

**Projet de fin d’études**

En vue de l’obtention du diplôme master spécialisé : Big Data & Cloud Computing

|  |
| --- |
| Diagnostic et la prise de décision des maladies cas : Diabète |

**Encadrée par : Présenté devant les membres de jury :**

Pr. Said Tkatek (Faculté des sciences Kenitra) -Pr. Jaafar Abouchabka  
 -Pr. Najat Rafalia  
 -Pr.

**Réalisé par :**

KARJOUT abdeslam

##### Année Universitaire 2020-2022

[Tableau 1 Différence entre apprentissage supervisé et non supervisé 17](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725206)

[Tableau 2: Random forest 24](#_Toc113725207)

# Table des matières

Contents

[Table des matières 3](#_Toc113725154)

[Table des figures 5](#_Toc113725155)

[Introduction Générale 7](#_Toc113725156)

[Structure du rapport 8](#_Toc113725157)

[Chapitre 1 : ’’Machine Learning et le domaine médicale’’ 8](#_Toc113725158)

[Chapitre 2 : ‘’LE DIABETE SUCRE’’ 8](#_Toc113725159)

[Chapitre3 : “Conception et Réalisation” 8](#_Toc113725160)

[Chapitre 4: “Tests et Évaluation des Résultats” 8](#_Toc113725161)

[Chapitre 1 : ’’Machine Learning et le domaine médicale’’ 9](#_Toc113725162)

[3. Introduction 10](#_Toc113725163)

[4. Machine Learning: 10](#_Toc113725164)

[4.1. Définition et Histoire: 10](#_Toc113725165)

[4.2. Méthodes d’apprentissage : 12](#_Toc113725166)

[4.2.1. Apprentissage supervisé : 13](#_Toc113725167)

[4.2.1.1. Classification : 13](#_Toc113725168)

[4.2.1.2. Régression : 15](#_Toc113725169)

[4.2.2. Apprentissage non supervisé : 15](#_Toc113725170)

[4.2.2.1. Clustering 16](#_Toc113725171)

[4.2.3. Différence entre apprentissage supervisé et non supervisé : 17](#_Toc113725172)

[4.2.4. Apprentissage semi-supervisé 18](#_Toc113725173)

[4.2.5. Apprentissage par Renforcement : 19](#_Toc113725174)

[5. Deep Learning : 20](#_Toc113725175)

[5.1. Machine Learning vs Deep Learning: 20](#_Toc113725176)

[5.2. Pourquoi Deep Learning : 22](#_Toc113725177)

[6. Random Forest : 23](#_Toc113725178)

[6.1. Définition : 23](#_Toc113725179)

[6.2. Principe de fonctionnement du random forest : 23](#_Toc113725180)

[6.3. Genèsedel’algorithme 24](#_Toc113725181)

[6.4. Tree bagging: 24](#_Toc113725182)

[6.5. Feature sampling 25](#_Toc113725183)

[6.7. Random Forest : intuition 27](#_Toc113725184)

[6.8. Avantages et inconvénients des random-forests 28](#_Toc113725185)

[7. K-nearset Neighbors (KNN) : 29](#_Toc113725186)

[7.1. Définition 29](#_Toc113725187)

[7.2. Applications de k-NN dans l'apprentissage automatique : 31](#_Toc113725188)

[7.3. Avantages et inconvénients de l'algorithme KNN : 32](#_Toc113725189)

# Table des figures

[Figure 1 : LE ML pour tous : la régresion.les données sont modélisées par y = a\*x+b. 11](#_Toc113725190)

[Figure 2: type d'apprentissage machine learning 13](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725191)

[Figure 3Indices de performance pour les problématiques de classification 14](#_Toc113725192)

[Figure 4l :a proportion de prédictions correctes (Accuracy en anglais) 14](#_Toc113725193)

[Figure 5 : clustering 16](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725194)

[Figure 6: Différence entre apprentissage supervisé et non supervisé 17](#_Toc113725195)

[Figure 7: Semi-supervised learning 18](#_Toc113725196)

[Figure 8 :Apprentissage par Renforcement: 19](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725197)

[Figure 9: AI & ML & DL 21](#_Toc113725198)

[Figure 10: ML vs DL 21](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725199)

[Figure 11: Construction de l'arbre N°1 du random forest 25](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725200)

[Figure 12: Prediction du random forest 26](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725201)

[Figure 13 : Mis en œuvre d’une forêt d'arbres 26](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725202)

[Figure 14 : Random forest illustration 27](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725203)

[Figure 15: Euclidienne 29](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725204)

[Figure 16: Manhattan 29](file:///C:\Users\Bitquark\OneDrive\Bureau\all\PFE\rapport\rapport_pfe.docx#_Toc113725205)

Remerciement

Je tiens à remercier dans un premier temps, toute l’équipe pédagogique de *la Faculté des sciences Kenitra* et les intervenants professionnels responsables de la formation *Big Data & Cloud Computing* pour avoir assuré la partie théorique de celle-ci.  
  
Je remercie également Monsieur *Said Tkatek* pour l’aide et les conseils concernant les missions évoquées dans ce rapport, qu’il/elle m’a apporté lors des différents suivis.

# Introduction Générale

* **Contexte**Le diabète est considéré depuis plusieurs années comme un des fléaux  
  du troisième millénaire. En 2006 on comptait 230 millions de diabétiques dans  
  le monde et 350 millions sont attendus en 2035 soit 120 millions de plus en  
  une génération. La proportion du diabète de type 2 et du diabète de type 1 est  
  respectivement de 90 et 5% (les 5% restants sont les autres types de diabète).
* **Problématique**Dans la plupart des cas le diabète est reconnu au stade de complications aiguës qui  
  est responsable d'une morbidité et d'une mortalité importante en l’absence de  
  traitement approprié et précoce.  
  La prise en charge correcte des malades pose d'énormes problèmes s'expliquant par :  
  - Le retard de diagnostic ;  
  - Le manque de formation du personnel de santé et de spécialistes ;  
  - L'insuffisance du plateau technique adapté ;  
  - Le coût élevé individuel et collectif de la prise en charge.  
  - Le manque d’éducation du diabétique et de son environnement social.

On a estimé le nombre de personnes adultes atteintes de cécité dû à une rétinopathie tardivement diagnostiquée à plus de 93 millions d’où l’importance d’une méthode complète et automatisée, reconnue par la communauté scientifique, pour aider à détecter une cette pathologie de façon précoce.

* **Solution proposée**  
  Notre principal objectif dans ce projet est d'analyser les données que nous avons collectées pour prédire si une personne est diabétique ou non. En outre, nous utiliserons également plusieurs techniques de Machine Learning pour nous aider à atteindre notre objectif. La réalisation de cette analyse est très importante car 9,3 % représente une part significative de la population adulte mondiale. L'observation des tendances et des modèles de données liés au diabète pourrait nous permettre de prédire plus facilement si les gens sont diabétiques ou non.

# Structure du rapport

Afin de bien mener ce projet, nous avons structuré le rapport de la manière suivant :

# Chapitre 1 : ’’Machine Learning et le domaine médicale’’

Définit essentiellement le Machine Learning ainsi que le AI dans le domaine médical.

# Chapitre 2 : ‘’LE DIABETE SUCRE’’

Décrit la diabétique, ainsi que l’organe touché par cette dernière, incluant les études effectuées à but de diagnostiquer le diabète.

# Chapitre3 : “Conception et Réalisation”

Présente la solution apportée à la problématique posée.

# Chapitre 4: “Tests et Évaluation des Résultats”

Décrit les résultats des tests d’évaluation effectués par rapports aux critères d’évaluation fixés.

