèle avec peu de variable (plus simple et plus explicable).

La régression Lasso est formalisée de la manière suivante :

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \left( \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \right)$$

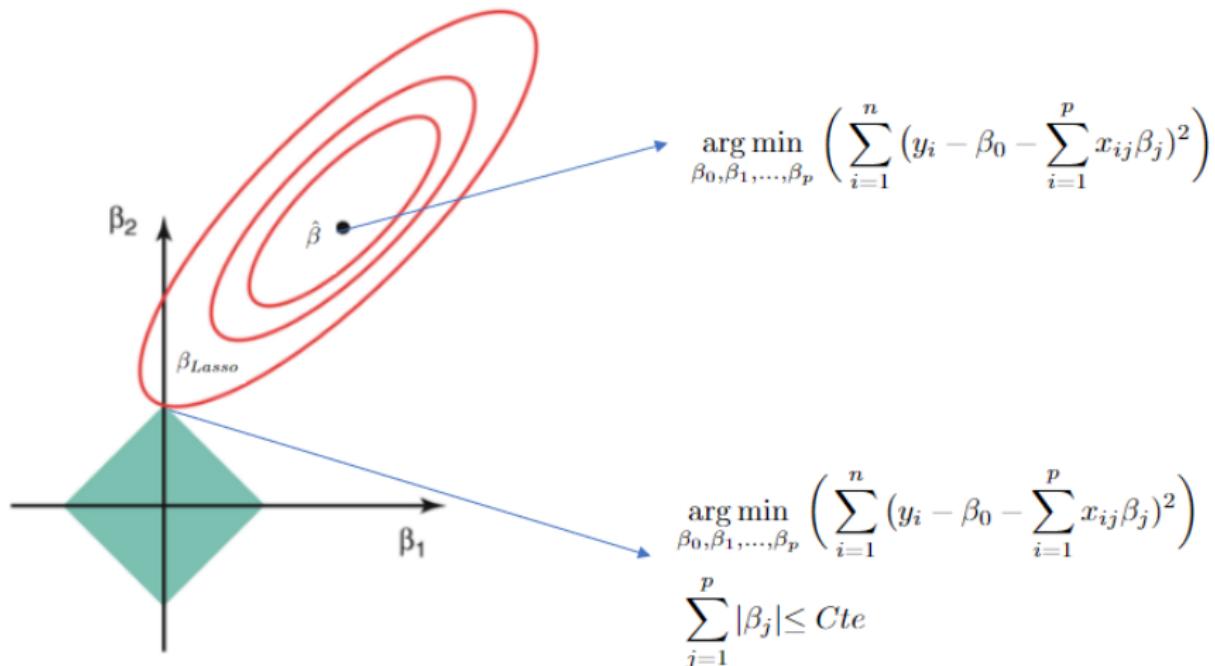
sous la contrainte :

$$\sum_{j=1}^p |\beta_j| \leq Cte$$

Où

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \left( \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j| \right)$$

# Interprétation géométrique de régression Lasso



# Elastic net

Elastic net combine les deux approches, Ridge et Lasso, pondérées avec un paramètre  $\alpha \in [0, 1]$ .

Le problème s'écrit ainsi :

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \left( \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \right)$$

sous la contrainte :

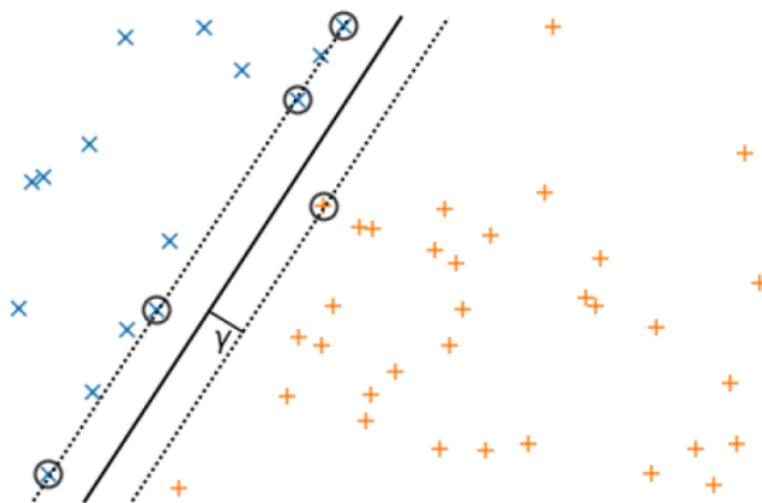
$$\sum_{j=1}^p (1 - \alpha)|\beta_j| + \alpha\beta_j^2 \leq Cte$$

