



# Intelligence Artificielle - Machine Learning

IMIE

Session 2 – du 16 au 19 avril 2019



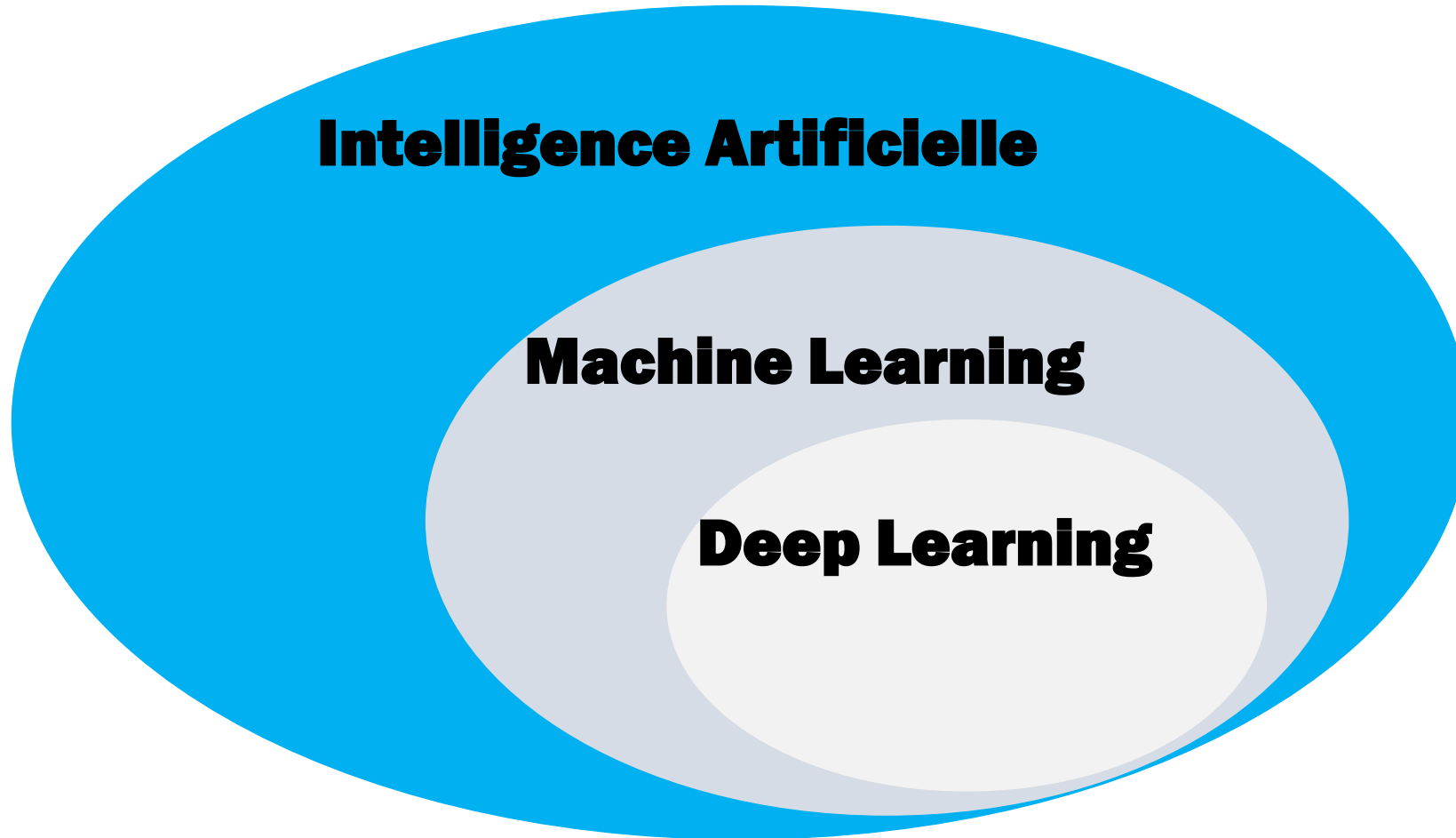
# Programme

1. Introduction à l'IA : familiarisation avec les notions et les intérêts de ces méthodes
2. Mener un projet Data impliquant l'utilisation du Machine Learning : déroulement du projet
3. **Focus sur les méthodes courantes de Machine Learning**
  - a) Apprentissage Non Supervisé
  - b) Apprentissage Supervisé
  - c) Introduction au Natural Language Processing / Text-mining
  - d) Introduction aux Réseaux de Neurones
4. **Mener un projet Data (Machine Learning) : les éléments essentiels à la réussite du projet**
  - a) Combien de temps pour un sujet de Machine Learning ?
  - b) Sources d'erreurs les plus fréquentes d'un projet ML : connaître les biais possibles
  - c) Quelques réglementations autour de l'utilisation des données : RGPD & CNIL
  - d) Dérouler les facteurs clés d'une stratégie Data
5. **Evaluation**

# RÉSUMÉ SESSION 1

*“The goal is to turn data into information,  
and information into insight.”*  
**Carly Fiorina**

# Intelligence Artificielle > Machine Learning > Deep Learning

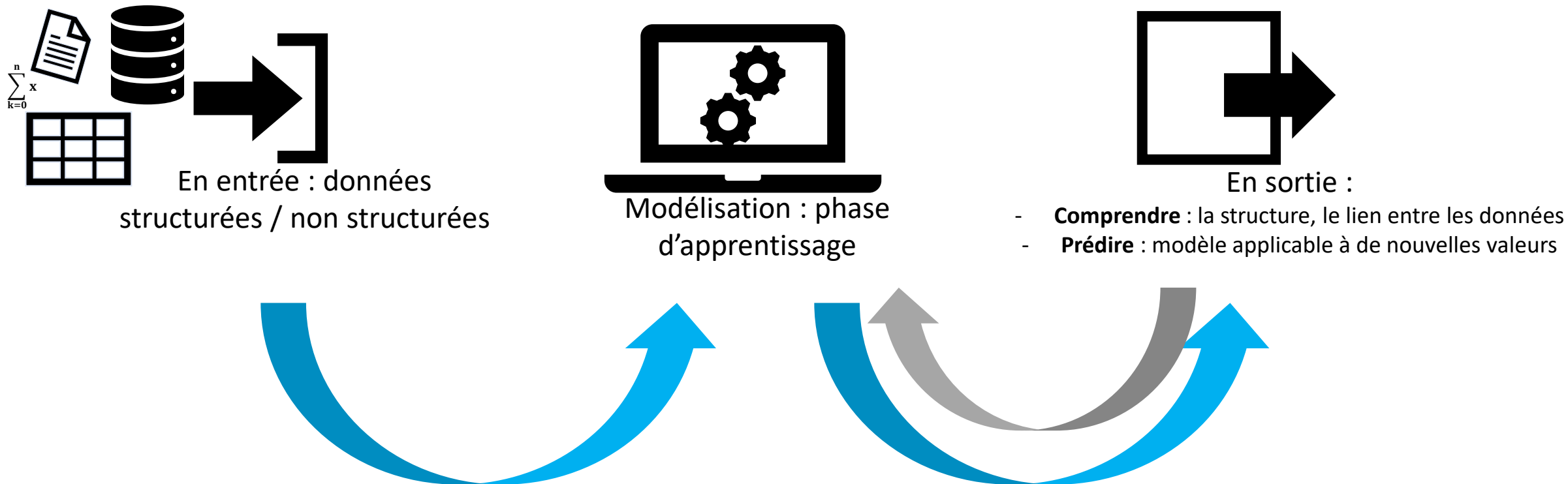




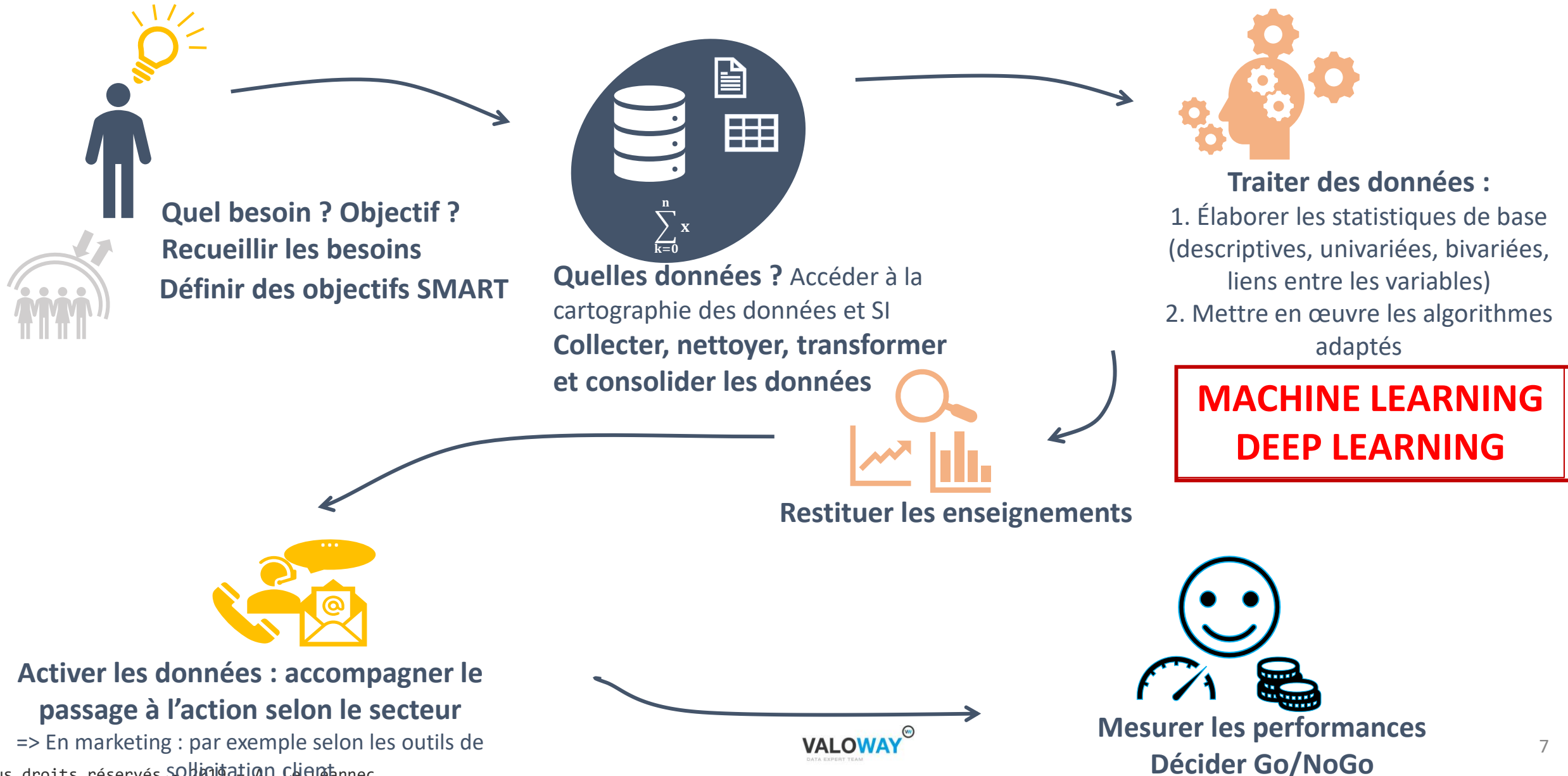
**Enjeu : Résoudre le problème soumis**

Apporter la bonne solution avec les  
moyens et outils adaptés

# Machine Learning : apprentissage automatique

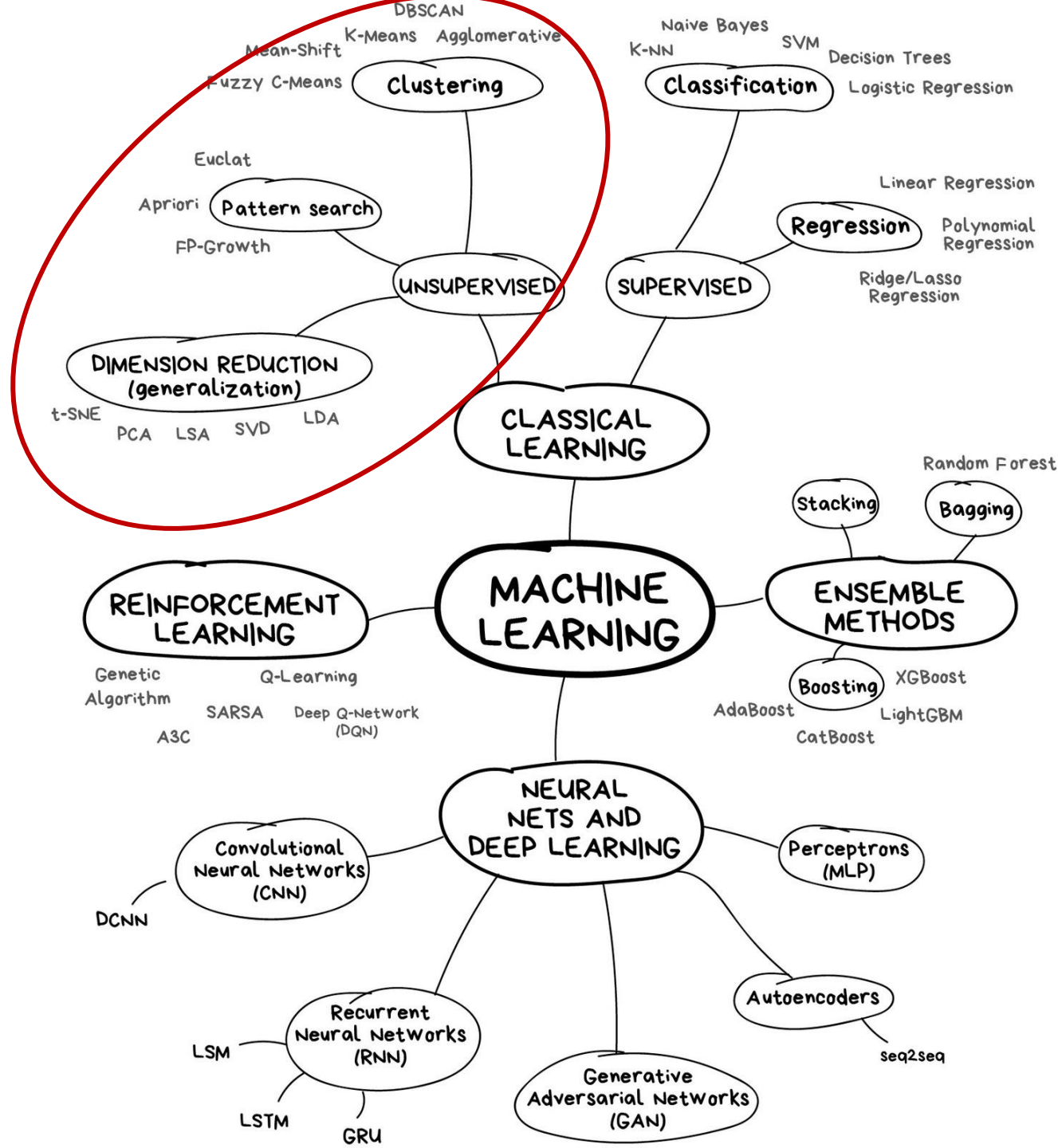


# Déroulement d'un projet Data / Machine Learning



**Machine Learning :**  
un ensemble de méthodes  
développées pouvant adresser  
diverses problématiques

Mathématiques  
Statistiques  
Probabilités  
Algèbre  
Géométrie  
Logique  
Multidimensionnel  
Vecteurs  
Lois  
Arithmétique

