

Introduction au Machine Learning

Un cours sur les fondements, les types et les applications de l'apprentissage automatique

Table des matières

- 1 Introduction au Machine Learning
- 2 Les Types d'Apprentissage
- 3 Classification
- 4 Régression
- 5 Evaluer un modèle
- 6 Défis de l'apprentissage automatique
- 7 Conclusion

Qu'est-ce que le Machine Learning ?

Définition (Arthur Samuel, 1959)

« [L'apprentissage automatique est la] discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés.»

Définition (Tom Mitchell, 1997)

« Étant donné une tâche **T** et une mesure de performance **P**, on dit qu'un programme informatique apprend à partir d'une expérience **E** si les résultats obtenus sur **T**, mesurés par **P**, s'améliorent avec l'expérience **E**.»

Exemple : Le filtre anti-spam

- **Tâche (T)** : Identifier les e-mails frauduleux (spam) parmi les nouveaux e-mails.
- **Expérience (E)** : Les données d'entraînement (exemples de spam et de "ham" signalés par les utilisateurs).
- **Mesure (P)** : Le pourcentage d'e-mails correctement classés (l'exactitude ou "accuracy").
- **Modèle** : La partie du système qui apprend et fait les prédictions (ex : un réseau de neurones, une forêt aléatoire).

Pourquoi utiliser le Machine Learning ?

Approche Traditionnelle (Ex : Filtre Spam)

- ① Examiner les spams (repérer "gratuit", "carte bancaire"....).
- ② Écrire des règles explicites (if "gratuit" in subject, then spam).
- ③ Tester et ajuster les règles manuellement.
- ④ Répéter à l'infini.

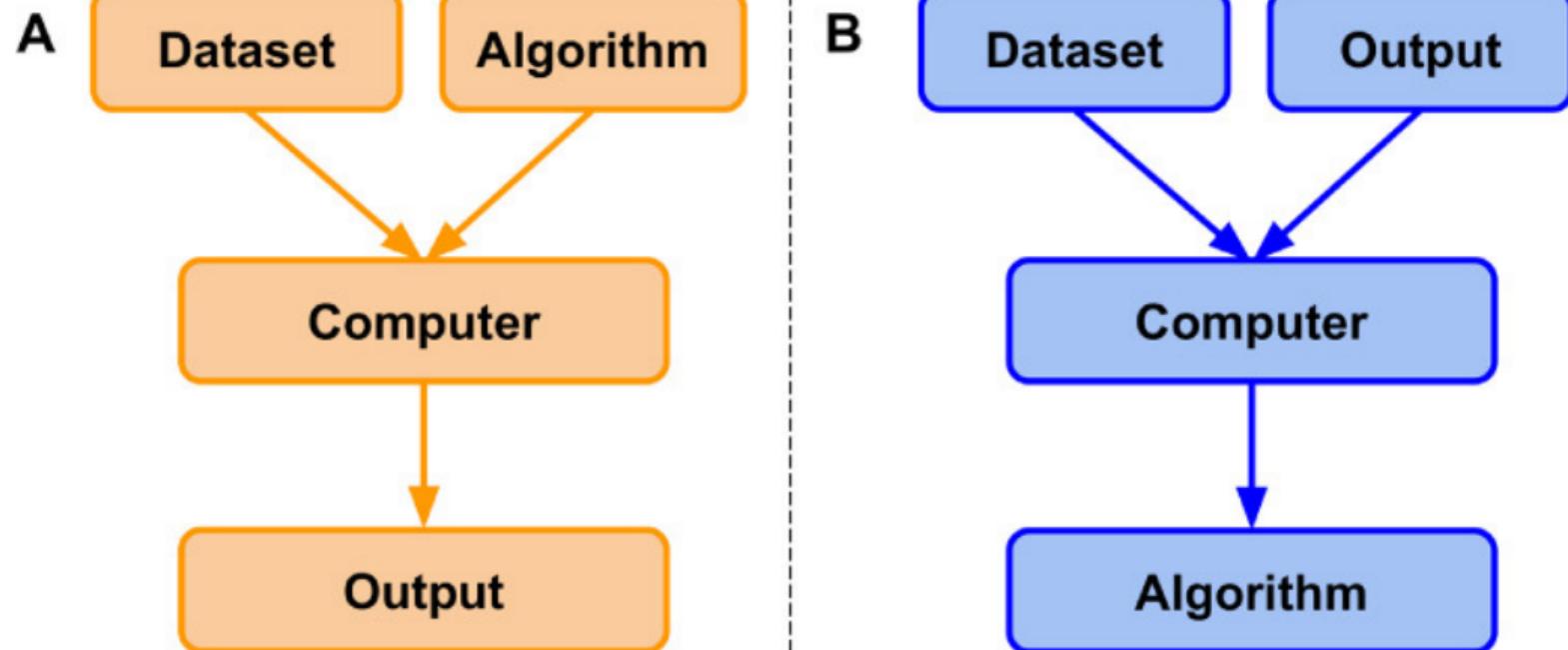
Problème : Devient une liste de règles complexes, difficile à maintenir.

Approche Machine Learning

- ① Fournir des exemples de spam et de "ham" à un algorithme.
- ② Le modèle apprend *automatiquement* les mots et associations qui prédisent le mieux le spam.

Avantage : Plus court, plus facile à maintenir, souvent plus performant.

Approche Traditionnelle vs Machine Learning



Résumé : Quand utiliser le ML ?

L'apprentissage automatique est excellent pour :

- Les problèmes nécessitant beaucoup d'ajustements fins ou de **longues listes de règles** (ex : détection de spam).
- Les **problèmes complexes** pour lesquels aucune solution traditionnelle n'existe (ex : reconnaissance vocale, analyse d'image).
- Les **environnements fluctuants** (le système peut se ré-entraîner et s'adapter à de nouvelles données).
- L'exploration de **gros volumes de données** pour y découvrir des structures cachées (Data Mining).