Ou

$$\arg \min_{\beta_0, \beta_1, \dots, \beta_p} \left( \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \left( \sum_{j=1}^p (1 - \alpha)|\beta_j| + \alpha\beta_j^2 \right) \right)$$

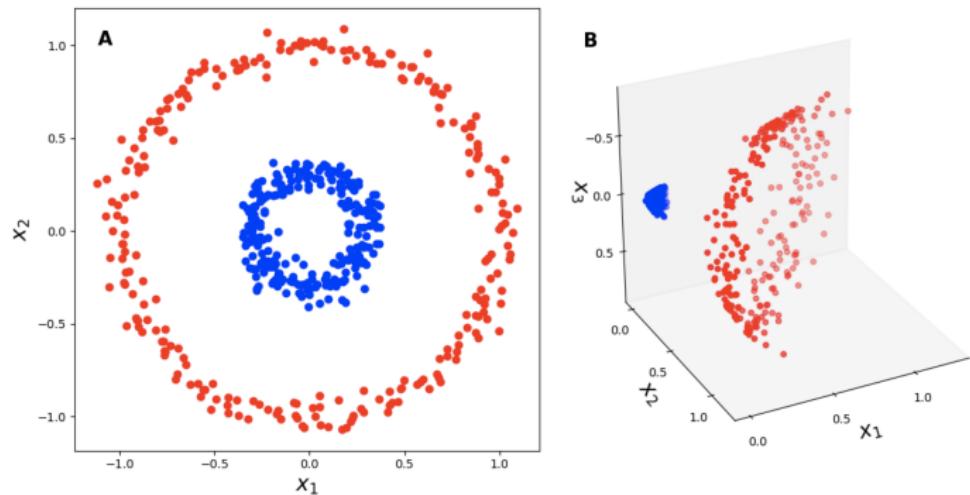
# Machines à vecteurs de support (SVM)

Considérons un problème de classification binaire. On recherche l'hyperplan séparateur qui maximise la marge  $\gamma$  entre les deux classes. La marge  $\gamma$  étant définie comme la distance entre cet hyperplan et les observations les plus proches.