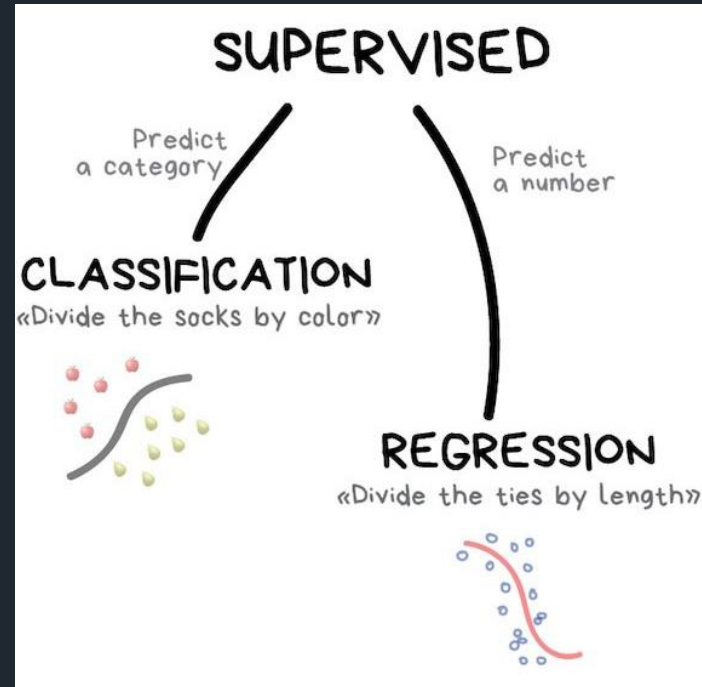




# Qualités d'un bon algorithme

- **Performance** : produire de bons résultats, être efficace
- **Robustesse** : privilégier des solutions fiables, pérennes dans le temps
- **Déployabilité** : solution actionnable dans la vraie vie
- **Transparence** : rendre compréhensible le modèle, le monitorer, suivre la dégradation/déviance dans le temps
- **Adéquation aux compétences** : algorithme accessible, ne nécessitant pas d'expertise trop poussée si non disponible en production, maintenable facilement
- **Proportionnalité** : mesurer le ROI, rester rentable

# MACHINE LEARNING ALGORITHMES SUPERVISES



*“The goal is to turn data into information,  
and information into insight.”*  
**Carly Fiorina**

# Algorithmes Supervisés : principes

- **Principes** :

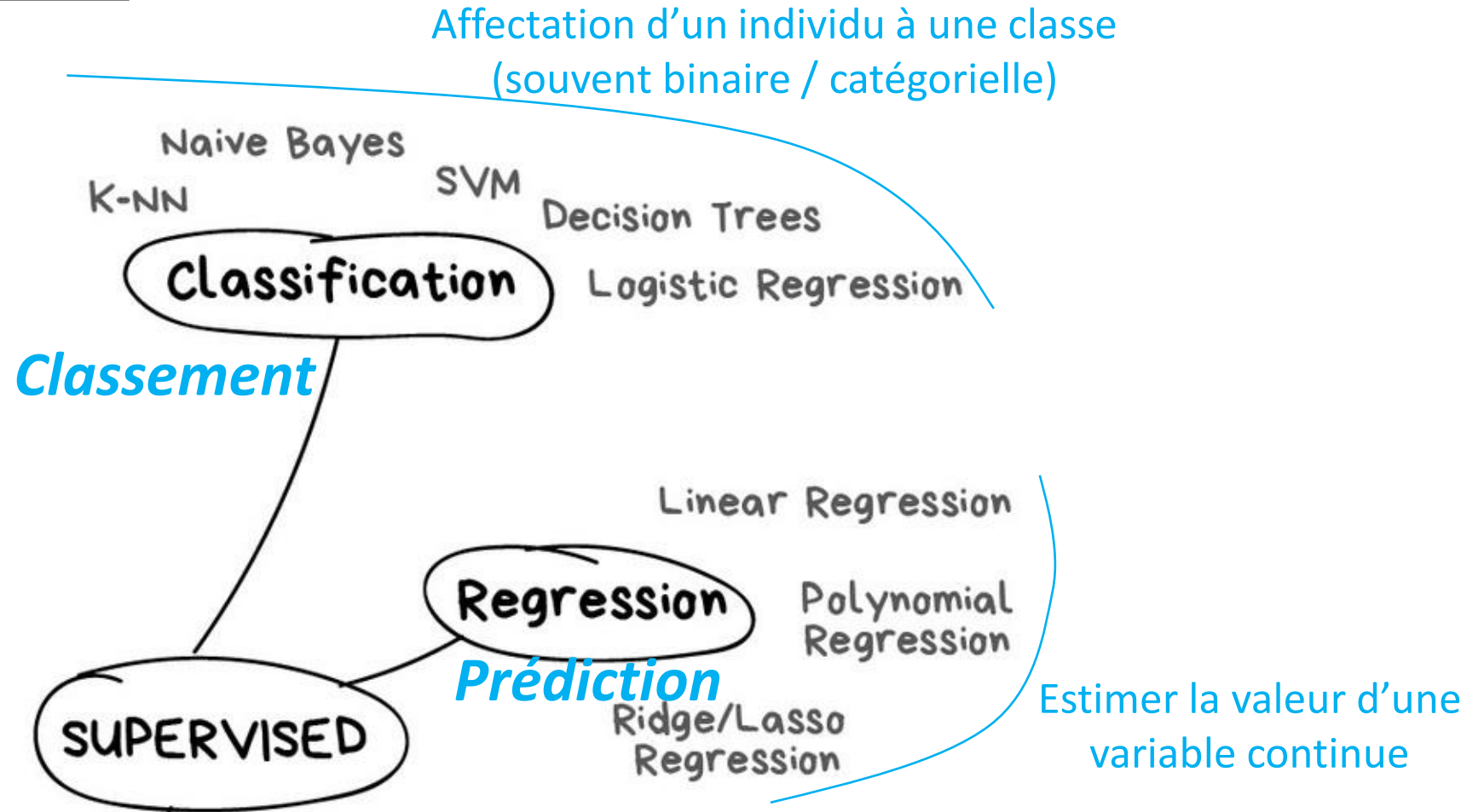
- **Variable cible (Y)** = variable à expliquer, aussi appelée variable dépendante ou endogène
- **Variables explicatives ( $X_i$ )** = une ou des variables indépendantes ou exogènes

- **Objectif** : trouver la meilleure combinaison de variables explicatives ( $X_i$ ) à travers un modèle (ie fonction) décrivant des caractéristiques et/ou distinguant des classes ou concepts **pour expliquer la variable cible** et ainsi **pouvoir prédire ses valeurs dans le futur**

- **Cas d'usage** : détection des grains de beauté présentant des anomalies, détection des images de bus, détection des appétents, prédire une valeur...etc
- **Algorithmes utiles** : algorithmes de régressions (régressions linéaires, arbres de régression...) ou algorithmes de classification (arbres de classement, régression logistique, réseaux de neurones ...), ...

# Algorithmes Supervisés : les modèles classiques

- Algorithmes utiles :



# Utilisation de la librairie Scikit-learn de Python contenant les méthodes de ML

- La librairie Scikit-Learn contient beaucoup de modèles de Machine Learning.
- Comme la plupart des librairies de Python, elle est bien documentée et illustrée avec des exemples.
- <https://scikit-learn.org/stable/>
- A noter : les sorties sous Python suite à l'utilisation des modèles de Machine Learning contiennent moins d'indicateurs que via d'autres logiciels « statistiques ».



## Classification

Identifying to which category an object belongs to.

**Applications:** Spam detection, Image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, ... — Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, ridge regression, Lasso, ... — Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, ... — Examples

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** PCA, feature selection, non-negative matrix factorization. — Examples

## Model selection

Comparing, validating and choosing parameters and models.

**Goal:** Improved accuracy via parameter tuning

**Modules:** grid search, cross validation, metrics. — Examples

## Preprocessing

Feature extraction and normalization.

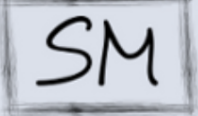
**Application:** Transforming input data such as text for use with machine learning algorithms.

**Modules:** preprocessing, feature extraction. — Examples

# D'autres librairies Python existent pour créer des modèles de ML

- La librairie StatsModels contient aussi beaucoup de modèles de Machine Learning.

- Elle peut présenter des similitudes avec R. Et au niveau de ses sorties, peut se rapprocher de sorties plus classiques obtenues avec des logiciels statistiques.



## StatsModels

Statistics in Python

[Install](#) | [Support](#) | [Bugs](#) | [Develop](#) | [Examples](#) | [FAQ](#) | [next](#) | [modules](#) | [index](#)

### Download

This documentation is for the **0.9.0** release. You can install it with pip:

```
pip install --upgrade --no-deps statsmodels
```

or conda:

```
conda install statsmodels
```

Documentation for the current development version is [here](#).

### Participate

Join the [Google Group](#):

Grab the source from [Github](#). Report bugs to the [Issue Tracker](#). Have a look at our [Developer](#) Pages.

### Quick search

## Welcome to Statsmodels's Documentation

**statsmodels** is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct. The package is released under the open source Modified BSD (3-clause) license. The online documentation is hosted at [statsmodels.org](#).

## Minimal Examples

Since version 0.5.0 of statsmodels, you can use R-style formulas together with `pandas` data frames to fit your models. Here is a simple example using ordinary least squares:

```
In [1]: import numpy as np
In [2]: import statsmodels.api as sm
In [3]: import statsmodels.formula.api as smf

# Load data
In [4]: dat = sm.datasets.get_rdataset("Guerry", "HistData").data

# Fit regression model (using the natural log of one of the regressors)
In [5]: results = smf.ols('Lottery ~ Literacy + np.log(Pop1831)', data=dat).fit()
```

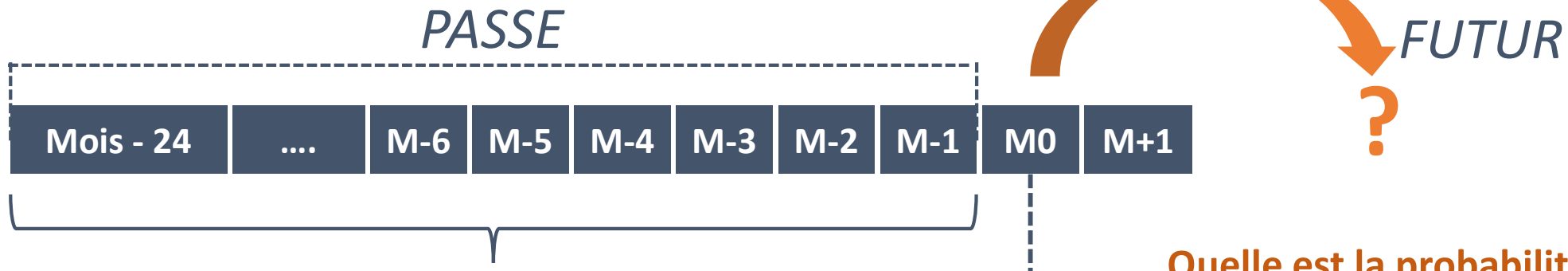


# Algorithmes Supervisés : fonctionnement de base

- Comment l'algorithme « devine »-t-il qu'un individu « statistique » est susceptible de réaliser l'évènement ?



**Utiliser le passé pour prédire l'avenir**



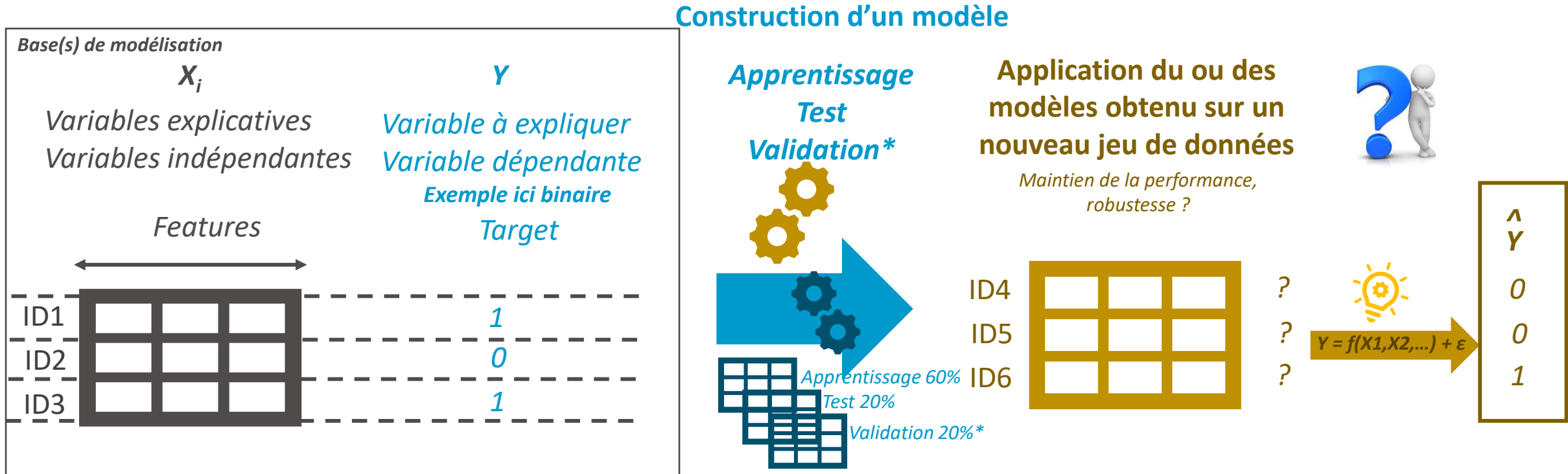
Construction d'une base de modélisation comportant l'ensemble des variables explicatives sur un historique et la variable à expliquer

Quelle est la probabilité pour que l'évènement se réalise ?

