# Machine Learning - Chapitre 1

*« Les machines un jour pourront résoudre tous les problèmes, mais jamais aucune d'entre elles ne pourra en poser un ! » - Albert Einstein*

Rémi Vaucher
EPITA - Laboratoire de Recherche de l'EPITA (LRE)
3 novembre 2026

Machine Learning - Chapitre 1 © 2026 Rémi Vaucher is licensed under CC BY 4.0.
To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/.

# 01

# Introduction

# Plan du cours


01 Introduction

02 Cadre formel

03 Généralisation

04 Régularisation

05 Protocole d'évaluation

06 Métriques d'évaluation



licensed under CC BY 4.0

# Qu'est-ce que l'apprentissage automatique ?

**Définition (Mitchell, 1997)**

Un programme apprend d'une expérience $E$ pour une tâche $T$ et une mesure de performance $P$ si sa performance sur $T$, mesurée par $P$, s'améliore avec $E$.

- On ne programme pas la règle de décision : on l'**estime** à partir de données.
- Objectif réel : bien prédire sur des données **jamais vues**.
- Ce n'est pas de l'optimisation pure : bien ajuster les données d'entraînement ne suffit pas.


licensed under CC BY 4.0

# Les grandes familles de tâches

**Supervisé** : données $(x_i, y_i)$

- *Régression* : $y \in \mathbb{R}$
  prix d'un logement, température
- *Classification* : $y \in \{1, \dots, K\}$
  spam, diagnostic, reconnaissance de chiffres

**Non supervisé** : données $x_i$ seules

- *Clustering* : regrouper
- *Réduction de dimension* : résumer
- *Estimation de densité*

- Autres cadres (hors cours) : semi-supervisé, auto-supervisé, apprentissage par renforcement.
- Ce CM : le cadre **supervisé**, commun à tout le reste du cours.


licensed under CC BY 4.0

# 02

# Cadre formel

# Données et hypothèse fondamentale

- Espace d'entrée $\mathcal{X}$ (souvent $\mathbb{R}^d$), espace de sortie $\mathcal{Y}$.
- Échantillon d'apprentissage :

$$\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

- **Hypothèse i.i.d.** : les couples $(x_i, y_i)$ sont tirés indépendamment selon une même loi $P$ sur $\mathcal{X} \times \mathcal{Y}$, **inconnue**.

**L'hypothèse i.i.d. n'est pas là pour rien**

Les futures données doivent suivre la même loi $P$. Si ce n'est pas le cas (dérive temporelle, changement de population), les garanties de généralisation tombent.


licensed under CC BY 4.0

# Fonction de perte

Une perte $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ mesure le coût de prédire $\hat{y}$ quand la vérité est $y$.

| Tâche | Perte | Expression |
|---|---|---|
| Régression | quadratique | $(y - \hat{y})^2$ |
| Régression | absolue | $\lvert y - \hat{y} \rvert$ |
| Classification | 0-1 | $\mathbb{1}[y \neq \hat{y}]$ |
| Classification | logistique | $\log(1 + e^{-yf(x)}), y \in \{-1, 1\}$ |
| Classification | charnière | $\max(0, 1 - yf(x))$ |

- La perte 0-1 est celle qui nous intéresse (pour l'instant), mais elle n'est ni convexe ni dérivable.
- On optimise donc souvent une **perte de substitution** convexe.


licensed under CC BY 4.0

# Risque et prédicteur de Bayes

## Risque (erreur de généralisation)

$$R(f) = \mathbb{E}_{(X,Y)\sim P}\big[\ell(Y, f(X))\big]$$

- C'est la quantité qu'on veut minimiser, mais $P$ est inconnue.
- Le meilleur prédicteur possible est le **prédicteur de Bayes** $f^\star = \arg\min_f R(f)$.

**Perte quadratique**

$$f^\star(x) = \mathbb{E}[Y \mid X = x]$$

**Perte 0-1**

$$f^\star(x) = \arg\max_k P(Y = k \mid X = x)$$

- $R(f^\star)$ est en général $> 0$ : bruit irréductible.


licensed under CC BY 4.0

# Minimisation du risque empirique (ERM)

## Risque empirique

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, f(x_i)\big)$$

- Par la loi des grands nombres, $\hat{R}_n(f) \to R(f)$ pour $f$ **fixée**.
- On choisit un **espace d'hypothèses** $\mathcal{F}$ et on résout :

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f)$$

- **Piège** : $\hat{f}_n$ dépend des données, donc $\hat{R}_n(\hat{f}_n)$ est un estimateur **optimiste** de $R(\hat{f}_n)$.


licensed under CC BY 4.0

# Exemple fil rouge : la régression linéaire

- Espace d'hypothèses : $\mathcal{F} = \{x \mapsto w^\top x + b\}$, et on optimise la perte quadratique.
- ERM = moindres carrés :

$$\hat{w} = \arg\min_w \frac{1}{n}\|y - Xw\|_2^2 \quad \Longrightarrow \quad \hat{w} = (X^\top X)^{-1} X^\top y$$

- La RL est supposée connue ici : elle sert d'**exemple** pour illustrer
  - le compromis biais-variance (via des features polynomiales),
  - la régularisation (ridge, lasso),
  - le protocole d'évaluation.