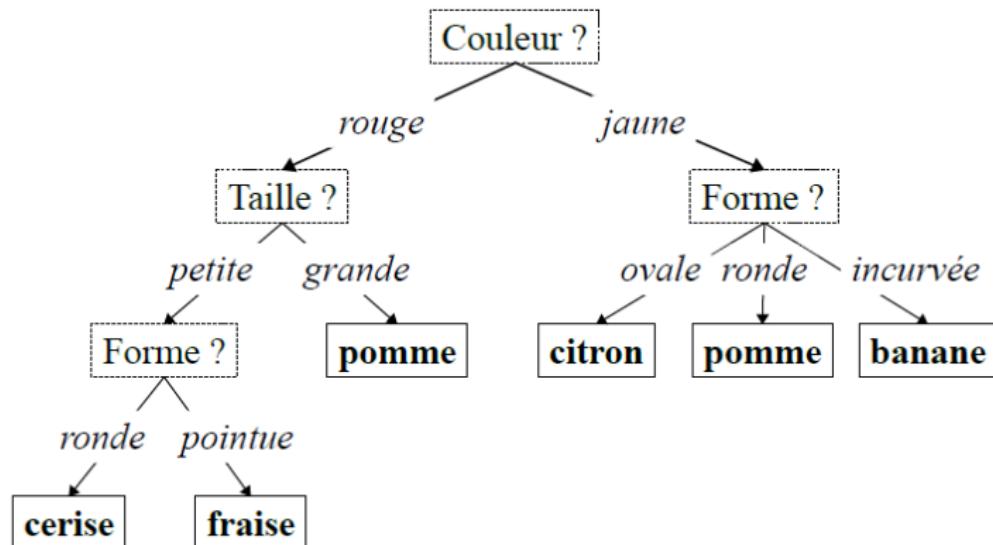
## SVM à noyau

On considère un problème de classification non linéaire. L'astuce du noyau consiste à augmenter la dimension du problème, pour le résoudre ensuite avec un séparateur linéaire dans le nouvel espace.



# Arbres de décision

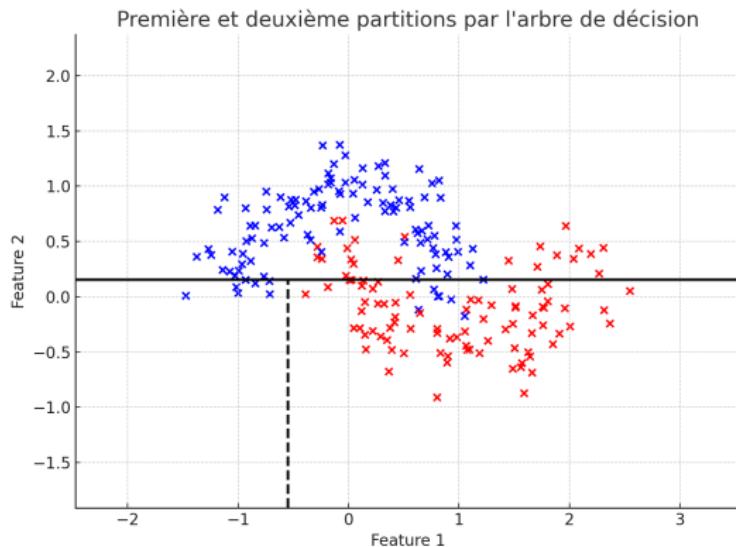
Les arbres de décisions sont des modèles dont le processus de décision est hiérarchique et prend la forme d'un arbre.



# Entraînement des arbres de décision

Les arbres de décisions sont généralement entraînés à l'aide de la technique CART (Classification And Regression Trees).

Etant donné un ensemble d'observations  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , les arbres de décision partitionnent cet espace en plusieurs régions  $R_1, R_2, \dots, R_m$ .



# Critères d'optimisation

Le critère d'optimisation dépend de la tâche en question.

- **Classification**

- Indice de Gini simplifié :  $\sum_{k=1}^K p_{mk}(1 - p_{mk})$
- Entropie croisée :  $-\sum_{k=1}^K p_{mk} \ln(p_{mk})$

- **Régression**

$$\sum_{i=n}^n (y_i - f(x_i))^2$$

# Forêts aléatoires (random forests)

La technique des forêts aléatoires (random forests) consiste à appliquer une approche de type bagging sur les arbres de décision.

L'algorithme random forests suit cette procédure :

- Tirer par bootstrap  $B$  échantillons de tailles  $n$  à partir de l'ensemble  $D$ .
- Pour chaque échantillon tiré, construire un arbre en répétant les étapes suivantes jusqu'à atteindre  $n_{min}$ .
  - Tirer d'une manière aléatoire  $m$  variables parmi les  $p$  variables.
  - Sélectionner la meilleure variable avec le meilleur point de partitionnement.
  - Partitionner le noeud en deux sous-branches.
- Agréger les arbres construits.

Les prédictions sont agrégées selon qu'il s'agisse de régression ou de classification :

- Régression : moyenne  $f^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$
- Classification : vote majoritaire.

