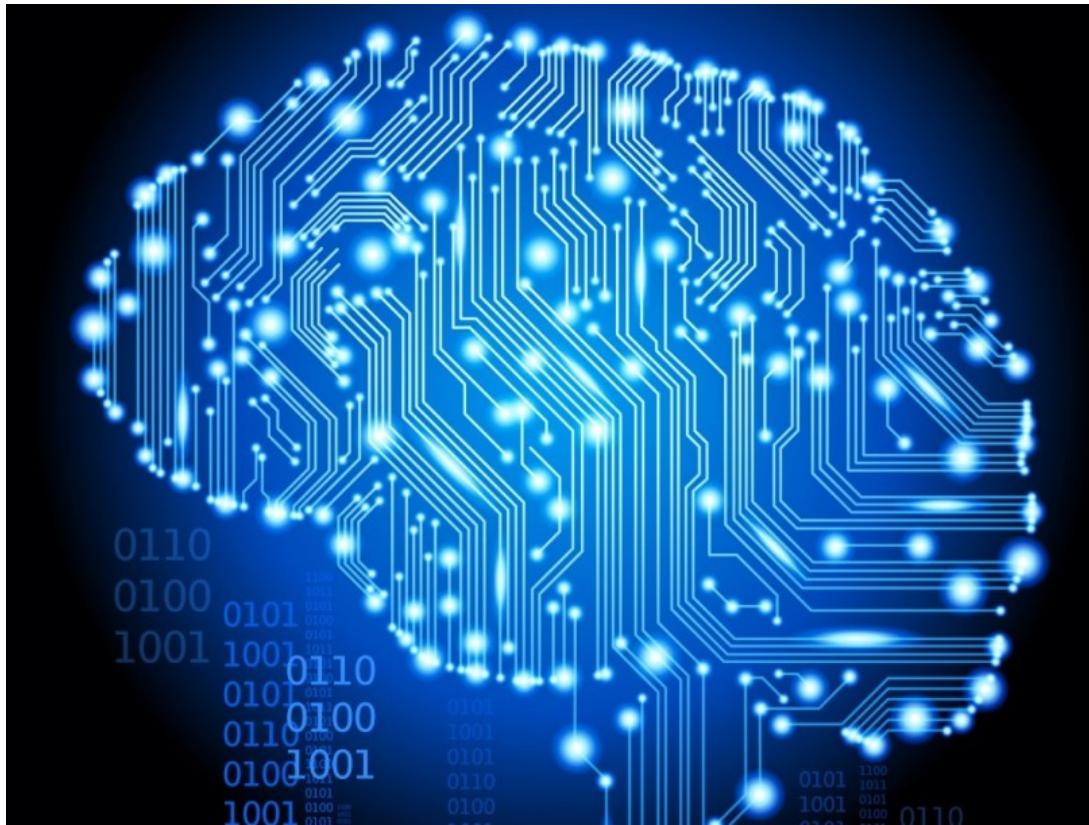


Synthèse du cours CNAM-STA211



Dépôt: https://gitlab.com/logrus_fr/CNAM-projets/

Auteur: Fabrice DUNAN <fabrice.dunan@laposte.net>

Tuteurs: N.Dieye <ndeye.niang_keita@cnam.fr> E.Jacubovitz <ej@stat4decision.com>

Table des matières

1. INTRODUCTION.....	3
2. LE "BIG DATA" ET L'ÉVOLUTION DE SON ANALYSE.....	4
2.1. Le "Big Data" et son contexte.....	4
2.2. Historique de la problématique, des méthodes et outils.....	7
2.3. Le "Data Mining".....	7
2.4. Les outils du DataScientist.....	8
3. PROCESSUS DU PROJET D'ANALYSE DE DONNÉES MASSIVES.....	9
4. ÉTUDES STATISTIQUES PRÉALABLES A L'ANALYSE DE DONNÉES MASSIVES.....	10
4.1. La statistique.....	10
4.2. Étude unidimensionnelle.....	12
4.3. Étude bidimensionnelle.....	13
5. ALGORITHMES D'APPRENTISSAGE AUTOMATIQUE.....	14
5.1. Famille non supervisée.....	15
5.1.1. Analyse factorielle.....	17
5.1.2. Classification / "clustering".....	18
5.2. Famille supervisée.....	22
5.2.1. Méthodologie autour des algorithmes supervisés.....	23
5.2.2. Classement.....	24
→ Modèles génératifs.....	24
→ Modèles discriminants.....	24
→ Modèles discriminants par fonction.....	24
→ Arbres de décision.....	25
5.2.3. Régression.....	25
→ Régression multilinéaire.....	25
5.3. Autres méthodes.....	26
5.3.1. Méthodes "ensemble".....	26
5.3.2. Méthodes duales.....	27
5.3.3. Réseaux de neurone.....	28
→ Perceptron.....	28
→ Réseau de neurone à rétro propagation de l'erreur.....	28
5.3.4. Deep Learning et réseaux de neurones.....	29
ANNEXES.....	30
Bibliographiques / web.....	30
Tableau synoptique des caractéristiques des algorithmes.....	30

1. INTRODUCTION

Ce document constitue ma synthèse des enseignements du cours CNAM-STA211 reçus entre octobre 2017 et mars 2018.

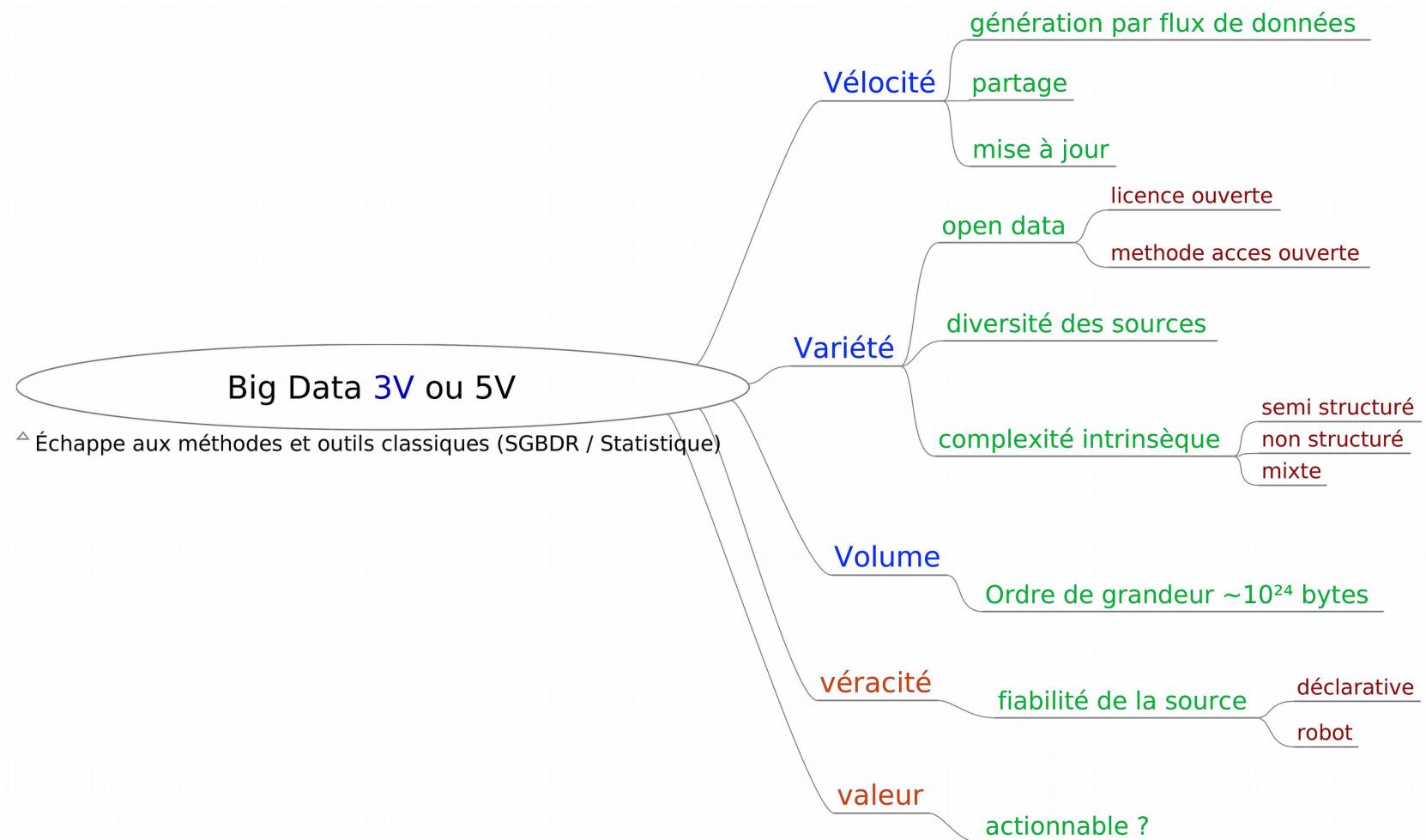
Cette synthèse suit globalement le plan du cours. Après une courte **introduction** sur le "big data", la **première partie** donnera un résumé de la méthode des projets d'analyse de données massives, la **seconde partie** décrira brièvement la phase préparatoire à l'exécution des algorithmes par l'usage des outils statistiques, la **troisième partie** exposera les différentes méthodes d'usage et de validation des algorithmes et enfin on résumera dans une **quatrième partie** l'ensemble des algorithmes par grandes familles et caractéristiques.

A chaque fois que possible, le sujet sera présenté sous forme de carte mentale [\[MM\]](#).

