

# Cours 1 - Les fondements de l'apprentissage machine

Neila Mezghani

15 décembre 2021 & 3 janvier 2022

# Plan du cours

## 1 Introduction

- Paramètres ou Hyperparamètres ?
- Modèles d'AM
  - Machine à support vectoriel
  - Arbres de décision
- Entraînement, test et validation du modèle

## 2 Évaluation d'un système de classification

## 3 Principales difficultés de l'AM

- Sous et sur-ajustement aux données d'entraînement
- Données d'apprentissage en nombre insuffisant
- Données d'entraînement non représentatives
- Variables non pertinentes

## 4 Vers... Apprentissage profond

# Introduction

## Qu'est ce que l'apprentissage machine ? (1/2)

Par définition, l'apprentissage automatique ou l'apprentissage machine (AM) (Machine Learning) est la science de programmer les ordinateurs de sorte qu'ils puissent apprendre à partir de données. Voici une définition un peu plus générale :

*« ...est la discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés. »*

*Arthur Samuel, 1959*

## Qu'est ce que l'apprentissage machine ? (2/2)

Une autre définition plus technique :

*« 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  $E$  . »*

*Tom Mitchell, 1997*

## Phases d'un système d'apprentissage machine (1/2)

Un système basé sur l'AM passe en général par deux phases :

- La phase d'entraînement ou d'apprentissage pendant laquelle l'algorithme détermine les paramètres du modèle.
- La phase de prédiction ou d'inférence pendant laquelle le système entraîné applique ce qu'il a appris sur de nouvelles données.

## Phases d'un système d'apprentissage machine (2/2)

Phase  
d'entraînement

Données  
d'entraînement



Algorithme  
d'entraînement



Modèle  
(d'IA)

Phase de  
prédiction

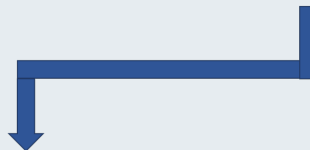
Nouvelle donnée



Algorithme de  
prédiction



Variable cible



# Plan du cours

## 1 Introduction

- Paramètres ou Hyperparamètres ?
- Modèles d'AM
  - Machine à support vectoriel
  - Arbres de décision
- Entraînement, test et validation du modèle

## 2 Évaluation d'un système de classification

## 3 Principales difficultés de l'AM

- Sous et sur-ajustement aux données d'entraînement
- Données d'apprentissage en nombre insuffisant
- Données d'entraînement non représentatives
- Variables non pertinentes

## 4 Vers... Apprentissage profond



## Paramètres ou Hyperparamètres ?



Les termes **Hyperparamètres** et **Paramètres** sont souvent utilisés de manière interchangeable bien qu'il existe une différence entre eux.

- Un hyperparamètre est un paramètre de l'algorithme d'apprentissage  $\implies$  un hyperparamètre ne fait pas partie des variables que l'on cherche à optimiser pendant l'apprentissage.
- Les paramètres sont les variables du modèle apprises pendant l'apprentissage

## Paramètres ou Hyperparamètres ?

- Dans la pratique, on lance l'algorithme d'apprentissage plusieurs fois, en essayant différentes valeurs des hyperparamètres, jusqu'à déterminer la combinaison de valeurs qui permet à l'algorithme d'apprentissage de produire un modèle satisfaisant.
- Pour évaluer chaque modèle, on utilise l'ensemble des données de validation  $\implies$  Ce réglage fin des hyperparamètres s'appelle le *hyperparameter tuning*.

## Paramètres ou Hyperparamètres ?

- Dans un modèle de régression : les hyperparamètres sont le nombre d'itération et la valeur de  $\eta$ . Les paramètres sont les coefficients qui seraient appris par la machine ( $\theta_0, \theta_1, \dots$ )
- Dans un réseau de neurone : les hyperparamètres sont le nombre de nœuds, le nombre de couches cachées, la fonction d'activation, etc. Les paramètres sont ceux qui seraient appris par la machine comme les poids synaptiques et les biais.

## Comment choisir les hyperparamètres ?

- Exemple : Comment choisir le nombre d'itérations  $N$  dans un réseau de neurone ?
  - Si  $N$  est trop faible  $\implies$  on sera très loin de la solution optimale lorsque l'algorithme s'arrêtera.
  - Si  $N$  est trop élevé  $\implies$  On perd du temps alors que les paramètres du modèle n'évolueront plus.
- Une solution simple consiste à choisir un très grand nombre d'itérations, mais à interrompre l'algorithme sous certaines conditions comme par exemple l'atteinte d'un certain taux de classification.

# Plan du cours

1

## Introduction

- Paramètres ou Hyperparamètres ?

- Modèles d'AM

  - Machine à support vectoriel

  - Arbres de décision

- Entraînement, test et validation du modèle

2

## Évaluation d'un système de classification

3

## Principales difficultés de l'AM

- Sous et sur-ajustement aux données d'entraînement

- Données d'apprentissage en nombre insuffisant

- Données d'entraînement non représentatives

- Variables non pertinentes

4

## Vers... Apprentissage profond

## Machine à support vectoriel

- Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais support-vector machine, SVM) sont un ensemble de techniques d'apprentissage supervisé
- Les SVM peuvent être utilisés pour résoudre des problèmes de classification (décider à quelle classe appartient une observation) ou de régression (prédire la valeur numérique d'une variable).

