Chapitre 6 - Apprentissage supervisé : la classification

Neila Mezghani

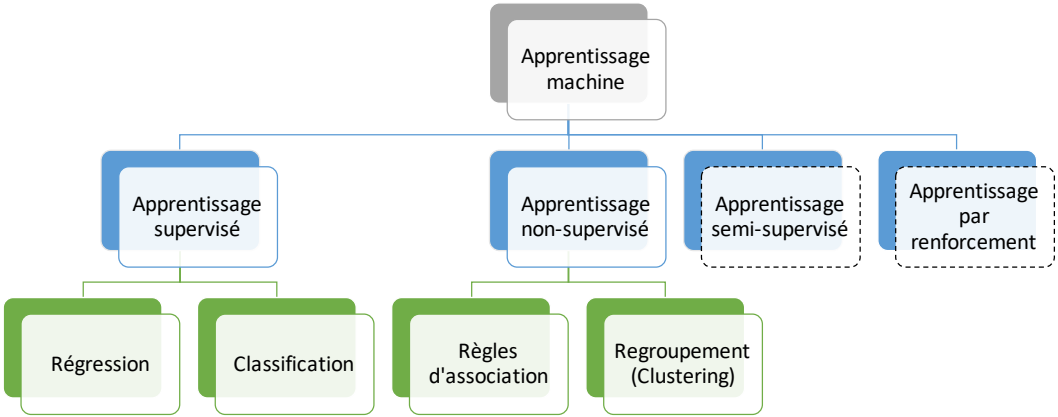
Hiver 2025

COLLÈGE DE
BOIS-DE-BOULOGNE

Plan du cours

Plan du cours

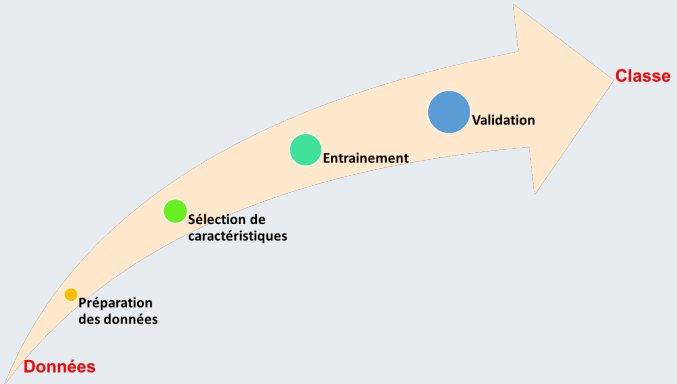
Techniques d'apprentissage machine



Classification de formes : Définition

- La classification de formes (en anglais, *pattern classification*) est « l'ensemble des techniques permettant à l'ordinateur de détecter la présence de formes, en comparant leurs caractéristiques avec celles de motifs de référence »(Office québécois de la langue française, 2010).
- La classification de formes est l'une des branches de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage machine ; plus particulièrement aux réseaux de neurones.

Cassification de formes : Étapes du processus



Plan du cours

Les données

- Un ensemble de données d'apprentissage comprend n individus décrits $D_n = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$ avec :
  - $\diamond$ $\mathbf{x}_i \in \mathbb{R}^n$ est le vecteur de caractéristiques du $i$ème individu.
  - $\diamond$ y_i est la classe associée à cet individu parmi les k classes.
- L'apprentissage supervisé consiste à déterminer une fonction $f : \mathbb{R}^n \rightarrow \{1, ..., k\}$ qui permet d'associer y à $f(x)$

Plan du cours

Entraînement, test et validation du modèle

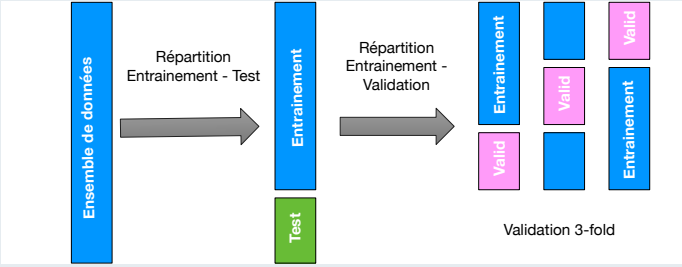
Entraînement, test et validation du modèle (1/4)

