Fouille de Données

Data Mining

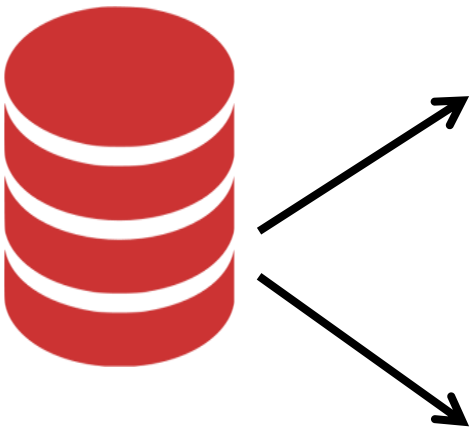
Classification - Partie 2

Plan du cours

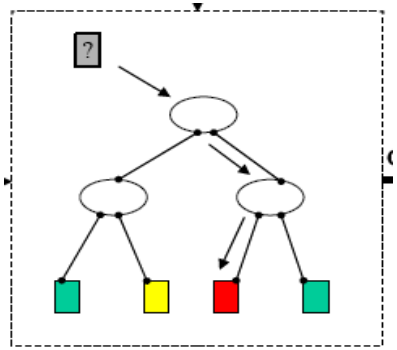
1. Classification associative
2. Méthodes d'évaluation d'un modèle
3. Combinaison de modèles
4. K plus proches voisins

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.

Choix de la bonne taille de l'arbre

Pré-élagage

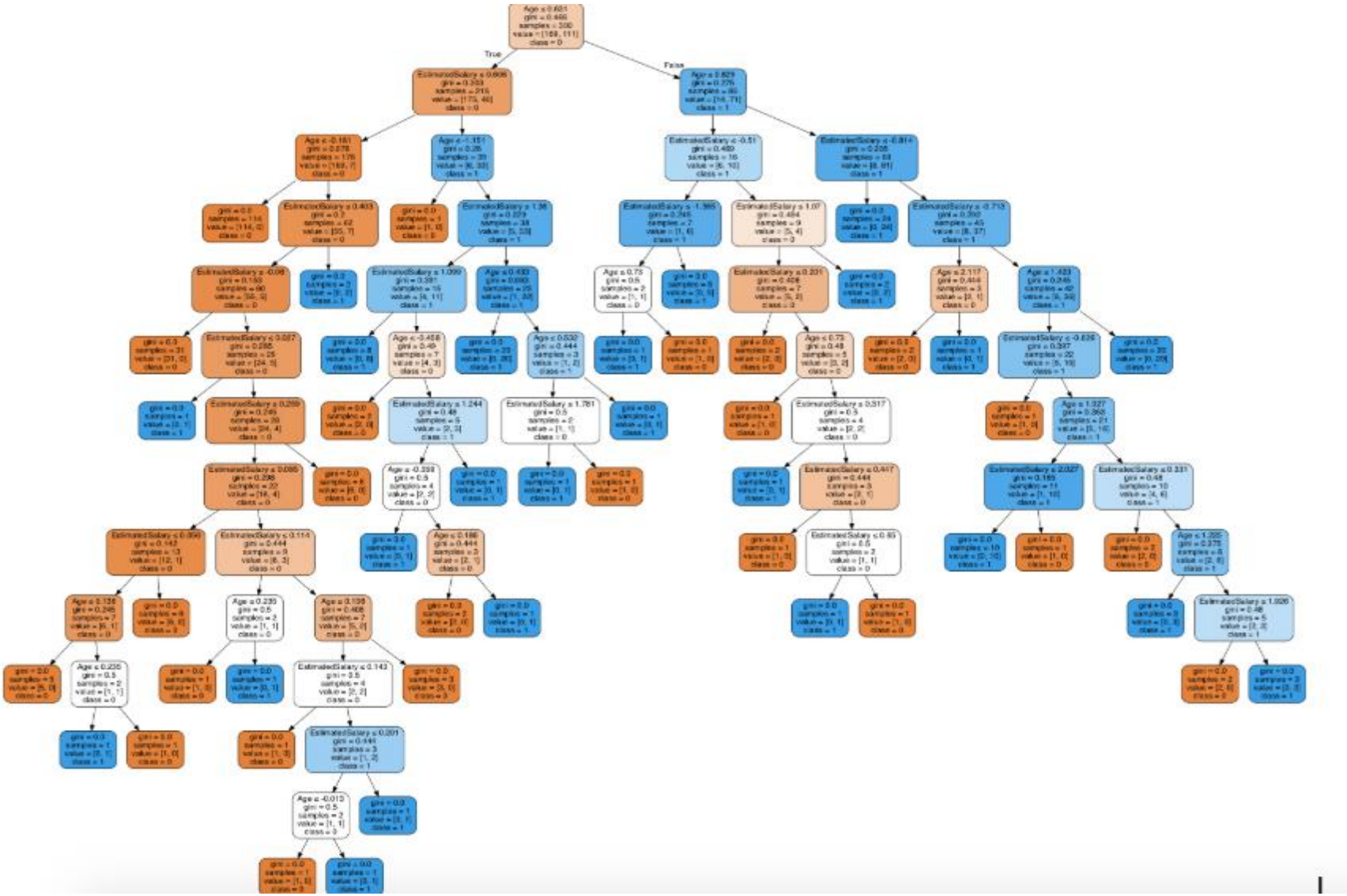
- Effectué **lors** de la construction de l'arbre.
- Au moment du calcul du gain d'information, décider de l'importance ou non de sa subdivision.
- Arrêter la construction lorsqu'il n'y a pas d'association statistiquement significative entre un attribut et la classe d'un nœud particulier.
- Couper complètement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

Choix de la bonne taille de l'arbre

Post-élagage

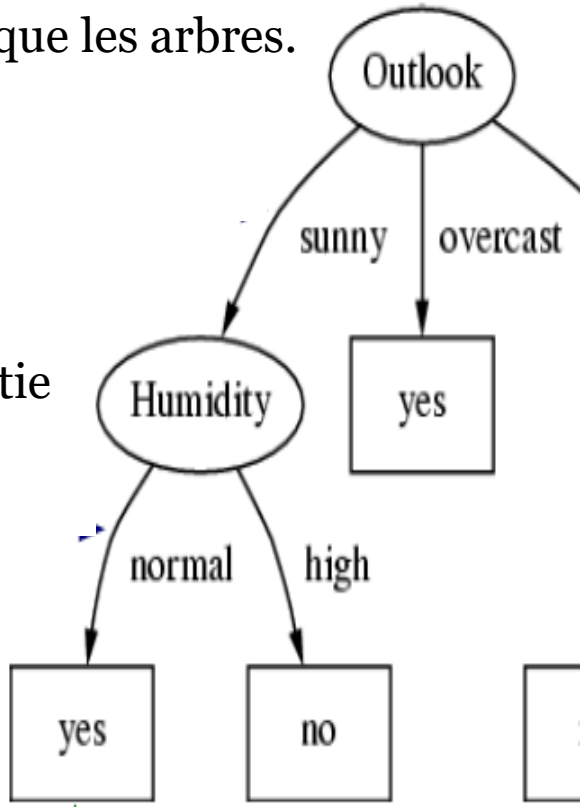
- Effectué **après** la construction de l'arbre en coupant des sous arbres entiers et en les remplaçant par des feuilles représentant la classe la plus fréquente dans l'ensemble des données de cet arbre.
- On commence de la racine et on descend.
- Pour chaque nœud interne (non feuille), on mesure sa 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.



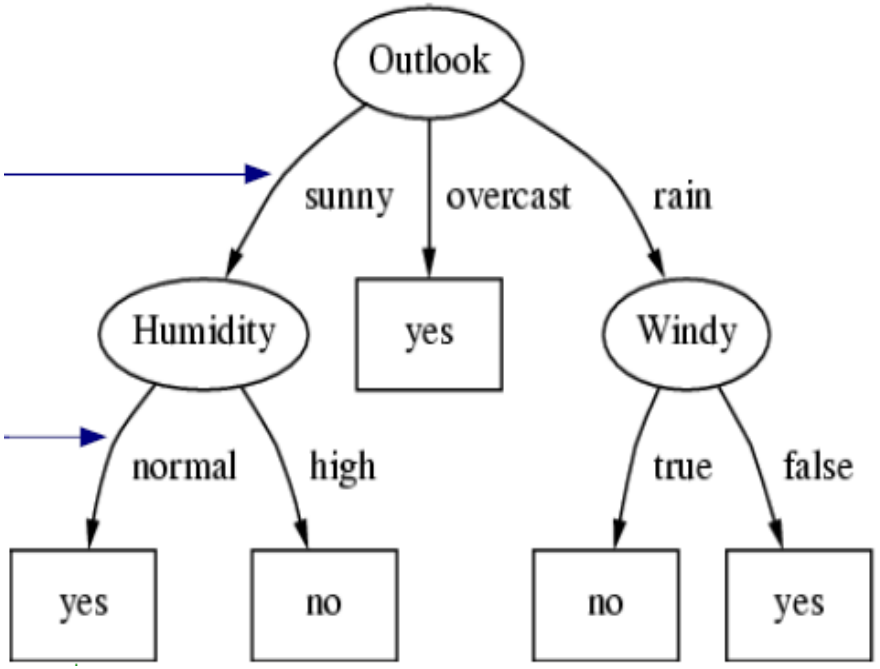
- Exemple :

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

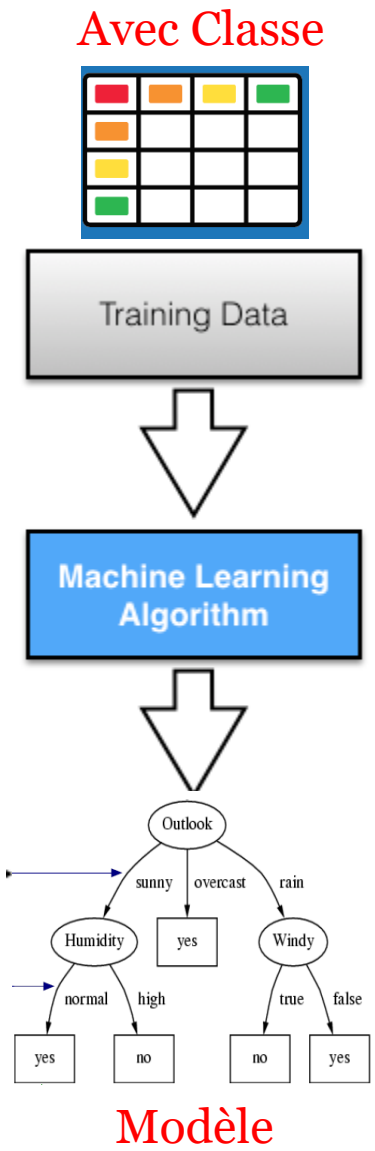
Outlook	Temp	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

OTHERWISE play=yes

Evaluation du modèle

Deux étapes :

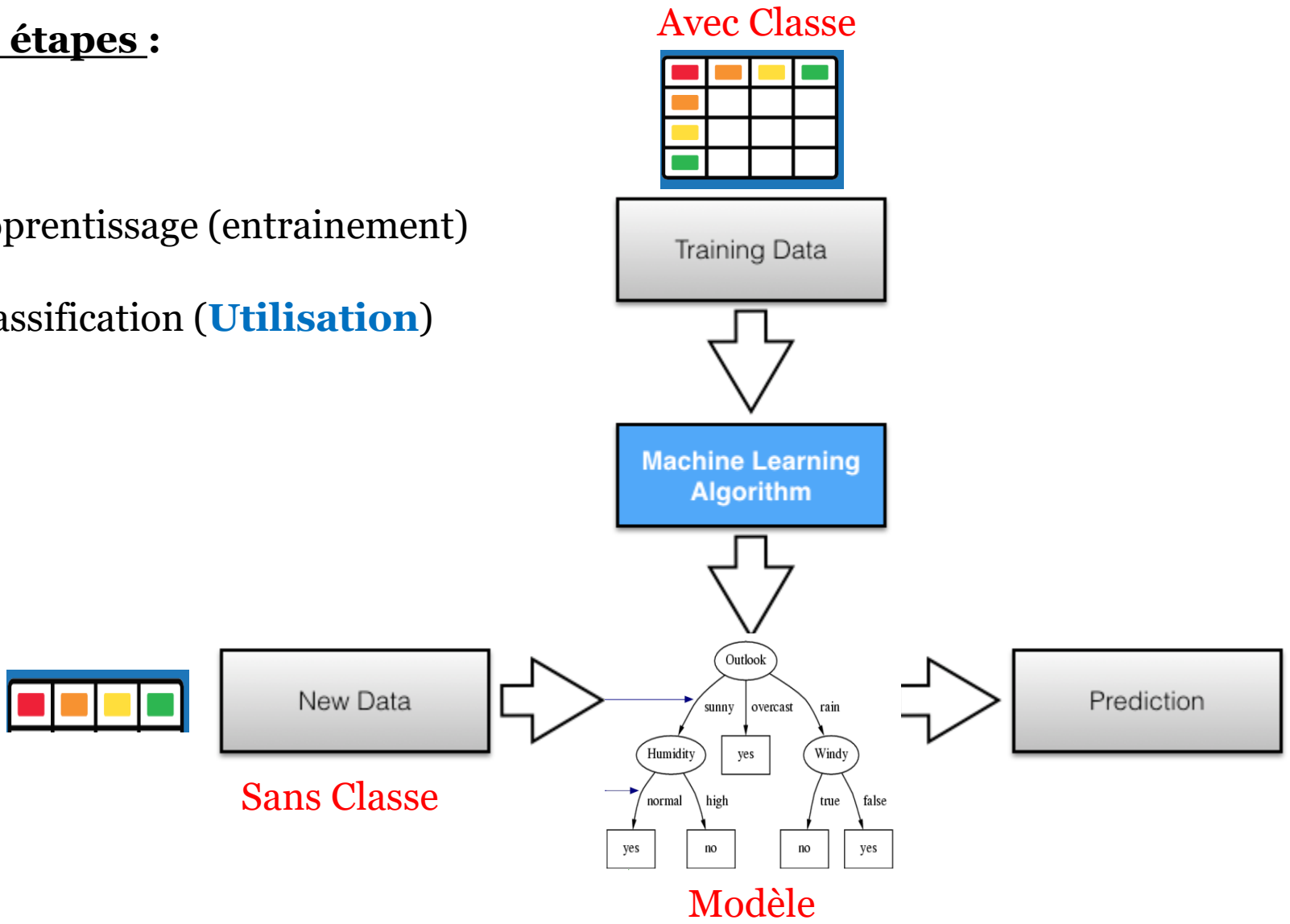
1. Apprentissage (**entrainement**)