# Chapitre 1 : ’’Machine Learning et le domaine médicale’’

# Introduction

Dans ce chapitre nous allons parler de l'émergence du Machine Learning et du Deep Learning, les raisons de leur apparition ainsi que leur impact, puis on verra un exemple de Random Forest et K-nearset (KNN) et LDA (linear Discriminant Analysis) tout en illustrant son fonctionnement interne dans le domaine médicale .

# Machine Learning:

# Définition et Histoire:

L'apprentissage automatique ou Machine Learning (ML) est une sous-section du domaine de l'intelligence artificielle ou IA en informatique, qui vise à apprendre aux machines à effectuer une tâche sans être explicitement programmé, et cela en utilisant un des algorithmes qu’on appellera modèles et de données.

Le concept d’intelligence artificielle est apparu dans les années 50 dans une assemblée rassemblant toute une flopée de savants célèbres en informatique et en mathématique dont **Alan Turing**. Ce savant de génie a aussi prédit le développement du Machine Learning tel qu’on le connaît.

Le ML a refait surface entre les années 70 à 80, l’idée derrière le concept était de créer des algorithmes ayant la capacité d’accumuler de l'expérience, et de la connaissance à partir de données sans être explicitement programmé pour effectuer cette tâche, et c’est vers la fin des années 80 qu’on a le retour des réseaux de neurones (créé plutôt 1943 par **Walter Pitts** et 9 Warren **McCulloch** inventeurs du premier Perceptron qu’on appellera par la suite **Neurone artificiel**. L’idée était de reproduire un neurone biologique d’un cerveau humain de façon artificielle en utilisant d’opérations mathématiques, malheureusement cette approche fut très limitée dans la résolution des problèmes. [1][2]

Pour qu’une machine apprenne et s’adapte automatiquement, la connaissance qu’elle peut extraire de ses expériences doit être stockée sous une certaine forme, de façon à pouvoir être utilisée dans un but précis. En général, le but est de prédire le futur (un comportement, un nombre, etc.), mais une machine peut également être construite pour générer artificiellement de nouvelles données (machines génératives) ressemblant statistiquement aux données originales, ou bien pour détecter des schémas de fonctionnement (causalités, structure d’un réseau, etc.).

Comme nous l’avons mentionné plus haut, le ML a beaucoup puisé dans les mathématiques et les statistiques. Classiquement, les statistiques s’attachent à permettre une compréhension globale des données : quelles sont les caractéristiques d’une variable (minimum, maximum, moyenne) ? De façon plus avancée, quelle est la distribution statistique ? Peut-on modéliser le processus de génération des données en essayant d’extraire le bruit contenu dans les données ? Le modèle le plus simple est la régression, très utilisée dans de nombreux domaines. Entraîner (faire apprendre) un modèle de régression c’est faire du ML. On peut donc faire remonter le machine learning au XVIIème siècle avec Legendre et Gauss et leur méthode des moindres carrés.

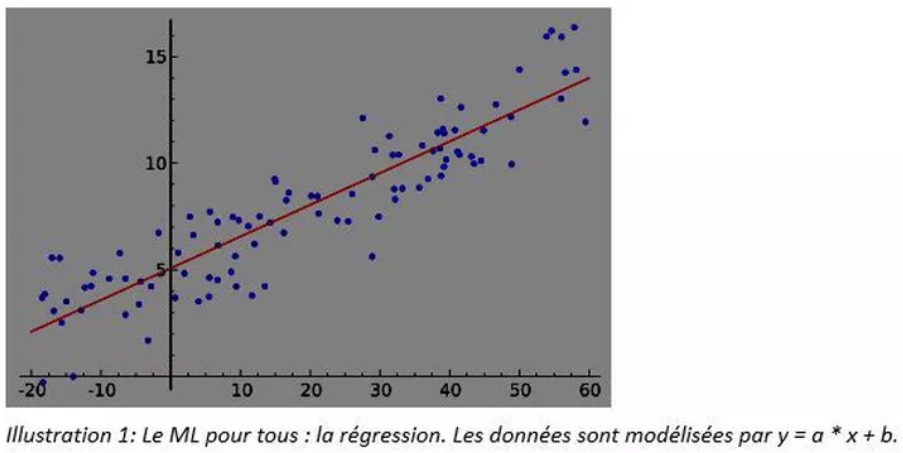


Figure 1 : LE ML pour tous : la régresion.les données sont modélisées par y = a\*x+b.

Vers la fin des années 80, on a vu le retour des réseaux de neurones, avec une réinvention de l’algorithme de **Rétropropagation (Backpropagation),** mais sans succès car le domaine sera finalement laissé à l'abandon faute de capacité de calcul et de stockage suffisantes. Il fallait attendre le milieu des années 2000 pour le grand retour des réseaux de neurones avec le **Deep Learning (3).**

Il faut aussi noter l’importance de l’arrivée du Big Data : la disponibilité d’ensembles de plus variés de données massives et l’augmentation des capacités de calcul et de stockage.

Le développement des technologies Big Data a permis de dépasser des limites importantes du ML : l’absence d’une quantité suffisante de données ne permet pas d’entraîner efficacement un algorithme de prédiction. Par conséquent, en l’absence de telles données, on ne dispose pas d’expérimentations probantes ; il est alors plus difficile de développer les fondements théoriques d’un algorithme de ML.

Le Big Data pose également des challenges d’ordre théorique pour le ML, comme les problèmes de grandes dimensions : le croisement de nombreuses sources de données diluent fortement l’information. On augmente ainsi le nombre de variables à explorer, mais pas le nombre d’exemples disponibles.

# Méthodes d’apprentissage :

Il y a de nombreux sous domaines ou sous sections. On dénote 3 principaux, ses dernières dépendent du type de problèmes que l’on voudrait traiter, on retrouve :

* **Supervised Learning**
* **Unsupervised Learning**
* **Reinforcement Learning.**
* **Semi-supervised Learning**

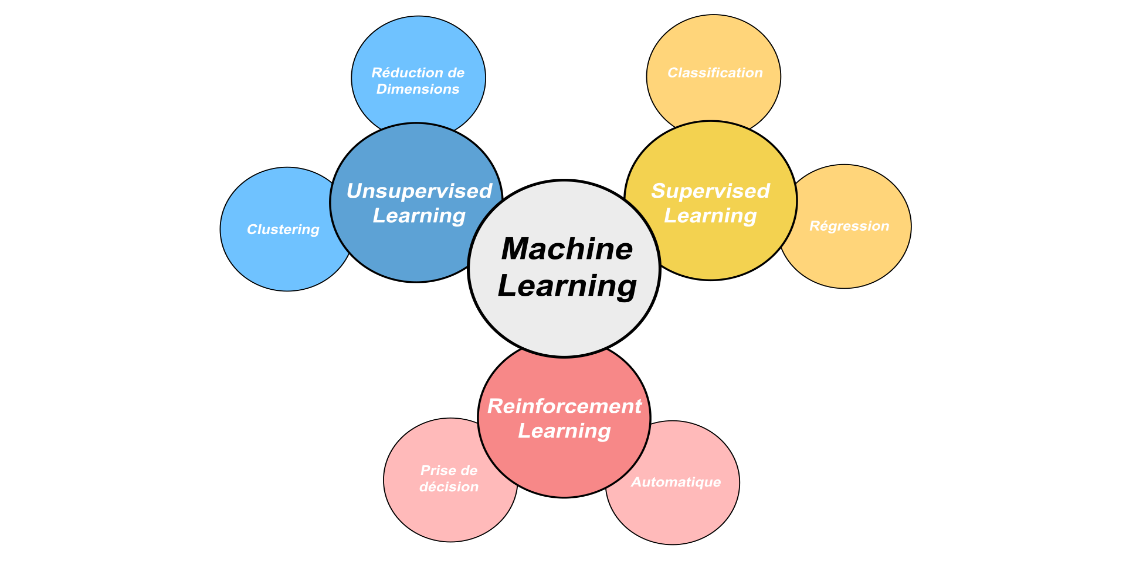
****

Figure 2: type d'apprentissage machine learning

# Apprentissage supervisé :

**L’Apprentissage supervisé** est un ensemble d'algorithmes qui permettent à l'ordinateur d'apprendre à prédire un résultat à partir d'un ensemble de prédicteurs. Le jeu de données doit inclure une variable dépendante aussi appelée variable Y. Il s’agit de la variable que l’ordinateur devra apprendre à prédire. Les autres variables sont les **Prédicteurs**, également appelés **variables X** et utilisées par l'ordinateur pour construire des modèles permettant de prédire Y.