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

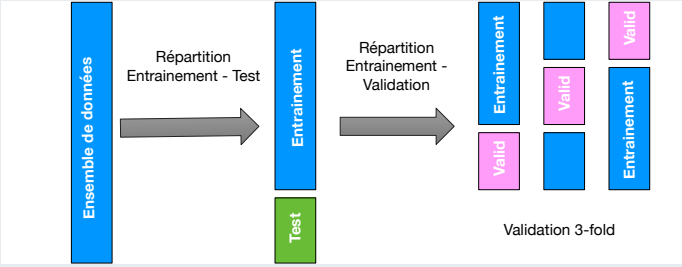
Entraînement, test et validation du modèle (2/4)

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



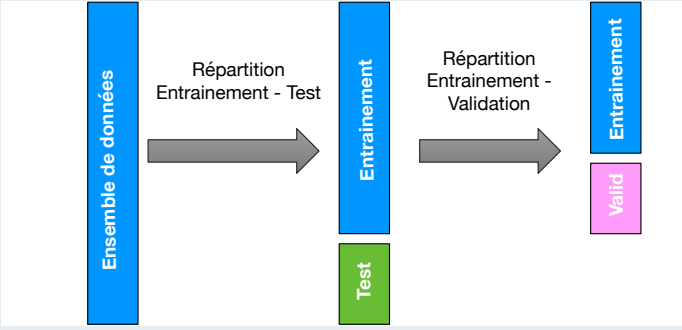
Entraînement, test et validation du modèle (3/4)

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 (4/4)

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.



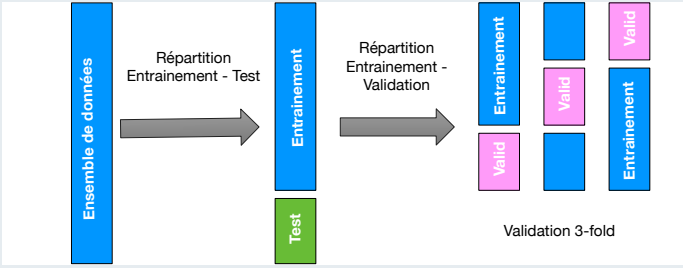
Validation croisée

Validation croisée

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

Validation croisée : Validation k-fold (1/2)

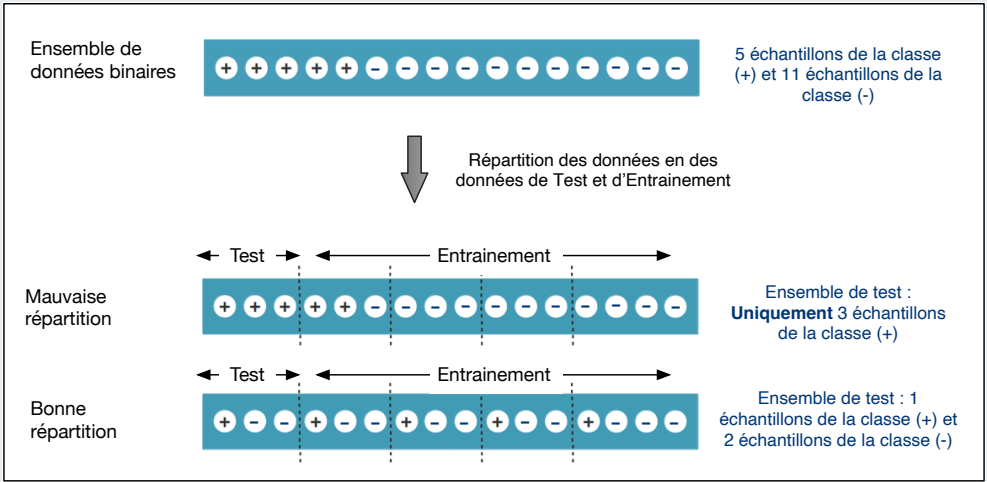
- Consiste à diviser l'ensemble original des n individus en k échantillons de même taille et à prendre un échantillon pour procéder à la validation.
- Le processus est répété k fois jusqu'au parcours des k échantillons.
- La fonction coût est calculé k fois $\Longrightarrow$ La moyenne est calculée.