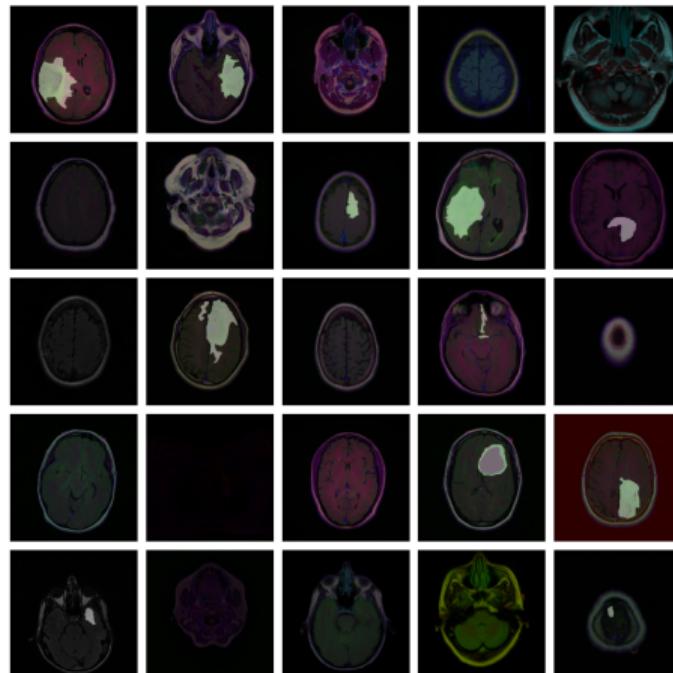
Exemples d'Applications : Classification d'images



Classification d'images

- **Exemple** : Tri de produits sur une chaîne de production
- **Techniques** : CNN, Transformateurs
- Identification automatique d'objets dans des images

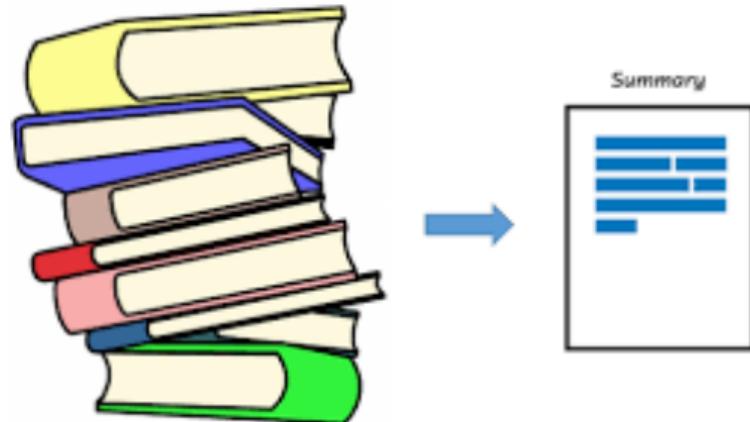
Exemples d'Applications : Segmentation sémantique



Segmentation sémantique

- **Exemple :** Détection de tumeurs sur scanners
- **Techniques :** CNN, Transformateurs
- Identification précise de régions dans des images médicales

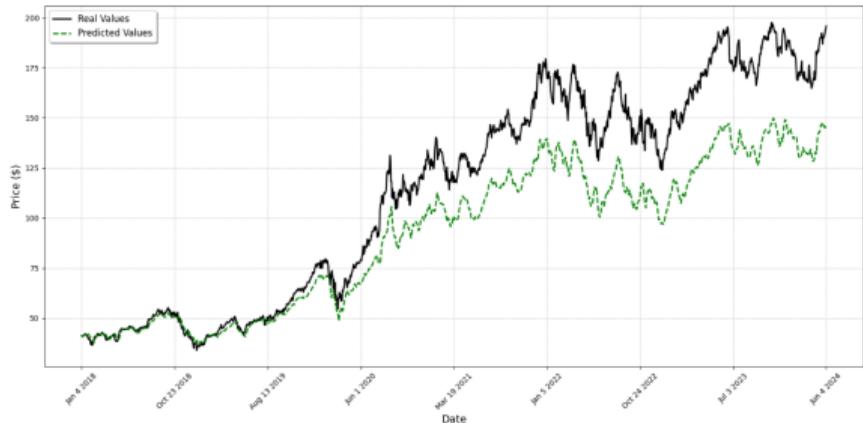
Exemples d'Applications : Traitement du Langage (TALN)



Traitement du Langage (TALN)

- **Exemples** : Classification d'articles, chatbots, résumé auto
- **Techniques** : RNR, Transformateurs
- Compréhension et génération de texte automatique

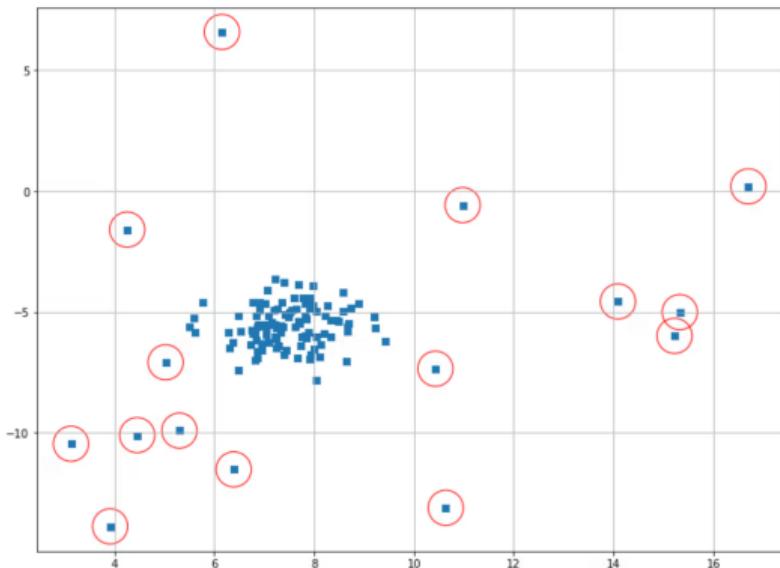
Exemples d'Applications : Régression (Prévision)



Régression (Prévision)

- **Exemple** : Prévoir les résultats financiers
- **Techniques** : Régression linéaire, Forêts aléatoires, RNR
- Prédiction de valeurs numériques continues

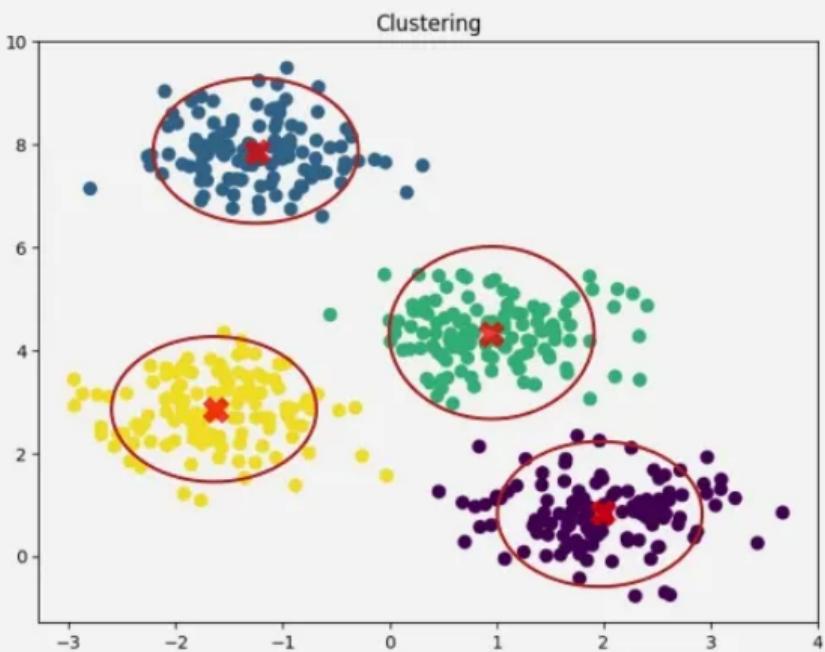
Exemples d'Applications : Détection d'anomalies



