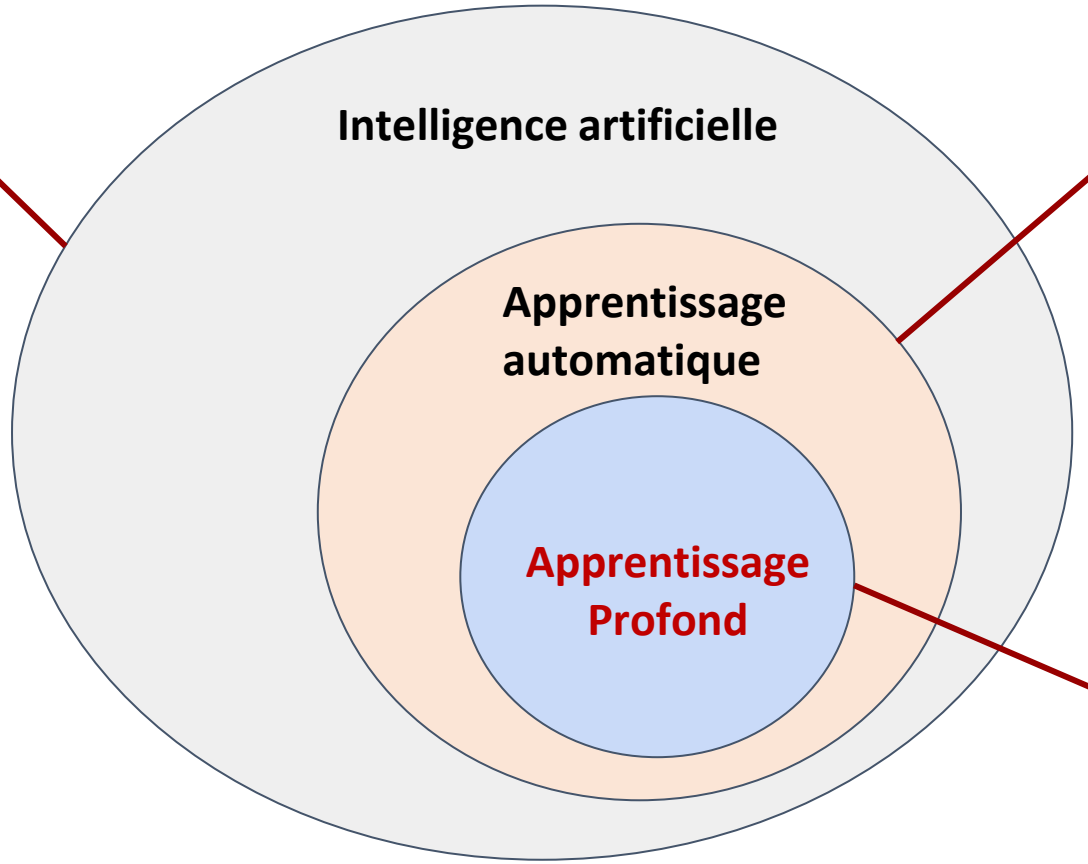
Intelligence Artificielle (IA)
- Apprentissage supervisé -

Groupe des étudiants : FISA4N

Khadidja OULD AMER
E-Mail : khadidja.ouldamer@isen-ouest.yncrea.fr

Disciplines de l'intelligence artificielle

Intelligence artificielle IA : vise à créer des systèmes intelligents capables d'imiter certaines fonctions du cerveau humain, comme observer, comprendre, analyser et agir en conséquence.



Machine Learning (Apprentissage automatique) ML : Sous-domaine de l'IA qui utilise des algorithmes permettant aux systèmes d'apprendre et de s'améliorer automatiquement à partir de données, sans programmation explicite.

Deep Learning (Apprentissage profond) DL : Type d'apprentissage automatique utilisant des réseaux de neurones artificiels multicouches (profonds) pour traiter des données complexes.

Limites de l’apprentissage automatique (classique)

Traitement inefficace des données brutes
Exemple : une image 1024×1024×3 pixels génère 3 millions de caractéristiques
Nécessité de réduire et sélectionner manuellement les features pertinentes

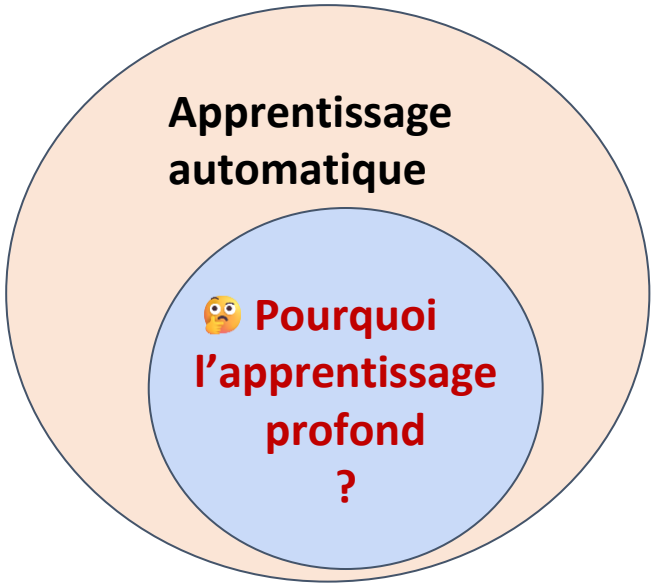
Ingénierie des caractéristiques manuelle (Feature Engineering)
Processus long et complexe nécessitant une expertise métier approfondie
Requiert une connaissance spécifique du domaine d'application
Exemple en reconnaissance d'images : extraction manuelle des bords, textures et formes

Dépendance à la qualité des features
Les performances du modèle dépendent directement de la qualité de l'ingénierie des features
Risque de perte d'informations importantes lors de la sélection

Inadapté aux données non structurées
Excellentes performances sur les données structurées (exemple : données tabulaires)
Difficultés avec les formats non structurés : Images, vidéo, son, texte brut

Capacité de modélisation limitée
Difficulté à capturer des relations non-linéaires complexes
Limité dans la modélisation d'interactions multiples entre variables

Problèmes de scalabilité
Entraînement difficile sur des jeux de données massifs
Limitations en taille et dimension des données
Performance qui peut diminuer avec l'augmentation de l'échelle



Pourquoi l ’Apprentissage Profond?

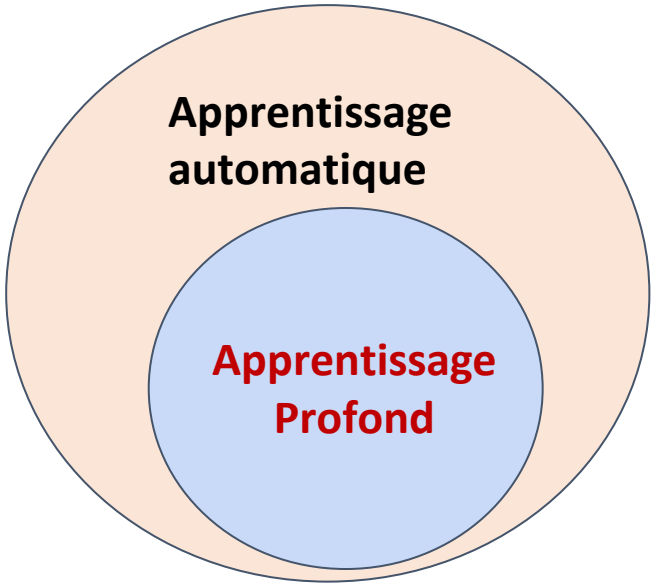
Apprentissage automatique des features (Feature Learning)
Les réseaux apprennent directement les représentations pertinentes à partir des données brutes.
Exemple : en vision, un CNN apprend automatiquement → bords → textures → formes → objets.

Adapté aux données non structurées
Deep Learning est devenu l’approche de référence (CNN, RNN, Transformers, etc.) pour les images, audio, texte.