Quelle valeur puis-je prédire ?

# Algorithmes Supervisés : comment parvient-on à prédire ?

- Comment l'algorithme « devine »-t-il qu'un individu « statistique » est susceptible de réaliser l'évènement ?

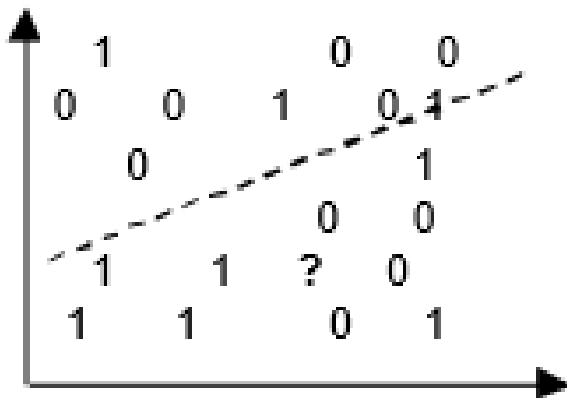


\* : % donné à titre indicatif, la construction de 2 ou 3 échantillons, leur répartition doit faire preuve de bon sens par rapport aux données disponibles, à l'objectif à atteindre

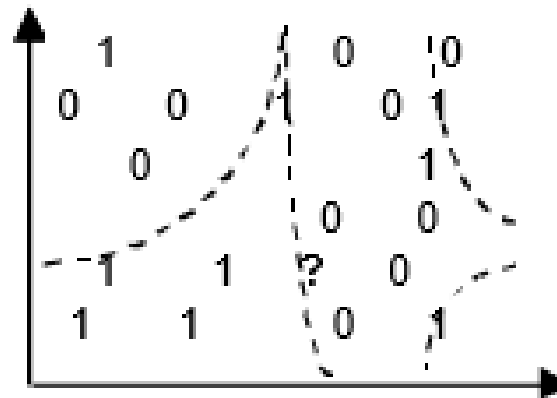


# Algorithmes Supervisés : représentation de différentes techniques de classement

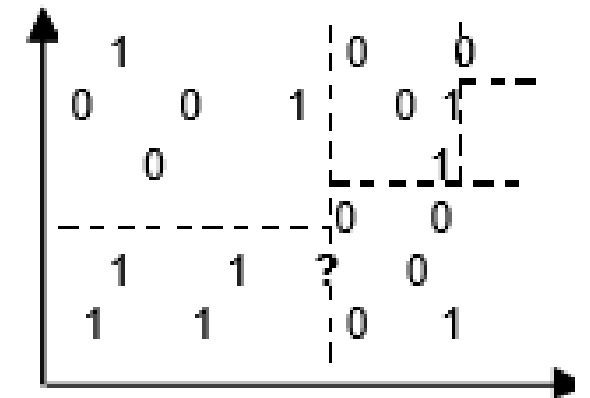
- Intérêt de différentes méthodes pour modéliser différentes formes : linéarité / non linéarité
- Représentation de différentes régions selon les techniques employées – frontières de prédiction



**Analyse discriminante**



**Réseau de neurones**



**Arbre de décision**

# Quelques algorithmes supervisés

*Note : Bien sûr, ces méthodes requiert au grand minimum 5 individus, voire un minimum de 30 !*

*Ce cours constitue une introduction, une acculturation aux méthodes les plus usuelles.  
Pour une pratique concrète, il sera nécessaire de consulter les théories mathématiques, d'approfondir l'application  
théorique et pratique des différentes méthodes (vérification des hypothèses de normalité...)*

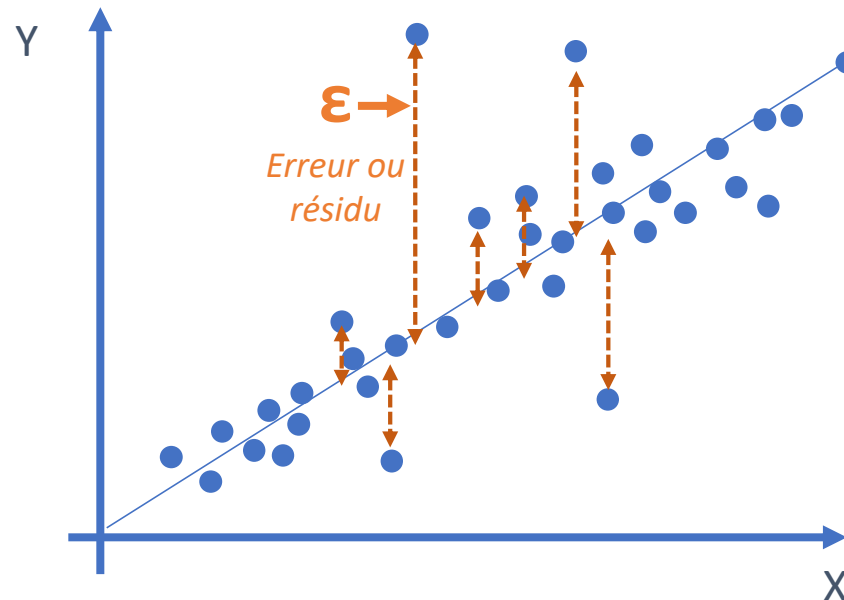
# Régression Linéaire Simple : le plus simple des modèles !

- **Objectif** : Expliquer Y (par exemple le chiffre d'affaire, une variable continue) en fonction de X (les ventes, une variable continue)
- **Prérequis** : Le **lien entre X et Y doit être linéaire**, souvent visible via une représentation des données de X et Y par un nuage de points (*pour que le modèle soit valide, préférable d'avoir plusieurs valeurs de X et Y*)
- **Equation du modèle – droite de régression** :  $y = ax + b$  (2 paramètres a et b) ou  $y = ax + b + \epsilon_i$  avec  $E(\epsilon_i) = 0$

Lien entre les 2 variables mesurable par le coefficient de corrélation  $R^2$

(pour rappel  $R^2$  varie entre 0 et 1)

$R^2$  est un indicateur de qualité d'ajustement entre les données estimées et les données observées



La droite de régression ajuste le nuage de points.

Part inexpliquée du modèle

= résidus, les erreurs  $\epsilon$

$$Y - \hat{Y}$$

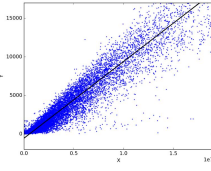
# Cas pratique

Nous avons obtenu l'équation suivante  $y = 3x + 5$  à partir d'un jeu de données

Question : si  $x$  prend la valeur de 556, quelle sera la valeur de  $y$  ?

1673

# Généralisation du modèle linéaire : régression linéaire multiple



- **Objectif** : Expliquer Y (par exemple le chiffre d'affaire) en fonction de plusieurs  $X_i$  (les ventes du produit A, les ventes du produits B, ..., la météo,...)
- **Prérequis : Hypothèses du modèle linéaire**
  - L'espérance conditionnelle  $E(Y/X=x)$  est une fonction linéaire
  - Les variables indépendantes doivent être non corrélées entre elles
  - Concernant les résidus :
    - Les résidus sont linéairement indépendants
    - Les résidus sont normalement distribués
    - La variance des résidus est la même pour toutes les valeurs de X
- **Equation du modèle – droite de régression** :  $y = a_1x_1 + a_2x_2 + \dots + a_nx_n + b$
- **Méthode des Moindres Carrés Ordinaires** : Minimiser  $\sum_i (y_i - b - ax_i)^2$  (recherche d'estimateurs de a et b minimisant les résidus)
- Variance Totale = Variance expliquée par la régression + Variance résiduelle
- Somme des carrés totale (SCT) = Somme des carrés dus à la régression (SCR) + Somme des carrés résiduels (SCE)
- **Qualité du modèle : Coefficient de détermination** : part de variabilité de Y décrite par le modèle  
 $R^2 = 1 - (SCE/SCT)$  si  $R^2 = 90\%$  soit 90% de la variable CA est expliqué par le modèle

# Régression linéaire : avantages & inconvénients



- **Simple à comprendre**
- **Pas un algorithme complexe** : le modèle repose sur une **expression mathématique explicite**, **calculer une prédiction est donc rapide**
- **Interprétation du modèle** : sélection simple des variables ayant un pouvoir discriminant par l'analyse de la p-value ( $< 0.05$ )

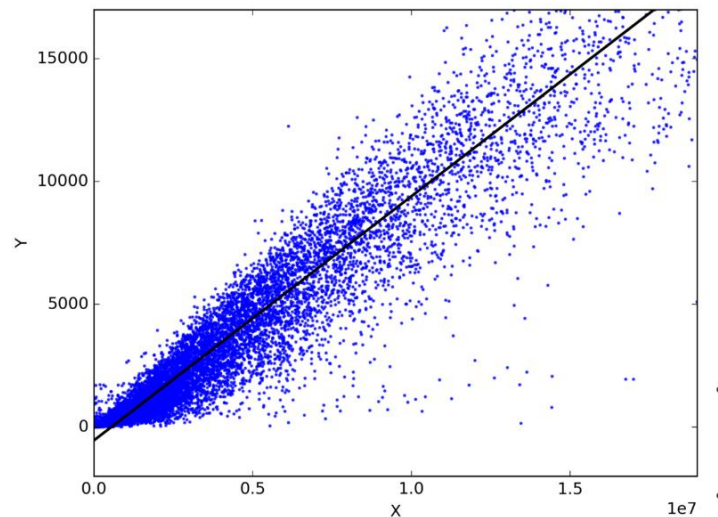
```
NOTE_SCORE= -0.00992062986
+ 0.0678990976* USG_M1
+ 0.0567889865* NB_APPELS_M1
+ 0.0456778754* NB_CNX_M1
+ 0.0345667643* ANC_T3
+ 0.0234556532* AGE_T3
+ 0.0123445421* TYPE_PDT
```



- **Algorithme linéaire** : présuppose une linéarité dans la liaison des variables (souvent illusoire sur les grands jeux de données), nécessite l'absence de multicolinéarité parmi les variables explicatives, néglige les interactions entre les variables prédictives, présuppose de vérifier l'ensemble des hypothèses pour être correctement appliqué
- **Algorithme sensible aux valeurs aberrantes**. Peut contourner ceci en utilisant des techniques de régularisation (lasso, ridge, elastic net)

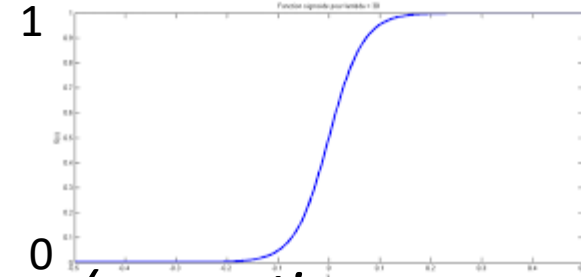
# Cas pratiques

- Réaliser une régression linéaire simple (jeu de données et programme Python fourni)
- A partir du jeu de données confié lors de la précédente session concernant la régression linéaire simple, recalculer la droite d'équation entre le CA et les ventes et le coefficient de corrélation, représenter le nuage de points et la droite l'ajustant
- Calculer avec l'aide de Python le  $\chi^2$  sur les 2 jeux de données confiés – comparer avec les résultats calculés sous excel lors de la précédente session
- Réaliser une régression linéaire multiple (jeu de données et programme Python fourni)



# Régression Logistique : méthode très usuelle

- **Modèle de classification linéaire utilisant une fonction de score  $S$  des variables prédictives** :  $S(x) = a_1x_1 + \dots + a_nx_n$
- **La variable cible (target) est binaire** :  $Y = 0$  ou  $1$
- **Les variables explicatives peuvent être continues ou binaires** (*en pratique, on peut mettre les continues en classes*).
- **Principe** :
  - Chercher des coefficients  $a_1, \dots, a_n$  tels que :
  - Le score  $S(x)$  soit positif lorsque **les chances d'appartenir à la classe 1 sont grandes** et inversement, le score  $S(x)$  soit négatif lorsque les chances d'appartenir à la classe 0 sont grandes
  - **Modèle probabiliste paramétrique et discriminant**
  - **Modéliser l'espérance conditionnelle** :  $E(Y/X=x) = P(Y=1 / X=x)$  sous la forme  $E(Y/X=x) = \beta_0 + \beta_1x_1 + \dots + \beta_nx_n$



$n$  : étant le nombre de variables

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) \text{ où } p \text{ est défini sur } ]0; 1[$$



# Régression Logistique : méthode très usuelle

- **Fonction de lien classique : **logit**** (il existe aussi le probit et log-log)

$$p(x) = \frac{e^{b_0 + \sum_j b_j x_j}}{1 + e^{b_0 + \sum_j b_j x_j}}$$

$$\text{Log}\left(\frac{p(x)}{1-p(x)}\right) = b_0 + b_1 x_1 + \dots + b_p x_p$$

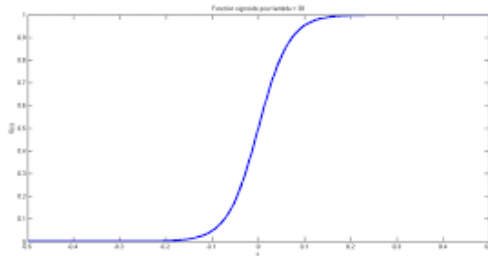
- **Interprétation : Odds Ratio et risques relatifs**

- **Odds pour la « cote » des parieurs. L'OR permet de mesurer l'effet relatif d'un facteur.**
- L'odds ratio est proche du risque relatif lorsque le nombre d'événements est faible.
- L'odds ratio est toujours supérieur ou égal à zéro et s'interprète par rapport à une modalité de référence (*explicite dans les sorties, que l'on peut choisir selon le logiciel*).
- Si l'odds ratio est :
  - proche de 1, l'événement est indépendant du groupe ;
  - supérieur à 1, l'événement est plus fréquent dans le groupe A que dans le groupe B ;
  - bien supérieur à 1, l'événement est beaucoup plus fréquent dans le groupe A que dans le groupe B ;
  - inférieur à 1, l'événement est moins fréquent dans le groupe A que dans le groupe B ;
  - proche de zéro, l'événement est beaucoup moins fréquent dans le groupe A que dans le groupe B.
- Ex : Un odd ratio de 3 signifie que l'odd de la variable discriminée dans le groupe A est **3 fois plus élevé** que dans le groupe B pour la caractéristique mesurée.

