

Fouille de Données

Data Mining

Classification - Partie 2

Plan du cours

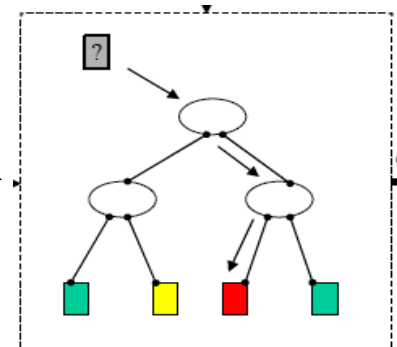
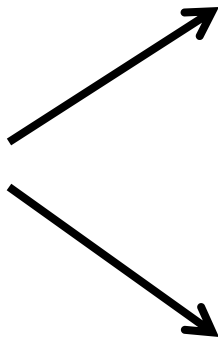
1. Classification associative
2. Méthodes d'évaluation d'un modèle
3. Combinaison de modèles
4. K plus proches voisins

Contexte

SAVOIR - PREDIRE - DECIDER



Données



Connaissances

Arbres de décision : **ID3**

Les arbres de décision

Algorithmes de construction d'arbres de décision

Plusieurs problèmes :

- Comment choisir l'attribut qui sépare le mieux l'ensemble d'exemples? On parle souvent de la variable de segmentation (Split).
- Comment choisir les critères de séparation d'un ensemble selon l'attribut choisi, et comment ces critères varient selon que l'attribut soit numérique ou nominal ?
- Quel est le nombre optimal du nombre de critères qui minimise la **taille de l'arbre** et maximise la précision ?
- Quels sont les critères d'arrêt de ce partitionnement, sachant que souvent l'arbre est d'une **taille** gigantesque ?

Les arbres de décision

Algorithmes de construction d'arbres de décision

➤ La bonne taille de l'arbre:

- Eviter l'overfitting – Sur-apprentissage: anomalies, bruits, erreurs, etc.
- L'arbre construit peut être d'une taille importante.
- => Opérations **d'élagage**.
- Eliminer les branches les moins significatives.
- Elagage avant ou après apprentissage : pré-élagage ou post-élagage.

Les arbres de décision

Choix de la bonne taille de l'arbre

Pré-élagage

- Effectué **lors** de la construction de l'arbre.
- Au moment du calcul du gain d'information, décider de l'importance ou non de sa subdivision.
- Arrêter la construction lorsqu'il n'y a pas d'association statistiquement significative entre un attribut et la classe d'un nœud particulier.
- Couper complètement des branches qui peuvent être générées.

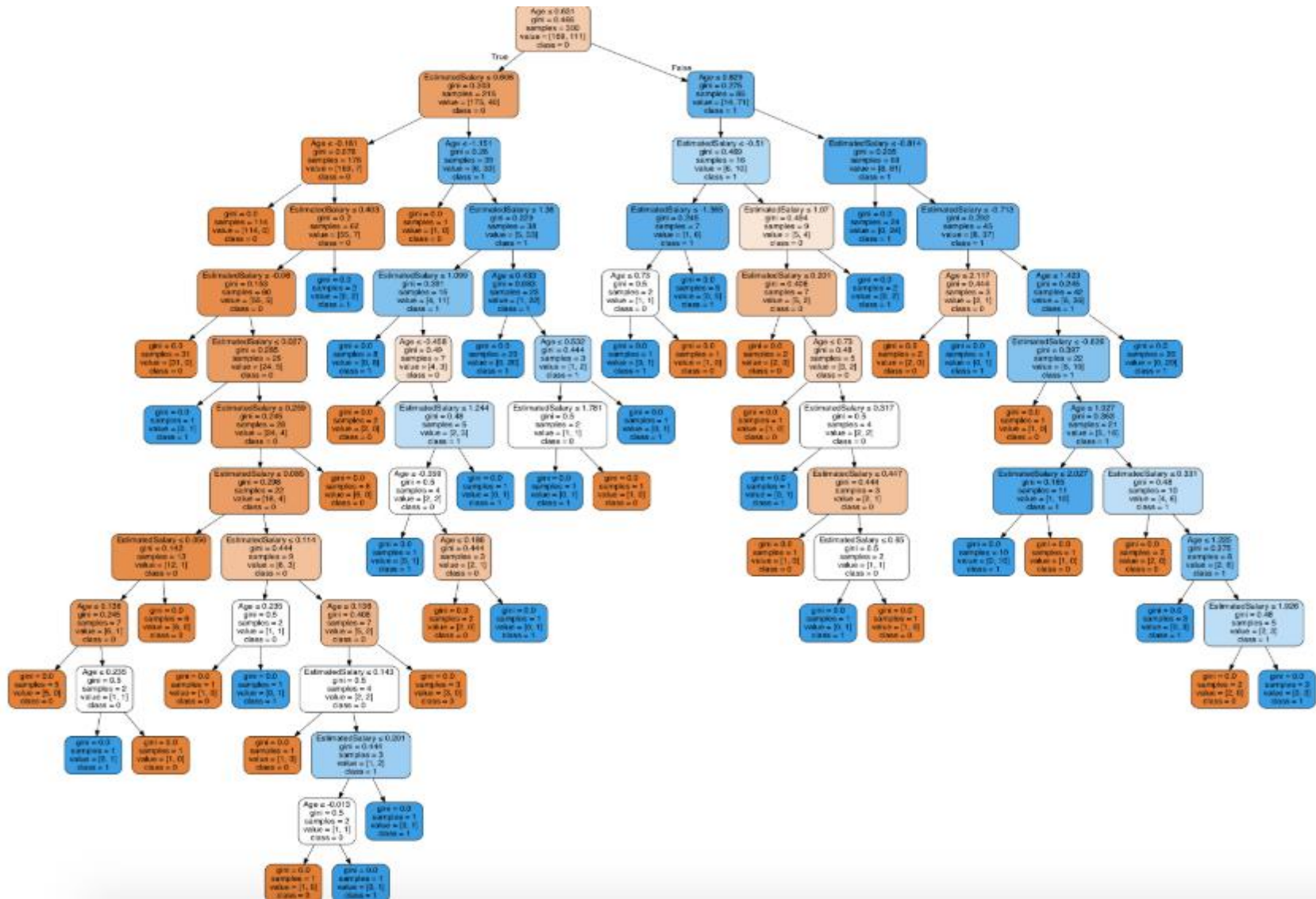
Les arbres de décision

Choix de la bonne taille de l'arbre

Post-élagage

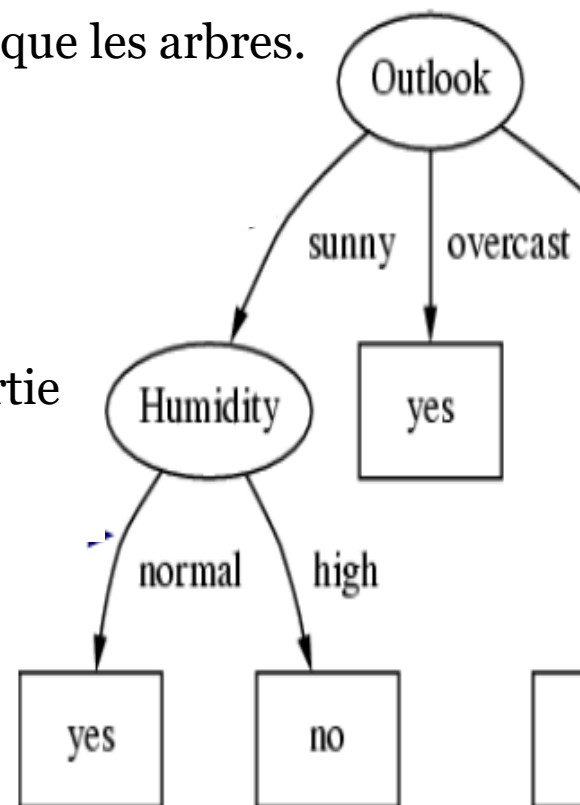
- Effectué **après** la construction de l'arbre en coupant des sous arbres entiers et en les remplaçant par des feuilles représentant la classe la plus fréquente dans l'ensemble des données de cet arbre.
- On commence de la racine et on descend.
- Pour chaque nœud interne (non feuille), on mesure sa performance avant et après sa coupure (son remplacement par une feuille).
- Si la différence est peu importante, on coupe le sous arbre et on le remplace par une feuille.

D'arbres de décision aux règles d'association



D'arbres de décision aux règles d'association

- Les règles sont plus **simples à lire et à interpréter** que les arbres.
- Une règle d'association pour chaque feuille.
- Connaissance sous forme de : **IF-THEN**
- Extraction des règles solides qui ont dans leur partie droite l'attribut classe.
- Exemple :

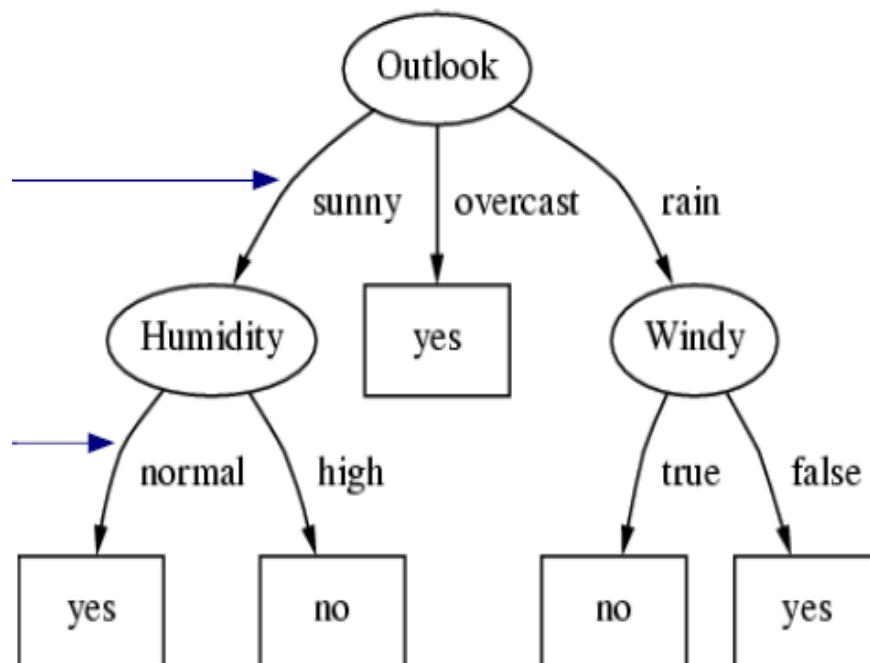


IF (Outlook=sunny) **AND** (Humidity=normal) **THEN** Play =yes

D'arbres de décision aux règles d'association

➤ Exemple :

- ✓ **IF** (Outlook=sunny) **AND** (Humidity=high) **THEN** Play =no
- ✓ **IF** (Outlook=rain) **AND** (Wind=true) **THEN** Play =no
- ✓ **OTHERWISE** play=yes



D'arbres de décision aux règles d'association

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

IF (Outlook=sunny) **and** (Humidity=high) **THEN** Play =no

D'arbres de décision aux règles d'association

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

IF (Outlook=rainy) **and** (Wind=true) **THEN** Play =no

D'arbres de décision aux règles d'association

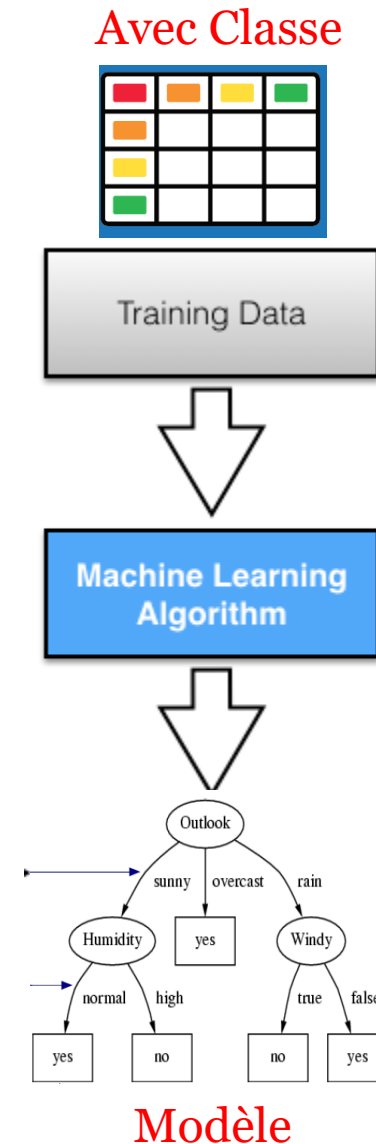
Outlook	Temp	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

OTHERWISE play=yes

Evaluation et validation du modèle

Deux étapes :

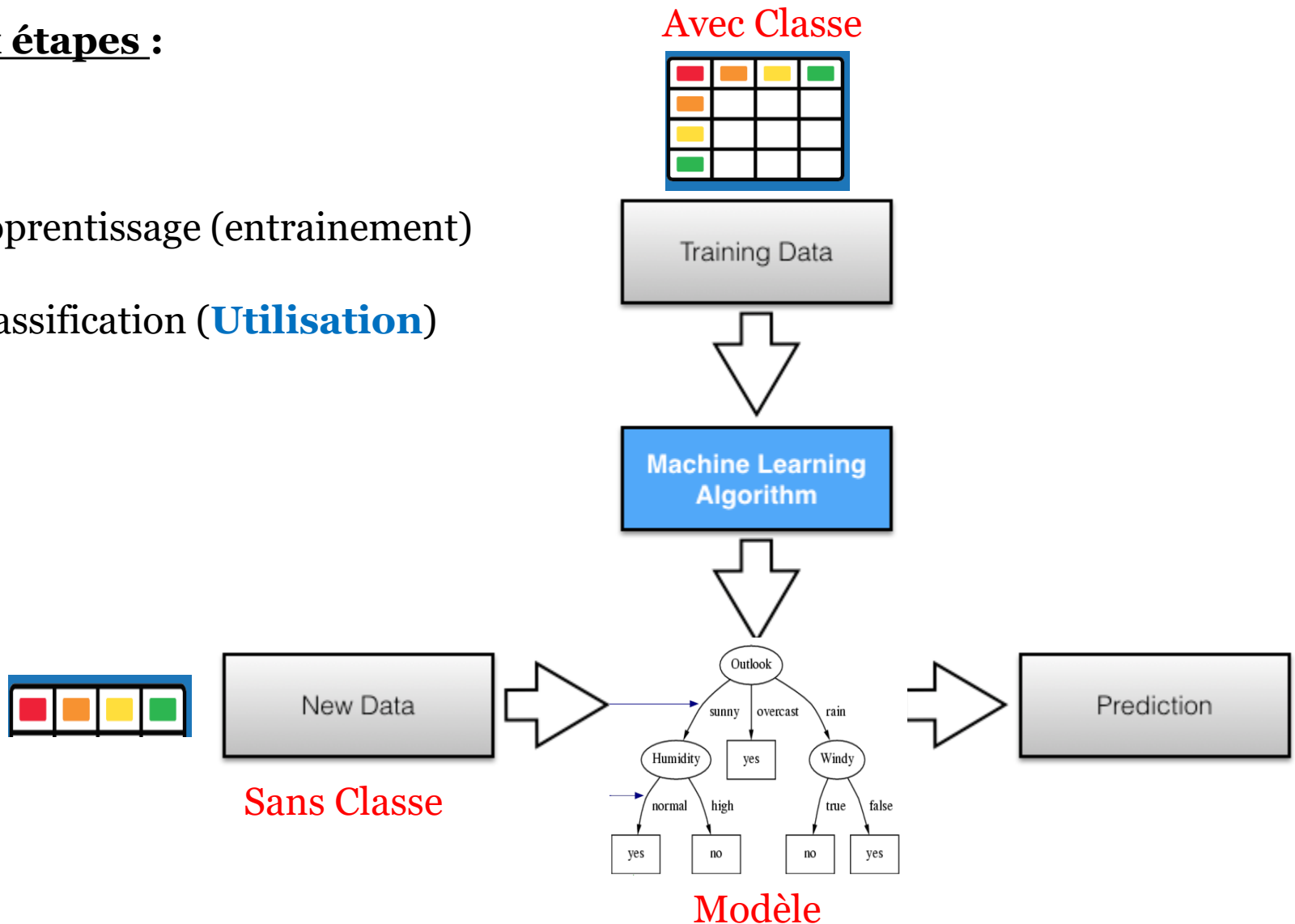
1. Apprentissage (**entraînement**)



Evaluation et validation du modèle

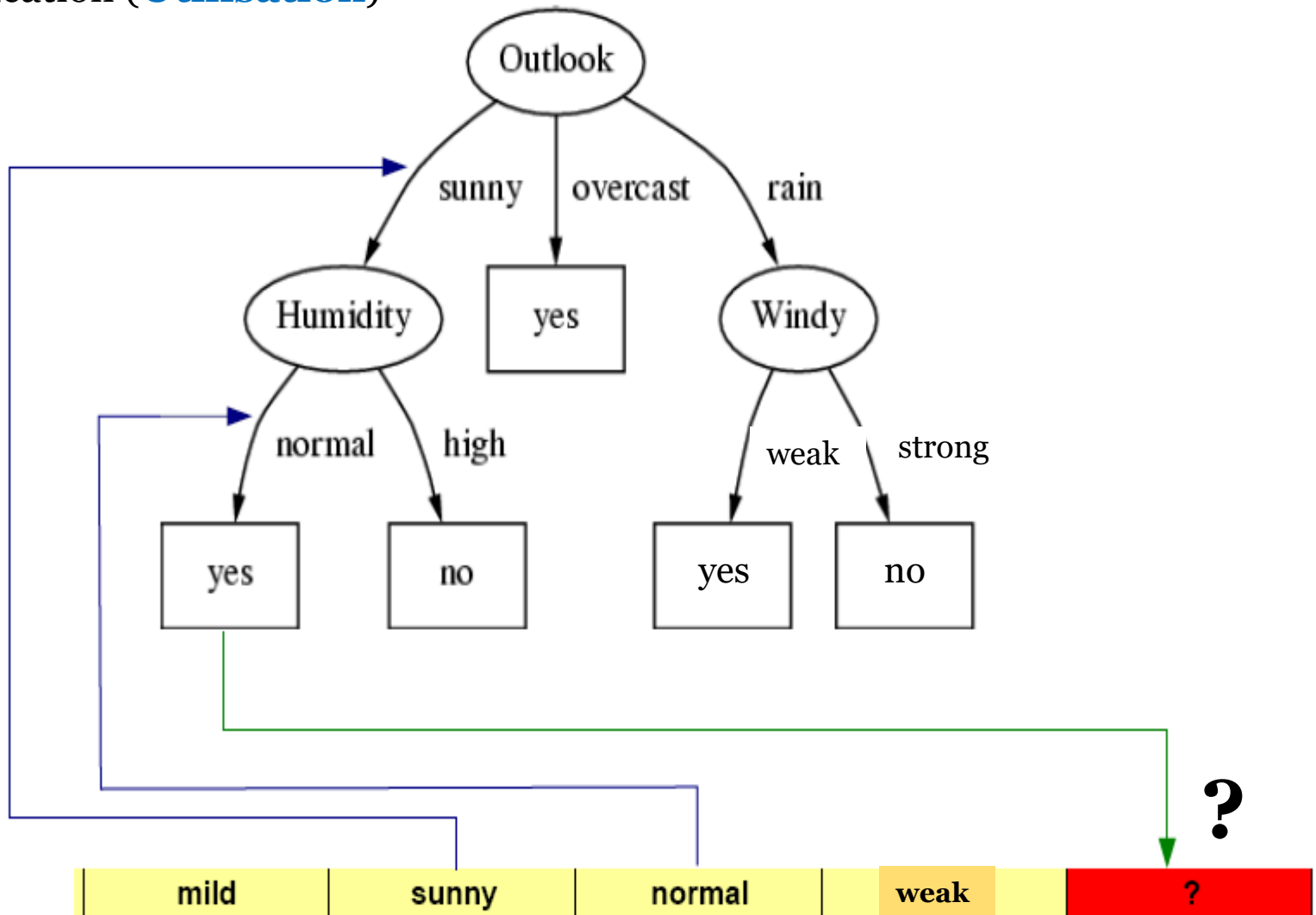
Deux étapes :