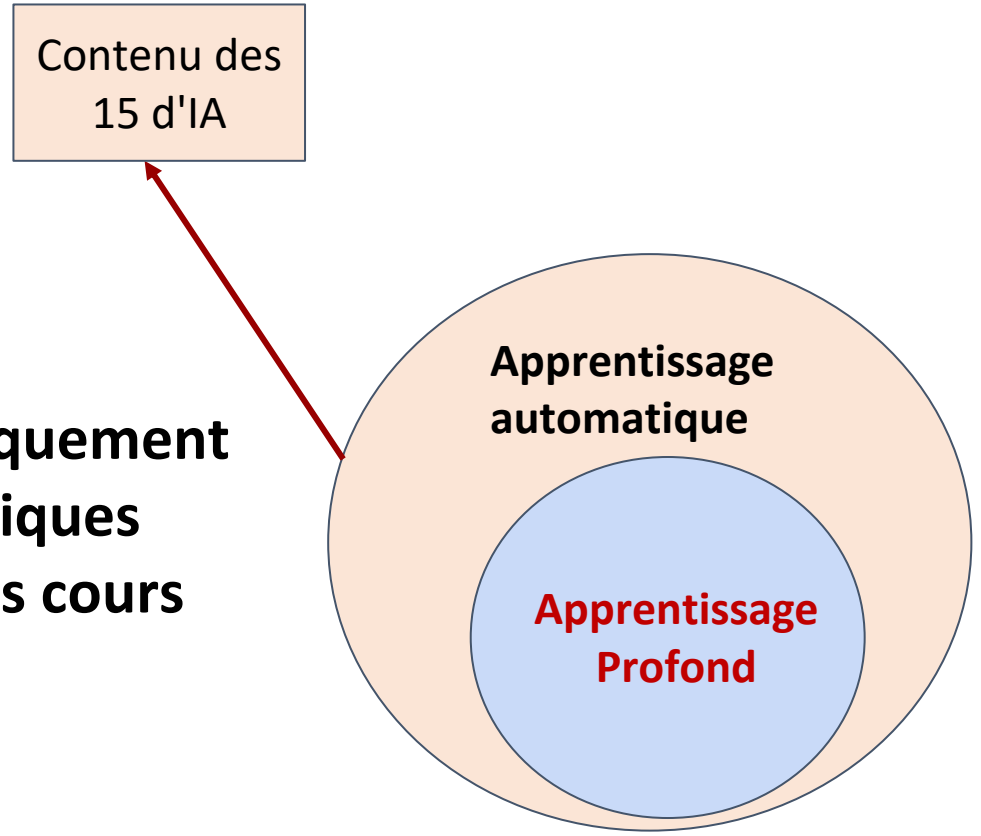
Capacité d’expression très élevée
Grâce à la profondeur (nombre de couches), un réseau peut approximer des relations extrêmement complexes, bien plus qu’un modèle linéaire ou un SVM.

Exploite les grandes quantités de données
Plus on fournit de données, meilleures sont les performances (contrairement à beaucoup de modèles classiques qui saturent rapidement).
Couplé à la puissance de calcul (GPU, TPU), le Deep Learning est hautement scalable.

Performances de pointe
Dans de nombreux domaines (vision, NLP, reconnaissance vocale, jeux, bioinformatique), le Deep Learning dépasse largement les approches classiques.



Ce cours est une introduction à l'IA et couvre uniquement l'apprentissage automatique classique. Les techniques d'apprentissage profond seront abordées dans les cours dédiés au Deep Learning (selon le DP).



Types d'Apprentissage Automatique

- **Apprentissage Supervisé (Supervised Learning)**

Consiste à entraîner un modèle à partir de données étiquetées, c'est-à-dire que chaque exemple d'entrée est associé à une étiquette correcte (classe ou valeur). Le but est de permettre au modèle de prédire les étiquettes de nouvelles données qu'il n'a jamais vues auparavant.

- **Apprentissage Non Supervisé (Unsupervised Learning)**

Utilise des données non étiquetées. Le modèle tente alors de découvrir des structures cachées dans ces données, par exemple en regroupant les données similaires en clusters.

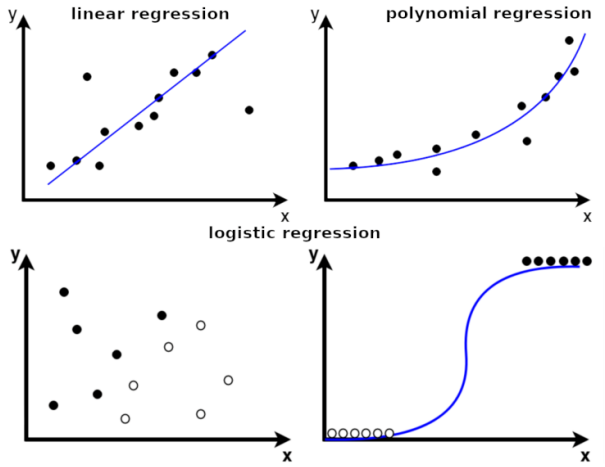
- **Apprentissage Semi-Supervisé (Semi-Supervised Learning)**

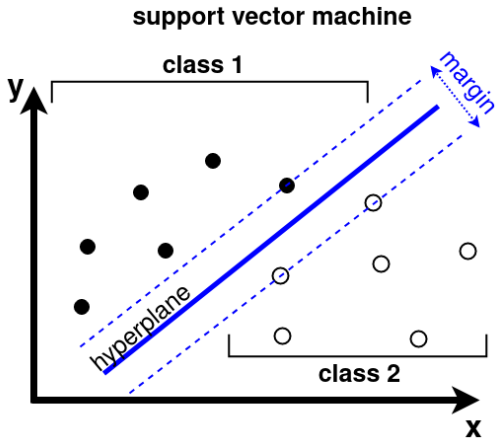
Combine à la fois des données étiquetées et non étiquetées. L'idée est d'exploiter le petit nombre de données étiquetées disponibles pour guider l'apprentissage sur un grand ensemble de données non étiquetées, afin d'améliorer les performances du modèle.

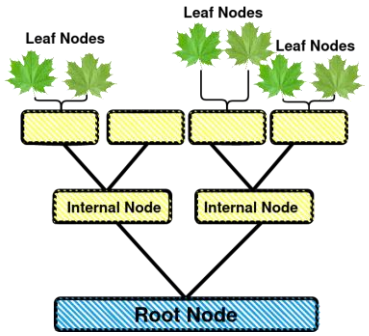
- **Apprentissage par Renforcement (Reinforcement Learning)**

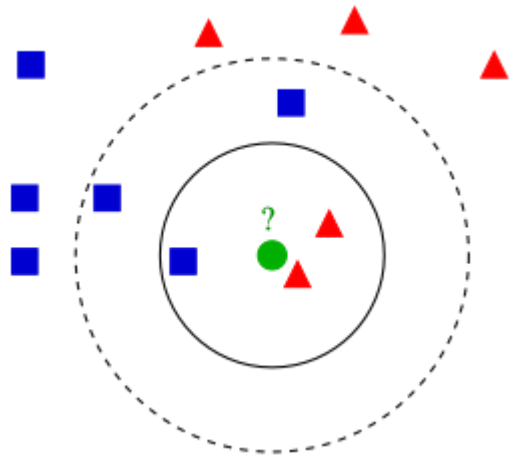
Repose sur l'interaction d'un agent avec un environnement. L'agent reçoit des récompenses ou des punitions en fonction de ses actions et apprend à maximiser la récompense cumulative au fil du temps.

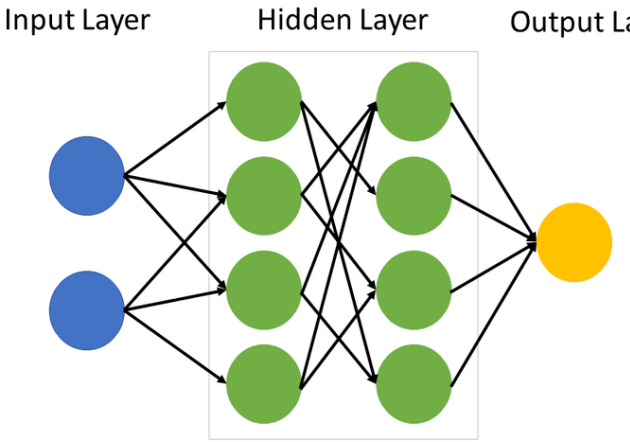
Les algorithmes d'apprentissage supervisé

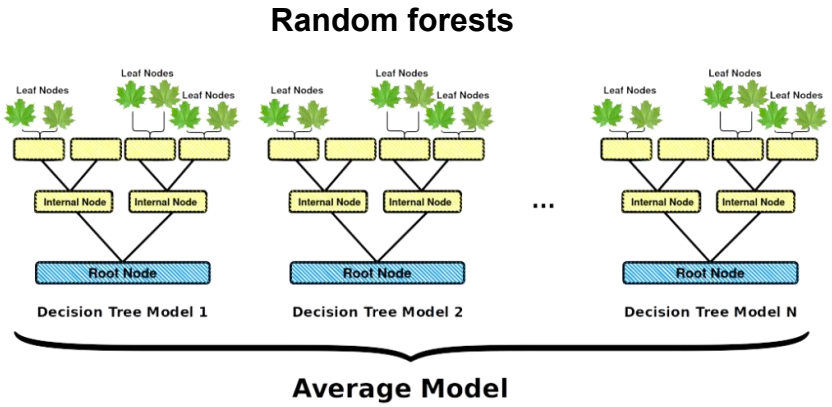












https://kapernikov.com/traditional-machine-learning-algorithms-for-machine-vision/