# Régression logistique : avantages & inconvénients

+

- **Classification d'une nouvelle observation rapide** puisqu'elle se résume à l'évaluation d'une fonction de score linéaire  $S$
- **Algorithme simple** : ainsi peu sensible au surapprentissage, traitement de variables explicatives qualitatives ou continues
- **Interprétation des coefficients du score simple** (avec les OR/probabilités).



-

- **L'hypothèse de linéarité du score** empêche de tenir compte des interactions entre variables. Méthode avec de fortes hypothèses à vérifier (multicolinéarité des variables explicatives...)
- **Données en entrée bien préparées** : sans valeur manquante, non redondantes, mise en classes (notamment pour éviter les valeurs aberrantes)...
- **Phase d'apprentissage pouvant être longue** : l'opération numérique d'optimisation des coefficients est complexe.
- **Algorithme plutôt limité aux variables cibles binaires**. Un enchainement de plusieurs régressions logistiques permet de surmonter cette limitation.

# Régression : sélection de variables – modèle step-by-step

Il existe plusieurs manières de faire tester la pertinence de l'entrée d'une variable dans un modèle :

**Conseil : tester plusieurs méthodes**

- **Ascendante / Forward**

- Principe de l'algorithme :

- Partir d'un modèle ne contenant aucune variable explicative
    - Ajouter séquentiellement une variable permettant d'optimiser les critères (selon un seuil de significativité, augmentation du  $R^2$ ) jusqu'à ce que plus aucune variable ne puisse améliorer le modèle – tests de N modèles puis test de N-1 modèles, ainsi de suite, etc

- **Descendante / Backward**

- Principe :

- Partir du modèle complet contenant toutes les variables
    - De manière itérative, éliminer les variables une à une (variable non suffisamment corrélée à la cible : la variable reste dans le modèle si sa p-value est en-dessous du seuil de significativité (souvent  $SL = 0,05$ ))

- **Progressive / Stepwise / Mixte**

- Principe :

- Aucune variable au départ
    - Ajout de variables corrélées à la cible à chaque étape et suppression de certaines variables si leur pouvoir discriminant est contenu dans une nouvelle combinaison de variables – choisir 2 seuils de significativité pour entrer et rester dans le modèle

- **Globale**

- Essai d'ajuster le  $R^2$  en comparant une partie de tous les modèles possibles / élimination des modèles les moins intéressants

# Régression Ridge, Lasso, Elastic Net

Des méthodes intégrant un **principe de pénalisation ou de régularisation** (ajout de contraintes).

**Méthodes plus adaptées aux grands jeux de données contenant des variables potentiellement corrélées.**

Ces méthodes nécessitent de centrer/réduire les variables.

**Elles s'appliquent à la régression linéaire et plus globalement aux modèles linéaires généralisés (logistique, Support Vector Machines...).**

- **Régression Ridge**

- Modèle performant, mais contenant beaucoup de variables...
- Modèle avec un fort pouvoir prédictif
- Modèle interprétable quoiqu'il utilise toutes les variables

- **Régression Lasso**

- Méthode qui permet aussi une **bonne sélection des variables**, mais décide de conserver arbitrairement une seule variable parmi un ensemble de variables corrélées
- Ne retrouve pas forcément les bonnes variables d'intérêt dans une base à très grande dimension
- Modèle synthétique (sélection des variables) et interprétable

- **Régression Elastic Net**

- Compromis des 2 méthodes permettant à la fois de rétrécir les coefficients et d'en annuler d'autres – optimisation sous contrainte

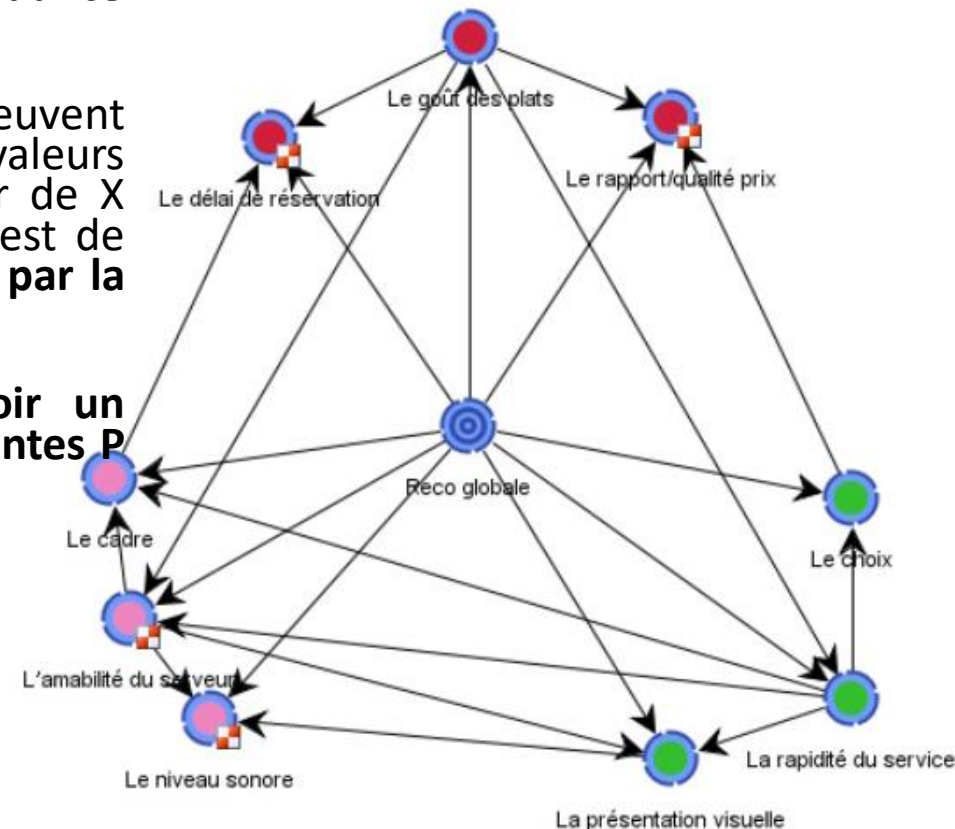
# Réseaux bayésiens : les probabilités conditionnelles $P(X/Y)$

- Méthode pouvant être classée selon son utilisation dans les méthodes non supervisées comme supervisées.
- *Repose sur des théories probabilistes :*

**$P(X|Y) = (P(Y|X) * P(X)) / P(Y)$  avec une hypothèse forte d'indépendance des probabilités conditionnelles des variables prédictives**

- **Objectif** : Découvrir les liens entre les variables. Ses résultats peuvent prédire la valeur d'une variable à partir de la connaissance des valeurs d'autres variables lui « étant liées » : quelle pourrait-être la valeur de X sachant qu'en temps normal quand il y a Y, la probabilité d'avoir X est de z%.... Cette méthode s'ouvre sur la **découverte de relations causales par la recherche des liens inter-variables.**
- **Prérequis** : Disposer d'un historique suffisant permettant d'avoir un maximum de probabilités entre « événements » : probabilités conjointes  $P(X \text{ et } Y)$ ,  $P(X/Y)$ ...

Représenter sous forme de graphe  
- Les nœuds sont étiquetés par les variables et reliés avec des probabilités annotées (conjointes, conditionnelles)



# Réseaux bayésiens : avantages & inconvénients

+

- Simplicité de l'algorithme assurant de bonnes performances
- Graphique, visuel : représentation des variables sous forme de nœuds et liées les unes aux autres
- Algorithme performant, utile et prédictif même dans les situations où l'hypothèse d'indépendance de probabilités conditionnelles des variables prédictives n'est pas possible à justifier

-

- Repose sur des théories probabilistes : les prédictions des probabilités pour les différentes classes sont erronées lorsque l'hypothèse d'indépendance conditionnelle est invalide

# Arbre de décision : méthode la plus lisible et visuelle ! (1/5)

- Peut s'appliquer à un apprentissage supervisé pour **une régression et pour un classement** !
- **Target (Y) :**
  - continue pour la régression
  - qualitative : binaire ou à plusieurs valeurs !
- **Variables explicatives (X) :**
  - Tout type !
  - Attention aux différences d'échelles entre les variables explicatives, il peut être bien de les mettre sur des échelles assez comparables.
- **Principe :**
  - **L'algorithme sélectionne la variable séparant, discriminant le mieux les individus d'une population selon la cible définie en sous-population appelées nœuds**
  - **Fonctionnement itératif : Procède ainsi de suite jusqu'à ce que les critères définis en amont soient atteints**
  - En termes de vocabulaire : branches, nœuds, feuilles (nœuds terminaux)
  - L'arbre peut être binaire (cas le plus fréquent ou non)

# Arbre de décision : méthode la plus lisible et visuelle ! (2/5)

- **Interprétation d'un arbre :**

- l'algorithme aboutit à **un ensemble de règles très facilement transcriposables en SQL** (et donc très faciles à mettre en production) et **interprétables** sans aucune connaissance préalable
- Méthode intuitive

- **Plusieurs types d'arbres existent, avec des spécificités notamment sur la manière de supprimer les branches de peu d'intérêt :**

- **CHAID** – Chi-Square Automation Interaction Detection – adapté à l'étude des variables explicatives discrètes – utilise le chi2 pour sélectionner la variable discriminante et le découpage des modalités, cible binaire ou non
- **CART** – Classification and Regression Tree : maximise la pureté des nœuds (homogénéité), adapté à tout type de variables explicatives, arbre binaire
- **C5.0 / C4.5** : maximise la gain d'information réalisé en affectant chaque individu à une branche de l'arbre, adapté à tout type de variables explicatives
- etc



# Arbre de décision : méthode la plus lisible et visuelle ! (3/5)

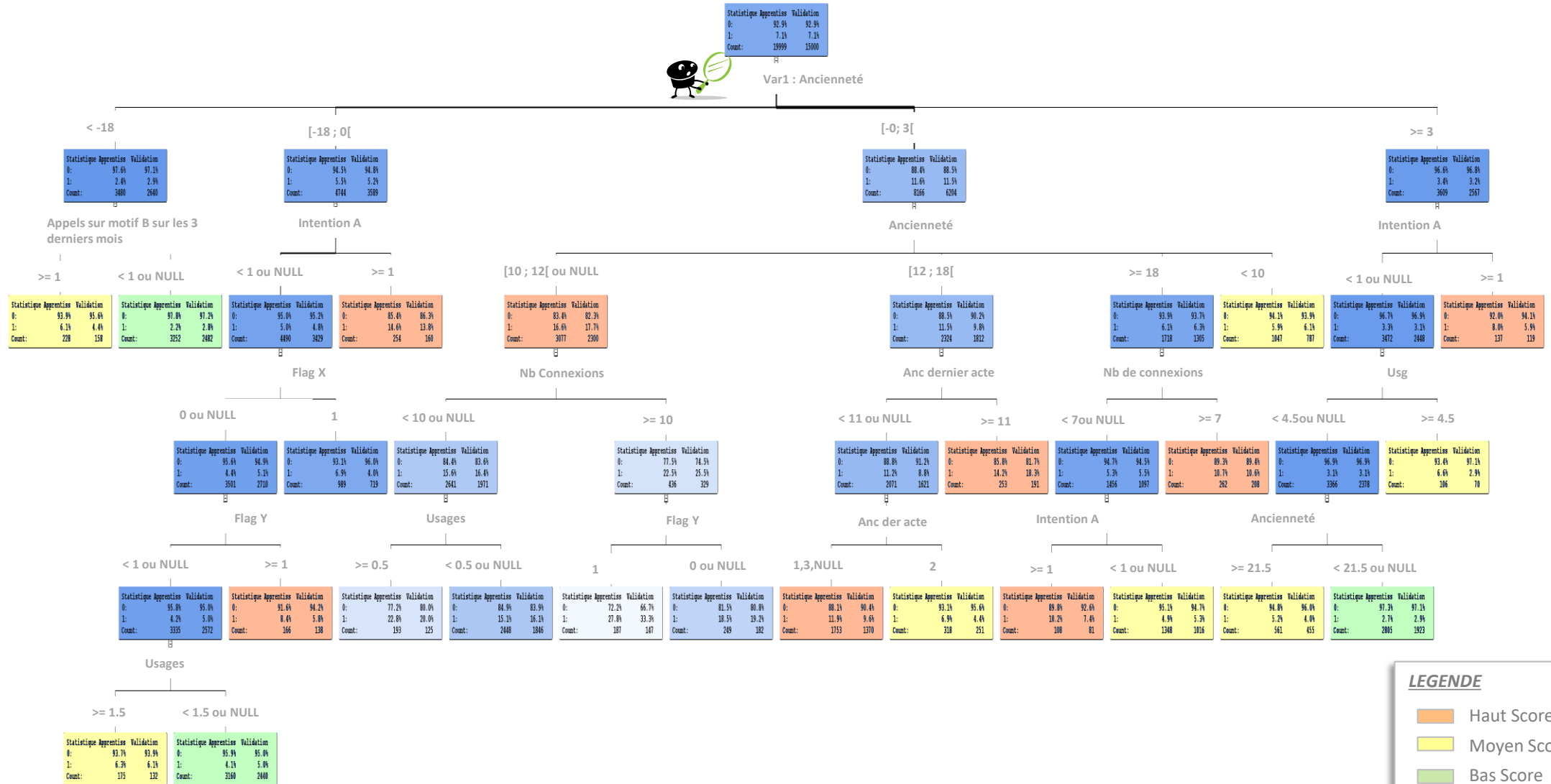
- **Critères de séparation :**

- **Indice de Gini** : utilisé dans CART
- **Entropie** : utilisé dans les arbres C4.5 et C5.0
- **Indice de Twoing** : utilisé dans CART lorsque la cible a au moins 3 modalités
- **Critère du  $\chi^2$**  : utilisé dans l'arbre CHAID
- Globalement, plus le nœud est pur, plus les indices de Gini ou entropie sont bas

- **Critères d'arrêt possibles d'un arbre** : à paramétrer en amont

- **Profondeur de l'arbre** : souvent on fixe une limite
- **Nombre de feuilles** : nombre de règles maximum
- **Effectif de chaque nœud** : fixe une valeur d'au moins 50 individus dans un nœud par ex
- **Qualité de l'arbre** : suffisante ou n'augmente plus
- Globalement, plus le nœud est pur, plus les indices de Gini ou entropie sont bas