## Machine à support vectoriel linéaire (1/2)

- La conception d'un SVM requiert une fonction  $h$  qui fait correspondre à un vecteur d'entrée  $\mathbf{x}$  une sortie  $y$  telle que

$$y = h(\mathbf{x})$$

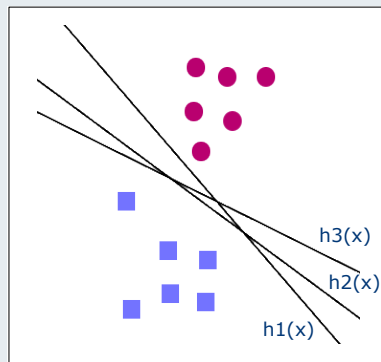
- Dans le cas simple, on a une fonction discriminante linéaire, obtenue par combinaison linéaire du vecteur d'entrée

$\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T$  avec un vecteur poids  $\mathbf{w} = (w_1, w_2, \dots, w_N)^T$  :

$$h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

## Machine à support vectoriel linéaire (2/2)

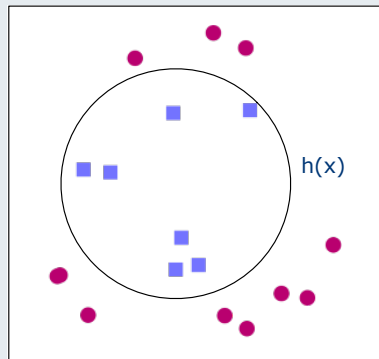
- La frontière de décision  $h(x) = 0$  est un hyperplan, appelé hyperplan séparateur, ou séparatrice.
- Le but est d'apprendre la fonction  $h(x)$  en se basant sur l'ensemble d'apprentissage





## Machine à support vectoriel non-linéaire

- lorsque les données sont non linéairement séparables (il n'existe pas d'hyperplan capable de séparer correctement les deux )  
     $\Rightarrow$  On utilise l'astuce du noyau (en anglais kernel trick)
- Consiste à reconsidérer le problème dans un espace de dimension supérieure.
- Dans ce nouvel espace, il est alors probable qu'il existe une séparation linéaire.



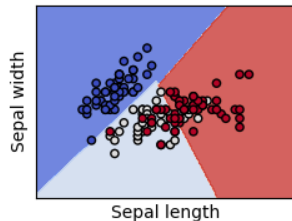
## Exemple de SVM (1/2)

Pour l'entraînement d'un SVM on peut utiliser la librairie : **Support Vector Machines**

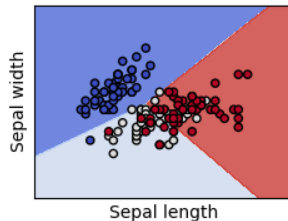
```
from sklearn.svm import SVC
# SVM Classifier model
svm_clf = SVC(kernel="linear", C = 1)
svm_clf.fit(X, y)
```

## Exemples de SVM (2/2)

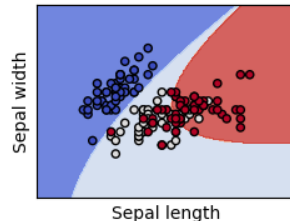
SVC with linear kernel



LinearSVC (linear kernel)



SVC with polynomial (degree 3) kernel



## Arbres de décision

- Un **arbre de décisions** (*decision tree*) est une structure très utilisée en forage de données. Son fonctionnement repose sur des heuristiques construites selon des techniques techniques d'apprentissage supervisées.
- Les arbres de décisions ont une structure hiérarchique et sont composés de **nœuds** et de **feuilles** (aussi appelées nœuds terminaux) reliés par des branches.

## Construction d'un arbre de décision (1/2)

- Les nœuds internes sont appelés des **nœuds de décision**. Ils peuvent contenir une ou plusieurs règles (aussi appelées tests, ou conditions).
- Plusieurs méthodes de construction d'arbres de décisions permettent de choisir entre les différentes variables.
- Exemples :
  - L'algorithme ID3
  - L'algorithme C4.5
  - L'algorithme CART

## Construction d'un arbre de décision (2/2)

- L'algorithme récursif CART (*Classification And Regression Trees*) permet la construction d'un arbre de décisions par la maximisation de l'indice de Gini.
- L'indice de Gini mesure l'impureté, qui est un concept très utile dans la construction des arbres de décisions.
- La qualité d'un nœuds et son pouvoir discriminant peuvent être évalués par son impureté.

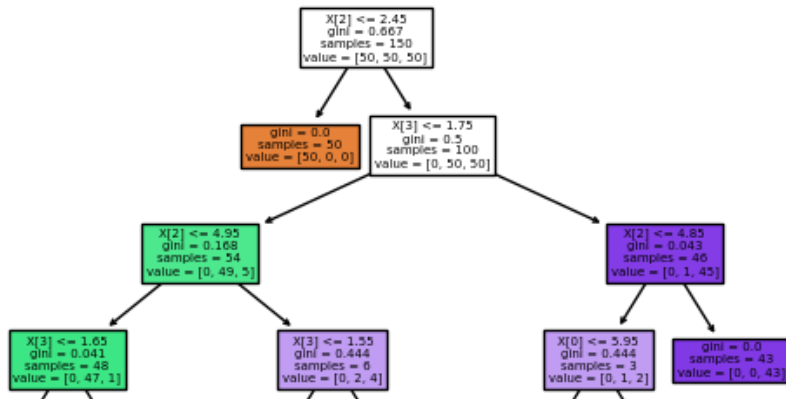
## Exemple d'un arbre de décision (1/2)

Pour l'entraînement d'un SVM on peut utiliser la librairie : **Decision Tree**

```
from sklearn import tree
# Decision tree Classifier model
tree_clf = tree.DecisionTreeClassifier()
tree_clf = clf.fit(X, y)
```

## Exemple d'un arbre de décision (2/2)

Decision tree trained on all the iris features





# Plan du cours

1

## Introduction

- Paramètres ou Hyperparamètres ?
- Modèles d'AM
  - Machine à support vectoriel
  - Arbres de décision
- Entraînement, test et validation du modèle