1. Apprentissage (entraînement)
2. Classification (**Utilisation**)



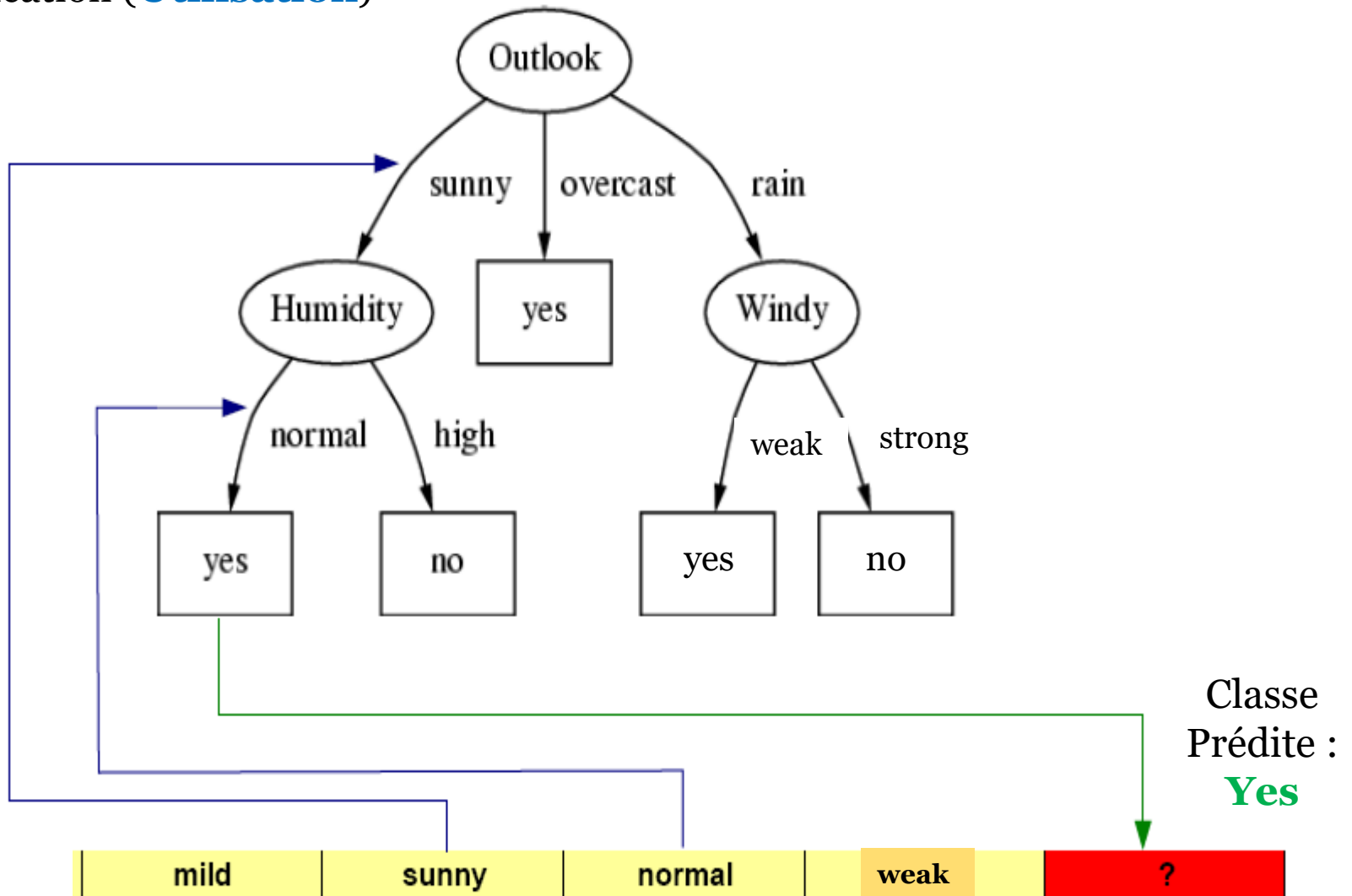
Evaluation et validation du modèle

Classification (**Utilisation**)



Evaluation et validation du modèle

Classification (**Utilisation**)



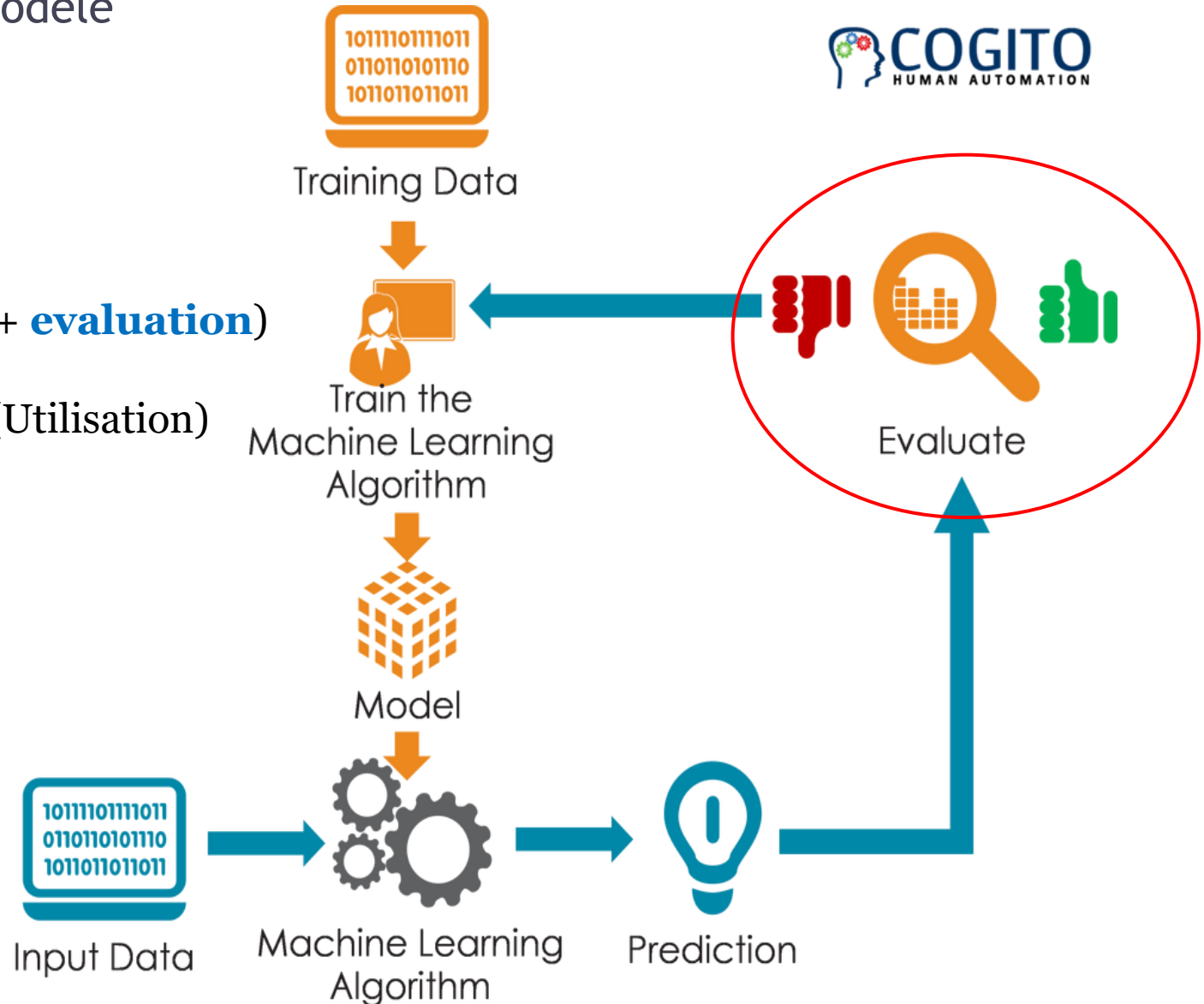
Méthodes d'évaluation et de validation

- L'apprentissage d'un modèle se fait à base de plusieurs paramètres.
- Le choix de leurs valeurs se fait à travers plusieurs essais et évaluations.
- Paramètres optimaux => précision de 100%.
- Problème d'overfitting => Mesure de précision non suffisante.
- D'où les méthodes de **validation** et d'**évaluation**.
- Tirer des conclusions sur le comportement d'un modèle face à tout l'espace d'exemples en limitant l'influence des exemples d'entraînement.

Evaluation du modèle

Deux étapes:

1. Apprentissage
(entraînement + **evaluation**)
1. Classification (Utilisation)



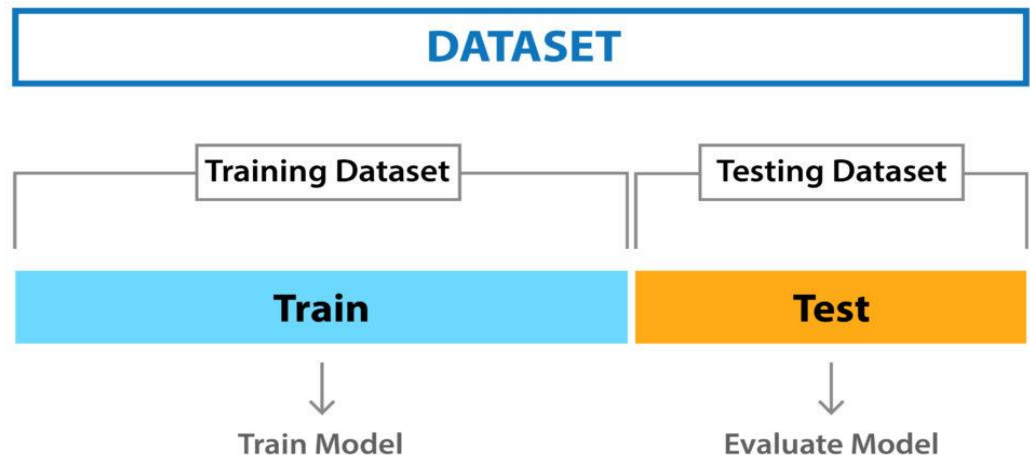
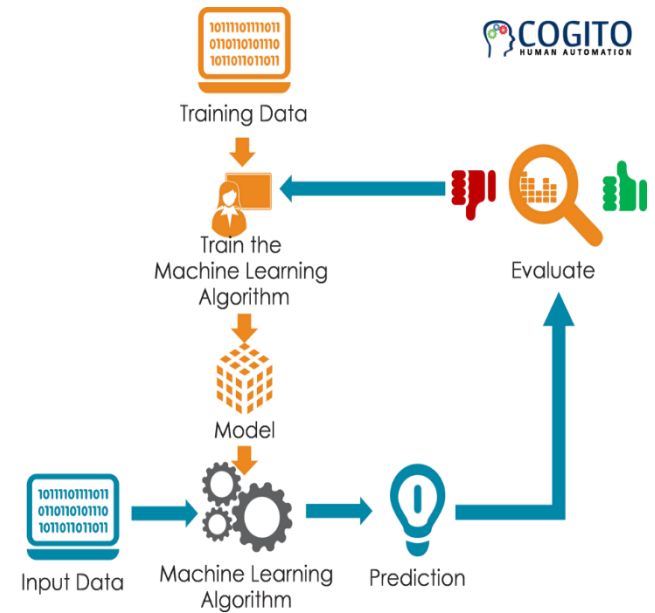
Méthodes d'évaluation

Deux étapes:

1. Apprentissage
(entraînement + **evaluation on Test Set**)
1. Classification (Utilisation)

Deux bases d'exemples:

1. Training Set
2. Test Set



Méthodes d'évaluation

Deux étapes:

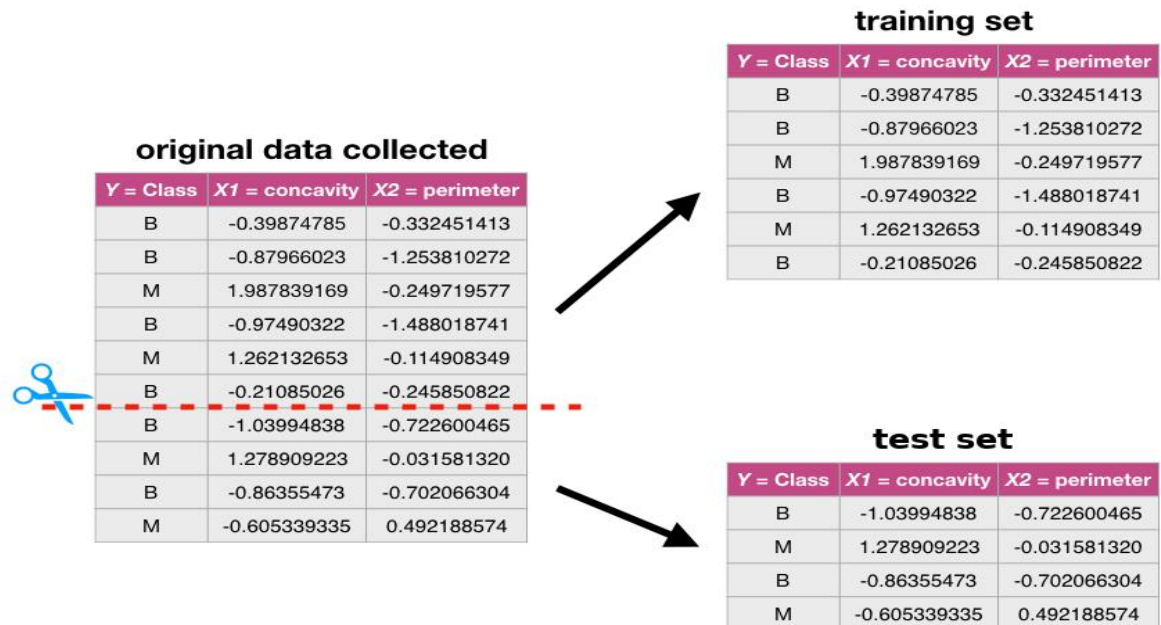
1. Apprentissage
(entraînement + **evaluation on Test Set**)

1. Utilisation

Deux bases d'exemples:

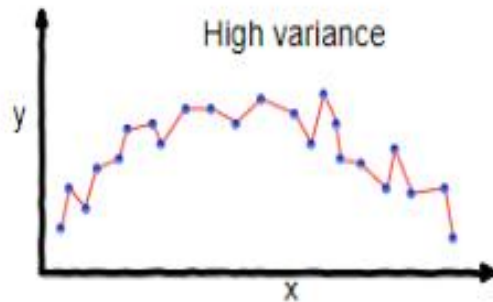
1. Training Set
2. Test Set

Creating the training and test sets

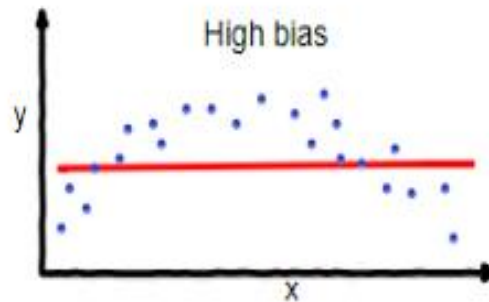


Evaluation du modèle

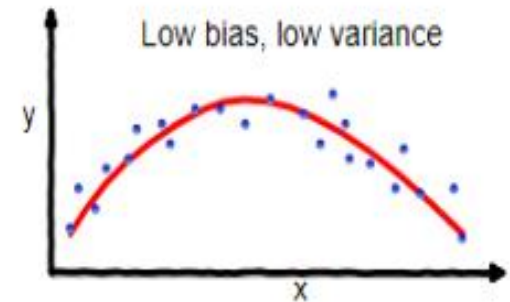
- Evaluer pour s'assurer de la **capacité de généralisation** en dehors des données d'entraînement.
- Overfitting/Underfitting – **Variance/Biais tradeoff**
- Permet de qualifier le comportement du modèle appris sur les données non utilisées lors de l'apprentissage.



overfitting



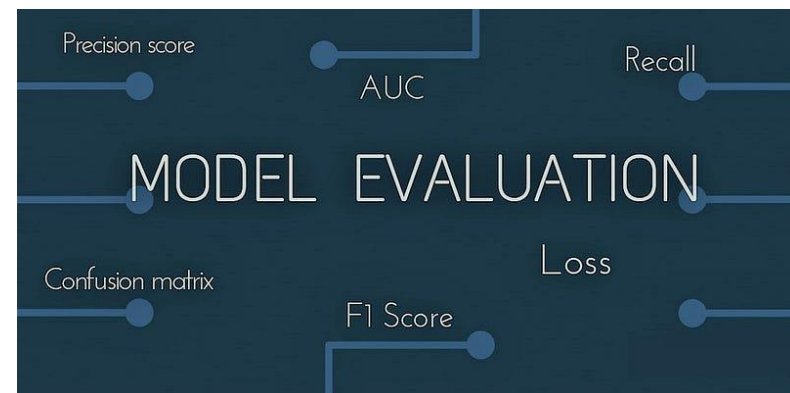
underfitting



Good balance

Evaluation du modèle

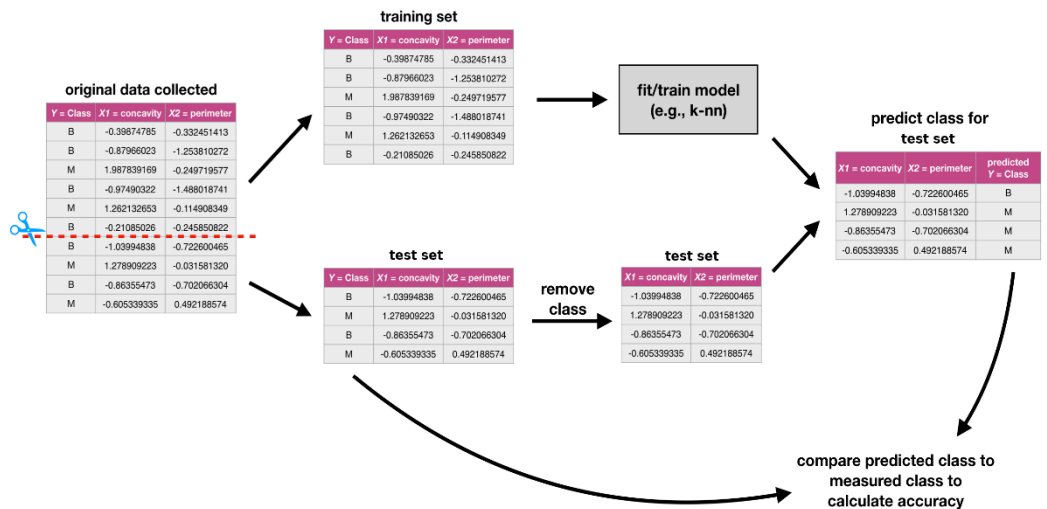
- Evaluer pour s'assurer de la capacité de généralisation en dehors des données d'entraînement.
- Overfitting/Underfitting – Variance/Biais tradeoff
- Permet de qualifier le comportement du modèle appris sur les données non utilisées lors de l'apprentissage.
- Sur les exemples d'entraînement ou autres exemples réservés pour le test.
- Différentes **mesures/metrics d'évaluation** – **classification** :
 - ✓ Accuracy
 - ✓ Sensitivité
 - ✓ Spécificité
 - ✓ Moyenne harmonique, F1 Score
 - ✓ Etc.