# Arbre de décision : illustration (4/5)



# Arbre de décision : avantages & inconvénients (5/5)

+

- **Variables explicatives de tout type** pouvant être qualitatives ou quantitatives
- **Phase de préparation des données réduite** (pas de normalisation, ni traitement de valeurs manquantes)
- **Prend en compte les interactions entre variables** (pas d'hypothèse de linéarité) – méthode non paramétrique
- Traite les problèmes de classification dans toute leur généralité (cible binaire, ou plus)
- **Très interprétable et délivre un ensemble de règles intelligibles**
- **Méthode relativement rapide**

-

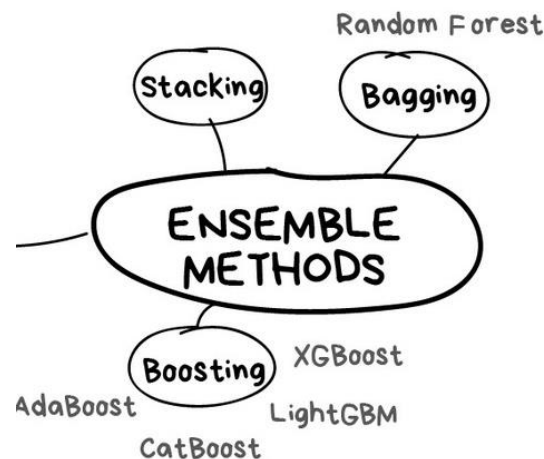
- **Risque de surapprentissage** (besoin d'élaguer l'arbre et de paramétrer l'arrêt)
- **Le critère de segmentation affecté au 1<sup>er</sup> nœud possède une très grande influence** sur le modèle de prédiction (besoin d'un jeu de données très représentatif) : détection d'optimums locaux et non globaux
- **Manque de robustesse**

# Existence d'autres méthodes non abordées dans ce cours

---

- En voici quelques unes :
  - Régression polynomiale
  - Régression PLS
  - Support Vector Machine : méthodes à noyau / Kernel SVM
  - K plus proches voisins
  - etc

# Amélioration des résultats d'un modèle



# Comment améliorer le modèle ?

- Des techniques de rééchantillonnage existent pour améliorer ses résultats ou se donner un maximum de chances d'aboutir à un modèle robuste (techniques notamment adaptées en cas d'échantillons de petite taille)
- En voici quelques unes :
  - **Bootstrap** : créer plusieurs échantillons en réalisant des tirages aléatoires avec remise
  - **Bagging** - Bootstrap Aggregating : construction d'une famille de modèles sur m échantillons bootstrap dont les modèles seront agrégés par vote ou moyenne des estimations
  - **Boosting** : même principe que la méthode précédente si ce n'est qu'on augmente à chaque itération le poids des individus précédemment mal classés et qu'on travaille sur toute la population

# Compétition ou Emboitement de modèles

# Comment choisir la meilleure méthode ?

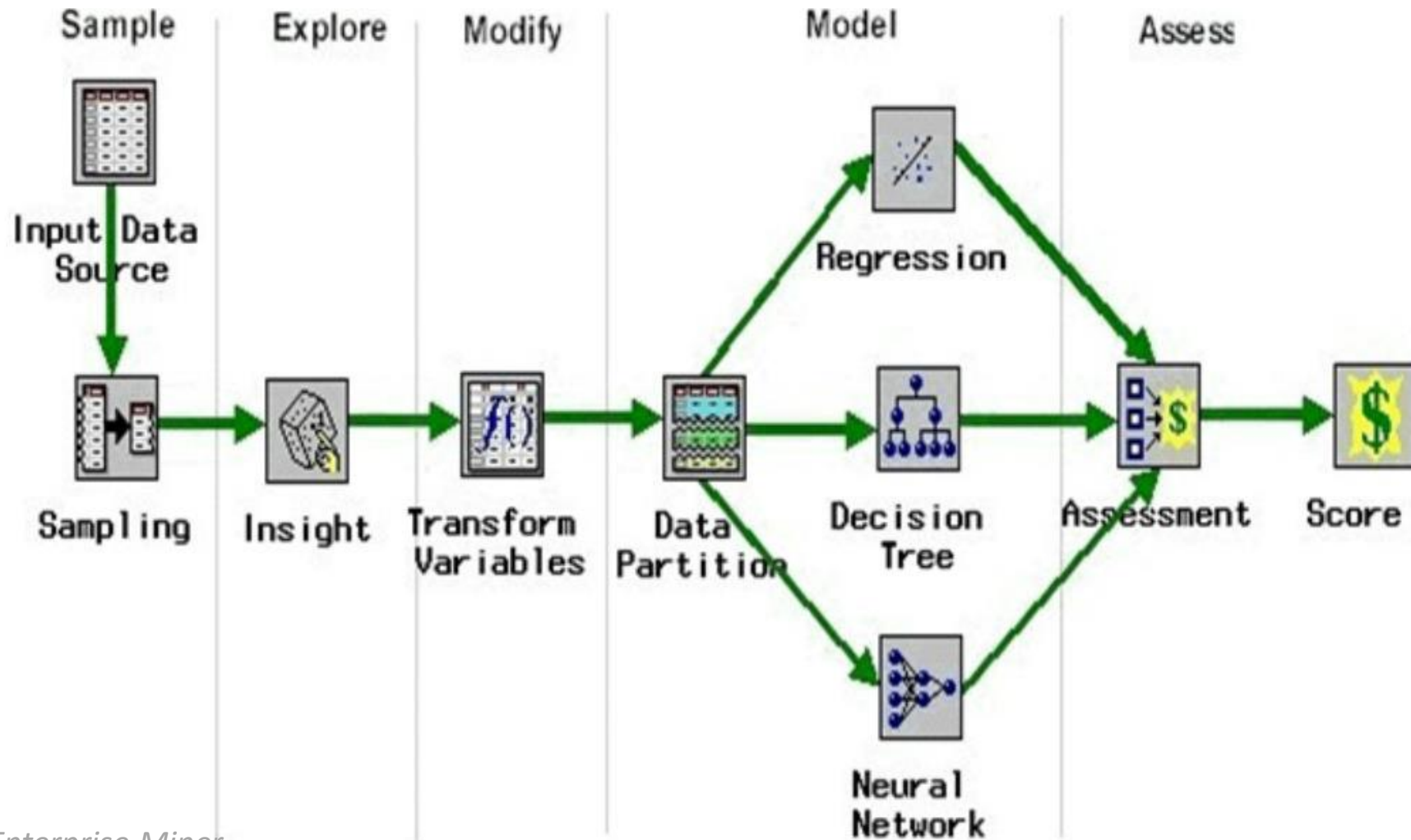
- Parmi les méthodes applicables : cible continue ou catégorielle

➔ **Tester plusieurs modèles, plusieurs paramétrages** : vérifier les indices de performance, la matrice de confusion, les taux d'erreurs, comparer les modèles, leurs performances, les variables conservées comme ayant un pouvoir prédictif...



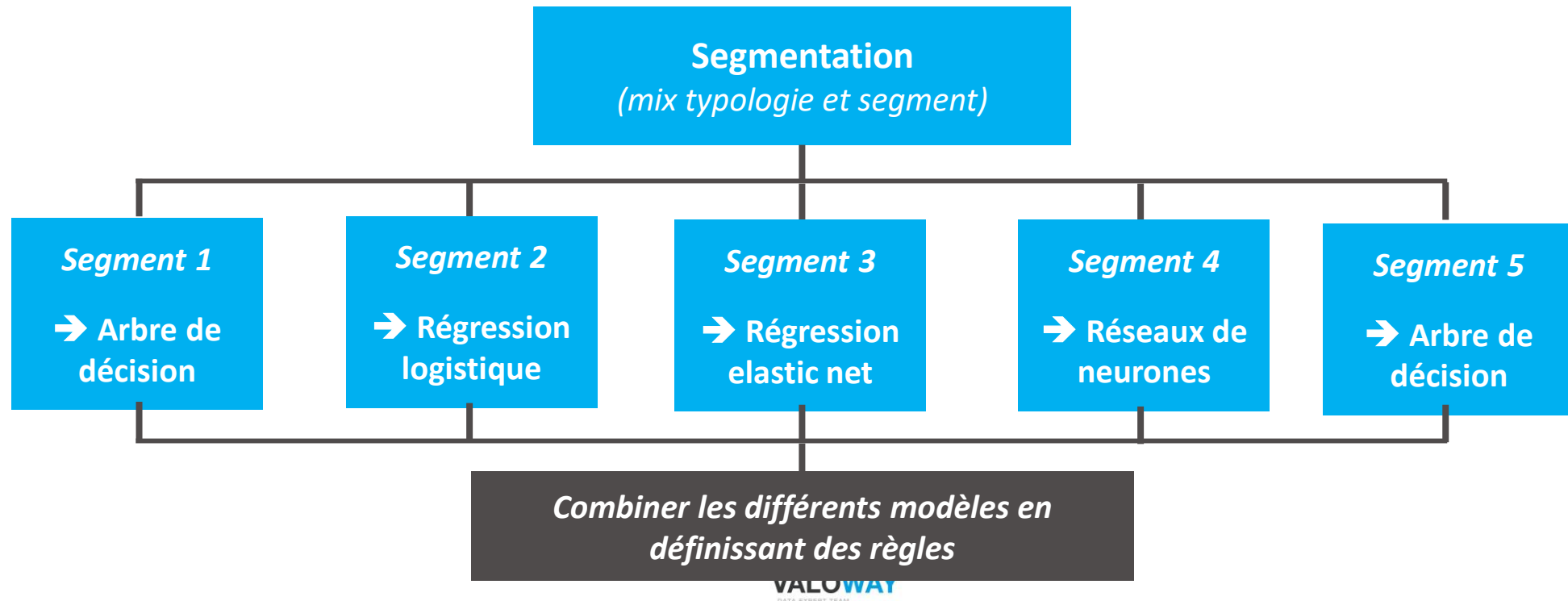
# Compétition des modélisations

Travail du data scientist : comparer différents modèles et choisir le meilleur selon les critères souhaités (plus robuste, plus performant...)



# Possibilité : combinaison / emboitement de modèles (1/2)

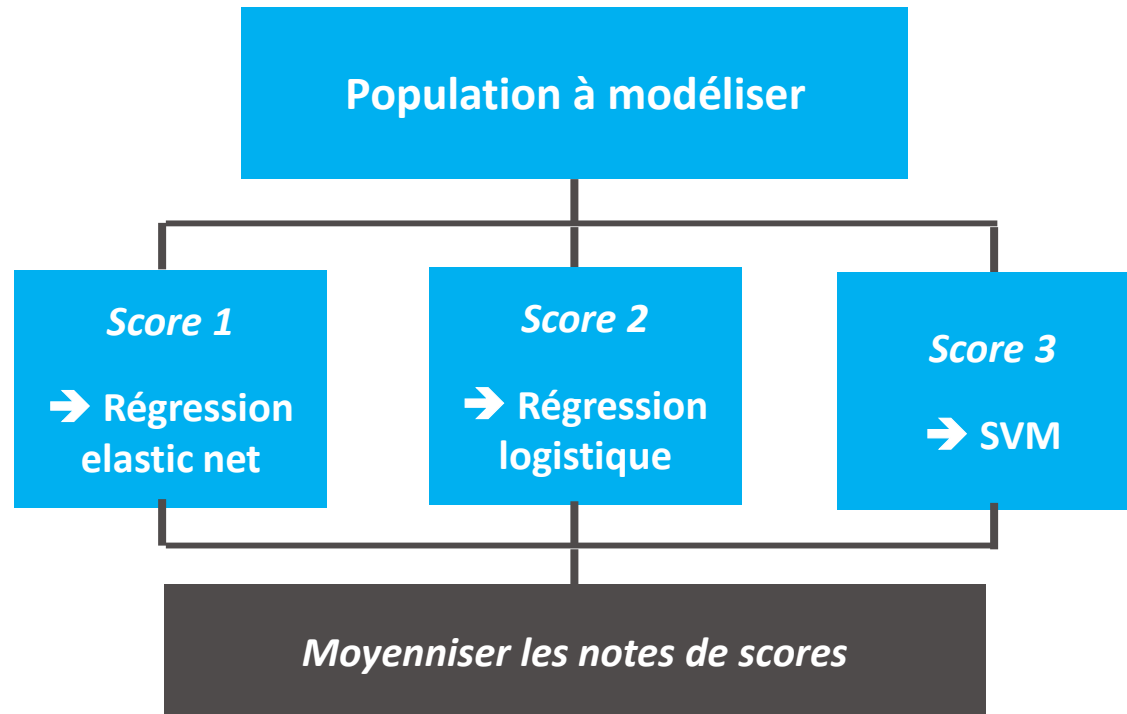
- Fréquent de **combiner différentes modélisations afin de traiter au mieux un sujet et / ou de les emboîter**
- Exemple : appliquer une segmentation sur une population puis différentes techniques de score par segment



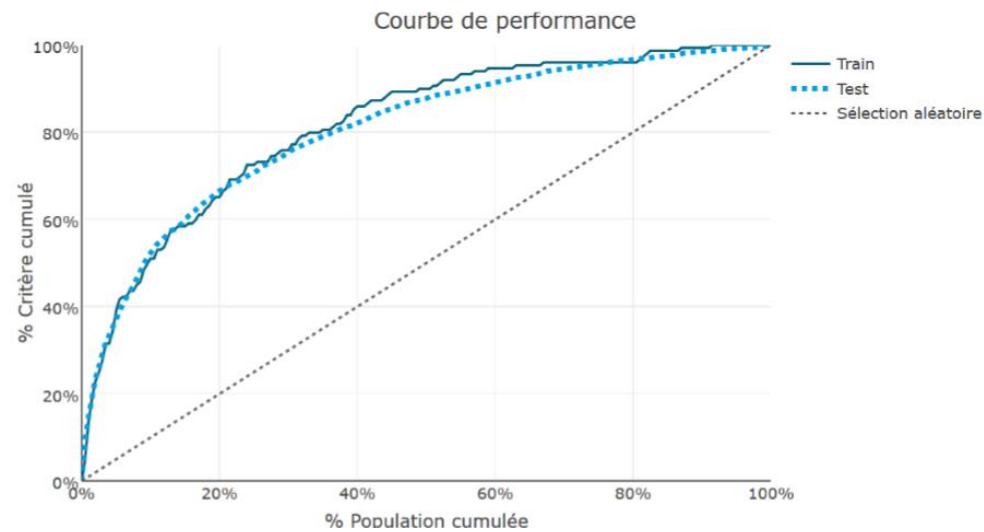
# Possibilité : combinaison / emboitement de modèles (2/2)

Exemple :

- calculer des modèles sur une même population à l'aide de différentes méthodes
- puis moyenniser le score obtenu par individu après réflexion sur les règles à mettre en place (moyenniser pour des clients appartenant à une même classe de risque ou de chance ? )



# Evaluation d'un modèle



# Pourquoi évaluer son modèle ?

- Choisir le meilleur modèle : « la meilleure équation » parmi les algorithmes et méthodes testés
- Tester la robustesse et la performance de son modèle sur d'autres jeux de données ayant les mêmes caractéristiques
- Verrouiller ces critères lorsque le modèle sera mis en production, déployé dans la vie réelle (maîtriser les taux d'erreurs, anticiper les déviations potentielles...)