2

## Évaluation d'un système de classification

3

## Principales difficultés de l'AM

- Sous et sur-ajustement aux données d'entraînement
- Données d'apprentissage en nombre insuffisant
- Données d'entraînement non représentatives
- Variables non pertinentes

4

## Vers... Apprentissage profond

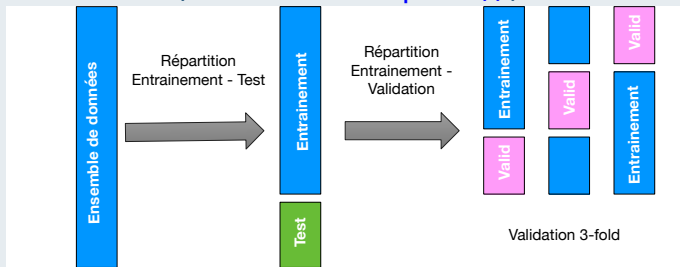
## Entraînement, test et validation du modèle (1)

L'ensemble des données considérées pour une analyse par apprentissage machine peut être réparti (échantillonné) en trois sous-ensembles :

- Un ensemble d'entraînement
- Un ensemble de validation
- Un ensemble de test

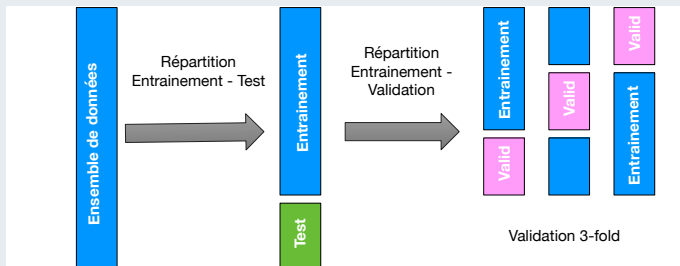
## Entraînement, test et validation du modèle (2)

**L'ensemble d'entraînement** qui est utilisé pour l'apprentissage des paramètres du modèle. (`train_test_split()`)



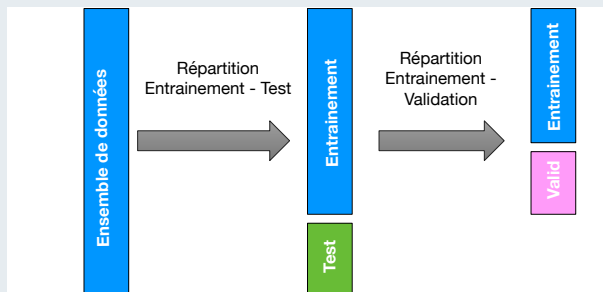
## Entraînement, test et validation du modèle (3)

**L'ensemble de validation** qui permet d'évaluer le modèle pendant la phase d'entraînement. Cette étape peut être omise en passant à la phase de test directement.



## Entraînement, test et validation du modèle (4)

**L'ensemble de test.** Une fois le modèle construit à partir de l'ensemble d'entraînement, le modèle est évalué en utilisant un ensemble de test : un ensemble d'échantillons n'ayant pas servi pour l'apprentissage.

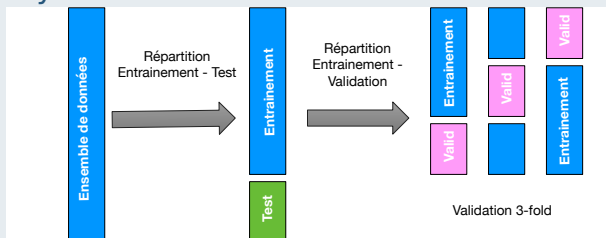


## Validation croisée

- Permet d'utiliser l'intégralité de l'ensemble de données pour l'entraînement et pour la validation.
- Les stratégies (les plus connues)
  - Validation  $k$ -fold ( $k$ -fold cross validation)
  - Leave-one out cross validation
  - Bootstrap

## Validation croisée : Validation $k$ -fold (1/3)

- Consiste à diviser l'ensemble original des  $n$  individus en  $k$  sous-ensemble de même taille et à prendre un sous-ensemble pour procéder à la validation.
- Le processus est répété  $k$  fois jusqu'au parcours des  $k$  sous-ensemble et le critère d'évaluation est calculé  $k$  fois pour en obtenir une moyenne.



## Validation croisée : Validation $k$ -fold (2/3)

- La stratification est le processus qui consiste à diviser la population générale en sous-groupes homogènes avant l'échantillonnage.
- L'objectif de la stratification est de créer des folds de sorte à ce qu'elles contiennent à peu près les mêmes proportions d'individus de chaque classe.
  - ⇒ d'éviter que l'ensemble d'entraînement contienne que des individus d'une classe particulière et par conséquent diminuer le risque d'affecter négativement la performance du modèle.



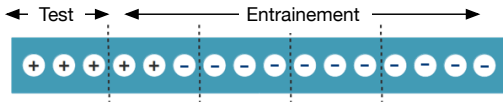
## Validation croisée : Validation $k$ -fold (3/3)

Ensemble de données binaires



Répartition des données en des données de Test et d'Entraînement

Mauvaise répartition

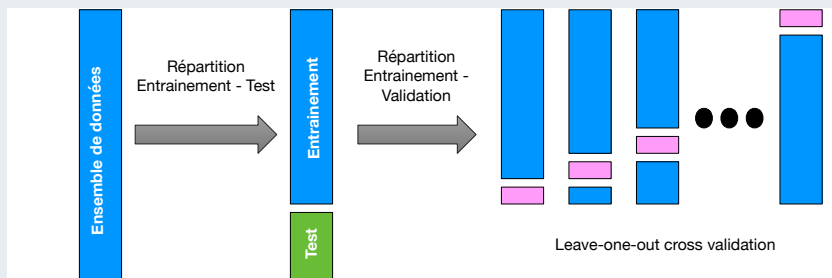


Bonne répartition



## Validation croisée : Leave-one out cross validation

- Consiste à diviser l'ensemble original des  $n$  individus en  $k$  échantillons (partitions) avec  $k = n \implies$  un seul individu est gardé pour le test et les  $k - 1$  restant pour l'entraînement