Evaluation du modèle

Deux étapes :

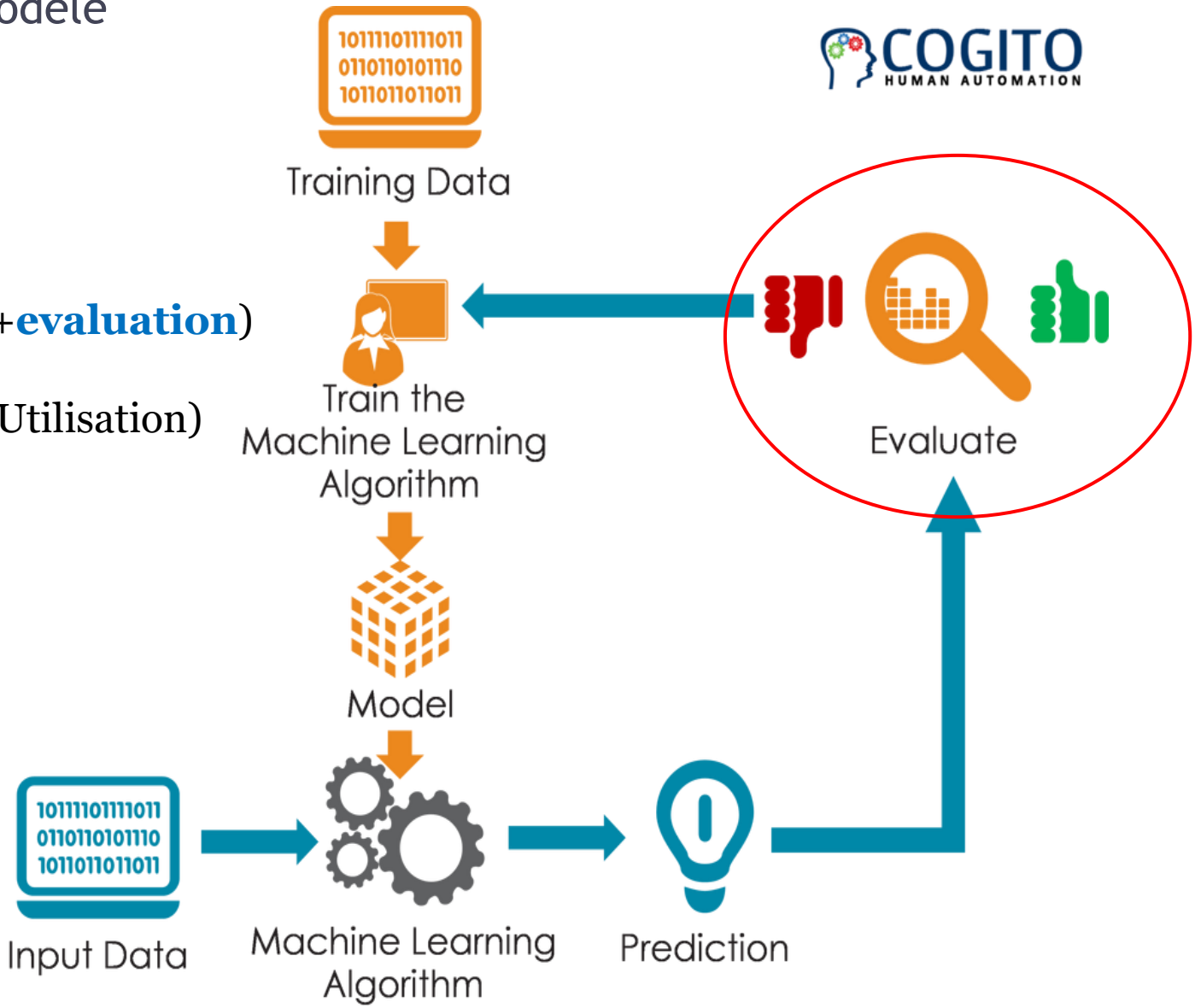
1. Apprentissage (entrainement)
2. Classification (**Utilisation**)



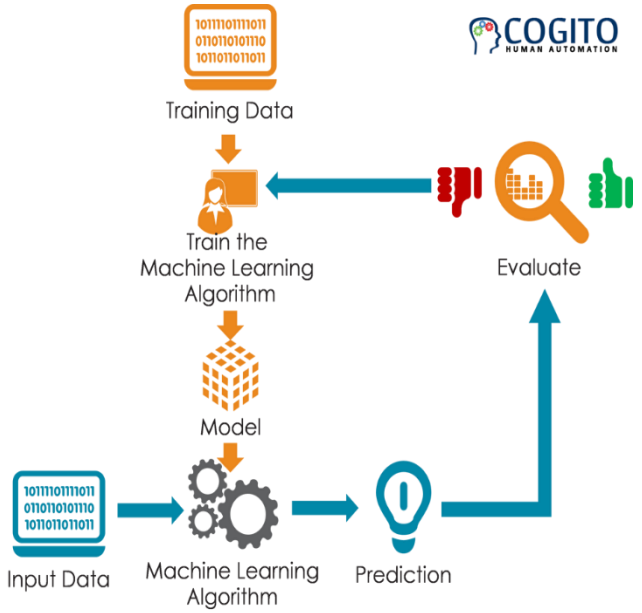
Evaluation du modèle

Deux étapes:

1. Apprentissage (entrainement+**evaluation**)
2. Classification (Utilisation)



Méthodes d'évaluation



Deux étapes:

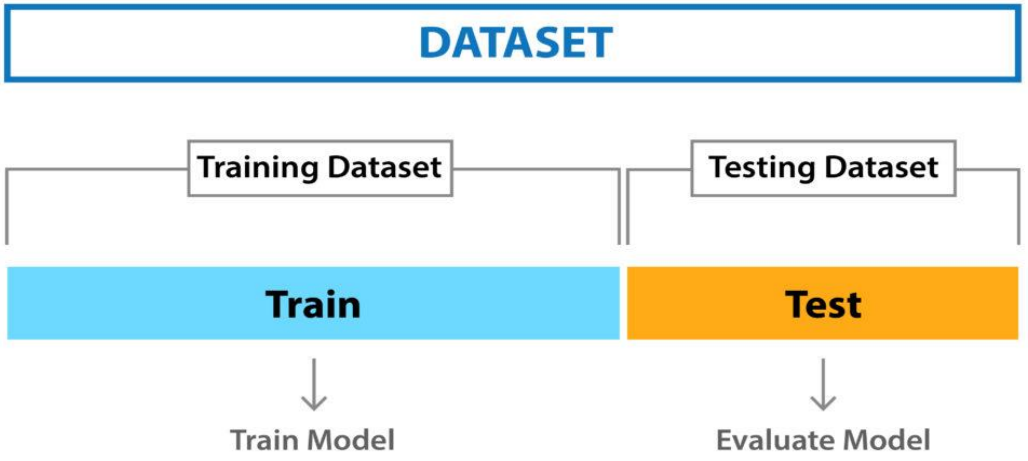
1. Apprentissage
   (entrainement + **evaluation on Test Set**)

1. Classification (Utilisation)

Deux bases d'exemples:

1. Training Set

2. Test Set



Méthodes d'évaluation

Deux étapes:

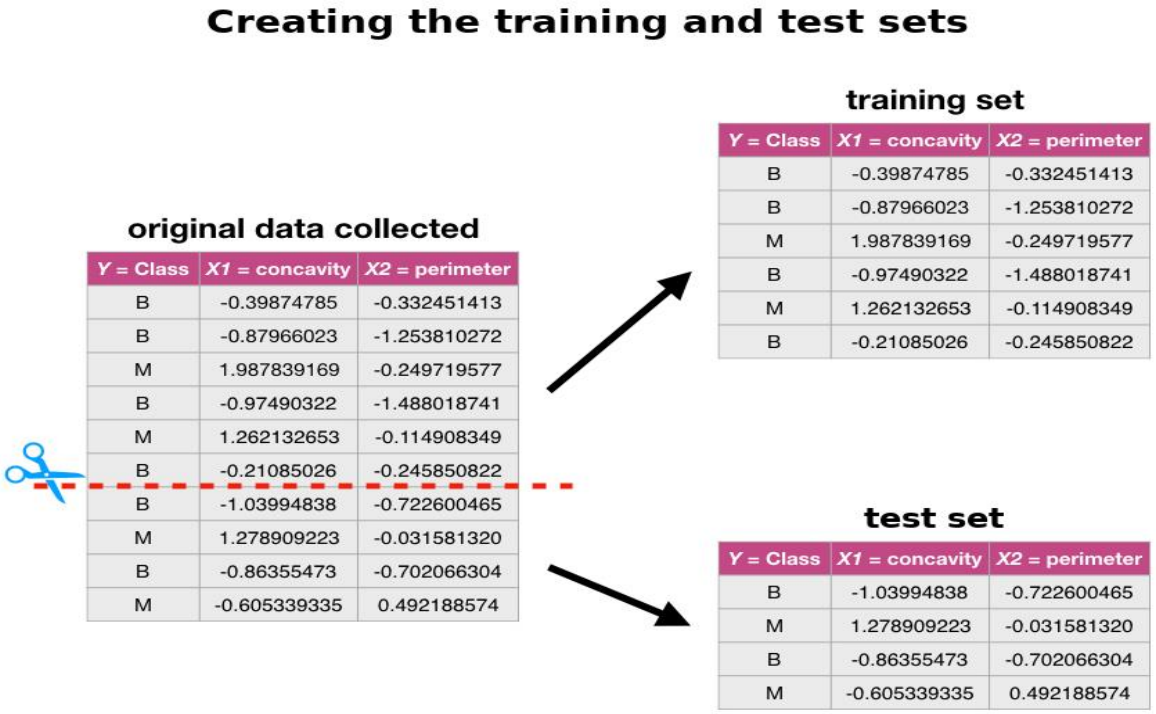
1. Apprentissage
   (entrainement + **evaluation on Test Set**)

1. Utilisation

Deux bases d'exemples:

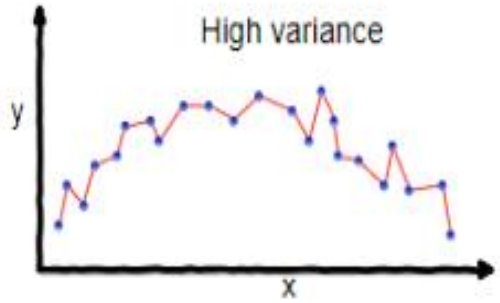
1. Training Set

2. Test Set

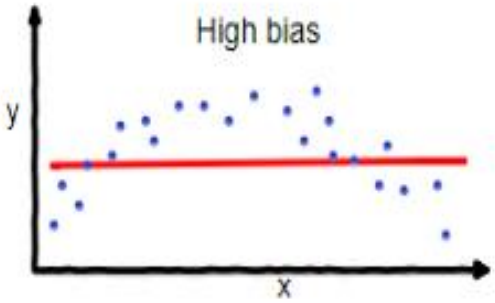


Evaluation du modèle

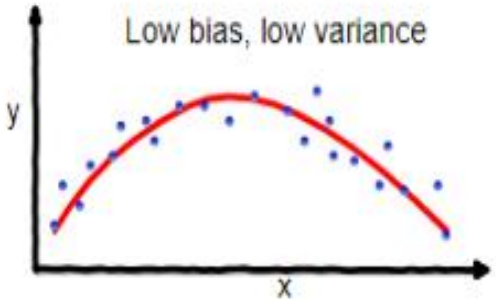
- Evaluer pour s'assurer de la capacité de généralisation en dehors des données d'entrainement.
- Overfitting/Underfitting – Variance/Biais tradeoff
- Permet de qualifier le comportement du modèle appris sur les données non utilisées lors de l'apprentissage.