Evaluation du modèle

Précision générale P (**Accuracy**) d'un modèle

- Métrique intuitive, qui représente le rapport entre le nombre d'exemples correctement classés et le nombre total des exemples testés.
- = > Pourcentage des exemples correctement classés.
- Taux d'erreurs = $100 - P$



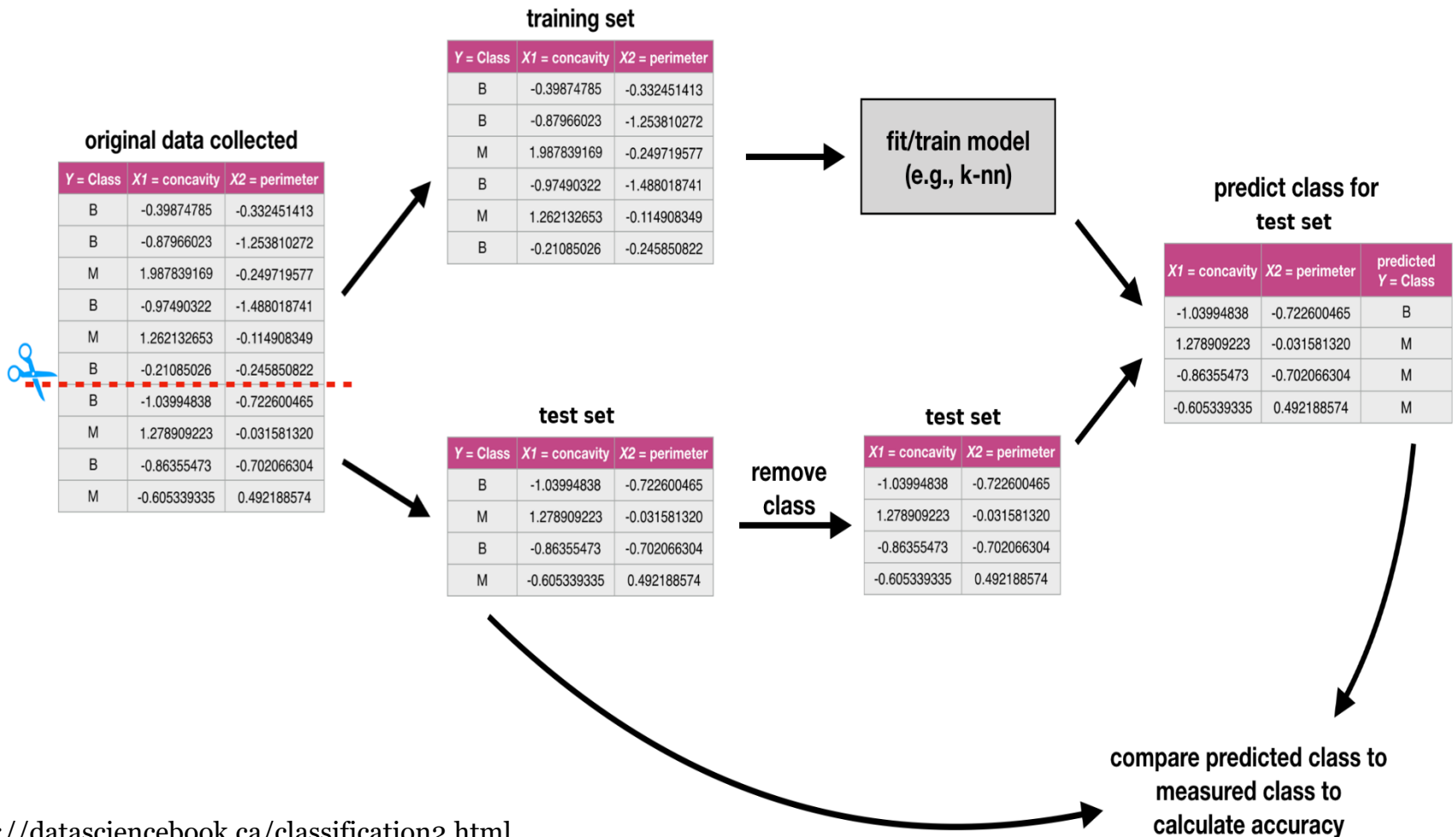
$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

$$\text{accuracy} = \frac{\# \text{ correct predictions}}{\# \text{ total predictions}}$$

Evaluation du modèle

Précision générale P (**Accuracy**) d'un modèle

$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$



Evaluation du modèle

Matrice de confusion

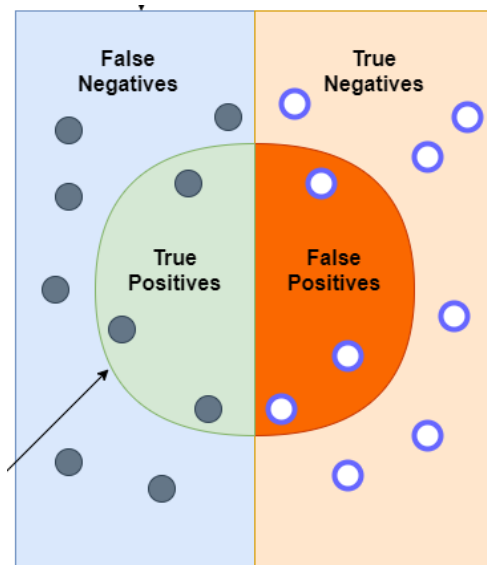
- Aucune information sur **la nature des erreurs**.
- Exemple : considérer un échantillon non cancéreux alors qu'il l'est, est beaucoup plus grave de considérer un échantillon cancéreux alors qu'il ne l'est pas.
- Cas de classification binaire : Observation (y) et Prédiction (f)

$$\left\{ \begin{array}{llll} \hat{f}(x_i) = \text{Yes} & \text{et} & y_i = \text{Yes} & \text{correcte positive} \\ \hat{f}(x_i) = \text{Yes} & \text{et} & y_i = \text{No} & \text{fausse positive} \\ \hat{f}(x_i) = \text{No} & \text{et} & y_i = \text{No} & \text{correcte négative} \\ \hat{f}(x_i) = \text{No} & \text{et} & y_i = \text{Yes} & \text{fausse négative} \end{array} \right.$$

Evaluation du modèle

Matrice de confusion

- **CP** : classe positive considérée positive - TP
- **CN** : classe negative considérée negative - TN
- **FP** : classe negative considérée positive - FP
- **FN** : classe positive considérée negative - FN



		Prédiction	
		Yes	No
Observation	Yes	CP	FN
	No	FP	CN

Evaluation du modèle

Matrice de confusion

➤ **Attention :**

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

		Predicted classes	
		Negative 0	Positive 1
Actual classes	Negative 0	TN	FP
	Positive 1	FN	TP

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Evaluation du modèle

Matrice de confusion

- Modèle sans erreurs : $CP + CN = N$
- Multi classes : nombre de colonnes = nombre de classes
- Accuracy (Précision globale):

$$P = \frac{CP + CN}{CP + FP + CN + FN}$$

		Prédiction	
		Yes	No
Observation	Yes	CP	FN
	No	FP	CN

Evaluation du modèle

Matrice de confusion

Exemple 1:

- Chat ou Chien ?
- N = 14 exemples/animaux testés
- Deux Classes : Chat, Chien
- En réalité : **8** Chats et **6** Chiens.
- Le modèle a prédit :
 - ✓ À partir des 8 chats : 5 l'étaient seulement, les 3 restants étaient Chiens.
 - ✓ A partir des 6 Chiens : 4 l'étaient seulement, les 2 autres étaient Chats.

		Prédiction	
		Chat	Chien
Observation	Chat	CP	FN
	Chien	FP	CN

Evaluation du modèle

Matrice de confusion

Exemple 1:

- Chat ou Chien ?
- N = 14 exemples/animaux testés
- Deux Classes : Chat, Chien
- En réalité : **8** Chats et **6** Chiens.
- Le modèle a prédit :
 - ✓ Depuis les 8 **Chats** : 5 l'étaient seulement, les 3 restants étaient Chiens.
 - ✓ Depuis les 6 **Chiens** : 4 l'étaient seulement, les 2 autres étaient Chats.

		Prédiction	
		Chat	Chien
Observation	Chat	5	3
	Chien	2	4

Evaluation du modèle

Matrice de confusion

Exemple 2:

		Prédiction		
		Chat	Chien	Lapin
Observation	Chat	5	3	0
	Chien	2	3	1
	Lapin	0	2	11

- Chat, Chien, ou Lapin ?
- N = 27 exemples/animaux testés
- Trois Classes : Chat, Chien, Lapin
- En réalité : **8** Chats et **6** Chiens, **13** Lapins.

Evaluation du modèle

Matrice de confusion

Exemple 2:

On considère Chat comme classe positive.

		Prédiction		
		Chat	Chien	Lapin
Observation	Chat	5	3	0
	Chien	2	3	1
	Lapin	0	2	11

		Prédiction	
		Chat	Chien&Lapin
Observation	Chat	5	3
	Chien&Lapin	2	17

Evaluation du modèle

Matrice de confusion

Exemple 2:

On considère Chat comme classe positive.

		Prédiction		
		Chat	Chien	Lapin
Observation	Chat	5	3	0
	Chien	2	3	1
	Lapin	0	2	11

		Prédiction	
		Chat	Chien&Lapin
Observation	Chat	5	3
	Chien&Lapin	2	17

		Predicted Class			
		C_1	C_2	...	C_N
Actual Class	C_1	$C_{1,1}$	FP	...	$C_{1,N}$
	C_2	FN	TP	...	FN

	C_N	$C_{N,1}$	FP	...	$C_{N,N}$

Evaluation du modèle

Matrice de confusion

Autres mesures :

- **Sensitivité** – True Positive Rate:

$$Sv = \frac{CP}{CP + FN}$$

- **Spécificité** – True Negative Rate:

$$Sp = \frac{CN}{CN + FP}$$

- **Moyenne harmonique** :

$$Mh = \frac{2 * Sv * Sp}{Sv + Sp}$$

		Prédiction	
		Yes	No
Observation	Yes	CP	FN
	No	FP	CN

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Evaluation du modèle

Matrice de confusion: Autres mesures

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{TP + FN}$ Recall
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{TN + FP}$
		Precision $\frac{TP}{TP + FP}$	Negative Predictive Value $\frac{TN}{TN + FN}$	Accuracy $\frac{TP + TN}{TP + TN + FP + FN}$

Evaluation du modèle

Matrice de confusion: Autres mesures

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Recall $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	F1 Score $2 * \frac{Precision * Recall}{Precision + Recall}$

Evaluation du modèle

Matrice de confusion

Autres mesures :

Metric	Formula	Evaluation focus
Accuracy	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$	Overall effectiveness of a classifier
Precision	$PRC = \frac{TP}{TP + FP}$	Class agreement of the data labels with the positive labels given by the classifier
Sensitivity	$SNS = \frac{TP}{TP + FN}$	Effectiveness of a classifier to identify positive labels. Also called true positive rate (TPR)
Specificity	$SPC = \frac{TN}{TN + FP}$	How effectively a classifier identifies negative labels. Also called true negative rate (TNR)
F_1 score	$F_1 = 2 \frac{PRC \cdot SNS}{PRC + SNS}$	Combination of precision (PRC) and sensitivity (SNS) in a single metric
Geometric mean	$GM = \sqrt{SNS \cdot SPC}$	Combination of sensitivity (SNS) and specificity (SPC) in a single metric
Area under (ROC) curve	$AUC = \int_0^1 SNS \cdot dSPC$	Combined metric based on the receiver operating characteristic (ROC) space (Powers, 2011)

Classification performance metrics based on the confusion matrix.

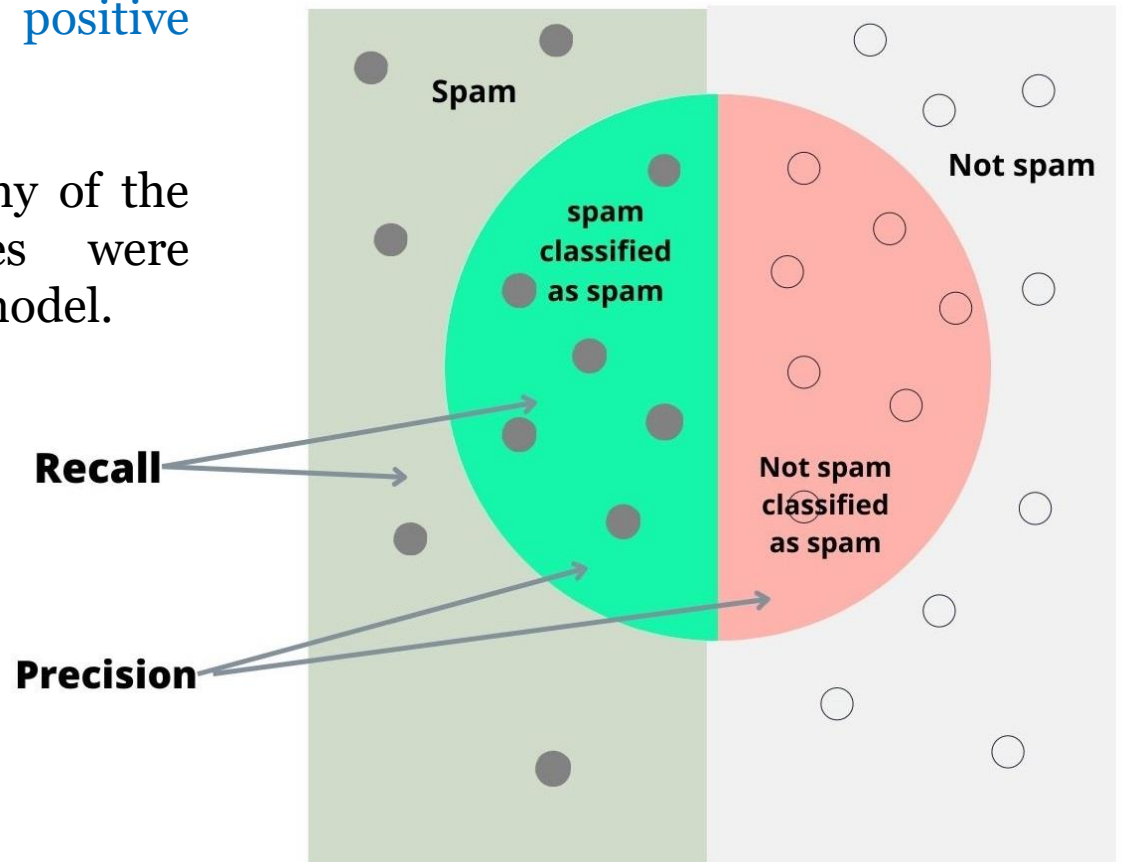
Evaluation du modèle

Precision and Recall Interpretation

Precision measures how many of the instances **predicted as positive** are actually correct.

Recall measures how many of the **actual positive** instances were correctly identified by the model.

TP	FN
FP	TN



Evaluation du modèle

Precision and Recall Interpretation

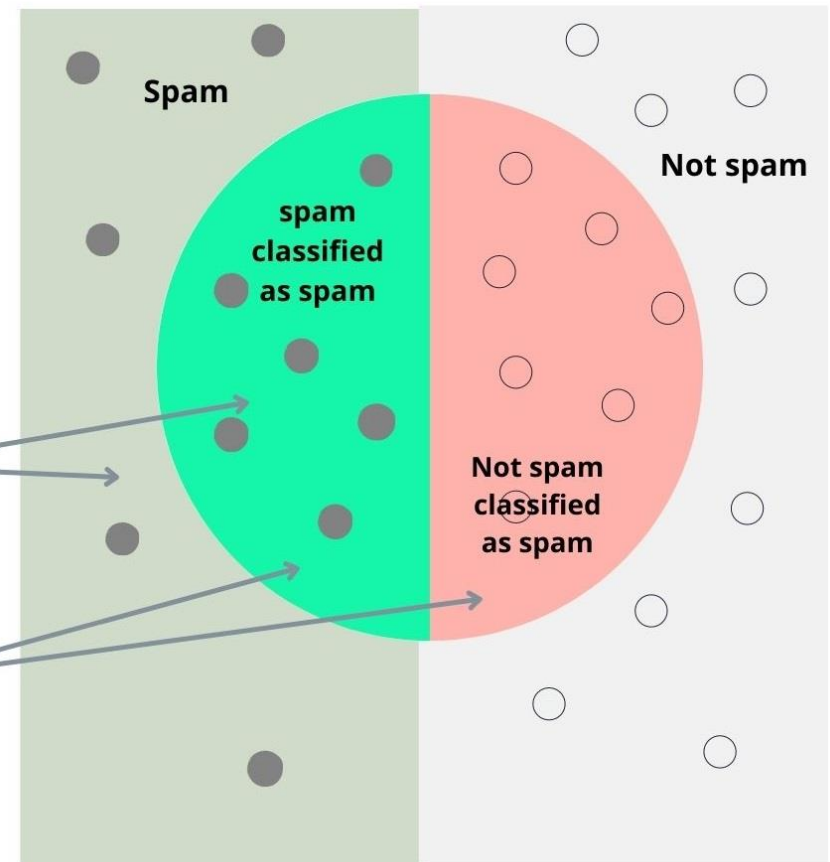
Precision: Of all the emails the model predicted as spam, how many were truly spam?

Ex: If it said 12 emails predicted spam and 6 really were, precision = $6/(6+6)$.

Precision focuses on the **model's predictions** — **how accurate** its positive predictions are.

Recall

Precision



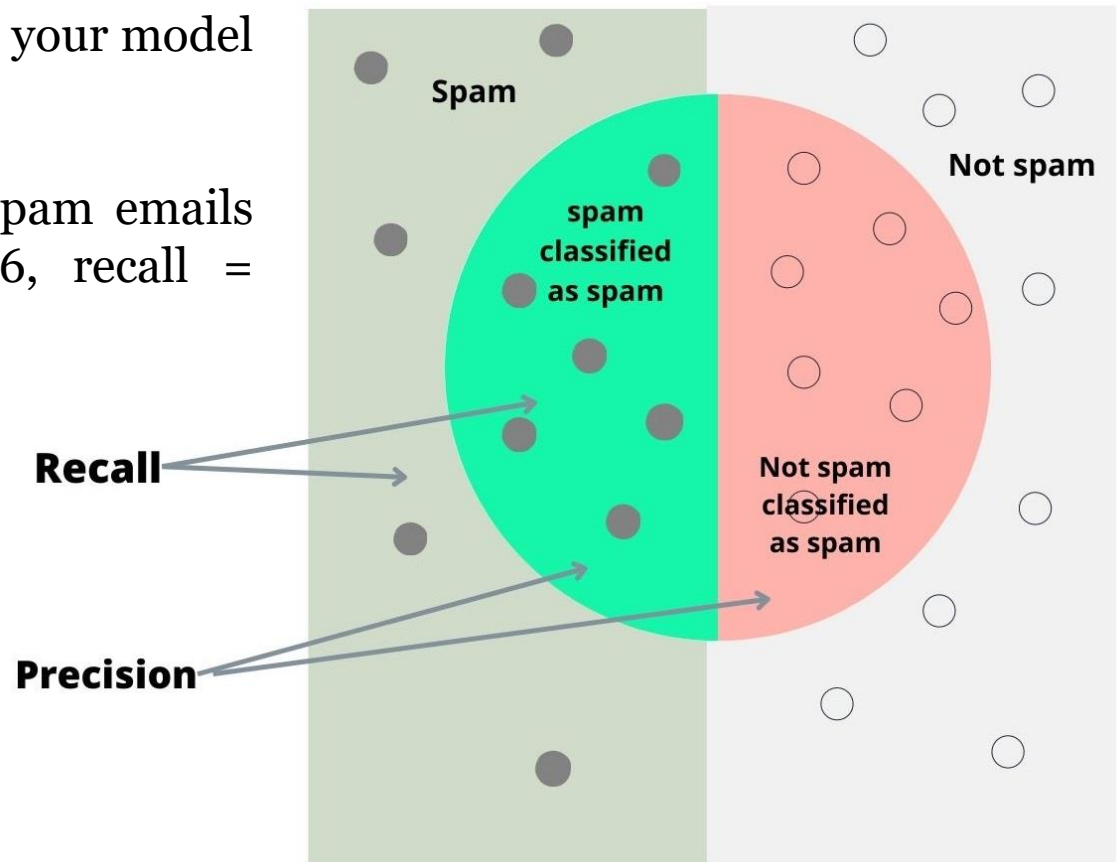
Evaluation du modèle

Precision and Recall Interpretation

Recall: Of all the real spam emails in your dataset, how many did your model catch?