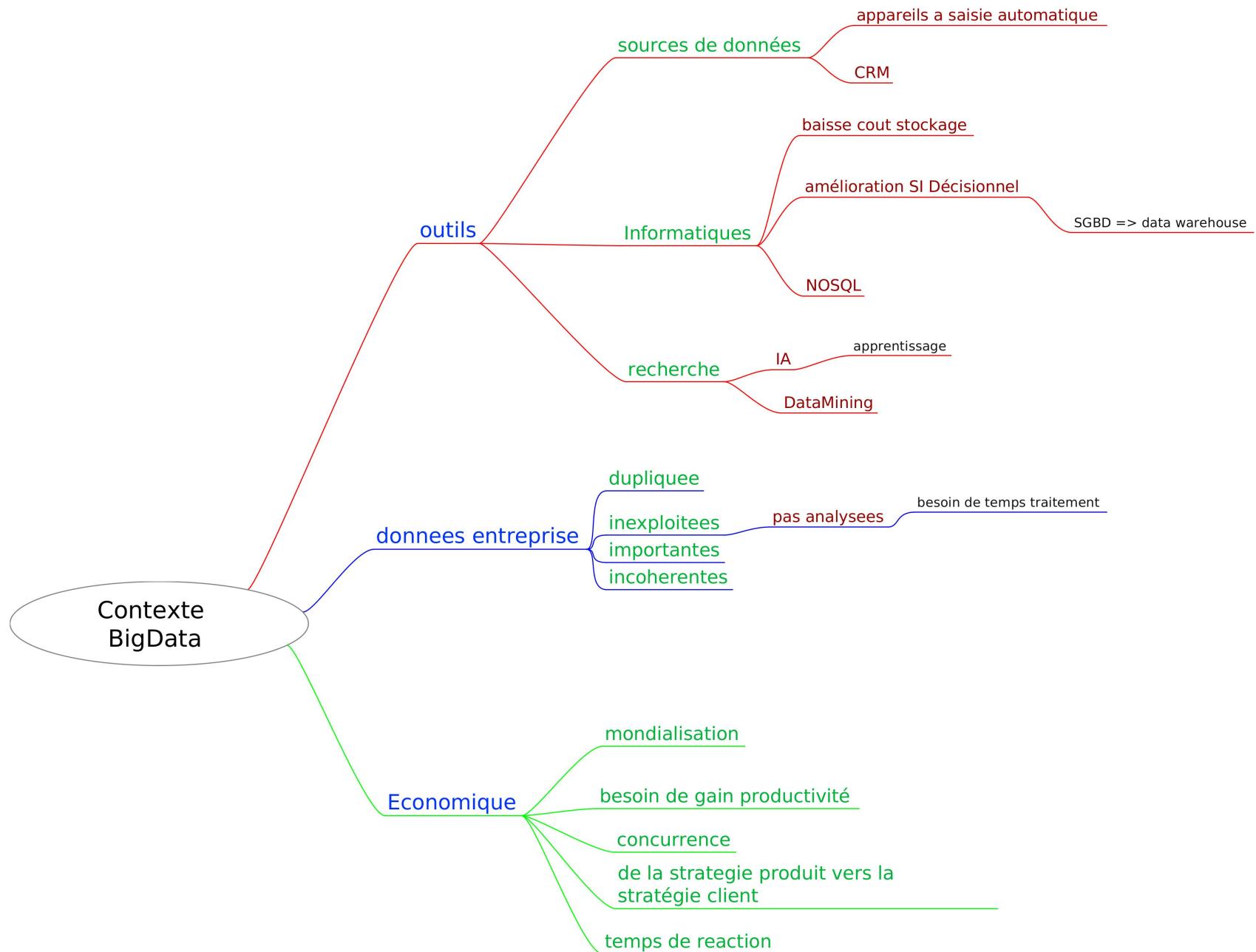
2. LE "BIG DATA" ET L'ÉVOLUTION DE SON ANALYSE

2.1. Le "Big Data" et son contexte

→ Schéma illustrant la problématique "Big Data"



→ Schéma illustrant le contexte de la problématique "Big data"



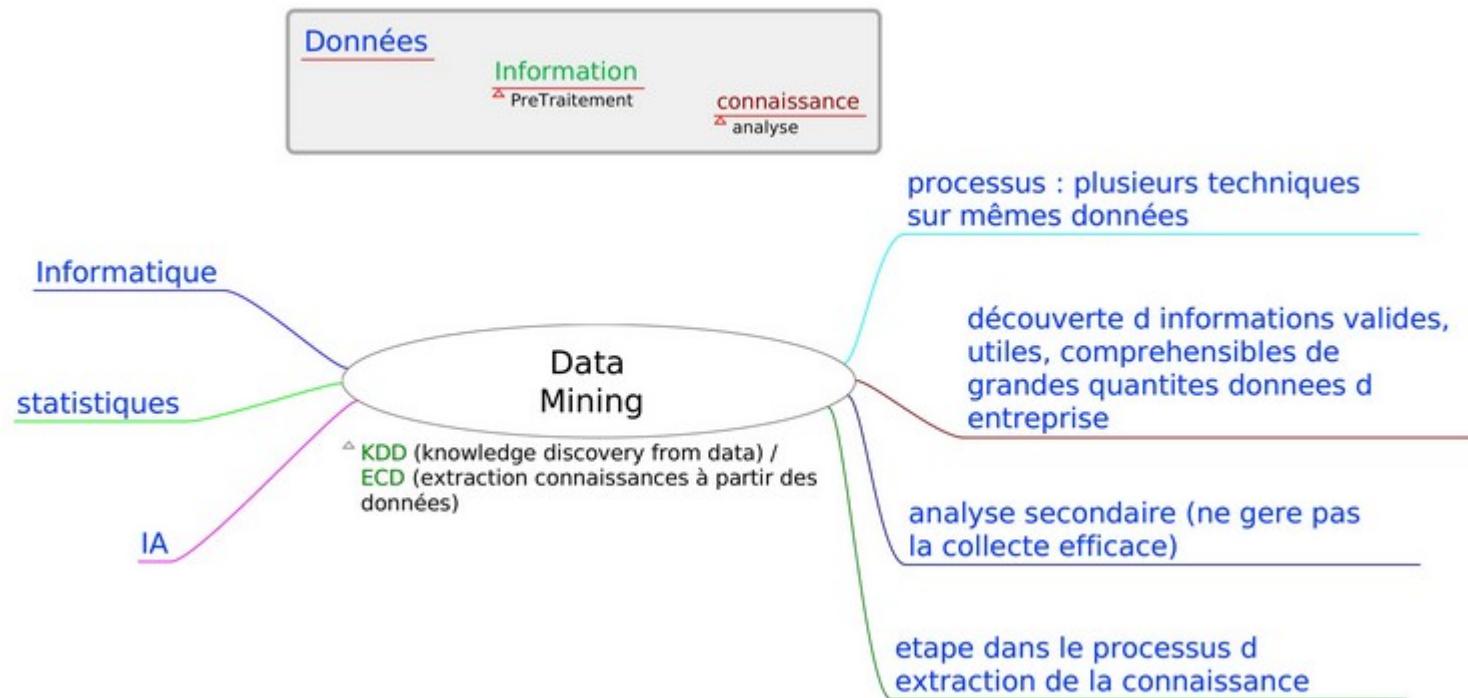
2.2. Historique de la problématique, des méthodes et outils

L'évolution du traitement des données massives a suivi plusieurs étapes démarrant en 1970 avec les méthodes statistiques classiques pour aboutir à l'explosion "Big Data" des années 2010.

Les problématiques outils et méthodes ne datent pas de la mode "Big Data" mais ont plutôt suivi une évolution continue.

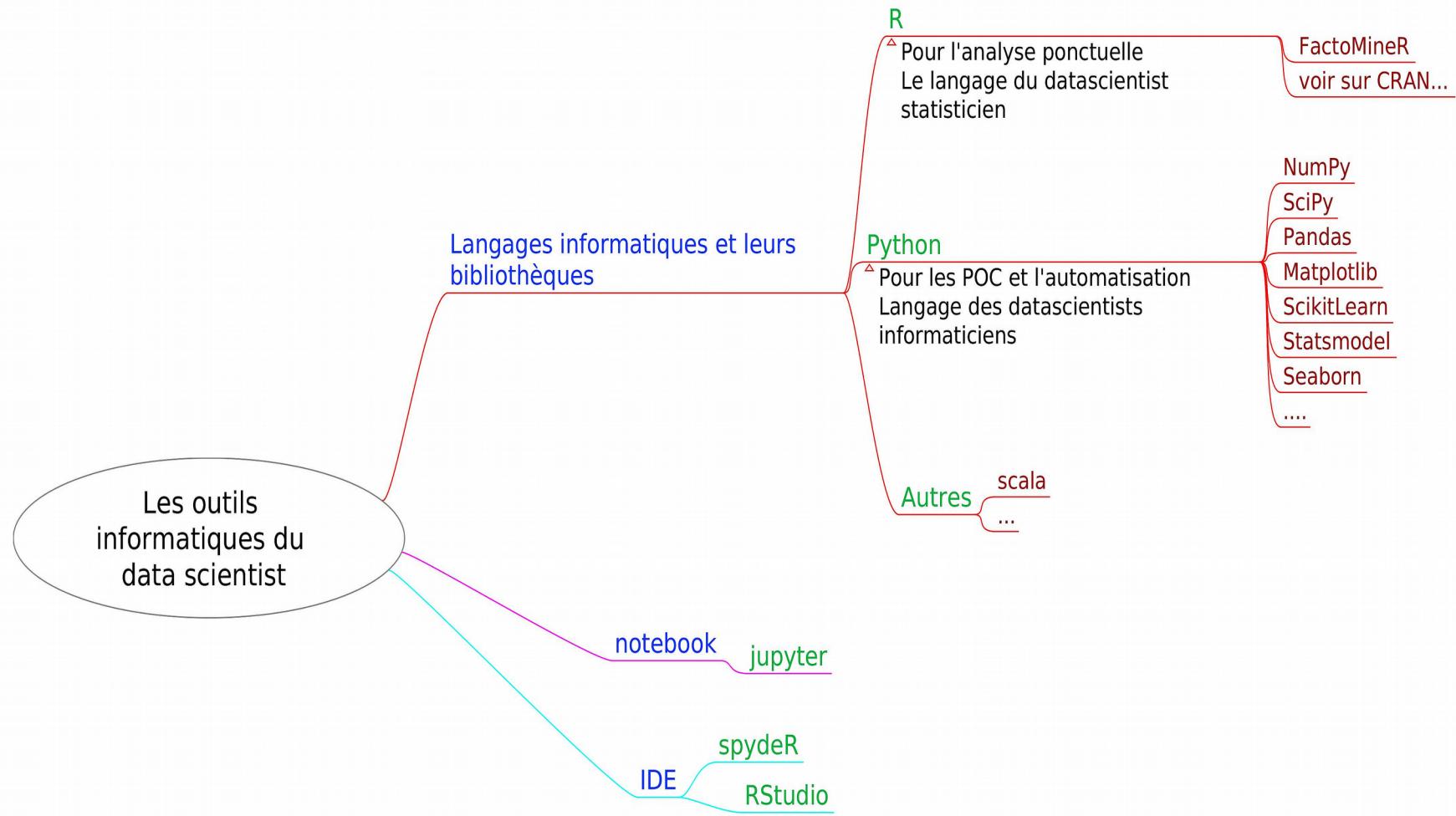
2.3. Le "Data Mining"

→ Qu'est ce que le "Data Mining" ?