overfitting



underfitting



Good balance

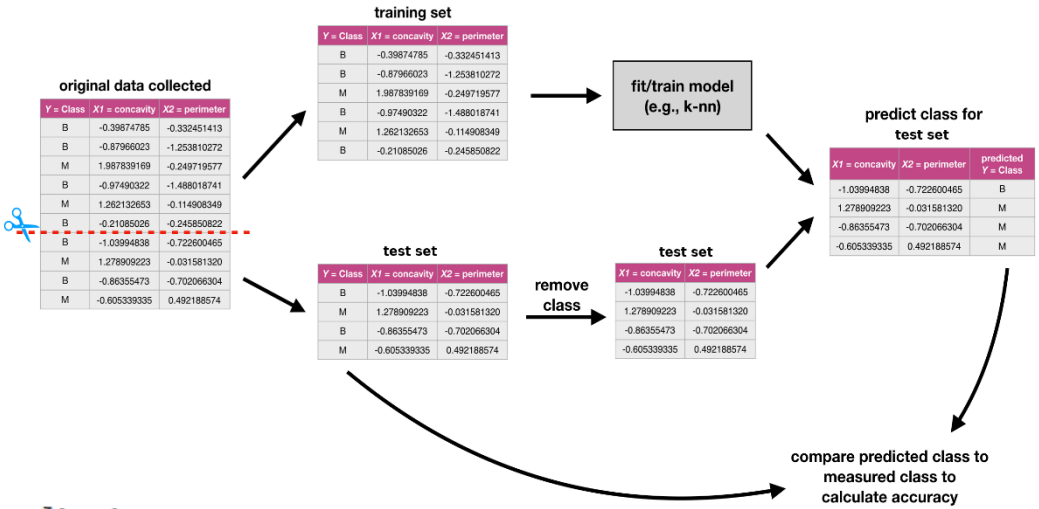
Evaluation du modèle

- Evaluer pour s'assurer de la capacité de généralisation en dehors des données d'entrainement.
- 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'entrainement ou autres exemples réservés pour le test.
- Différentes mesures/metrics d'évaluation :
  - ✓ Accuracy
  - ✓ Sensitivité
  - ✓ Spécificité
  - ✓ Moyenne harmonique, F1 Score
  - ✓ Etc.

Evaluation du modèle

Précision ***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*



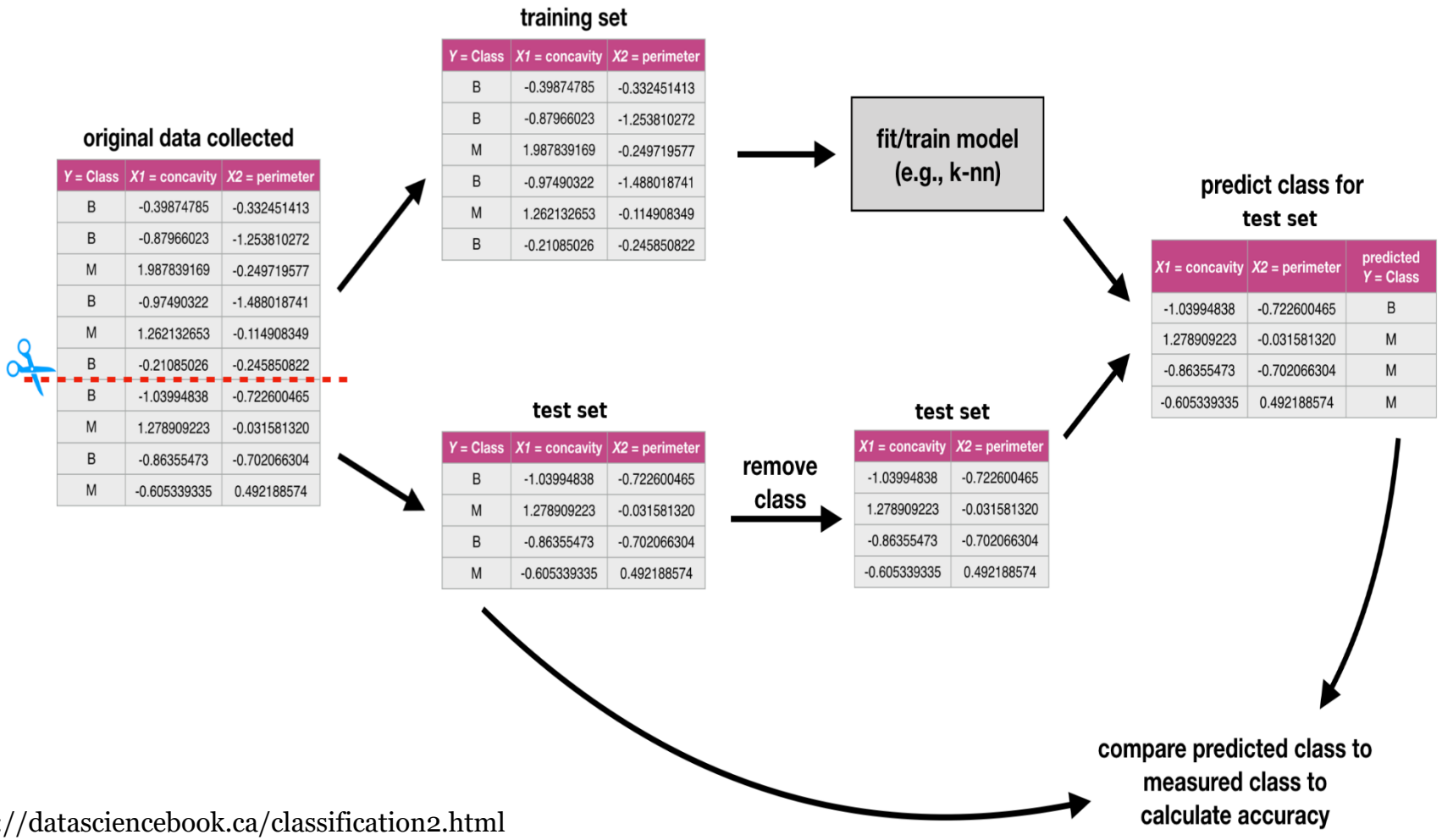
$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

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

Evaluation du modèle

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

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



https://datasciencebook.ca/classification2.html

Matrice de confusion

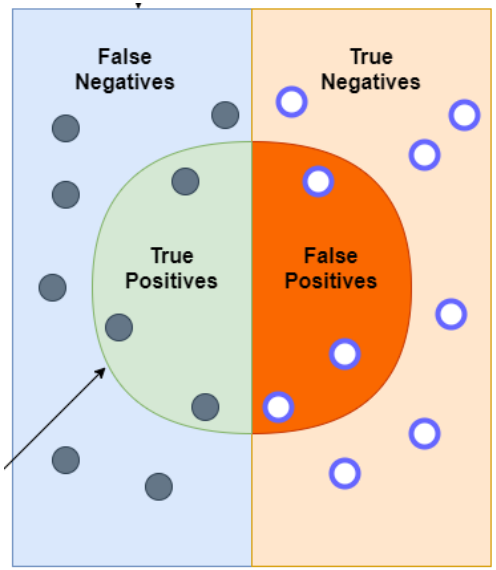
- Aucune information sur la nature des erreurs.
- Exemple : considérer un échantillon non cancéreux alors qu'il l'est, est beaucoup plus grave de considérer un échantillon cancéreux alors qu'il ne l'est pas.
- Cas de classification binaire : Observation (y) et Prédiction (f)

$$\begin{cases} \hat{f}(x_i) = \text{Yes} \quad et \quad y_i = \text{Yes} \quad correcte \;\; positive \\ \hat{f}(x_i) = \text{Yes} \quad et \quad y_i = \text{No} \quad fausse \;\; positive \\ \hat{f}(x_i) = \text{No} \quad et \quad y_i = \text{No} \quad correcte \;\; négative \\ \hat{f}(x_i) = \text{No} \quad et \quad y_i = \text{Yes} \quad fausse \;\; négative \end{cases}$$

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

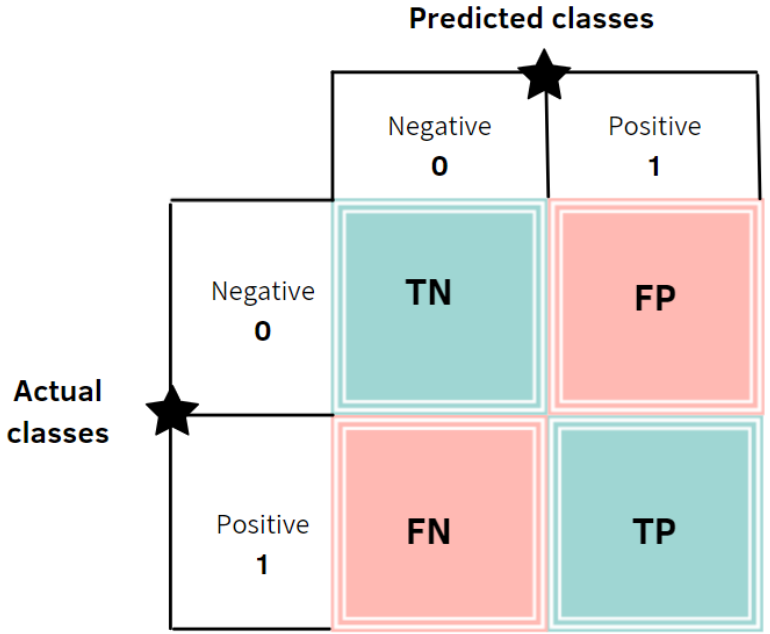


		Prédiction	
		Yes	No
Observation	Yes	**CP**	**FN**
	No	**FP**	**CN**

Evaluation du modèle

Matrice de confusion

- **Attention :**



		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

		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:

		Prédiction	
		Chat	Chien
Observation	Chat	**CP**	**FN**
	Chien	**FP**	**CN**

- 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.

Evaluation du modèle

Matrice de confusion

Exemple 1:

		Prédiction	
		Chat	Chien
Observation	Chat	**5**	**3**
	Chien	**2**	**4**

- 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.

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

		Prédiction	
		Yes	No
Observation	Yes	**CP**	**FN**
	No	**FP**	**CN**

Autres mesures :

- Sensitivité – True Positive Rate:

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

- Spécificité – True Negative Rate:

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

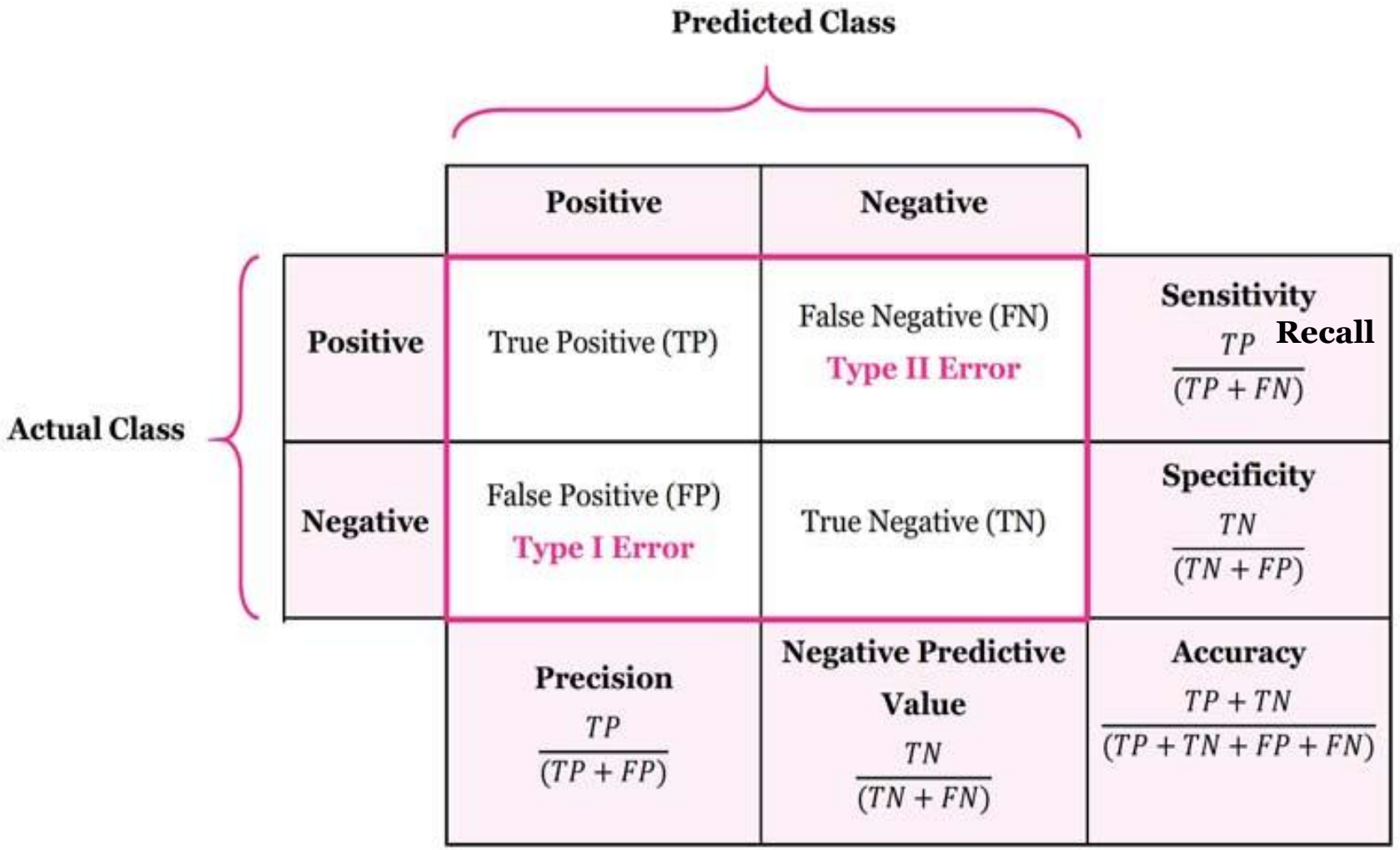
- Moyenne harmonique :

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

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Evaluation du modèle

Matrice de confusion: Autres mesures



Evaluation du modèle

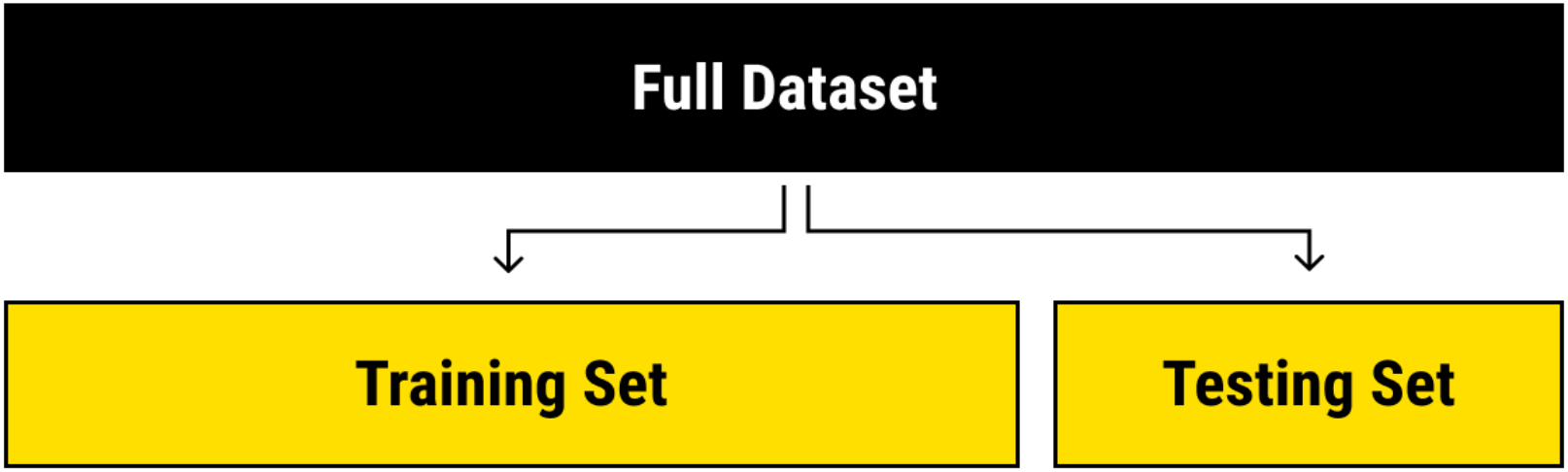
Matrice de confusion

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*)

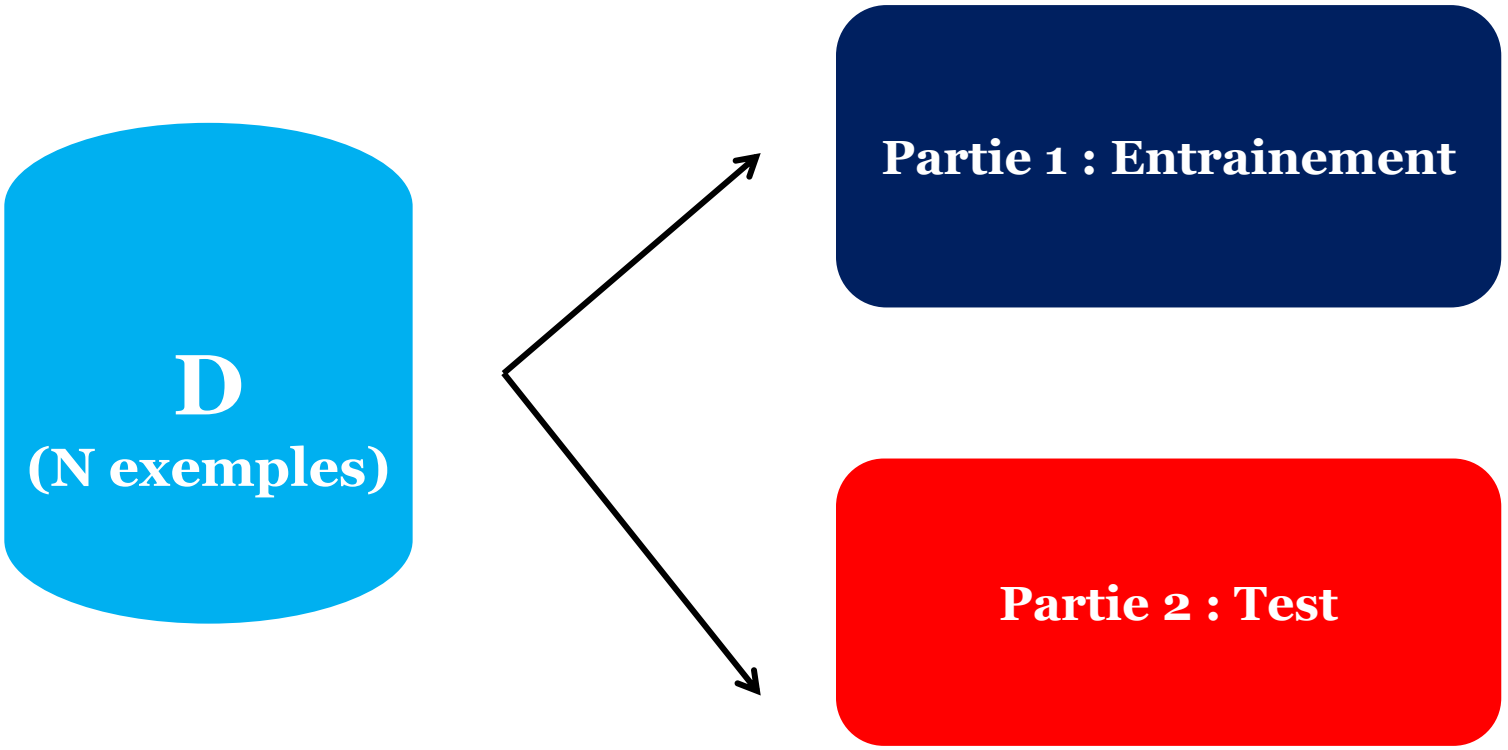
Classification performance metrics based on the confusion matrix.