Quand Y est une variable **qualitative** (= catégorielle), nous travaillons sur une problématique de **classification**. Quand Y est quantitative, il s’agit d’une problématique de **régression**.

# Classification :

C’est une tâche d’apprentissage supervisé dont la sortie a des étiquettes définies (valeur discrète) ce qu’on appelle classes ou catégories.

On peut avoir des classifications **Binaire** ou alors **Multiclasse**.

La **classification binaire** prédit pour un modèle (0 ou 1) vrai ou faux, dans le cas de **multiclasses** le modèle va prédire plus d’une classe.

Le but ici reste le même quel que soit le nombre de classes, **prédire** si **les valeurs discrètes** appartiennent à une classe particulière.

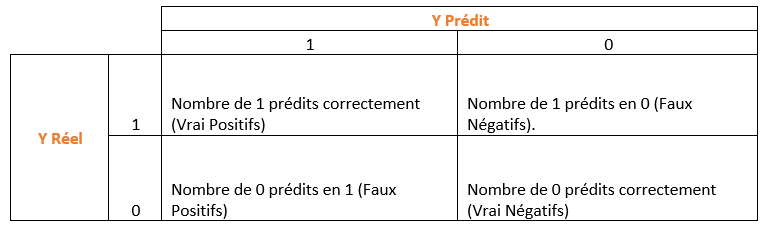


Figure 3Indices de performance pour les problématiques de classification

L’indice le plus intuitif que nous pouvons calculer à partir de cette matrice est la proportion de prédictions correctes (Accuracy en anglais) :

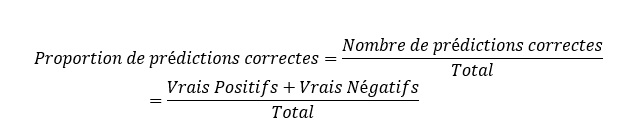


Figure 4l :a proportion de prédictions correctes (Accuracy en anglais)

Cet indice n'est pas fiable en cas de déséquilibre important entre les catégories de Y dans les données. Par exemple, lorsque l'algorithme apprend à détecter une fraude, les données d'entrée contiennent souvent beaucoup plus de cas historiques non frauduleux que de cas frauduleux. D'autres indices peuvent être utilisés en cas de déséquilibre de classe. Par exemple, l'indice d’Aire Sous la Courbe (AUC) calculé à partir d’une [courbe ROC](https://help.xlstat.com/fr/6627-roc-curve-analysis-excel-tutorial).

**Exemple :**

**E-Commerce**

L'ordinateur apprend à recommander des articles aux clients sur la base de données antérieures, notamment l'historique des achats et les profils d'autres clients.

**Détection de fraude (assurance)**

L'ordinateur apprend à détecter des fraudes à partir d’une base de données de comportement passées, liées à des cas confirmés frauduleux ou non-frauduleux.

#### Médecine

L'ordinateur apprend à prédire la prédisposition de patients à une maladie sur la base de données génétiques associées à des cas de patients confirmés malades ou sains.

#### Web Marketing

L'ordinateur apprend à prédire le risque de désinscription de clients (*churn*) en fonction des données de comportement passées.

#### Industrie

L'ordinateur apprend à prédire une défaillance future sur une chaîne de production en fonction des données de défaillance passées associées à différents signaux. Cela permet aux ingénieurs et techniciens d'intervenir suffisamment tôt sur la chaîne pour éviter la panne. C'est ce qu'on appelle la maintenance prédictive, composante majeure de l'industrie 4.0.

# Régression :

Il s’agit d’une tâche d’apprentissage supervisé dont la sortie a une **valeur continue.**

Le but ici est de prédire une valeur aussi proche de la valeur de sortie réelle que notre modèle le permet, puis l'évaluation est effectuée en calculant la valeur d'erreur. Plus l'erreur est petite, plus la précision de notre modèle de régression est grande.

#### Exemples de problématiques de régression

#### Finance

L'ordinateur apprend à prédire le prix de propriétés immobilières en fonction des caractéristiques et du prix de vente d’autres propriétés vendues récemment dans la région.

# Apprentissage non supervisé :

L'apprentissage non supervisé ou unsupervised est un type d'algorithme d'apprentissage  
automatique utilisé pour tirer des inférences à partir d'ensembles de données constitués de  
données d'entrée sans réponses étiquetées.

Ce type d’apprentissage est principalement utilisé quand on souhaite étudier un ensemble de données non labellisés

# Clustering

Le clustering permet de séparer les données entrées en un ensemble ou groupe de données  
qui ont des traits similaires et de les affecter à un cluster. Contrairement à la classification  
dans l’apprentissage supervisé ces différents clusters ou groupe ne sont pas connus à l’avance,  
c'est l’algorithme lui-même qui va séparer les données aux nombres de clusters qu’il faut.  
Comme l’on peut le voir pouvez le voir dans l'exemple *Figure* *5*, les points de jeu de  
données ont été divisés en groupes identifiables par les couleurs rouge, vert et bleu.

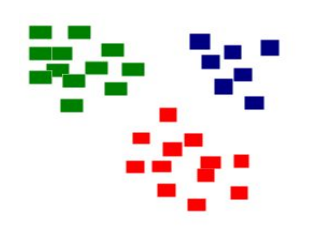


Figure 5 : clustering

* + 1. Différence entre apprentissage supervisé et non supervisé :

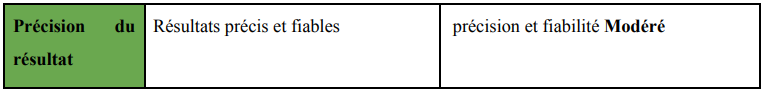
La majeure différence qu’on peut trouver entre ces 2 types d’apprentissages est la disposition des données d’entrée, mais il existe tout de même d’autres différences qu’on peut trouver dans le tableau ci-dessous.[4] Ce tableau résume la différence entre apprentissage supervisé et non supervisé

Tableau 1 Différence entre apprentissage supervisé et non supervisé

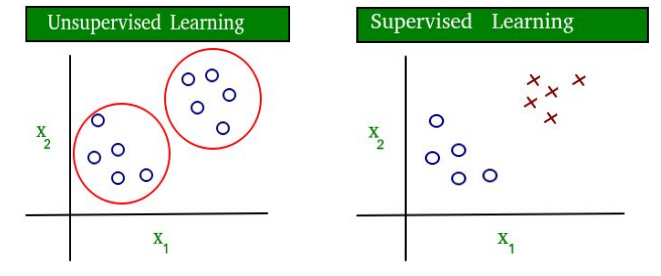
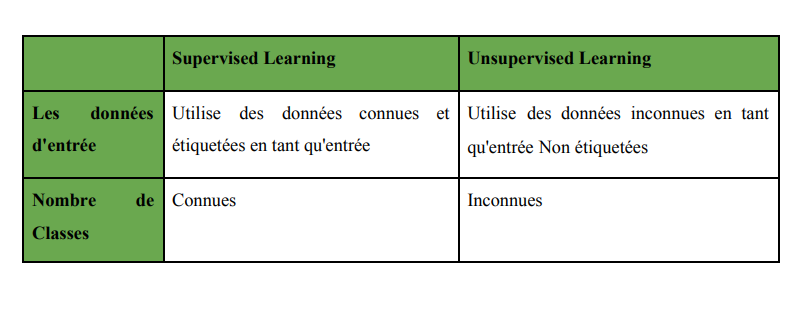


Figure 6: Différence entre apprentissage supervisé et non supervisé

* + 1. Apprentissage semi-supervisé :

Nous avons préalablement vue l’apprentissage supervisé et non supervisé, dont la majeure différence réside dans le fait que les données soient étiquetées ou non, et à cela s’ajoute les méthodes adéquates utilisées pour traiter ses données.