## Remarques

- L'algorithme de random forests intègre nativement une forme de validation croisée. Les performances mesurées sur  $\bigcup_{b_i \neq b_k}$  (out of bag ou OOB) sont souvent proches de celles que l'on pourrait mesurer avec une validation croisée.
- Le nombre de variable tirées pour chaque noeud est généralement donné par  $\sqrt{p}$  pour la classification et  $\frac{p}{3}$  pour la régression. Cet hyperparamètre dépend cependant du problème considéré.
- Lorsque le nombre de variable est élevé alors que le nombre de variables réellement partinente est faible, la probabilité que les  $p$  variables sélectionnées pour chaque partitionnement incluent des variables pertinentes devient faible, et les performances du modèle en termes de généralisation peuvent se détériorer considérablement.
- L'algorithme random forests permet de restituer des informations sur l'importance des variable (feature importance).

# Classification Bayésienne

La classification bayésienne repose sur l'utilisation de la probabilité conditionnelle et le théorème de Bayes pour prédire la catégorie d'une nouvelle observation. Elle est particulièrement efficace dans les situations où la dimensionnalité des données est élevée ou lorsque les données sont incomplètes.

La formule fondamentale de la classification bayésienne est le calcul de la probabilité a posteriori pour chaque classe  $C_k$  donnée une observation  $\mathbf{x}$  :

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}|C_k) \cdot P(C_k)}{P(\mathbf{x})}$$

où :

- $P(C_k|\mathbf{x})$  est la probabilité a posteriori de la classe  $C_k$  étant donné l'observation  $\mathbf{x}$ .
- $P(\mathbf{x}|C_k)$  est la vraisemblance de l'observation  $\mathbf{x}$  dans la classe  $C_k$ .
- $P(C_k)$  est la probabilité a priori de la classe  $C_k$ .
- $P(\mathbf{x})$  est la probabilité marginale de l'observation  $\mathbf{x}$ , souvent calculée comme la somme des vraisemblances de  $\mathbf{x}$  sur toutes les classes pondérée par leur probabilité a priori.

# Classification avec Naive Bayes

Le classificateur Naive Bayes est un modèle probabiliste basé sur le théorème de Bayes, avec une hypothèse simplificatrice d'indépendance conditionnelle entre les caractéristiques données la classe.

Points clés :

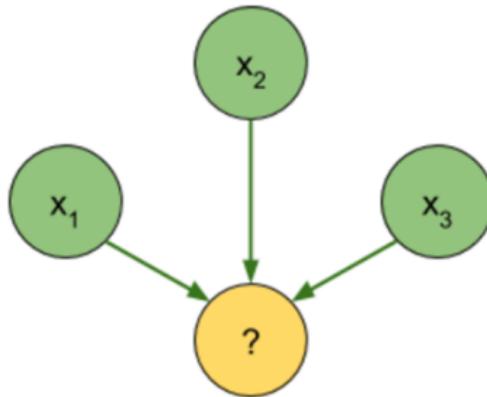
- Facile à construire et particulièrement utile pour de très grands ensembles de données.
- Malgré son hypothèse naïve d'indépendance, il peut être étonnamment efficace.
- Largement appliqué dans la classification de documents, le filtrage anti-spam et les systèmes de recommandation.

## Hypothèse d'indépendance dans Naive Bayes

L'hypothèse d'indépendance de Naive Bayes suppose que la présence (ou l'absence) d'une caractéristique particulière est indépendante de la présence (ou l'absence) de toute autre caractéristique, donnée la classe.

Cela simplifie le calcul de  $P(x|C_k)$  car :

$$P(x_1, \dots, x_n|C_k) = P(x_1|C_k) \times \dots \times P(x_n|C_k)$$



# Avantages et limites de Naive Bayes

## Avantages :

- Efficacité en temps et en espace, même avec de grands ensembles de données.
- Performance robuste et souvent compétitive avec des classificateurs plus sophistiqués.
- Bonne performance sur des données multidimensionnelles et catégorielles.

## Limites :

- L'hypothèse d'indépendance peut être irréaliste et peut affecter la performance.
- Moins performant si les caractéristiques sont corrélées.
- Peut être biaisé si l'ensemble de données d'entraînement ne représente pas bien toutes les classes.

# Apprentissage non supervisé

# Apprentissage non supervisé

Dans l'apprentissage non supervisé, on considère  $n$  observations sans labels. On s'intéresse fondamentalement à la probabilité jointe de ces observations.