Ex: If there were 11 real spam emails and your model caught 6, recall = $6/(6+5)$.

Recall focuses on the **actual dataset** - **how complete** the model's detection of real positives is.



Série TD 2 - Exercice 1

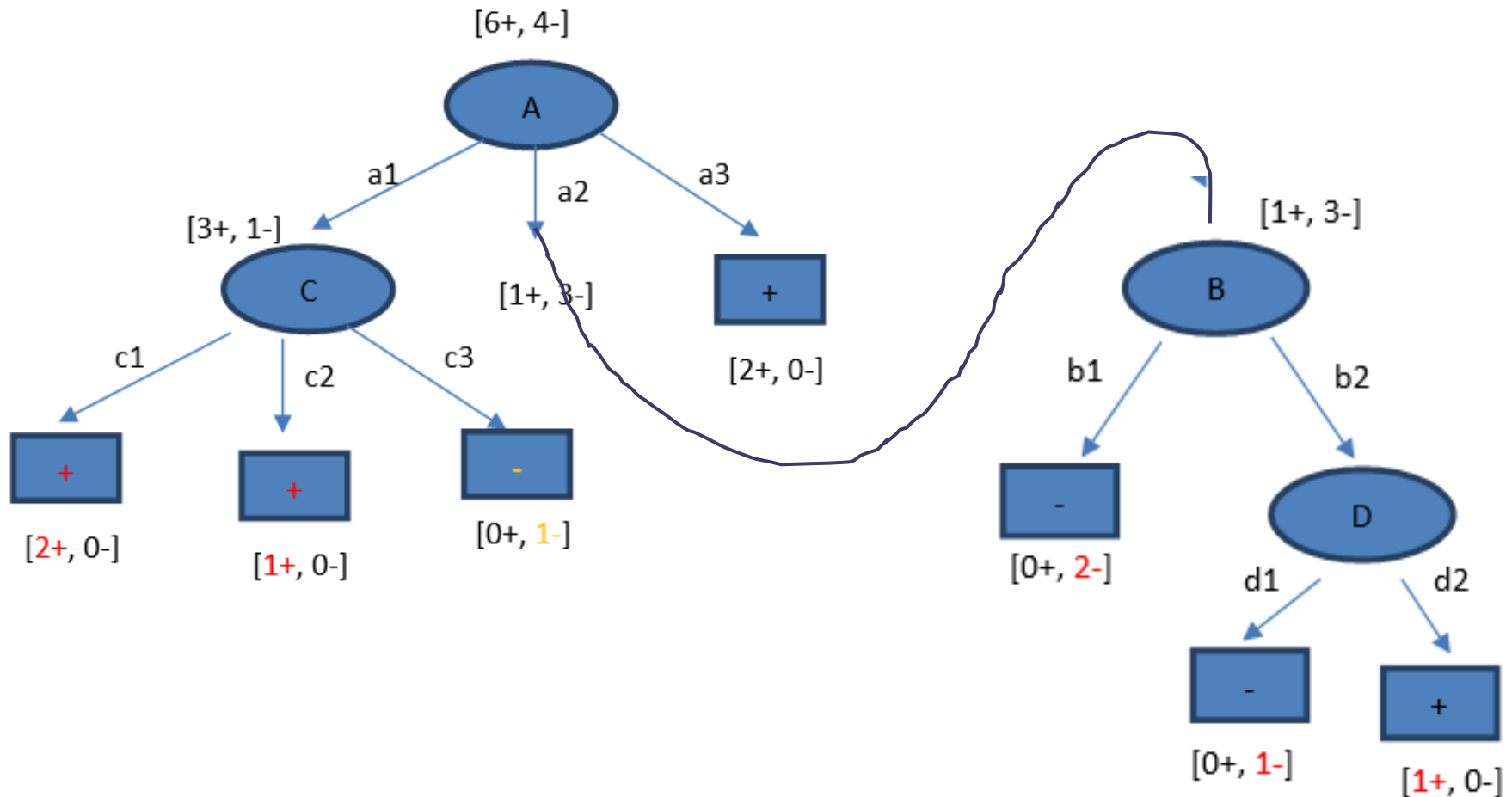
	A	B	C	D	Classe
E1	a1	b1	c1	d2	+
E2	a1	b2	c2	d2	+
E3	a1	b2	c3	d1	-
E4	a2	b1	c1	d1	-
E5	a2	b2	c1	d1	-
E6	a2	b2	c1	d2	+
E7	a1	b1	c1	d1	+
E8	a2	b1	c2	d2	-
E9	a3	b1	c3	d1	+
E10	a3	b2	c2	d2	+

Série TD 2 - Exercice 1

1. Calculer l'entropie de l'ensemble S par rapport à la valeur de la classe.
2. Construire l'arbre de décision correspondant à l'ensemble S en utilisant l'algorithme ID3.
3. Etablir la matrice de confusion pour la table d'entraînement ci-dessus.
4. Donner la précision générale (accuracy) de l'arbre obtenu sur la table d'entraînement. Calculer la moyenne harmonique sur la même table.
5. Trouver la classe de l'exemple ayant les attributs (a2, b1, c3, d1) selon l'arbre construit.

Série TD 2 - Exercice 1

2. Construction de l'arbre de décision avec ID3 :



Série TD 2 - Exercice 1

1. Calculer l'entropie de l'ensemble S par rapport à la valeur de la classe.
2. Construire l'arbre de décision correspondant à l'ensemble S en utilisant l'algorithme ID3.
3. Etablir la **matrice de confusion** pour la **table d'entraînement** ci-dessus.

Série TD 2 - Exercice 1

3. Etablir la **matrice de confusion** pour la **table d'entraînement** ci-dessus.

	A	B	C	D	Classe	Classe Prédite
E1	a1	b1	c1	d2	+	+
E2	a1	b2	c2	d2	+	+
E3	a1	b2	c3	d1	-	-
E4	a2	b1	c1	d1	-	-
E5	a2	b2	c1	d1	-	-
E6	a2	b2	c1	d2	+	+
E7	a1	b1	c1	d1	+	+
E8	a2	b1	c2	d2	-	-
E9	a3	b1	c3	d1	+	+
E10	a3	b2	c2	d2	+	+

Série TD 2 - Exercice 1

3. La matrice de confusion : On pose la classe + comme **classe positive**.

	A	B	C	D	Classe	Classe Prédite	
E1	a1	b1	c1	d2	+	+	CP
E2	a1	b2	c2	d2	+	+	CP
E3	a1	b2	c3	d1	-	-	CN
E4	a2	b1	c1	d1	-	-	CN
E5	a2	b2	c1	d1	-	-	CN
E6	a2	b2	c1	d2	+	+	CP
E7	a1	b1	c1	d1	+	+	CP
E8	a2	b1	c2	d2	-	-	CN
E9	a3	b1	c3	d1	+	+	CP
E10	a3	b2	c2	d2	+	+	CP

Série TD 2 - Exercice 1

3. La matrice de confusion :

		Prédiction	
		Classe +	Classe -
Observation	Classe +	CP : 6	FN : 0
	Classe -	FP : 0	CN : 4

Série TD 2 - Exercice 1

4. La précision générale (accuracy) et la moyenne harmonique :

$$\textbf{Accuracy} = (CP+CN)/(CP+CN+FP+FN) = (6+4)/(6+4+0+0) = 1$$

→ 100%

$$\textbf{Moyenne harmonique} = (2*S_v*S_p)/(S_v+S_p) = (2*1*1)/(1+1) = 1$$

- $S_v = CP / (CP+FN) = 6/6 = 1$
- $S_p = CN / (CN+FP) = 4/4 = 1$

		Prédiction	
		Classe +	Classe -
Observation	Classe +	CP : 6	FN : 0
	Classe -	FP : 0	CN : 4

Série TD 2 - Exercice 1

$$\mathbf{F1\ Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) = (2 * 1 * 1) / (1 + 1) = 1$$

=> 100%

- **Precision** = $CP / (CP + FP) = 6/6 = 1$
- **Recall** = $CP / (CP + FN) = 6/6 = 1$

		Prédiction	
		Classe +	Classe -
Observation	Classe +	CP : 6	FN : 0
	Classe -	FP : 0	CN : 4

Série TD 2 - Exercice 1

1. Calculer l'entropie de l'ensemble S par rapport à la valeur de la classe.
2. Construire l'arbre de décision correspondant à l'ensemble S en utilisant l'algorithme ID3.
3. Etablir la **matrice de confusion** sur la **table de test** suivante

	A	B	C	D	Classe
E1	a1	b1	c1	d2	+
E2	a1	b2	c2	d2	-
E3	a1	b2	c3	d1	+
E4	a2	b1	c1	d1	+
E5	a2	b2	c1	d1	-

Série TD 2 - Exercice 1

1. Calculer l'entropie de l'ensemble S par rapport à la valeur de la classe.
2. Construire l'arbre de décision correspondant à l'ensemble S en utilisant l'algorithme ID3.
3. Etablir la **matrice de confusion** sur la **table de test** suivante

	A	B	C	D	Classe	Classe Prédite	
E1	a1	b1	c1	d2	+	+	CP
E2	a1	b2	c2	d2	-	+	FP
E3	a1	b2	c3	d1	+	-	FN
E4	a2	b1	c1	d1	+	-	FN
E5	a2	b2	c1	d1	-	-	CN

Série TD 2 - Exercice 1

3. La matrice de confusion sur la table de test :

		Prédiction	
		Classe +	Classe -
Observation	Classe +	1	2
	Classe -	1	1

$$\text{Accuracy} = (CP+CN)/(CP+CN+FP+FN) = (1+1)/(1+2+1+1) = 2/5 = 0,4$$

=> 40%

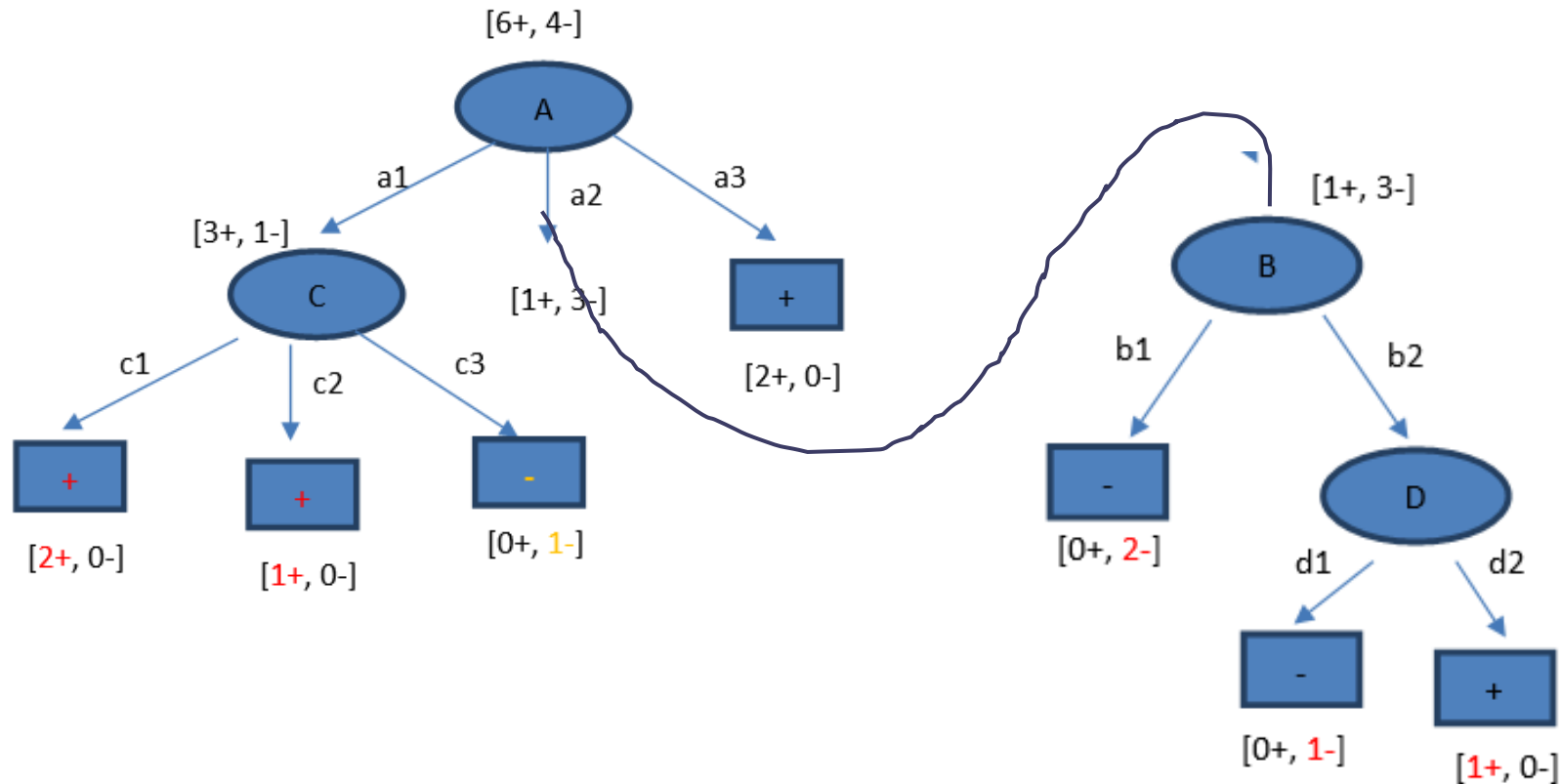
Série TD 2 - Exercice 1

1. Calculer l'entropie de l'ensemble S par rapport à la valeur de la classe.
2. Construire l'arbre de décision correspondant à l'ensemble S en utilisant l'algorithme ID3.
3. Etablir la matrice de confusion pour la table d'entraînement ci-dessus.
4. Donner la précision générale (accuracy) de l'arbre obtenu sur la table d'entraînement. Calculer la moyenne harmonique sur la même table.
5. Trouver la classe de l'exemple ayant les attributs (a2, b1, c3, d1) selon l'arbre construit.

Série TD 2 - Exercice 1

5. La classe de l'exemple $X = (a2, b1, c3, d1)$ selon l'arbre :

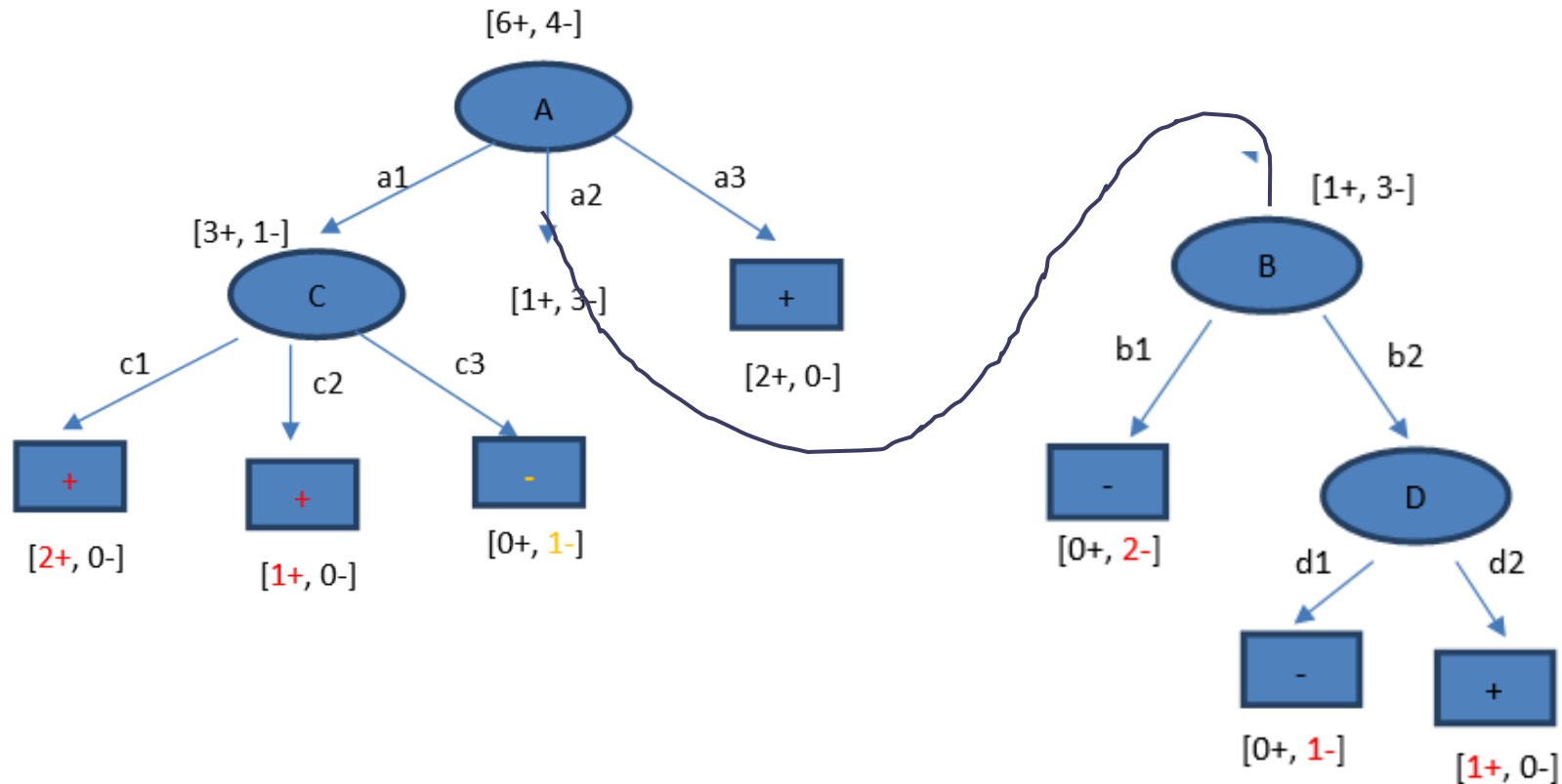
En parcourant l'arbre : $A \rightarrow a2 \rightarrow B \rightarrow b1 \rightarrow -$, donc X est de classe -