Les algorithmes d'apprentissage supervisé

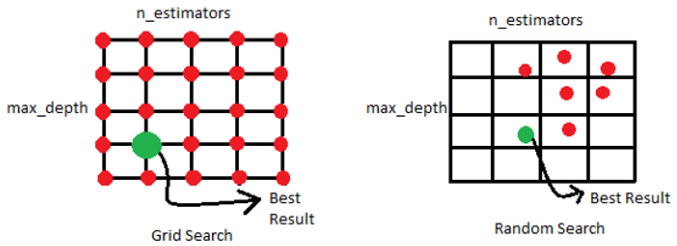
⚙️ Hyperparamètres

- Définis avant l'entraînement par l'utilisateur
- Contrôlent le fonctionnement global de l'algorithme
- Ne sont pas appris automatiquement

Exemples **:**

- Taux d'apprentissage (**learning rate**)
- Nombre d'arbres dans une forêt aléatoire
- Nombre de neurones par couche dans un réseau de neurones

Outils : **Grid Search**, **Random Search**

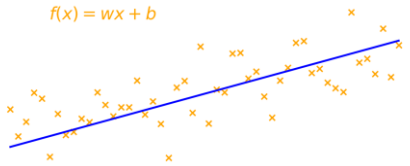


📈 Paramètres entraînables

- Estimés automatiquement pendant l'entraînement
- Ajustés pour minimiser l'erreur entre prédictions et étiquettes réelles
- Dépendent directement des données d'apprentissage

Exemples **:**

- Coefficients d'une régression linéaire
- Poids et biais d'un réseau de neurones



Entraînement d'un modèle

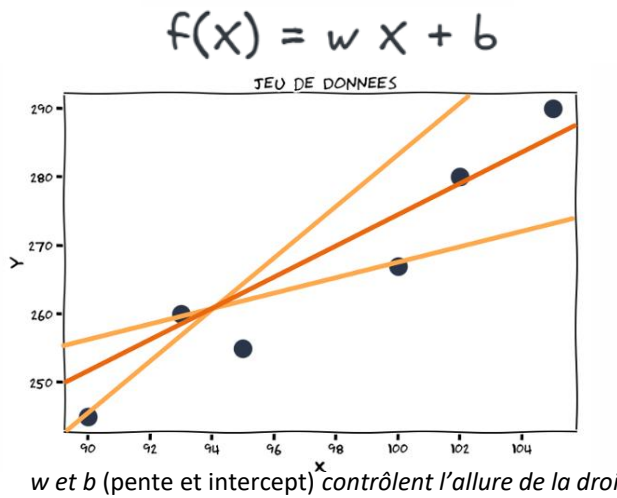
Exemple d'entraînement d'un modèle simple de régression linéaire

Un algorithme peut apprendre à prédire une **valeur de sortie** à partir de **caractéristiques d'entrée**.
Cet exemple illustre le principe général d'apprentissage supervisé appliqué à un modèle de **régression linéaire simple**.
L'idée est d'ajuster une droite qui passe au mieux parmi les points du jeu de données, afin de minimiser l'erreur entre les valeurs prédites et les valeurs réelles.

Fonctionnement

- **Entrées** : X(valeurs d'entrée / caractéristiques)
- **Sorties réelles** : Y(valeurs observées)
- **Sorties prédites** : $\hat{Y}$
- **Fonction de prédiction linéaire** :

$$\boldsymbol{f(x) = \hat{Y} = WX + b}$$



w et b (pente et intercept) *contrôlent l'allure de la droite*

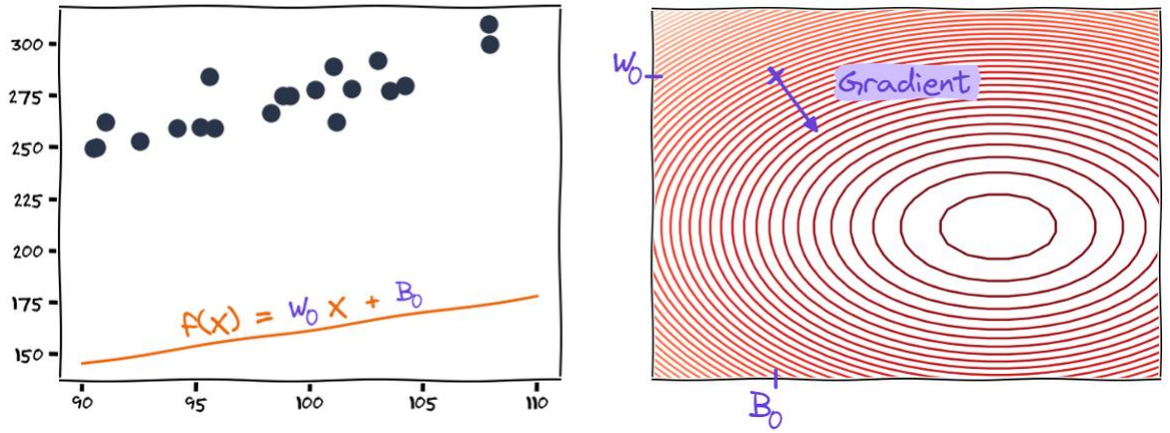
➢ **Objectif : trouver les paramètres Wet b (paramètres entraînables) qui permettent à la droite de mieux s'ajuster aux données.**

📝 Cette présentation est un exemple simple de régression linéaire. D'autres algorithmes supervisés peuvent utiliser des modèles plus complexes et des fonctions de prédiction non linéaires.

Entraînement d'un modèle

Étapes de l'entraînement

1. Initialisation des paramètres



L'algorithme démarre avec des paramètres (w_0, b_0) choisis au hasard, puis les ajuste progressivement pour réduire l'erreur.

Entraînement d'un modèle

Étapes de l'entraînement

1. Initialisation des paramètres

Les paramètres du modèle (W et b) sont généralement initialisés de manière aléatoire.

2. Propagation en avant (Forward Pass)

Le modèle calcule les prédictions à partir des entrées :

$$\hat{Y} = WX + b$$

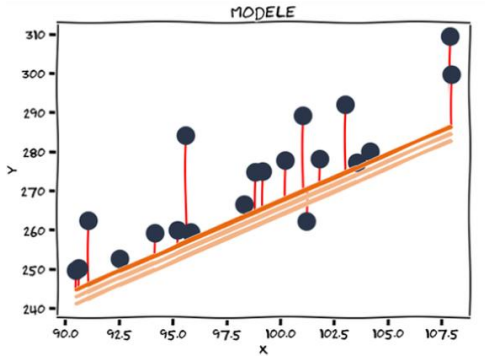
3. Calcul de l'erreur (fonction de coût)

On mesure l'écart entre les valeurs prédites $\hat{y}$ et les valeurs réelles y à l'aide d'une fonction de coût.

Exemple : **Erreur quadratique moyenne (MSE)**

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

.



- **Si $J(w,b)$ est élevé, cela signifie que les prédictions sont éloignées des valeurs réelles.**
- **L'objectif de l'apprentissage est de réduire cette erreur en ajustant les paramètres Wet b**

Entraînement d'un modèle

Étapes de l'entraînement

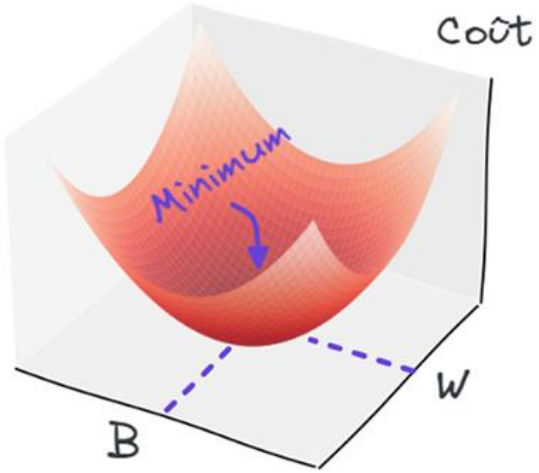
1. Initialisation des paramètres
2. Propagation en avant (Forward Pass)
3. Calcul de l'erreur (fonction de coût)
4. Rétropropagation (Calcul des gradients)
On calcule le gradient de la fonction de coût par rapport à chaque paramètre :

$$\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b}$$

5. Mise à jour des paramètres (Descente de gradient)
On met ensuite à jour les paramètres à l'aide de la descente de gradient, selon les formules :

$$w_{t+1} = w_t - \alpha \frac{\partial J}{\partial w_t} \qquad b_{t+1} = b_t - \alpha \frac{\partial J}{\partial b_t}$$

La descente de gradient est l'un des algorithmes d'apprentissage les plus utilisés en ML. Il consiste à calculer le **gradient de la fonction** coût, c'est-à-dire comment celle-ci évolue lorsque w et b varient légèrement, pour ensuite faire un pas dans la direction où la fonction coût diminue. D'où le nom de **« descente »** de gradient.



