



EPITECH

DEV- 810

ZOIDBERG 2.0



ZOIDBERG 2.0

KICK OFF

1. Le contexte
2. Votre mission
3. Votre boîte à outil
4. Votre rendu
5. Planning



LE CONTEXTE



Grâce aux applications de l'IA à la médecine, il est désormais possible d'analyser massivement toutes sortes d'images dans le but de dépister les tumeurs et autres anomalies.

En radiologie, en dermatologie, ou encore en ophtalmologie, l'IA permet de détecter des maladies invisibles à l'œil nu et d'établir des prévisions. Elle peut aussi aider à adapter et à personnaliser les traitements. Elle est même utilisée à titre expérimental aux urgences pour orienter plus rapidement les patients.

VOTRE MISSION

Un collectif de médecins fait appel à vous pour les aider à mieux diagnostiquer les cas de pneumonies sur leurs patients

Vous devez donc créer un programme basé sur du **Machine learning** qui permettrait de détecter les cas de pneumonie.

Vous disposez pour cela de 3 datasets contenant les images médicales de leurs patients sur lesquels vous entraînerez votre modèle

VOTRE MISSION

Les différentes étapes :

- Explorer/ transformer vos dataset
- Choisir à minima 2 algorithmes de ML
- Entraîner les modèles
- Les évaluer et les optimiser grâce à la cross validation
- Les comparer

*Vous pouvez utiliser des réseaux de neurones, du deep learning ou tout autres algorithmes, mais ne perdez pas de temps dessus. **Concentrez vous sur des algo simples de ML et la réduction de dimensions** de vos données.*

VOTRE BOÎTE À OUTIL

Vous allez donc avoir recours à différentes techniques :

- De la réduction de dimension → PCA pour simplifier vos données d'entrées
- Des algorithmes de Machine Learning → des algos de classification –cas de ML supervisé
- Du feature engineering → pour optimiser les hyperparamètres de vos algos et obtenir les meilleurs résultats possibles

Vous

LE RENDU ATTENDU

Le rendu attendu devra contenir les principaux éléments suivants :

1. Un **rapport** synthétisant votre méthodologie, vos traitements des données , vos choix d'algorithmes utilisés, et vos métriques d'évaluations.
2. Un notebook python qui sache **prédire si un patient a une pneumonie ou non** au regard de sa radiographie et un **html-file** qui permettrait d'éviter de rerunner votre code
3. Une présentation power point le jour de votre oral

PLANNING

- 09/04/24 : Kick off bootstrap → découverte du ML supervisé et des familles d'algo , prise en main de la librairie scikitlearn
- 28/05/24 : Follow up n°1 → réduction de dimension et implémentation de 2 algos minimum
- 11/06/24 : Follow up n°2 → évaluation et optimisation des hyperparamètres
- 09/07/24 : Evaluation

ZOIDBERG 2.0

BOOTSTRAP

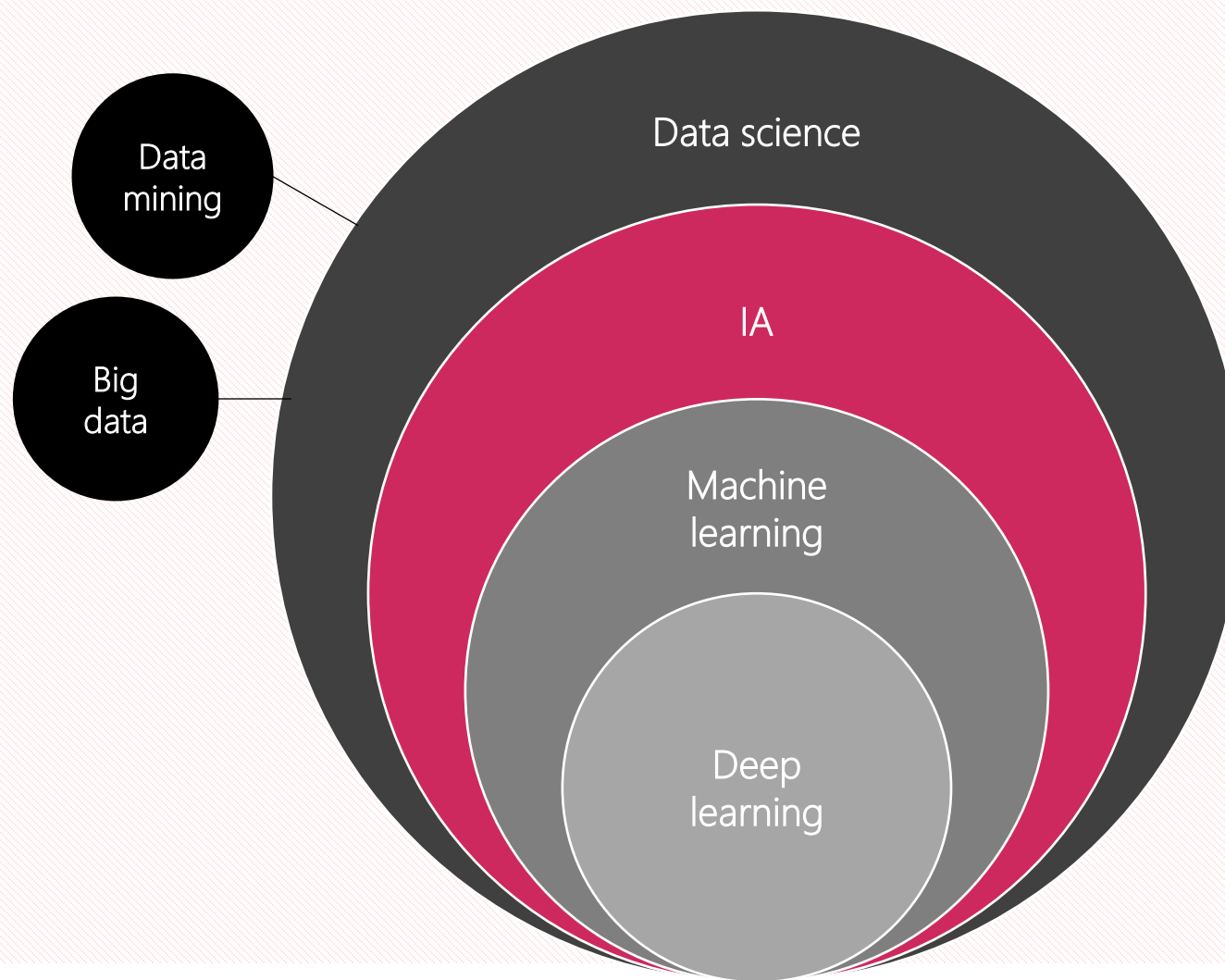
1. Introduction au Machine Learning
2. Réaliser un projet de machine learning
3. Réduction de dimension
4. Les algorithmes de classification
5. Évaluation du modèle
6. Optimisation des hyperparamètres

ZOIDBERG 2.0

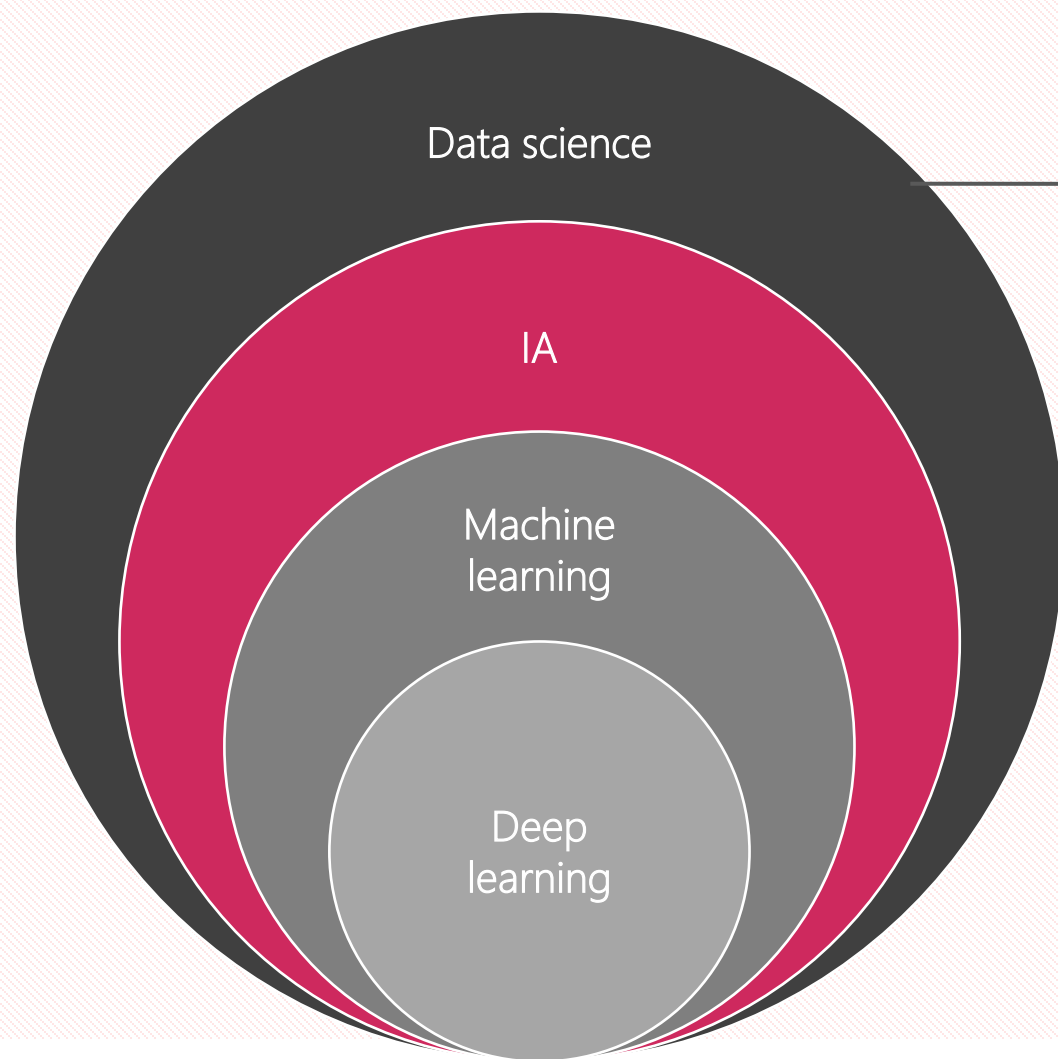
BOOTSTRAP

1. Introduction au Machine Learning
2. Réaliser un projet de machine learning
3. Réduction de dimension
4. Les algorithmes de classification
5. Évaluation du modèle
6. Optimisation des hyperparamètres

LES NOTIONS CLÉS

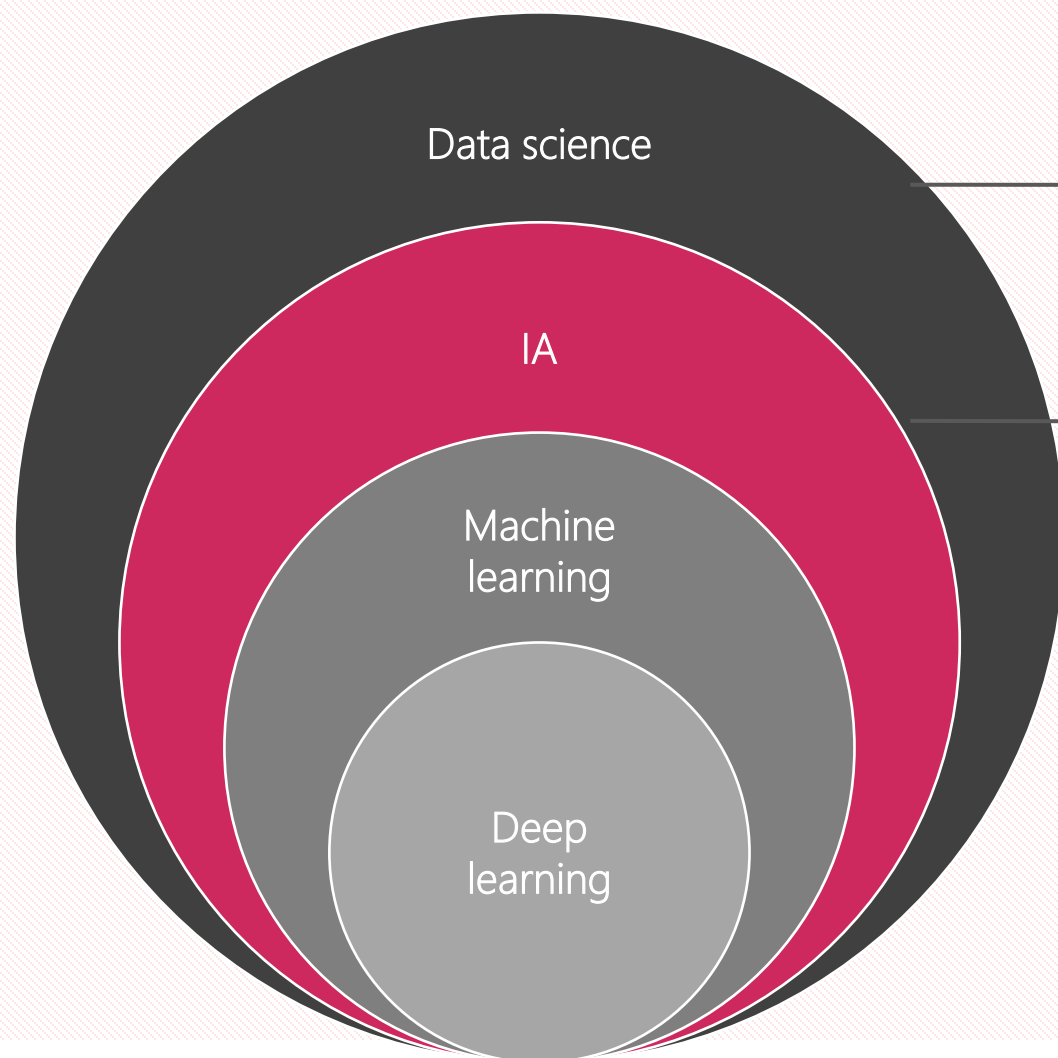


LES NOTIONS CLÉS



Inclus tout ce qui est lié à l'extraction, la collecte, la préparation et l'analyse des données dans le but de découvrir des informations, donner du sens, résoudre des problématiques c'est-à-dire **générer des informations exploitables**.