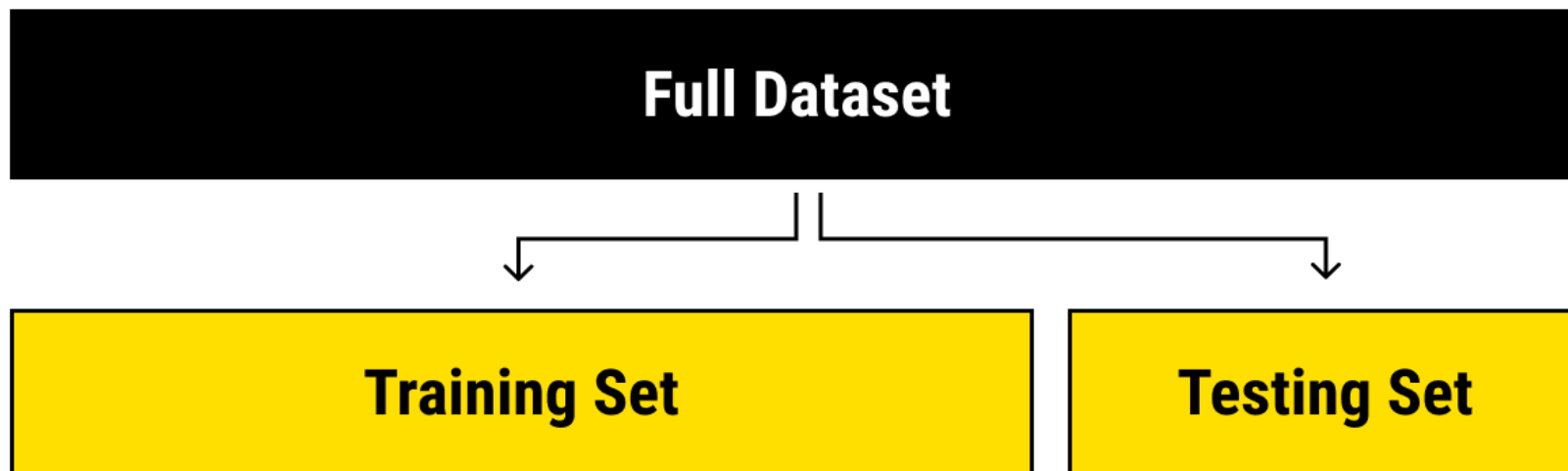
Série TD 2 - Exercice 1

5. La classe de l'exemple $X = (a1, b2, c2, d1)$ selon l'arbre :

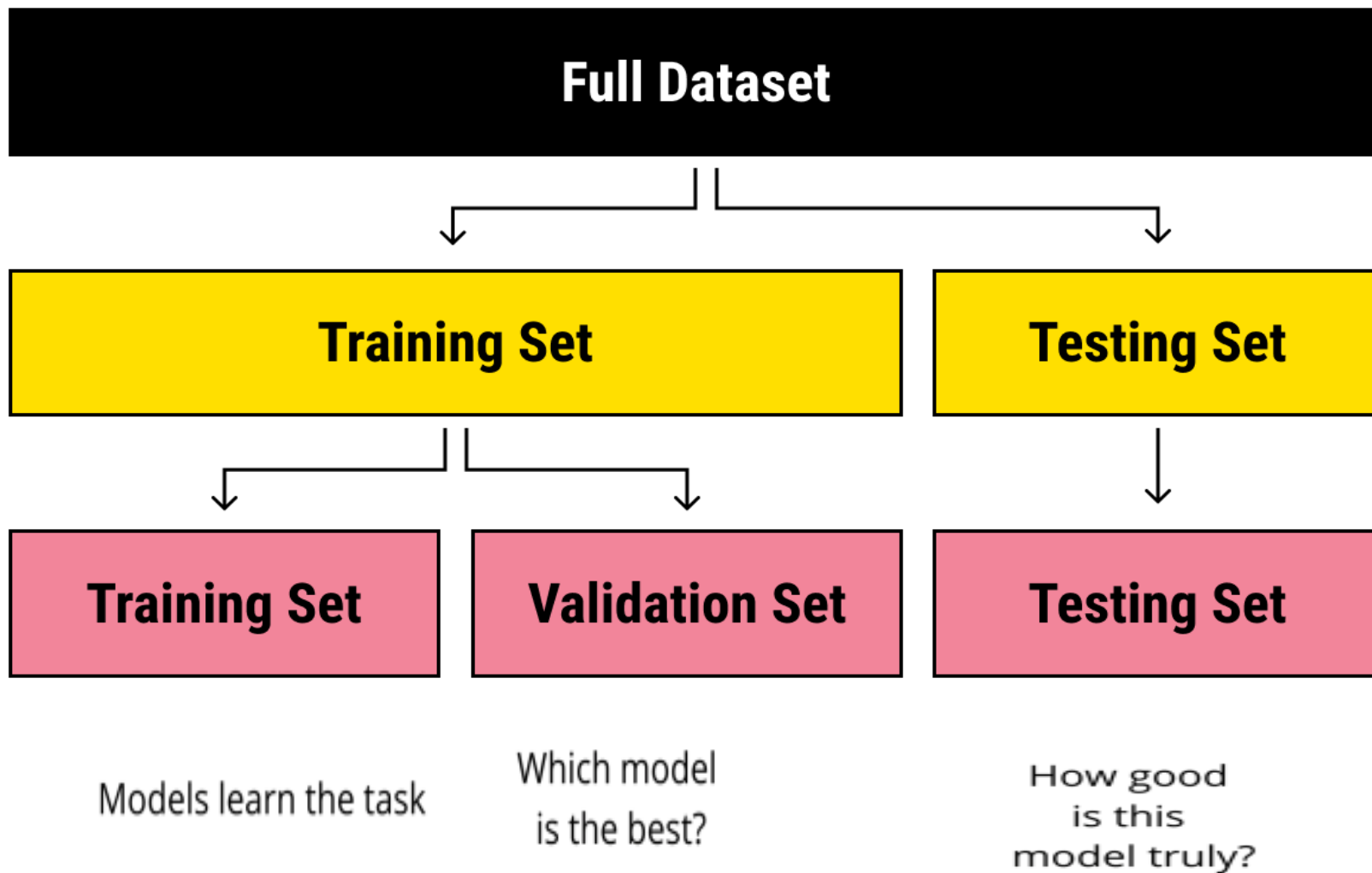
En parcourant l'arbre : $A \rightarrow a1 \rightarrow C \rightarrow c2 \rightarrow +$, donc X est de classe $+$



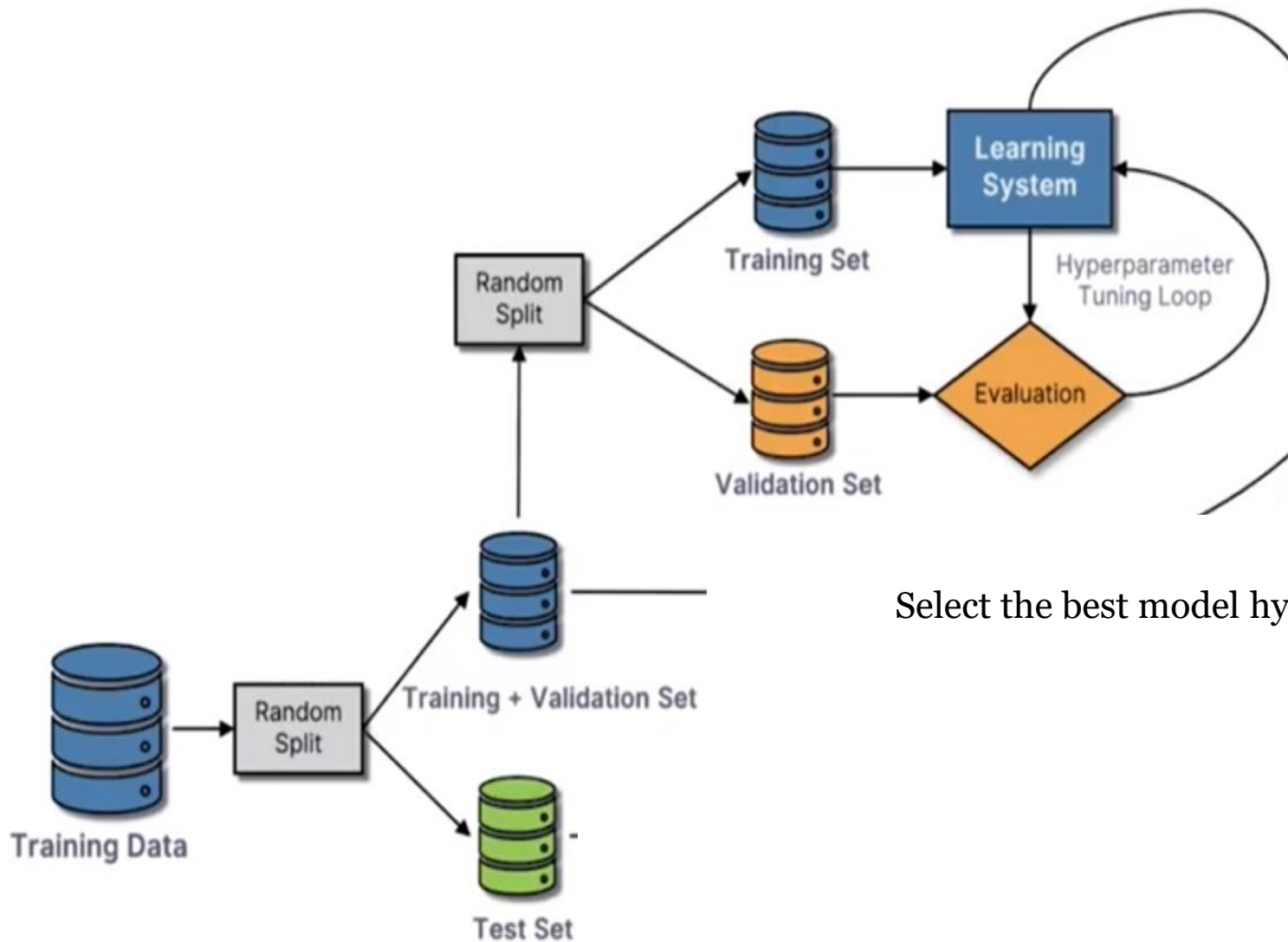
Méthodes de validation



Méthodes de validation

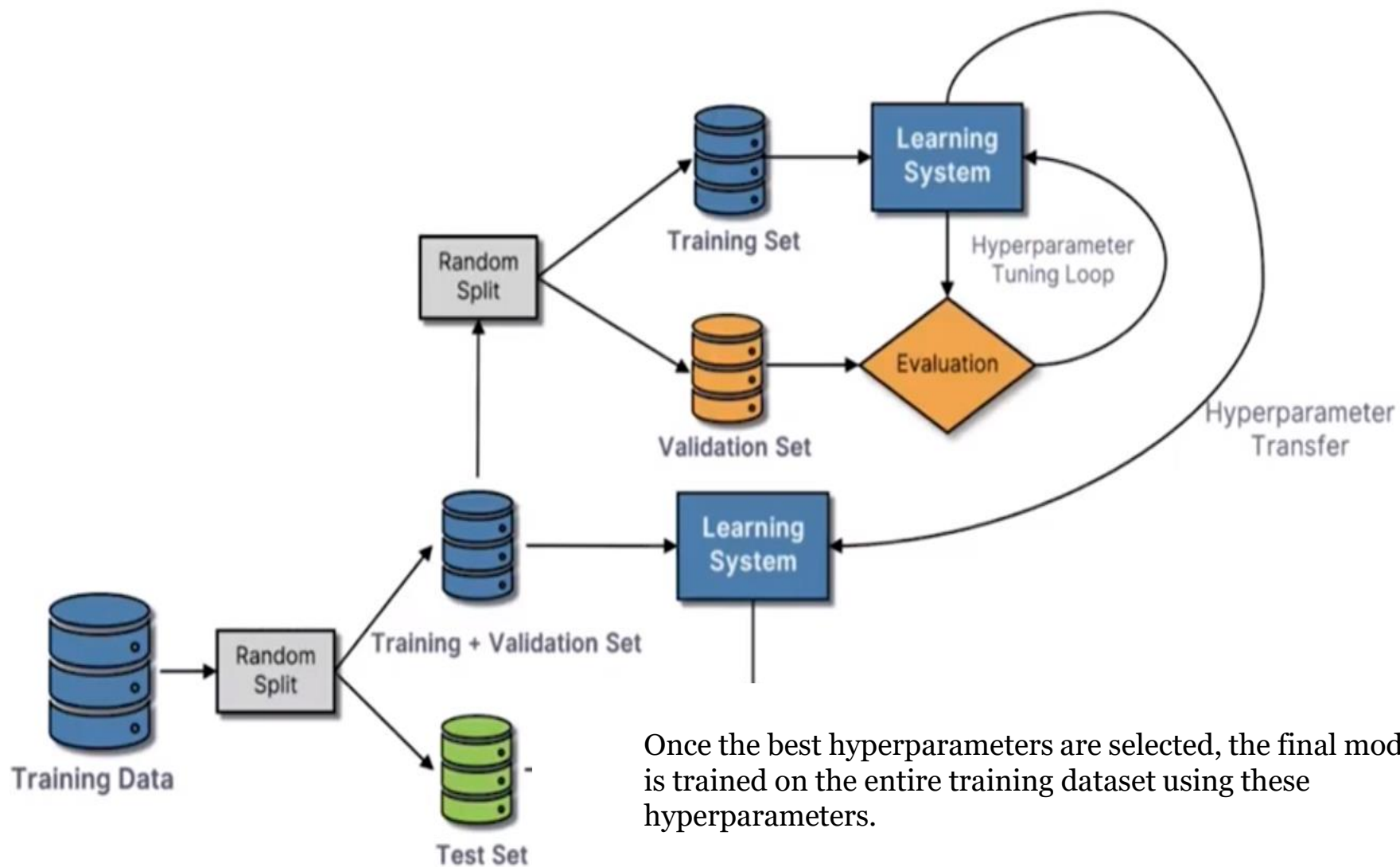


Méthodes de validation

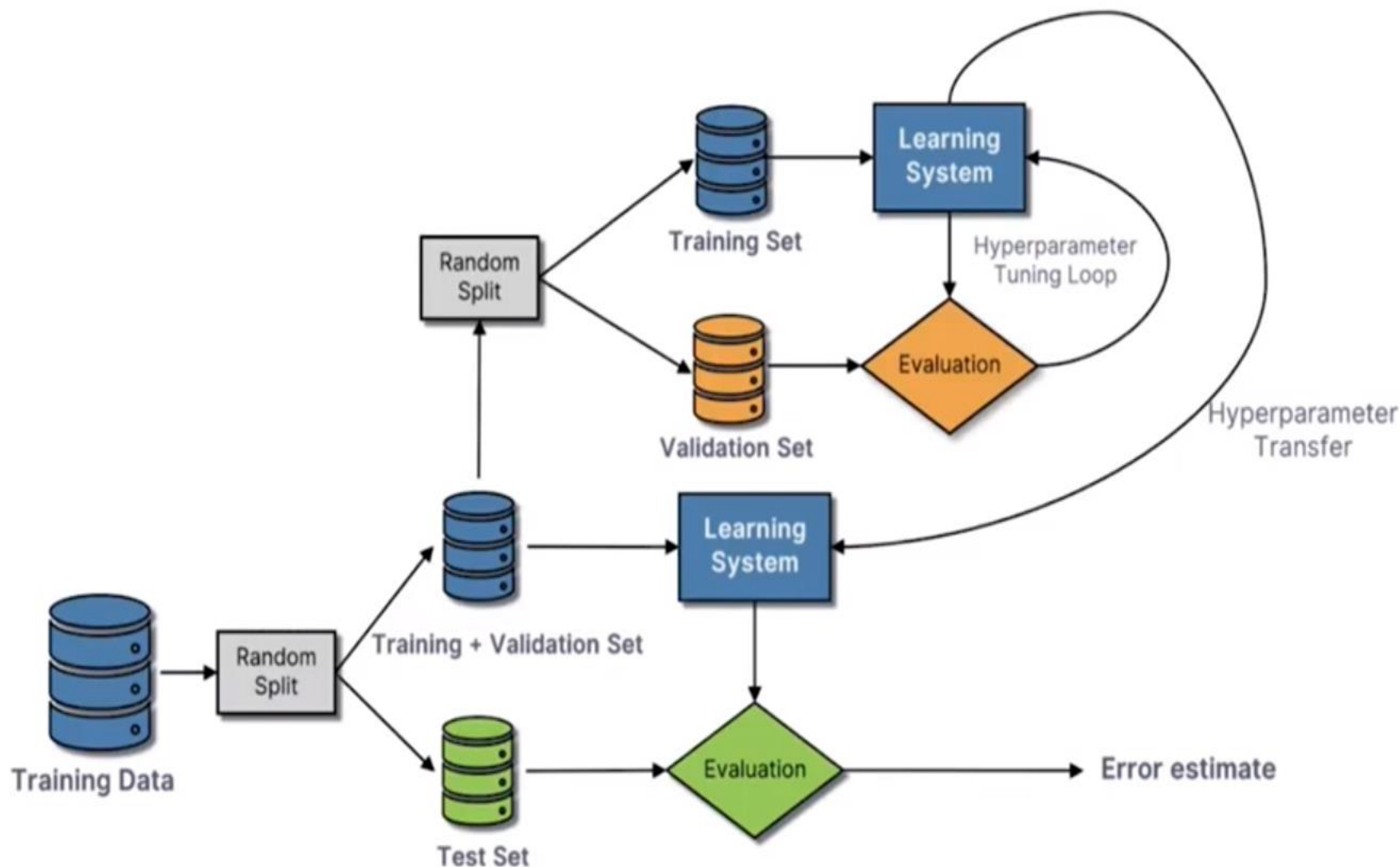


Select the best model hyperparameters

Méthodes de validation

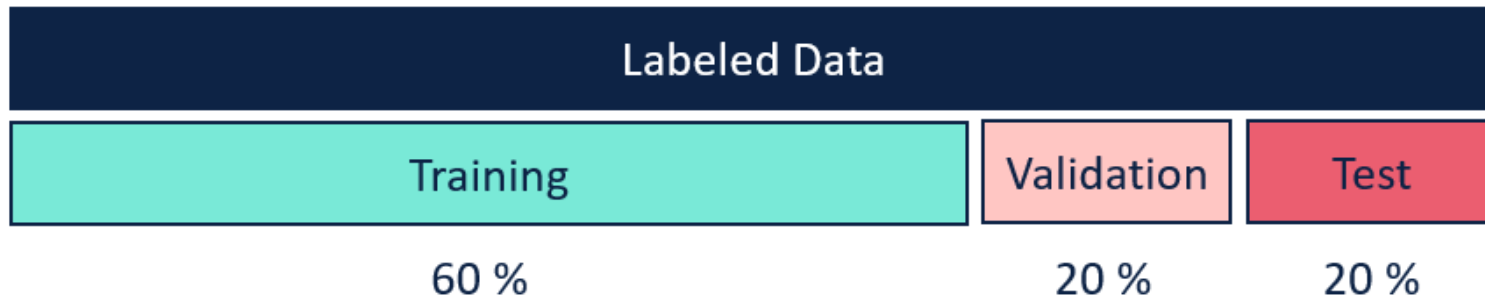
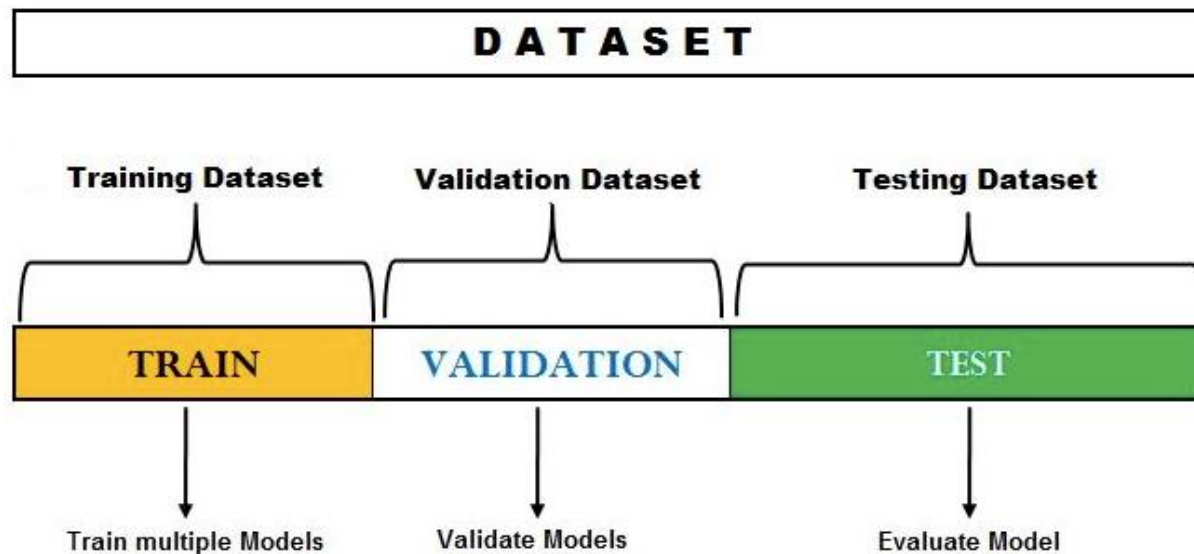