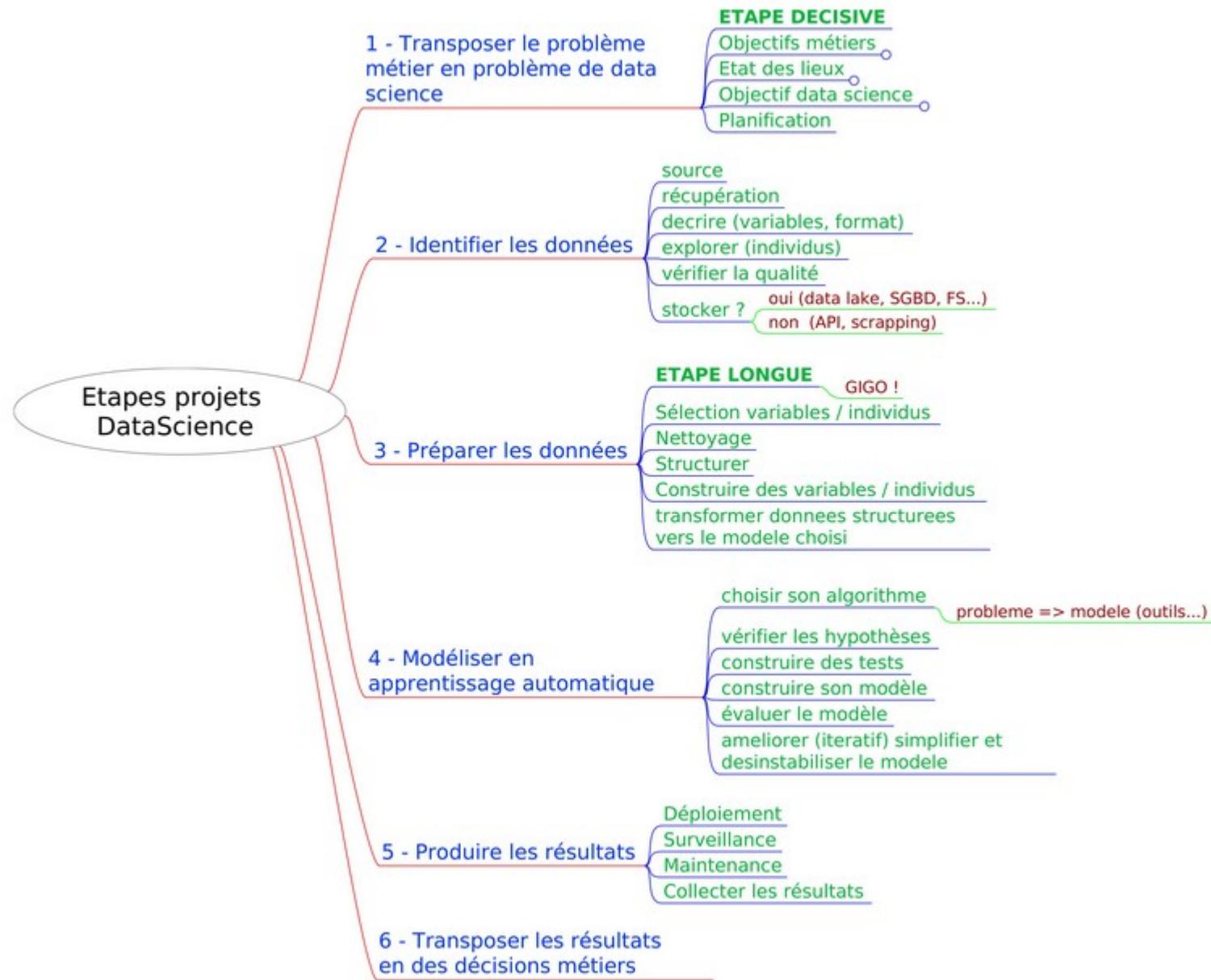
2.4. Les outils du DataScientist

→ Les langages bibliothèques et IDE



3. PROCESSUS DU PROJET D'ANALYSE DE DONNÉES MASSIVES

→ Schéma des grandes étapes d'un projet d'analyse de données massives



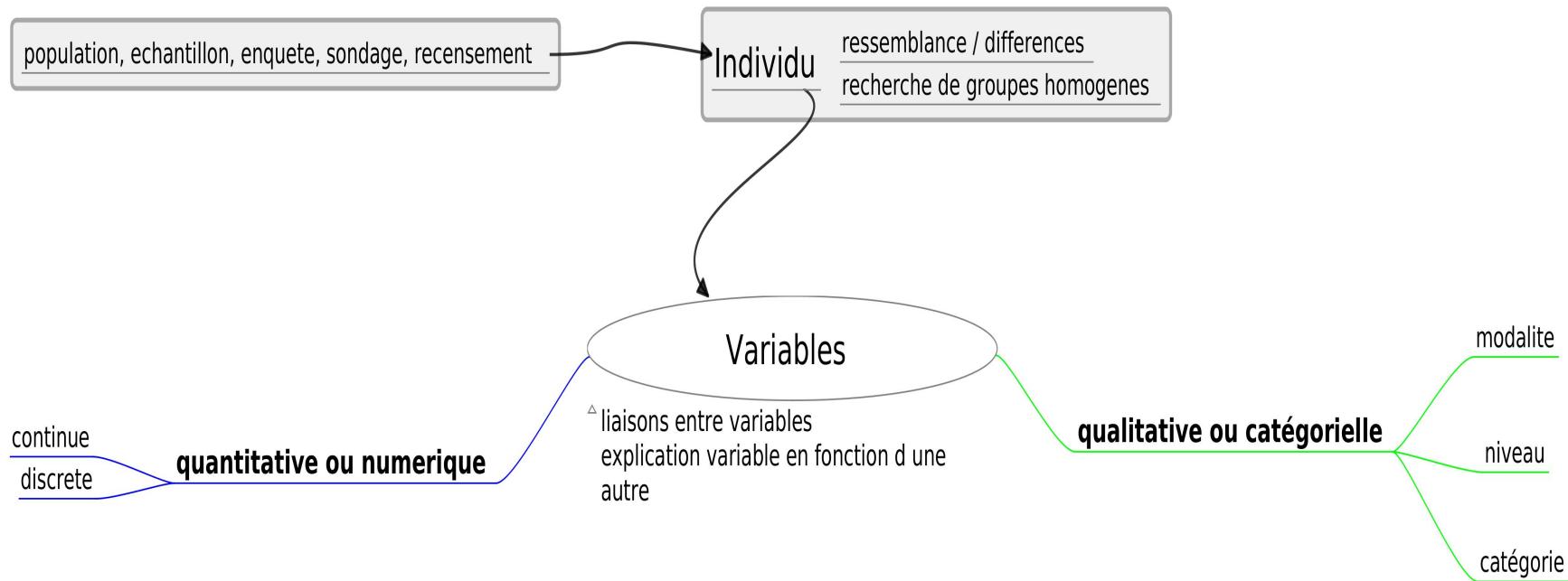
4. ÉTUDES STATISTIQUES PRÉALABLES A L'ANALYSE DE DONNÉES MASSIVES

4.1. La statistique

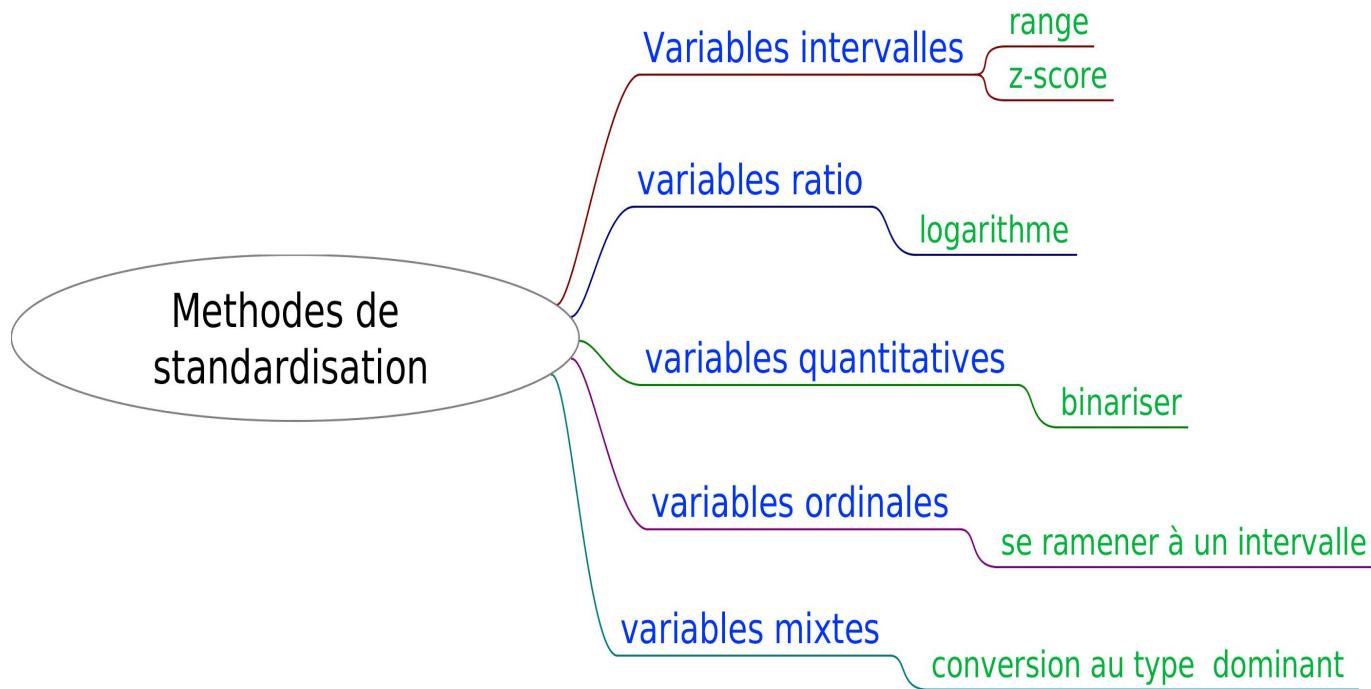
Définition:

La statistique, au travers de son aspect inférentiel, fournit des échantillons, permet d'estimer les paramètres algorithmiques et enfin permet de tester les hypothèses formulées. Pour décrire les données, elle fournit des résumés numériques, des graphiques et des tableaux.