# Matrice de confusion : indicateurs de qualité

- **Matrice de confusion** : affichant les taux de bien classés / mal classés en confrontant les valeurs réelles de Y et prédites.
- *Ici, modèle de classification binaire*

Y  
Réalité

Prédictions

Matrice de confusion	0	1	
0	8290	85	8375
1	135	1490	1625
	8425	1575	10000

Matrice de confusion	0	1	
0	83%	1%	84%
1	1%	15%	16%
	84%	16%	100%

Réalité

Prédictions

Matrice de confusion	Négatif	Positif
Négatif	Vrai Négatif	Faux Positif
Positif	Faux Négatif	Vrai Positif

Erreur de 1<sup>ère</sup> espèce

Erreur de 2<sup>ème</sup> espèce

Précision globale du modèle

**Accuracy Rate – Taux de bien classés** = Taux de correct / total

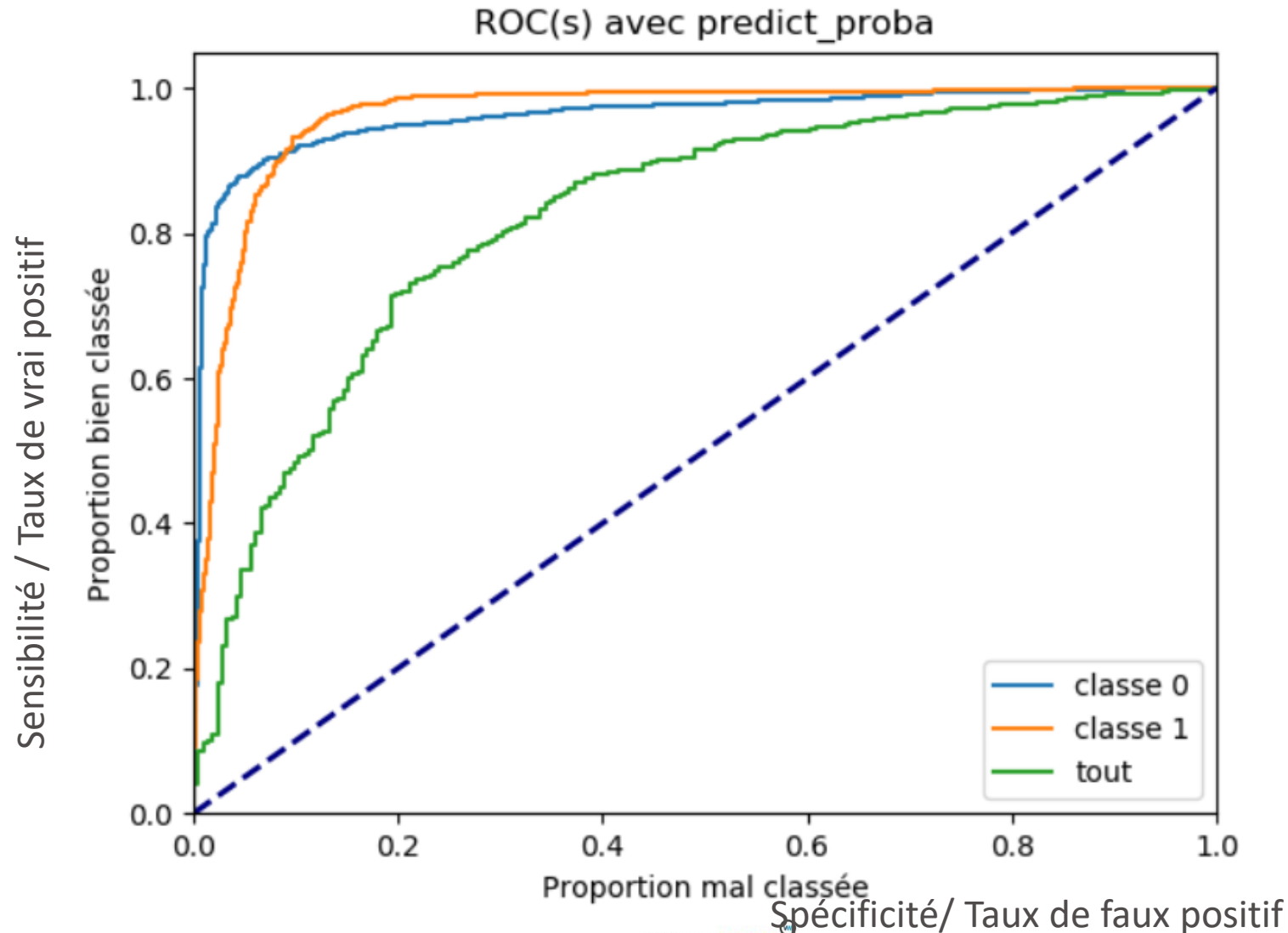
**Taux d'erreur** = Taux de non correct / total

# La courbe de ROC

*Receiving Operating Characteristics*

- Parfaitement adaptée pour comparer les performances de différents modèles
  - La courbe de ROC permet d'obtenir le meilleur modèle ayant le plus possible de vrais positifs avec le moins de faux positifs en représentant la proportion  $y$  de vrais positifs en fonction de la proportion  $x$  de faux positifs lorsque l'on fait varier le seuil  $s$  du score
- Permet d'obtenir **l'AUC (Area Under Curve)**, l'aire sous la courbe
  - AUC = probabilité pour que la fonction SCORE place un positif devant un négatif
    - L'AUC varie de 0,5 à 1.
      - 1 étant un modèle optimal (maximum).
      - En aléatoire, l'AUC est à 0,5 (un modèle à 0,5 n'est pas un bon modèle performant)

# La courbe de ROC (*Receiving Operating Characteristics*/curve)



Mesure de qualité  
d'un modèle : l'aire  
sous la courbe de  
ROC, AUC (Area  
Under Curve)  
 $\frac{1}{2} \leq \text{AUC} \leq 1$



# La courbe de Lift (*de concentration*)

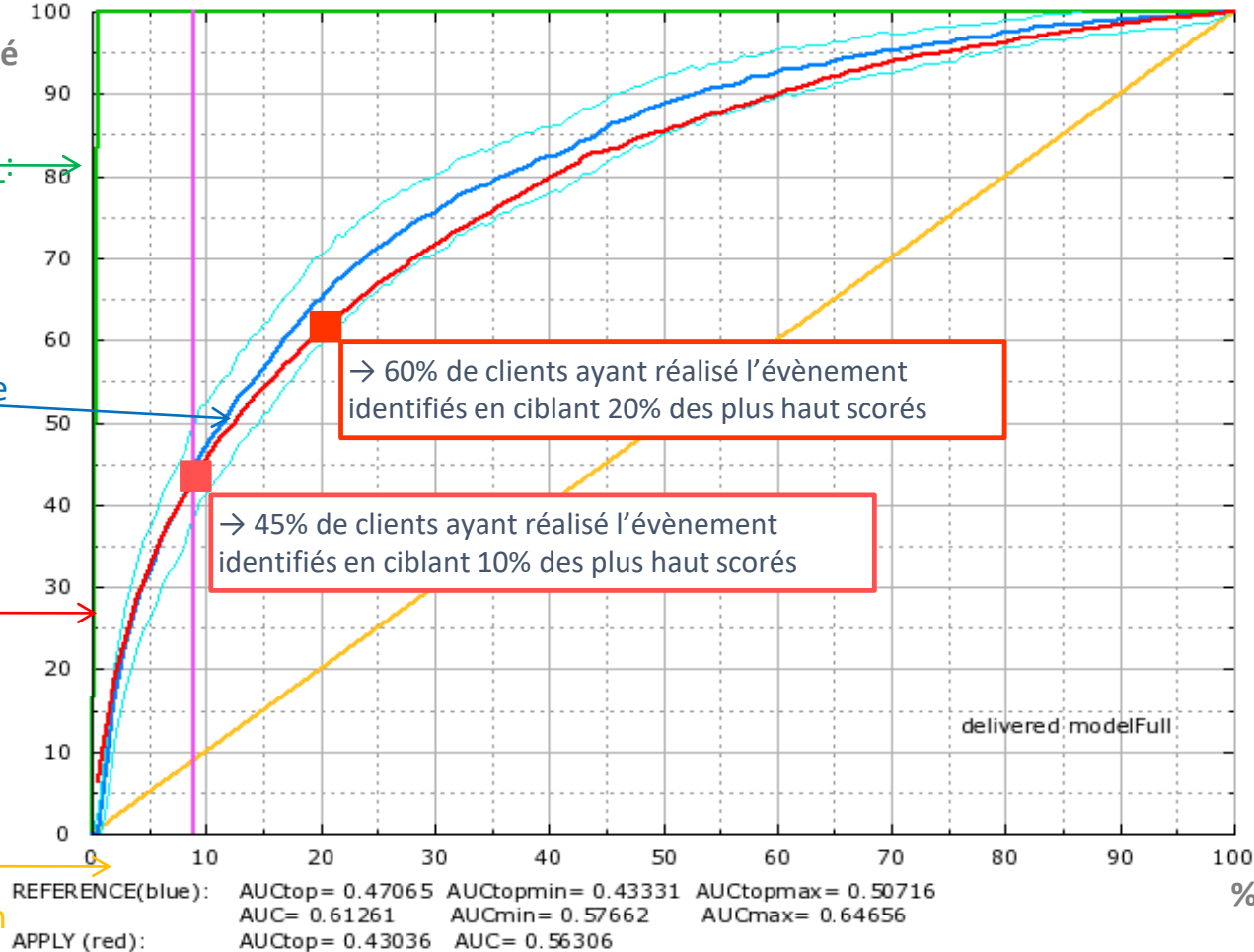
% d'individus pour lesquels  
l'évènement s'est déclenché

La concentration maximale :  
tous les individus sont  
regroupés (idéal)

La courbe de lift du modèle  
d'apprentissage sur les  
données de validation

La courbe de lift du modèle  
sur de nouvelles données

L'aléatoire : les individus  
sont répartis dans  
l'ensemble de la population



AUC : indicateur phare  
Aire sous la courbe de  
ROC, plus cette aire est  
élevée, plus le modèle  
est performant

- L'AUC du modèle de test est de **0,56**, ce qui reste une performance acceptable.
- Le modèle est relativement robuste en réapplication

# A noter : existence d'autres indices

- D'autres indices peuvent aider à évaluer un modèle : mesure la performance, l'ajustement entre  $\hat{Y}$  et  $Y$ 
  - Le coefficient de détermination (déjà rencontré à plusieurs reprises dans ce cours) + erreur quadratique moyenne (RMSE)
  - L'indice de Gini (ratio) : plus il est élevé, meilleur est le modèle
  - AIC : Akaike : plus il est faible, meilleur est le modèle
  - SC : Schwarz : plus il est faible, meilleur est le modèle
  - $-2 \log (L(\beta_k) - L(\beta_{\max}))$ : déviance, plus elle est proche de 0, plus le modèle colle à la réalité
  - $R^2$  ajusté : introduit un facteur de pénalisation (par rapport à l'ajout de nouvelles variables faisant augmenter le  $R^2$ ) permettant de fiabiliser le coefficient de détermination
  - ...

# Modèle choisi

# Modélisation de classification binaire : *résultat de l'application du modèle*

*Pour illustration\**

	Probabilité que l'évènement se réalise	Classes
Client1	99,99%	A
Client2	99,95%	A
Client3	98,50%	A
Client4	98,00%	A
Client5	97,50%	A
Client6	97,00%	A
.		
.		
Client500	75%	B
.		
.		
.		
Client nième	0%	C

5%\* des clients présentent des chances très élevées de réaliser l'évènement = HAUTS SCORES

25%\* des clients présentent des chances moyennement élevées de réaliser l'évènement = MOYENS SCORES

70%\* des clients présentent des chances plus faibles de réaliser l'évènement = BAS SCORES

**Classement à réaliser  
en fonction des besoins des  
commanditaires, du potentiel  
qu'ils souhaitent adresser**


# Cycle de vie d'un score « pérenne » *(et autres solutions ML)*

- Modélisation finie et mise en production passée :

- Le travail du data scientist est-il fini ?
- Le score une fois effectué a-t-il besoin de révision(s) ?

NON

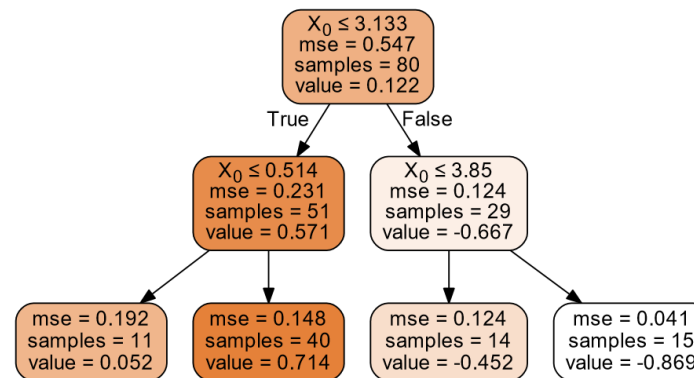
OUI

 **Un score dévie dans le temps** : d'autant plus que le marché est mouvant, que les comportements modélisés évoluent...ou que les données qui l'alimentent changent !