Méthodes de validation



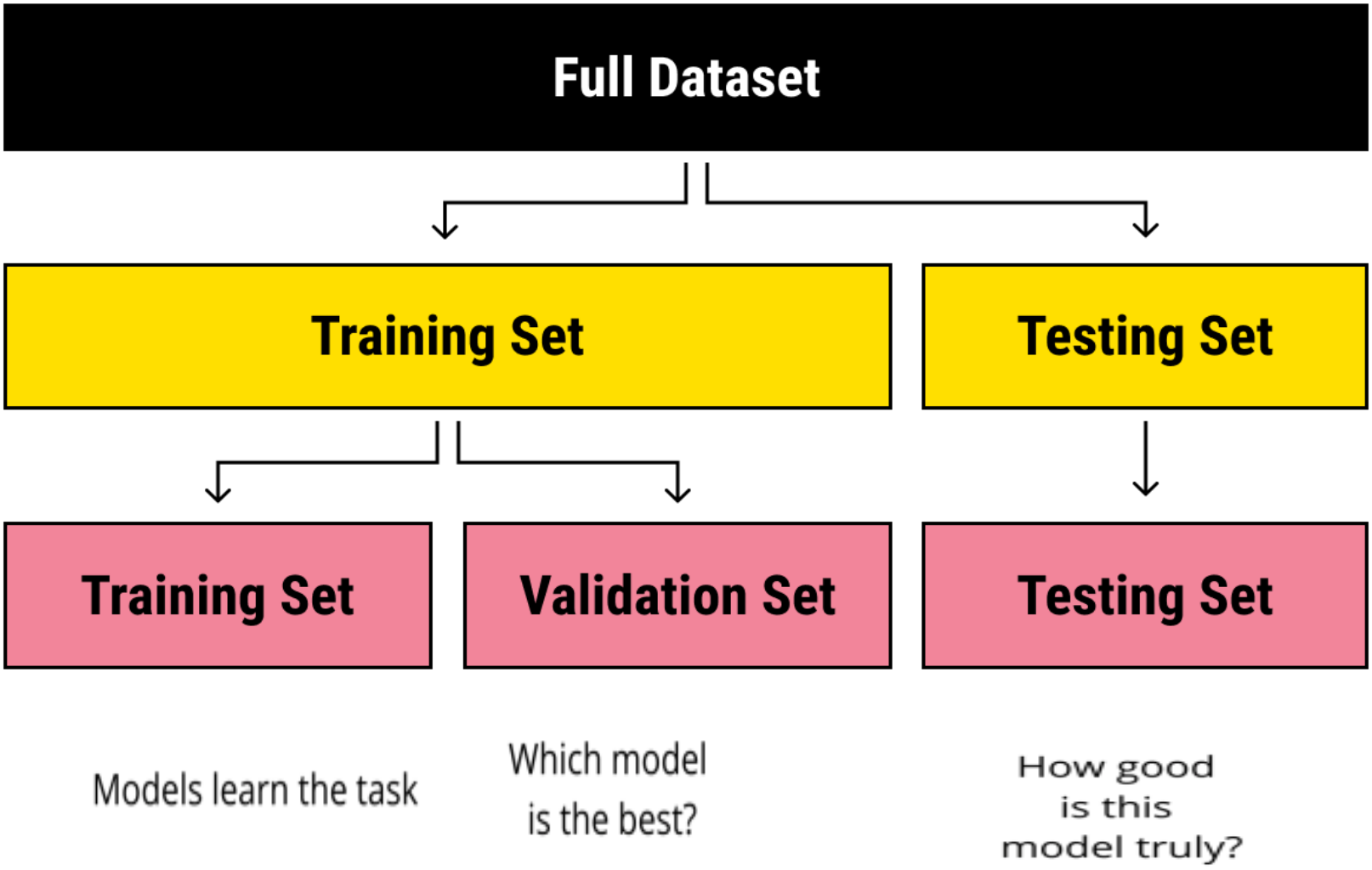
Méthodes de validation

Méthode HoldOut

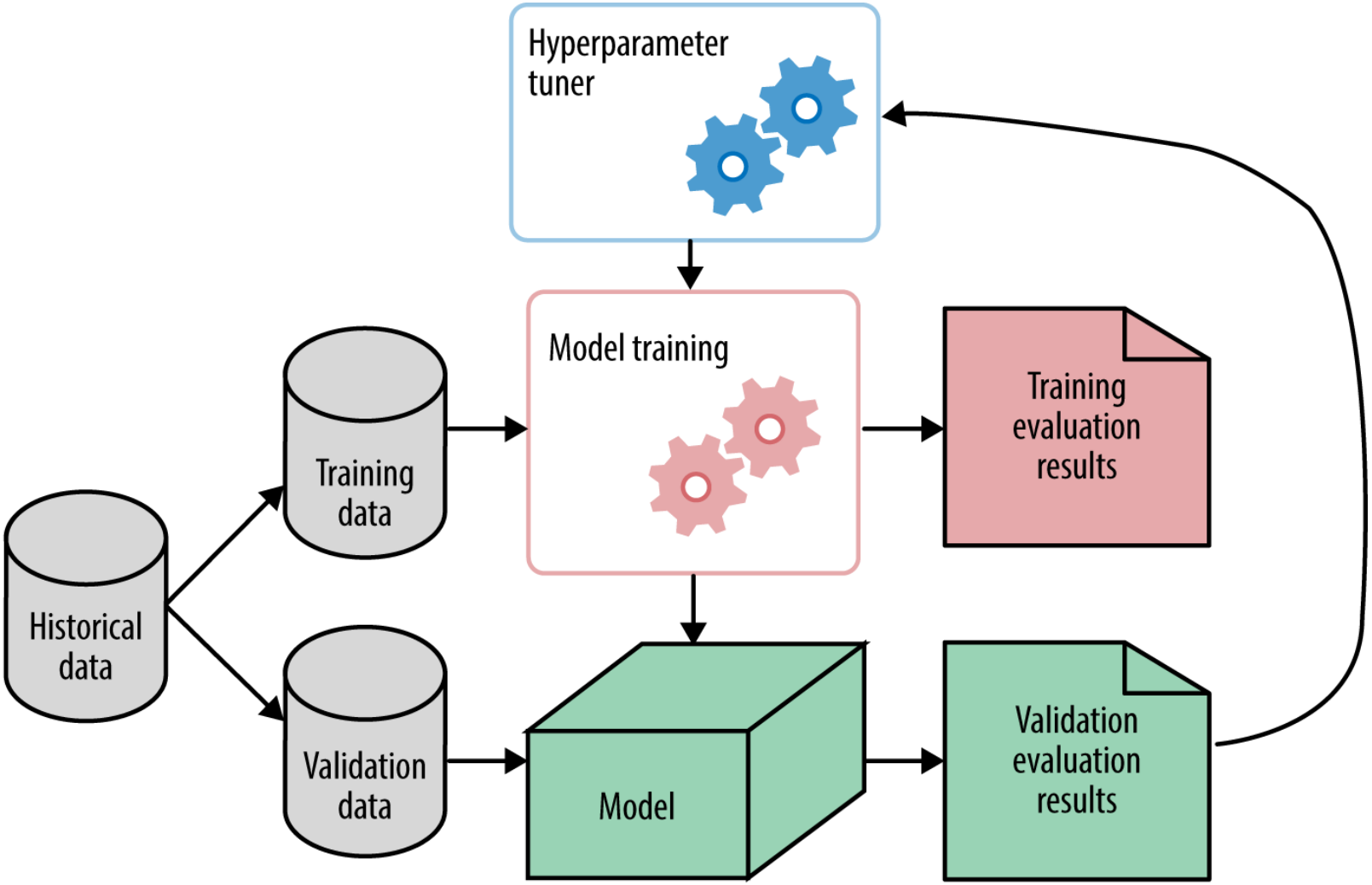


Maximiser précision Test => Maximiser précision Modèle

Méthodes de validation



Méthodes de validation



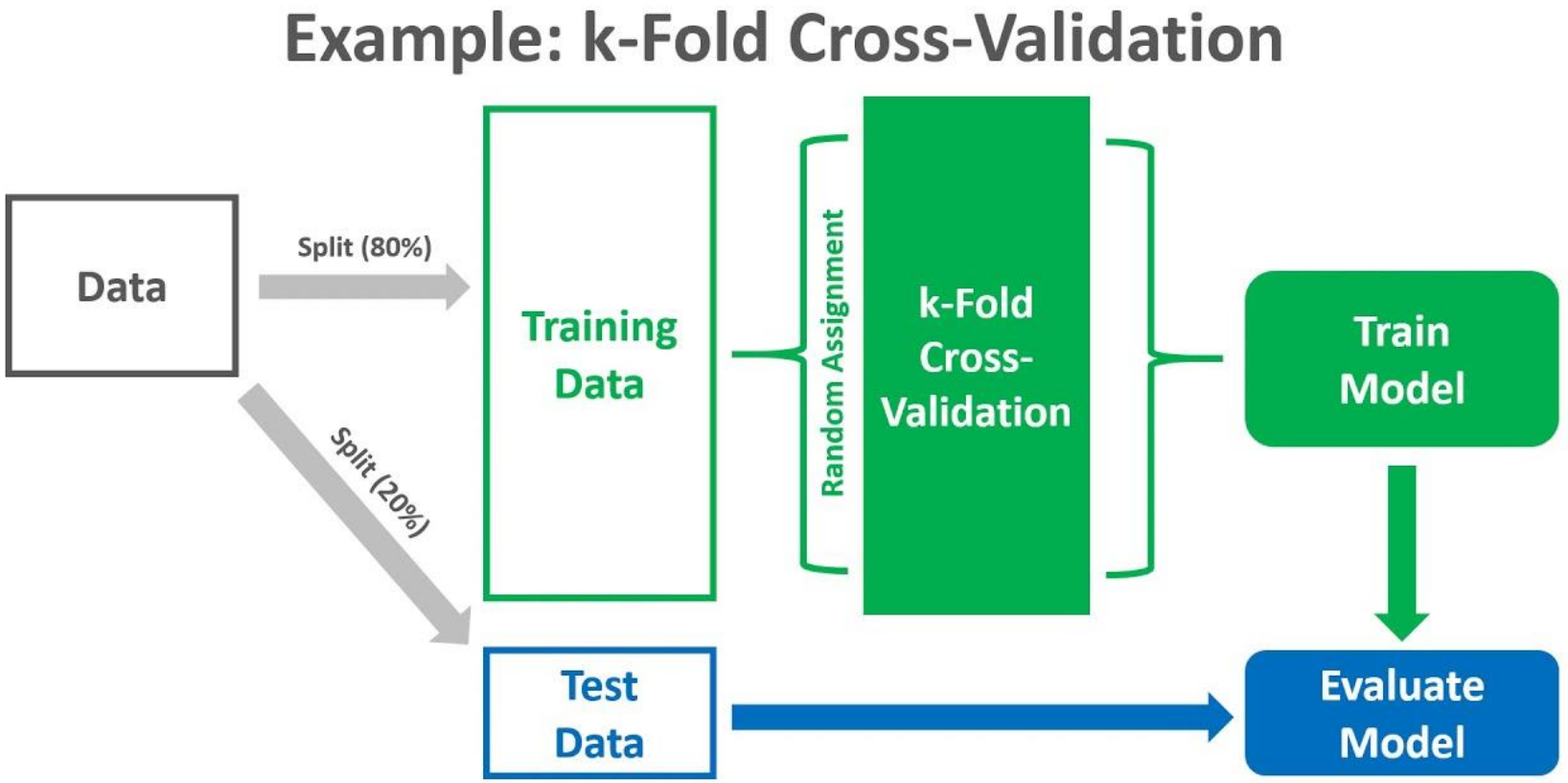
Méthodes de validation

- 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'entrainement.

Méthodes de validation

Méthode validation croisée

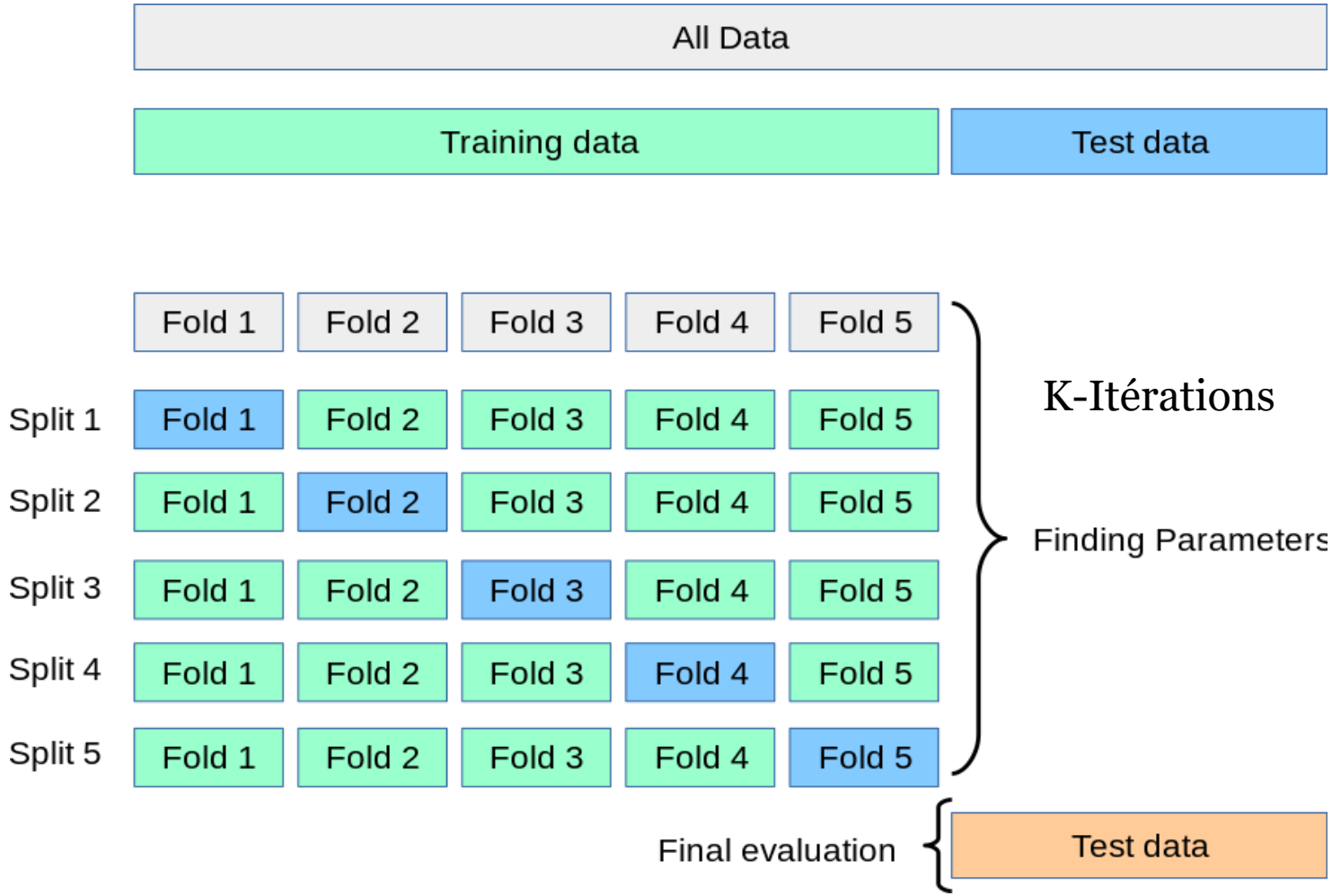
Subdiviser ***D*** en ***k*** sous-ensembles de même taille - Folds.