→ Les variables et les individus en statistique

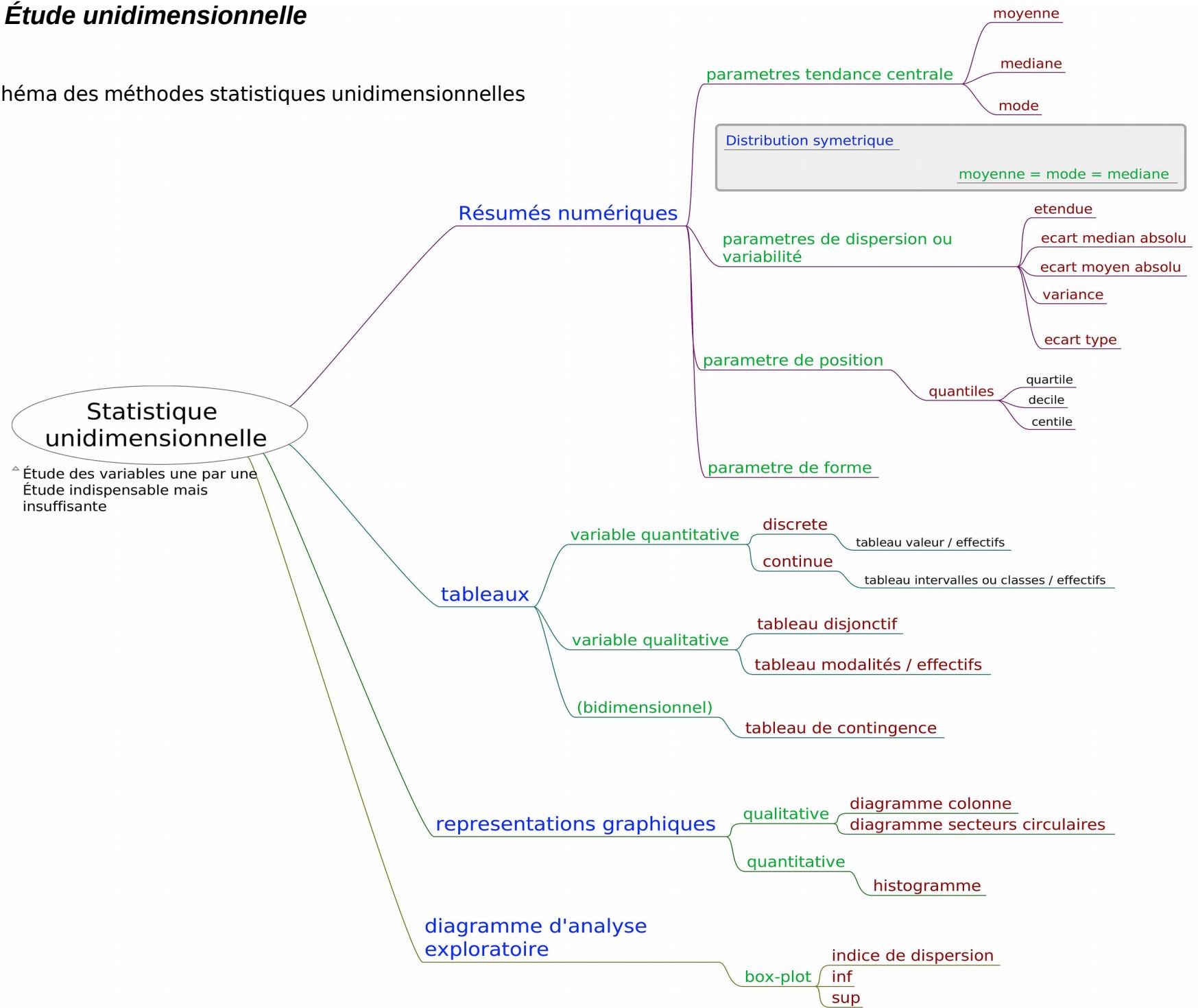


→ Méthodes de standardisation de la donnée



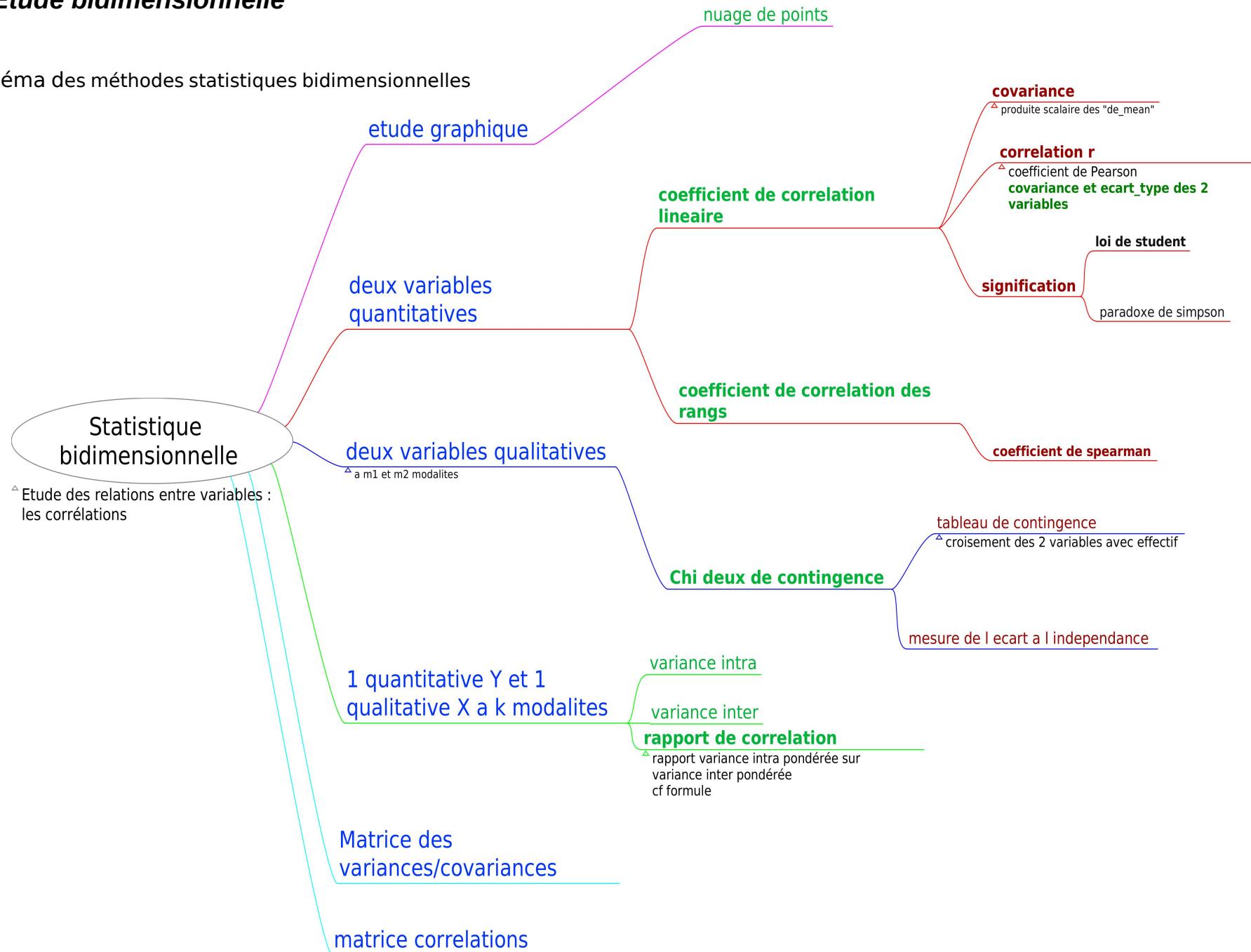
4.2. Étude unidimensionnelle

→ Schéma des méthodes statistiques unidimensionnelles



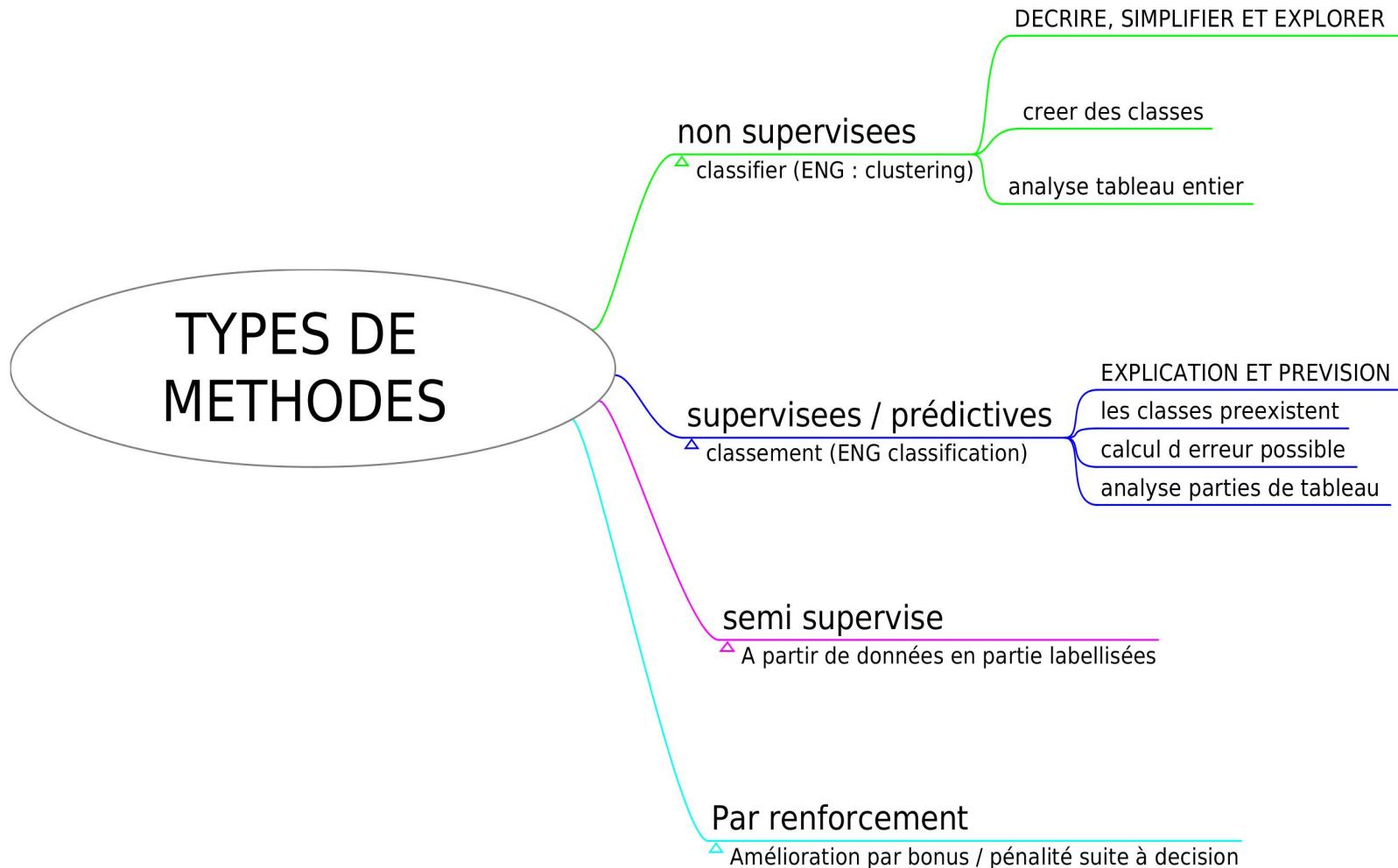
4.3. Étude bidimensionnelle

→ Schéma des méthodes statistiques bidimensionnelles



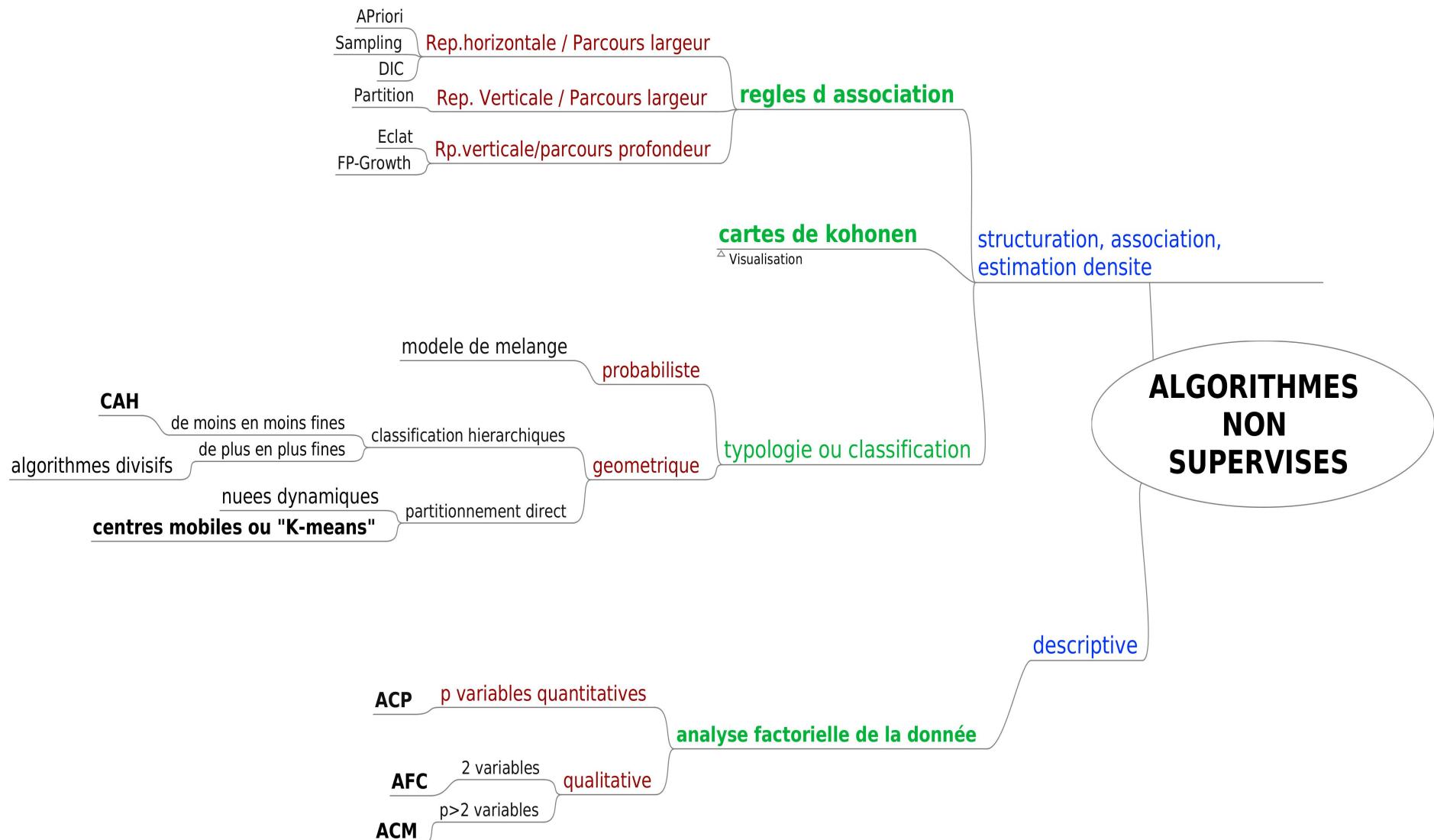
5. ALGORITHMES D'APPRENTISSAGE AUTOMATIQUE

→ Les grands types de systèmes d'apprentissage automatique:



5.1. Famille non supervisée

→ Les algorithmes principaux et les conditions d'usage relatives à la nature de la donnée



5.1.1. Analyse factorielle