On peut distinguer deux grandes catégories d'apprentissage non supervisé :

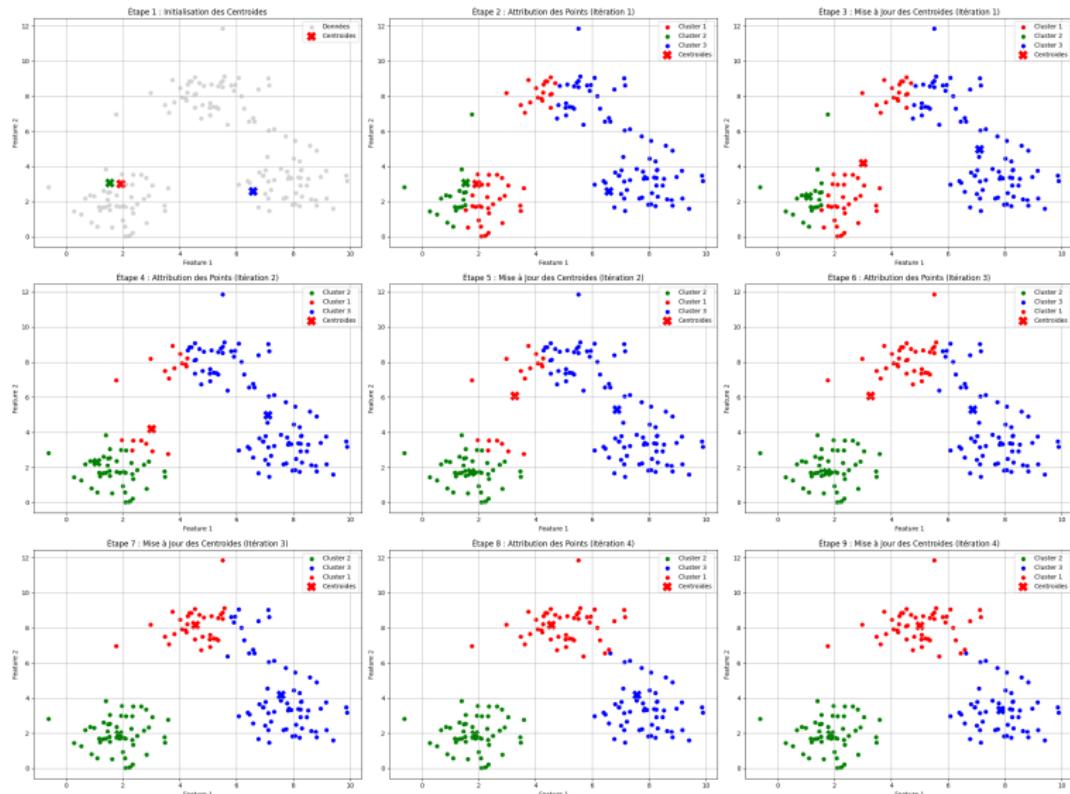
- **Clustering (partitionnement)** : cela consiste à partitionner les  $n$  observations en  $K$  groupes pertinents (généralement le critère de pertinence a une signification d'un point de vue métier).
- **Réduction de dimension** : il s'agit de trouver une représentation des données originelles dans un nouvelles espace de plus petite dimension. Cela peut être effectué à différentes fins : visualisation des données, compression des données, amélioration des performances du modèles (modèles plus robuste, plus explicables, etc.).
- **Détection d'anomalie** : il s'agit de détecter des observations qui présentent un profil (des features) inhabituels par rapport au profil moyen de la majorité des observations.

# K-means

L'algorithme de Lloyd tente de regrouper les données en clusters en minimisant les distances entre les points d'un même cluster tout en maximisant les distances entre points appartenant à différents clusters.