**Enrichir $\mathcal{F}$ sans changer d'algorithme**

Features polynomiales : $x \mapsto (x, x^2, \dots, x^p)$, puis régression linéaire. Le degré $p$ contrôle la **complexité** du modèle.


licensed under CC BY 4.0

# Décomposition de l'erreur

$$R(\hat{f}_n) - R(f^\star) = \underbrace{R(\hat{f}_n) - \inf_{f \in \mathcal{F}} R(f)}_{\text{erreur d'estimation}} + \underbrace{\inf_{f \in \mathcal{F}} R(f) - R(f^\star)}_{\text{erreur d'approximation}}$$

- **Approximation** : $\mathcal{F}$ est-il assez riche pour contenir un bon prédicteur ? Diminue quand $\mathcal{F}$ grandit.
- **Estimation** : peut-on trouver le meilleur de $\mathcal{F}$ avec $n$ points ? Augmente quand $\mathcal{F}$ grandit, diminue quand $n$ grandit.

**Conséquence**

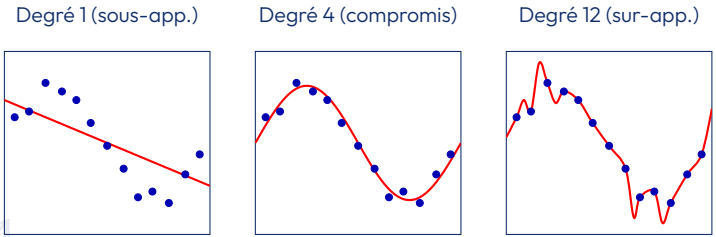
Choisir $\mathcal{F}$ est un **compromis** : ni trop pauvre, ni trop riche.


licensed under CC BY 4.0

# 03

# Généralisation

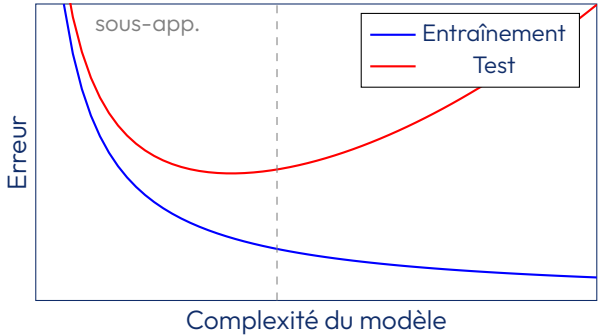
# Sous- et sur-apprentissage




- **Sous-apprentissage** : erreur élevée sur l'entraînement *et* le test.
- **Sur-apprentissage** : erreur quasi nulle sur l'entraînement, élevée sur le test.


licensed under CC BY 4.0

# Erreur d'entraînement vs erreur de test




- L'erreur d'entraînement décroît toujours avec la complexité.
- L'erreur de test est en U : c'est elle qui guide le choix du modèle.


licensed under CC BY 4.0

# Compromis biais-variance

**Retour à la régression :** $Y = f^\star(X) + \varepsilon$ avec $\mathbb{E}[\varepsilon] = 0$, $\mathrm{Var}(\varepsilon) = \sigma^2$. En un point $x$, en moyennant sur les échantillons $\mathcal{D}_n$ :

$$\mathbb{E}\big[(Y - \hat{f}_n(x))^2\big] = \underbrace{\big(f^\star(x) - \mathbb{E}[\hat{f}_n(x)]\big)^2}_{\text{biais}^2} + \underbrace{\mathrm{Var}\big(\hat{f}_n(x)\big)}_{\text{variance}} + \underbrace{\sigma^2}_{\text{bruit}}$$