Objectif :

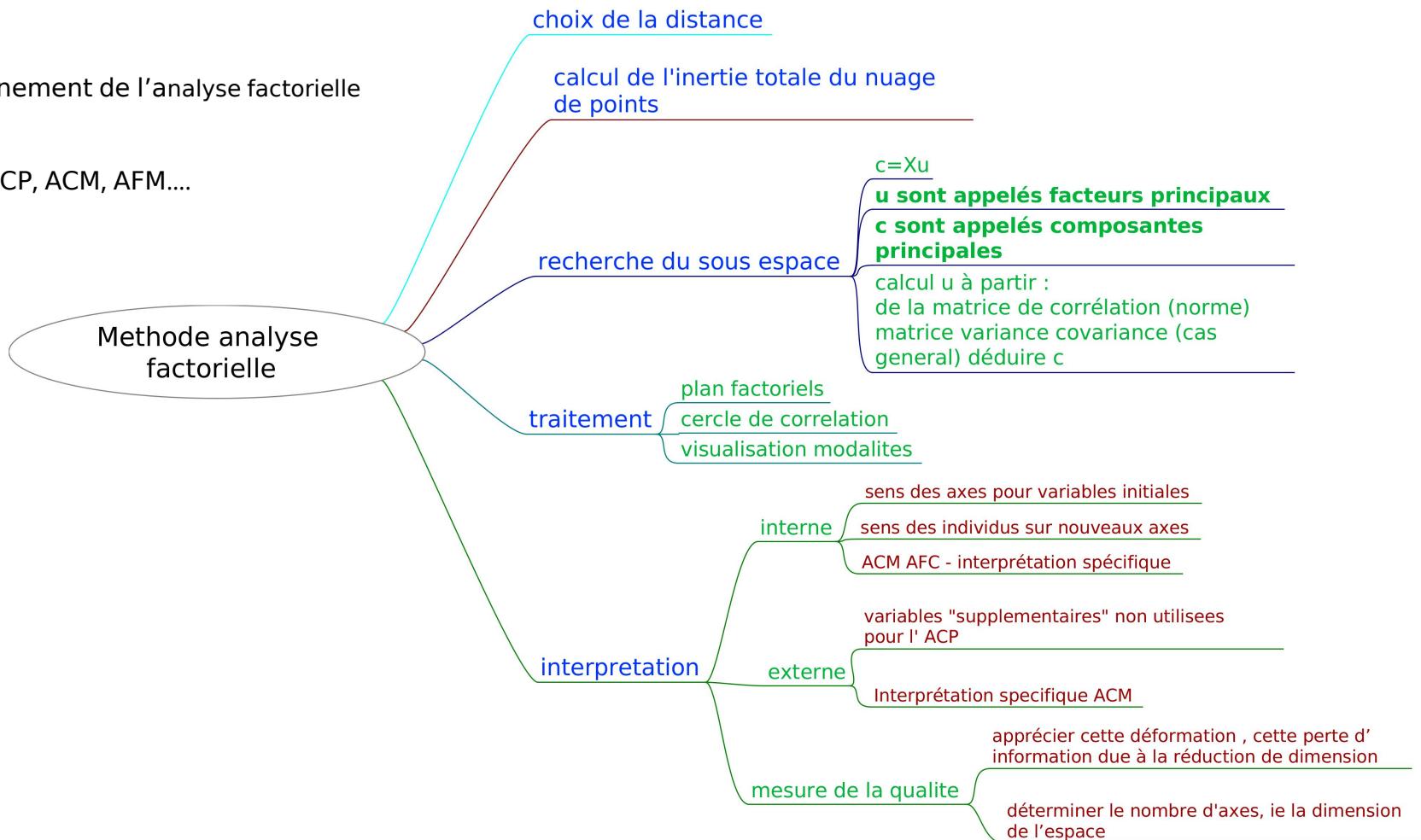
- réduction du nombre de variables

Principes :

- trouver un sous espace de dimension faible pour mieux "voir" les individus
 - trouver de nouvelles variables combinaisons linéaires des variables initiales conservant le maximum de l'information du nuage initial
- ⇒ ces nouvelles variables contiennent les coordonnées (ou projections) des individus dans le nouvel espace

→ Fonctionnement de l'analyse factorielle

exemple: ACP, ACM, AFM....



5.1.2. Classification / "clustering"

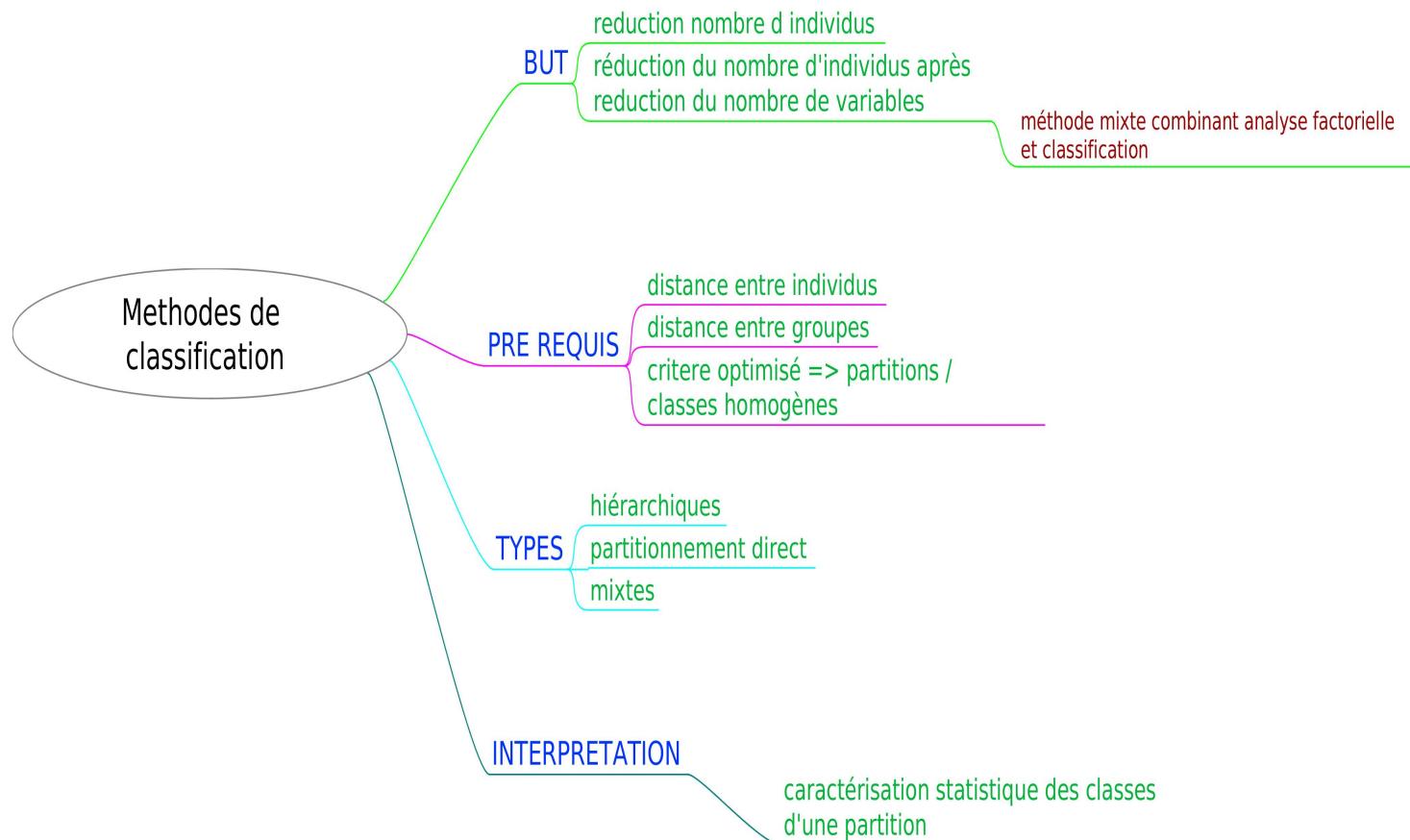
La définition de "classifier" ("clustering" en anglais) est la production de classe, de groupe.