Détection d'anomalies

- **Exemple :** Fraude de carte bancaire
- **Techniques :** Forêts d'isolation, Autoencodeurs
- Identification de comportements inhabituels

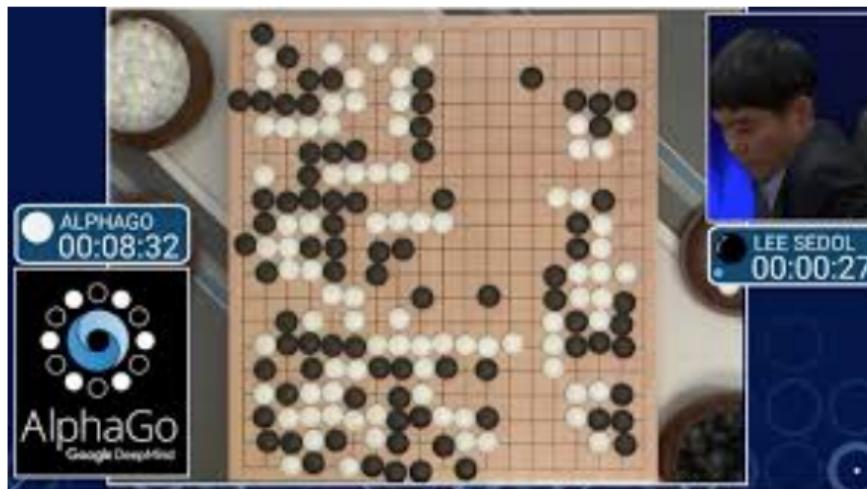
Exemples d'Applications : Partitionnement (Clustering)



Partitionnement (Clustering)

- **Exemple :** Segmentation de clientèle
- **Techniques :** k-moyennes, DBSCAN
- Découverte de groupes naturels dans les données

Exemples d'Applications : Apprentissage par renforcement



Apprentissage par renforcement

- Exemple : IA pour les jeux comme AlphaGo
- Apprentissage par interaction avec l'environnement
- Optimisation de stratégies par essai-erreur

Table des matières

1 Introduction au Machine Learning

2 Les Types d'Apprentissage

3 Classification

4 Régression

5 Evaluer un modèle

6 Défis de l'apprentissage automatique

7 Conclusion

La principale distinction se fait selon **la nature et l'importance de la supervision humaine** durant l'entraînement.

- ① Apprentissage Supervisé
- ② Apprentissage Non Supervisé

(Il en existe d'autres, comme l'apprentissage semi-supervisé ou par renforcement, mais nous nous concentrerons sur ces deux-là.)

Le système apprend avec un "professeur".

- Les données d'entraînement fournies à l'algorithme **comportent les solutions désirées**.
- Ces solutions sont appelées des **étiquettes** (en anglais, *labels*).
- L'objectif est d'apprendre à prédire l'étiquette pour de nouvelles données.

Tâches de l'Apprentissage Supervisé

1. Classification

- L'étiquette est une **catégorie** ou une classe.
- **Exemple** : Le filtre anti-spam. L'étiquette est "Spam" ou "Ham" (Normal).
- **Exemple** : Reconnaissance de chiffres. L'étiquette est 0, 1, 2, 3, 4, 5, 6, 7, 8, ou 9.

2. Régression

- L'étiquette est une **valeur numérique** continue.
- **Exemple** : Prédire le prix d'une voiture.
- Les données d'entrée (kilométrage, âge, marque) sont les **prédicteurs** ou **caractéristiques** (features).
- La valeur à prédire (prix) est la **cible** (target).

Le système apprend sans "professeur".

- Les données d'apprentissage **ne sont pas étiquetées**.
- Le système essaie de **découvrir des structures ou des motifs cachés** dans les données par lui-même.

- **Partitionnement (Clustering)**

- Déetecter des groupes d'observations similaires.
- Ex : Segmenter les visiteurs d'un blog en groupes (visiteurs du soir, passionnés de SF, etc.).

- **Visualisation / Réduction de dimension**

- Représenter des données complexes de grande dimension en 2D ou 3D pour les humains.
- Simplifier les données sans perdre trop d'information (ex : agréger "âge" et "kilométrage" en "vétusté").

- **Détection d'anomalies (ou de nouveautés)**

- Apprendre à quoi ressemblent des données "normales" pour détecter ce qui est "inhabituel".
- Ex : Transactions de carte bancaire frauduleuses, défauts de fabrication.

Table des matières

- 1 Introduction au Machine Learning
- 2 Les Types d'Apprentissage
- 3 Classification
- 4 Régression
- 5 Evaluer un modèle
- 6 Défis de l'apprentissage automatique
- 7 Conclusion

Définition Formelle

Soit un ensemble de données d'entraînement $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ où :

- $x^{(i)} \in \mathbb{R}^n$ est le vecteur de caractéristiques de la i -ème instance.
- $y^{(i)} \in \mathcal{Y}$ est l'étiquette de classe, où \mathcal{Y} est un ensemble fini et discret de valeurs.

L'objectif est d'apprendre une fonction de prédiction $f : \mathbb{R}^n \rightarrow \mathcal{Y}$ telle que $f(x) \approx y$ pour les nouvelles données.

- **Essentiel** : Contrairement à la régression (où $\mathcal{Y} = \mathbb{R}$), ici la sortie est une **catégorie**.
- **Exemple** : x est un email, $\mathcal{Y} = \{\text{Spam}, \text{Non-Spam}\}$.

Cas d'Étude : Le "Hello World" du ML

L'ensemble de données MNIST

- **Quoi** : 70 000 petites images de chiffres manuscrits.
- **Instances** : 70 000 images.
- **Caractéristiques (Features)** : Chaque image fait 28x28 pixels. Elle est "aplatie" en un vecteur de **784 caractéristiques** (une par pixel).
- **Valeurs** : Intensité du pixel (0 = blanc, 255 = noir).
- **Étiquettes (Classes)** : Le chiffre représenté (de "0" à "9").
- **Problème** : Classification multiclasse (10 classes).
- **Séparation** : 60 000 images d'entraînement, 10 000 images de test (pour évaluer la généralisation).



La Classification Binaire

- Cas particulier où l'ensemble des étiquettes ne contient que deux valeurs : $\mathcal{Y} = \{0, 1\}$ ou $\{-1, 1\}$.
- Souvent, on cherche à détecter la présence d'une classe spécifique (Classe Positive).

Exemple : Le "DéTECTEUR de 5" (depuis MNIST)

- **Classe Positive** : "Est un 5"
- **Classe Négative** : "N'est pas un 5" (donc 0, 1, 2, 3, 4, 6, 7, 8, ou 9).

C'est la base de nombreux systèmes (Spam/Non-Spam, Malade/Sain, etc.).