# Évaluation d'un système de classification

## Taux de classification (1/6)

- Soit  $S$  un ensemble d'échantillons d'apprentissage, et  $T$  un ensemble d'échantillons de test.
- L'estimation du taux de bonne classification est mesurée sur l'ensemble de test selon :

$$\tau = \frac{\text{nbr bien classifié}(T)}{\text{nbr}(T)}$$

## Taux de classification (2/6)

- Le taux de bonne classification est généralement donné en pourcentage. Il lui correspond une valeur complémentaire à 100 correspondant au taux d'erreur

$$\epsilon(\%) = 100 - \tau(\%)$$

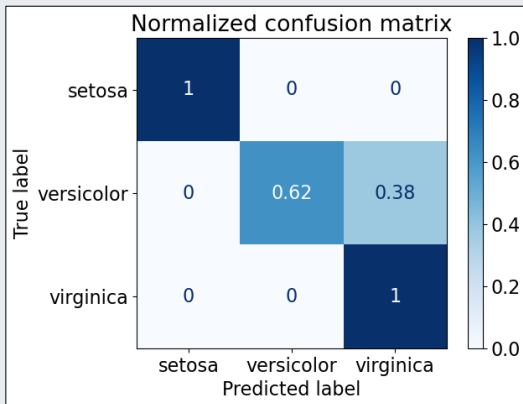
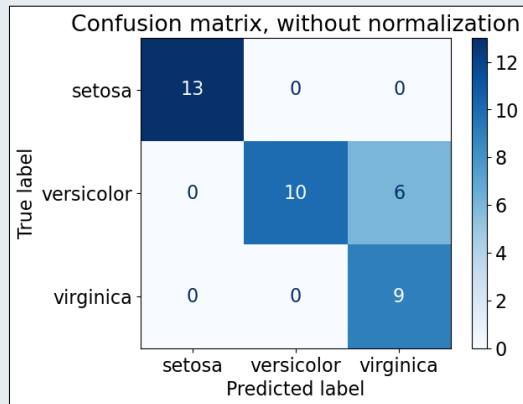
## Matrice de confusion (3/6)

- La matrice de confusion est aussi connue sous les termes matrice d'erreur, tableau de contingence ou matrice d'erreur de classification.
- C'est une matrice affichant les statistiques de la précision de classification et plus particulièrement les taux de classification par classes.
- Généralement, l'information des lignes (données horizontales) correspond aux classes réelles des formes et colonnes (données verticales) contiennent l'information prédite résultant de la classification.

## Matrice de confusion (4/6)

- Les valeurs de la diagonale de la matrice représentent le nombre de formes correctement classifiées.
- La somme des valeurs par colonne correspond au nombre d'échantillons de test par classe.
- Le taux de classification par classes est donné par la valeur à la diagonale divisée par la somme des valeurs par colonne  $\tau_i$ ,  
 $i = 1, 2, 3, \dots, c$ .  
 $c$  étant le nombre de classes.

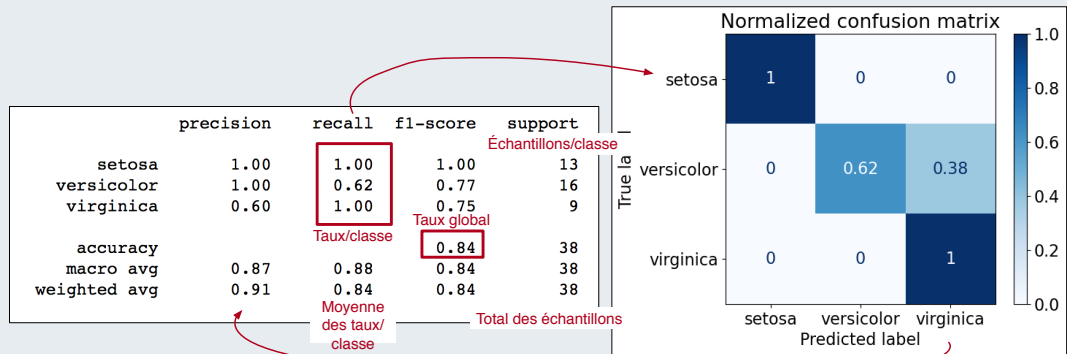
## Matrice de confusion (5/6)



Se référer à : **Confusion matrix**



## Matrice de confusion (6/6)



## Matrice de confusion : cas binaire (1/3)

- Dans le cas binaire, nous avons deux classes (+ et - ou bien vrai et faux)
- Le choix du + et de - a un sens dans le cas de variables non symétriques
- Exemple : Prenons un algorithme qui prédit s'il y a un incendie à un endroit donné. Déclencher une alerte incendie quand il n'y a pas le feu est moins grave que de ne pas déclencher d'alerte quand l'appartement est en flamme.