**Biais élevé**

- modèle trop simple
- rate la structure
- stable d'un échantillon à l'autre

**Variance élevée**

- modèle trop flexible
- ajuste le bruit
- change beaucoup avec les données


licensed under CC BY 4.0

# Leviers pour agir sur le compromis

| Action | Biais | Variance |
|---|---|---|
| Augmenter la complexité de $\mathcal{F}$ | $\searrow$ | $\nearrow$ |
| Ajouter de la régularisation | $\nearrow$ | $\searrow$ |
| Augmenter $n$ | $=$ | $\searrow$ |
| Ajouter des features pertinentes | $\searrow$ | $\nearrow$ |
| Moyenner plusieurs modèles (bagging) | $=$ | $\searrow$ |

- *Diagnostic pratique* : **courbes d'apprentissage** (erreur en fonction de $n$).
- Écart train/validation large $\Rightarrow$ variance
- Les deux élevées $\Rightarrow$ biais.


licensed under CC BY 4.0

# Malédiction de la dimension

- En dimension $d$, pour couvrir $[0,1]^d$ avec une grille de pas $0.1$, il faut $10^d$ points.
- Pour capturer une fraction $r$ des données uniformes dans un hypercube local, son côté doit valoir $r^{1/d}$ :

$$r = 1\%,\ d = 10 \ \Rightarrow\ \text{côté} \approx 0{,}63$$

- Les points deviennent tous à peu près **équidistants** : la notion de voisinage se dégrade.

**À retenir**

Plus $d$ est grand, plus il faut de données ou d'hypothèses (régularité, parcimonie) pour bien généraliser.


licensed under CC BY 4.0

# 04

# Régularisation

# ERM régularisé

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, f(x_i)\big) + \lambda\, \Omega(f)$$

- $\Omega(f)$ pénalise la complexité tandis que $\lambda \geq 0$ règle le compromis.
- $\lambda = 0$ : ERM classique.
- $\lambda \to \infty$ : modèle trivial.
- $\lambda$ est un **hyperparamètre** : il ne s'apprend pas sur l'entraînement.

## Ridge (L2)

$$\Omega(w) = \|w\|_2^2$$

Coefficients rétrécis, solution explicite : $(X^\top X + n\lambda I)^{-1} X^\top y$

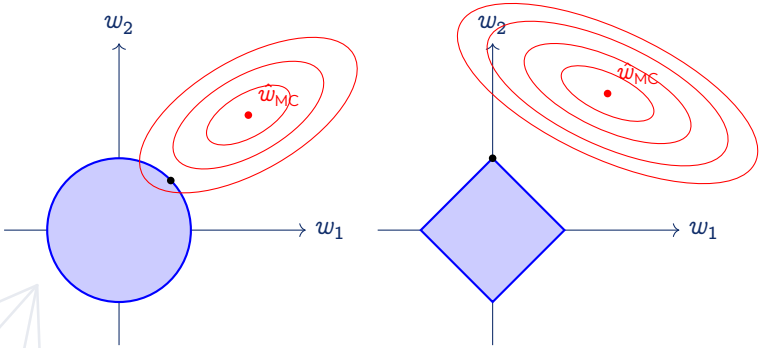
## Lasso (L1)

$$\Omega(w) = \|w\|_1$$

Coefficients exactement nuls : sélection de variables


licensed under CC BY 4.0

# Pourquoi le lasso produit des zéros




- Forme contrainte : minimiser la perte sous $\Omega(w) \leq t$.
- Les lignes de niveau de la perte touchent la boule L1 sur un **coin**, donc sur un axe.


licensed under CC BY 4.0

# Régularisation : points pratiques

- **Standardiser** les variables avant : la pénalité dépend de l'échelle des coefficients.
- Ne pas pénaliser l'intercept $b$.
- **Elastic net** : $\alpha\|w\|_1 + (1-\alpha)\|w\|_2^2$, utile avec des variables corrélées.
- Chemin de régularisation : tracer les coefficients en fonction de $\lambda$.
- Choix de $\lambda$ : par **validation croisée** (partie suivante).

**La régularisation est partout**

Paramètre $C$ des SVM, profondeur des arbres, dropout et weight decay des réseaux de neurones : même idée, contrôler la complexité.


licensed under CC BY 4.0

05

# Protocole d’évaluation

# Estimer le risque : séparer les données



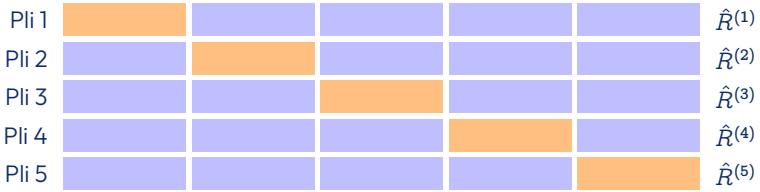

- **Entraînement** : ajuster les paramètres ($w$).
- **Validation** : choisir les hyperparamètres et le modèle ($\lambda$, degré, famille).
- **Test** : estimer la performance finale, **une seule fois**, à la fin.