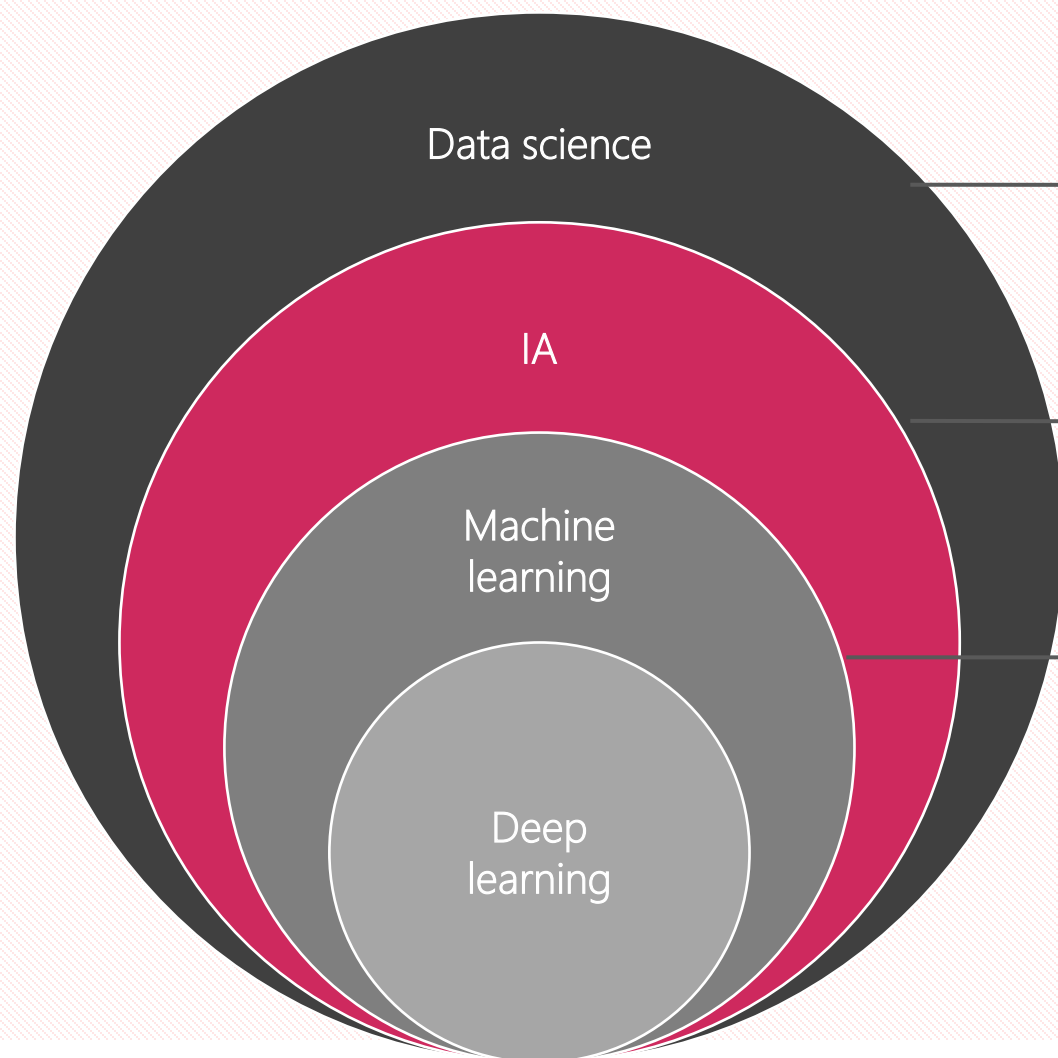
LES NOTIONS CLÉS



Inclus tout ce qui est lié à l'extraction, la collecte, la préparation et l'analyse des données dans le but de découvrir des informations, donner du sens, résoudre des problématiques c'est-à-dire **générer des informations exploitables**.

« l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine »
Larousse. Permet **aux ordinateurs d'imiter l'humain**

LES NOTIONS CLÉS

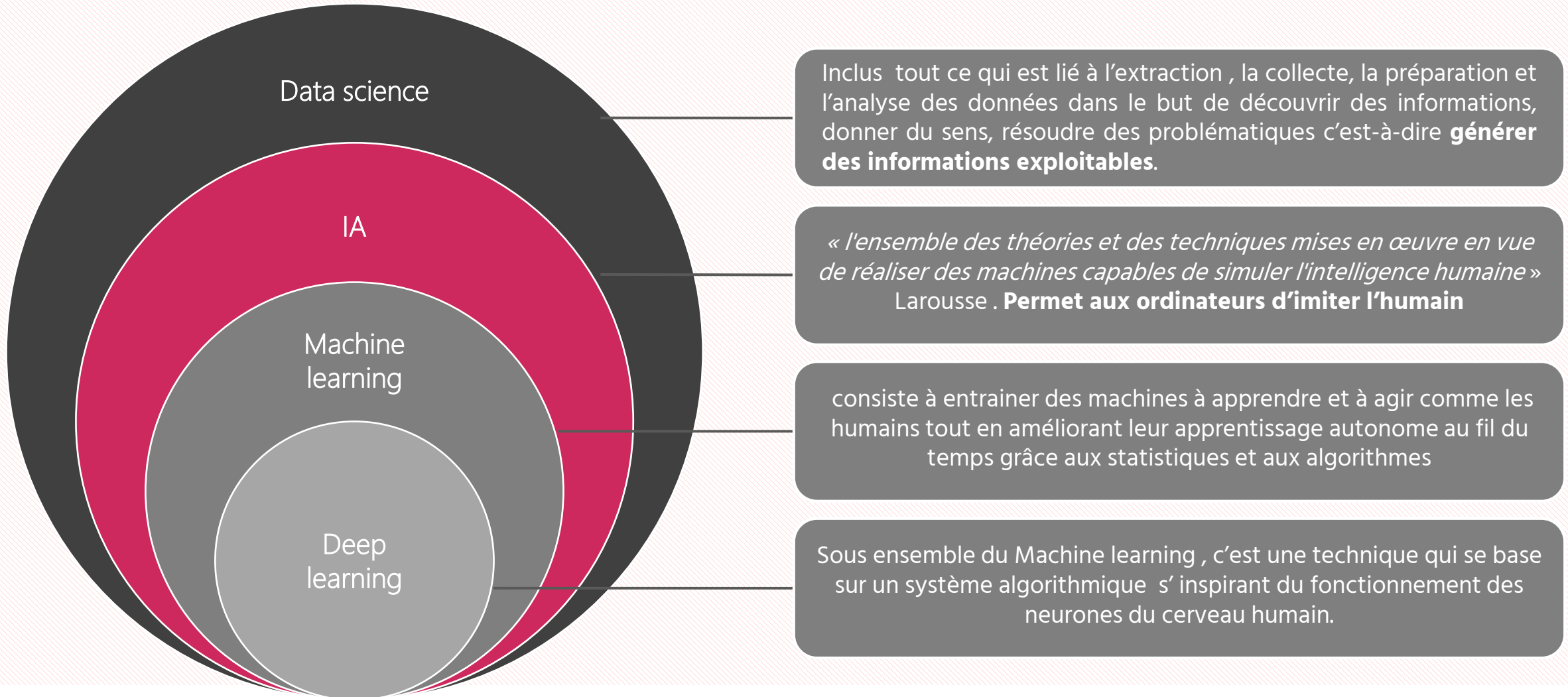


Inclus tout ce qui est lié à l'extraction, la collecte, la préparation et l'analyse des données dans le but de découvrir des informations, donner du sens, résoudre des problématiques c'est-à-dire **générer des informations exploitables**.

« l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine »
Larousse . **Permet aux ordinateurs d'imiter l'humain**

consiste à entraîner des machines à apprendre et à agir comme les humains tout en améliorant leur apprentissage autonome au fil du temps grâce aux statistiques et aux algorithmes

LES NOTIONS CLÉS



COMMENT ÇA MARCHE

LE PRINCIPE

Expérience

Les inputs :

- Le prix d'une action
- Des données clients
- image
- Etc.

Tâche

Définition des tâches que l'on veut faire :

- Classification d'image
- Segmentation
- Prédiction de prix
- Optimisation du parcours client

Performance

Définition des indices de performances :

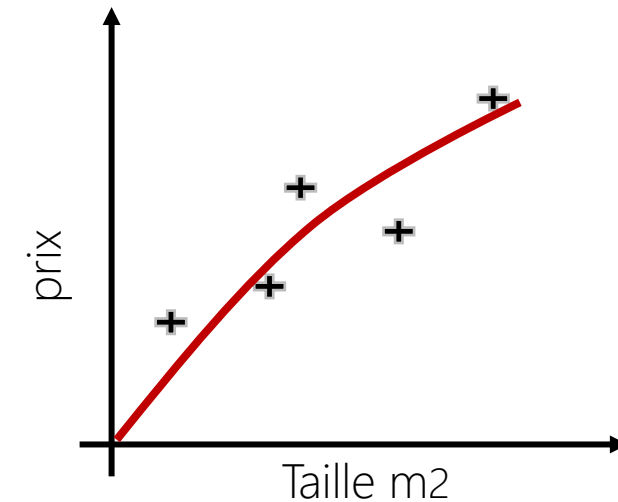
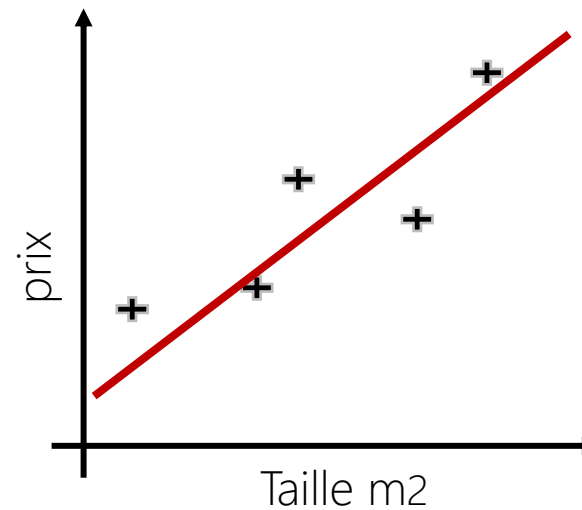
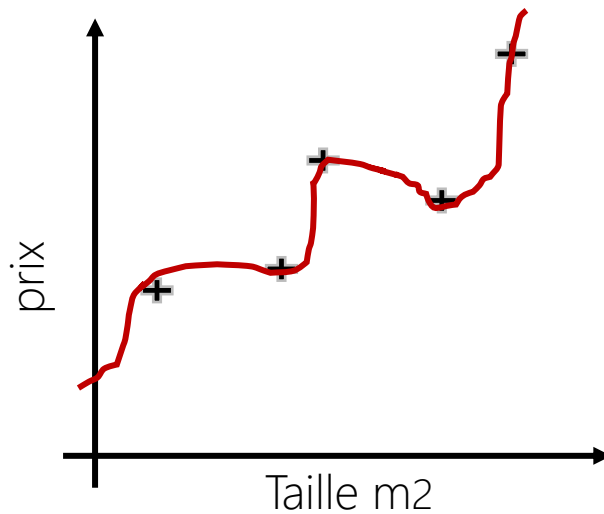
- KPI
- Classification correcte d'image
- Segmentation client cohérente

Un ordinateur apprend d'une expérience si pour réaliser une tâche précise son indice de performance s'améliore au cours du temps

COMMENT ÇA MARCHE

LE MODÈLE

A partir d'un **dataset**, on crée un modèle : une fonction mathématique qui relie les variables d'un problème entres elles. Les coefficients de cette fonction sont les **paramètres** du modèle.