## Matrice de confusion : cas binaire (2/3)

| Confusion Matrix |       | Predicted           |                     |
|------------------|-------|---------------------|---------------------|
|                  |       | FALSE               | TRUE                |
| Actual           | FALSE | True Negative (TN)  | False Positive (FP) |
|                  | TRUE  | False Negative (FN) | True Positive (TP)  |

Précision

Recall = Rappel = Sensibilité

## Matrice de confusion : cas binaire (3/3)

- À partir de la matrice de confusion, on peut dériver plusieurs critères de performance  $\implies$  fonction coût
  - La sensibilité (sensitivity)
  - La spécificité (specificity)
  - La précision
  - Mesure F1
- De manière générale, on préfère les donner sous la forme d'une fraction d'erreurs à un nombre total d'erreurs

## Matrice de confusion : Sensibilité

- Le rappel (recall) ou sensibilité (sensitivity), est le taux de vrais positifs, c'est à dire la proportion de positifs que l'on a correctement identifiés.

| Confusion Matrix |       | Predicted           |                     |
|------------------|-------|---------------------|---------------------|
|                  |       | FALSE               | TRUE                |
| Actual           | FALSE | True Negative (TN)  | False Positive (FP) |
|                  | TRUE  | False Negative (FN) | True Positive (TP)  |

Précision

↓

Recall = Rappel = Sensibilité

→

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

## Matrice de confusion : Spécificité

- La spécificité (specificity) est le taux de vrais négatifs
- C'est une mesure complémentaire de la sensibilité.

| Confusion Matrix |       | Predicted           |                     |
|------------------|-------|---------------------|---------------------|
|                  |       | FALSE               | TRUE                |
| Actual           | FALSE | True Negative (TN)  | False Positive (FP) |
|                  | TRUE  | False Negative (FN) | True Positive (TP)  |

Précision

Recall = Rappel = Sensibilité

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

## Matrice de confusion : Précision

- La précision est la proportion de prédictions correctes parmi les points que l'on a prédits positifs.

| Confusion Matrix |       | Predicted           |                     |
|------------------|-------|---------------------|---------------------|
|                  |       | FALSE               | TRUE                |
| Actual           | FALSE | True Negative (TN)  | False Positive (FP) |
|                  | TRUE  | False Negative (FN) | True Positive (TP)  |

Diagram illustrating the Confusion Matrix and the calculation of Precision and Recall:

- Precision** is indicated by a red arrow pointing to the **TRUE** column (Predicted).
- Recall = Rappel = Sensibilité** is indicated by a red arrow pointing to the **TRUE** row (Actual).

$$Precision = \frac{TP}{TP + FP}$$

## Matrice de confusion : *Mesure – F*

- Pour évaluer un compromis entre rappel et précision, on peut calculer la "F-mesure", qui est leur moyenne harmonique.

| Confusion Matrix |       | Predicted           |                     |
|------------------|-------|---------------------|---------------------|
|                  |       | FALSE               | TRUE                |
| Actual           | FALSE | True Negative (TN)  | False Positive (FP) |
|                  | TRUE  | False Negative (FN) | True Positive (TP)  |

Précision

Recall = Rappel = Sensibilité

$$Mesure - F = \frac{2 \times TP}{2 \times TP + FN + FP}$$



## Principales difficultés de l'AM

## Principales difficultés de l'AM

- Les principales étapes d'un système d'AM consistent à sélectionner un algorithme d'apprentissage et à l'entraîner sur un ensemble de données.
- Les systèmes d'AM peuvent rencontrer de nombreuses difficultés :
  - 1 Sous et sur-ajustement aux données d'entraînement
  - 2 Données d'apprentissage en nombre insuffisant
  - 3 Données d'entraînement non représentatives
  - 4 Variables non pertinentes

# Plan du cours

## 1 Introduction

- Paramètres ou Hyperparamètres ?
- Modèles d'AM
  - Machine à support vectoriel
  - Arbres de décision
- Entraînement, test et validation du modèle

## 2 Évaluation d'un système de classification

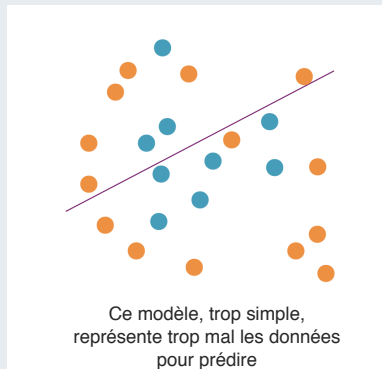
## 3 Principales difficultés de l'AM

- **Sous et sur-ajustement aux données d'entraînement**
- Données d'apprentissage en nombre insuffisant
- Données d'entraînement non représentatives
- Variables non pertinentes

## 4 Vers... Apprentissage profond

## Sous-apprentissage (1/2)

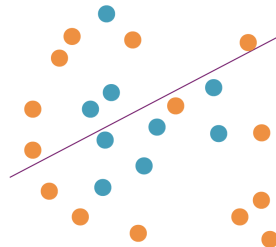
- Le sous-apprentissage (*underfitting*) sous entend que le modèle prédictif généré lors de la phase d'apprentissage, s'adapte mal aux données d'entraînement.
- Bien évidemment, le modèle prédictif ne pourra pas généraliser sur des données qu'il n'a pas encore vu.



## Sous-apprentissage (2/2)

Le sous-apprentissage survient :

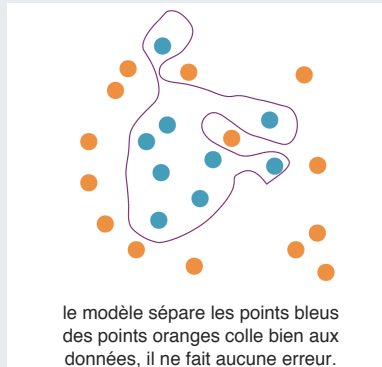
- Lorsqu'une classe considérée n'est pas suffisamment représentée pour pouvoir décrire la diversité présente dans les données
- Lorsque le modèle n'est pas suffisamment entraîné