Table des matières

- 1 Introduction au Machine Learning
- 2 Les Types d'Apprentissage
- 3 Classification
- 4 Régression
- 5 Evaluer un modèle
- 6 Défis de l'apprentissage automatique
- 7 Conclusion

Définition Formelle

Soit un ensemble de données d'entraînement $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ où :

- $x^{(i)} \in \mathbb{R}^n$ est le vecteur de caractéristiques de la i -ème instance.
- $y^{(i)} \in \mathbb{R}$ est la valeur cible continue.

L'objectif est d'apprendre une fonction de prédiction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ telle que $f(x) \approx y$ pour les nouvelles données.

- **Essentiel** : Contrairement à la classification (où \mathcal{Y} est discret), ici la sortie est une **valeur continue**.
- **Exemple** : x est la surface d'une maison, $y \in \mathbb{R}$ est son prix.

Exemple de Régression : California Housing

Le Jeu de Données California Housing

- **Contexte** : Données sur le logement en Californie (recensement de 1990).
- **Objectif (y)** : Prédire la valeur médiane des maisons dans un district (en centaines de milliers de \$).
- **Caractéristiques (x)** : 8 variables, dont :
 - MedInc : Revenu médian du district.
 - HouseAge : Âge médian des maisons.
 - AveRooms : Nombre moyen de pièces par logement.
 - Latitude/Longitude : Localisation géographique.

Pourquoi cet exemple ?

C'est un problème de **régression multiple** classique ($x \in \mathbb{R}^8 \rightarrow y \in \mathbb{R}$). Il montre comment plusieurs facteurs influencent une valeur continue.

Table des matières

- 1 Introduction au Machine Learning
- 2 Les Types d'Apprentissage
- 3 Classification
- 4 Régression
- 5 **Evaluer un modèle**
- 6 Défis de l'apprentissage automatique
- 7 Conclusion

Mesurer la Performance

- Entraîner un modèle, c'est bien.
- Savoir s'il est *bon*, c'est mieux.
- Comment évaluer un classifieur ou un régresseur ?

Métrique 1 : L'Exactitude (Accuracy)

- **Définition** : Le ratio de prédictions correctes.
- **Formule** :

$$\text{Accuracy} = \frac{VP + VN}{VP + VN + FP + FN}$$

- **Intuition** : Si le modèle classe 95 images sur 100 correctement, il a 95% d'exactitude.
- **Problème** : L'exactitude est une métrique trompeuse pour les **jeux de données déséquilibrés**.

Le Piège de l'Exactitude

- **Jeu de données déséquilibré (Skewed dataset)** : Une classe est beaucoup plus fréquente que l'autre.
- **Exemple "DéTECTEUR de 5"** : Dans MNIST, environ 10% des images sont des "5", et 90% sont des "non-5".
- **Expérience de pensée** : Créons un classifieur stupide, le "ClassifieurJamais5", qui prédit toujours "non-5".
- **Performance :**
 - Il aura raison sur tous les "non-5" (90% des données).
 - Il aura tort sur tous les "5" (10% des données).
 - **Son exactitude est de 90% !**
- **Conclusion** : Un score d'exactitude élevé peut être atteint par un modèle totalement inutile. **L'exactitude n'est pas une bonne métrique ici.**

Métrique 2 : La Matrice de Confusion

- Une manière beaucoup plus robuste d'évaluer.
- **Idée** : Compter le nombre de fois où la classe A est classée comme classe B.
- Pour un classifieur binaire (Positif / Négatif) :

	Prédit : Négatif	Prédit : Positif
Réel : Négatif	Vrais Négatifs (VN)	Faux Positifs (FP)
Réel : Positif	Faux Négatifs (FN)	Vrais Positifs (VP)

Comprendre la Matrice de Confusion

- **Vrais Positifs (VP)** : Un "5" (réel) qui a été prédict comme "5" (correct).
- **Vrais Négatifs (VN)** : Un "3" (réel) qui a été prédict comme "non-5" (correct).
- **Faux Positifs (FP)** : Un "3" (réel) qui a été prédict comme "5" (incorrect).
 - *Erreur de Type I.*
- **Faux Négatifs (FN)** : Un "5" (réel) qui a été prédict comme "non-5" (incorrect).
 - *Erreur de Type II.*

Note

Un classifieur parfait n'aurait que des VP et des VN. Les cases FP et FN seraient à zéro.

Métrique 3 : Précision et Rappel

La matrice de confusion nous donne deux métriques bien meilleures :

1. Précision (Precision)

- **Question** : *Parmi toutes les fois où le modèle a dit "5", quel pourcentage de fois avait-il raison ?*
- Mesure l'exactitude des prédictions **positives**.

2. Rappel (Recall) / Sensibilité

- **Question** : *Parmi tous les "5" qui existaient réellement, quel pourcentage le modèle a-t-il réussi à trouver ?*
- Mesure le taux de détection des instances **positives**.

Précision vs. Rappel : L'Intuition

Précision = $\frac{VP}{VP+FP}$ (Focus sur les colonnes "Prédit Positif")

- Haute Précision = Peu de Faux Positifs.
- Crucial quand un FP coûte cher.
- Exemple : Filtre "sûr pour les enfants". On préfère bloquer une vidéo sûre (FN) plutôt que de laisser passer une vidéo dangereuse (FP).

Rappel = $\frac{VP}{VP+FN}$ (Focus sur les lignes "Réel Positif")

- Haut Rappel = Peu de Faux Négatifs.
- Crucial quand un FN coûte cher.
- Exemple : Détection de voleurs. On préfère signaler un innocent (FP) plutôt que de manquer un vrai voleur (FN).

Métrique 4 : Le Score F1

- Il est pratique d'avoir une seule métrique qui combine Précision et Rappel.
- **Score F1** : La **moyenne harmonique** de la précision et du rappel.

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Pourquoi une moyenne harmonique ?

- Elle donne beaucoup plus de poids aux **faibles valeurs**.
- Un modèle n'aura un score F1 élevé que si **la précision ET le rappel sont élevés**.
- *Exemple* : Précision 90%, Rappel 10% → Moyenne simple = 50% (trompeur). Score F1 = 18% (reflète bien le mauvais rappel).

Le Compromis Précision/Rappel

- On ne peut presque jamais avoir les deux à 100%.
 - "**L'augmentation de la précision réduit le rappel, et vice-versa.**"
 - **Comment ça marche ?** Les classifieurs calculent un "score" (ex : "à quel point cette image ressemble à un 5").
 - On applique un **seuil de décision** à ce score pour dire "oui" ou "non".
- ① Augmenter le seuil** (être plus strict) :
- Moins de Faux Positifs → **Précision augmente**.
 - Plus de Faux Négatifs → **Rappel diminue**.
- ② Abaisser le seuil** (être plus laxiste) :
- Moins de Faux Négatifs → **Rappel augmente**.
 - Plus de Faux Positifs → **Précision diminue**.