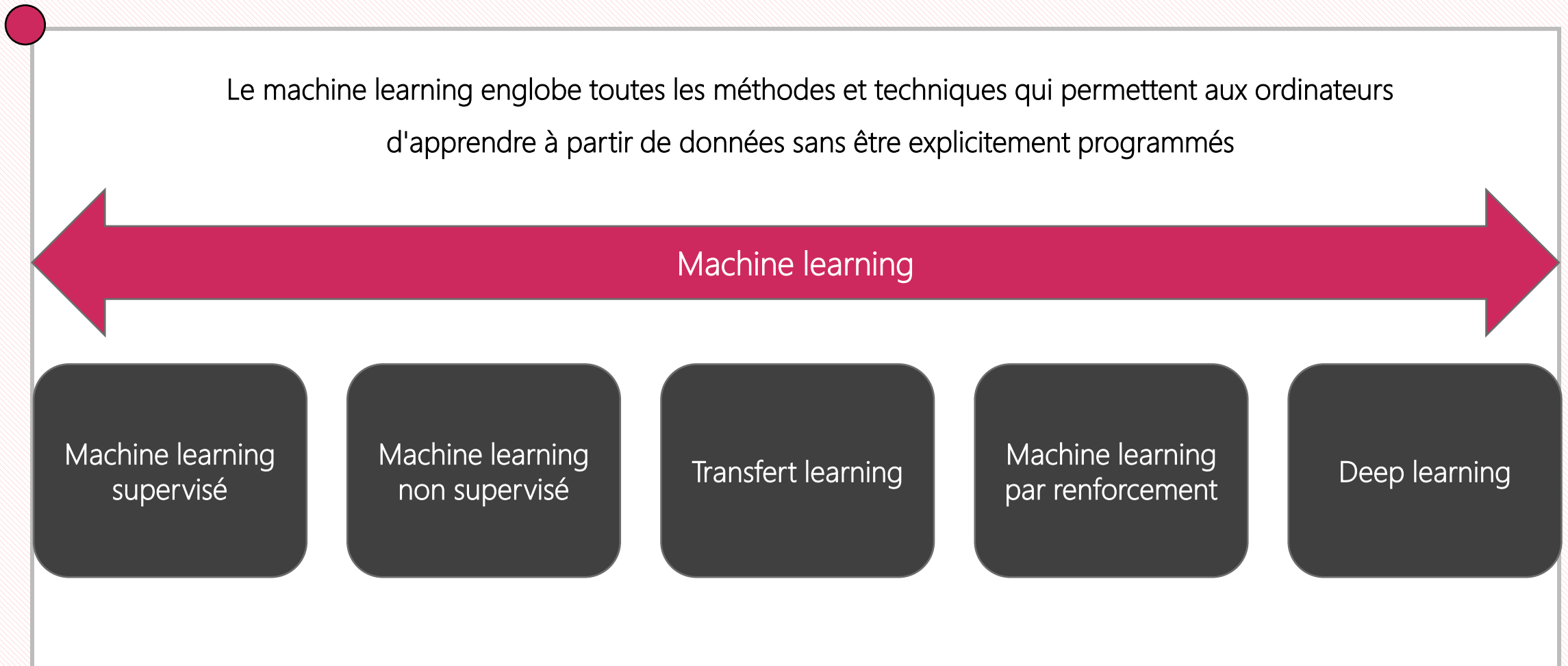


Le modèle doit être **généralisable** et **interprétable**

modèle d'apprentissage : trouve les paramètres pour faire marcher le modèle

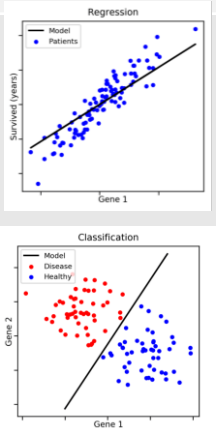
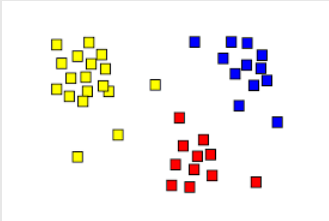
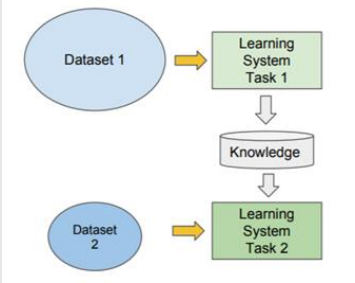
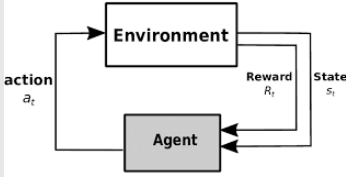
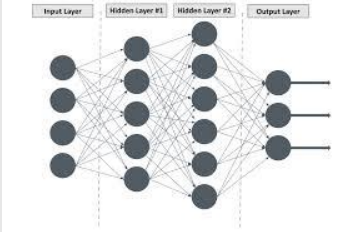
Modèle d'inférence : modèle qui va nous permettre de faire des prédictions

LES FAMILLES D'ALGORITHMES



LES FAMILLES D'ALGORITHMES



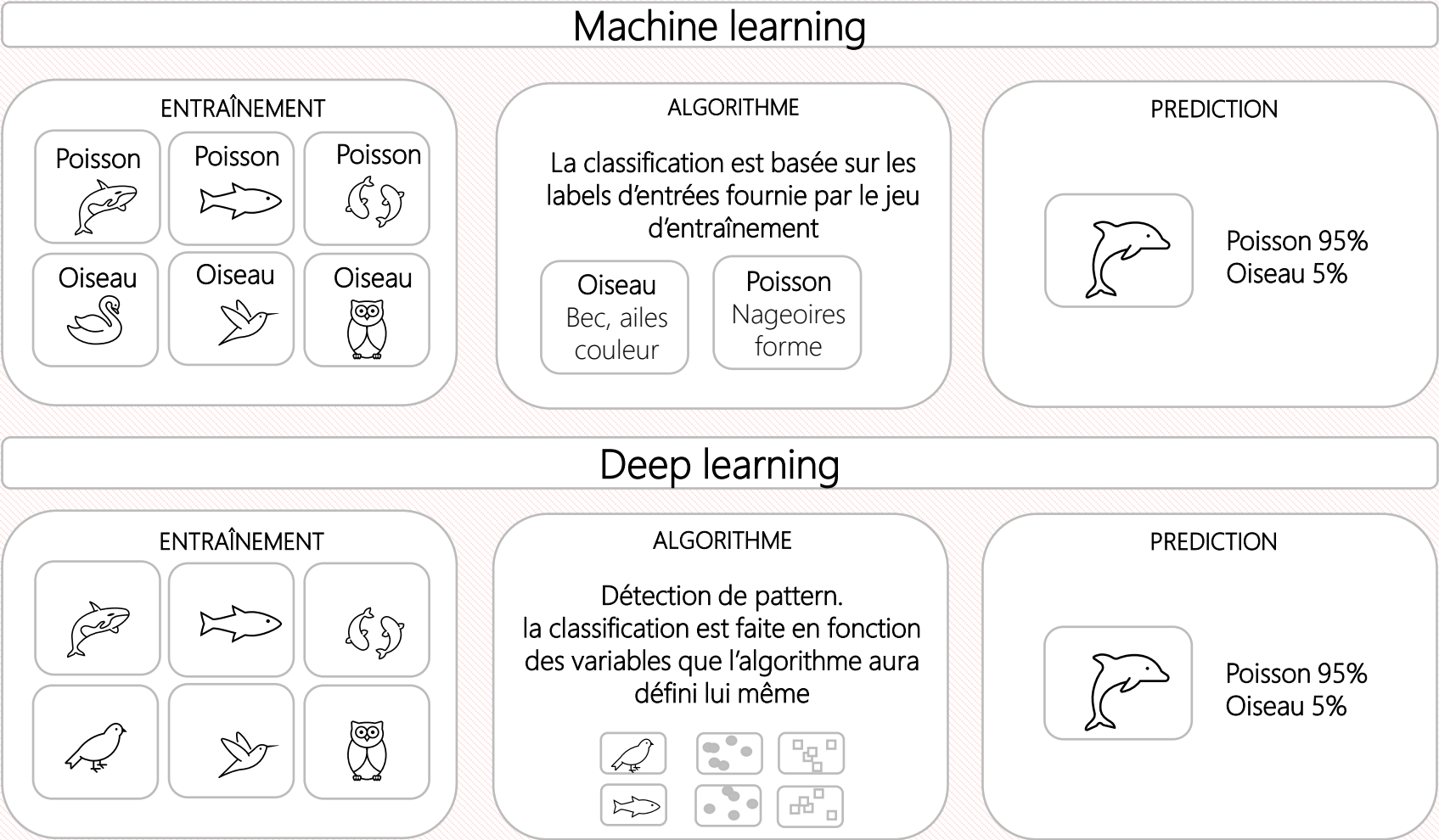
	ML supervisé	ML non supervisé	Transfert learning	Reinforcement learning	Deep learning
Types de problèmes	Régression / classification	Clustering / association / détection d'anomalie	modèles pré- entraînés	système de récompense Optimisation de processus	NLP reconnaissance d'image et de vidéos
					

LES FAMILLES D'ALGORITHMES

	ML supervisé	ML non supervisé	Transfert learning	Reinforcement learning	Deep learning
Approche	Le modèle est entraîné sur des données labellisées où il apprend à faire des prédictions en se basant sur les correspondances entre les entrées et les labels.	Le modèle cherche des structures ou des modèles dans les données sans être guidé par des labellisations préexistantes.	Transfert des connaissances d'un modèle pré-entraîné à une tâche similaire pour réduire le besoin de données d'entraînement et de temps de calcul.	L'agent apprend en interagissant avec un environnement. Mise en place d'un système de récompense pour réaliser les bonnes actions	Utilise des réseaux de neurones profonds pour réaliser des prédictions
Types de données	données labellisées	données non labellisées	données d'images ou de texte.	Pas de données fournies	données de grande dimensionnalité comme des images, du texte ou des séquences temporelles.

LES FAMILLES D'ALGORITHMES

MACHINE LEARNING VS DEEP LEARNING



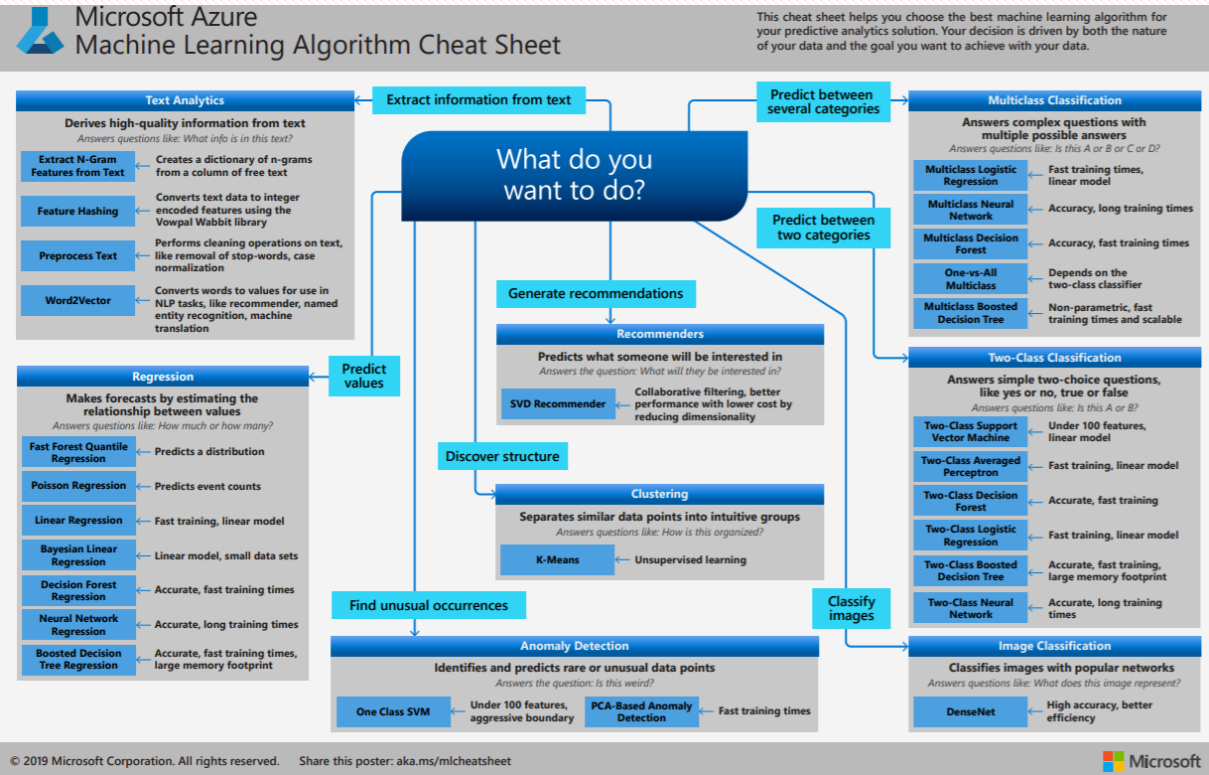
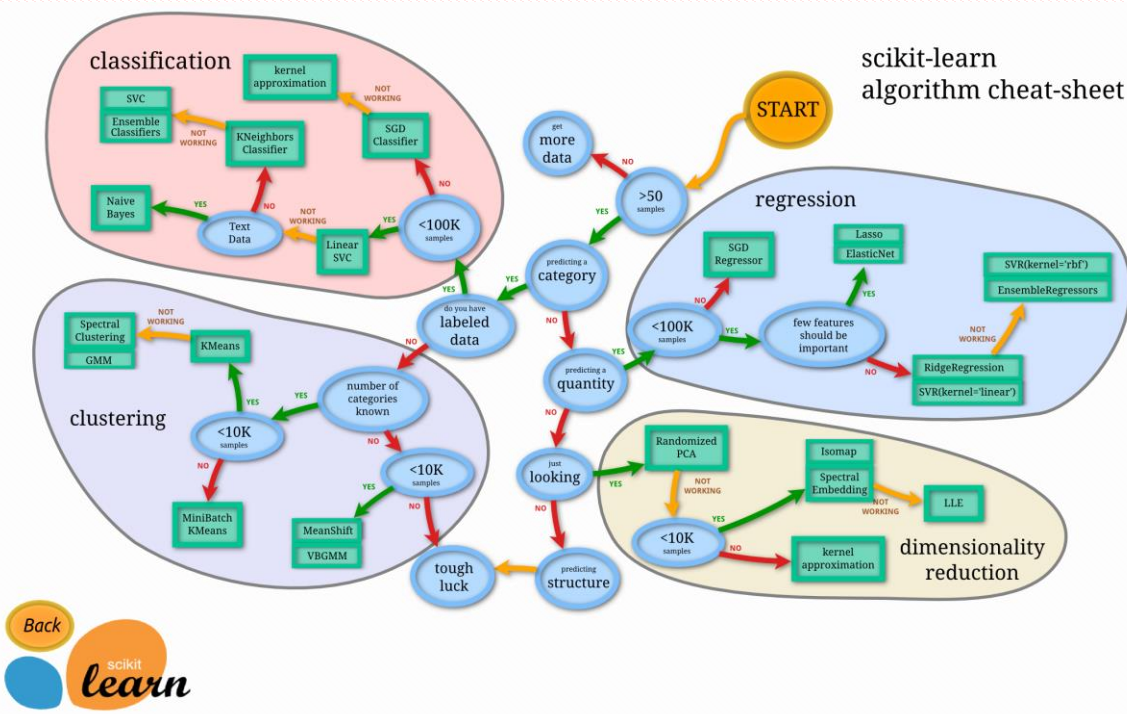
LES FAMILLES D'ALGORITHMES