Méthodes de validation



Méthodes de validation

Méthode validation **HoldOut**

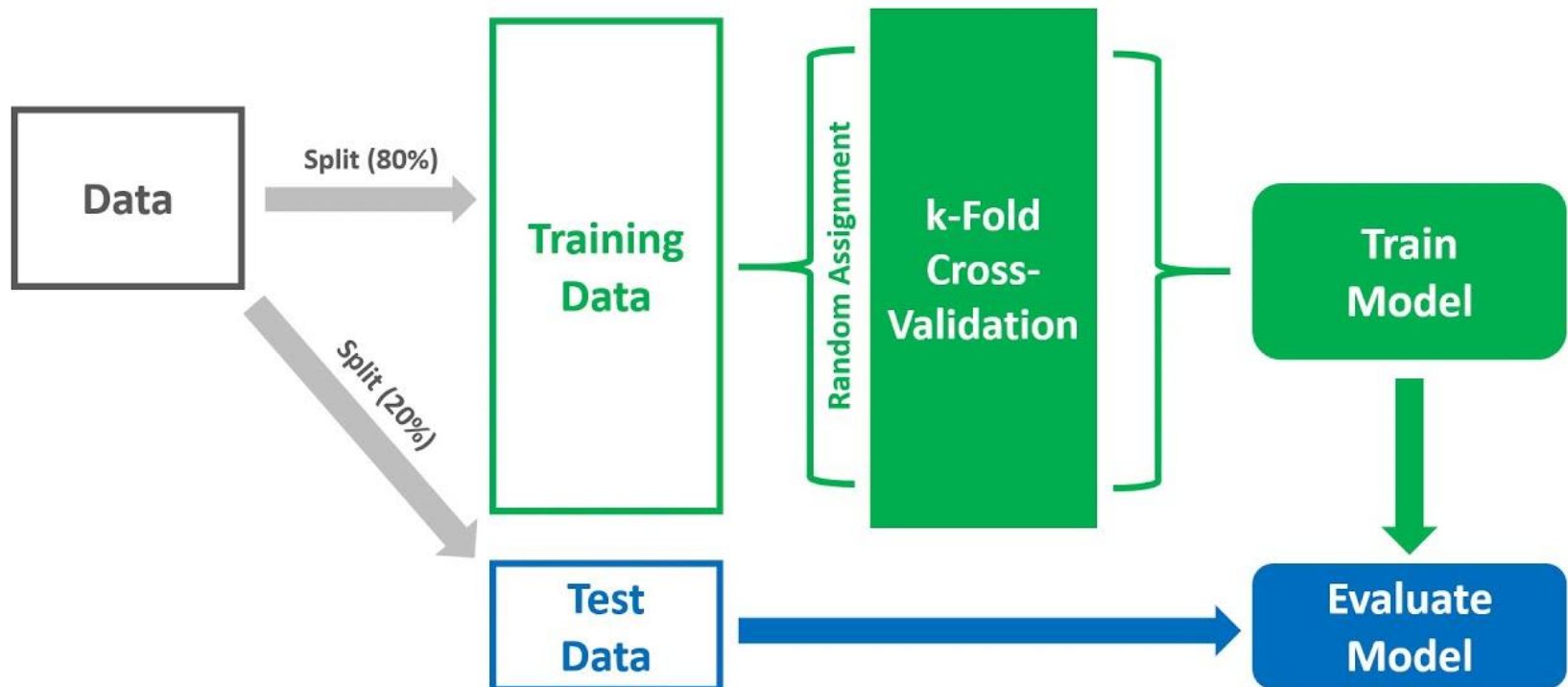


Méthodes de validation

Méthode validation croisée

Subdiviser D en k sous-ensembles de même taille – **K Folds**.

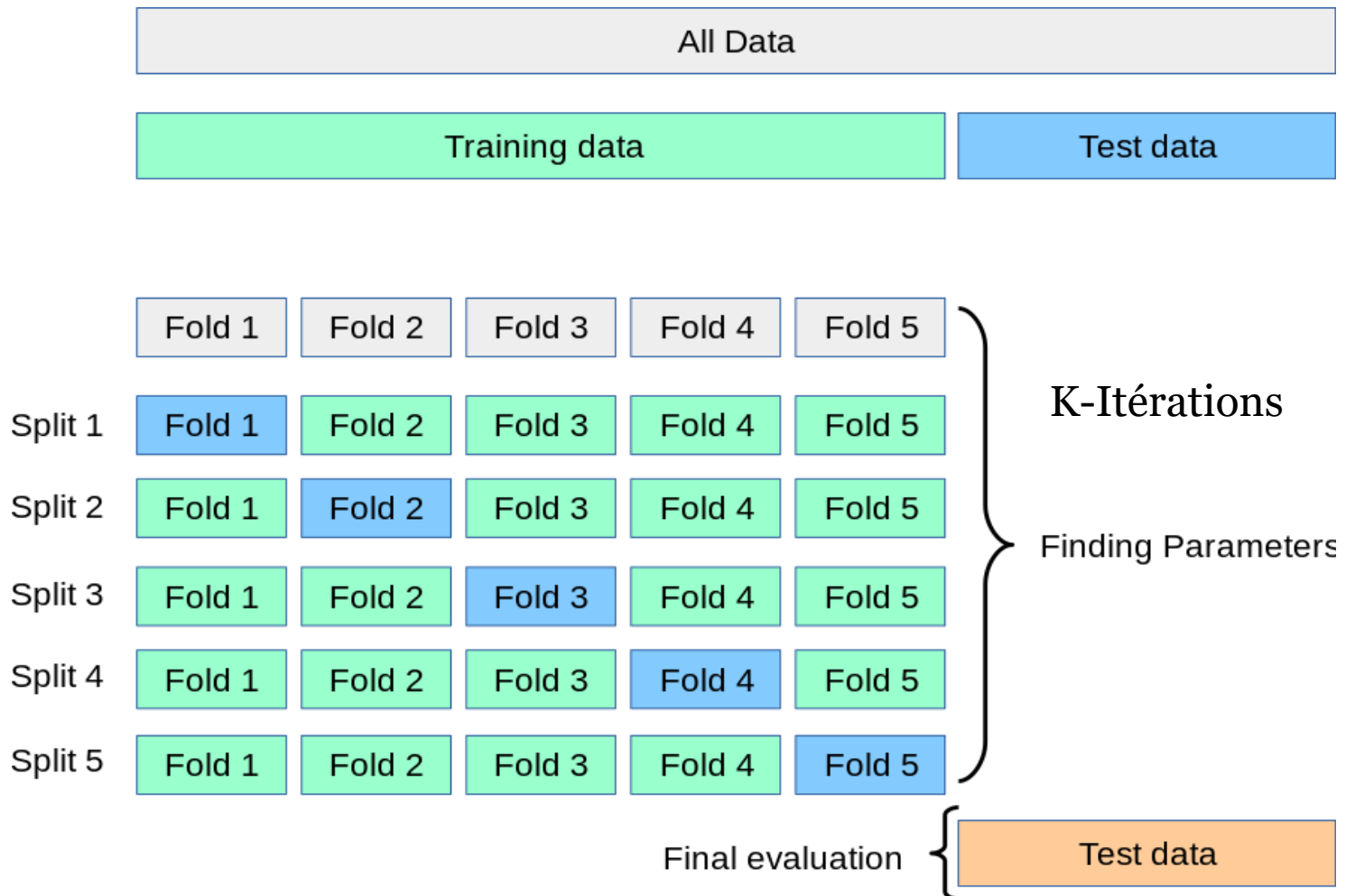
Example: k-Fold Cross-Validation



Méthodes de validation

Subdiviser D en k sous-ensembles de même taille – **K Folds**.

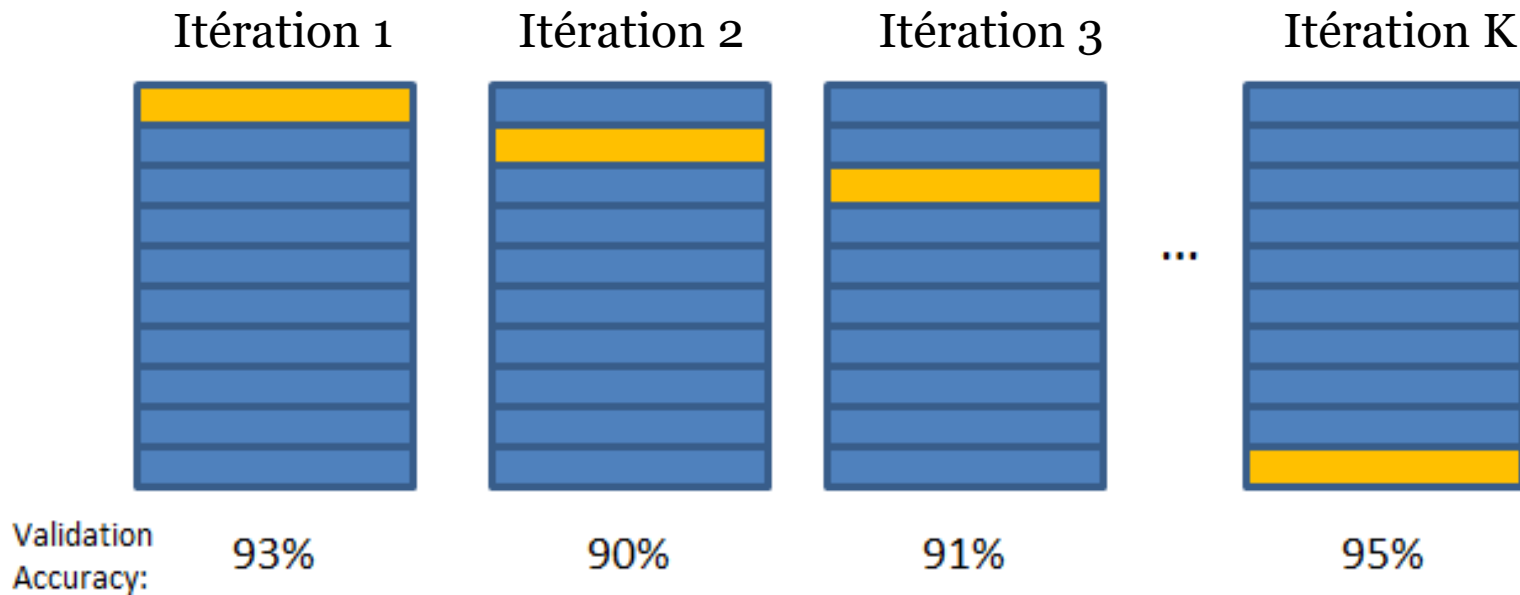
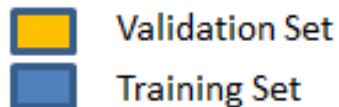
Méthode validation croisée



Méthodes de validation

Méthode validation croisée

Si $k=N$ (i.e. test sur un seul exemple exclu) =>
Méthode **Leave-One-Out**



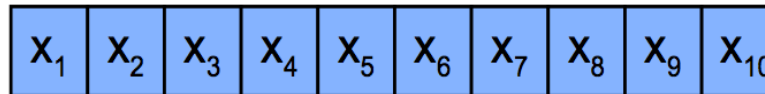
Accuracy Finale = Moyenne (Itération 1, Itération 2, ...)

Méthodes de validation

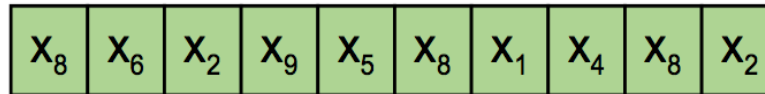
Subdiviser D en k sous-ensembles
aléatoires – Par remplacement.

Méthode Bootstrap

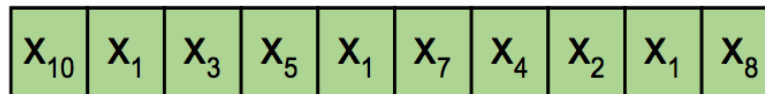
Original Dataset



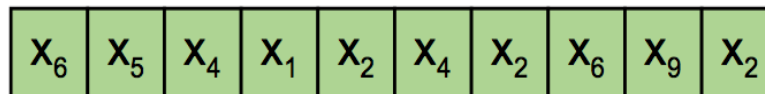
Bootstrap 1



Bootstrap 2

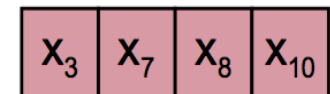
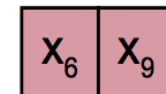
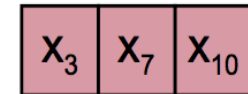


Bootstrap 3



Training Sets

Sampling with
Replacement

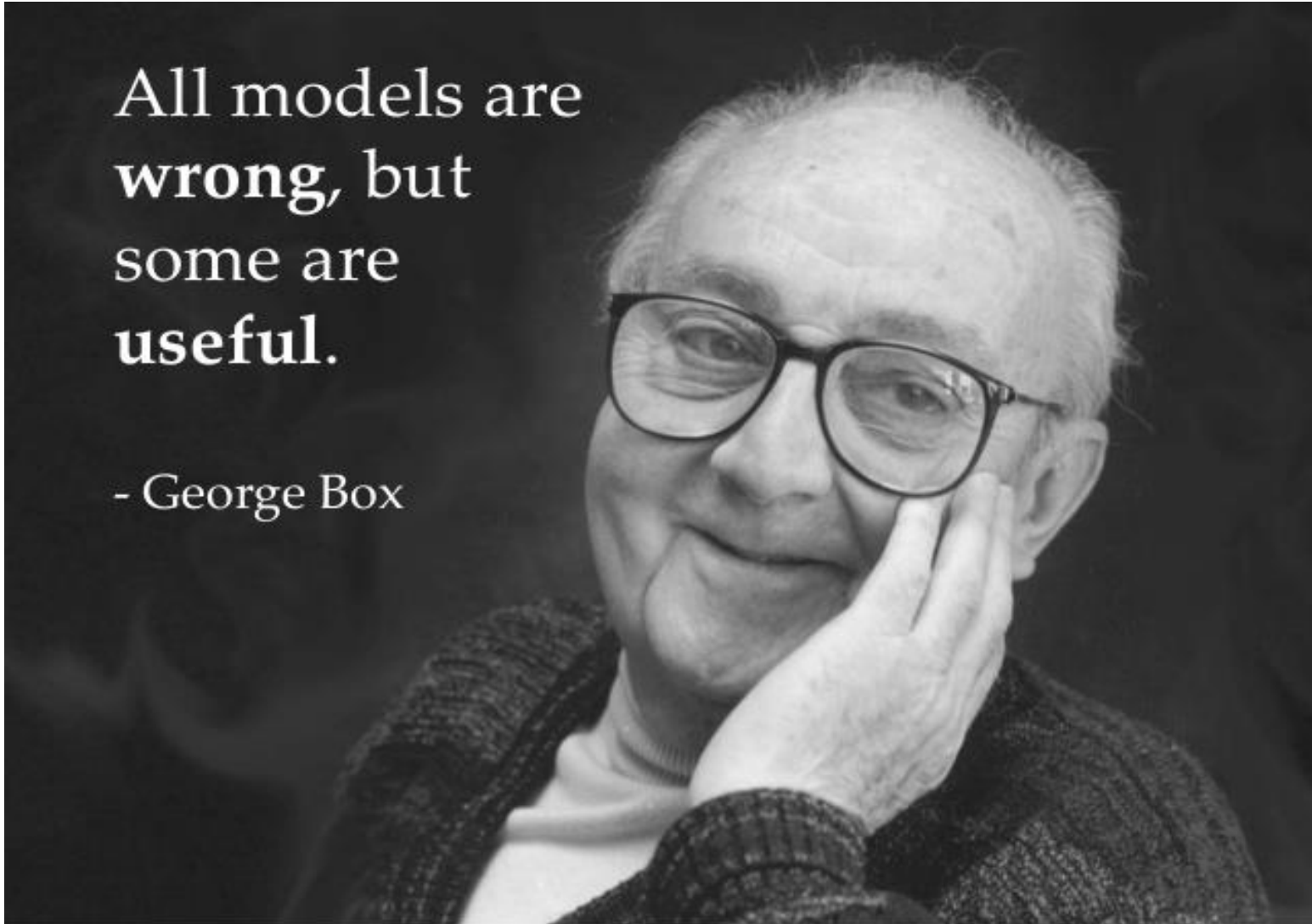


Validation Sets

Combinaison de modèles - Ensemble Learning

All models are
wrong, but
some are
useful.

- George Box



HOW TO CONFUSE MACHINE LEARNING



Combinaison de modèles

Labradoodle or fried chicken



Chihuahua or muffin



Sheepdog or mop

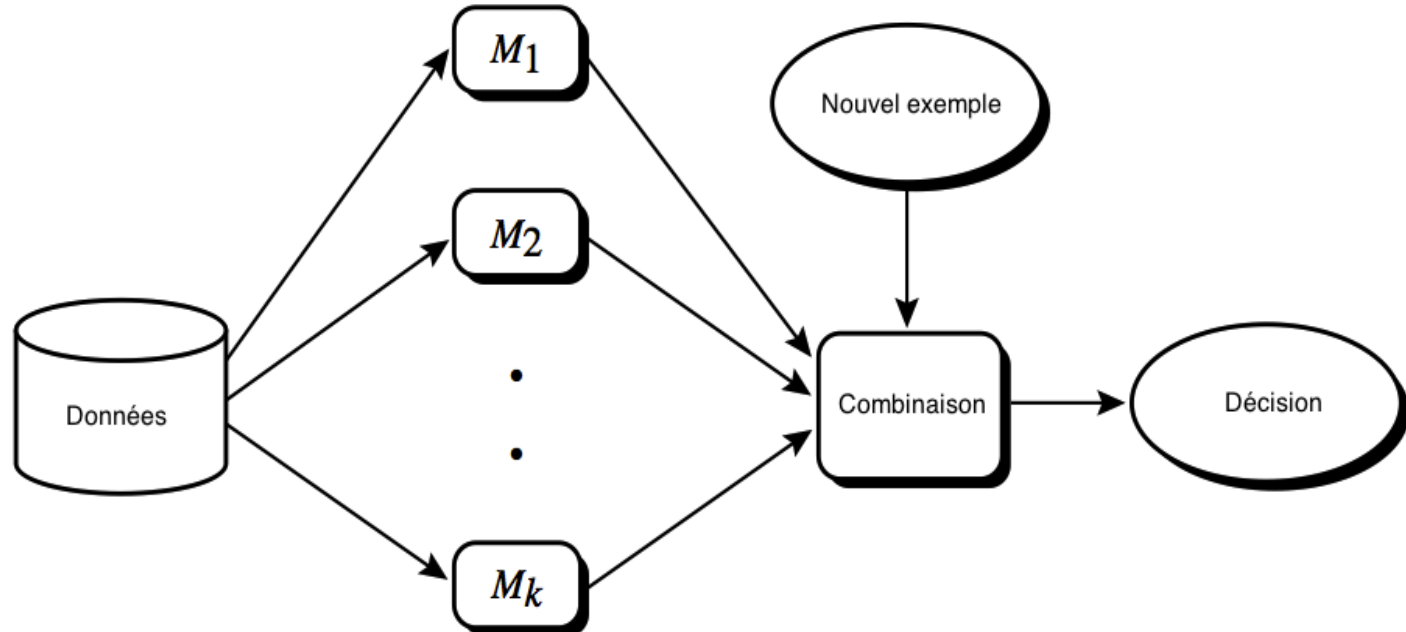


Barn owl or apple



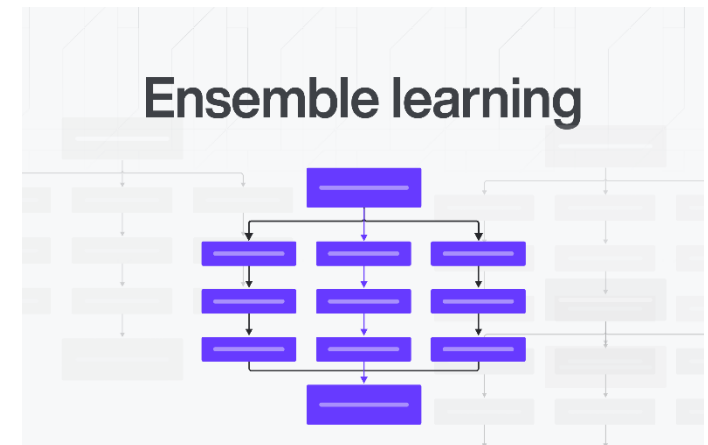
Combinaison de modèles - Ensemble Learning

- Combiner plusieurs modèles avec des performances faibles (weak learners) permettant d'obtenir un modèle plus efficace (meta-learner).
- Créer un grand nombre de petits modèles rapidement puis développer un modèle qui les rassemble.