Méthodes de validation

Méthode validation croisée

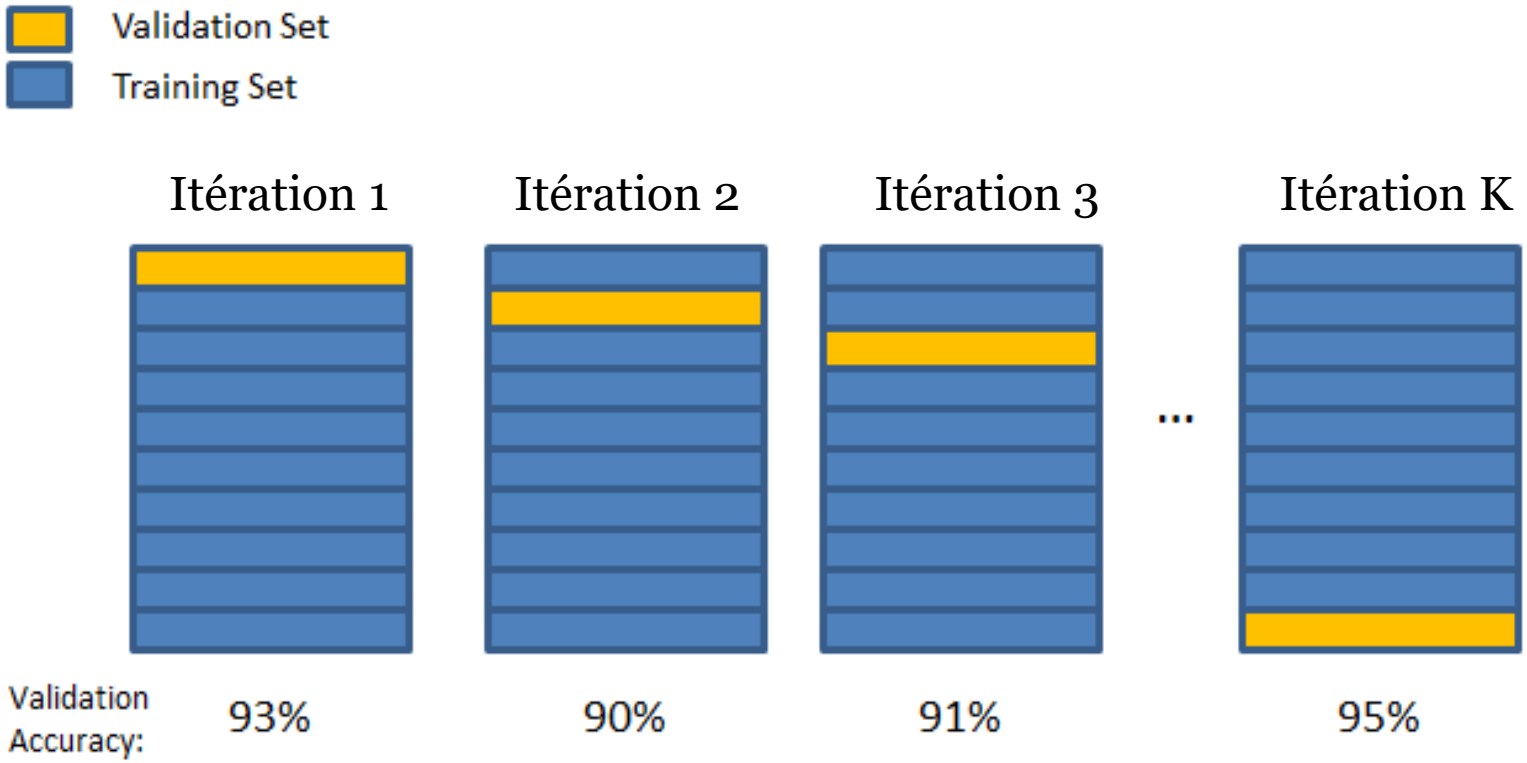
Subdiviser ***D*** en ***k*** sous-ensembles de même taille - Folds.



Méthodes de validation

Méthode validation croisée

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



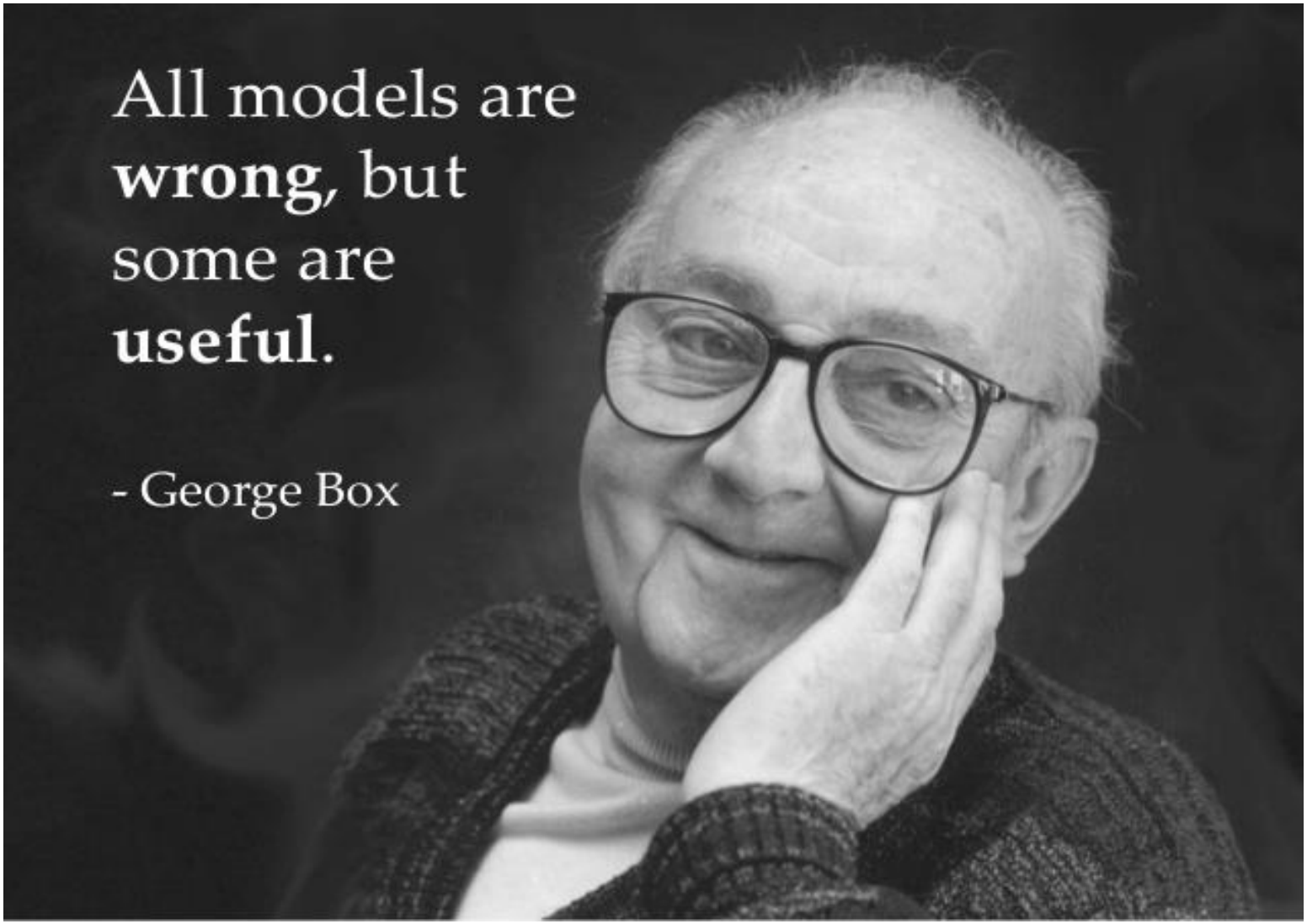
Précision Finale = Moyenne (Itération 1, Itération 2, ...)

Méthodes de validation

Méthode Bootstrap

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

Combinaison de modèles – Ensemble Learning



Combinaison de modèles

C. Michael Gibson MD
@CMichaelGibson

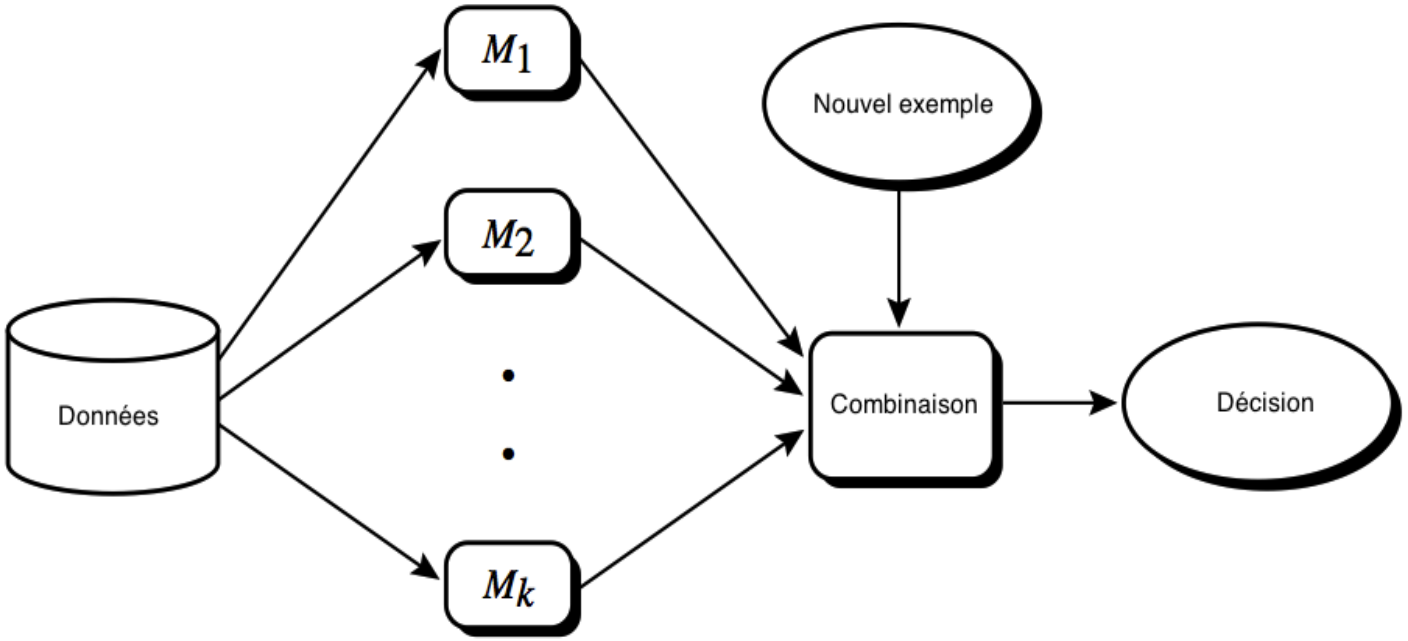
Suivre

Determining if an image is a Chihuahua or muffin is a tough problem in artificial intelligence

Combinaison de modèles – Ensemble Learning

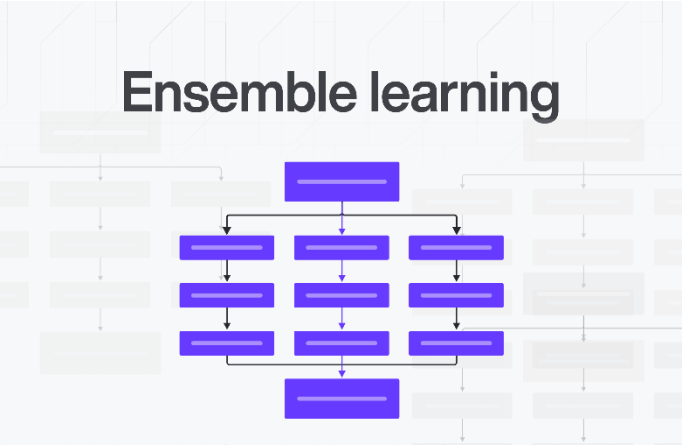
- Combiner plusieurs modèles avec des performances faibles (weak learners) permettant d'obtenir un modèle plus efficace (meta-learner).

- Créer un grand nombre de petits modèles rapidement puis développer un modèle qui les rassemble.