MACHINE LEARNING VS DEEP LEARNING

	Machine learning	Deep learning
Nombre de données en entrées	Peut utiliser de petites quantités de données pour faire des prédictions	A besoin de grandes quantités de données
Temps d'entraînement	Relativement faible	Long du aux nombreuses couches
Approche d'apprentissage	Le processus d'apprentissage est fait étape par étape	Résolve le problème de bout en bout par rétroaction
Sorties attendues	Généralement une sortie numérique	Peut avoir différents formats :texte son ou image
fonctionnement	Les données peuvent être labelisées et les variables définies	C'est l'algorithme qui définit les patterns et trouvent des variables

LES FAMILLES D'ALGORITHMES

QUEL ALGORITHME CHOISIR ET POUR QUELS TYPES DE PROBLÉMATIQUES



[Choosing the right estimator — scikit-learn 1.1.1 documentation](#)

[Machine Learning Algorithm Cheat Sheet - designer - Azure Machine Learning | Microsoft Docs](#)

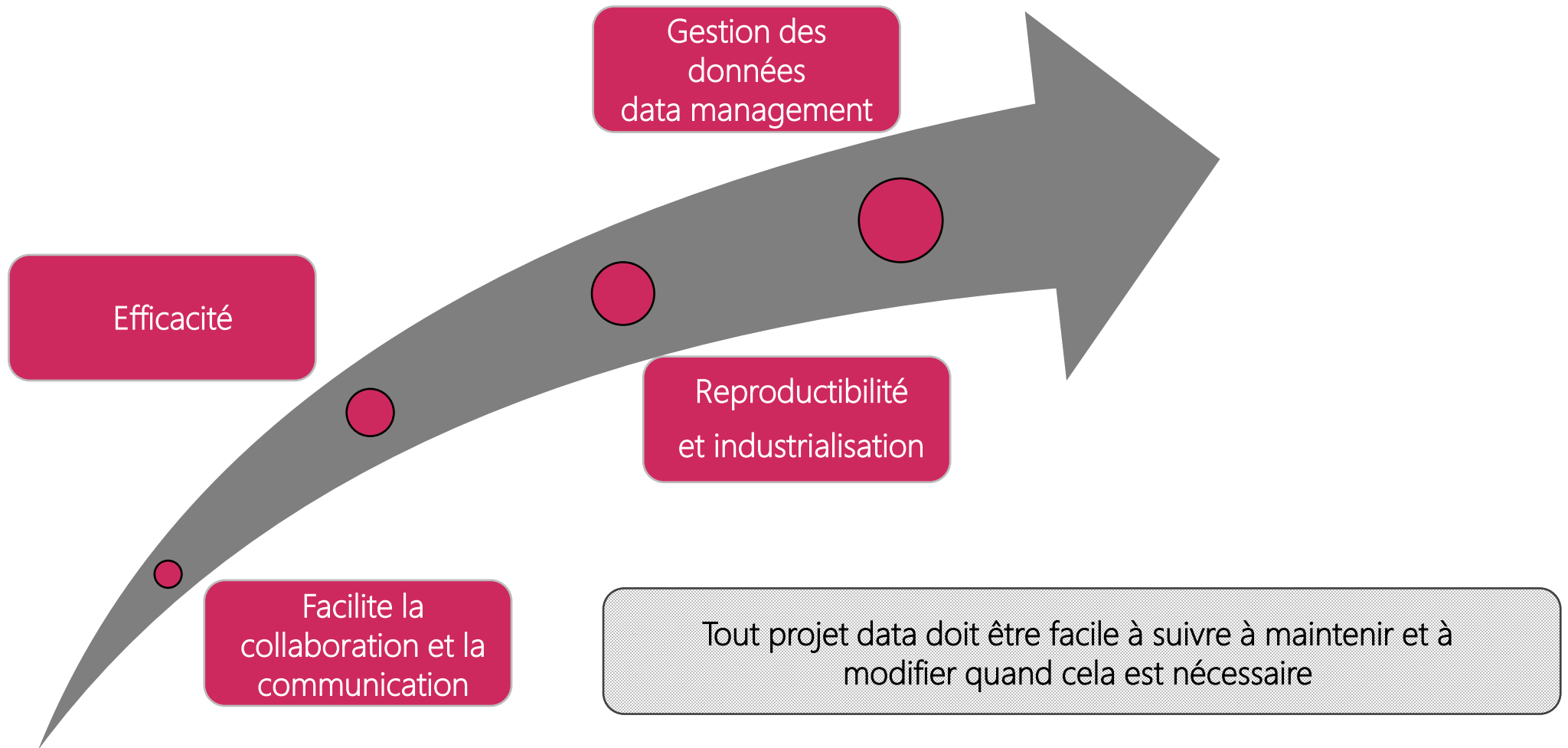


ZOIDBERG 2.0

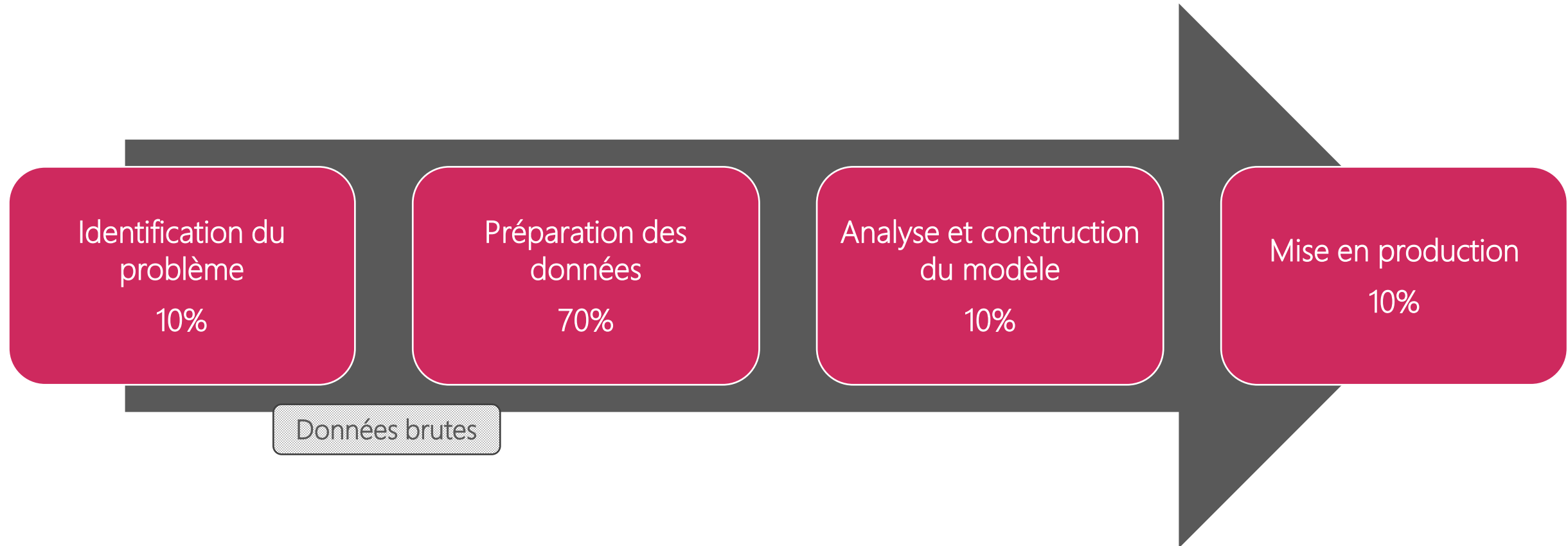
BOOTSTRAP

1. Introduction au Machine Learning
2. Réaliser un projet de machine learning
3. Réduction de dimension
4. Les algorithmes de classification
5. Évaluation du modèle
6. Optimisation des hyperparamètres

À QUOI SERT DE BIEN STRUCTURER SON PROJET



CYCLE DE DÉVELOPPEMENT D'UN PROJET IA



CYCLE DE DÉVELOPPEMENT D'UN PROJET IA



Comprendre les enjeux du problème



Interroger toutes les parties prenantes



Faire l'inventaire des ressources disponibles



Collecter les données nécessaires manquantes



CYCLE DE DÉVELOPPEMENT D'UN PROJET IA



Description du jeu de données



Nettoyage ou data cleaning



Encodage des données




Premières visualisations filtrage des variables ,
transformations et si nécessaire création de
nouvelles



DESCRIPTION DES DONNEES

Décrire son jeu de donnée consiste à savoir :

- 
- La dimension de nos données
 - Le nom et la signification des variables
 - Le type de variables
 - La présence de valeurs manquantes ou nulles
 - Le nombre de valeurs uniques

DESCRIPTION DES DONNEES

Décrire son jeu de donnée consiste à savoir :



ENCODAGE DES DONNÉES

1. LE PRINCIPE

L'objectif : convertir les données qualitatives en quantitatives !

Encodage Ordinal

- Associe à chaque classe une valeur unique (0, ...n classe-1)
- `OrdinalEncoder()` → s'applique à plusieurs variables , les variables explicatives ou **features**
- `LabelEncoder()` → s'applique à une seule variable, la variable cible ou **target**
- Attention ! La plupart des variables n'ont pas d'ordre et cela risque de poser un problème pour certains algorithmes de ML

Encodage One Hot

- chaque catégorie est représentée par un vecteur binaire unique ex : [001]
- Chaque classe à un poids équivalent
- `OneHotEncoder()` → s'applique à plusieurs variables , les variables explicatives ou **features**
- `LabelBinarizer()` → s'applique à une seule variable, la variable cible ou **target**

ENCODAGE DES DONNÉES

2. UN EXEMPLE

Encodage Ordinal

Petit	→	0
Petit		0
Moyen	→	1
Grand	→	2
Moyen		1

=

0
0
1
2
1

Encodage One Hot

Pomme	Pomme	Banane	Poire
1	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
0	1	0	0

ENCODAGE DES DONNÉES

3. ENCODAGE GET_DUMMIES

Encodage One Hot

Pomme

Pomme

Banane

Poire

Banane

=

Pomme

Banane

Poire

1 0 0

1 0 0

0 1 0

0 0 1

0 1 0

Encodage get dummies

Pomme

Pomme

Banane

Poire

Banane

=

1 0

1 0

0 1

0 0

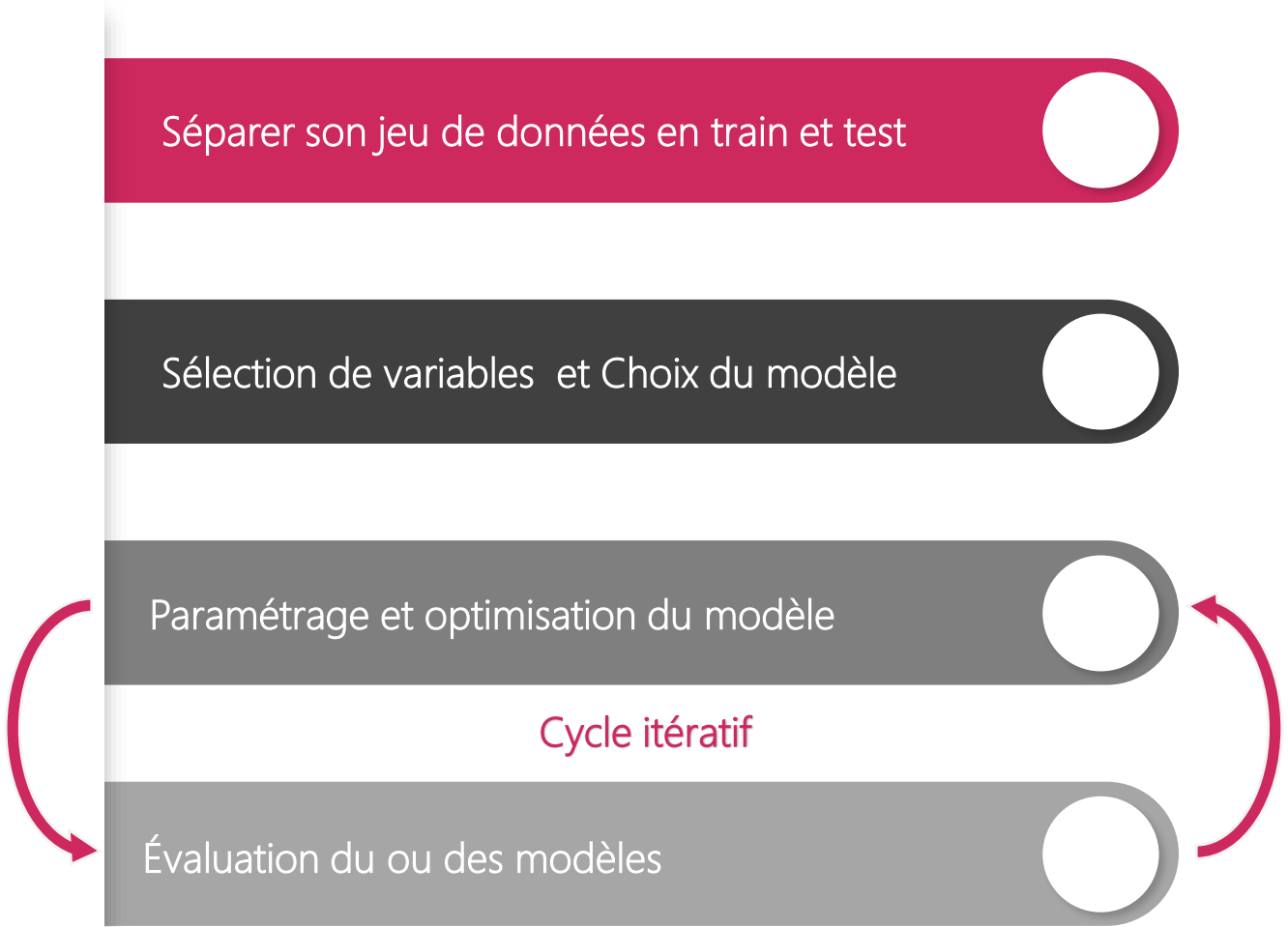
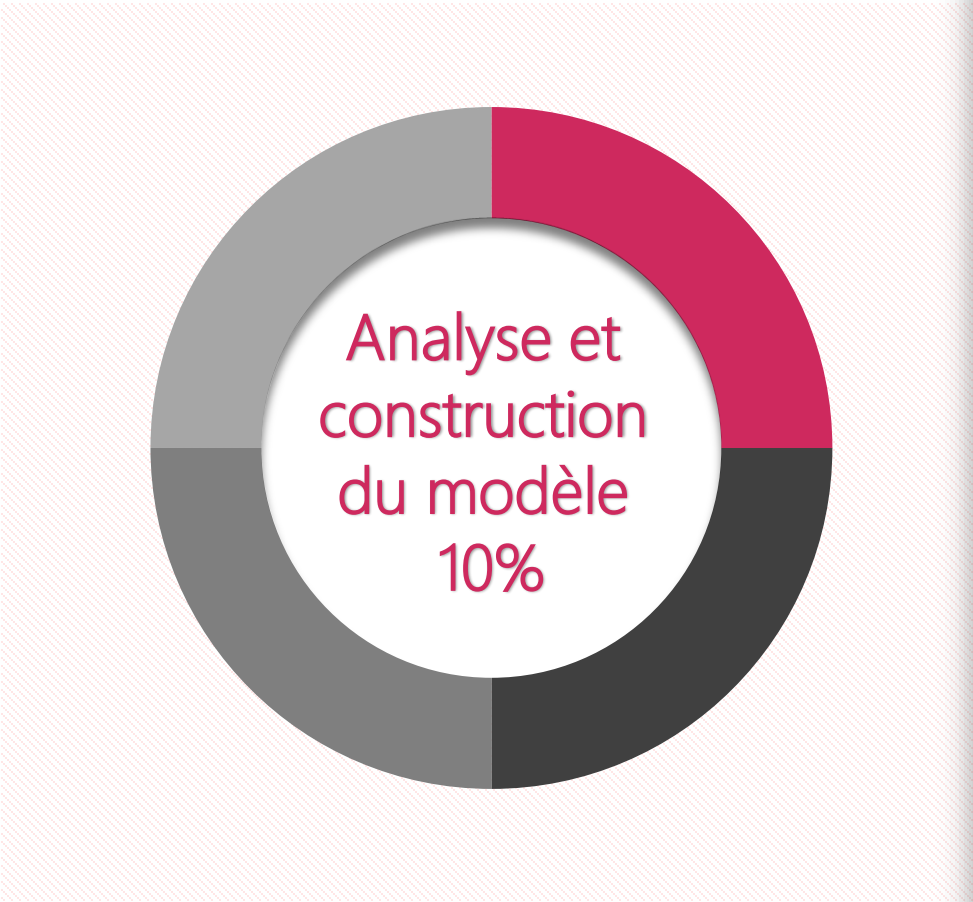
0 1