Objectif :

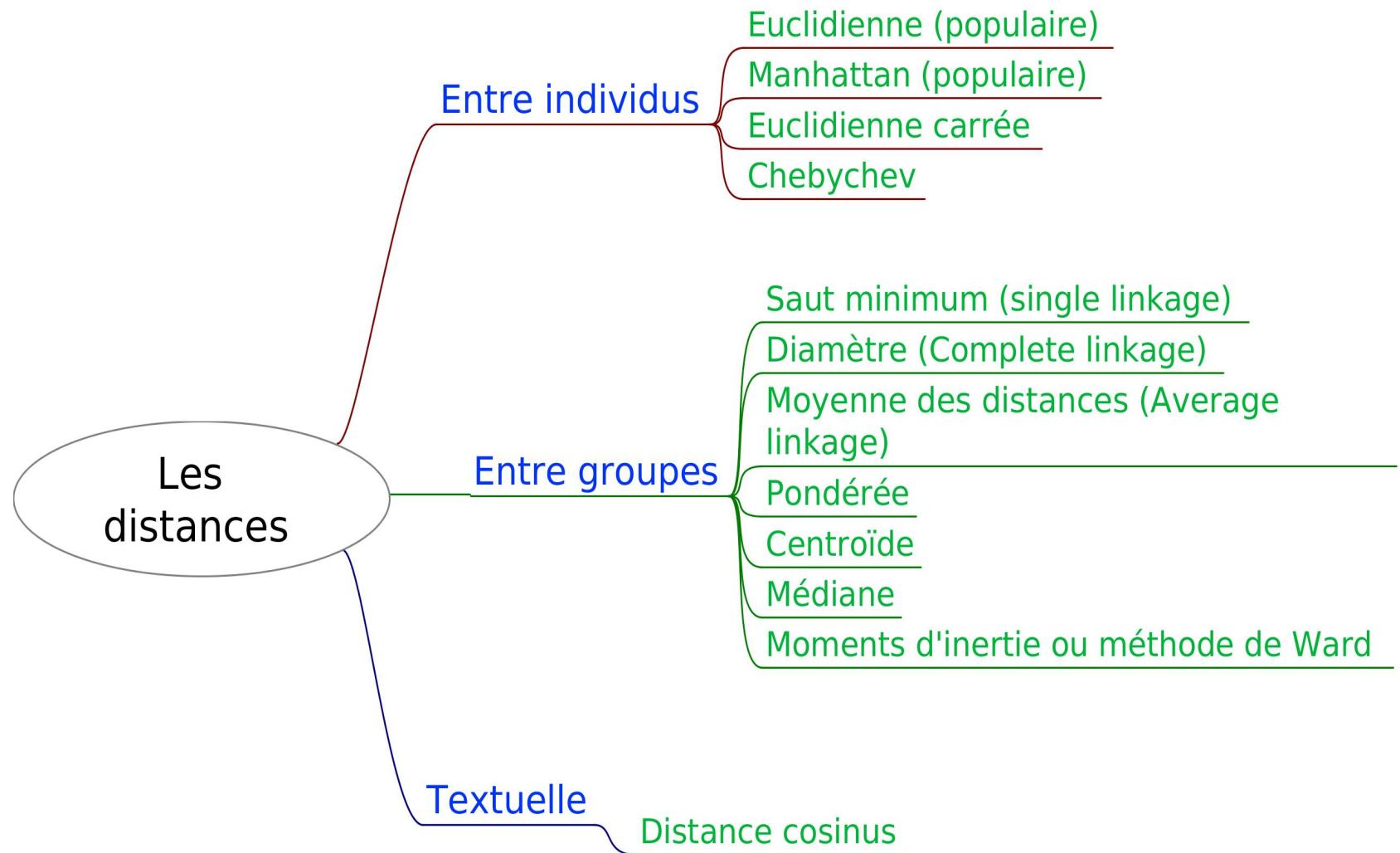
- réduction du nombre d'individus

Éléments communs:

→ Généralités sur les méthodes de création de classes



→ Distances utilisées dans les méthodes de création de classes



Méthodes de partitionnement

Objectif :

Les algorithmes visent à obtenir un partitionnement des données, plus éventuellement de trouver une donnée « représentative » (« prototype ») par groupe.

*Exemple : **k-means(ou centres mobiles)***

Méthodes hiérarchiques

Objectif :

Créer un hiérarchie de regroupements, qui fournit une information plus riche concernant la structure de similarité des données. Noter qu'à partir d'une telle hiérarchie, il est facile d'extraire plusieurs partitionnements, à des niveaux de « granularité » différents.

Exemple : CAH

Méthodes mixtes

Objectif :

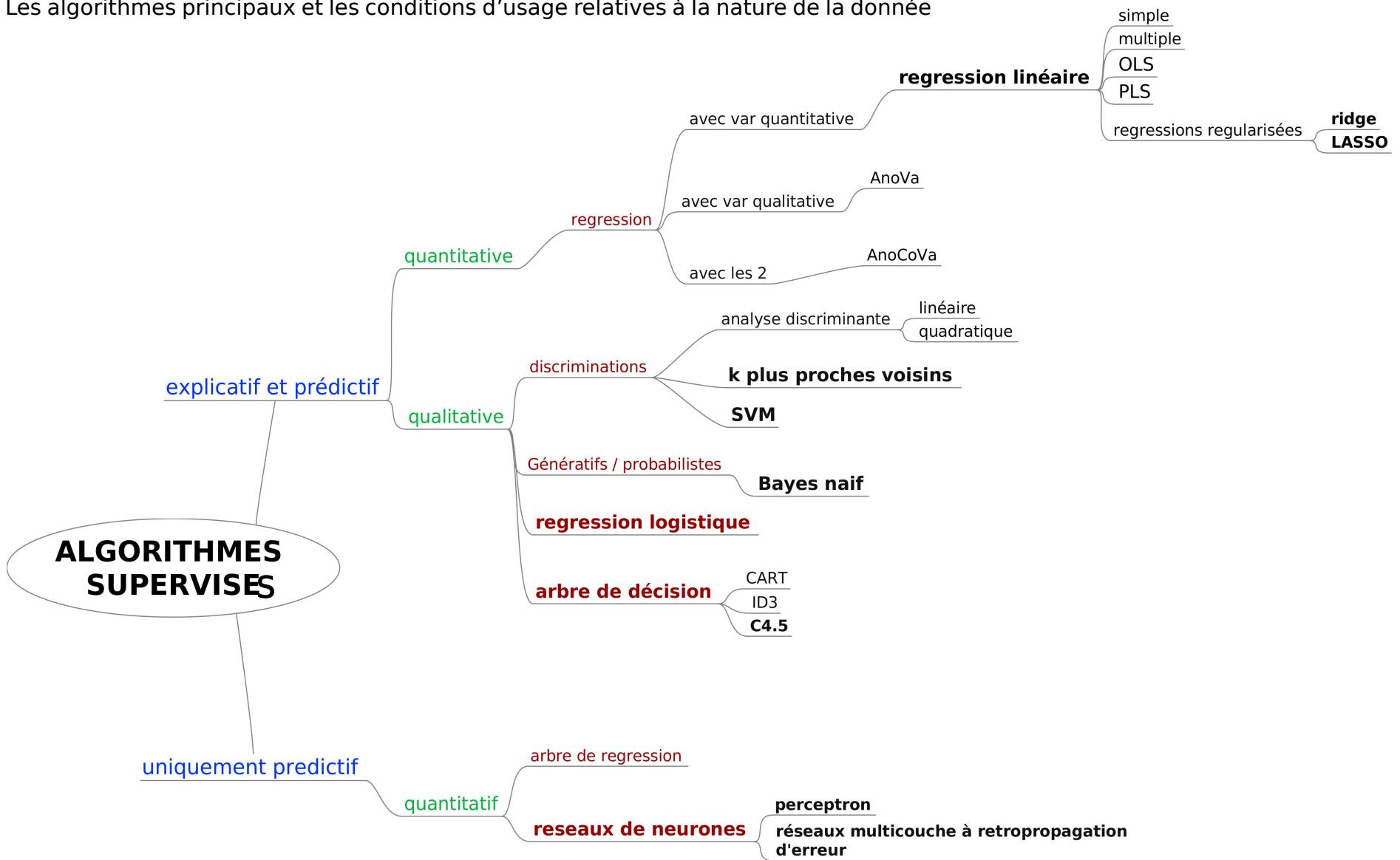
Dépasser les limitations des algorithmes classiques face à un nombre d'objets importants en enchaînant les différents types d'algorithmes.

*Exemple: **k-means + CAH sur groupes obtenus + consolidation par k-means.***

5.2. Famille supervisée

Le classement ("classification" en anglais) est l'action qui vise à trier les individus dans les classes produites par la classification. C'est la démarche des méthodes non supervisées.

→ Les algorithmes principaux et les conditions d'usage relatives à la nature de la donnée