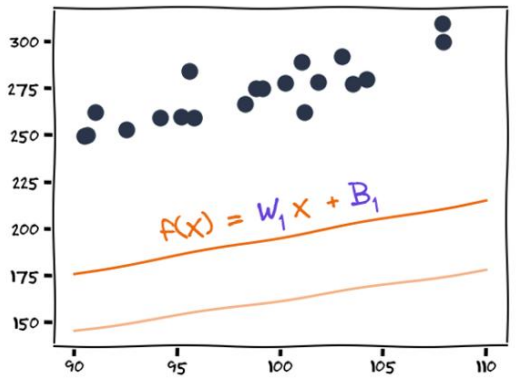
Descente de gradient
https://www.machinelearnia.com/

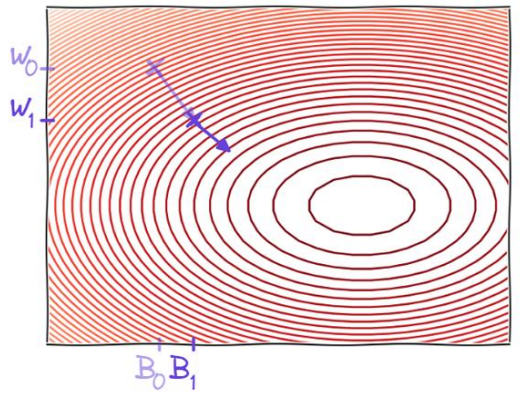
Entraînement d'un modèle

Étapes de l'entraînement

1. **Initialisation des paramètres**
2. **Propagation en avant (Forward Pass)**
3. **Calcul de l'erreur (fonction de coût)**
4. **Rétropropagation (Calcul des gradients)**
5. **Mise à jour des paramètres (Descente de gradient)**

On met ensuite à jour les paramètres à l'aide de la descente de gradient, selon les formules:





$$w_{t+1} = w_t - \alpha \frac{\partial J}{\partial w_t} \qquad b_{t+1} = b_t - \alpha \frac{\partial J}{\partial b_t}$$

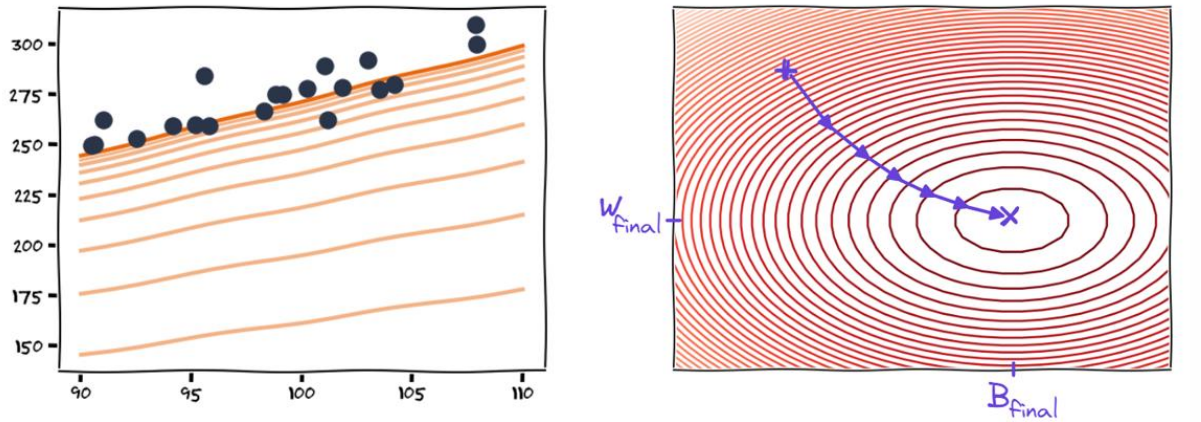
où α est le taux d'apprentissage (learning rate), un hyperparamètre qui définit la vitesse de convergence de l'algorithme

https://www.machinelearnia.com/

Entraînement d'un modèle

Étapes de l'entraînement

1. Initialisation des paramètres
2. Propagation en avant (Forward Pass)
3. Calcul de l'erreur (fonction de coût)
4. Rétropropagation (Calcul des gradients)
5. Mise à jour des paramètres (Descente de gradient)
6. Itération



Itération : On répète les étapes 2 à 5 jusqu'à convergence du modèle (erreur suffisamment faible) ou jusqu'à atteindre un nombre maximal d'itérations.

https://www.machinelearnia.com/

Apprentissage Supervisé

- **Régression**

Prédit une **valeur continue**

Exemples : prix d'une maison, température, revenus

- **Classification :**

La sortie est une **catégorie ou une classe (discrète)**

Types de classification :

- **Classification binaire**
- **Classification multi-classes**
- **Classification multi-labels**
- **Classification multi-outputs**

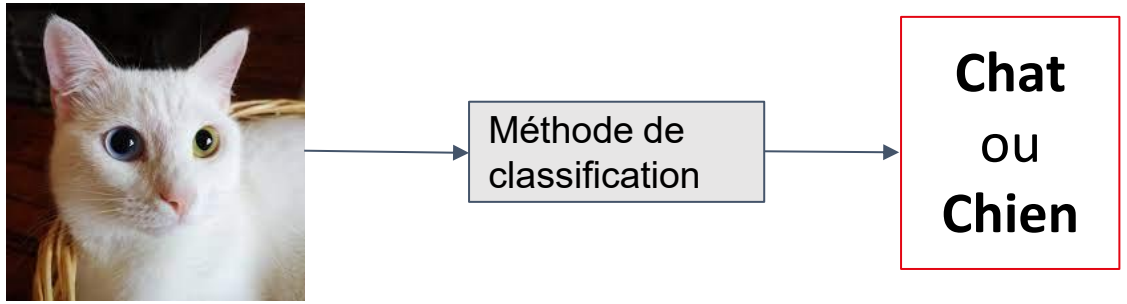
Régression
Cours1/TP1

Classification
Cours2/TP2

Apprentissage Supervisé : Classification

Types de classification

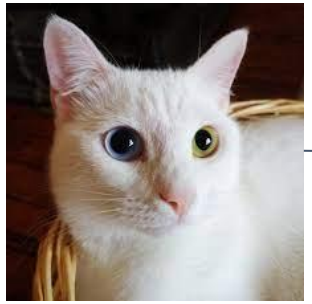
Classification binaire = une parmi 2 classes par instance



Types de classification

Classification binaire = une parmi 2 classes par instance

Classification multi-classes = une parmi plusieurs classes par instance



Méthode de classification

Chat
ou
Chien
ou
Tigre
ou
Lion

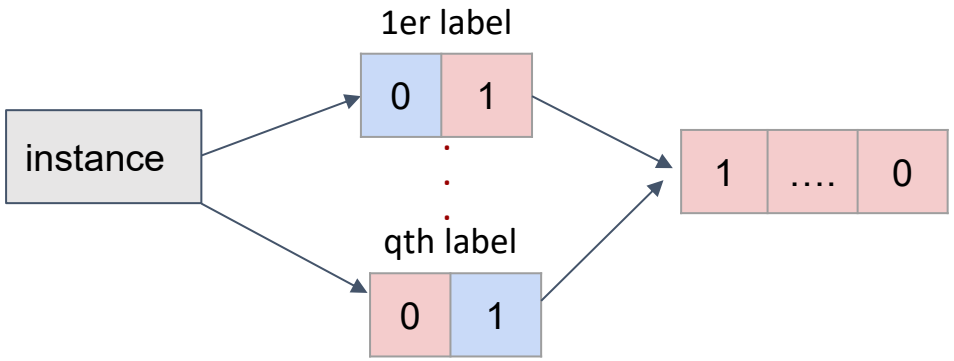
Types de classification

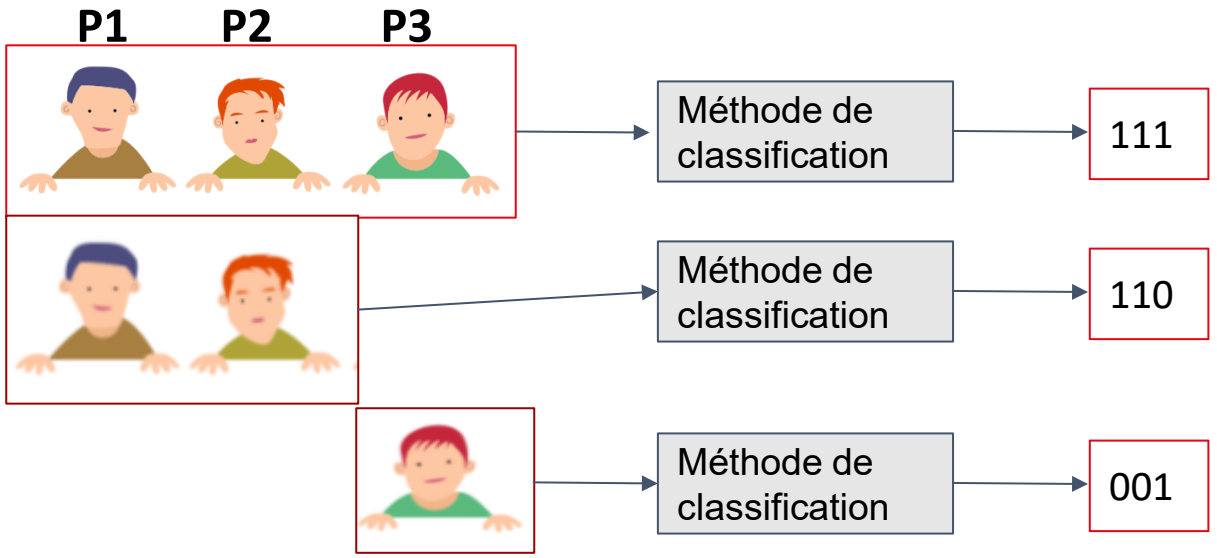
- **Types de méthodes de classification** :