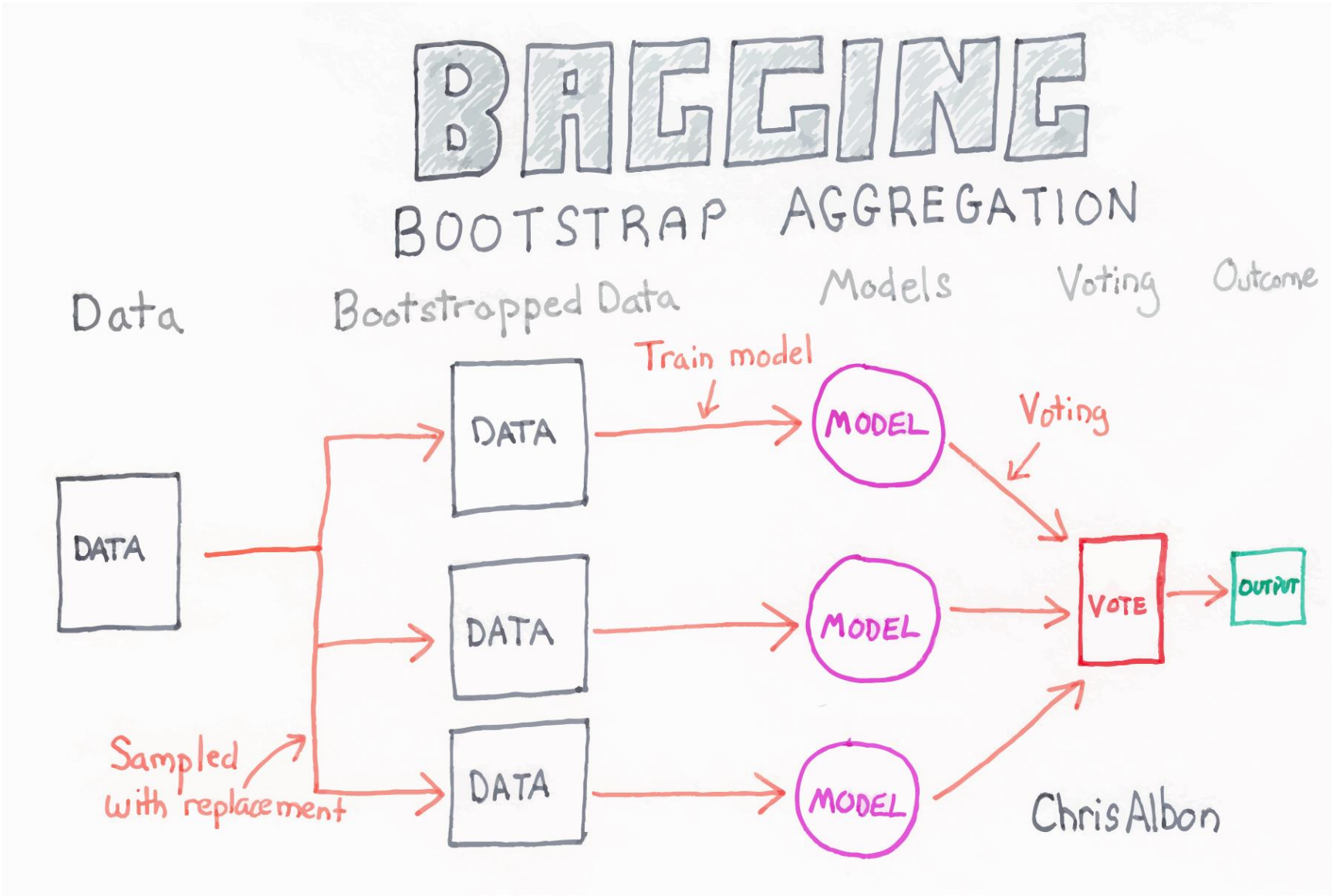


Combinaison de modèles – Ensemble Learning

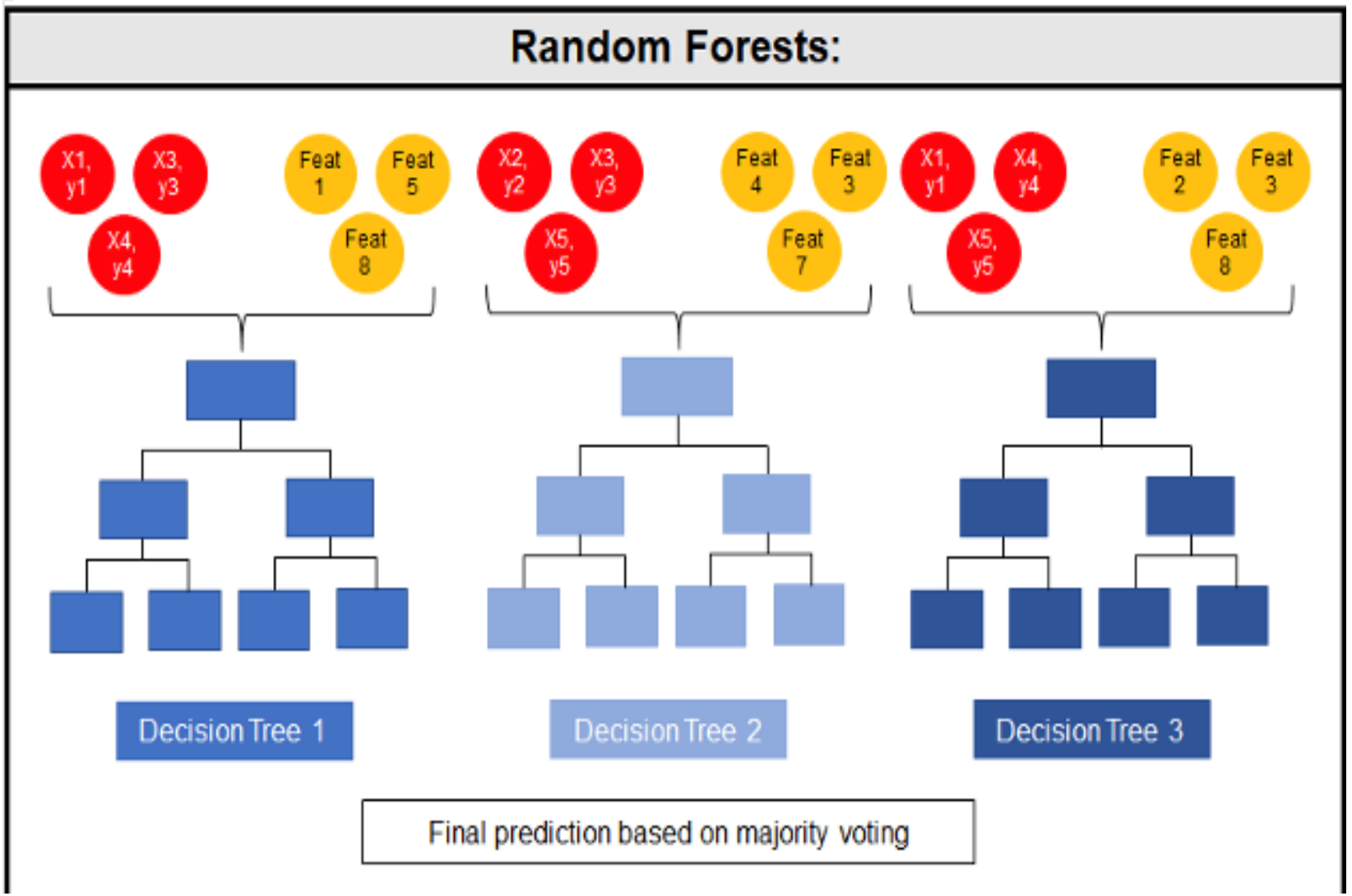
- Deux méthodes : Parallèles et séquentielles.
- Méthodes **parallèles** : Bagging et Random Forest.
- Méthodes **séquentielles** : Boosting (AdaBoost, Gradient Boosting, XGBoost etc.)
- **Stacking**, Stacked Generalization (Wolpert, 1992).



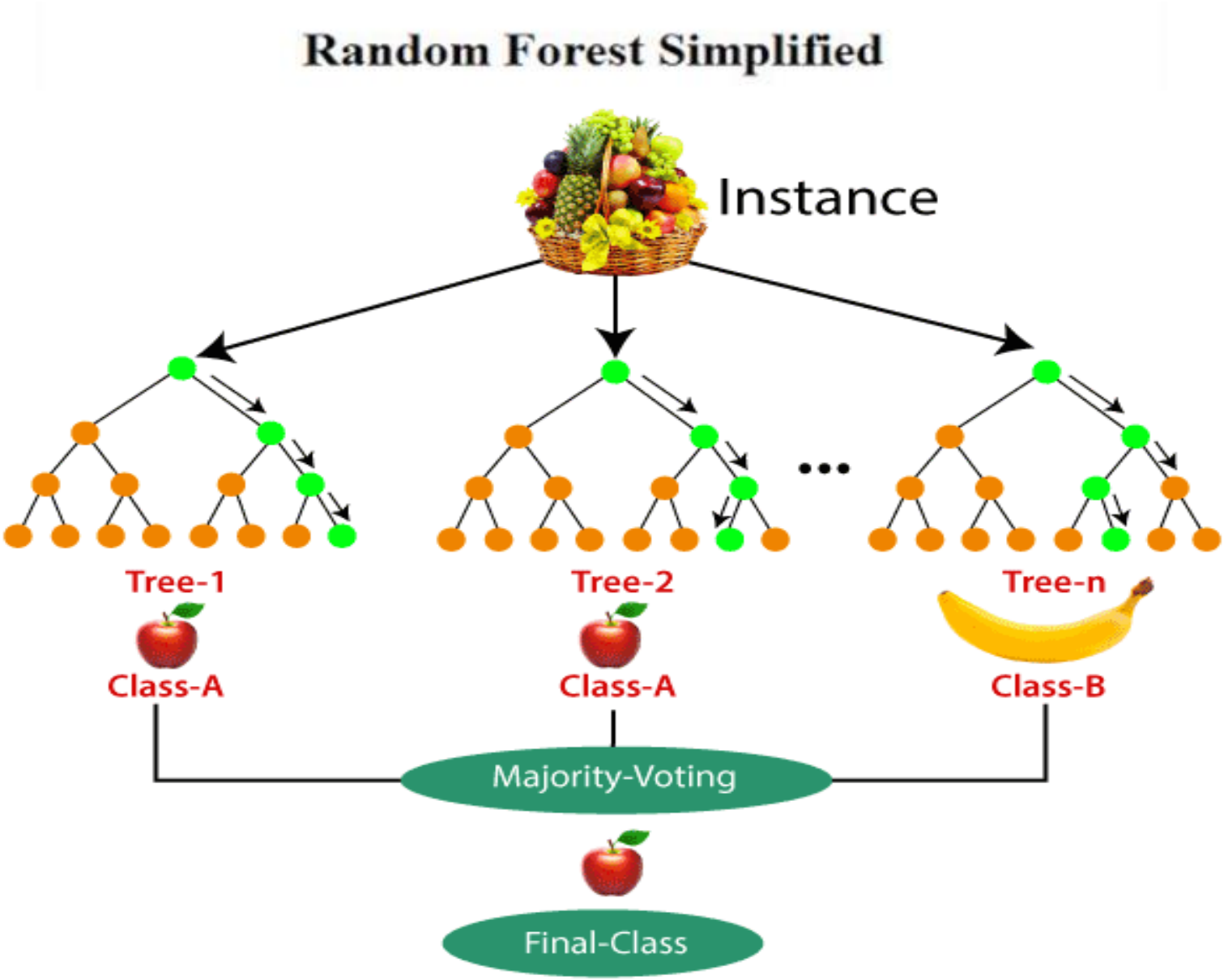
Combinaison de modèles – Ensemble Learning