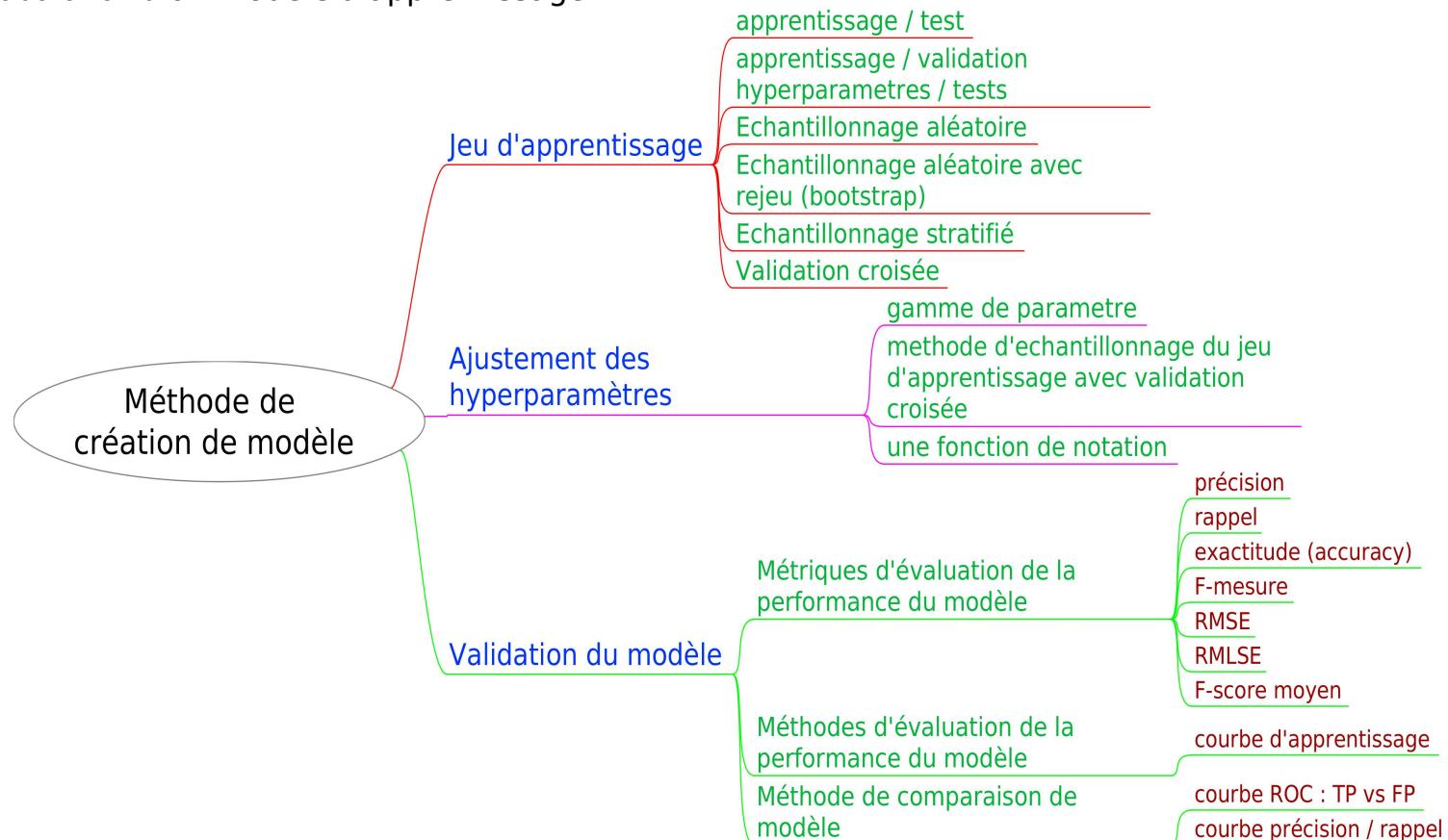


5.2.1. Méthodologie autour des algorithmes supervisés

Étapes d'utilisation d'un algorithme supervisé et contraintes sur les données :

1. Construction du modèle
Constitution des données d'apprentissage
2. Évaluation du modèle
Constitution des données de test (distinctes des précédentes)
3. Utilisation du modèle sur des données futures

→ Méthode d'élaboration d'un modèle d'apprentissage



5.2.2. Classement

L'objectif est l'apprentissage d'une fonction discrète : On veut prédire une classe. Ces classes peuvent être binaires ou multiples. Certains algorithmes peuvent nativement prédire des multi-classes, d'autres uniquement binaires peuvent s'y ramener.

→ Modèles génératifs

On cherche la probabilité qu'une donnée x soit dans une classe C_k :

- * Calculer $p(x|C_k)$ et $P(C_k)$
- * utiliser le théorème de Bayes pour trouver $P(C_k|x)$

Exemple : **Bayes naïf**

→ Modèles discriminants

Même besoin mais ici : On calcule directement $P(C_k|x)$ et on l'utilise dans la décision.

Exemple : **Analyse discriminante linéaire, régression logistique, voire SVM (maximisation marge)**

→ Modèles discriminants par fonction

On cherche une fonction qui fait correspondre la donnée à prédire à une classe sans calculer de probabilité.
L'algorithme peut être paresseux : seule la collection des données est réalisée à l'apprentissage, la sélection des données proches et le calcul de la fonction (polynôme) à l'exécution.

Exemple: **K plus proches voisins**

→ Arbres de décision

Algorithme glouton décidant, lors de la phase d'apprentissage, les choix matérialisés par les branches de l'arbre optimums. Ces choix se fondent sur la sélection de la variable. La variable sélectionnée doit être discriminante et donc constituer des groupes homogènes lors des choix. Des indices peuvent mener à une meilleure sélection de variable.

5.2.3. Régression

L'objectif est l'apprentissage d'une fonction continue : On veut prédire une valeur.

→ Régression multilinéaire

Les méthodes de régression linéaire, qui presupposent un lien linéaire entre variables prédicteurs et à prédire, peuvent :

- par transformation s'adapter à des relation non linéaire
- être optimisées par des méthodes de pénalité (ex: Ridge, Lasso) sur les coefficients du polynôme.

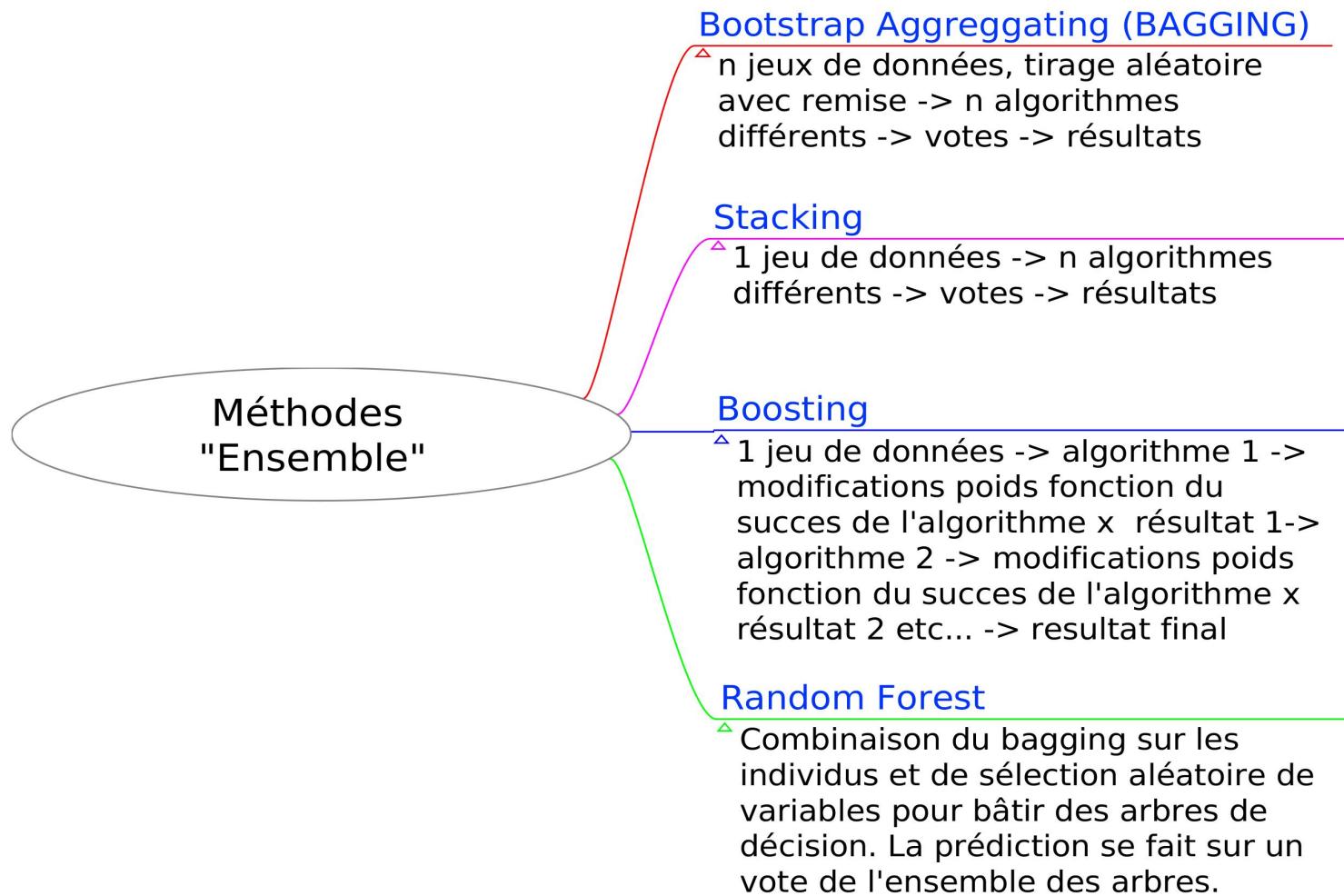
5.3. Autres méthodes

5.3.1. Méthodes "ensemble"

Ces méthodes combinent plusieurs exécutions d'algorithmes pour améliorer la performance de prédiction.

En effet ces algorithmes pris individuellement sont moins efficaces ("weak learner") qu'une fois combinés ("strong learners").

→ Récapitulatif des méthodes ensemble



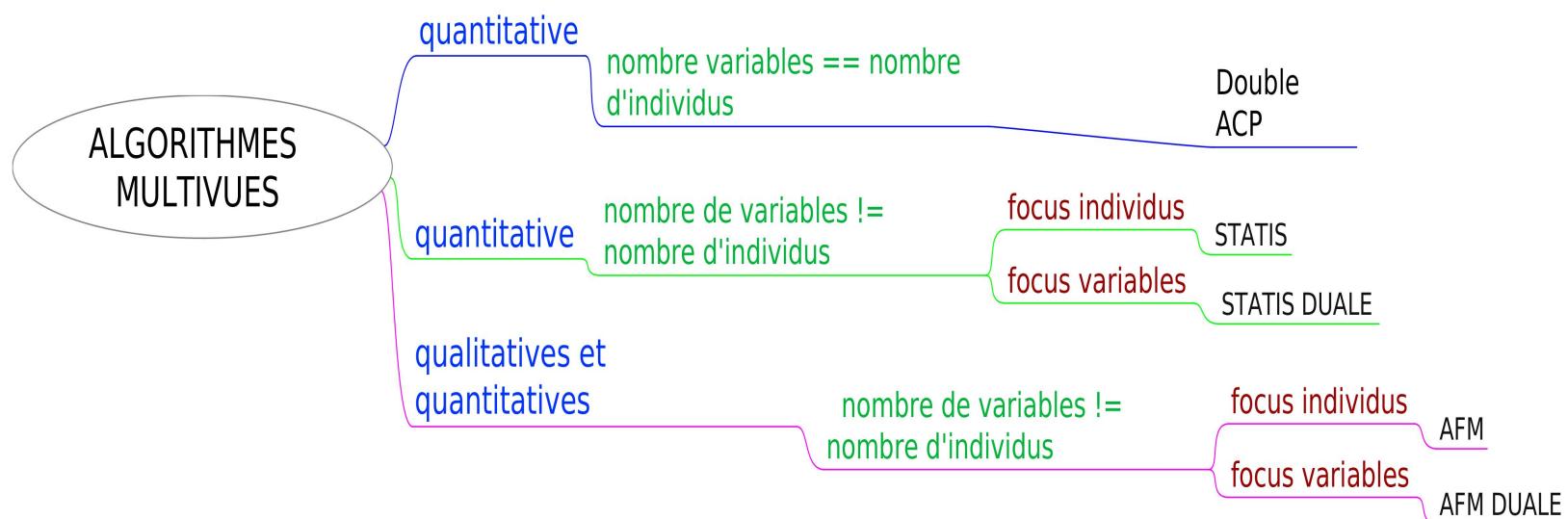
5.3.2. Méthodes duales

Ces méthodes permettent l'étude de l'évolution des matrices au cours du temps. Elles sont fondées sur l'application de méthodes d'analyse factorielle. Elles sont utilisées en non supervisé ou en supervisé selon la nature du problème.