**Règle d'or**

Toute décision prise en regardant le test le contamine : son score devient optimiste.


licensed under CC BY 4.0

# Validation croisée à $k$ plis




$$\hat{R}_{\text{CV}} = \frac{1}{k} \sum_{j=1}^{k} \hat{R}^{(j)}$$

- Chaque point sert une fois en validation ($k = 5$ ou 10 en pratique).
- On rapporte la **moyenne** et l'**écart-type** des $k$ scores.
- Plus coûteux ($k$ entraînements), mais bien moins dépendant du découpage.


licensed under CC BY 4.0

# Variantes de la validation croisée

- **Stratifiée** : conserve la proportion des classes dans chaque pli (classification).
- **Par groupes** : tous les échantillons d'un même groupe (patient, utilisateur, sujet) restent dans le même pli.
- **Temporelle** : on valide toujours sur le futur, jamais sur le passé.
- **Leave-one-out** : $k = n$, peu biaisée mais coûteuse et de forte variance.
- **Imbriquée** : une CV interne choisit les hyperparamètres, une CV externe estime la performance de toute la procédure.

**Principe**

Le découpage doit reproduire la situation de déploiement réelle.


licensed under CC BY 4.0

# Fuite de données

**Fuite** : une information indisponible au moment de la prédiction entre dans l'entraînement.

- **Prétraitement avant le découpage** : standardisation, imputation, sélection de variables calculées sur tout le jeu.
- **Doublons** ou mesures du même individu dans train et test.
- **Variable cible déguisée** : une feature calculée après coup (date de sortie d'hôpital pour prédire la durée de séjour).
- **Mélange temporel** : entraîner sur le futur pour prédire le passé.

**Symptôme typique**

Un score « trop beau pour être vrai ».
Remède : tout prétraitement dans un `Pipeline`, ajusté uniquement sur l'entraînement.


licensed under CC BY 4.0

06

# Métriques d’évaluation

# Métriques de régression

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad \text{RMSE} = \sqrt{\text{MSE}}$$

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

- RMSE et MAE sont dans l'unité de $y$ : plus faciles à interpréter.
- MSE pénalise fortement les grosses erreurs mais MAE est plus robuste aux valeurs aberrantes.
- $R^2$ : part de variance expliquée.


licensed under CC BY 4.0

# Matrice de confusion

| | | Prédit | |
|---|---|---|---|
| | | Positif | Négatif |
| Réel | Positif | VP | FN |
| | Négatif | FP | VN |

$$\text{Exactitude} = \frac{VP + VN}{n}$$

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

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

$$F_1 = \frac{2 \cdot \text{Précision} \cdot \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

- **Précision** : parmi les alertes, combien sont vraies ?
- **Rappel** : parmi les vrais positifs, combien sont détectés ?


licensed under CC BY 4.0

# Le piège des classes déséquilibrées

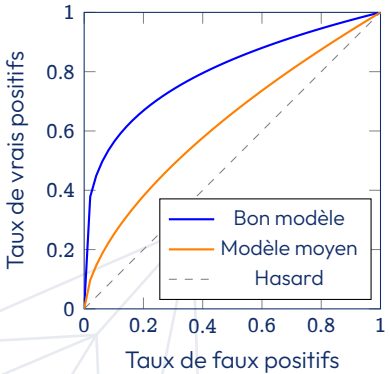
**Exemple**

Détection de fraude : 1 % de fraudes. Le classifieur « jamais de fraude » atteint **99 % d'exactitude**… et un rappel de 0.

- L'exactitude est trompeuse dès que les classes sont déséquilibrées.
- Préférer précision, rappel, $F_1$, exactitude équilibrée, PR-AUC.
- Le bon choix dépend du **coût des erreurs** : rater un cancer (FN) ou déclencher une fausse alerte (FP) ?
- Leviers : pondération des classes, choix du seuil, rééchantillonnage (uniquement sur l'entraînement).


licensed under CC BY 4.0

# Seuil de décision et courbe ROC




- Le modèle produit un score $s(x)$ : on prédit positif si $s(x) \geq \tau$.
- Faire varier $\tau$ trace la courbe ROC.
- **AUC** : probabilité qu'un positif ait un score plus élevé qu'un négatif.
- AUC $= 0.5$ : hasard / $1$ : parfait.
- Classes rares : préférer la courbe **précision-rappel**.


licensed under CC BY 4.0