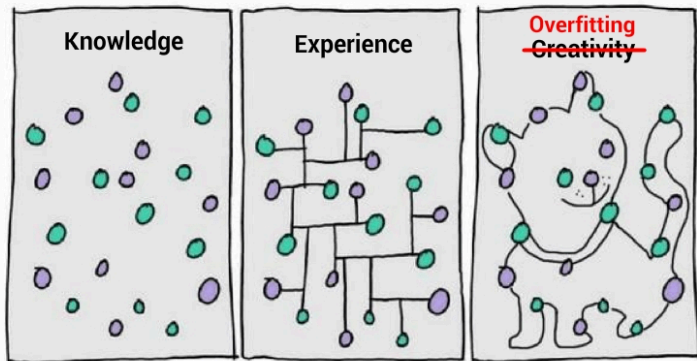
Ce modèle, trop simple,  
représente trop mal les données  
pour prédire

## Sur-apprentissage (1/2)

- Le sur-apprentissage (*overfitting*) survient lorsque le modèle apprend « par cœur » des données = une sorte de mémorisation.
- Le modèle colle trop aux données sur lesquelles il a appris
- Un tel modèle aura de très bonnes performances sur les données d'entraînement mais sera mauvais sur de nouvelles données.

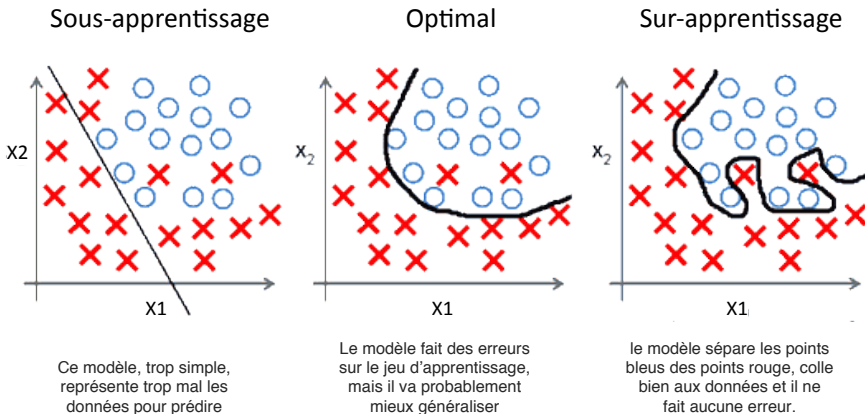


## Sur-apprentissage : Illustration (2/2)



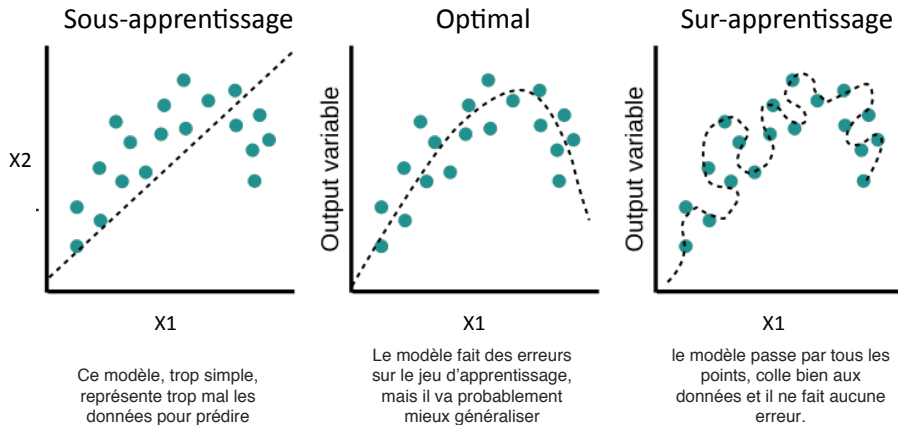
<https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d>

## Sous-apprentissage et sur-apprentissage : cas de la classification

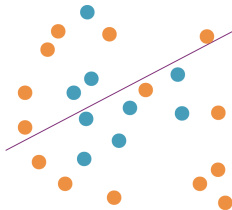




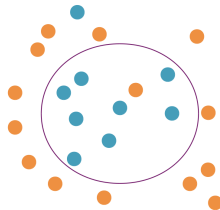
## Sous-apprentissage et sur-apprentissage : cas de la régression



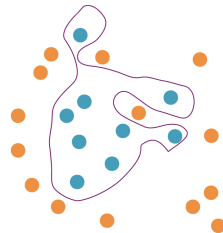
## Sous-apprentissage et sur-apprentissage : Résumé



Ce modèle, trop simple, représente trop mal les données pour prédire

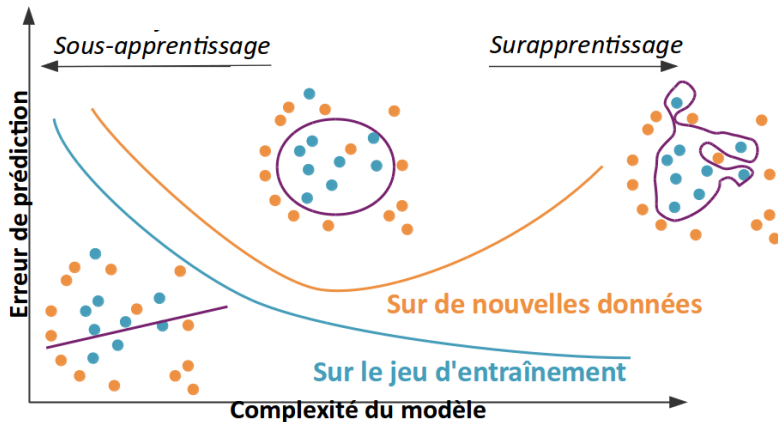


Le modèle fait des erreurs sur le jeu d'apprentissage, mais il va probablement mieux généraliser



le modèle sépare les points bleus des points oranges colle bien aux données, il ne fait aucune erreur.