Classification binaire = une parmi 2 classes par instance

Classification multi-classes = une parmi plusieurs classes par instance

Classification multi-labels = plusieurs labels binaires par instance



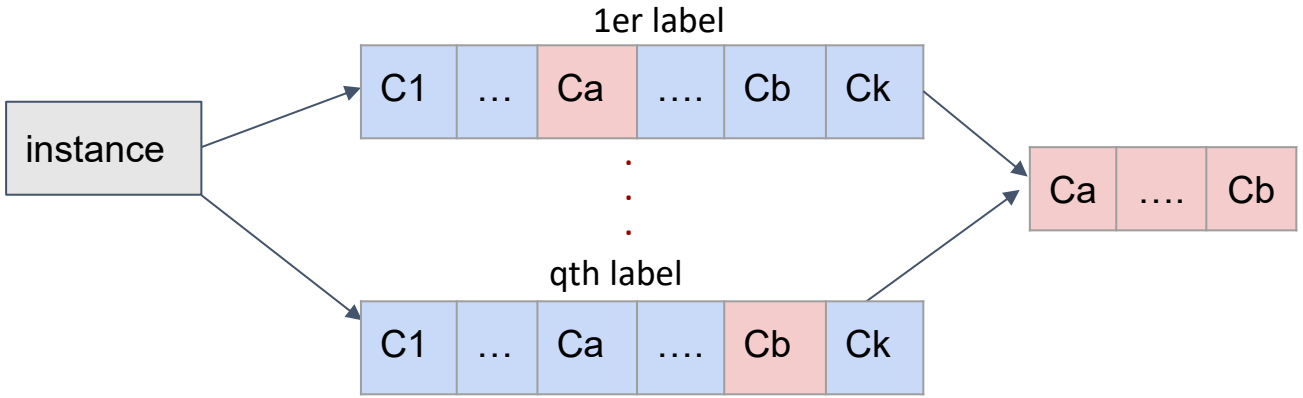


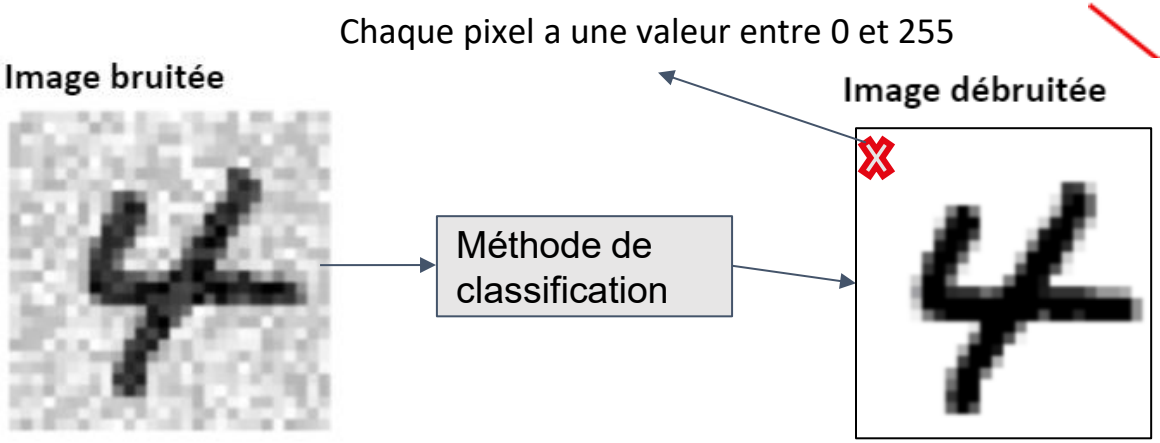
Types de classification

Classification binaire = une parmi 2 classes par instance

Classification multi-classes = une parmi plusieurs classes par instance

Classification multi-outputs = plusieurs labels multi-classes par instance



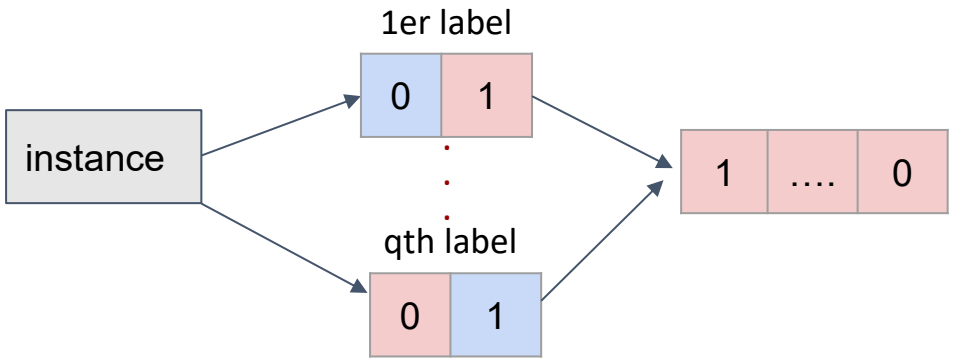


Types de classification

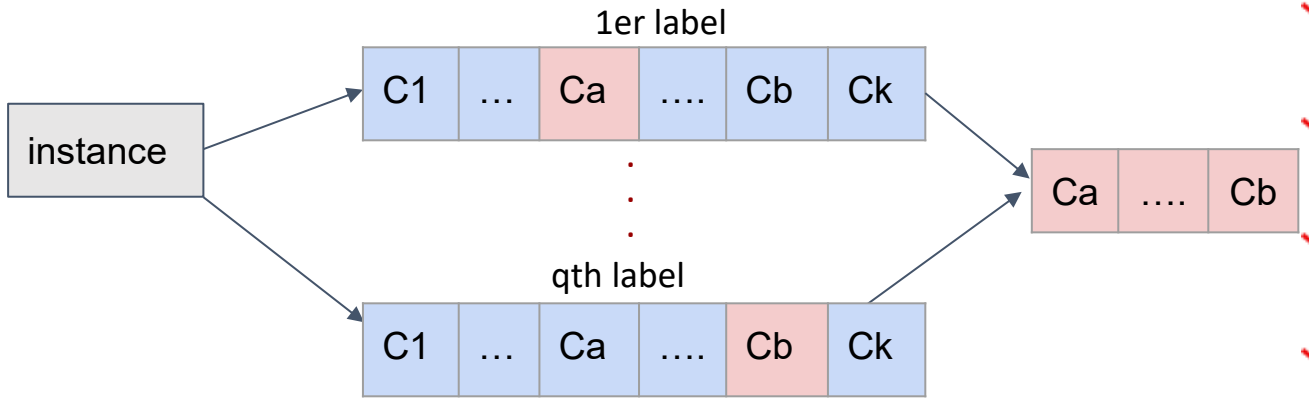
Classification binaire = une parmi 2 classes par instance