Le choix du seuil dépend de l'objectif métier (ex : filtre enfant vs. détection voleur).

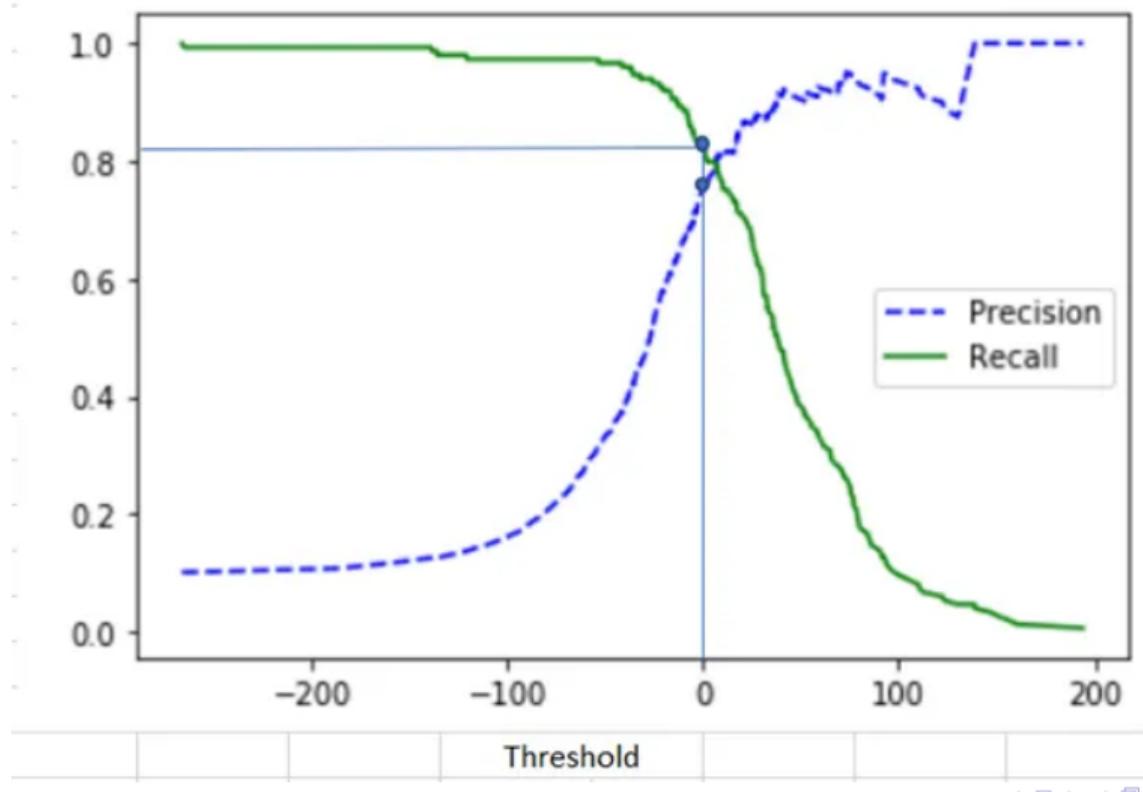
Visualiser : La Courbe Précision-Rappel (PR)

- Pour choisir le meilleur seuil, on peut calculer la précision et le rappel pour *tous* les seuils possibles.
- On trace ensuite la **Précision en fonction du Rappel**.

Avantages

- Permet de choisir le meilleur compromis.
- On cherche souvent le point "juste avant la chute" de la précision.
- Idéalement, la courbe est dans le coin supérieur droit.

Exemple de Courbe Précision-Rappel



Métrique 5 : La Courbe ROC

- Un autre outil très populaire pour les classifiants binaires.
- Trace : **Taux de Vrais Positifs (TVP)** en fonction du **Taux de Faux Positifs (TFP)**.

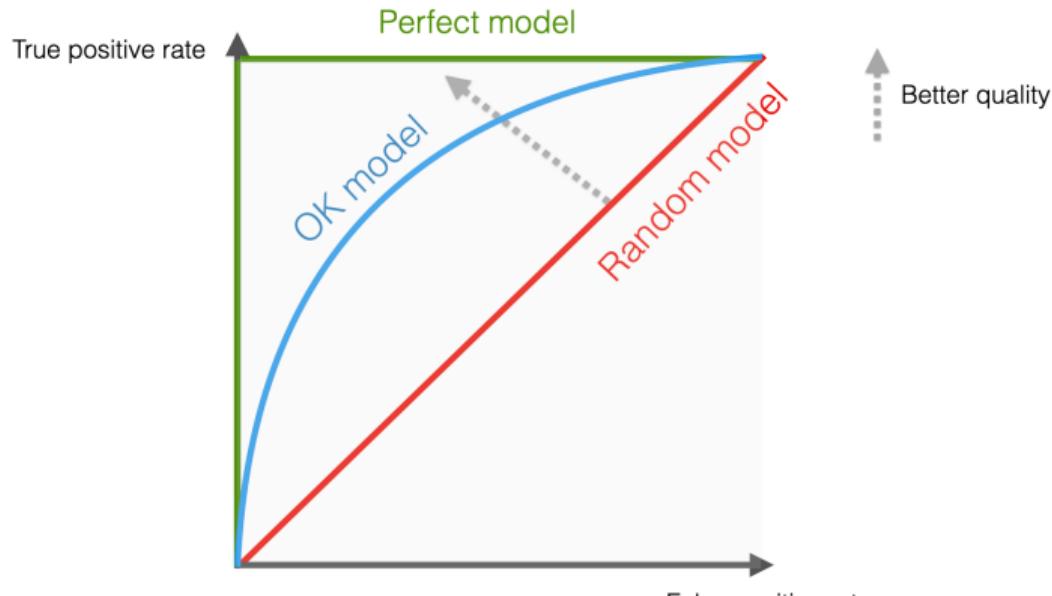
Taux de Vrais Positifs (TVP)

- $\frac{VP}{VP+FN}$
- C'est juste un autre nom pour le **Rappel** (ou Sensibilité).

Taux de Faux Positifs (TFP)

- $\frac{FP}{FP+VN}$
- Le ratio d'instances *négatives* qui sont *incorrectement* classées positives.
- Un classifieur aléatoire suit la diagonale.
- Un classifieur parfait est dans le **coin supérieur gauche** ($TVP = 1$, $TFP = 0$).
- On mesure l'**Aire Sous la Courbe (AUC)** : $1.0 = \text{parfait}$, $0.5 = \text{aléatoire}$.

Exemple de Courbe ROC



Courbe PR vs. Courbe ROC

Quand utiliser quoi ?