Validation croisée : Validation k-fold (2/2)

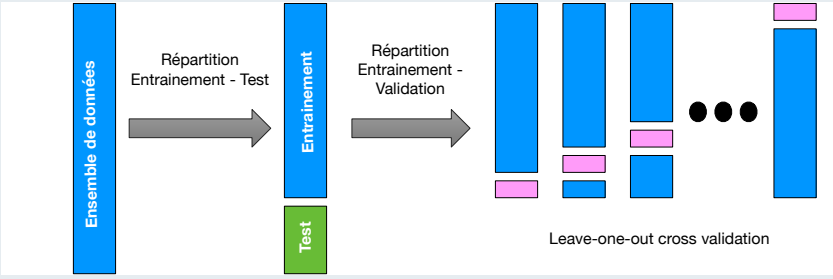
- La stratification est le processus qui consiste à diviser la population générale en sous-groupes homogènes avant l'échantillonnage.
- L'objectif de la stratification est de créer des folds de sorte à ce qu'elles contiennent à peu près les mêmes proportions d'individus de chaque classe.
  $\Longrightarrow$ 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 (2/2)

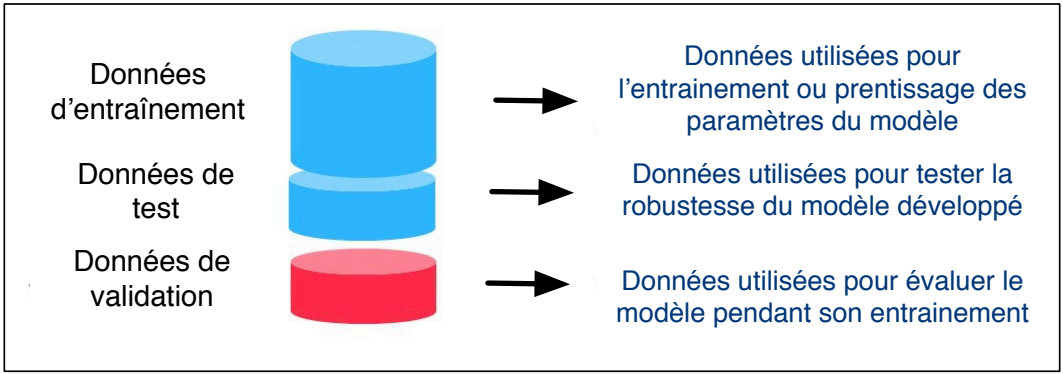


Validation croisée : Leave-one out cross validation

- 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'entrainement



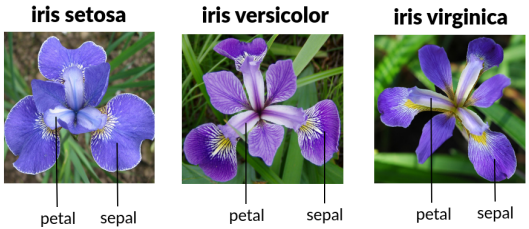
Répartition des données : Résumé



Base de données IRIS

La base de données iris comprend 150 échantillons.

- Les trois classes sont setosa, versicolor et virginica
- Les fleurs sont caractérisées par : sepal_length, sepal_width, petal_length et petal_width



Exemple iris : Répartition des données train, test

```
1 data.groupby('species').size()
```

```
species
setosa        50
versicolor    50
virginica     50
dtype: int64
```

—> Contenu de la base de données au complet

```
1 train, test = train_test_split(data, test_size = 0.3, stratify = data['species'], random_state = 10)
2 test.groupby('species').size()
```

```
species
setosa        15
versicolor    15
virginica     15
dtype: int64
```

—> Répartition de la base de données en 70 pour l'entrainement et 30% pour le test

```
1 train, test = train_test_split(data, train_size = 0.9, stratify = data['species'], random_state = 10)
2 test.groupby('species').size()
```

```
species
setosa        5
versicolor    5
virginica     5
dtype: int64
```

—> Répartition de la base de données en 90% pour l'entrainement et 10% pour le test

Exemple iris : Création des matrices des caractéristiques (*features*) et des calsses

```
X_train = train[['sepal_length','sepal_width','petal_length','petal_width']]
y_train = train.species
X_test = test[['sepal_length','sepal_width','petal_length','petal_width']]
y_test = test.species
fn = ["sepal_length", "sepal_width", "petal_length", "petal_width"]
cn = ['setosa', 'versicolor', 'virginica']
```

Plan du cours

Sélection des caractéristiques

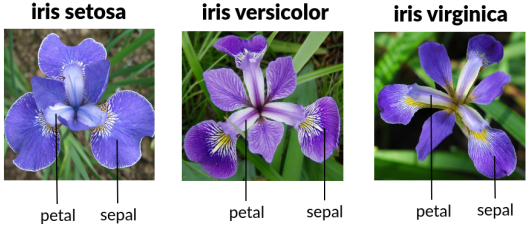
Qu'est ce que l'extraction des caractéristiques ? (1/3)

- Parmi les aspects importants de l'apprentissage machine, on retrouve l'extraction et la sélection de caractéristiques.
- L'extraction des caractéristiques permet d'obtenir un ensemble de variables informatives.