Classification multi-classes = une parmi plusieurs classes par instance

Classification multi-labels = plusieurs labels binaires par instance



Classification multi-outputs = plusieurs labels multi-classes par instance

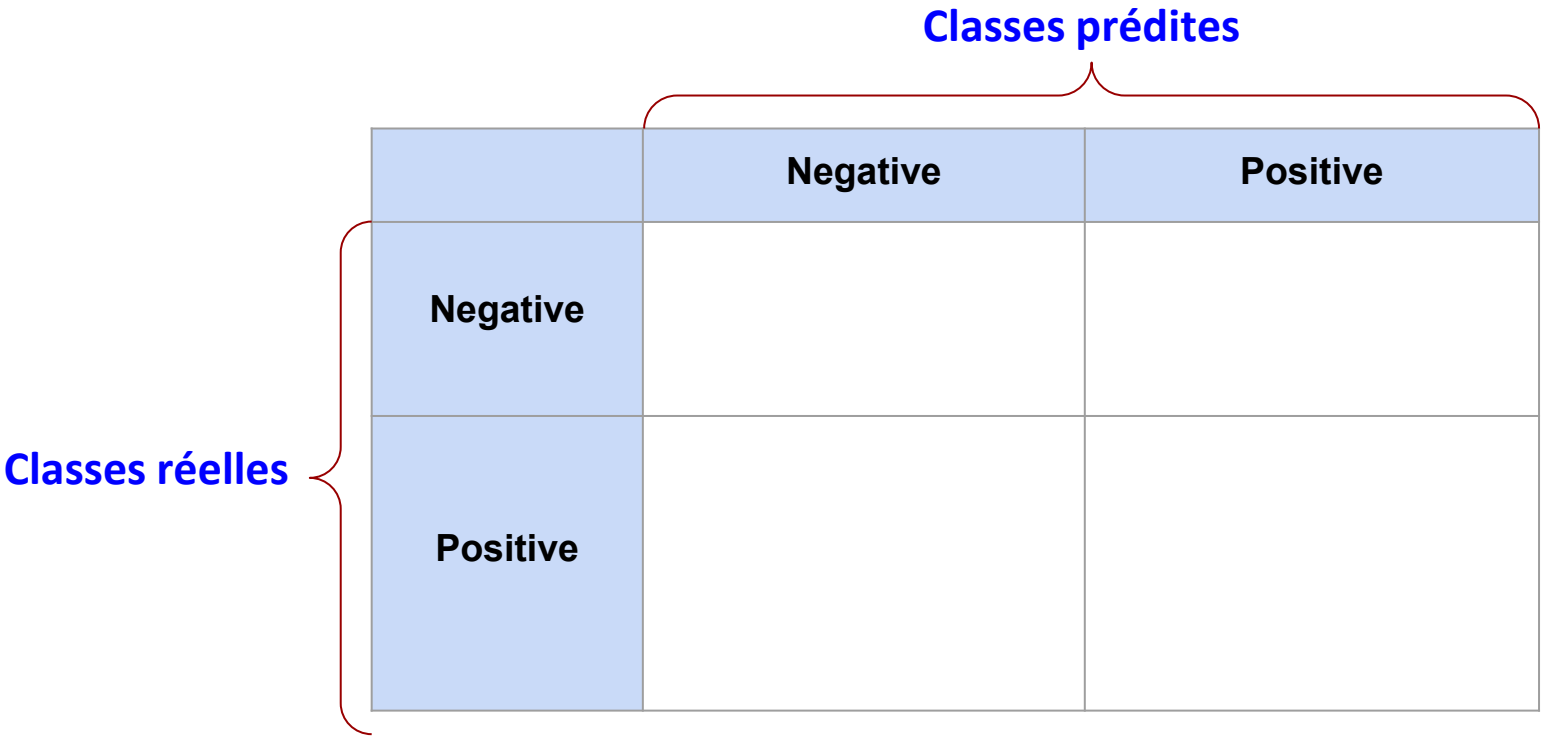


Classification

- **Types de méthodes de classification** :
  - Classification binaire
  - Classification multi-classes
  - Classification multi-labels
  - Classification multi-outputs

- **Quelques méthodes de classification** :
  - Algorithme du gradient stochastique
  - Support Vector Machines
  - K-plus proche Voisin
  - …

Evaluation des méthodes de classification



Matrice de confusion d'une classification binaire

Positif = Le modèle détecte la présence d'une caractéristique (ex : "malade", "spam", "défaut").
Négatif = Le modèle détecte l'absence de cette caractéristique (ex : "sain", "non spam", "pas de défaut").

Evaluation des méthodes de classification

- **Matrice de confusion d'une classification binaire**

Classes prédites

Classes réelles	Negative	Positive
Negative	✅ **True Negative (TN)** *Le modèle a bien prédit "négatif"*	❌ **False Positive (FP)** *Le modèle a prédit "positif", mais c'était "négatif"*
Positive	❌ **False Negative (FN)** *Le modèle a prédit "négatif", mais c'était "positif"*	✅ **True Positive (TP)** *Le modèle a bien prédit "positif"*

Evaluation des méthodes de classification

- **Métriques calculées à partir de la matrice de confusion :**

Classes prédites

Classes réelles	Negative	Positive	
Negative	True Negative (TN)	False Positive (FP)	$F_1 = 2 \times \frac{précision \times rappel}{précision + rappel}$
Positive	False Negative (FN)	True Positive (TP)	$rappel = \frac{TP}{TP+FN}$
		$précision = \frac{TP}{TP+FP}$	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

Evaluation des méthodes de classification

Accuracy (Exactitude) : Mesure la proportion de prédictions correctes (positives et négatives) parmi l'ensemble des prédictions.

Parmi toutes les prédictions effectuées, combien sont correctes ?

$$\mathbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Objectif : Maximiser la proportion des prédictions correctes.

Evaluation des méthodes de classification

Accuracy (Exactitude) : Mesure la proportion de prédictions correctes (positives et négatives) parmi l'ensemble des prédictions.

Parmi toutes les prédictions effectuées, combien sont correctes ?

Exemple : Test COVID

Classes réelles \ Classes prédites	Negative	Positive
Negative	True Negative (TN) 42	False Positive (FP) 8
Positive	False Negative (FN) 5	True Positive (TP) 25

$$Précision = \frac{TP + TN}{TP + TN + FP + FN} = \frac{67}{80} = 83{,}75\%$$

Le test est correct dans 83,75% des cas, toutes classes confondues.

Limites de l'Accuracy

L'accuracy peut être trompeuse lorsque les **classes sont déséquilibrées.** Un modèle peut obtenir une accuracy élevée en favorisant systématiquement la classe majoritaire.

Exemple : Sur 100 cas avec 95 négatifs et 5 positifs, un modèle qui prédit toujours "négatif" aurait :

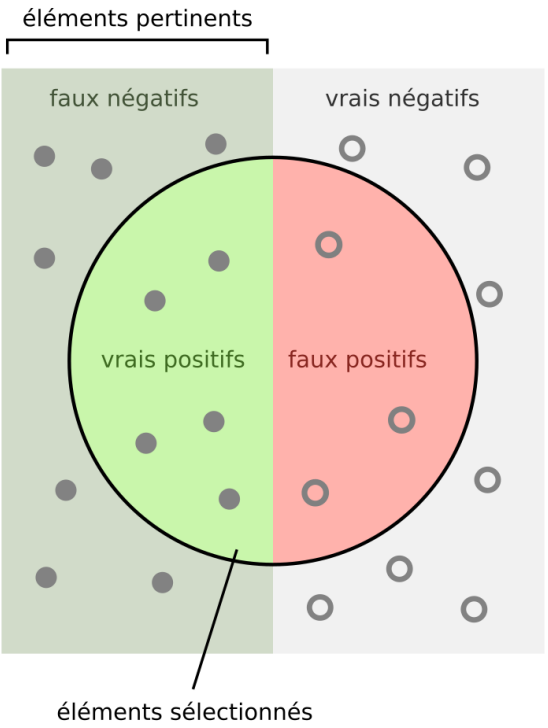
Accuracy = 95%, mais la détection des positifs = 0%

→ **Métrique inadaptée pour évaluer la performance sur la classe minoritaire.**

Evaluation des méthodes de classification

Précision : Parmi toutes les prédictions positives (correctes TP ou incorrectes FP), combien sont réellement correctes ?

$$Précision = \frac{TP}{TP + FP}$$



Combien
de candidats sélectionnés
sont pertinents ?

Précision = $\frac{\text{vrais positifs}}{\text{éléments sélectionnés}}$

Interprétation : Une haute précision indique peu de **faux positifs (FP)**.
Exemple (domaine médical) : Un faux positif correspond à un patient sain classé à tort malade → stress et examens inutiles.

Evaluation des méthodes de classification

Précision : Parmi toutes les prédictions positives (correctes TP ou incorrectes FP), combien sont réellement correctes ?

Exemple : Test COVID

Classes réelles \ Classes prédites	Negative	Positive
Negative	True Negative (TN) 42	False Positive (FP) 8
Positive	False Negative (FN) 5	True Positive (TP) 25

$$\text{Précision} = \frac{TP}{TP + FP} = \frac{25}{25 + 8} = 75,76\%$$

75,76% des tests prédits positifs sont réellement positifs. Les 24,24% restants sont des **faux positifs** (personnes saines classées à tort comme malades), entraînant stress et examens inutiles.

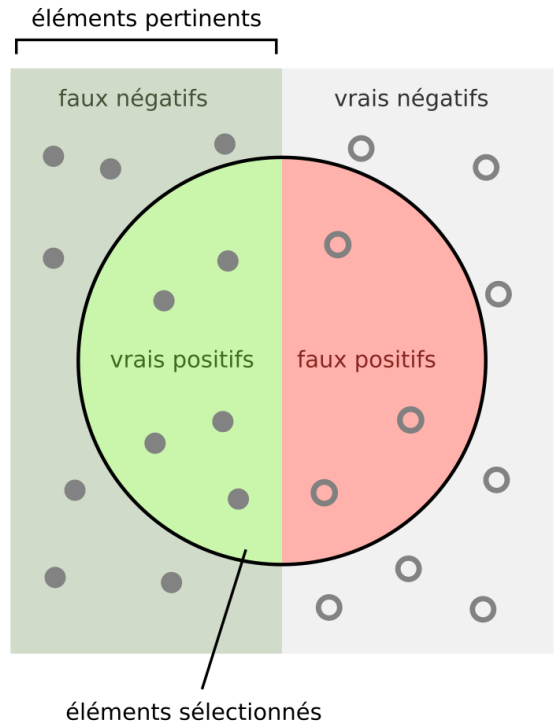
Limites de la précision

La précision ignore les **faux négatifs (FN)**.
Dans le contexte COVID : les faux négatifs correspondent à des patients malades non détectés → risque de propagation de la maladie.