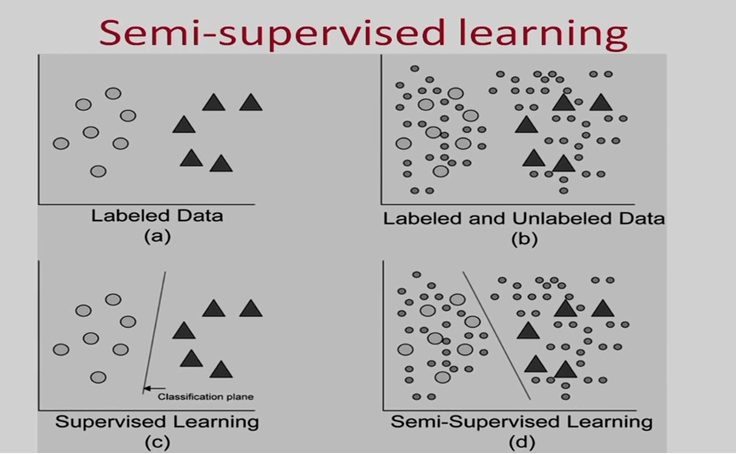


Figure 7: Semi-supervised learning

L’apprentissage semi-supervisé regroupe ses deux principes, il prend un ensemble réduit de données étiquetées avec un autre ensemble de données non étiquetées du même types. L’avantage de ce type d’apprentissage réside principalement dans le processus d’étiquetage des données prend beaucoup de temps et souvent coûteux. Donc paradoxalement le non étiquetage devient bénéfique pour le processus d’apprentissage, et la construction du modèle et moins coûteuse. [5]

# Apprentissage par Renforcement :

Entraînement par renforcement est une technique dans laquelle on immerge un agent dans un environnement où celui-ci interagit avec son environnement dans le but d’apprendre.

Un agent dans un environnement est muni de capteurs pour capter les informations de son environnement, ainsi que d’actionneurs qui lui permettront d’agir dans son environnement.

Les agents observent l'entrée, puis il effectue une action en prenant des décisions. Une fois l’action réalisées, l'agent reçoit des récompenses en conséquence, ce qui renforce le modèle en stockant ses informations dans une base de données. [6][7].

La récompense peut être positive, négative ou nulle selon les actions effectuées.

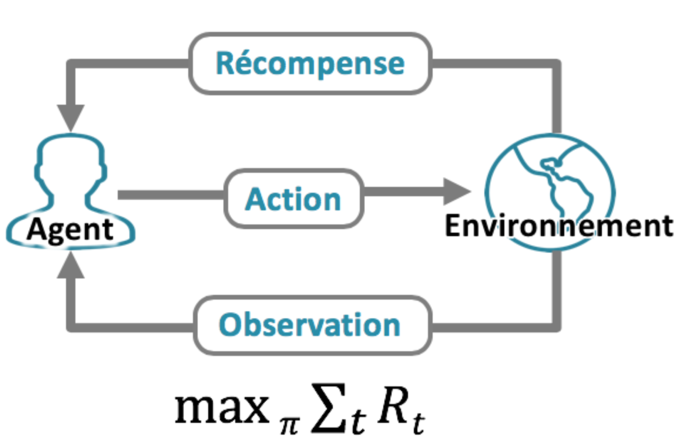


Figure 8 :Apprentissage par Renforcement:

**Example** : Alpha GO, robot domestique, ou IA dans les jeux vidéo.

# Deep Learning :

Deep Learning ou apprentissage en profondeur ou DL est une branche du Machine Learning entièrement basée sur **des réseaux de neurones artificiels**. [19]

Le concept d'apprentissage en profondeur existe depuis plusieurs années, mais il a été laissé à l'abandon faute de moyens nécessaires.

Dans le milieu des années 2000 le Machine Learning fait rage dans les compétitions de reconnaissance visuelle, en 2012 **Deep Mind** une startup dans le domaine de l’IA arrive dans la compétition avec un algorithme de Deep Learning qui bat largement tous les autres compétiteurs, l’année suivante tous les compétiteurs se sont tournés vers le Deep Learning au vu des résultats obtenus.

L'avancée du DL est dû à l’augmentation en exponentiel qu'ont connu les machines en capacités de **calculs**, et de **stockages**, ainsi que la disponibilité de données de masses (big data), ses 3 ingrédients étaient nécessaires pour exploiter le potentiel du DL qui fût chose impossible dans les années 90.

# Machine Learning vs Deep Learning:

La majeure différence qu’on note entre ses 2 concepts provient de la manière dont les données sont présentées au système (modèle).

* Les algorithmes de ML nécessitent presque toujours des données structurées, alors que les réseaux d’apprentissage approfondis reposent sur des couches de réseaux de neurones artificiels (RNA).
* On voit aussi une différence au sein de l’architecture des modèles qui les composent, on note que les modèles type DL sont plus profond que les modèles type ML.
* Deep Learning n’utilise que les réseaux de neurones, alors que pour le ML les réseaux de neurones sont qu’une approche de conception des modèles parmi tant d’autres.

En considérant le fait que le DL est la prochaine étape de l’évolution du ML inculquant aux machines la manière de prendre leurs décisions de façon précise sans l’intervention de l’expert humain

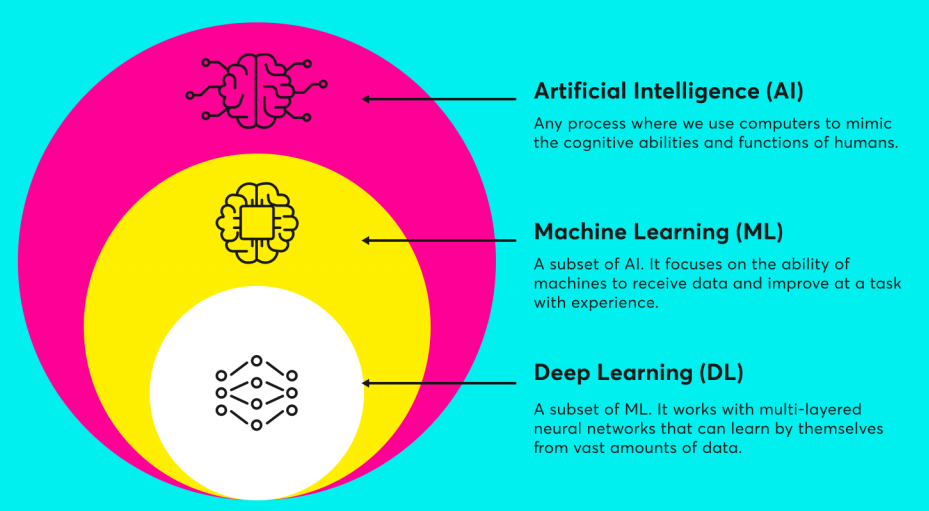


Figure 9: AI & ML & DL

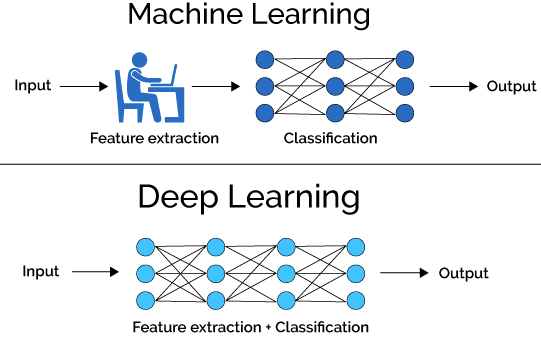


Figure 10: ML vs DL

# Pourquoi Deep Learning :

Il s’agit d’une combinaison de facteurs dont :