Qu'est ce que l'extraction des caractéristiques ? (2/3)

Par exemple, dans le cas de la base de données `iris`

- Les trois classes sont `setosa`, `versicolor` et `virginica`
- Les caractéristiques sont déjà extraites. Il s'agit de `sepal_length`, `sepal_width`, `petal_length` et `petal_width`



Qu'est ce que l'extraction des caractéristiques ? (3/3)

	A	B	C	D	E
1	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
2	5.1	3.5	1.4	0.2	Iris-setosa
3	4.9	3	1.4	0.2	Iris-setosa
4	4.7	3.2	1.3	0.2	Iris-setosa
5	4.6	3.1	1.5	0.2	Iris-setosa
6	5	3.6	1.4	0.2	Iris-setosa
7	5.4	3.9	1.7	0.4	Iris-setosa

Plan du cours

Structure

- 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 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)

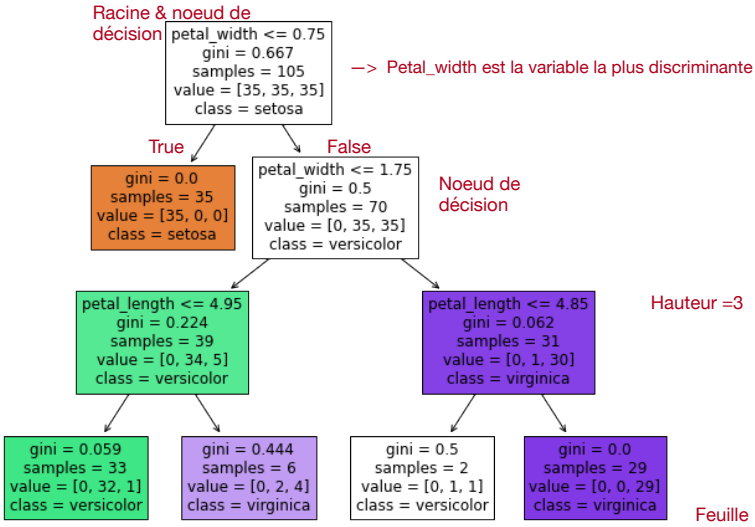
- L'algorithme récursif CART (*Classification And Regression Trees*) permet la construction d'un arbre de décisions par la maximisation de l'indice de Gini.
- L'indice de Gini mesure l'impureté, qui est un concept très utile dans la construction des arbres de décisions.
- La qualité d'un nœuds et son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

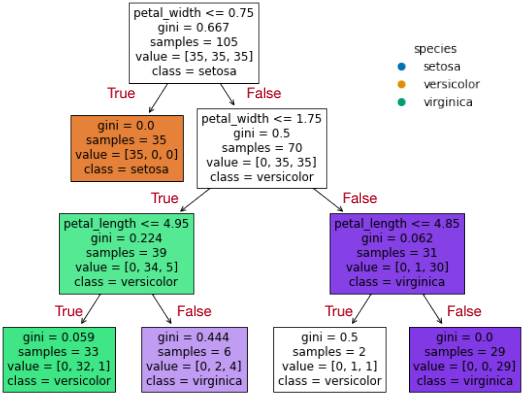
Exemple iris : Entrainement de l'arbre de décision (par défault le critère = gini)