## Sous-apprentissage et sur-apprentissage : Résumé



# Plan du cours

## 1 Introduction

- Paramètres ou Hyperparamètres ?
- Modèles d'AM
  - Machine à support vectoriel
  - Arbres de décision
- Entraînement, test et validation du modèle

## 2 Évaluation d'un système de classification

## 3 Principales difficultés de l'AM

- Sous et sur-ajustement aux données d'entraînement
- **Données d'apprentissage en nombre insuffisant**
- Données d'entraînement non représentatives
- Variables non pertinentes

## 4 Vers... Apprentissage profond

## Données d'apprentissage en nombre insuffisant

- L'apprentissage d'un système d'IA requiert un nombre suffisant de données pour être performant.
- Ce nombre dépend de la complexité du problème.
- Par exemple, la reconnaissance vocale et l'analyse d'image sont des problèmes qui nécessitent plusieurs milliers (voir millions) d'exemples.

# Plan du cours

## 1 Introduction

- Paramètres ou Hyperparamètres ?
- Modèles d'AM
  - Machine à support vectoriel
  - Arbres de décision
- Entraînement, test et validation du modèle

## 2 Évaluation d'un système de classification

## 3 Principales difficultés de l'AM

- Sous et sur-ajustement aux données d'entraînement
- Données d'apprentissage en nombre insuffisant
- **Données d'entraînement non représentatives**
- Variables non pertinentes

## 4 Vers... Apprentissage profond

## Données d'entraînement non représentatives

- Pour qu'un système d'IA basé sur l'AM soit utile, il doit avoir un bon pouvoir de généralisation  $\implies$  prendre la bonne décision sur de nouvelles données
- Il est donc essentiel qu'il soit entraîné sur des données d'entraînement représentatives.

# Plan du cours

## 1 Introduction

- Paramètres ou Hyperparamètres ?
- Modèles d'AM
  - Machine à support vectoriel
  - Arbres de décision
- Entraînement, test et validation du modèle

## 2 Évaluation d'un système de classification

## 3 Principales difficultés de l'AM

- Sous et sur-ajustement aux données d'entraînement
- Données d'apprentissage en nombre insuffisant
- Données d'entraînement non représentatives
- **Variables non pertinentes**

## 4 Vers... Apprentissage profond



## Variables non pertinentes (1/2)

- Un système d'IA ne sera performant que s'il est entraîné sur suffisamment de variables pertinentes et pas trop de variables non pertinentes.
- La réussite d'un système d'IA repose en partie sur le choix d'un bon ensemble de variables  $\implies$  c'est ce qu'on appelle ingénierie des variables (feature engineering) qui regroupe :
  - ◇ L'extraction de variables
  - ◇ La sélection de variables

## Variables non pertinentes (2/2)

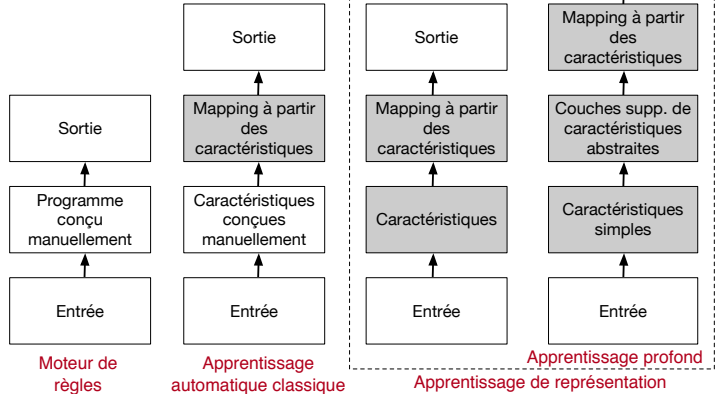
- Extraction des variables : consiste à convertir les données brutes en caractéristiques adaptées à la modélisation
- La sélection de variables : consiste à sélectionner les plus utiles à entraîner le modèles (parmi l'ensemble de toute les variables)

## Vers... Apprentissage profond

## Apprentissage profond : Définition

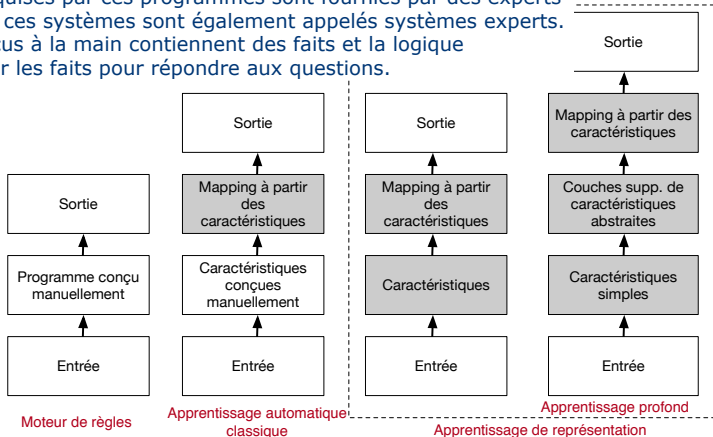
"L'apprentissage profond ou apprentissage en profondeur (deep learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires." (dictionnaire terminologique de la langue)

Organigramme descriptif des  
différentes parties d'un système d'IA.  
Les cases ombrées indiquent les  
éléments qui se basent sur l'AM