Evaluation des méthodes de classification

Rappel (Sensibilité) : Parmi tous les cas positifs réels, combien sont correctement détectés ?

$$Rappel = \frac{TP}{TP + FN}$$



Combien d'éléments pertinents sont sélectionnés ?

$$Rappel = \frac{\text{■}}{\text{■}}$$

Interprétation : Un rappel élevé indique peu de **faux négatifs (FN)**.
Exemple (domaine médical) : Un faux négatif correspond à un patient malade non détecté → risque pour la santé.

Evaluation des méthodes de classification

Rappel (Sensibilité) : Parmi tous les cas positifs réels, combien sont correctement détectés ?

Exemple : Test COVID

Classes réelles \ Classes prédites	Negative	Positive
Negative	True Negative (TN) 42	False Positive (FP) 8
Positive	False Negative (FN) 5	True Positive (TP) 25

$$\text{rappel} = \frac{TP}{TP + FN} = \frac{25}{5 + 25} = 83{,}33\%$$

Limites :

Le rappel ne tient pas compte des **faux positifs**.

Dans le contexte COVID : les faux positifs correspondent à des personnes saines classées à tort comme malades → stress et examens inutiles.

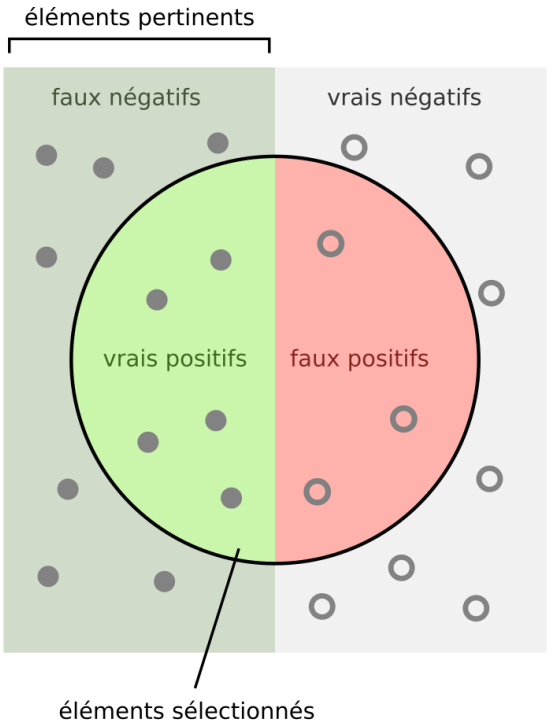
Le test détecte 83,33% des patients malades, mais rate 16,67% de cas. Ces **faux négatifs** (malades classés à tort comme sains) présentent un risque de propagation de la maladie.

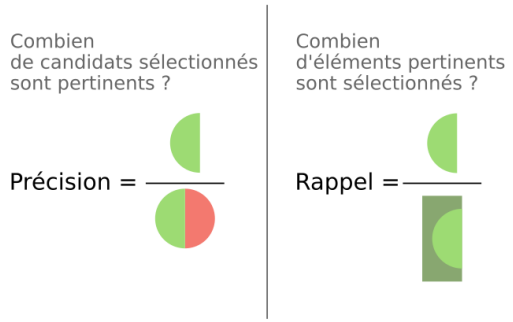
Evaluation des méthodes de classification

F1-Score : combine la **précision** et le **rappel** en une seule métrique.

- **Précision :** proportion des prédictions positives correctes.
- **Rappel :** proportion des cas positifs réels correctement détectés.

$$F1score = 2 \cdot \frac{Précision \cdot Rappel}{Précision + Rappel}$$





Objectif : Équilibrer précision et rappel
Exemple domaine médical : Détecter un maximum de vrais malades tout en limitant les diagnostics erronés

Evaluation des méthodes de classification

F1-Score : combine **précision** et **rappel** en une seule métrique.

Exemple : Test COVID

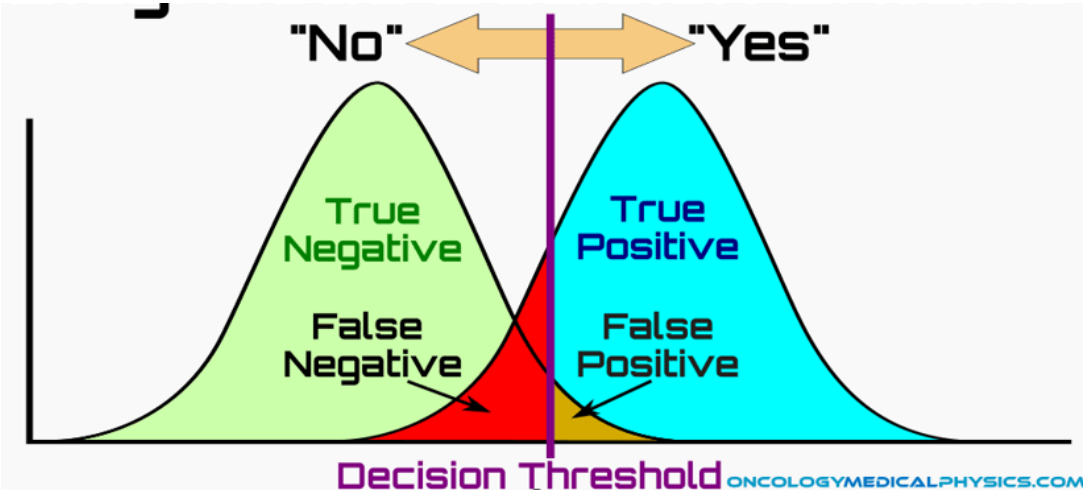
Classes réelles \ Classes prédites	Negative	Positive
Negative	True Negative (TN) 42	False Positive (FP) 8
Positive	False Negative (FN) 5	True Positive (TP) 25

$$F1 - score = 2 \cdot \frac{Précision \cdot Rappel}{Précision + Rappel} = 2 \cdot \frac{75{,}76 \times 83{,}33}{75{,}76 + 83{,}33} = 79{,}37\%$$

Interprétation : Le F1-score de 79,37% représente un équilibre entre la capacité du modèle à éviter les faux positifs (précision) et à détecter les vrais malades (rappel).

Evaluation des méthodes de classification

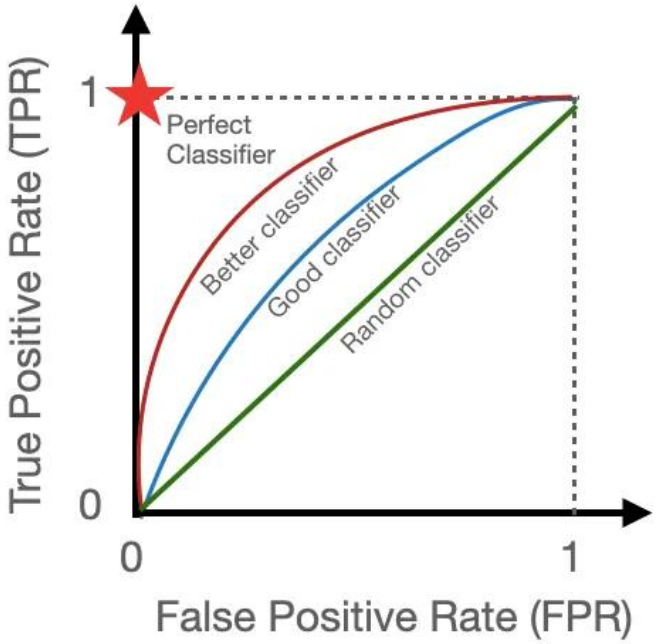
Classification et seuil de décision : Un algorithme de classification attribue une probabilité à chaque instance afin de déterminer son appartenance à une classe. Cette probabilité est ensuite comparée à un seuil de classification pour assigner une catégorie.



Le seuil de décision sépare les prédictions en deux catégories : **Négative** et **Positive**. Ajuster ce seuil influence l'équilibre entre **faux positifs** et **faux négatifs**, ce qui a un impact sur la performance du modèle.

Evaluation des méthodes de classification

Courbe ROC (Receiver Operating Characteristic) : Visualise le compromis entre la **sensibilité (True Positive Rate - TPR)** et le **taux de faux positifs (False Positive Rate - FPR)** pour différents seuils de classification.
L'**AUC (Area Under the Curve)** permet de comparer facilement plusieurs modèles, indépendamment d'un seuil spécifique.