→ Synthèse des étapes de la méthode:

- étude des tableaux
- remplacer les tableaux par leur centre de gravité
- application de l'analyse factorielle
- visualisation sur un graphique de la projection des tableaux

→ Schéma des déclinaisons de la méthode:



5.3.3. Réseaux de neurone

Caractéristiques:

Utilisable dans un contexte supervisé ou non supervisé, et maintenant quelque soit la nature (image mais aussi texte) de la donnée.

Principes:

Le neurone applique une fonction sur la combinaison d'un ensemble pondéré d'entrées.

→ Perceptron

Caractéristiques:

- n entrées, 1 neurone, 1 sortie
- classificateur linéaire

→ Réseau de neurone à rétro propagation de l'erreur

Caractéristiques:

- n entrées, n neurone d'entrée, p neurones intermédiaires (couche cachée), t neurones de sorties, 1 sortie
- Spécificité de l'apprentissage : la rétro propagation de l'erreur :
n entrées ⇒ résultat ⇒ mesure erreur ⇒ rétro propagation ⇒ modification des poids ⇒ nouveau résultat, etc...

5.3.4. Deep Learning et réseaux de neurones

Cette méthode traite une problématique nouvelle :

On traite plusieurs tableaux en parallèle, dont les données brutes sans filtrage préalable, grâce aux capacités des réseaux de neurones (multiples entrées, modèle non linéaire universel).

Dans le cas d'usage du traitement d'image, il existe une nouvelle contrainte : l'interversion possible des individus; ce qui jusqu'à présent n'est pas possible dans le cas des pixels des images.

Le côté "Deep" de cette méthode vient de la profondeur de la couche de neurones intermédiaires : on peut, en théorie, remplacer la couche cachée des réseaux de neurones par autant de couches de neurones intermédiaires que nécessaire.

ANNEXES

Bibliographiques / web

- carte heuristique sur Wikipédia : https://fr.wikipedia.org/wiki/Carte_heuristique

Tableau synoptique des caractéristiques des algorithmes

- Table 1. tableau synthétique des caractéristiques des Algorithmes NON supervisés

Nom(s) algorithme(s)	Nature des variables	Objectif	+	-	Commentaire
Analyse en composantes principales (ACP)	quantitatives	réduire les dimensions extraction de variable synthétique	flexible	interprétation nouvelle variable choix du nombres de valeurs propres perte d'information	<u>Complexité</u> : $O(m^3)$, m nombre de variables restreinte aux k plus grandes valeurs propres : $O(Nmk)$, N nombre de données
Analyse factorielle des correspondances (AFC)	qualitatives binaires	réduire les dimensions exploratoire décisionnel	-id-	-id-	-id-
Analyse des correspondances multiples (ACM)	Qualitatives	réduire les dimensions exploratoire	-id-	-id- + pas de cercle de visualisation	-id-
k -moyens, "k-means", centroides,	Toutes (transformation)	Classification - partition	Simple Efficace	-difficulté à définir k -pas déterministe	C : $O(tkN)$ <u>améliorations</u> :

Nom(s) algorithme(s)	Nature des variables	Objectif	+	-	Commentaire
méthode des centres mobiles	des qualitatives requise)			-sensible aux aberrations -problème avec des formes de données irrégulières	-supprimer aberrations -échantillonner les données -populaire
Classification hiérarchique ascendante (CAH)	-id-	Classification - hiérarchique visualisation	représentation graphique	-pas grands data sets -choix de la distance	C : O(N ²) -produit un dendrogramme -améliorables par BIRCH ou ré-échantillonnage -populaire
Classification divisive	-id-	Classification - hiérarchique	représentation graphique	-id-	inusitée
Cartes de kohonen, carte topologique auto organisatrice, SOM (self organized Map)	-id-	Classification - vectorielle visualisation	<u>polyvalente :</u> -réduction -dimensionnalité -visualisation -clustering	- pas vraiment de classes - coûteux	-moins connue -but descriptif
Règles d'association : Apriori Sampling DIC Partition Éclat	Toutes	classification - règles, probabilistes	simple nombre minimal de candidats optimisable		<u>Suivant les variantes :</u> -représentation horizontale ou verticale -parcours largeur / profondeur -nombre de lecture de l'ensemble des données +/- important -coût en ressources (CPU, RAM) différent

Nom(s) algorithme(s)	Nature des variables	Objectif	+	-	Commentaire
FP-Growth					-nombre de candidats générés -simplicité de la structure de données

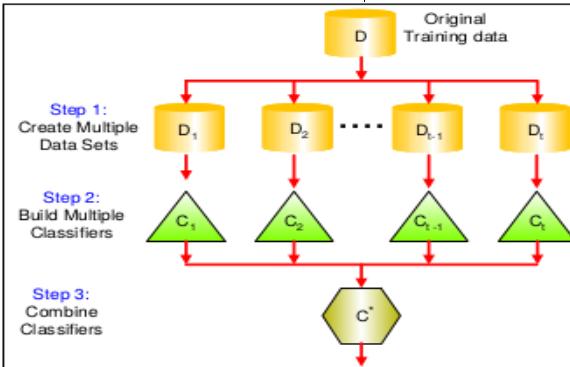
Table 2. tableau synthétique des caractéristiques des algorithmes supervisés

Nom algorithme	Nature des données	Objectifs	+	-	Commentaires
Régression multilinéaire - OLS (Ordinary Least Square)	Supervisé	Régression – Classement <i>binaire</i>	simple nombre minimal de candidats optimisable	<u>hypothèses :</u> linéaire distribution gaussienne var E/S rééchelonner les entrées <u>sensible :</u> bruit colinéarité	<u>Calcul précision :</u> RSE R^2 <u>Augmenter la variance par "pénalité":</u> Ridge / LASSO <u>Régulariser:</u> PLS / PCR / Ridge
Régression logistique	Supervisé	Classement - probabiliste	pas d'hypothèse de distribution extension multi-classe vue probabiliste des classes rapide pour l'apprentissage très rapide pour les données inconnues bonne précision pour les	Hypothèse frontière de décision linéaire	calcule la probabilité conditionnelle $P(Y=\text{classe} X=\text{données})$

Nom algorithme	Nature des données	Objectifs	+	-	Commentaires
			data sets simples résistant à au surapprentissage coefficients du modèle = importance des caractéristiques		
Support Vector Machine (SVM)	Supervisé	Classement - <i>binaire</i> , probabiliste discriminatif	classificateur linéaire ou non-linéaire -efficace avec beaucoup de variables -marche si séparation claire -efficace sur dimensions > échantillons -occupation mémoire	<u>sensible au bruit</u> <u>que deux classes</u> <u>mauvais</u> avec beaucoup d'échantillons si bruit = subduction des classes pas d'estimation probabiliste dans coûteux 5-fold cross-validation	- maths du "kernel trick" complexes <u>-A définir</u> fonction "kernel" paramètres kernel (gamma) C contrôle des var correctrices d overfitting
Arbre de décisions	Supervisé	Classement - binaire (Régressions)	facile à construire extrêmement rapide pour les records inconnus facile à interpréter pour les petits arbres précision comparables aux autres pas de superposition sur les variables	- Sensibles aux faibles variations des données d'apprentissage -coûteux à l'apprentissage -complexité des arbres à la généralisation	Exemples :ID3, C4.5, CART gloutons "pureté du nœud" : Info Gain, GINI, Misclassification error overfitting : Prepruning /

Nom algorithme	Nature des données	Objectifs	+	-	Commentaires
			pas de paramètres sélection de variable dans l'algorithme	-expressivité limitée	postpruning
Naïf Bayésien	Supervisé	Classement - probabiliste (génératif)	facile à construire pour les très grands datasets très performant : meilleur que regression logistique meilleur avec les variables qualitatives robuste au bruit ignore les valeurs manquantes multi-classe natif	-hypothèse indépendance - raté d'une catégorie dans les données d'apprentissage (0-frequency \Rightarrow Laplace estimation) -mauvais estimateur	Extension Bayesian Belief Network améliorable : * transformer les variables en distributions normales * supprimer les variables corrélées * bien sélectionner les variables et les paramètres
k-plus proches voisins (KNN)	Supervisé	Classement -	faible temps d'apprentissage	temps de test classer records inconnus coûteux mauvais en grande dimension pas déterministe	algorithme feignant besoin de techniques de réduction dimension

Table 3. Tableau synthétique "autres méthodes"

Nom algorithme	Nature des données	Objectifs	+	-	Commentaires
Méthodes d'ensemble	Supervisé	bagging	<ul style="list-style-type: none"> réduire l'erreur réduire le bruit modèle stable 		<ul style="list-style-type: none"> Split n training set build n classificateurs algorithmes différents hyper paramètres différents combinaison par vote ou poids <p>plus efficace que ses constituants</p> <p>n jeu de données → n méthodes différentes → vote ⇒ résultat</p>
Méthodes d'ensemble	Supervisé	boosting	<ul style="list-style-type: none"> réduire l'erreur réduire le bruit modèle stable 	trouver le bon poids pour les modèles	<ul style="list-style-type: none"> -1 jeu de données → méthode 1 → poids x méthode 1 (résultat1) → etc... ⇒ vote pondère le résultat final -poids dépend succès/erreur modèle -poids des exemples augmente si modèle fait des erreurs
Méthodes d'ensemble	Supervisé	stacking	<ul style="list-style-type: none"> réduire l'erreur réduire le bruit 	<ul style="list-style-type: none"> black box k-fold cross validation 	<ul style="list-style-type: none"> - 1 jeu de données → n méthodes différentes → - résultat « méta learner » remplace le vote

Nom algorithme	Nature des données	Objectifs	+	-	Commentaires
			modèle stable	obligatoire pour données test	
Random Forest, forêt aléatoires	Supervisé	Ensemble, classement	Populaire , précis, réutilisable bon résultat sur grand nombre données / individus donne l'importance des variables donne l'erreur de génération au cours du training process gère la donnée manquante et la donnée "déséquilibrée" multi classificateur natif	pas de visualisation comme les arbres sur apprentissage en cas de bruit biais des qualitatives avec beaucoup de rangs	sélection N nombre « training set » sélection M nombre variables modèle - apprentissage : sélection aléatoire de variable training set : bootstrap sample - résultat = vote tous les arbres
Réseau de neurones - perceptron	+ Supervisé que non supervisé	classificateur linéaire			vecteur $S = f(\sum w_i E_i)$ avec choix f f échelon f linéaire / morceaux f sigmoïde
Réseau multicouche à rétro-propagation de l'erreur	+ Supervisé que non supervisé	classificateur linéaire			erreur sortie pondérée $f'(Z_{h_k})$ correction poids $Z_{t+1} = Z_t + \Delta Z$ avec pas d'apprentissage erreur couche cachée

Nom algorithme	Nature des données	Objectifs	+	-	Commentaires
					<p>pondérée $f'(Wx_k)$</p> <p>correction poids $W_{t+1} = W_t + \Delta W$ avec pas d'apprentissage</p>