**Besoin d'établir un monitoring des scores qui suivra la déviance du score et alertera lorsqu'il faudra le refondre ou le réajuster !**

# Cas pratiques

- Réalisons une régression de type logistique
- Question : comment pourrions-nous repérer quelle fonction de Python est utile si je souhaite produire une régression ridge, elastic net ou autre ?
- Testons justement un autre type de classement
- Réalisons un arbre de décision : cherchez les fonctions avec la document de scikit-learn pour produire un arbre de décision de type « classifier »



# Moteurs de recommandation & Auto-apprentissage

*Et le temps réel !*

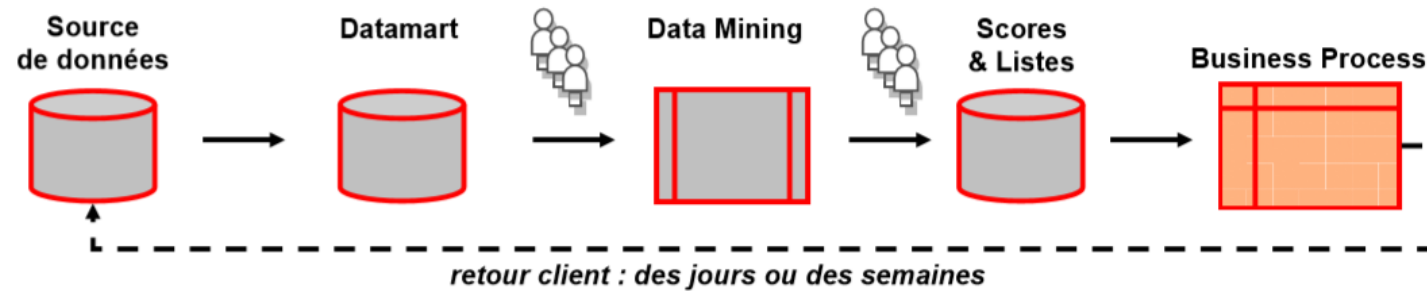
# Moteur de recommandations en temps réel : les avantages de l'auto-apprentissage

Illustration avec une solution existante d'Oracle, mais plusieurs autres solutions existent

Comment ?

*L'Intelligence Analytique intégrée au processus opérationnel sans délais*

## Apprentissage Traditionnel : Un Processus long et Manuel

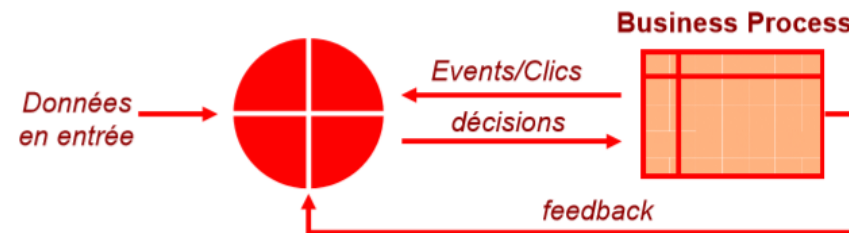


## Auto-Apprentissage : processus automatique et sans délais

### Avantages:

- Cycle de vie des Modèles auto-gérés
- Spectre d'analyse étendu
- Simple à implémenter et à maintenir

### Modèles Prédictifs



*Tuning automatique des hyperparamètres*

**Les modèles RTD automatisent l'intégration des retours clients**

Jusqu'à 80% de temps gagné



# Exemple de fonctionnement macro d'un moteur de recommandation pour un score efficace dès J+1

## Illustratif

**Inputs décisionnels / données accessibles seulement en temps réel**

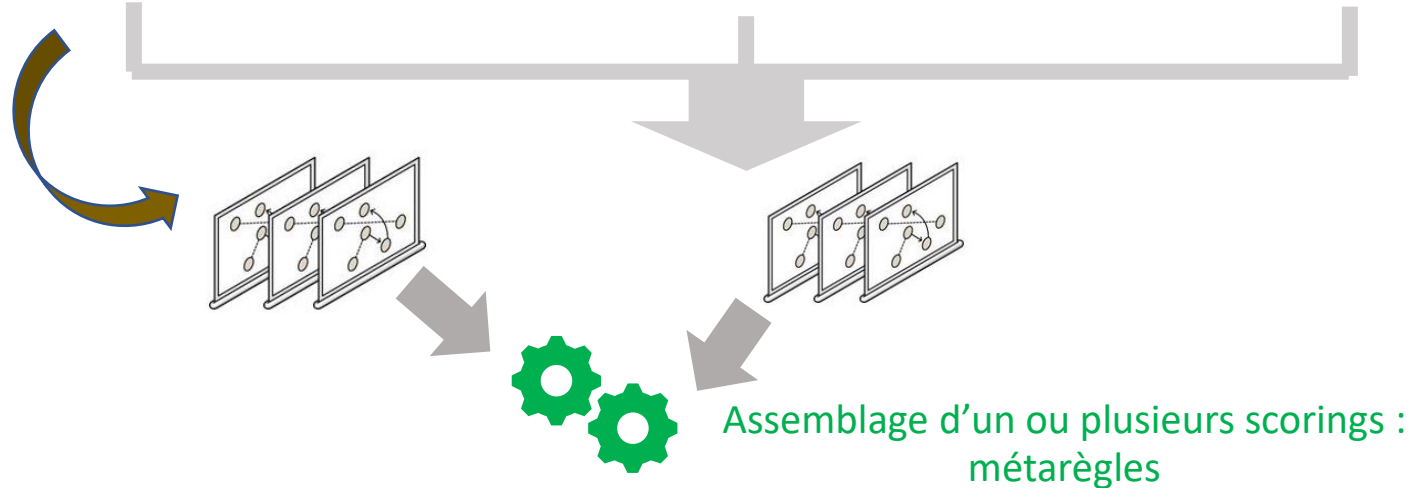
  
Données provenant des entrepôts décisionnels

  
Données des enseignes, du réseau, etc en provenance du système source x

  
Données de navigation en provenance des systèmes sources y

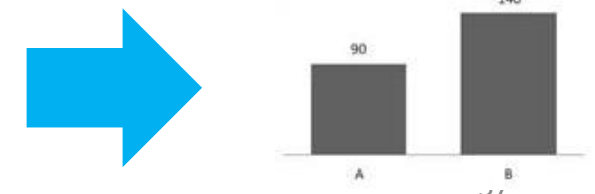
**Mécanisme de calcul de scores temps réel pour classer les clients en x niveaux de risque**

**Mécanisme d'assemblage des différents scores calculés pour aboutir à une note finale**



	HS	MS	BS
Règle 1	Flux 25%		
Règle 2	Flux 25%		
Règle 3	Flux 50%		

**Boucle d'apprentissage : mesure de l'incrément et test&learn**



# Moteur de recommandations en temps réel : exemple de la solution RTD d'Oracle

## RTD pour l'Optimisation des Décisions

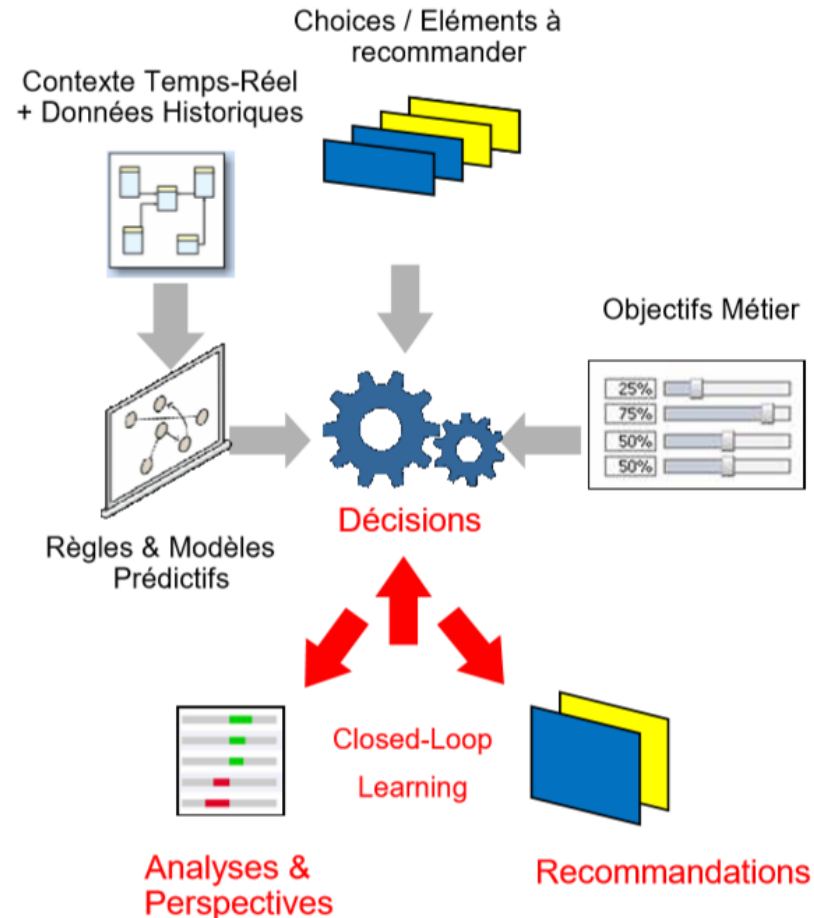
### Que font les Services de Décision RTD ?

#### Optimisent les processus métier en temps réel

- En exploitant les données historiques et temps réel
- Recommandent le meilleur produit et le meilleur message

#### Règles + Statistiques Prédicatives + Auto-Apprentissage

- Définir ce qui est "optimal" en fonction de multiples objectifs métier antagonistes
- Apprendre et prédire selon les retours client pour améliorer les résultats régulièrement



ORACLE

# Moteur de recommandations en temps réel : l'autoapprentissage

## Quelques cas d'usages

### Personnalisation Web

- Par intelligence statistique
- A/B testing en temps réel
- 1 to 1 digital marketing

### Marketing aux interactions entrantes

- Cross Sell / Up Sell / Rétention
- Recommandations de biens & services

**62% of consumers find them useful according to Forrester Research**

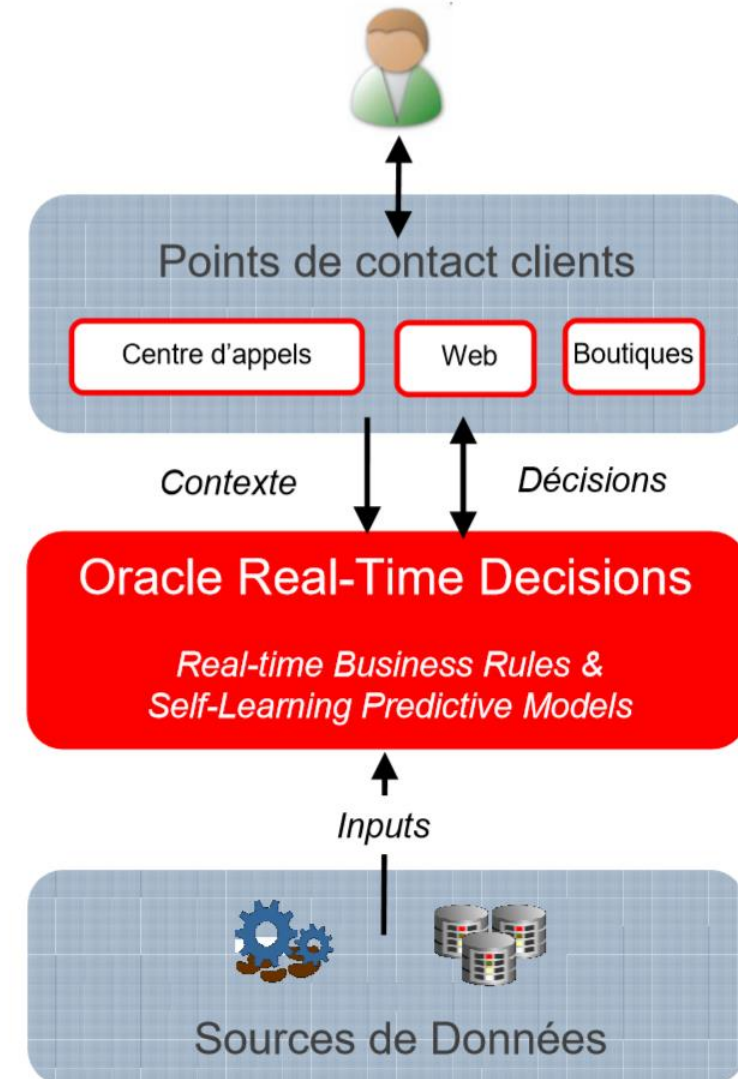
- Optimisation/personnalisation des Emails selon tous les apprentissages (visites en ligne, contenus investigués)

### Détection de fraude et processus préventifs

- Décisions selon scoring et calcul des probabilités de fraude

### Reporting en temps réel

- Analyses de la performance des décisions



# Moteur de recommandations en temps réel

## Personnalisation Web

*En temps record*

The screenshot shows the Belgacom website interface. Several elements are highlighted with red boxes and annotated with yellow speech bubbles:

- Top Navigation:** Includes links for "Television", "Packs and Promotions", "e-Services", and "Help".
- Main Banner:** Features a "Happy week" promotion (monday 7th June from 10:00) and a "Web exclusive offers" badge. A speech bubble asks: "RTD affine et ajuste le contenu personnalisé en capturant le comportement du Client (Clics)".
- Product Recommendations:** Below the banner, there are boxes for "Buy online and get a extra disc", "Which does?", and "Are you going to move?". A speech bubble asks: "Quels produits?".
- Service Categories:** Below the product recommendations, there are sections for "Telephone" and "Internet". A speech bubble asks: "Quel Contenu?".
- Right Sidebar:** Includes a "Log in" section, a "Direct access" section with links like "View my last", "Manage my", "Purchase", and "Check my", and a "Help" section with links like "Contact", "Points-of-sale", "Online help", "Organise your move", "Become a Belgacom customer again", and "Install Belgacom TV". A speech bubble asks: "Quelle Action proposer?".
- Search and Navigation:** At the top right, there is a search bar and a "belgacom" logo. A speech bubble asks: "Quelle Promotion?".

**Règles et Modèles déterminent le contenu optimal pour chaque interaction**

Source : Oracle

Tous droits réservés - 2019 -

Copyright ©2010, Oracle. All rights reserved.