- **Courbe ROC** : Très utilisée, mais peut être **trompeuse** si les classes sont très déséquilibrées.
 - *Pourquoi* ? Le TFP (axe X) a les VN au dénominateur. S'il y a énormément de Vrais Négatifs (ex : les "non-5"), le TFP reste très bas même si le nombre de Faux Positifs augmente, donnant une courbe faussement optimiste.
- **Courbe PR (Précision-Rappel)** :
 - **À préférer lorsque la classe positive est rare** (datasets déséquilibrés).
 - *Pourquoi* ? La Précision (axe Y) a les FP au dénominateur. Une augmentation des FP fait chuter la courbe directement et visiblement. Elle donne une image plus honnête de la performance sur des données déséquilibrées.

Mesurer la Performance (Régression)

- En classification, on comptait les succès/échecs (Exactitude, Précision...).
- En régression, on doit quantifier la **magnitude de l'erreur**.
 - Une prédiction de 15 251 € n'est pas "100% fausse" si la cible est 15 250 €. Elle est juste "proche".
- **Objectif** : Mesurer à quel point, en moyenne, les prédictions (\hat{y}) sont éloignées des valeurs réelles (y).

Métriques principales :

- ① RMSE (Root Mean Squared Error)
- ② MAE (Mean Absolute Error)
- ③ R² (Coefficient de Détermination)

Métrique 1a : MSE (Erreur Quadratique Moyenne)

- Pour comprendre le RMSE, il faut d'abord comprendre le MSE.

MSE (Mean Squared Error) :

On calcule la moyenne des *carrés* des erreurs.

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

- **Problème** : L'unité est au carré (ex : "euros carrés"). Difficile à interpréter.

Métrique 1b : RMSE (Root Mean Squared Error)

RMSE (Racine de l'EQM) :

On prend la racine carrée du MSE pour revenir aux unités d'origine.

$$\text{RMSE} = \sqrt{\text{MSE}}$$

- **Interprétation** : C'est l'erreur de prédiction "typique" du modèle (ex : "le modèle se trompe en moyenne de 8 500 €").
- **Caractéristique clé** : Très sensible aux **outliers** (valeurs aberrantes) car l'erreur est élevée au carré avant la moyenne. Une grosse erreur est lourdement pénalisée.

Métrique 2 : MAE (Mean Absolute Error)

- Une alternative robuste au RMSE.

MAE (Erreur Absolue Moyenne) :

C'est la moyenne des *valeurs absolues* des erreurs.

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |\hat{y}^{(i)} - y^{(i)}|$$

- **Interprétation** : Très directe (ex : "en moyenne, les prédictions s'écartent de 5 000 € de la réalité").
- **Caractéristique clé : Robuste aux outliers.** Les grosses erreurs contribuent de manière linéaire, pas quadratique.
- **Quand l'utiliser ?** Idéal si vos données contiennent beaucoup de bruit ou d'erreurs de saisie que vous ne voulez pas voir dominer la métrique.

Métrique 3 : Coefficient de Détermination (R^2)

- Ce n'est pas une métrique d'erreur, mais une métrique de **qualité d'ajustement**.
- **Concept** : Mesure la proportion de la variance de la cible (y) qui est "expliquée" par les caractéristiques (x) du modèle.

Interprétation :

- $R^2 = 1.0$: Le modèle explique 100% de la variance (ajustement parfait).
- $R^2 = 0.0$: Le modèle n'est pas meilleur que de simplement prédire la moyenne de y .
- $R^2 < 0.0$: Le modèle est *pire* que la simple moyenne.

Limitations :

- Le R^2 augmente (ou stagne) toujours lorsque vous ajoutez des caractéristiques, même inutiles.
- (Utiliser le R^2 ajusté pour compenser ce problème).

Diagnostic : L'Analyse des Résidus

- Les métriques (RMSE, R²) donnent un seul chiffre, mais ne disent pas *comment* le modèle se trompe.
- Un **résidu** est simplement l'erreur pour une seule instance : $e_i = y_{\text{réel}}^{(i)} - \hat{y}_{\text{prédit}}^{(i)}$.
- **Outil de diagnostic** : On trace les valeurs prédites (\hat{y}) sur l'axe X et les résidus (e) sur l'axe Y.

Interprétation :

- **Un bon modèle** : Le graphique doit être un nuage de points **aléatoire**, centré sur 0, sans structure. Les erreurs sont du "bruit".
- **Forme de cône** : Hétéroscédasticité (l'erreur augmente quand la prédiction augmente ; le modèle est moins fiable pour les grosses valeurs).
- **Forme de courbe (U)** : Le modèle linéaire manque une relation non-linéaire.

Table des matières

- 1 Introduction au Machine Learning
- 2 Les Types d'Apprentissage
- 3 Classification
- 4 Régression
- 5 Evaluer un modèle
- 6 Défis de l'apprentissage automatique
- 7 Conclusion

Les Principaux Défis du ML

- La tâche principale est de sélectionner un algorithme et de l'entraîner sur des données.
- Les deux principaux écueils sont donc :
 - ➊ De **mauvaises données**
 - ➋ Un **mauvais algorithme**
- Commençons par les défis liés aux données.

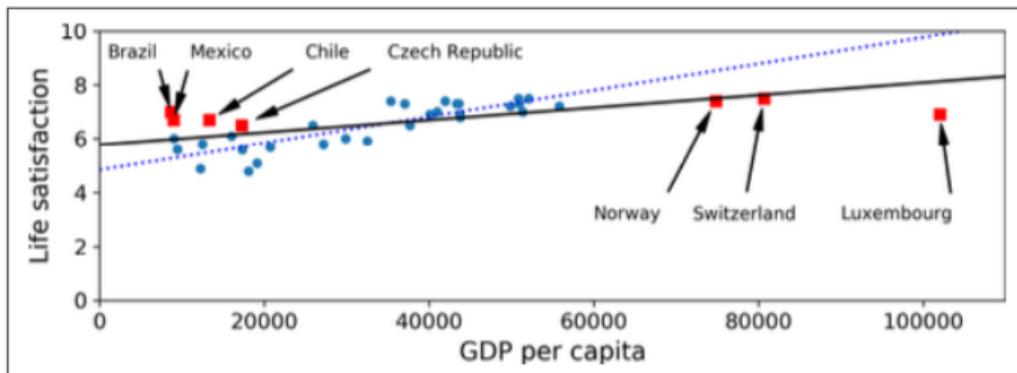
Défi Données 1 : Quantité Insuffisante

- **L'analogie humaine** : Un enfant apprend ce qu'est une "pomme" avec peu d'exemples.
- **La réalité du ML** : La plupart des algorithmes ont besoin de *beaucoup* de données pour fonctionner correctement.
 - **Problèmes simples** : Souvent des milliers d'exemples.
 - **Problèmes complexes** (reconnaissance d'image, parole) : Souvent des millions d'exemples (sauf si l'on réutilise un modèle existant).
- Un manque de données rend difficile la détection de motifs fiables.