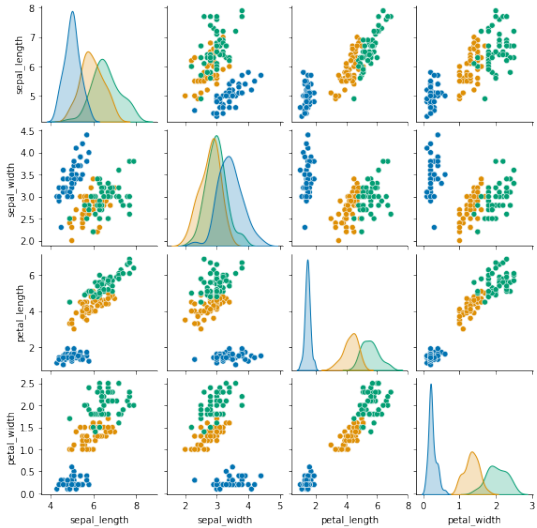
```
1 from sklearn.tree import DecisionTreeClassifier, plot_tree
2
3 dt = DecisionTreeClassifier(max_depth = 3, random_state = 1)
4 dt.fit(X_train,y_train)
```

Visualisation de l'arbre de décision

```
1 from sklearn.tree import DecisionTreeClassifier, plot_tree
2
3 plt.figure(figsize = (10,8))
4 plot_tree(dt, feature_names = fn, class_names = cn, filled = True)
```


Racine & noeud de décision
petal_width <= 0.75
gini = 0.667
samples = 105
value = [35, 35, 35]
class = setosa
—> Petal_width est la variable la plus discriminante
True
False
gini = 0.0
samples = 35
value = [35, 0, 0]
class = setosa
petal_width <= 1.75
gini = 0.5
samples = 70
value = [0, 35, 35]
class = versicolor
Noeud de décision
petal_length <= 4.95
gini = 0.224
samples = 39
value = [0, 34, 5]
class = versicolor
petal_length <= 4.85
gini = 0.062
samples = 31
value = [0, 1, 30]
class = virginica
Hauteur =3
gini = 0.059
samples = 33
value = [0, 32, 1]
class = versicolor
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
gini = 0.5
samples = 2
value = [0, 1, 1]
class = versicolor
gini = 0.0
samples = 29
value = [0, 0, 29]
class = virginica
Feuille


petal_width <= 0.75
gini = 0.667
samples = 105
value = [35, 35, 35]
class = setosa
True
False
gini = 0.0
samples = 35
value = [35, 0, 0]
class = setosa
petal_width <= 1.75
gini = 0.5
samples = 70
value = [0, 35, 35]
class = versicolor
True
False
petal_length <= 4.95
gini = 0.224
samples = 39
value = [0, 34, 5]
class = versicolor
petal_length <= 4.85
gini = 0.062
samples = 31
value = [0, 1, 30]
class = virginica
True
False
True
False
gini = 0.059
samples = 33
value = [0, 32, 1]
class = versicolor
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
gini = 0.5
samples = 2
value = [0, 1, 1]
class = versicolor
gini = 0.0
samples = 29
value = [0, 0, 29]
class = virginica
species
setosa
versicolor
virginica


sepal_length
sepal_width
petal_length
petal_width

Plan du cours

Taux de classification

- 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 :

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

- 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 (1/3)

- 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 (2/3)

- Les valeurs de la diagonale de la matrice représentent le nombre de formes correctement classifié.
- 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 τ_i, $i = 1, 2, 3, ...c$.
  c étant le nombre de classes.

Matrice de confusion (3/3)

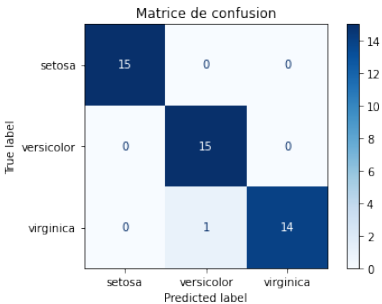
- Taux de classification par classe :
  $\tau_{Cat} = 4/6$
  $\tau_{Fish} = 2/10$
  $\tau_{Hen} = 6/9$
- Nombre d'échantillons par classe
  $n_{Cat} = 6$, $n_{Fish} = 10$ et $n_{Hen} = 9$

		Classe réelle (Real)		
		Cat	Fish	Hen
Classe prédite (Predicted)	Cat	4	6	3
	Fish	1	2	0
	Hen	1	2	6

https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2

Exemple iris : Matrice de confusion

```
disp = metrics.plot_confusion_matrix(dt, X_test, y_test,
                                     display_labels=cn,
                                     cmap=plt.cm.Blues,
                                     normalize=None)
disp.ax_.set_title('Matrice de confusion');
```



Exemple iris : Matrice de confusion