Pomme

Banane

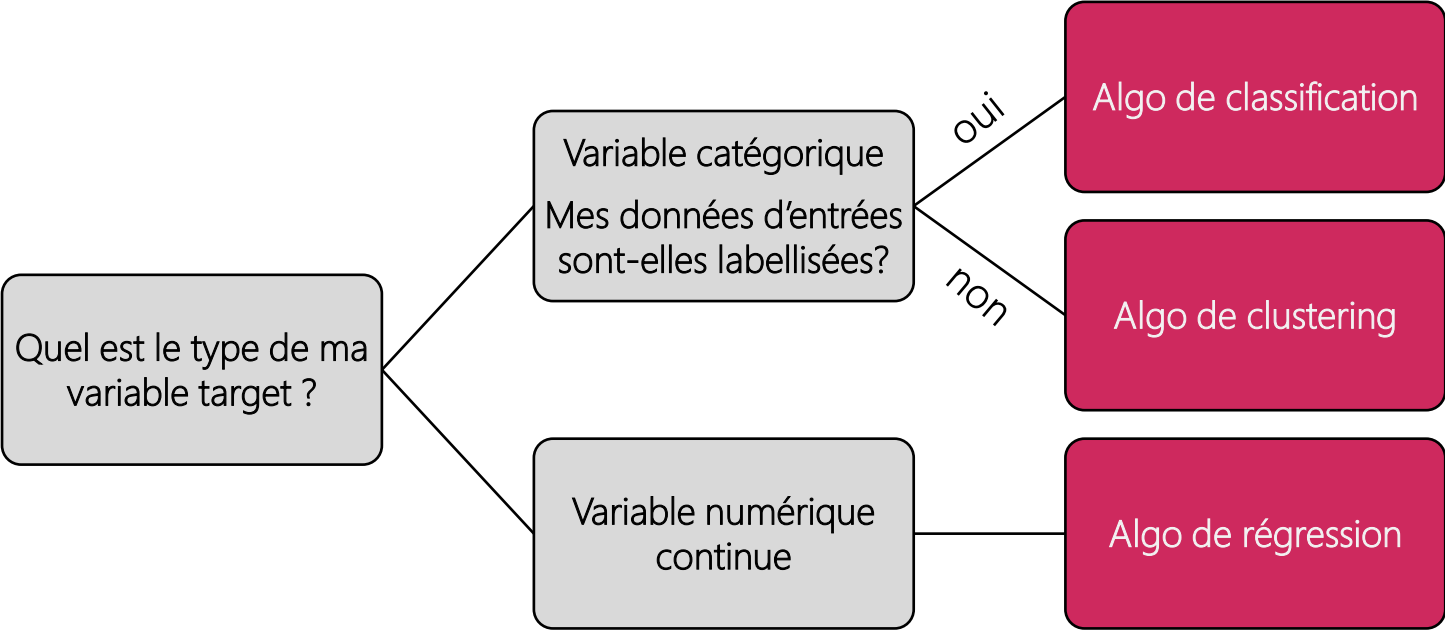
Poire

CYCLE DE DÉVELOPPEMENT D'UN PROJET IA

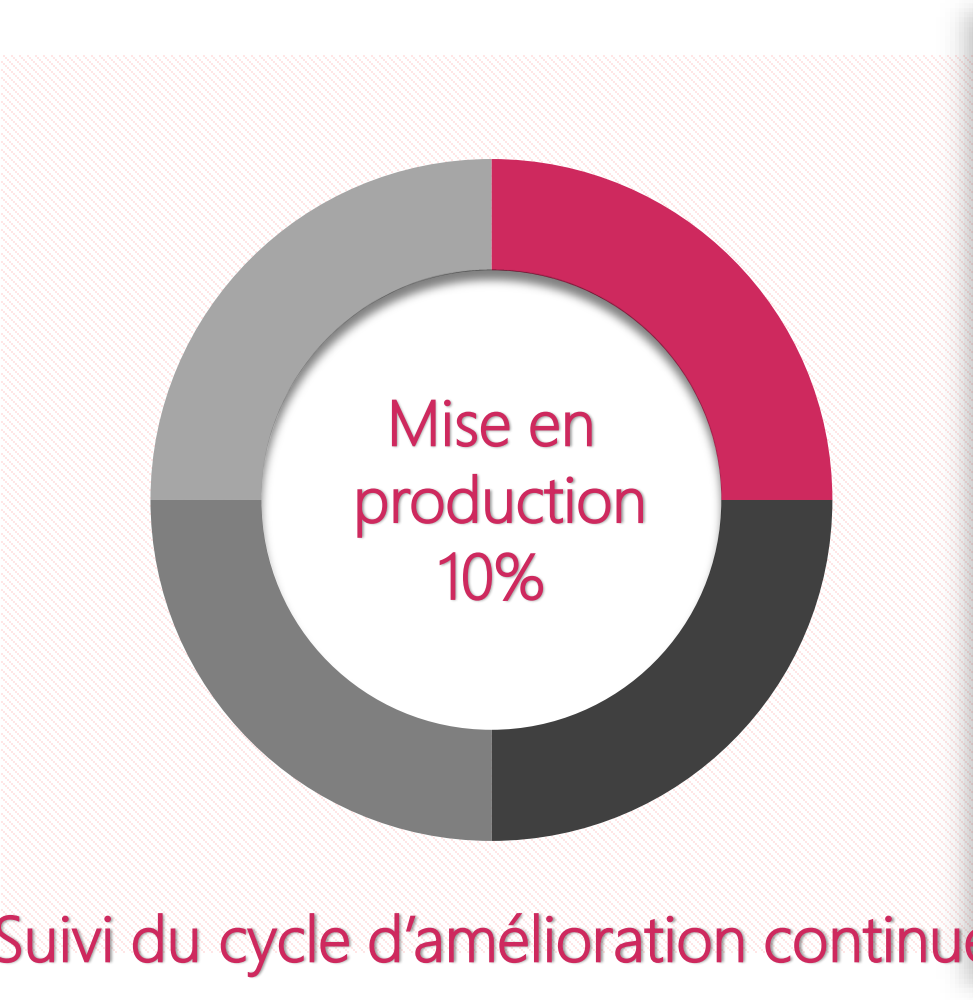


POUR CHOISIR SON ALGO

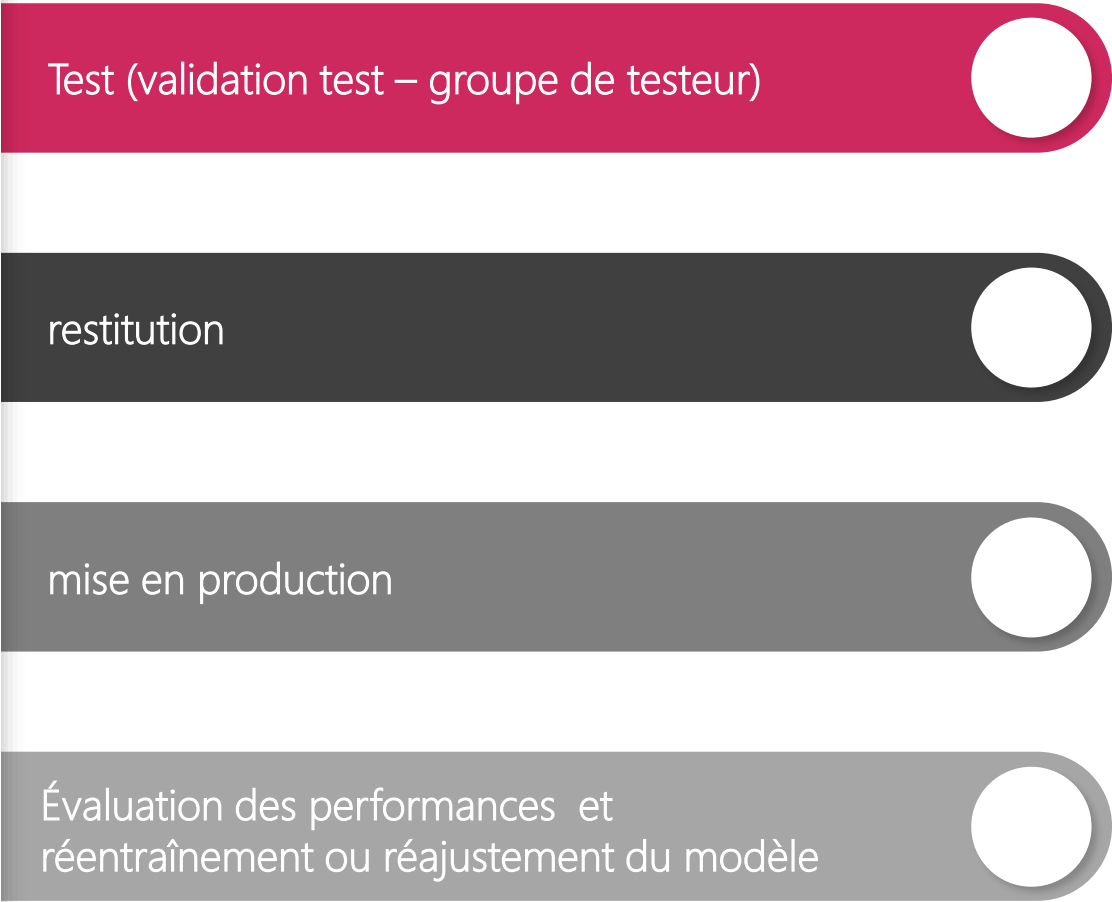
Déterminer le type de variable vous aide à choisir l'algorithme adéquat