Défi Données 2 : Données Non Représentatives

- **Objectif** : Le modèle doit bien *généraliser* à de nouveaux cas inconnus.
- **Condition** : Pour cela, les données d'entraînement *doivent être représentatives* de ces nouveaux cas.
- **Bruit d'échantillonnage** : Si l'échantillon est trop petit, les données peuvent être non représentatives par pur hasard.
- **Biais d'échantillonnage** : Si la *méthode* de collecte est défectueuse, même un grand échantillon sera biaisé.

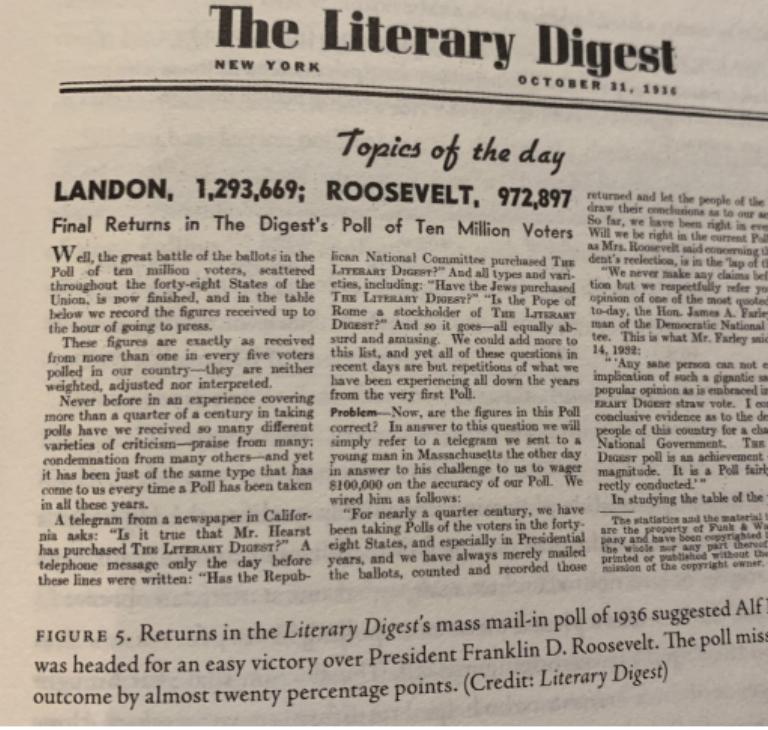
Exemple : Données Non Représentatives



Exemple : Prédire la satisfaction de vie

- Entraîner un modèle (ligne pointillée) sur des données partielles (ex : quelques pays riches) donne un résultat très différent du modèle entraîné sur l'ensemble des données (ligne pleine).

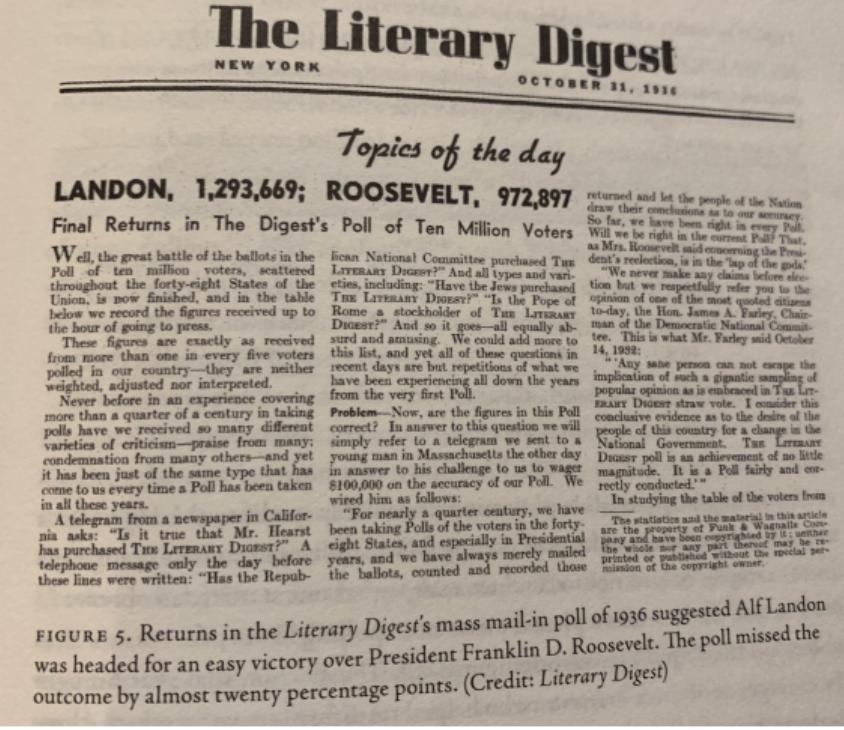
Cas d'Étude : Biais d'Échantillonnage (1936)



Le sondage :

- Le *Literary Digest* envoie 10M de bulletins pour l'élection présidentielle US (Landon vs. Roosevelt).
- La réponse : 2,4M de réponses reçues.
- La prédition : Victoire écrasante de Landon.
- Le résultat : Victoire écrasante de Roosevelt.

Cas d'Étude : Biais d'Échantillonnage (1936) - Pourquoi l'échec ?



Pourquoi l'échec ? Le Biais d'échantillonnage

- 1 Biais de source :** Les adresses provenaient d'annuaires téléphoniques et de listes de membres de clubs. Ces listes favorisaient les personnes plus riches, plus susceptibles de voter Républicain (Landon).
- 2 Biais de non-réponse :** Moins de 25% ont répondu. Ceux qui ont répondu n'étaient pas représentatifs de l'ensemble des votants (ex : excluant les moins politisés).

Défi Données 3 : Données de Mauvaise Qualité

- Votre système aura du mal à apprendre si les données sont pleines d'erreurs, de valeurs aberrantes (outliers) et de bruit.
- Le nettoyage des données ("data cleaning") est une étape cruciale et souvent longue du travail d'un data scientist.

Actions de nettoyage courantes :

- Si des instances sont des **valeurs aberrantes** claires : les supprimer ou les corriger manuellement.
- Si des instances ont des **caractéristiques manquantes** (ex : âge non renseigné) :
 - Ignorer l'instance.
 - Ignorer la caractéristique (si elle manque trop souvent).
 - Combler ("imputer") les valeurs (ex : par l'âge médian).
 - Entraîner des modèles avec et sans la caractéristique.

Défi Données 4 : Caractéristiques Non Pertinentes

- **Adage** : "Garbage In, Garbage Out" (À données médiocres, résultats médiocres).
- Le système ne peut apprendre que si les données contiennent suffisamment de caractéristiques **pertinentes** et pas trop **d'inutiles**.

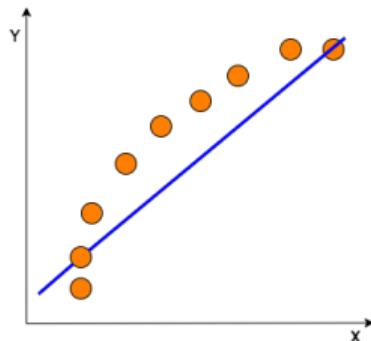
L'Ingénierie des Fonctionnalités (Feature Engineering)

- **Sélection de caractéristiques** : Choisir les plus utiles parmi celles existantes.
- **Extraction de caractéristiques** : Combiner des caractéristiques pour en créer une plus utile (ex : "vétusté" = "âge" + "kilométrage").
- **Création de nouvelles caractéristiques** : Collecter de nouvelles données pour créer de nouvelles variables.