# Exercice

# Choisir une méthode d'apprentissage supervisée : régression ou classification

Objectif : prédire la température à partir de variables continues connues

➡ **Régression (multiple)**

Objectif : prédire la survenue d'un tremblement de terre

➡ **Classification**

Objectif : détecter des usages frauduleux connus

➡ **Classification**

Objectif : prédire l'âge en fonction de caractéristiques connues

➡ **Régression**

Objectif : reconnaître des individus pouvant potentiellement acheter le produit Y

➡ **Classification**

Objectif : affecter chacun des individus à 4 équipes existantes

➡ **Classification**

ALLER PLUS LOIN

# INTRODUCTION TEXT MINING & NATURAL LANGUAGE PROCESSING

*“You can have data without information, but you cannot have information without data.”*

**Daniel Keys Moran**



# Textmining & Natural Language Processing (NLP) : qu'est-ce ?

- **Ensemble des techniques de traitement automatique de données textuelles en langage naturel reposant sur les domaines de la linguistique, de la sémantique, du langage, des statistiques et de l'informatique**
- **Ce qu'on appelle textmining fait plus référence à l'analytique et repose davantage sur l'utilisation de méthodes de Machine Learning plus classiques, et NLP sous-entend l'utilisation de méthodes de Deep Learning**



# NLP : objectifs

- **Objectifs : Dégager et structurer le contenu, repérer les thèmes récurrents, classer et affecter des documents, des verbatims...**

## Deux approches :

- **Descriptifs / Non Supervisé : dégager les contenus clés**
  - Rechercher de thèmes abordés dans un ensemble (corpus) de documents, sans a priori
  - Synthétiser un texte via la détection des mots clés les plus utilisés
  - Mettre en avant des informations clés issues des textes
- **Prédictifs / Supervisé : affecter les textes, les mots à une classe**
  - Détecter ou établir des règles permettant de catégoriser, d'affecter automatiquement un document à un thème, parmi plusieurs thèmes prédéfinis
  - Rechercher des informations précises sur la base de données textuelles « cibles »

# NLP : étapes souvent employées

- 1) Décomposer le texte comme une succession de phrases qui deviennent une succession de mots (« tokenise ») afin de structurer l'information pour le(s) traitement(s) : construire une matrice contenant les différents mots (uniques) = **matrice de sac de mots** (BOW = Bag Of Words)
- 2) Remplir la matrice de mots en calculant le **nombre d'occurrences de chacun des mots**
- 3) **Gestion des mots** : étape pouvant être relativement longue et fastidieuse à réaliser
  - **Etablir un dictionnaire des données** (réunir les mots ayant « une même racine » apportant une information similaire, les mots-composés, les synonymes, le vocabulaire spécifique au thème étudié...)
  - **Identifier une liste des mots non utiles / « vides »** (le, la...), de caractères spéciaux (regex\*)
- 4) **Conserver uniquement les mots sémantiquement saillants**

# NLP : Vocabulaire

- **Lemmatisation** : Analyse lexicale
  - Regroupement des mots d'une même famille dans un texte, afin de réduire ces mots à leur forme canonique (le lemme), comme petit, petite, petits, et petites. Certaines conjugaisons peuvent rendre cette tâche complexe pour des ordinateurs, comme retrouver la forme canonique «avoir» depuis «eussions eu». En revanche, « des avions » et « nous avions » n'ont pas le même lemme.
- **Racinisation** : Analyse sémantique
  - Regroupement des mots ayant une racine commune et appartenant au même champ lexical. Par exemple, pêche, pêcher, pêcheur ont la même racine, mais ni la pêche (le fruit), ni le péché, ne font partie du même champ lexical.
- **Désambiguïsation lexicale** :
  - Problème encore non résolu, consistant à déterminer le sens d'un mot dans une phrase, lorsqu'il peut avoir plusieurs sens possibles, selon le contexte général.
- **Étiquetage morpho-syntaxique** :
  - Assigne chaque mot d'un texte à sa catégorie grammaticale. Par exemple, le mot *ferme* peut être un verbe dans « il ferme la porte » et un nom dans « il va à la ferme ».

# NLP : Difficultés & Challenges

- Les difficultés du text-mining :
  - Gérer la langue (français, anglais...), les abréviations, les mots « inutiles » et très fréquents, les fautes d'orthographe....
  - Gérer les ambiguïtés des langues : polysémie des mots, homographes, ironie...



# NLP : techniques utilisables

- Plusieurs techniques de machine learning sont applicables sur un ou des jeux de données textuelles :
  - Méthodes de classement : Arbres de Décision, SVM...
  - Réseaux de Neurones
  - Méthodes d'analyses factorielles
  - ....
- Possibilité d'utiliser seulement des mots clés déclencheurs si on sait d'avance ce qu'on cherche :
  - Exemple : recherche de tous les mots clés de type « intelligence artificielle »

# NLP : Quels supports, quelles données ?

De nombreuses données peuvent nécessiter le recours à l'usage de techniques de type textmining :

- **Tout fichier texte ou données textuelles détenus** : courriers, lettres de réclamation, e-mails adressés au Service Client afin d'analyser les motifs, les enquêtes d'opinion, les appels téléphoniques retranscrits en texte...
- **Récupération de contenu textuel sur internet** (Web Scraping – BeautifulSoup) : avis clients, articles, FAQ, ....



**Rester dans la légalité au niveau de l'exploitation des contenus textuels, notamment sur internet : s'assurer d'un droit d'utilisation**

# NLP : Quelques cas d'usages



## Classement et affectation de documents

- Filtres anti-spam
- Redirection de CV, de mails...
- Sélection de newsletters



## Rechercher et extraire des informations

- Repérer les thèmes les plus abordés dans les discussions
- Détection des centres d'intérêts des personnes
- Détection de mots clés utilisés par les visiteurs afin d'utiliser le même vocabulaire qu'eux



## Analyse de sentiments

- Avis clients
- ...

Synthèse des recommandations délivrées par Villani :  
*quels sont les 20 à 30 mots significatifs les plus utilisés  
dans le texte ?*

<https://www.aiforhumanity.fr/>



Synthèse des recommandations délivrées par Villani –  
rapport complet :

*Pourriez-vous retrouver les 20 à 30 mots les plus utilisés  
pour chacun des 5 secteurs déroulés dans le rapport  
complet ?*

### 30 mots significatifs les plus fréquents dans le texte :

ia	niveau
données	permettre
acteurs	nécessaire
recherche	numérique
intelligence	penser
artificielle	écologique
développement	partie
publique	économiques
doit	chercheurs
valeur	professionnelle
pourrait	autour
innovation	notamment
formation	état
place	Développer
doivent	mission

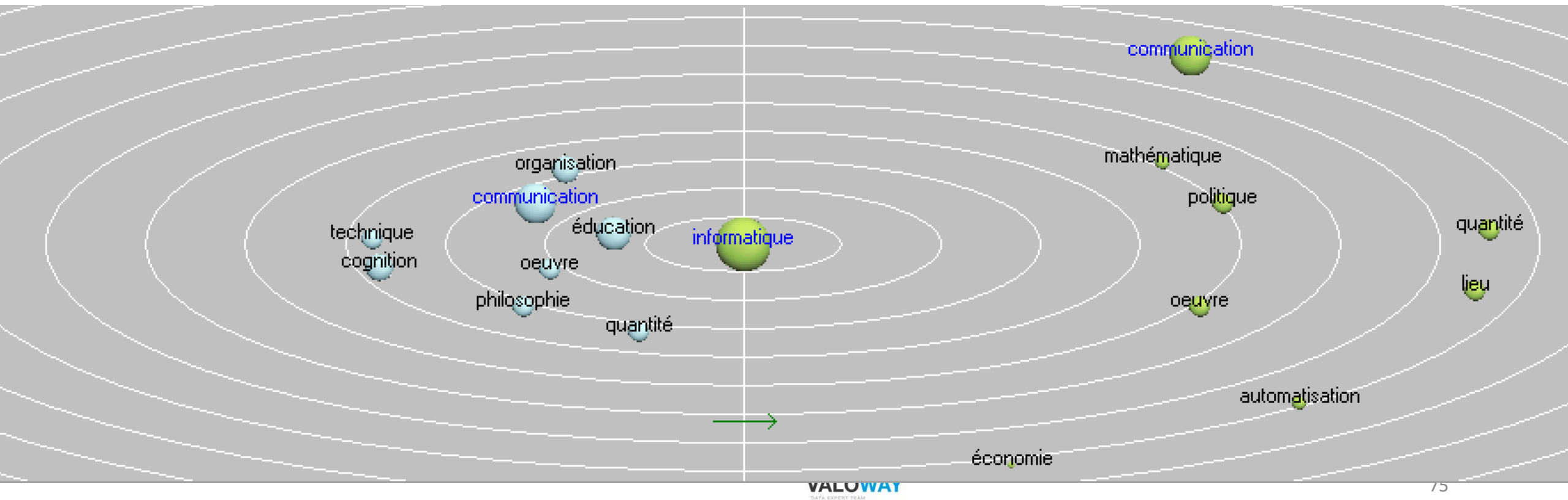


# NLP : cas pratique avec Tropes

- **Info utile** si besoin d'analyser des textes rapidement sans coder :

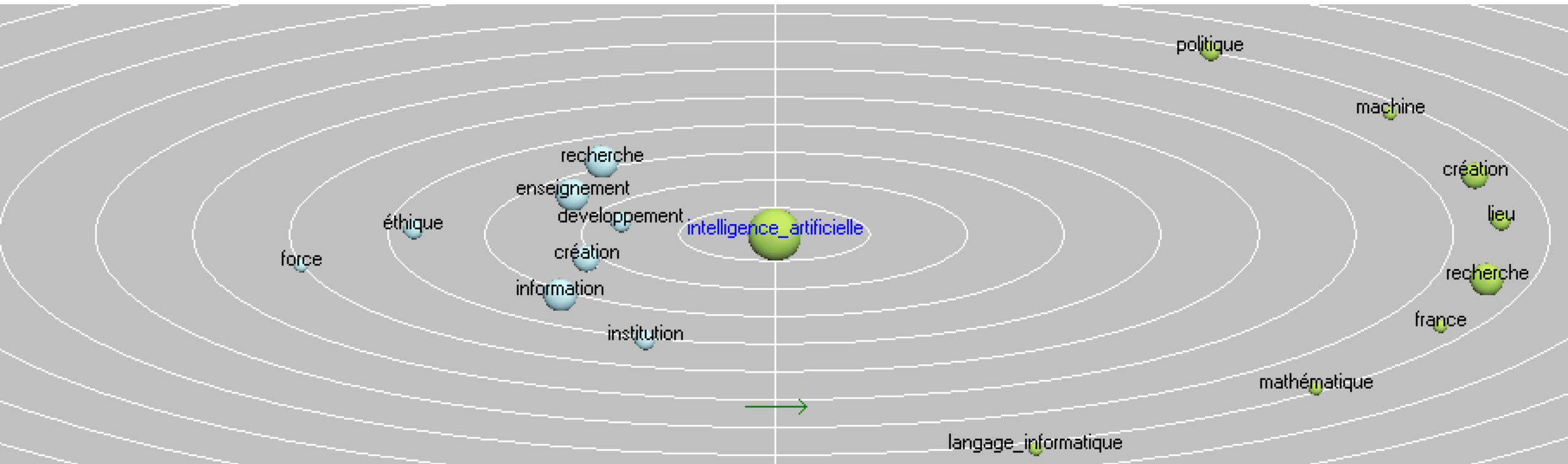
➔ **Logiciel Tropes : permet une analyse rapide et complète**

**Délivre des univers de référence** : dans notre exemple, les recommandations listées par Villani (synthèse) font référence (sans surprise) à l'informatique, notion utilisée assez souvent suite à des notions d'éducation, de communication, d'organisation....



# NLP : cas pratique avec Tropes

**Univers de référence de niveau 2 :** Tropes indique que la première notion de référence est l'intelligence artificielle souvent liée et précédée des notions de développement, de création, d'enseignement, de recherche...



### 30 mots significatifs les plus fréquents dans le texte :

apprentissage	orientation
ia	intelligence
pédagogique	artificielle
données	dispositifs
edtech	développer
élèves	expérimentations
pédagogiques	acteurs
enseignants	nouvelles
ministère	parcours
éducation	apprenants
apprenant	éducatifs
transformer	pratiques
temps	politiques
développement	effet
enseignement	équipe
éducatives	





# NLP : cas pratique avec Python – Sante

**30 mots significatifs les plus fréquents dans le texte :**

santé  
données  
ia  
recherche  
intelligence  
artificielle  
médicales  
patient  
professionnels  
système  
informations  
heure  
médical  
innovation  
patients  
accès

usages  
médecine  
effet  
service  
cliniques  
temps  
fins  
politiques  
améliorer  
médicale  
production  
dossier  
publique  
terme  
France



### 30 mots significatifs les plus fréquents dans le texte :

données	exploitations
agriculture	exploitants
agricoles	ia
faire	numériques
agricole	augmentée
france	économie
pourrait	développement
exemple	agroalimentaire
agriculteurs	ouverture
capacités	effet
secteur	préserver
ensemble	terme
valeur	soutenir
innovation	émerger
doit	recherche
démarches	



# NLP : cas pratique avec Python – Transports

**30 mots significatifs les plus fréquents dans le texte :**

données	ia
Secteur	services
acteurs	autour
développement	rupture
transport	doit
niveau	exemple
autonome	cadre
européen	autonomes
mobilité	également
politique	france
véhicule	recherche
véhicules	certain
pourrait	domaine
innovation	nouveaux
transports	publics
	plateformes

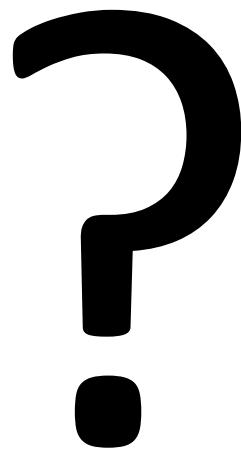




### 30 mots significatifs les plus fréquents dans le texte :

ia	cadre
données	applications
systèmes	opérationnels
sécurité	particulier
défense	afin
domaine	recherche
techniques	expérimentation
notamment	bien
exemple	missions
développement	armées
mise	cas
ministère	permettre
faire	œuvre
contexte	place
information	service
technologies	





**MERCI POUR VOTRE  
ATTENTION**

# Liste des liens utiles

- Kaggle : <https://www.kaggle.com/>
- MOOC's :
  - Codecademy, Udemy, Coursera, ...
  - Python : <https://www.codecademy.com/learn/learn-python-3>
  - Data Science :  
<https://www.coursera.org/learn/machine-learning>  
<https://www.codecademy.com/learn/paths/data-science>
- Cours ou documentation d'enseignants-chercheurs experts sur ce domaine :
  - [http://eric.univ-lyon2.fr/~ricco/cours/cours\\_programmation\\_python.html](http://eric.univ-lyon2.fr/~ricco/cours/cours_programmation_python.html)
- Documentation des librairies sous Python :
  - <https://scikit-learn.org/stable/index.html>

# ANNEXES

# Pour continuer l'acculturation à l'IA, Stéphane Mallat

---

L'intelligence artificielle : un enjeu scientifique?



# Pour continuer l'acculturation à l'IA, Jean-Gabriel Ganascia

Ep 40 - Faut-il craindre l'intelligence artificielle et la robotique ? Avec Jean-Gabriel Ganascia



*Ganascia à Rennes :*

<https://www.youtube.com/watch?v=J1eB1K2VIOA>