Sensibilité (TPR - True Positive Rate)

$$TPR = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

Taux de faux positifs (FPR - False Positive Rate)

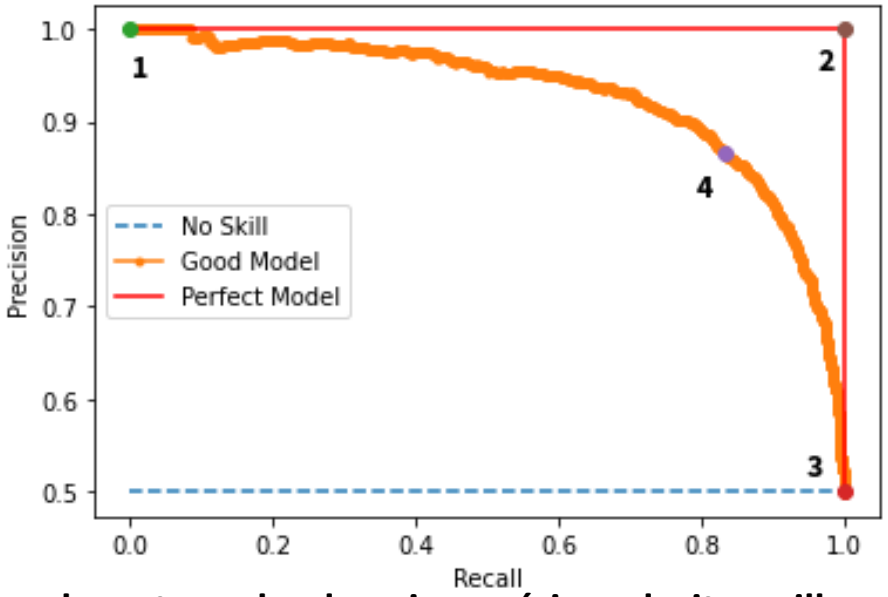
$$FPR = 1 - Specificity = \frac{FP}{FP + TN}$$

Interprétation de la courbe ROC

- **Classifieur aléatoire** : prédit les classes au hasard (diagonale)
- **Classifieur parfait** : coin supérieur gauche ($\mathrm{TPR} = 1$ et $\mathrm{FPR} = 0$)
- **Plus la courbe se rapproche du coin supérieur gauche, meilleure est la performance**
- Plus l'AUC est proche de 1, meilleur est le modèle

Evaluation des méthodes de classification

La courbe Precision-Recall (PR) : Visualise le compromis entre la **Précision** et le **Rappel** pour différents seuils de classification.
Particulièrement utile pour les jeux de données déséquilibrés, car ces métriques restent informatives malgré le déséquilibre des classes.



$$Précision = \frac{TP}{TP + FP}$$

$$Rappel = \frac{TP}{TP + FN}$$

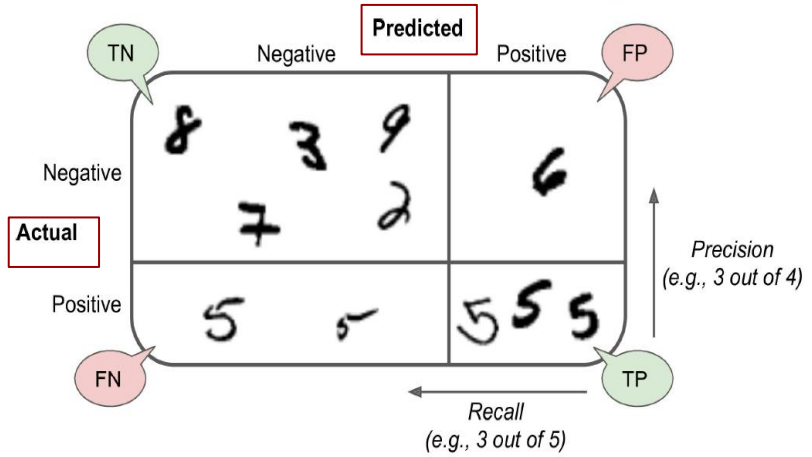
Plus la courbe est proche du coin supérieur droit, meilleur est le modèle
Compromis Précision-Rappel

- Augmenter le rappel → souvent diminue la précision
- Augmenter la précision → souvent diminue le rappel
- **Average Precision (AP)** : Aire sous la courbe PR, indicateur global de performance.

TP 2
Apprentissage supervisé - Classification

Evaluation des méthodes de classification

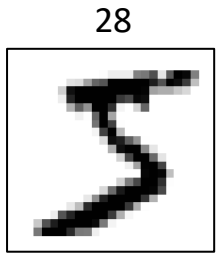
- **Exemple de 5-detector (classification binaire) :**



	classe Non-5	classe 5	
classe Non-5	**5**	**1**	
classe 5	**2**	**3**	**Rappel (Recall)** = 3/5 = 60 % → Le classifieur détecte 60% des instances comme 5
		Précision (Precision) = 3/4 = 75 % → Quand le classifieur prétend qu'une image représente un 5, elle n'est correcte que 75% du temps.	**Taux de classification (Accuracy)** = 8/11 = 72.73%

TP2

Base de données : MNIST



28*28 = 784 pixels

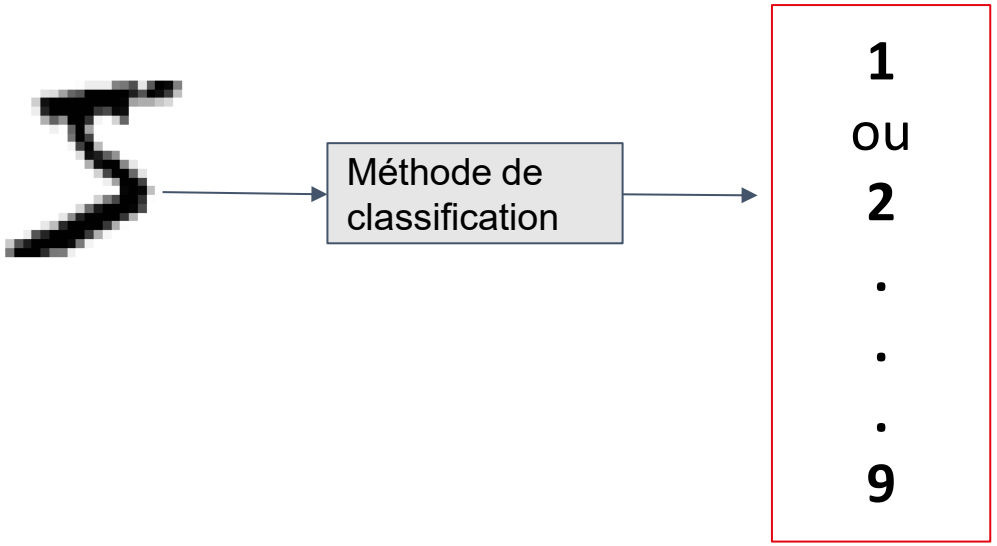
	Classes = Labels	Features		
Instance 1	1	attribut 1	……..	attribut 784
.	…			
.				
.				
Instance 70 000	9			

Y

X

TP2

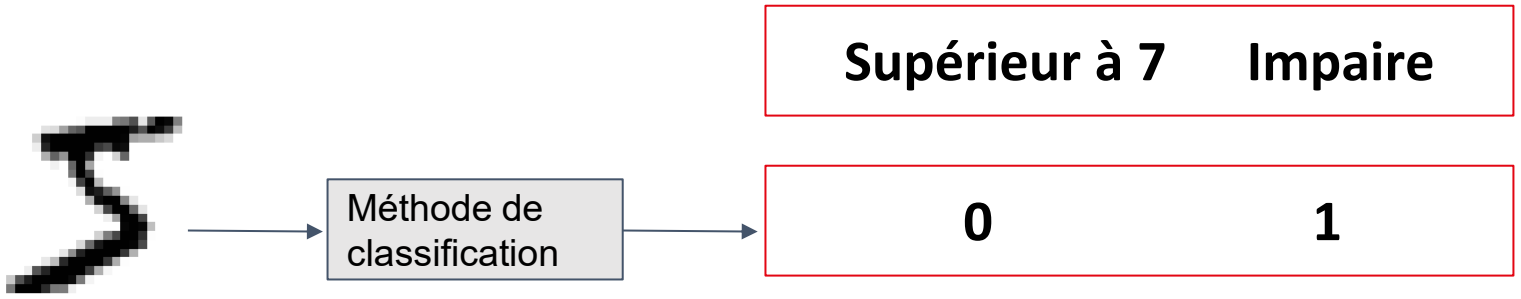
1. Classification multi-classe



TP2

1. Classification multi-classe

2. **Classification multi-label**



TP2

1. Classification multi-classe
2. Classification multi-label
3. **Classification multi-output**