* **L’omniprésence des données :** Nous somme dans l’ère de l’informatisation (Internet Of Things), et le propre du Deep-Learning est de tirer parti d’une grande quantité de données pour en estimer une représentation abstraite et en tirer parti.
* **La puissance de calcul** : La théorie des réseaux de neurones existe depuis quelques décennies, mais c’est grâce à la puissance de calcul accessible aujourd’hui qui se démocratise, notamment depuis que les GPUs sont devenus la plateforme de choix pour le Deep-Learning.
* Des **besoins croissants dans le domaine de l’IA** : vision par ordinateur, reconnaissance vocale, traitement du langage, …etc.
* Un **effet de mode**. On a tendance à vouloir appliquer le Deep-Learning partout alors que ça reste un moyen et non une fin. Certains problèmes sont tout à fait solubles par d’autres méthodes d’apprentissage statistique. Cela dit, si beaucoup de gens sont prêts à investir dans le Deep-Learning, il est normal qui deviennent si populaire.
* **Capacité de Stockage :** qui sont devenu beaucoup plus accessible à prix raisonnable.
* Apparition de plateformes et communautés forte encouragent l’évolution de ce domaine, ainsi que sa démocratisation.

# Random Forest :

# Définition :

Le random forest est un algorithme incontournable en machine learning. Random forest signifie « forêt aléatoire ». Proposé par Leo Breiman en 2001, c'est un algorithme qui se base sur l’assemblage d’arbres de décision. Il est assez intuitif à comprendre, rapide à entraîner et il produit des résultats généralisables. Seul bémol, le random forest est une boîte noire qui donne des résultats peu lisibles, c’est-à-dire peu explicatifs.

Il est néanmoins possible de limiter cet inconvénient par d’autres techniques de machine Learning ce sera l’objet de notre article N°7 autour de LIME, un algorithme clé pour rendre un modèle explicable.

# Principe de fonctionnement du random forest :

Un random forest est constitué d'un ensemble d'arbres de décision indépendants.

Chaque arbre dispose d'une vision parcellaire du problème du fait d'un double tirage aléatoire :

* Un tirage aléatoire avec remplacement sur les observations (les lignes de votre base de données). Ce processus s'appelle le **tree bagging**.
* Un tirage aléatoire sur les variables (les colonnes de votre base de données). Ce processus s'appelle le **feature sampling**.

A la fin, tous ces arbres de décisions indépendants sont assemblés. La prédiction faite par le random forest pour des données inconnues est alors la moyenne (ou le vote, dans le cas d'un problème de classification) de tous les arbres.

L'idée de base de cet algorithme est assez intuitive. A titre d’exemple, si votre banque vous refuse votre demande de crédit, il y a fort à parier que vous irez consulter une ou plusieurs autres banques. Effectivement, un seul avis ne suffit pas en général pour prendre la meilleure décision.

Le random forest fonctionne sur ce même principe : plutôt que d'avoir un estimateur complexe capable de tout faire, le random forest utilise plusieurs estimateurs simples (de moins bonne qualité individuelle). Chaque estimateur a une vision parcellaire du problème. Ensuite, l'ensemble de ces estimateurs est réuni pour obtenir la vision globale du problème. C'est l'assemblage de tous ces estimateurs qui rend extrêmement performante la prédiction.

# Genèsedel’algorithme

Le défaut majeur de l'arbre de décision est que sa performance est fortement dépendante de l'échantillon de données de départ. Par exemple, l'ajout de quelques nouvelles données dans la base d'apprentissage peut modifier radicalement le modèle et les résultats.

Pour lutter contre ce défaut, on peut utiliser une multitude d’arbres : une forêt d'arbres. Vous comprenez maintenant le terme de forest contenu dans l’anglicisme random forest.

Et vous l’aurez compris, le terme “random” vient du processus de double tirage aléatoire que l’on applique à chaque arbre, à la fois sur les variables et sur les observations.

**\*\*Illustration pratique de l'algorithme**

Tableau 2: Random forest

|  |
| --- |
| Une formule à retenir**: Random Forest = tree bagging + feature sampling.** |

# Tree bagging:

Le Bagging signifie “bootstrap aggregation”. C'est un processus de tirage aléatoire sur les observations (lignes de données) déterminé par 3 étapes clés :

* Construction de n arbres de décisions en tirant aléatoirement n échantillons d'observations,
* Entraînement de chaque arbre de décision,
* Pour faire une prévision sur de nouvelles données, il faut appliquer chacun de n arbres et prendre la majorité parmi les n prévisions.

# Feature sampling

C'est un processus de tirage aléatoire sur les variables (colonnes de données). Par défaut, on tire Racine n variables pour un problème à n variables au total.

Pour reprendre l'exemple précédent de l’acceptation de crédit, l'idée de base du feature sampling c'est de demander à chaque banque d’étudier votre demande de prêt à partir d'un accès limité aux informations du client. L'une des banques rendra sa décision en ayant, par exemple, uniquement accès aux informations relatifs à l’âge, à la CSP et au revenu annuel du client. Une autre banque quant à elle, aura uniquement pris connaissance des informations relatives à la situation maritale, au sexe et au nombre de crédits en cours du client, ...