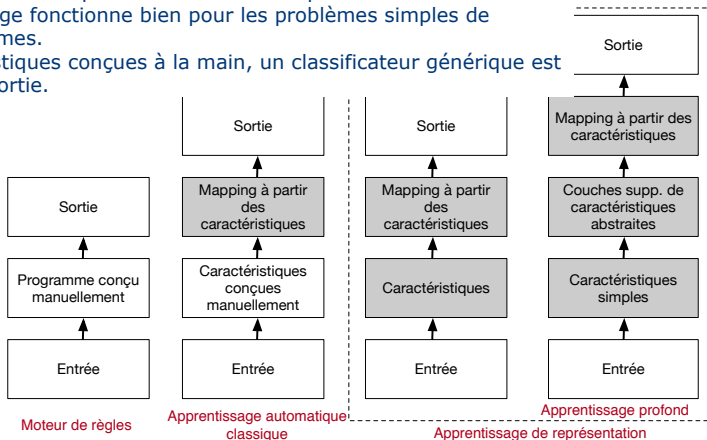
- **Les systèmes à base de règles** sont des programmes d'IA sont conçus à la main.
- Les connaissances requises par ces programmes sont fournies par des experts du domaine concerné : ces systèmes sont également appelés systèmes experts.
- Les programmes conçus à la main contiennent des faits et la logique permettant de combiner les faits pour répondre aux questions.

(2/6)



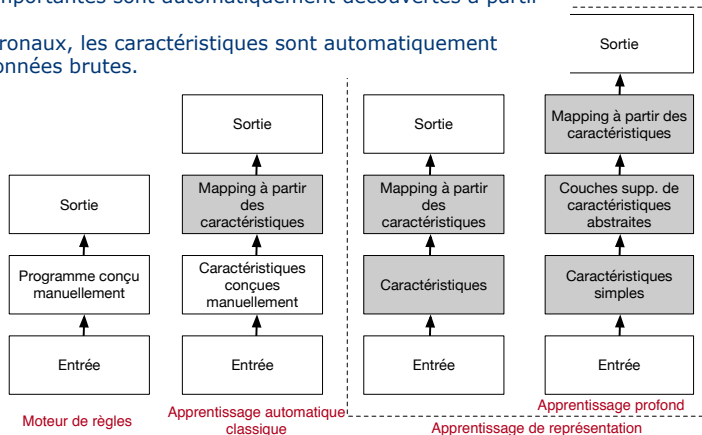
- Dans l'**apprentissage automatique classique**, les caractéristiques importantes de l'entrée sont conçues manuellement et le système apprend automatiquement à faire correspondre les caractéristiques aux sorties.
- Ce type d'apprentissage fonctionne bien pour les problèmes simples de reconnaissance des formes.
- Une fois les caractéristiques conçues à la main, un classificateur générique est utilisé pour obtenir la sortie.

(3/6)



- **L'apprentissage par représentation** va plus loin et élimine la nécessité de concevoir manuellement les caractéristiques.
- Les caractéristiques importantes sont automatiquement découvertes à partir des données.
- Dans les réseaux neuronaux, les caractéristiques sont automatiquement apprises à partir des données brutes.

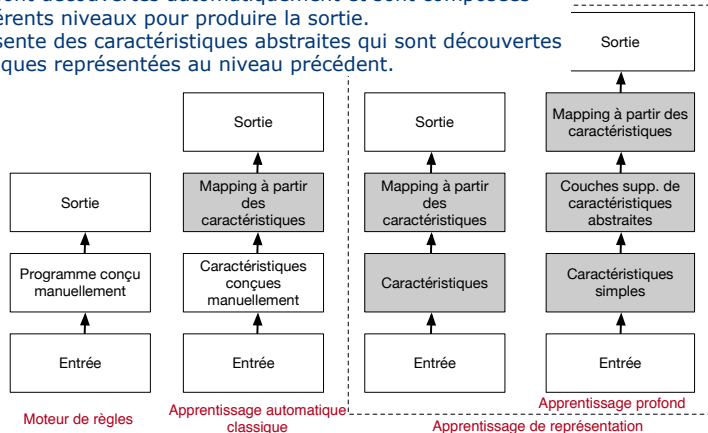
(4/6)





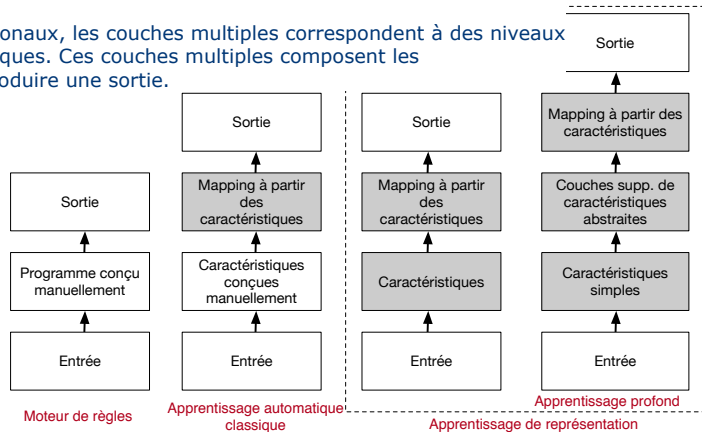
- **L'apprentissage profond** est un type d'apprentissage par représentation dans lequel il existe plusieurs niveaux de caractéristiques.
- Ces caractéristiques sont découvertes automatiquement et sont composées ensemble dans les différents niveaux pour produire la sortie.
- Chaque niveau représente des caractéristiques abstraites qui sont découvertes à partir des caractéristiques représentées au niveau précédent.

(5/6)



- Dans **l'apprentissage profond** Le niveau d'abstraction augmente avec chaque niveau.
- Ce type d'apprentissage permet de découvrir et de représenter des abstractions de plus haut niveau.
- Dans les réseaux neuronaux, les couches multiples correspondent à des niveaux multiples de caractéristiques. Ces couches multiples composent les caractéristiques pour produire une sortie.

(6/6)



## Comparaison entre apprentissage machine et profond