# Algorithme de Lloyd

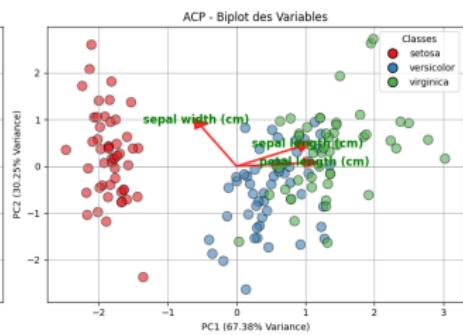
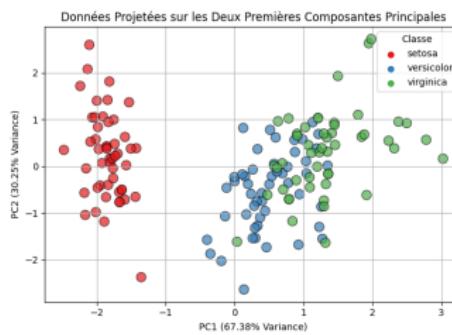
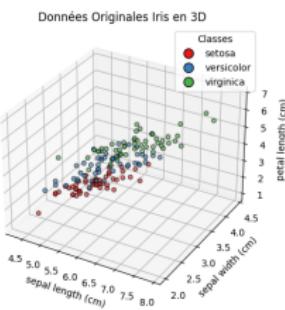


## Remarques

- L'algorithme des k-means étant basé sur une distance euclidienne, il est nécessaire de normaliser les données avant de l'exécuter.
- L'algorithme des k-means est très sensible aux données aberrantes (outliers). Il faut donc considérer les données d'une manière attentive. Cependant, cela permet également d'utiliser l'algorithme des k-means pour la détection automatique des outliers.
- Les centroïdes étant initialisés d'une manière aléatoire, les clusters obtenus ne sont pas stables ; les clusters peuvent changer d'une exécution à l'autre. Il existe cependant une variante plus stable, appelée k-means++, qui permet de sélectionner les centroïdes d'une manière semi-aléatoire.
- Il est possible de partitionner les données avec une métrique plus générale que la distance euclidienne. On peut définir un algorithme k-means à noyau sur un espace de Hilbert pour aller au-delà de la métrique euclidienne.
- **K-means n'est pas adapté aux données en grande dimension.**

# Analyse en Composantes Principales (ACP)

L'Analyse en Composantes Principales (ACP) est une technique statistique de réduction de dimensionnalité. Elle transforme les données en un nouveau système de coordonnées où la plus grande variance est capturée sur les premiers axes, appelés composantes principales.



# Formulation de l'ACP

Soit  $X$  une matrice de données de dimension  $n \times p$  ( $n$  observations,  $p$  variables), centrée (moyenne nulle). L'ACP cherche à trouver les vecteurs propres et les valeurs propres de la matrice de covariance  $C = \frac{1}{n-1} X^T X$ .

La matrice de covariance  $C$  peut être décomposée comme suit :

$$C = VLV^T$$

où  $V = [u_1, u_2, \dots, u_p]$  est la matrice des vecteurs propres et  $L$  est une matrice diagonale des valeurs propres  $\lambda_k$ .

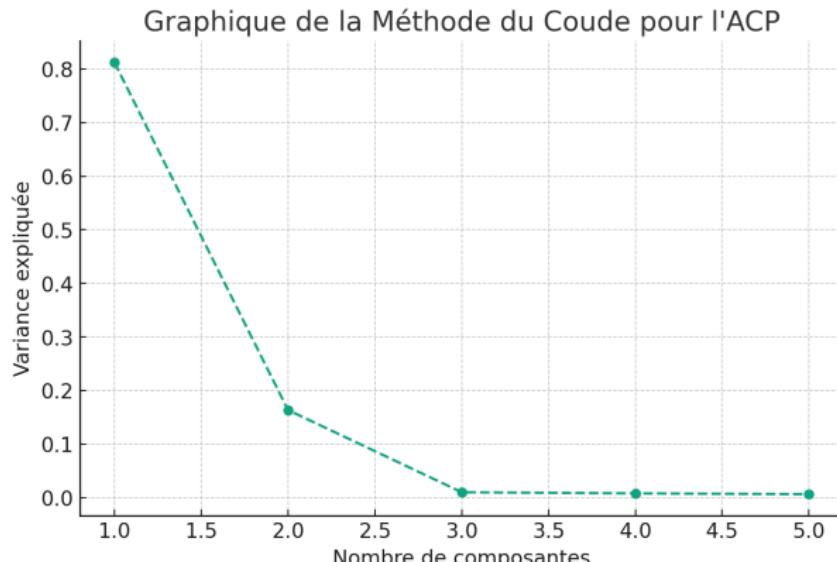
La contribution de chaque composante principale à la variance totale est donnée par :