CYCLE DE DÉVELOPPEMENT D'UN PROJET IA



Suivi du cycle d'amélioration continue

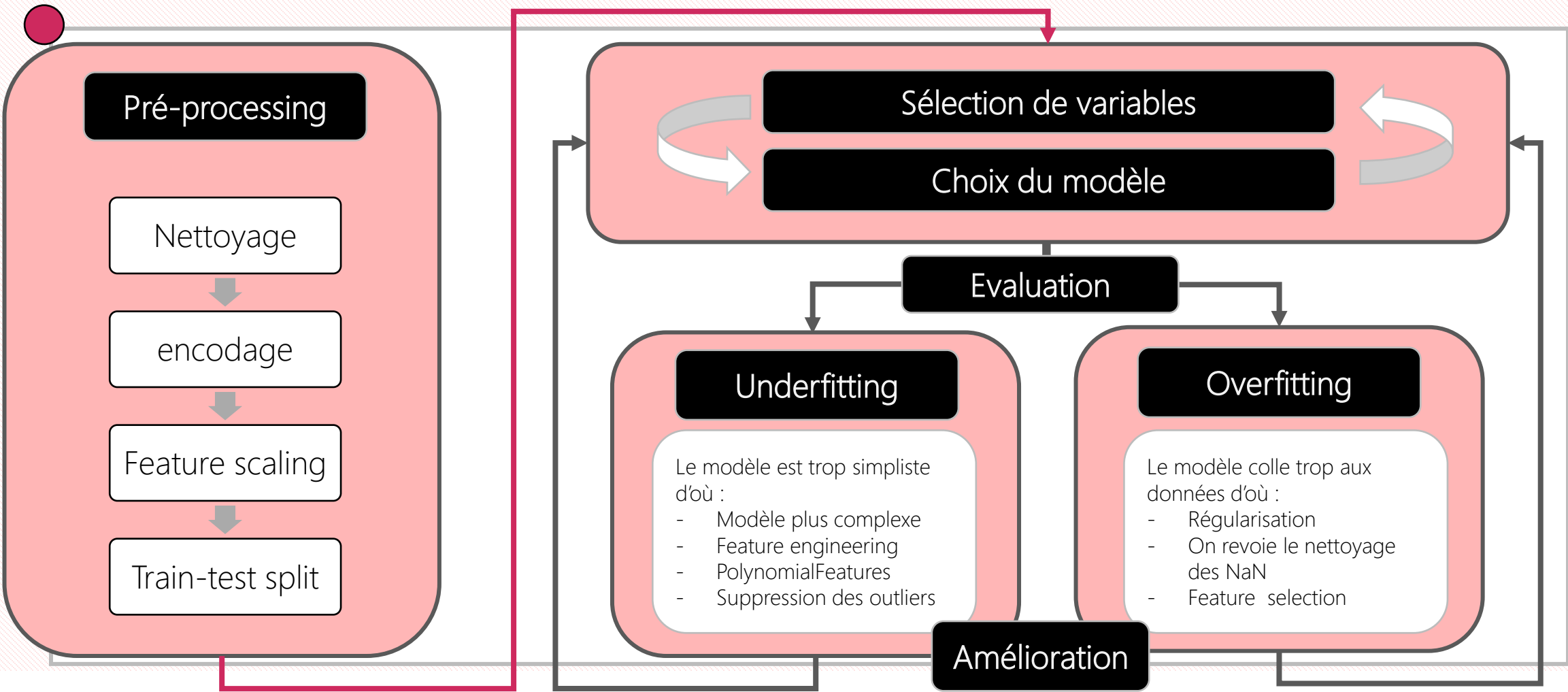


ZOIDBERG 2.0

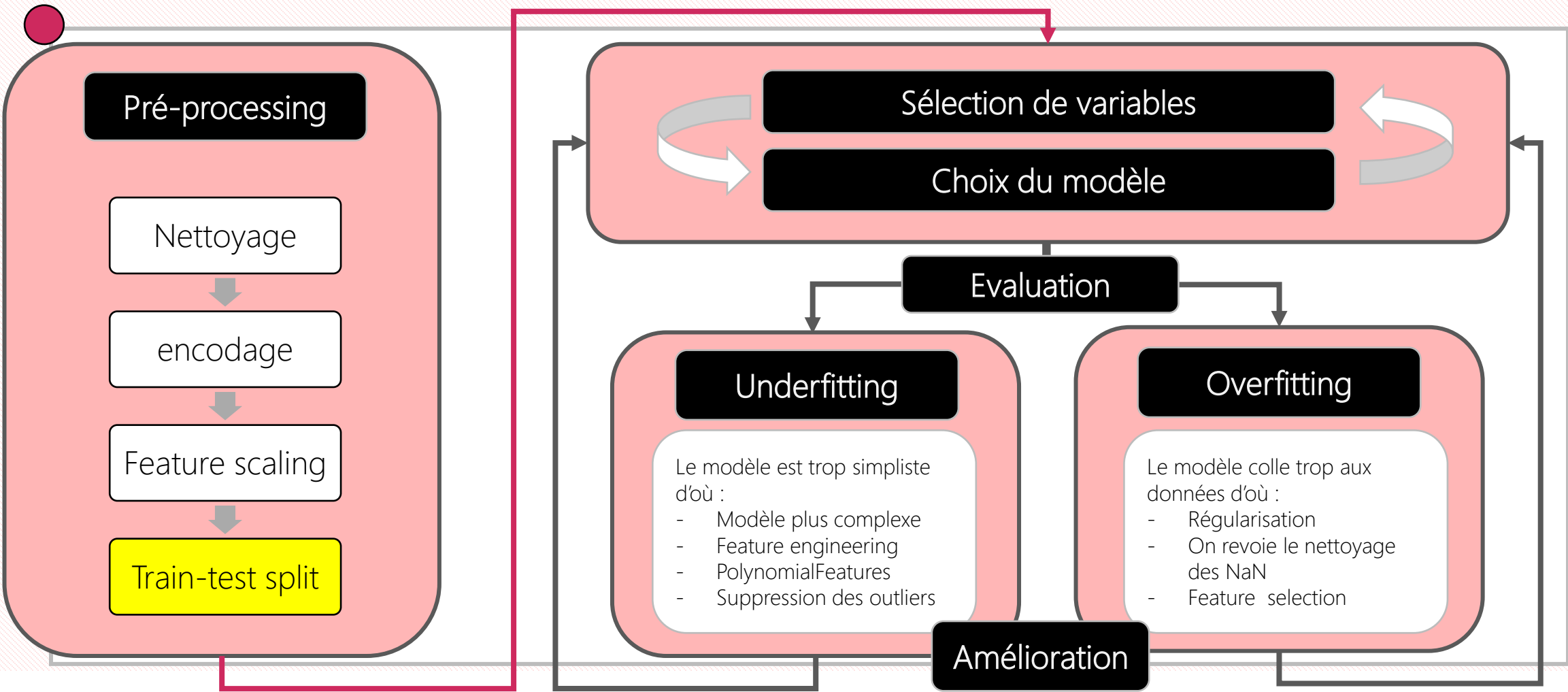
BOOTSTRAP

1. Introduction au Machine Learning
2. Réaliser un projet de machine learning
3. Modélisation
4. Réduction de dimension
5. Les algorithmes de classification
6. Évaluation du modèle
7. Optimisation des hyperparamètres

MODÉLISATION

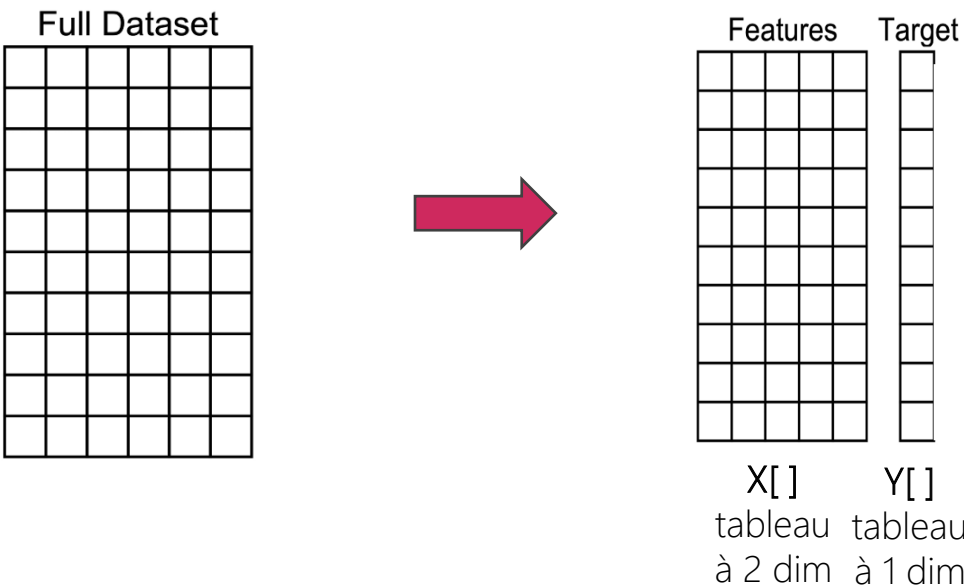


MODÉLISATION



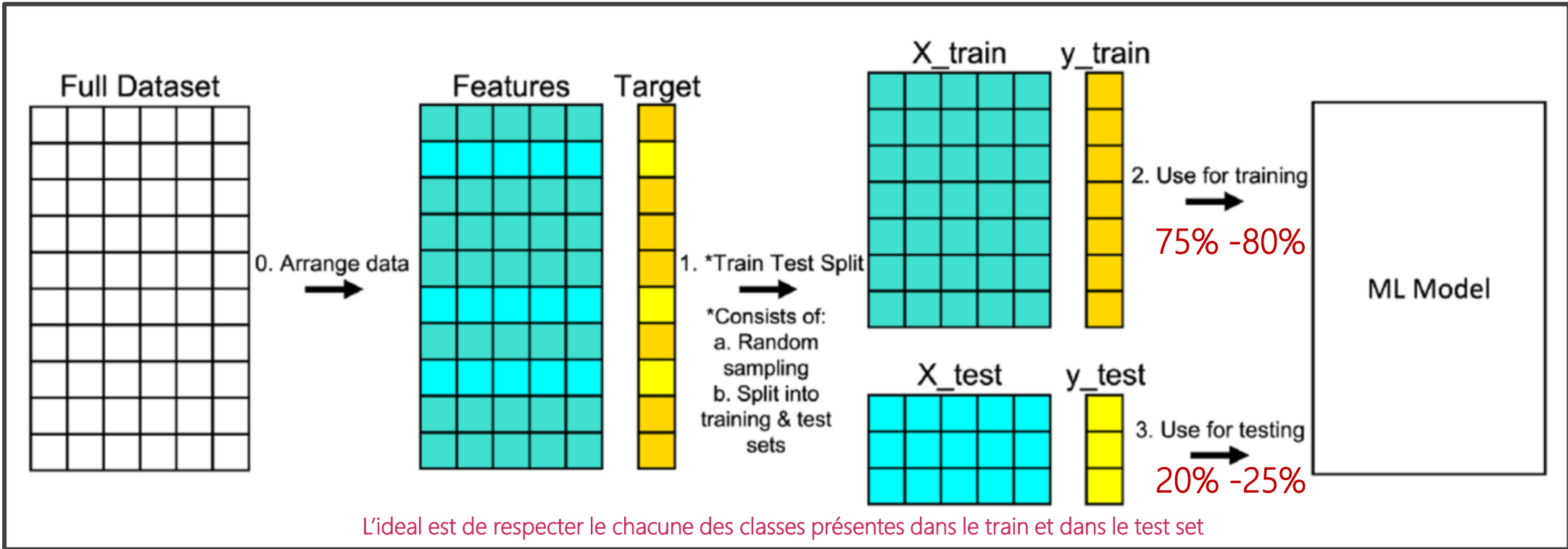
MODÉLISATION

on arrange son dataset de sorte à séparer les variables explicatives de la variable target

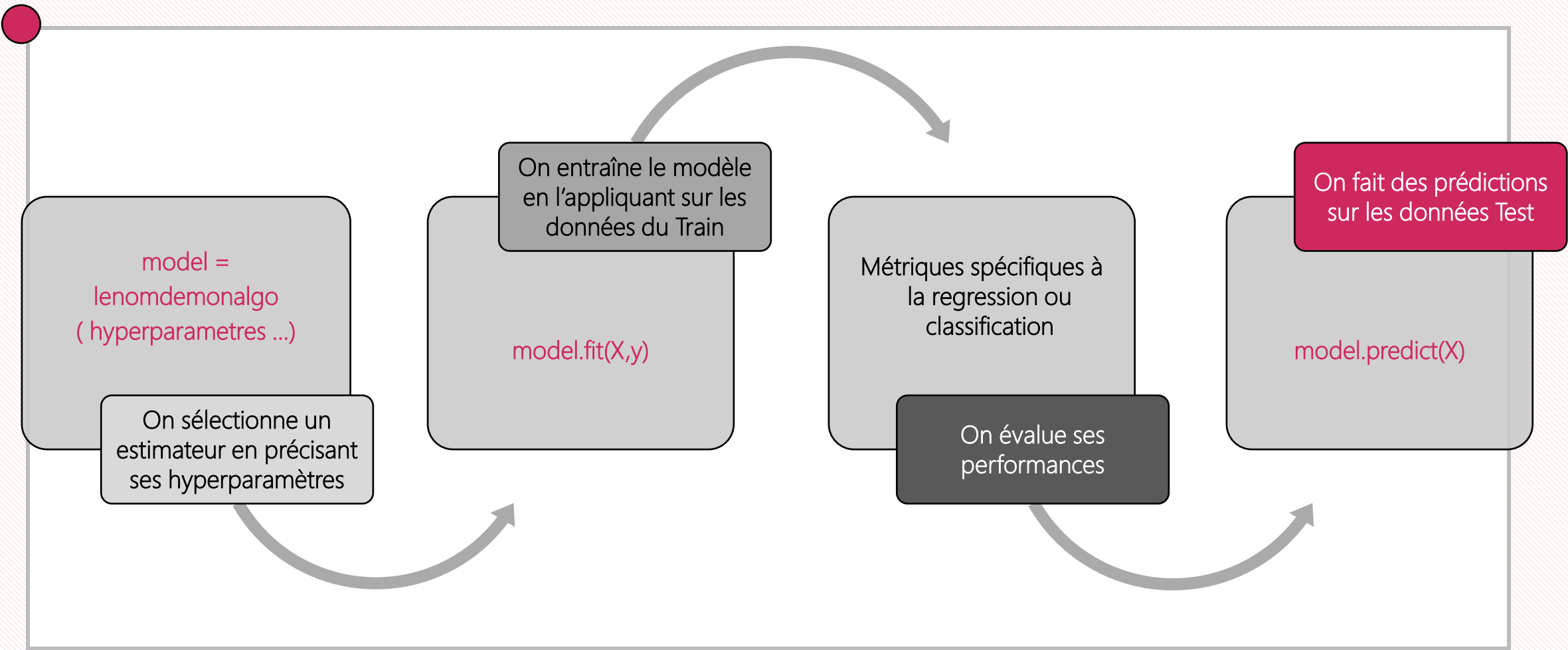


MODÉLISATION

! On ne peut pas tester son modèle sur les données avec lesquelles on l’a entraîné !
On sépare donc le dataset en trainset et test set



MODÉLISATION



ZOIDBERG 2.0

BOOTSTRAP

1. Introduction au Machine Learning
2. Réaliser un projet de machine learning
3. Modélisation
4. Réduction de dimension
5. Les algorithmes de classification
6. Évaluation du modèle
7. Optimisation des hyperparamètres

CHOIX DES VARIABLES



Une image peut contenir des milliers, voire des millions de pixels. Chaque pixel représente une couleur ou une intensité lumineuse à un emplacement spécifique de l'image.

Vous risquez donc de vous retrouver avec un nombre très importants d'informations ou features

Ne pas faire de sélection / transformations de variables peut cependant grandement réduire les performances du modèle.