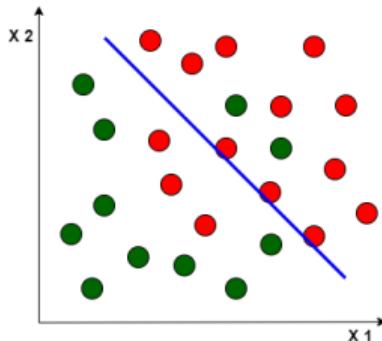
Les Défis Liés à l'Algorithme

- Nous avons vu les problèmes de "mauvaises données".
- Voyons maintenant les problèmes de "mauvais algorithme".
- Les deux défis majeurs sont les extrêmes opposés :
 - ① **Le sous-apprentissage (Underfitting)**
 - ② **Le surapprentissage (Overfitting)**

Défi Algorithme 1 : Le Sous-apprentissage (Underfitting)



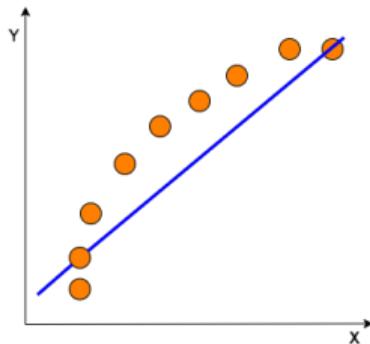
Linear Regression



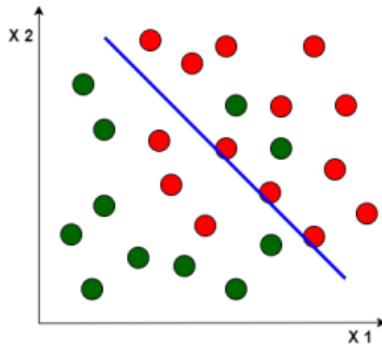
Logistic Regression

- **Définition :** Le modèle est **trop simple** pour apprendre la structure sous-jacente des données.
- **Exemple :** Tenter d'utiliser un modèle linéaire (une ligne droite) pour modéliser une relation non-linéaire (une courbe).
- **Symptôme :** Les prédictions sont mauvaises, **même sur les données d'entraînement**. Le modèle n'arrive tout simplement pas à "coller" aux données.

Défi Algorithme 1 : Le Sous-apprentissage (Underfitting) - Solutions



Linear Regression

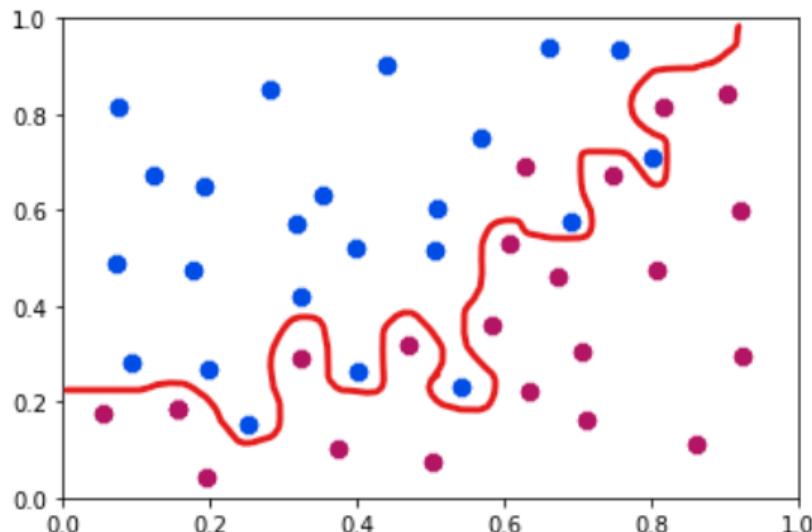


Logistic Regression

Solutions principales :

- ① Sélectionner un modèle plus puissant (ex : passer d'un modèle linéaire à un modèle polynomial).
- ② Fournir de meilleures caractéristiques au modèle (Feature Engineering).
- ③ Réduire les contraintes sur le modèle (ex : réduire la régularisation).

Défi Algorithme 2 : Le Surapprentissage (Overfitting)



- **Définition** : Le modèle fonctionne très bien sur les données d'entraînement, mais il généralise mal aux nouvelles données.
- **Analogie** : L'humain qui, arnaqué par un taxi, conclut que *tous* les taxis de ce pays sont des voleurs (surgénéralisation).
- **Cause** : Le modèle est trop complexe par rapport à la quantité ou au bruit des données.
 - Il n'apprend pas les vrais "signaux" des données, il mémorise le "bruit".
 - Ex : Il trouve un motif (par hasard) que "les pays avec un 'W' dans le nom ont une satisfaction de vie > 7 ".

- **Objectif :** Simplifier le modèle pour éviter qu'il n'apprenne le bruit.

Solutions principales :

① Simplifier le modèle :

- Choisir un modèle avec moins de paramètres (ex : linéaire au lieu de polynomial).
- Réduire le nombre de caractéristiques.
- Contraindre le modèle.

② Rassembler plus de données d'entraînement (pour que le "vrai" signal domine le "bruit").

③ Réduire le bruit dans les données (nettoyer les données).

Solutions au Surapprentissage - La Régularisation

Concept clé : La Régularisation

- C'est le processus de **contraindre un modèle** pour le rendre plus simple et réduire le risque de surapprentissage.
- L'objectif est de trouver le **juste équilibre** entre bien coller aux données d'entraînement et rester simple pour bien généraliser.

Comment ça marche ?

- La régularisation ajoute une pénalité aux paramètres du modèle.
- Plus les paramètres sont grands, plus la pénalité est élevée.
- Cela force le modèle à utiliser des valeurs de paramètres plus petites, ce qui le rend plus simple et moins sujet au surapprentissage.

Table des matières

1 Introduction au Machine Learning

2 Les Types d'Apprentissage

3 Classification

4 Régression

5 Evaluer un modèle

6 Défis de l'apprentissage automatique

7 Conclusion

Conclusion & Questions

Nous avons vu :

- ① Ce qu'est le ML (Tâche, Expérience, Performance).
- ② Les types majeurs (Supervisé vs. Non Supervisé).
- ③ Focus sur la **Classification** (Binaire vs Multiclasse).
- ④ Focus sur la **Régression** (Linéaire Simple vs Multiple).
- ⑤ L'**Évaluation** des modèles (Métriques de Classification et Régression).
- ⑥ Les **Défis Principaux** (Données et Algorithmes).

Des questions ?