$$\frac{\lambda_k}{\sum_{i=1}^p \lambda_i}$$

# Choix du nombre de composantes principales

Le nombre de composantes à retenir est déterminé en fonction du pourcentage de variance totale que l'on souhaite expliquer.

On utilise généralement la méthode du coude (Scree plot). Il s'agit d'un graphique montrant la proportion de la variance expliquée en fonction du nombre de composantes.



# Isolation Forest : Principe

L'Isolation Forest est une technique de détection d'anomalies basée sur l'isolement des observations. Son efficacité repose sur l'hypothèse que les anomalies sont "faciles à isoler" par rapport aux observations normales.

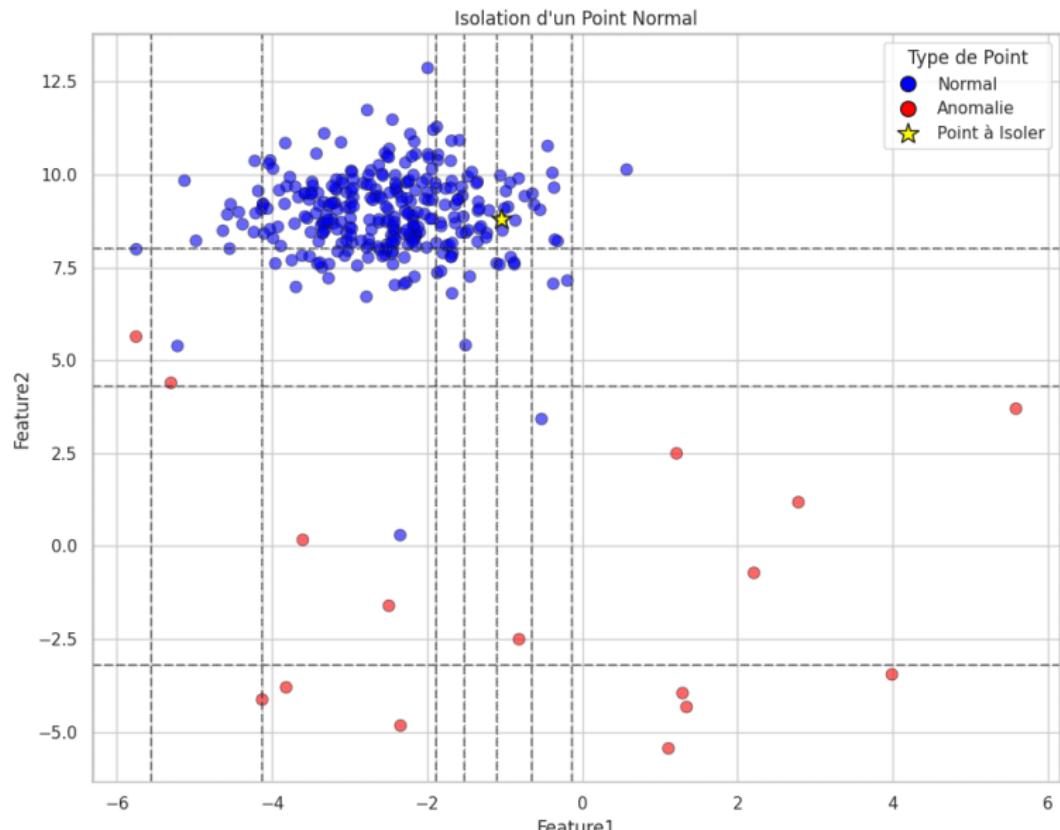
- Fonctionne en construisant des arbres d'isolement à partir de sous-ensembles de données.
- Isoler une observation signifie la séparer des autres par des divisions aléatoires de l'espace des caractéristiques.
- Moins de divisions sont nécessaires pour isoler une anomalie, ce qui constitue le fondement du score d'anomalie.