D'après vous pourquoi ?

CHOIX DES VARIABLES

Trop de variables :

- Introduit du bruit (informations non utiles) dans le modèle
- peut entraîner de l'overfitting
- Le modèle devient trop complexe et inexplicable : c'est ce qu'on appelle la black box
- Certaines variables peuvent être corrélées , redondantes
- Trop grand temps de calcul
- Peut introduire des biais

CHOIX DES VARIABLES

- Plus on a de features et plus l'algorithme aura besoin de données d'entraînement pour obtenir de bons résultats. Le manque de données nécessaires à l'apprentissage du modèle explose très vite, ce phénomène est appelé '**le fléau de la dimension**'.
- Pour modéliser en ML le comportement d'un ensemble d'observations il faut pouvoir généraliser les phénomènes observés et donc **réduire la complexité du phénomène**.

CHOIX DES VARIABLES

- Plus on a de features et plus l'algorithme aura besoin de données d'entraînement pour obtenir de bons résultats. Le manque de données nécessaires à l'apprentissage du modèle explose très vite, ce phénomène est appelé '**le fléau de la dimension**'.
- Pour modéliser en ML le comportement d'un ensemble d'observations il faut pouvoir généraliser les phénomènes observés et donc **réduire la complexité du phénomène**.

La réduction de dimension consiste à réduire la quantité d'informations contenues dans cette image tout en essayant de préserver les informations essentielles.

RÉDUCTION DE DIMENSIONS

Sélection de variables Feature selection

Méthode de filtrage

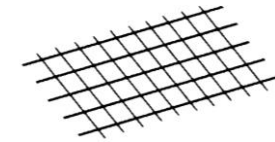
Méthode wrapped

Méthode embedded

Transformation de variables Feature extraction

Linéaire :
ACP AFD & ACM

Non linéaire :
T-SNE LLE Isomap KPCA



RÉDUCTION DE DIMENSIONS

Sélection de variables Feature selection

Méthode de filtrage

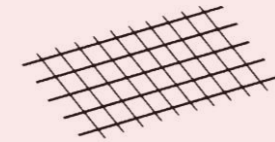
Méthode wrapped

Méthode embedded

Transformation de variables Feature extraction

Linéaire :

ACP AFD & ACM



Non linéaire :

T-SNE LLE Isomap KPCA



On construit de nouvelles variables à partir des variables initiales.
NB : les nouvelles variables construites sont rarement interprétables

RÉDUCTION DE DIMENSIONS

L'objectif général des méthodes d'analyse factorielle est la recherche de **facteurs** permettant de **résumer les données** ou leurs caractéristiques (la réduction de la redondance entre variables ou l'amélioration de la séparation entre classes d'observations)

RÉDUCTION DE DIMENSIONS

L'objectif général des méthodes d'analyse factorielle est la recherche de **facteurs** permettant de **résumer les données** ou leurs caractéristiques (la réduction de la redondance entre variables ou l'amélioration de la séparation entre classes d'observations)

3 méthodes factorielles linéaires :

- L'analyse en composantes principales ACP → adaptée à des données décrites par des variables quantitatives.
- L'analyse factorielle discriminante AFD → adaptée à des données décrites par des variables quantitatives et appartenant à plusieurs classes.
- L'analyse des correspondances multiples ACM → adaptée à des données décrites par des variables nominales.

RÉDUCTION DE DIMENSIONS

L'Analyse en Composantes Principales

C'est une méthode qui va permettre de construire un nouveau système de représentation en **synthétisant l'information**, malgré une perte de données qui reste tout de même contrôlée.

RÉDUCTION DE DIMENSIONS

L'Analyse en Composantes Principales

C'est une méthode qui va permettre de construire un nouveau système de représentation en **synthétisant l'information**, malgré une perte de données qui reste tout de même contrôlée.

Pour ce faire on projette les données sur des axes appelés **composantes principales**, dans un espace de plus petite dimension , en cherchant à **minimiser la distance entre nos points et leur projection**. Ces axes sont une combinaison linéaire des variables initiales.

on réduit ainsi la dimension de notre dataset tout en **préservant le maximum de variance** de nos données.

RÉDUCTION DE DIMENSIONS

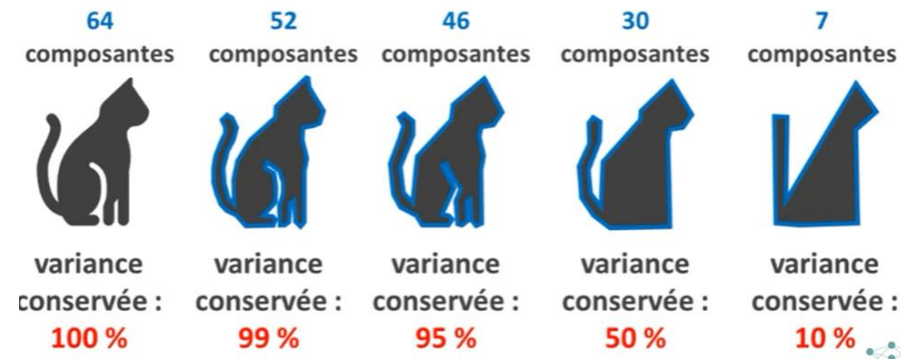
Quelques remarques :

- Les résultats obtenus sont en général des approximations de ceux obtenus sur les données complètes. **Il y a donc toujours une perte d'information** ! Mais celle-ci reste contrôlée en choisissant le bon nombre de K composantes principales préservant une variance suffisante .
- L'ACP peut être mise en échec si les données sont à faible densité d'informations :
 - Soit parce que l'échantillon de données est trop petit
 - Soit parce que la réduction de dimension appliquée est trop forte

RÉDUCTION DE DIMENSIONS

Quelques remarques :

- Les résultats obtenus sont en général des approximations de ceux obtenus sur les données complètes. Il y a donc **toujours une perte d'information** ! Mais celle-ci reste contrôlée en choisissant le bon nombre de K composantes principales préservant une variance suffisante .
- L'ACP peut être mise en échec si les données sont à faible densité d'informations :
 - Soit parce que l'échantillon de données est trop petit
 - Soit parce que la réduction de dimension appliquée est trop forte

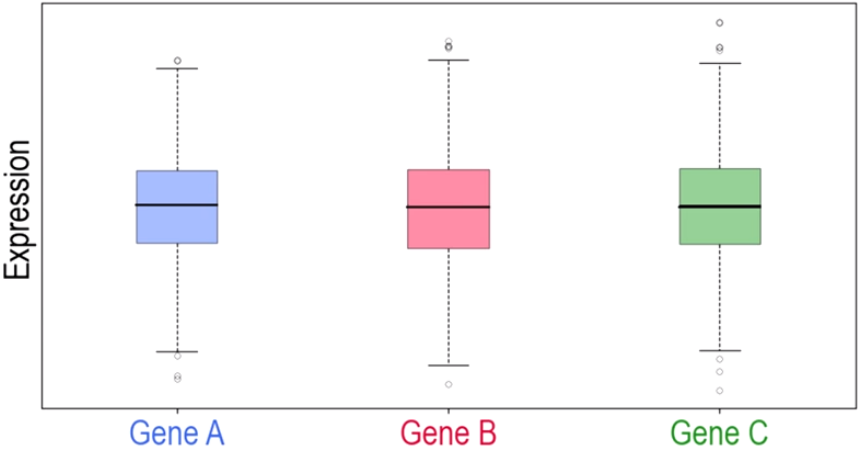


RÉDUCTION DE DIMENSIONS

Prenons un exemple :

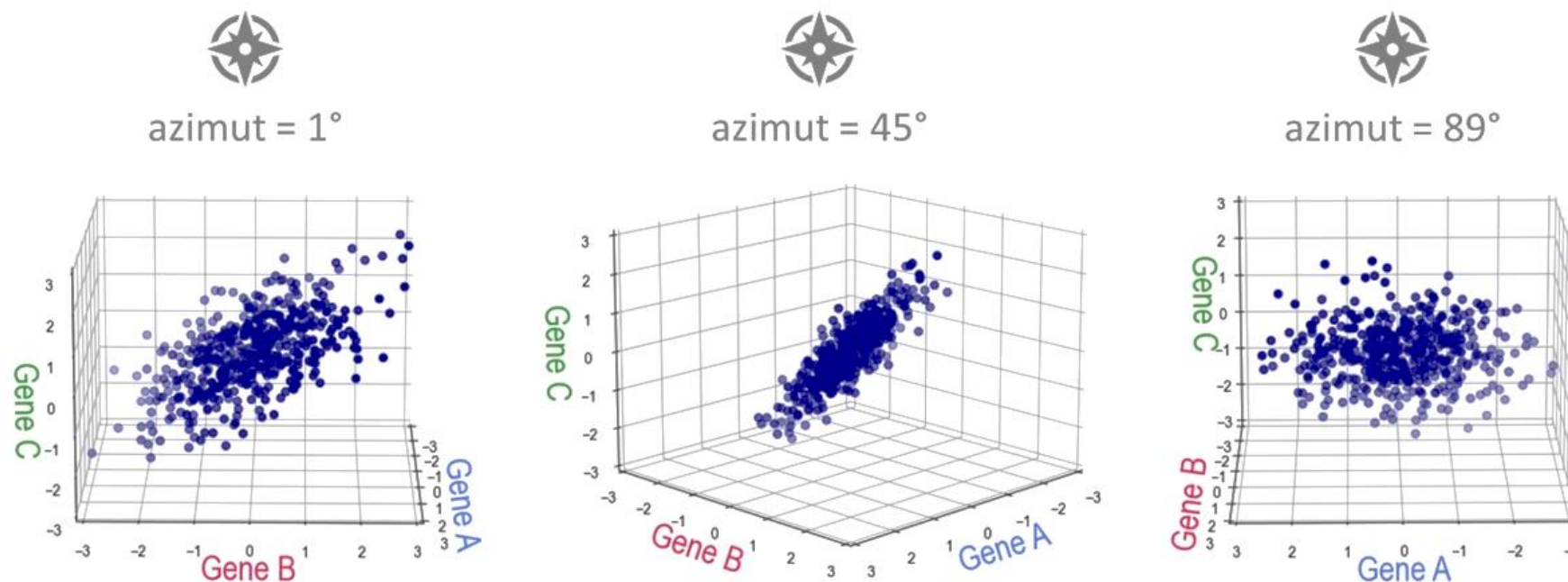
Données d'expression de gènes
pour un échantillon de 500
individus

échantillon	Gène A	Gène B	Gène C
001	-0.61	0.71	1.61
003	-0.15	-0.025	0.474
⋮			
500	-1.43	0.09	-1.028



RÉDUCTION DE DIMENSIONS

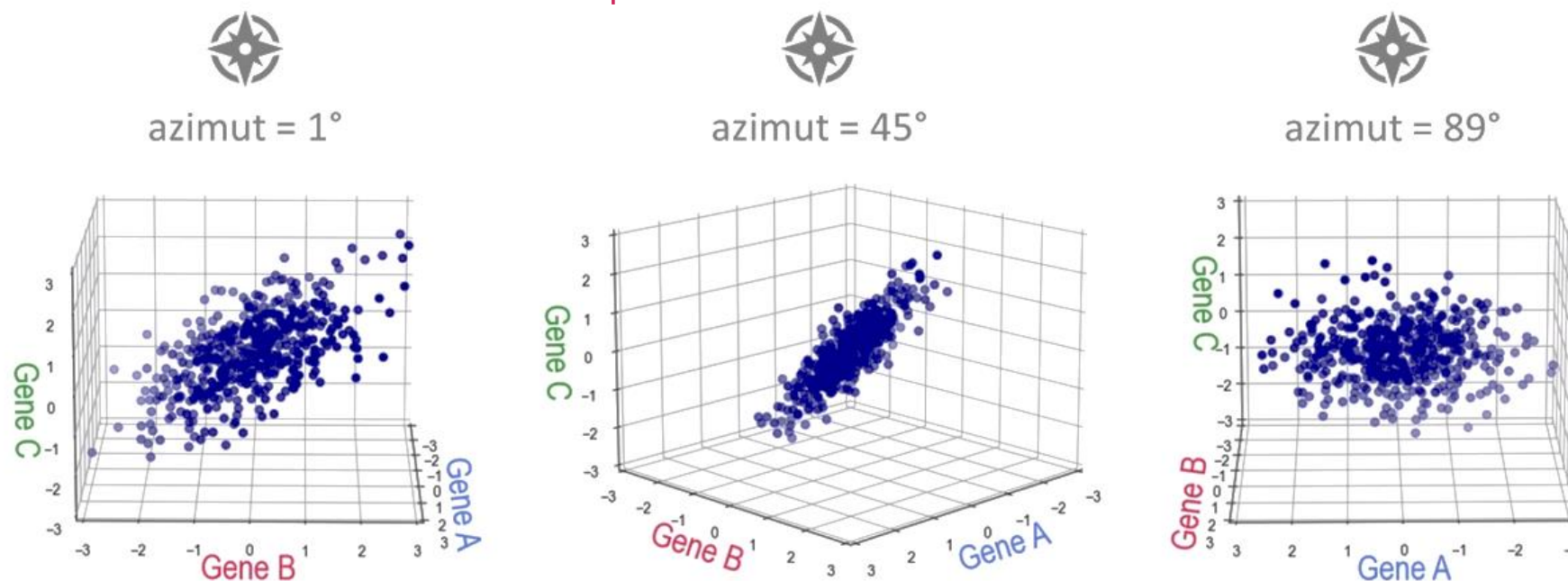
Les données ne sont pas distribuées de façon aléatoire , elles se situent dans un plan laissant pour le reste de l'espace vide