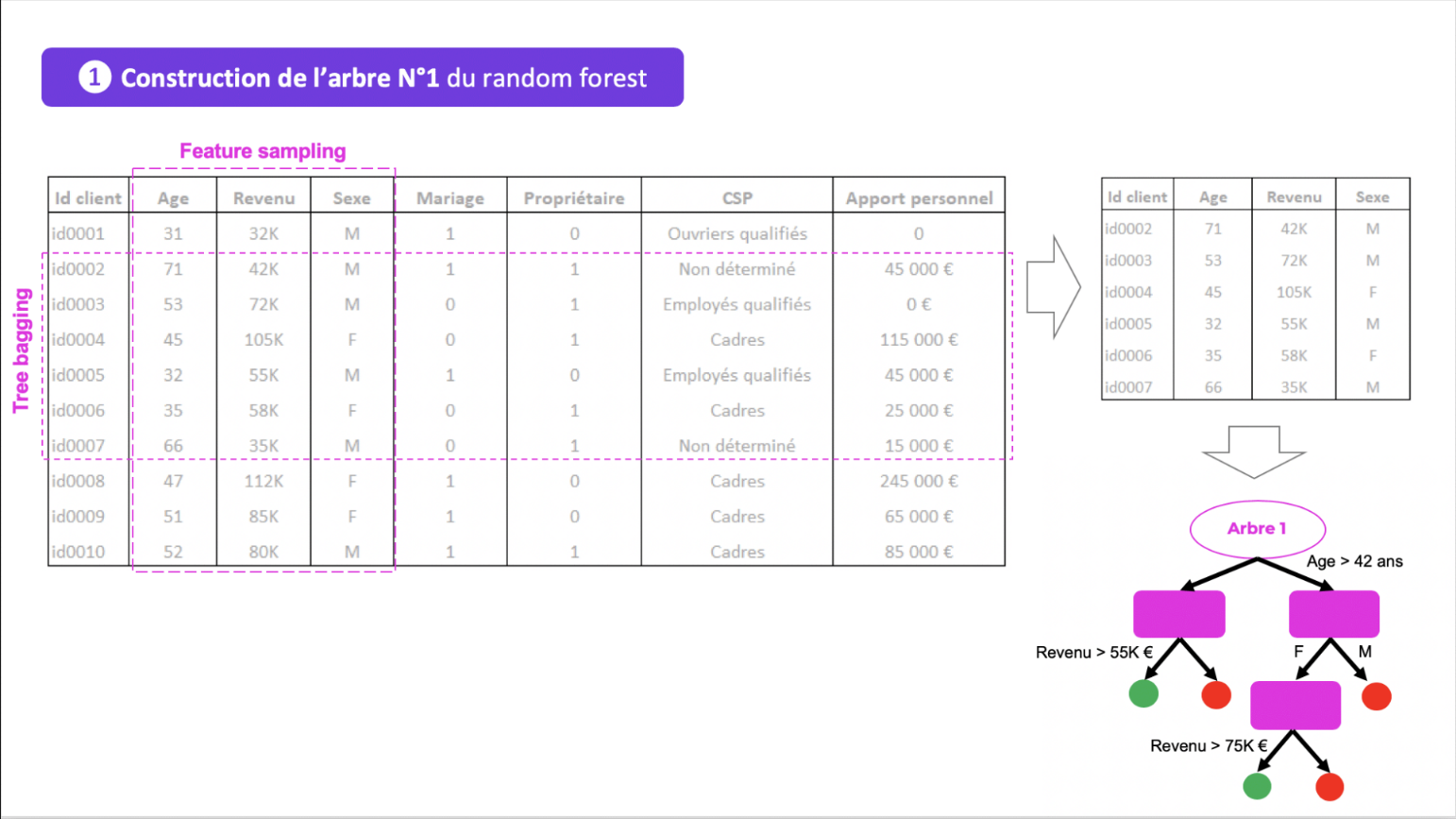
Ce processus permet de baisser la corrélation entre les arbres qui pourrait perturber la qualité des résultats. En statistique, on dit que le feature sampling permet de réduire la variance de l'ensemble créé.

Figure 11: Construction de l'arbre N°1 du random forest

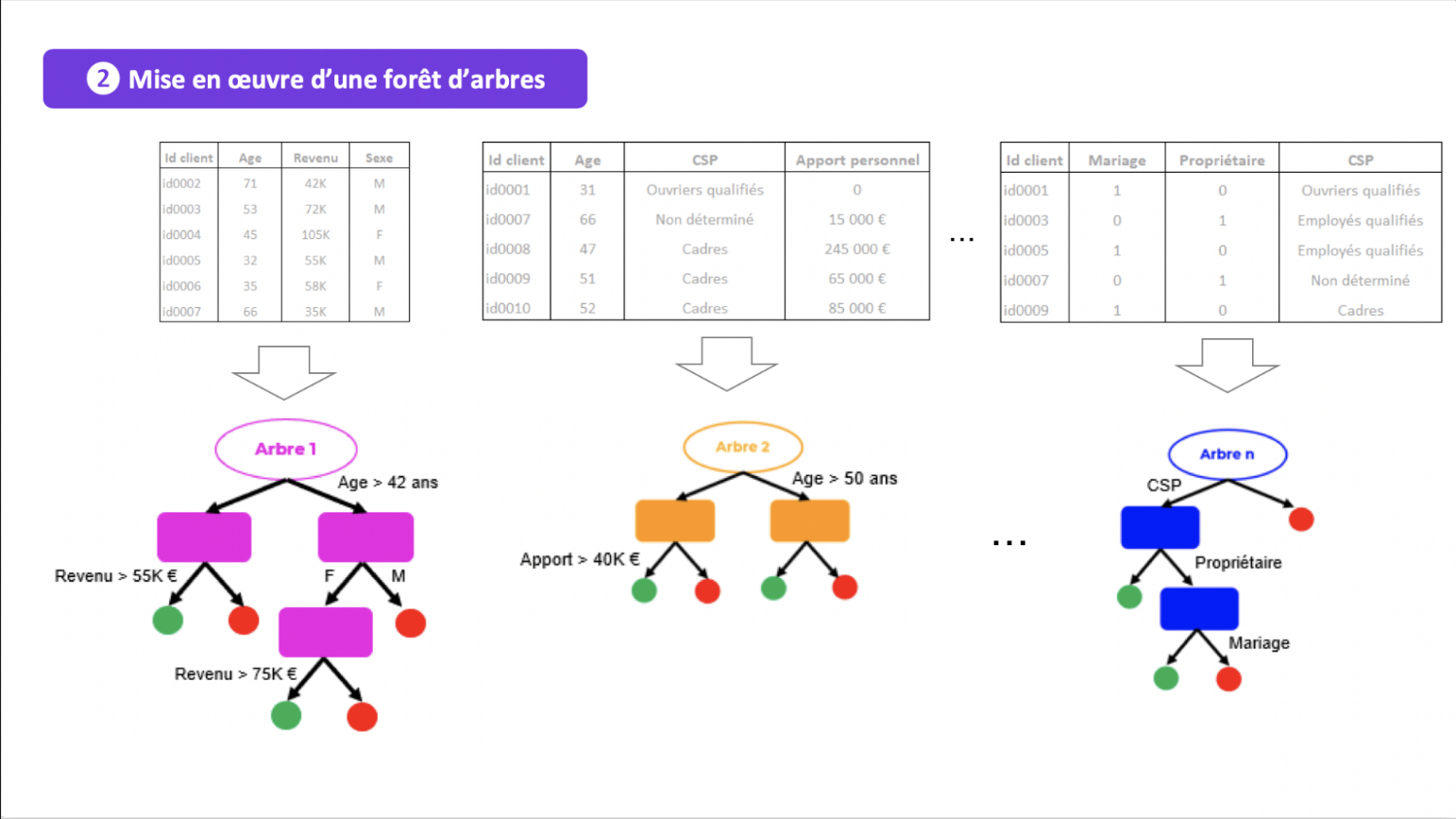
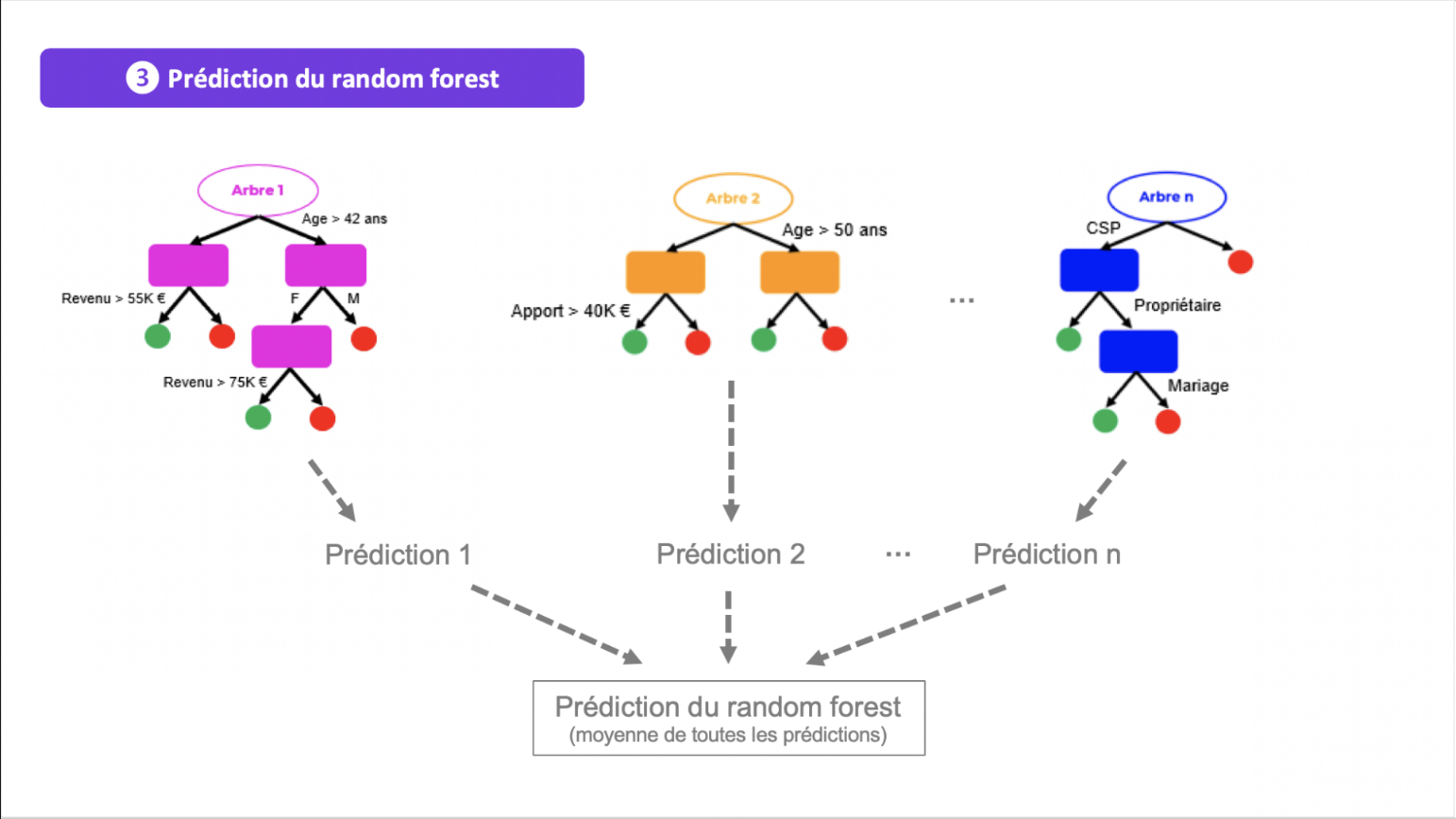