Combinaison de modèles – Ensemble Learning

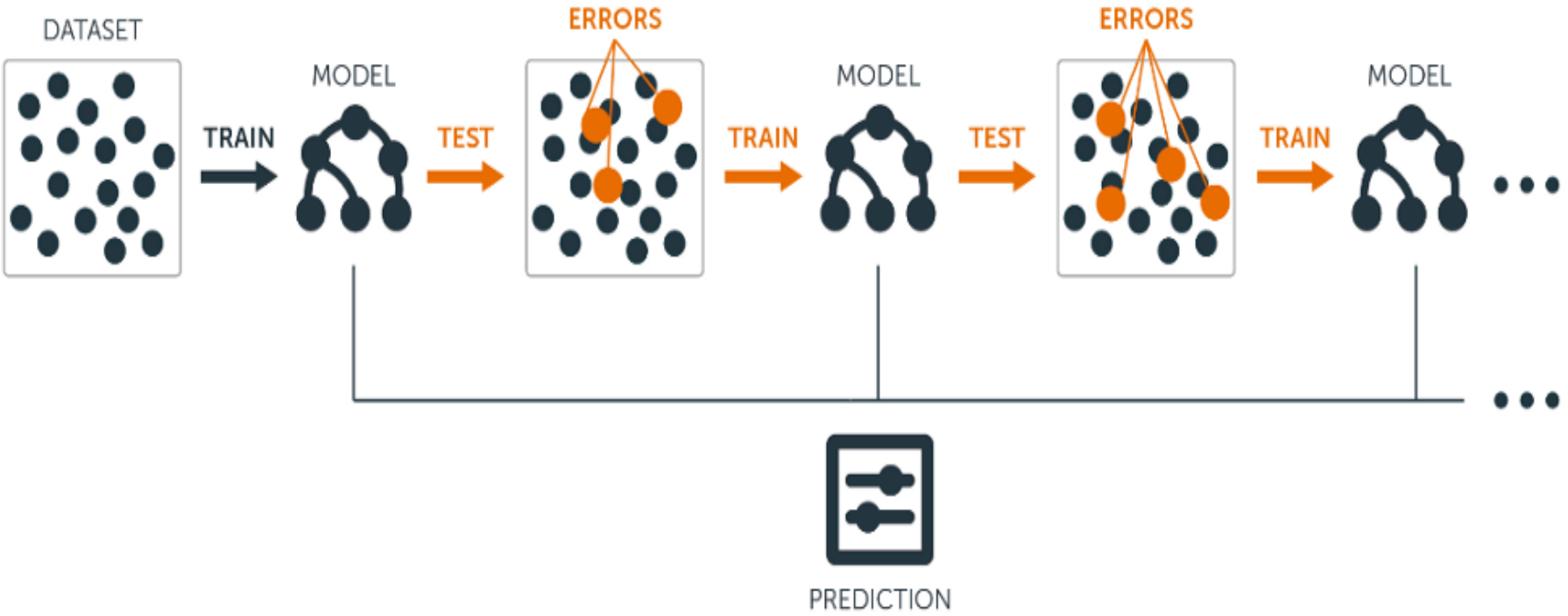


Combinaison de modèles – Ensemble Learning



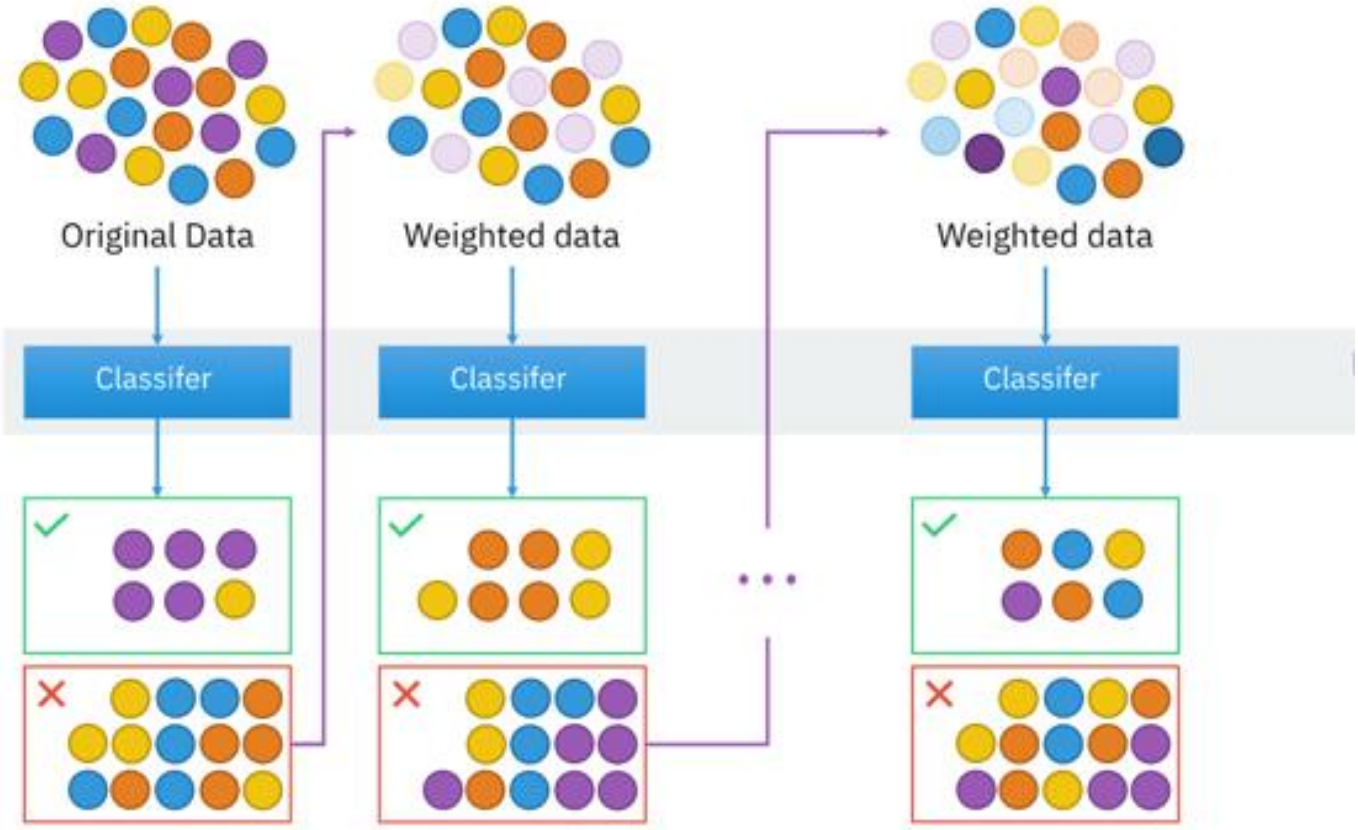
Boosting

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



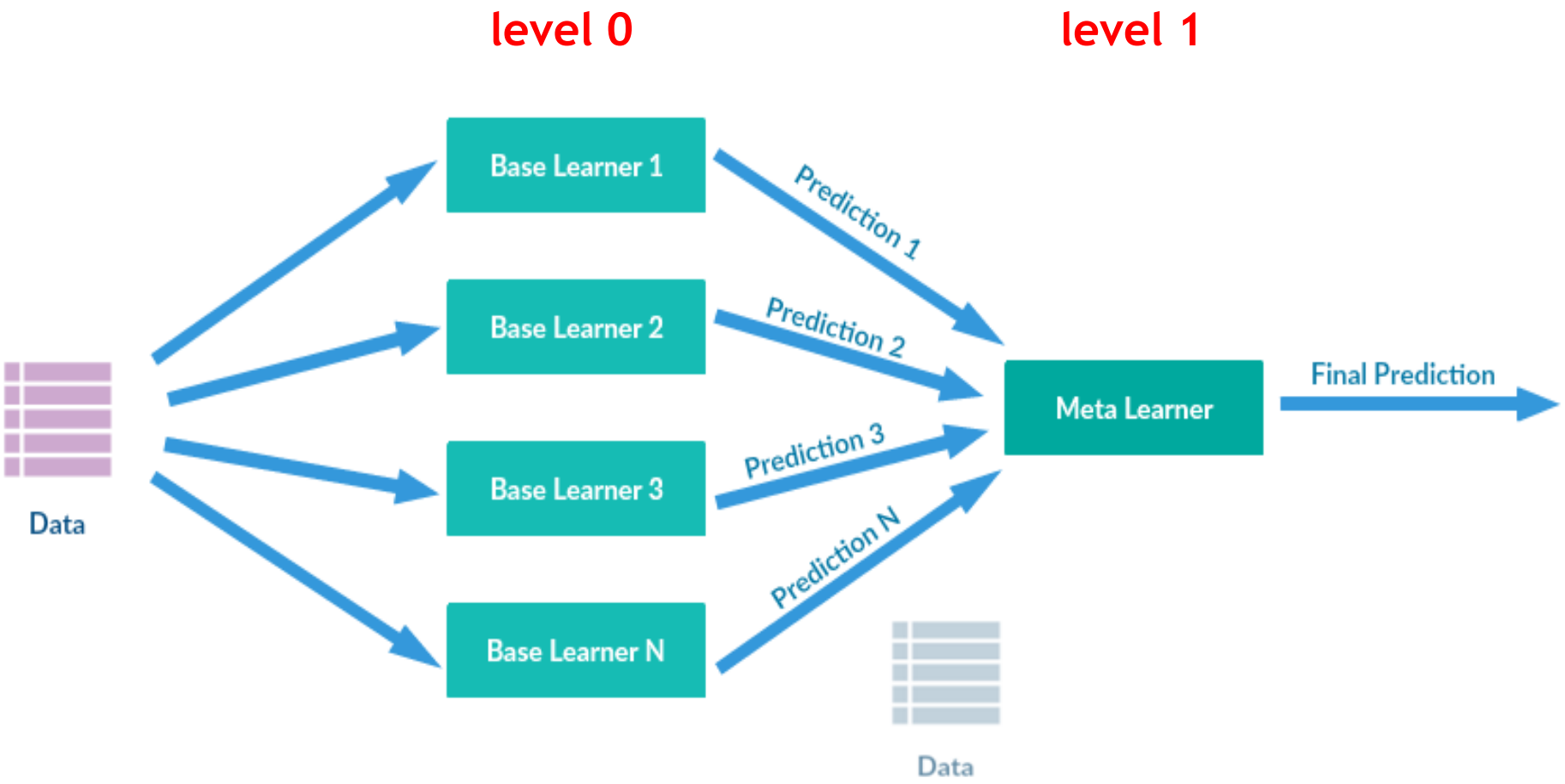
Boosting

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



Stacking/Blending

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



Stacking/Blending

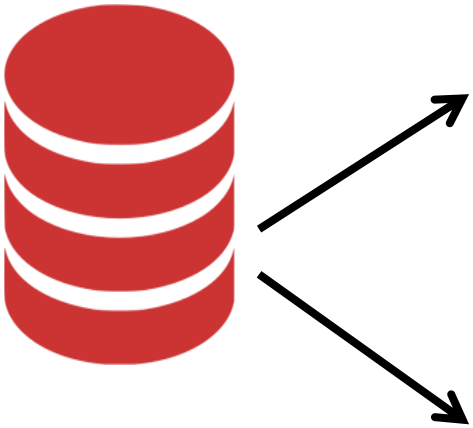
- Deux niveaux d'algorithmes. Principalement, algos de types différents.

Level 1 training data

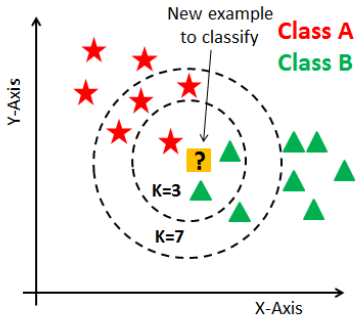
Data Point #	prediction from base learner 1	prediction from base learner 2	prediction from base learner 3	prediction from base learner M	actual
1	$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$	$\hat{y}_{1M}$	y_1
2	$\hat{y}_{21}$	$\hat{y}_{22}$	$\hat{y}_{23}$	$\hat{y}_{2M}$	y_2
…	…	…	…	…	…
N	$\hat{y}_{N1}$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

K plus proche voisins

SAVOIR - PREDIRE - DECIDER



Données

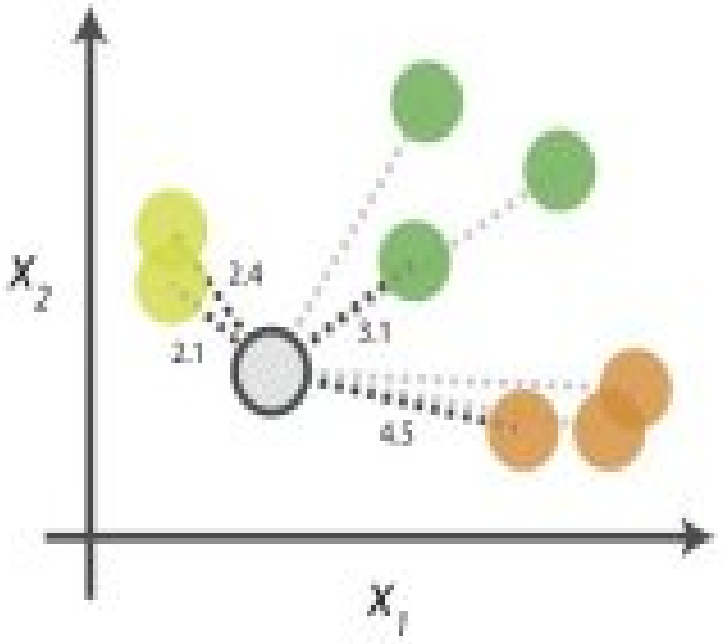


Connaissances

K-Nearest Neighbors

K plus proche voisins

- KNN – K Nearest Neighbors
- Algorithme de classification les plus simple.
- Principe : Calcul la distance entres tous les exemples de la base et le nouvel exemple qu'on cherche à classifier.
- 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	sqrt[(7-3)²+(7-7)²]=4
7	4	Bon	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 $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Les 3 plus proches exemples:

	Scénario	Jeu d'acteurs	Classe	Distance
3	7	7	Bon	sqrt[(7-3)²+(7-7)²]=4
	~~7~~	~~4~~	~~Bon~~	~~sqrt[(7-3)²+(4-7)²]=5~~
1	3	4	Mauvais	3
2	1	4	Mauvais	3.60

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

K plus proches voisins

Exemple : K =3

Euclidean

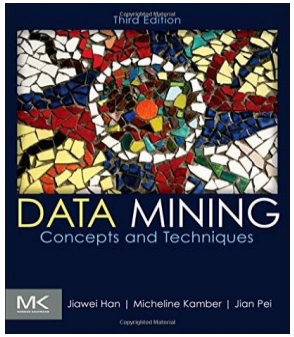
$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Choix de la classe majoritaire

	Scénario	Jeu d'acteurs	Classe	Distance
3	7	7	Bon	sqrt[(7-3)²+(7-7)²]=4
	~~7~~	~~4~~	~~Bon~~	~~sqrt[(7-3)²+(4-7)²]=5~~
1	3	4	Mauvais	3
2	1	4	Mauvais	3.60

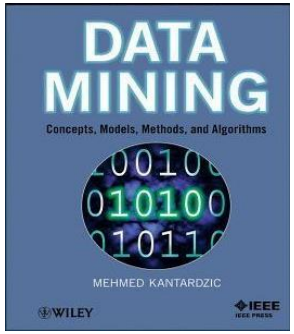
Test Data : Scénario = **3** , Jeu d'acteurs = **7**, classe = **Mauvais**

Ressources



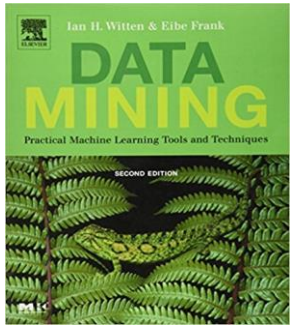
Data Mining : concepts and techniques, 3rd Edition

- ✓ 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/