Les arbres d'isolement sont construits de manière récursive :

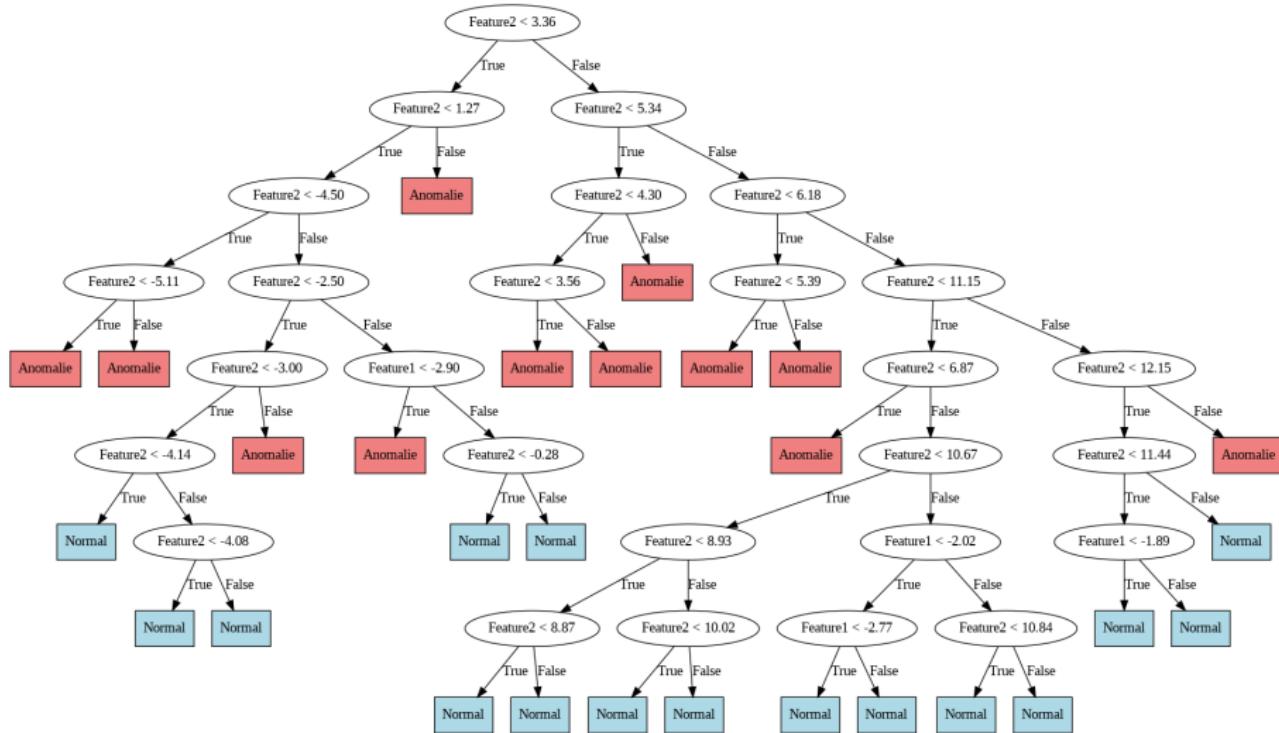
- ➊ Sélection aléatoire d'un sous-ensemble de données.
- ➋ Choix aléatoire d'une caractéristique et d'une valeur de seuil pour diviser le sous-ensemble.
- ➌ Répétition des divisions jusqu'à l'isolement des observations ou atteinte d'une limite de profondeur prédéfinie.

Chaque arbre est ainsi unique, offrant une perspective différente sur les données.

# Isolation tree : fonctionnement 1/2



## Isolation tree : fonctionnement 2/2



# Merci de votre attention

redha.moulla@axia-conseil.com