Combinaison de modèles - Ensemble Learning

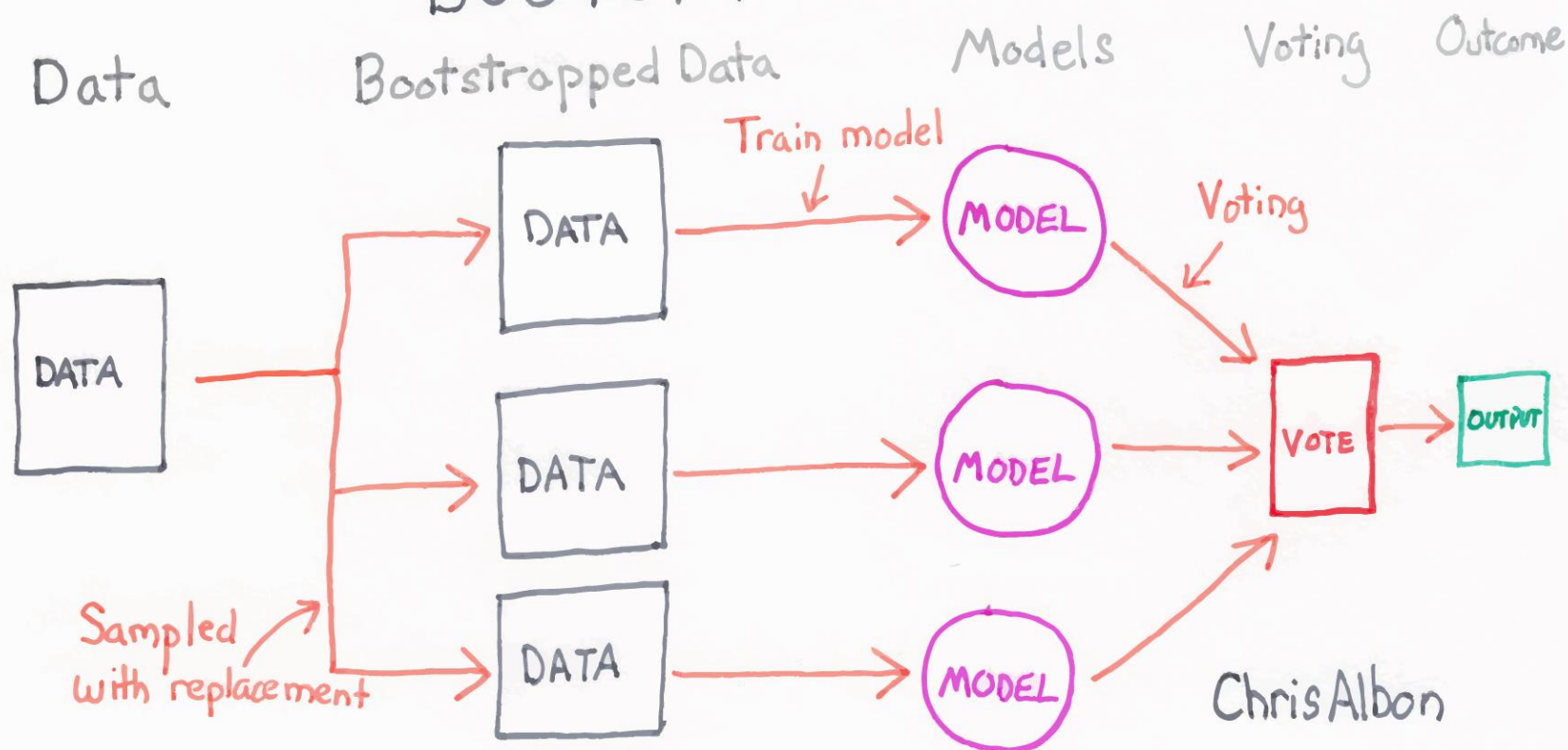
- Deux méthodes : Parallèles et séquentielles.
- Méthodes **parallèles** : Bagging et Random Forest.
- Méthodes **séquentielles** : Boosting (AdaBoost, Gradient Boosting, XGBoost etc.)
- **Stacking**, Stacked Generalization (Wolpert, 1992).



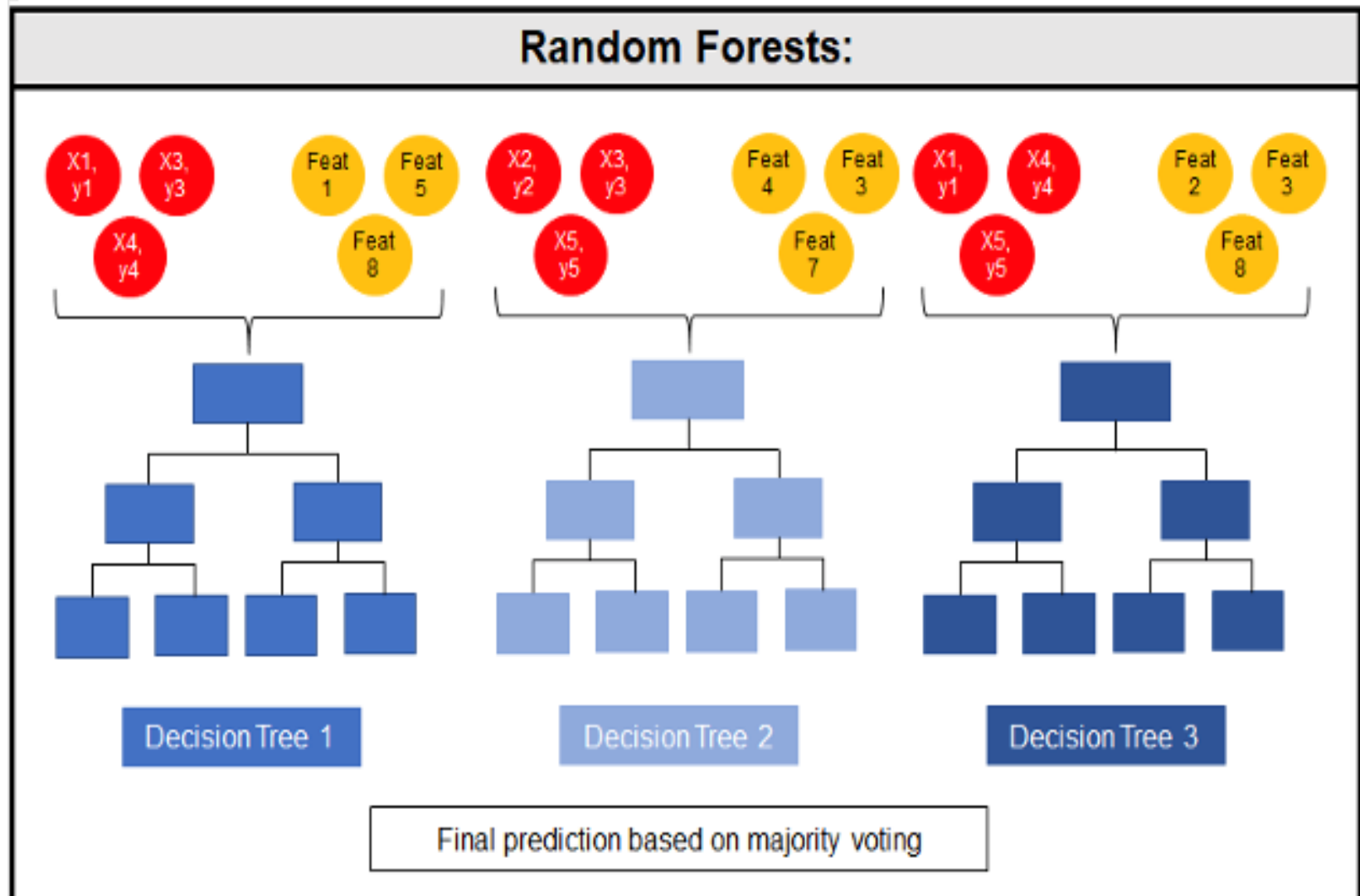
Combinaison de modèles - Ensemble Learning

BAGGING

BOOTSTRAP AGGREGATION

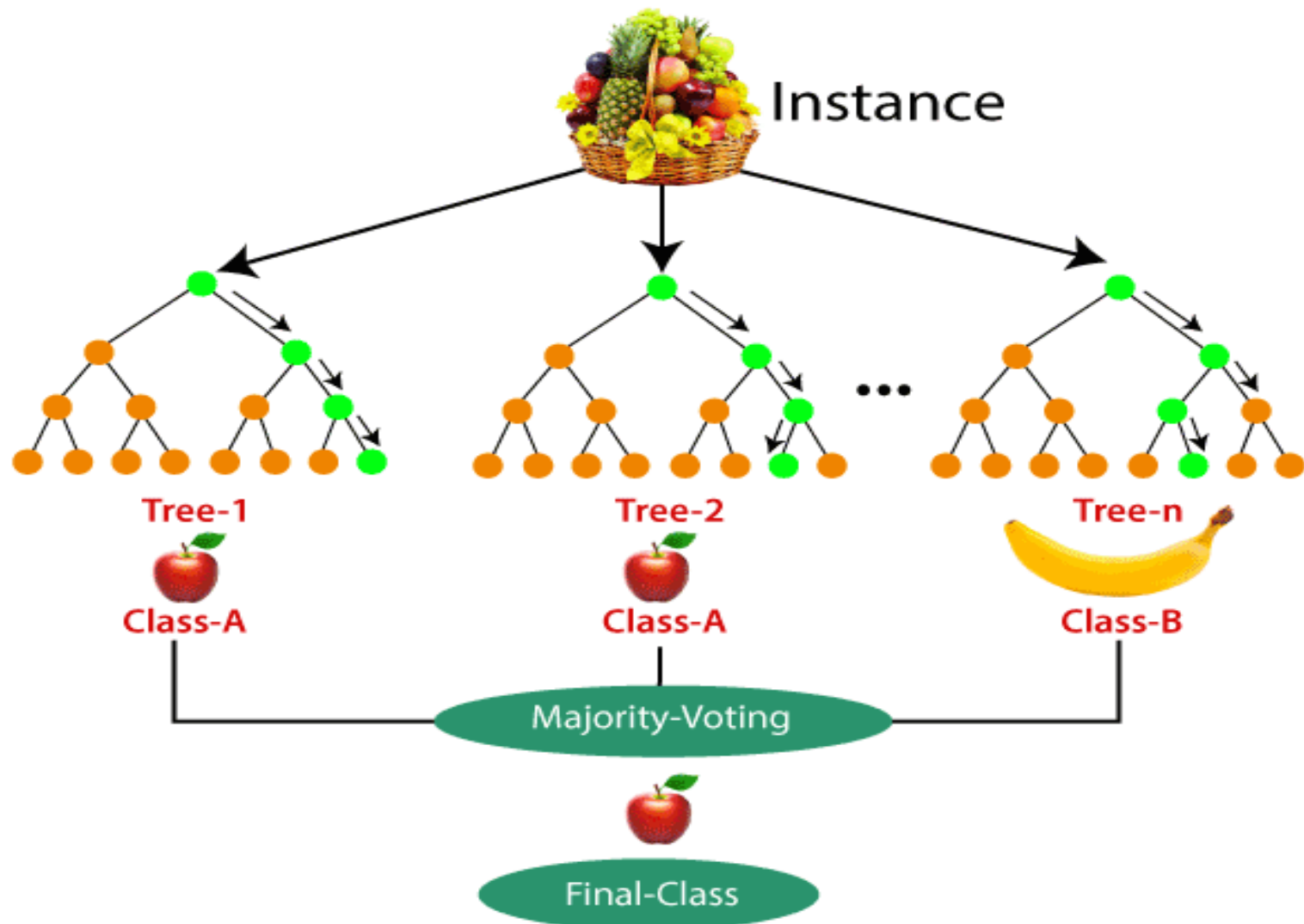


Combinaison de modèles - Ensemble Learning



Combinaison de modèles - Ensemble Learning

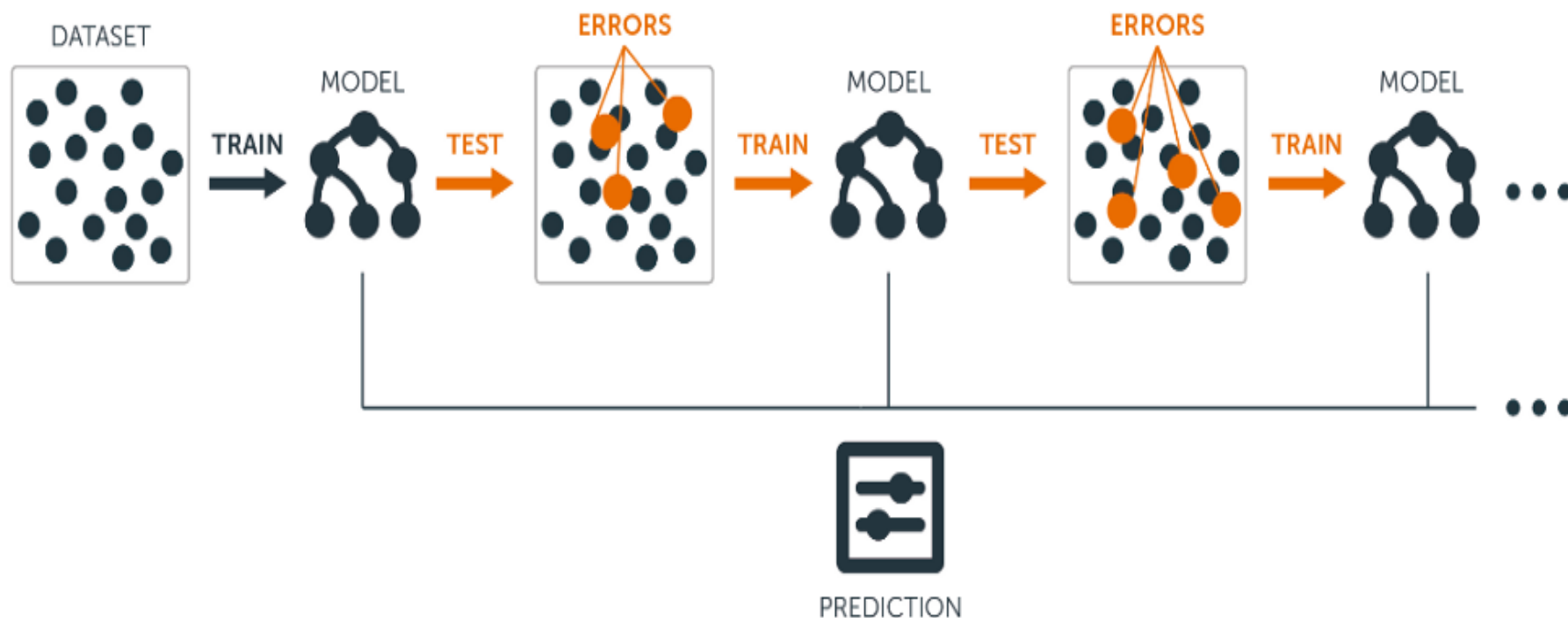
Random Forest Simplified



Combinaison de modèles - Ensemble Learning

Boosting

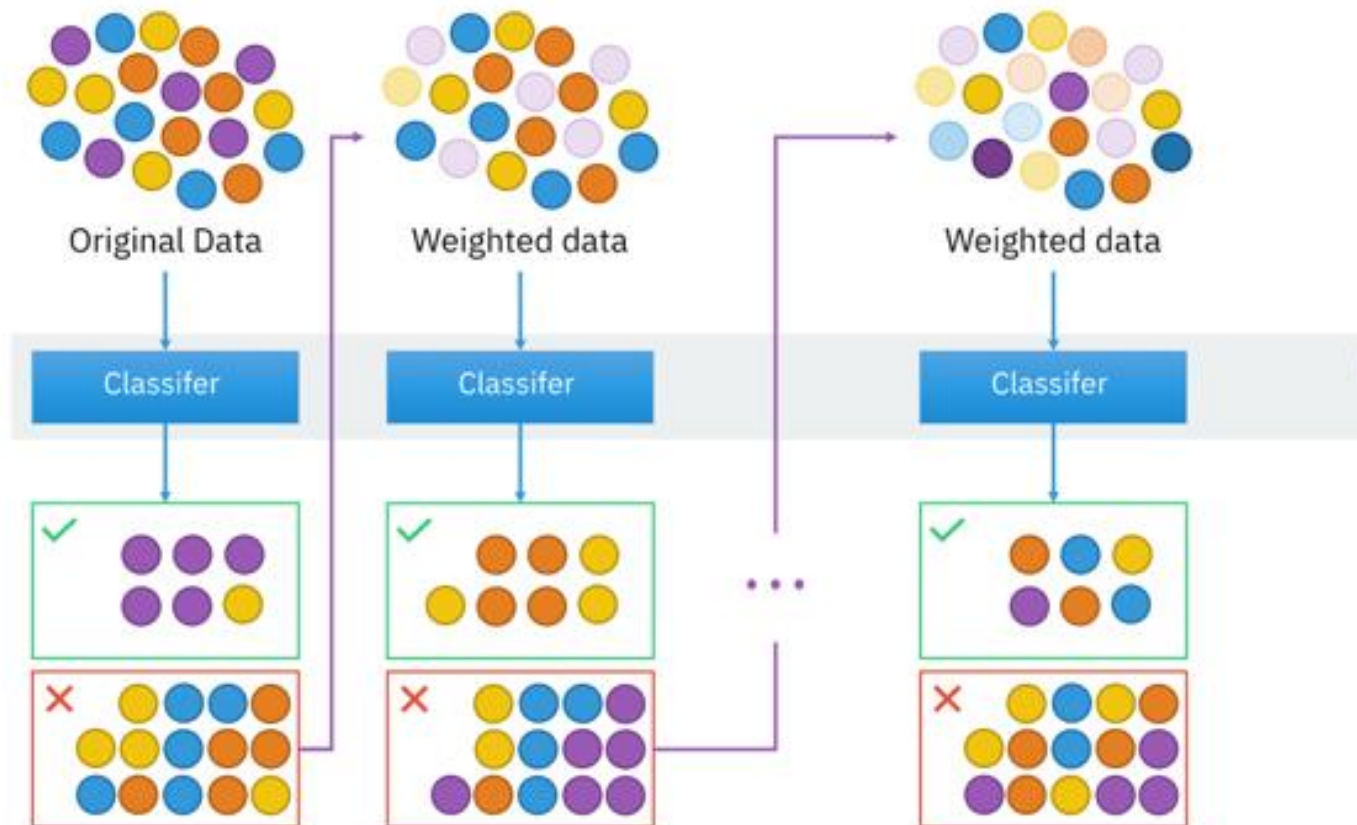
- Les exemples mal classés sont *boostés* (mettre à jour leur **poids**) pour qu'ils aient davantage d'importance vis-à-vis du classifieur faible au prochain tour, afin qu'il pallie le manque.