Figure 12: Prediction du random forest

Figure 13 : Mis en œuvre d’une forêt d'arbres

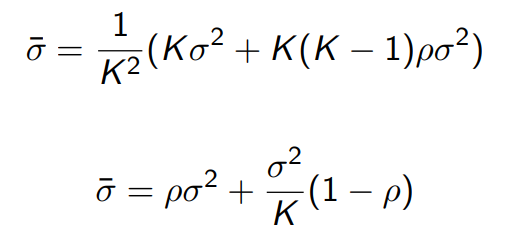
* 1. Critère de division/split

Vous le savez un arbre de décisions construit des sous-populations par séparations successives des feuilles d'un arbre.

Il existe différents critères de séparation pour construire un arbre :

* Le **critère de Gini** organise la séparation des feuilles d'un arbre en se focalisant sur la classe la plus représentée dans le jeu de données : il faut la séparer le plus rapidement possible.
* Le **critère d'entropie** est basé sur la mesure du désordre (comme en thermodynamique) qui règne dans la population étudiée. La construction de l'arbre vise à baisser l'entropie globale des feuilles de l’arbre à chaque étape.

# Random Forest : intuition

Si arbres sont identiquement distribués, de variance , avec un coefficient de corrélation deux à deux ρ la variance de leur moyenne est alors :

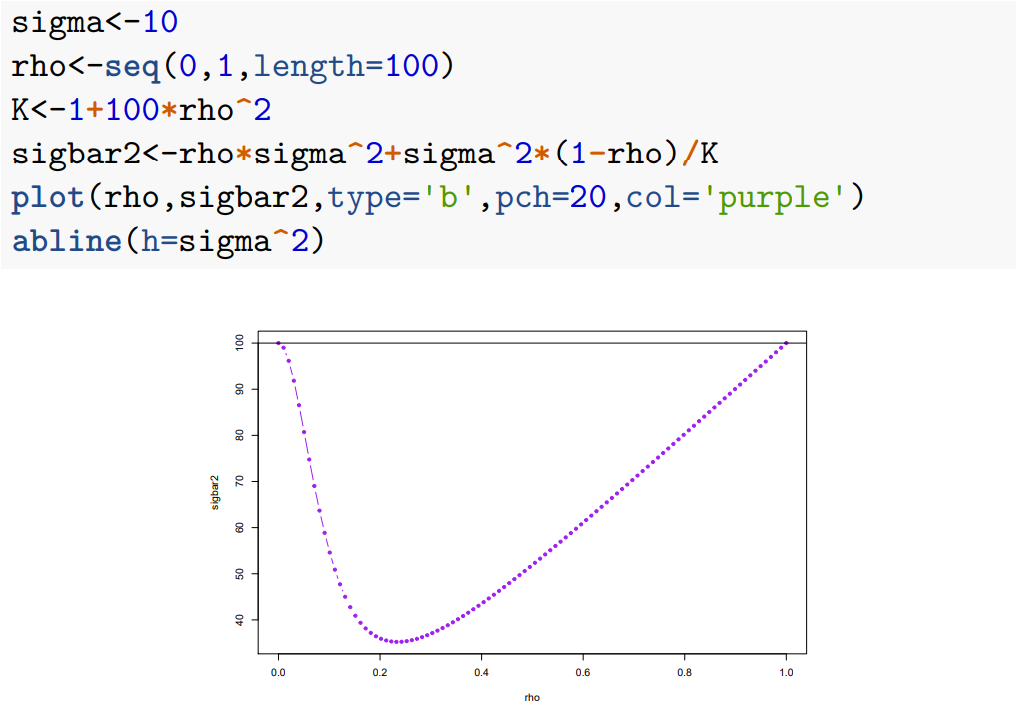


Figure 14 : Random forest illustration

# Avantages et inconvénients des random-forests

#### Avantages

* Pas de sur-apprentissage
* En général : meilleure performance que les arbres de décision, calcul de  
  l’erreur “Out-of-Bag” direct
* Paramètres faciles à calibrer
* Parallélisation possible
* Souvent utilisées comme benchmark dans les compétitions de machine learning

#### Inconvénients

* Boite noire : difficilement interprétable, difficilement améliorable
* Entrainement plus lent

Les random forest fonctionnent tout le temps bien mais excellent plus rarement même si les forêts sont robustes à un nombre de variable important, en grande dimension (n << p) il est nécessaire d’effectuer de la sélection de variable.

# K-nearset Neighbors (KNN) :

* 1. Définition :

L’algorithme des K plus proches voisins ou K-nearest Neighbors (kNN) est un algorithme de Machine Learning qui appartient à la classe des algorithmes d’apprentissage supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression. Dans cet article, nous allons revenir sur la définition de cet algorithme, son fonctionnement ainsi qu’une application directe en programmation.