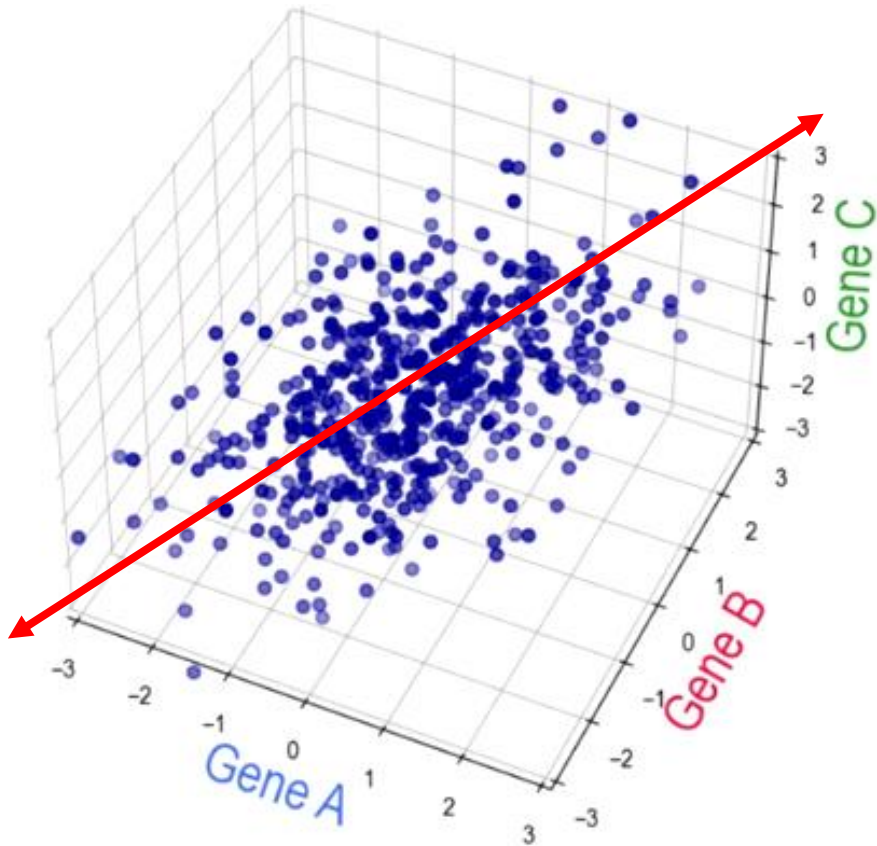
RÉDUCTION DE DIMENSIONS

Rq : on visualise mieux les données sous certains angles

L'ACP va nous permettre de trouver la projection pour laquelle l'orientation de la représentation des données permet au mieux de visualiser la variance



RÉDUCTION DE DIMENSIONS

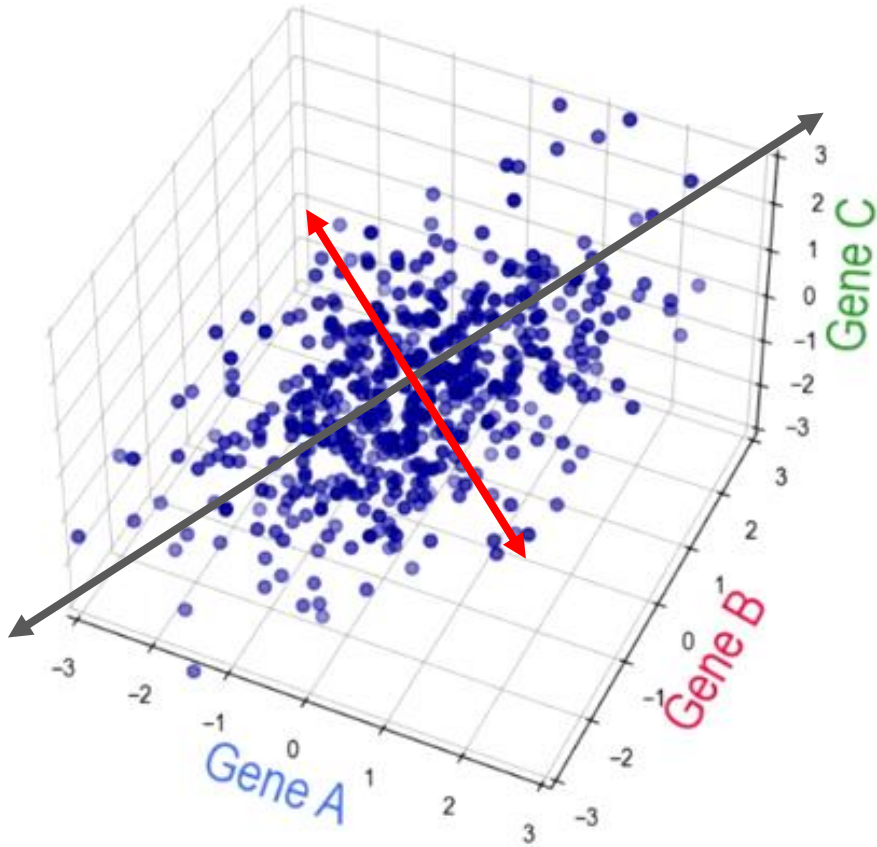


L'ACP identifie l'axe selon lequel la variance des données est maximale

Cet axe est appelé : **1^{ère} composante principale**

Selon cet axe, les données varient le plus
C'est également sur cette droite que l'écart entre les données et leur projection est le plus faible

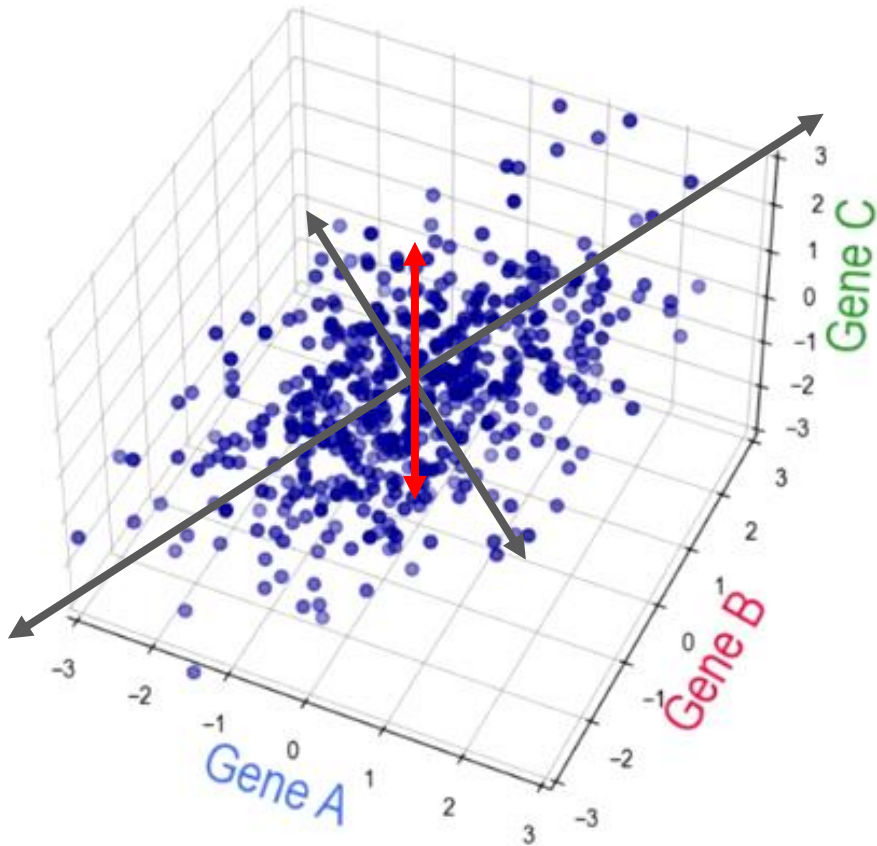
RÉDUCTION DE DIMENSIONS



L'ACP identifie le 2nd axe selon lequel la variance des données est maximale parmi les directions restantes.

Ce 2nd axe **est perpendiculaire au premier**. Il est appelé : 2^{ème} composante principale

RÉDUCTION DE DIMENSIONS

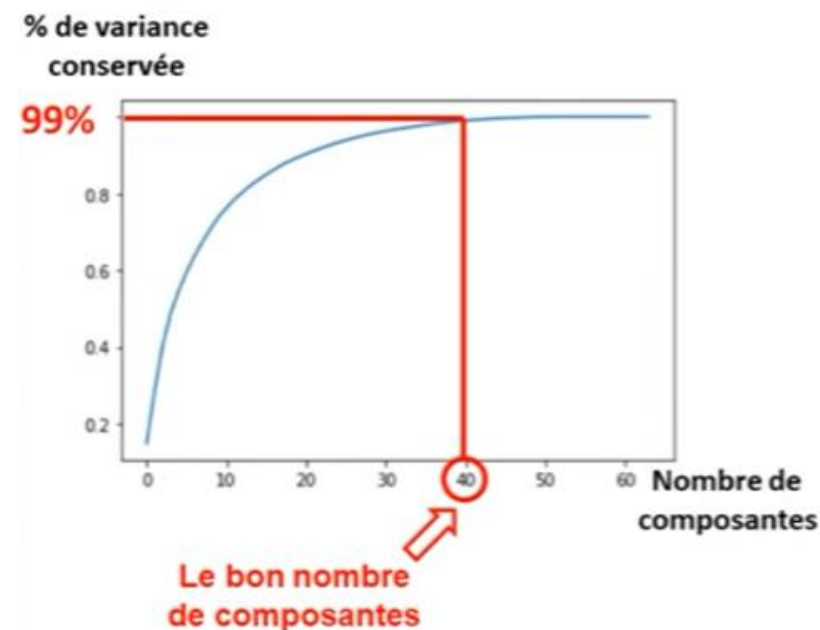


Et ainsi de suite, l'ACP va identifier autant de composantes principales que de variables dans nos données. Dans cet exemple nous avons 3 variables donc 3 composantes principales.

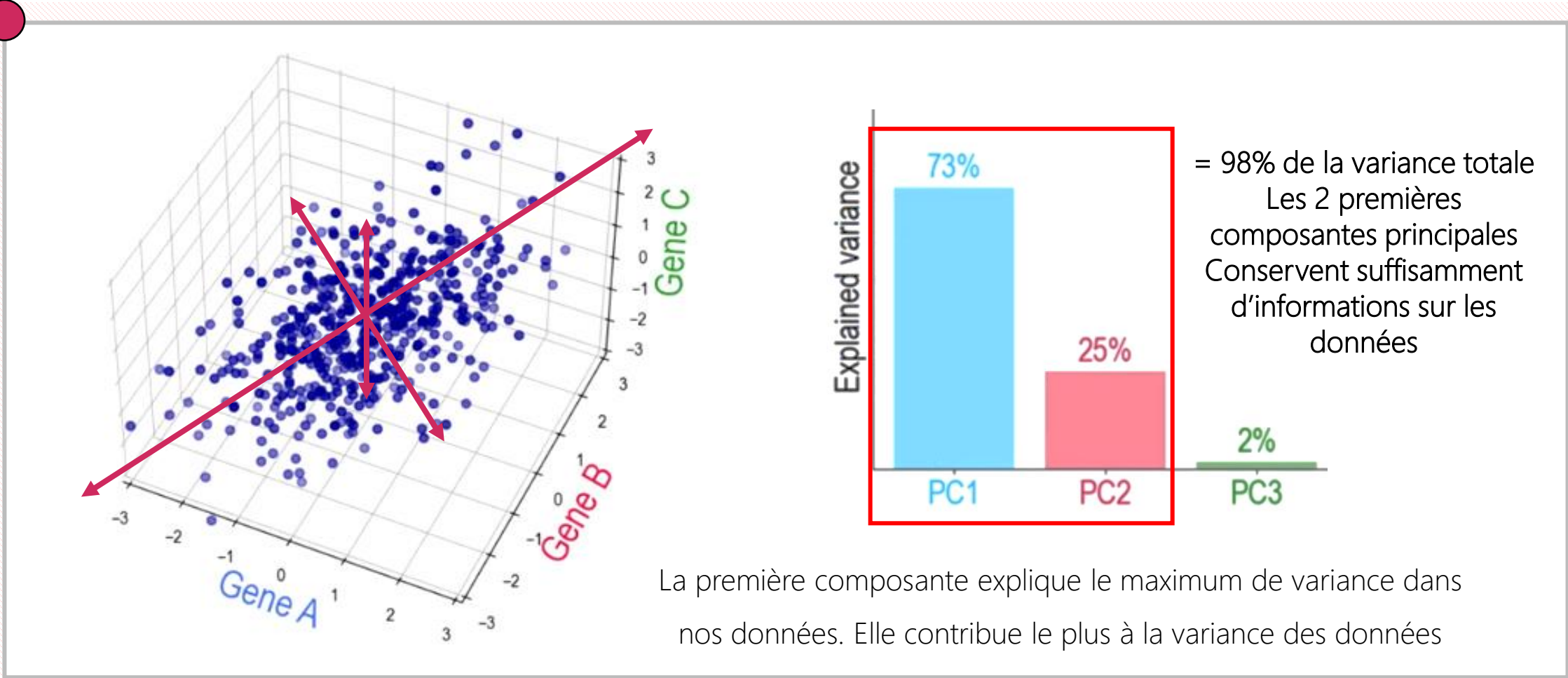
RÉDUCTION DE DIMENSIONS

Combien d'axes choisir ?

- On peut choisir un certain seuil de variance : généralement entre 95% et 98%
- On peut également considérer le nombre d'axes de l'ACP comme un paramètre à faire varier en utilisant gridsearchCV pour sélectionner un algorithme



RÉDUCTION DE DIMENSIONS



RÉDUCTION DE DIMENSIONS

```
# Import du module :  
from sklearn.decomposition import PCA  
  
# On définit l'estimateur :  
model = PCA(n_components = nombre de composantes principales souhaités )  
  
# On applique aux données:  
Model.fit_transform(X)  
  
# quelques méthodes et attributs utiles  
model.components_           # renvoie la combinaison linéaire de toutes les variables contribuant l'axe  
model.explained_variance_ratio_ # renvoie le pourcentage de variance préservée par chaque variable  
model.inverse_transform (X)   # pour retrouver les données initiales (attention , on retrouve uniquement les  
                               projections)
```