Combinaison de modèles - Ensemble Learning

Boosting

- Les exemples mal classés sont *boostés* (mettre à jour leur **poids**) pour qu'ils aient davantage d'importance vis-à-vis du classifieur faible au prochain tour, afin qu'il pallie le manque.

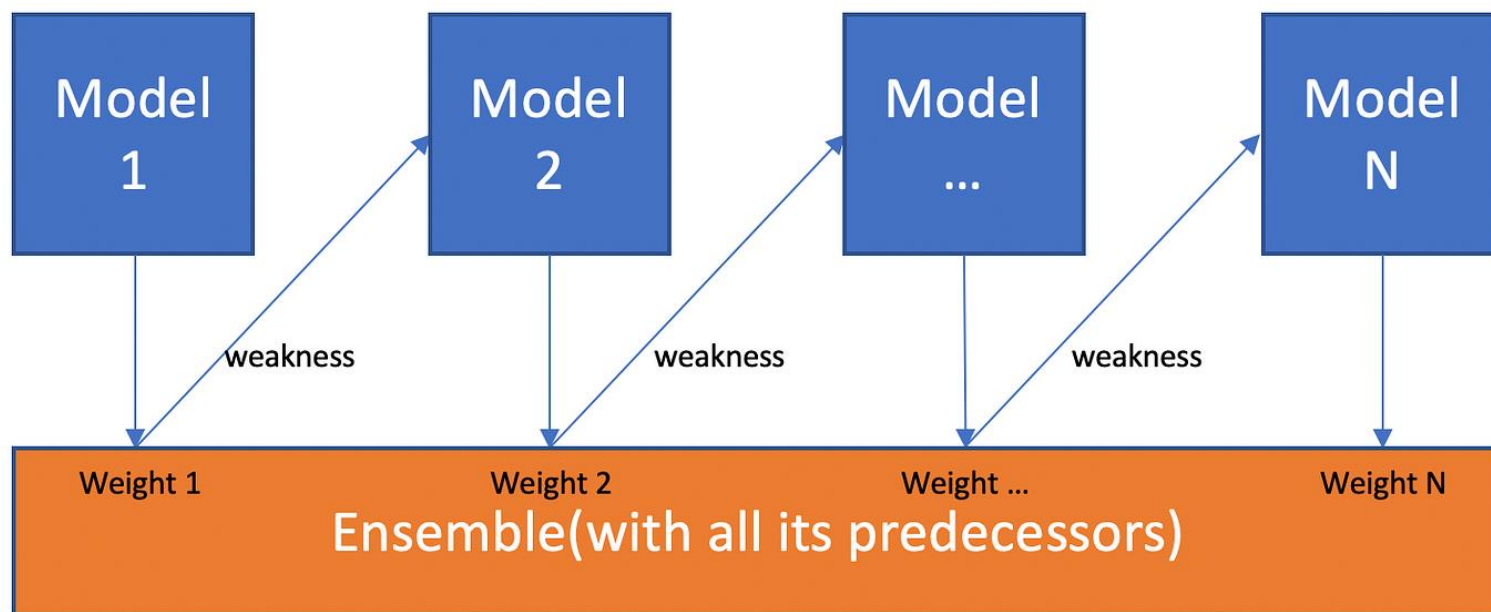


Combinaison de modèles - Ensemble Learning

Boosting

- Les exemples mal classés sont *boostés* (mettre à jour leur **poids**) pour qu'ils aient davantage d'importance vis-à-vis du classifieur faible au prochain tour, afin qu'il pallie le manque.

Model 1,2,..., N are individual models (e.g. decision tree)



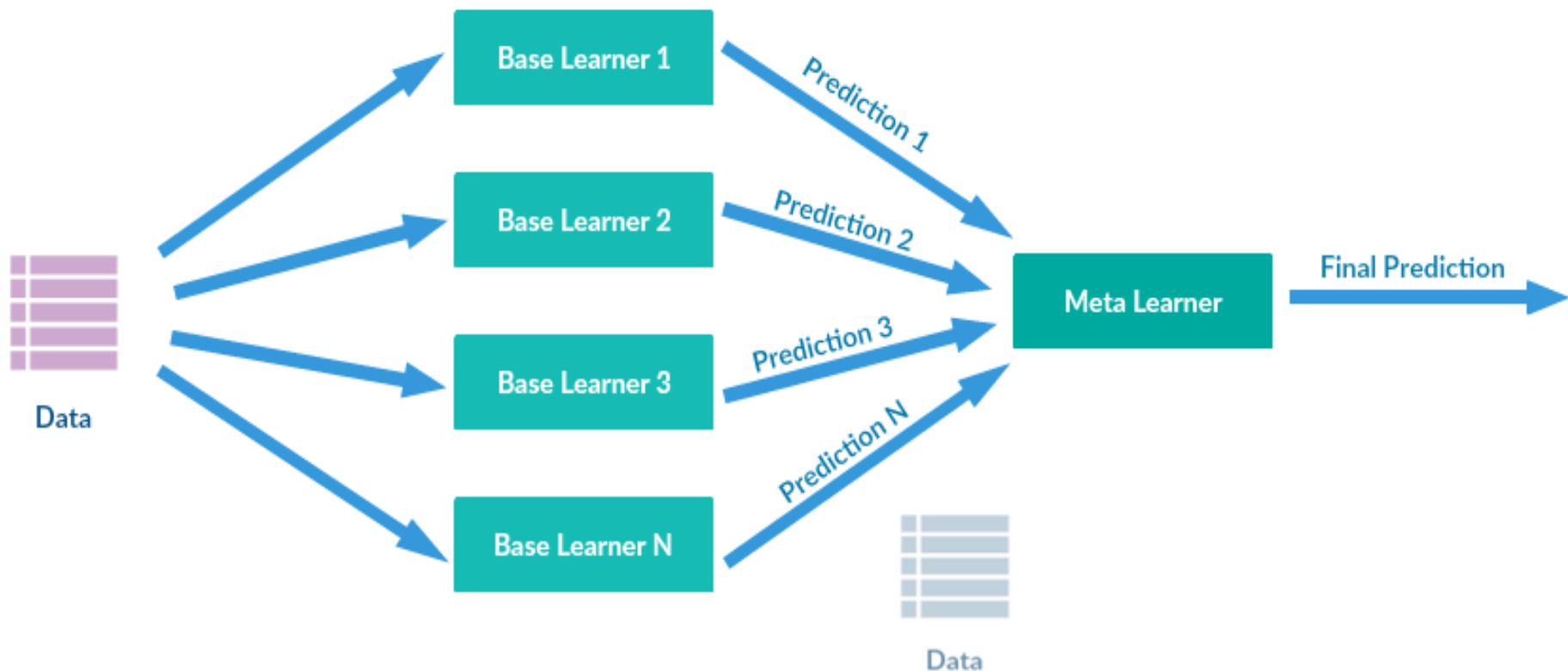
Combinaison de modèles - Ensemble Learning

Stacking/Blending

- Deux niveaux d'algorithmes. Principalement, algos de types différents.

level 0

level 1



Combinaison de modèles - Ensemble Learning

Stacking/Blending

- Deux niveaux d'algorithmes. Principalement, algos de types différents.

Level 1 training data

Data Point #	prediction from base learner 1	prediction from base learner 2	prediction from base learner 3	prediction from base learner M	actual
1	$y_{11}^{\hat{}}$	$y_{12}^{\hat{}}$	$y_{13}^{\hat{}}$	$y_{1M}^{\hat{}}$	y_1
2	$y_{21}^{\hat{}}$	$y_{22}^{\hat{}}$	$y_{23}^{\hat{}}$	$y_{2M}^{\hat{}}$	y_2
...
N	$y_{N1}^{\hat{}}$	$y_{N2}^{\hat{}}$	$y_{N3}^{\hat{}}$	$y_{NM}^{\hat{}}$	y_N

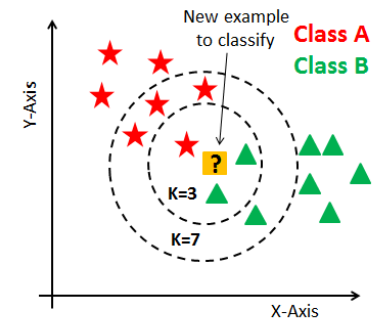
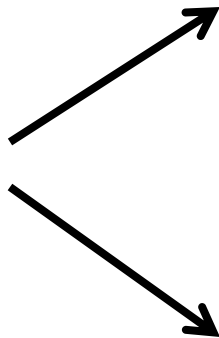


K plus proche voisins

SAVOIR - **PREDIRE** - DECIDER



Données

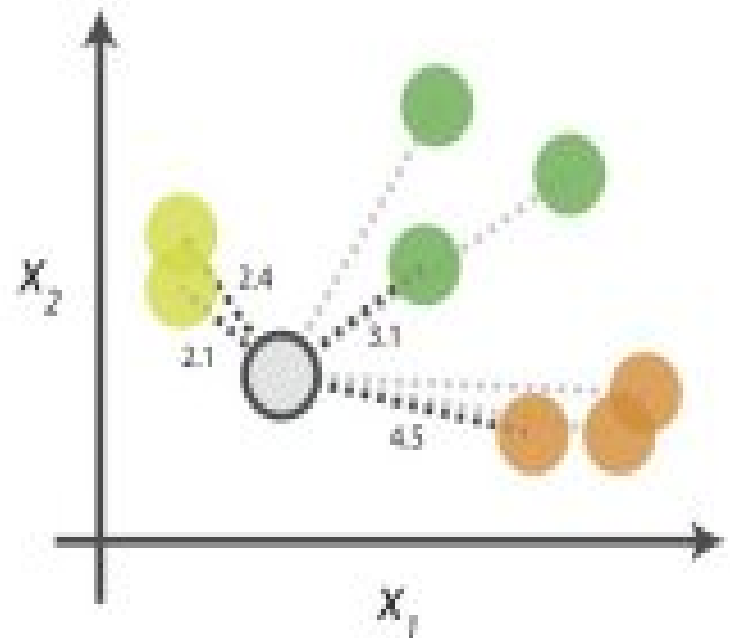


Connaissances

K-Nearest Neighbors

K plus proches voisins

- KNN – K Nearest Neighbors
- Algorithme de classification le plus simple.
- Principe : **Calcul la distance** entre tous les exemples de la base et le nouvel exemple qu'on cherche à classer.
- Choisir la classe majoritaire parmi les K-distances les plus petites.
- Les exemples sont représentés par des vecteurs de coordonnées.
- Distance euclidienne, Manhattan, Minkowski, Hamming, etc.



K plus proches voisins

Exemple : On pose **K = 3**

Scénario	Jeu d'acteurs	Classe
7	7	Good
7	4	Good
3	4	Bad
1	4	Bad

Test Data : Scénario = 3 , Jeu d'acteurs = 7, classe = ?

K plus proches voisins

Exemple : K = 3

Calculer la distance :

Euclidean

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Scénario	Jeu d'acteurs	Classe	Distance
7	7	Bon	$\text{sqrt}[(7-3)^2 + (7-7)^2] = 4$
7	4	Bon	$\text{sqrt}[(7-3)^2 + (4-7)^2] = 5$
3	4	Mauvais	3
1	4	Mauvais	3.60

Test Data : Scénario = 3 , Jeu d'acteurs = 7 , classe = ?

K plus proches voisins

Exemple : **K=3**

Les 3 plus proches exemples:

Euclidean

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

3

1

2

Scénario	Jeu d'acteurs	Classe	Distance
7	7	Bon	$\text{sqrt}[(7-3)^2+(7-7)^2]=4$
7	4	Bon	$\text{sqrt}[(7-3)^2+(4-7)^2]=5$
3	4	Mauvais	3
1	4	Mauvais	3.60

Test Data : Scénario = **3** , Jeu d'acteurs = **7**, classe = **?**

K plus proches voisins

Exemple : **K=3**

Euclidean

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Choix de la classe majoritaire

3

1

2

Scénario	Jeu d'acteurs	Classe	Distance
7	7	Bon	$\text{sqrt}[(7-\mathbf{3})^2 + (7-\mathbf{7})^2] = 4$
7	4	Bon	$\text{sqrt}[(7-\mathbf{3})^2 + (4-\mathbf{7})^2] = 5$
3	4	Mauvais	3
1	4	Mauvais	3.60

Test Data : Scénario = **3** , Jeu d'acteurs = **7** , classe = **Mauvais**

K plus proches voisins

Points clés

- **Choix de K** : Les performances de KNN dépendent fortement de la valeur de K (nombre de voisins). Un K faible peut rendre le modèle sensible au bruit, tandis qu'un K élevé peut atténuer des détails importants.
- **Métrique de distance** : KNN s'appuie sur une mesure de distance (comme la distance euclidienne, de Manhattan ou cosinus) pour trouver les voisins les plus proches, et le choix de la métrique peut influencer les résultats.
- KNN fonctionne mieux avec des attributs **numériques**, car il s'appuie sur des calculs de distance (comme la distance euclidienne ou de Manhattan).

K plus proches voisins

Points clés

- Il est facile à comprendre et efficace pour les petits ensembles de données.
- Cependant, il peut être **lent** et **gourmand** en **mémoire** sur les grands ensembles de données, car il doit comparer chaque nouveau point à tous les points existants.
- Apprentissage **paresseux** : KNN est un algorithme d'apprentissage paresseux : il n'entraîne **pas de modèle au préalable**, mais effectue des calculs pendant la phase de prédiction.
- Cela explique sa lenteur pour les grands ensembles de données.

K plus proches voisins

Points clés

- KNN ne fait **aucune hypothèse sur la distribution** des données et ne suppose pas que les données suivent un modèle spécifique.
- Ainsi, même si vos données sont désordonnées, non linéaires ou de forme irrégulière, KNN peut fonctionner car il s'adapte aux données telles qu'elles sont.
- KNN est un algorithme d'apprentissage automatique **non paramétrique**.
- Cela signifie qu'il ne suppose aucune forme ni équation fixe pour les données : il apprend directement à partir des exemples d'apprentissage au lieu d'apprendre des paramètres (comme les pondérations).

Série TD 3 - Exercice 2

1. Trouver le risque du client ayant les attributs (**Age=48 et Prêt=142,000**) en utilisant l'algorithme des **K plus proches voisins (KNN)**.

NB : On pose **K = 3** et distance = **distance euclidienne**.

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Age	Prêt	Non-remboursement
25	40,000	N
35	60,000	N
45	80,000	N
20	20,000	N
35	120,000	N
52	18,000	N
23	95,000	Y
40	62,000	Y
60	100,000	Y
48	220,000	Y
33	150,000	Y

Série TD 3 - Exercice 2

(Age = 48 et Prêt = 142,000)

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Age	Prêt	Non-remboursement	Distance
25	40,000	N	
35	60,000	N	
45	80,000	N	
20	20,000	N	
35	120,000	N	
52	18,000	N	
23	95,000	Y	
40	62,000	Y	
60	100,000	Y	
48	220,000	Y	
33	150,000	Y	

Série TD 3 - Exercice 2

(Age = 48 et Prêt = 142,000)

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Age	Prêt	Non-remboursement	Distance
25	40,000	N	$\text{sqrt}((25-48)^2 + (40000-142000)^2) = 102\ 000$
35	60,000	N	
45	80,000	N	
20	20,000	N	
35	120,000	N	
52	18,000	N	
23	95,000	Y	
40	62,000	Y	
60	100,000	Y	
48	220,000	Y	
33	150,000	Y	

Série TD 3 - Exercice 2

(Age = 48 et Prêt = 142,000)

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Age	Prêt	Non-remboursement	Distance
25	40,000	N	$\text{sqrt}((25-48)^2 + (40000-142000)^2) = 102\ 000$
35	60,000	N	82 000
45	80,000	N	62 000
20	20,000	N	122 000
35	120,000	N	22 000
52	18,000	N	124 000
23	95,000	Y	47 000
40	62,000	Y	80 000
60	100,000	Y	42 000
48	220,000	Y	78 000
33	150,000	Y	8000

Série TD 3 - Exercice 2

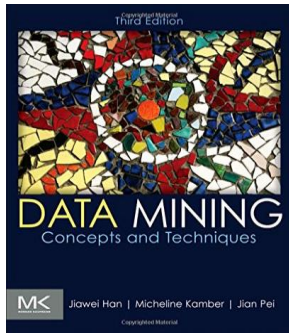
K=3

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Age	Prêt	Non-remboursement	Distance
25	40,000	N	$\text{sqrt}((25-48)^2 + (40000-142000)^2) = 102\ 000$
35	60,000	N	82 000
45	80,000	N	62 000
20	20,000	N	122 000
35	120,000	N	22 000
52	18,000	N	124 000
23	95,000	Y	47 000
40	62,000	Y	80 000
60	100,000	Y	42 000
48	220,000	Y	78 000
33	150,000	Y	8000

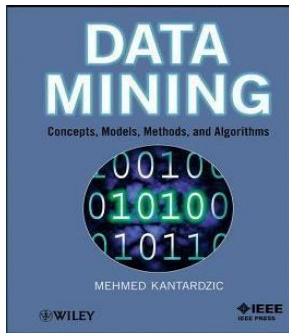
Y est majoritaire → Le risque de X est Y

Ressources



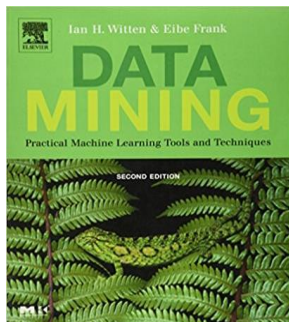
Data Mining : concepts and techniques, 3rd Edition

- ✓ Auteur : Jiawei Han, Micheline Kamber, Jian Pei
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2011 - 744 pages - ISBN 9780123814807



Data Mining : concepts, models, methods, and algorithms

- ✓ Auteur : Mehmed Kantardzi
- ✓ Éditeur : John Wiley & Sons
- ✓ Edition : Aout 2011 – 552 pages - ISBN : 9781118029121



Data Mining: Practical Machine Learning Tools and Techniques

- ✓ Auteur : Ian H. Witten & Eibe Frank
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2005 - 664 pages - ISBN : 0-12-088407-0

Ressources

Cours – Abdelhamid DJEFFAL – Fouille de données avancée

✓ www.abdelhamid-djeffal.net

WekaMOOC – Ian Witten – Data Mining with Weka

✓ <https://www.youtube.com/user/WekaMOOC/featured>

Cours - Laboratoire ERIC Lyon - DATA MINING et DATA SCIENCE

✓ https://eric.univ-lyon2.fr/~ricco/cours/supports_data_mining.html

Gregory Piatetsky-Shapiro - KDNuggets

✓ <http://www.kdnuggets.com/>