L’intuition derrière **l’algorithme des K** plus proches voisins est l’une des plus simples de tous les algorithmes de Machine Learning supervisé :

* **Etape1 :** Sélectionnez le nombre K de voisins
* **Etape 2 :** Calculez la distance

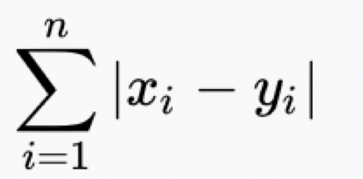


Figure 15: Euclidienne

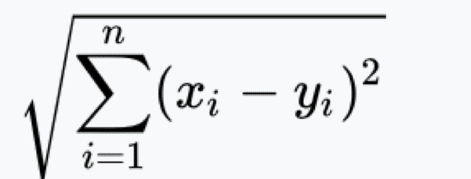
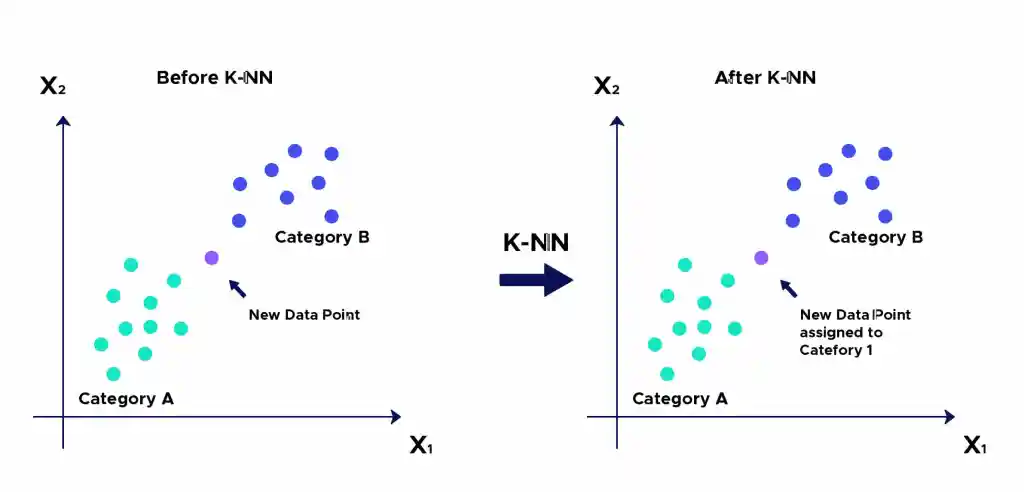
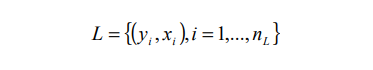


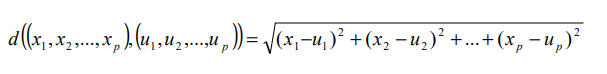
Figure 16: Manhattan

Du point non classifié aux autres points.

* **Étape 3** : Prenez les K voisins les plus proches selon la distance calculée.
* **Étape 4** : Parmi ces K voisins, comptez le nombre de points appartenant à chaque catégorie.
* **Étape 5** : Attribuez le nouveau point à la catégorie la plus présente parmis ces K voisins.
* **Étape 6** : Notre modèle est prêt :

La méthode du plus proche voisin est une méthode non paramétrique où une nouvelle observation est classée dans la classe d’appartenance de l’observation de l’échantillon d’apprentissage qui lui est la plus proche, au regard des Co variables utilisées. La détermination de leur similarité est basée sur des mesures de distance.

Formellement, soit L l’ensemble de données à disposition ou échantillon d’apprentissage :

où yi ∈{1,...,c} dénote la classe de l’individu i et le vecteur ) xi =(,.., ou x représente les variables prédicatrices de l’individu i. La détermination du plus proche voisin est basée sur un fonction distance arbitraire d (.,.). La distance euclidienne ou dissimilarité entre deux individus caractérisés par p Co variables est définie par :

# Applications de k-NN dans l'apprentissage automatique :

L'algorithme k-NN a été utilisé dans une variété d'applications, principalement dans le cadre de la classification. Certains de ces cas d'utilisation incluent :

* **Pré-traitement des données** : les jeux de données ont souvent des valeurs manquantes, mais l'algorithme KNN peut estimer ces valeurs dans un processus connu sous le nom d'imputation des données manquantes.
* **Moteurs de recommandation** : en utilisant les données de parcours de navigation des sites Web, l'algorithme KNN a été utilisé pour fournir des recommandations automatiques aux utilisateurs sur du contenu supplémentaire. Cette [recherche](https://www.researchgate.net/publication/267572060_Automated_Web_Usage_Data_Mining_and_Recommendation_System_using_K-Nearest_Neighbor_KNN_Classification_Method) (lien externe à ibm.com) montre qu'un utilisateur est affecté à un groupe particulier et, en fonction du comportement de l'utilisateur de ce groupe, il reçoit une recommandation. Cependant, étant donné les problèmes de mise à l'échelle avec KNN, cette approche peut ne pas être optimale pour les jeux de données plus volumineux.
* **Finance** : il a également été utilisé dans une variété de cas d'utilisation financière et économique. Par exemple, un [article](https://iopscience.iop.org/article/10.1088/1742-6596/1025/1/012114/pdf) (PDF, 391 Ko)  (lien externe à ibm.com) montre comment l'utilisation de KNN sur les données de crédit peut aider les banques à évaluer le risque d'un prêt à une entreprise ou à un individu. Il est utilisé pour déterminer la solvabilité d'un demandeur de prêt. Un autre [journal](https://www.ijera.com/papers/Vol3_issue5/DI35605610.pdf) (PDF, 447 Ko) (lien externe à ibm.com) met en évidence son utilisation dans les prévisions boursières, le change de devises, les contrats à terme et les analyses de blanchiment d'argent.
* **Soins de santé** : KNN a également eu des applications dans l'industrie des soins de santé, en faisant des prédictions sur le risque de crises cardiaques et de cancer de la prostate. L'algorithme fonctionne en calculant les expressions géniques les plus probables.
* **Reconnaissance de modèle** : KNN a également aidé à identifier des modèles, tels que la [classification de texte et de caractères](https://www.researchgate.net/profile/D-Adu-Gyamfi/publication/332880911_Improved_Handwritten_Digit_Recognition_using_Quantum_K-Nearest_Neighbor_Algorithm/links/5d77dca692851cacdb30c14d/Improved-Handwritten-Digit-Recognition-using-Quantum-K-Nearest-Neighbor-Algorithm.pdf) (lien externe à ibm.com). Cela a été particulièrement utile pour identifier les numéros manuscrits que vous pourriez trouver sur des formulaires ou des enveloppes postales.

# Avantages et inconvénients de l'algorithme KNN :

Comme tout algorithme d'apprentissage automatique, k-NN a ses forces et ses faiblesses. Selon le projet et l'application, il peut s'avérer être le bon choix ou le mauvais.

* **Avantages**
  + **Facile à implémenter :** compte tenu de la simplicité et de la précision de l'algorithme, c'est l'un des premiers discriminants qu'un nouveau spécialiste des données apprendra.
  + **Il s'adapte facilement** : à mesure que de nouveaux échantillons d'apprentissage sont ajoutés, l'algorithme s'ajuste pour tenir compte de toute nouvelle donnée, toutes les données d'apprentissage étant stockées en mémoire.
  + **Peu d'hyperparamètres :** KNN ne nécessite qu'une valeur k et une mesure de distance, ce qui est peu par rapport aux autres algorithmes d'apprentissage automatique.
* **Inconvénients**
  + **Mise à l'échelle difficile :** puisque KNN est un algorithme paresseux, il utilise plus de mémoire et de stockage de données par rapport aux autres discriminants. Cela peut être coûteux en termes de temps et d'argent. Davantage de mémoire et de stockage augmenteront les dépenses de l'entreprise et plus de données peuvent prendre plus de temps à calculer. Alors que différentes structures de données, telles que Ball-Tree, ont été créées pour remédier aux inefficacités de calcul, un discriminant différent peut s'avérer idéal en fonction de la problématique métier.
  + **Malédiction de la dimensionnalité** : l'algorithme KNN a tendance à être victime de la malédiction de la dimensionnalité, ce qui signifie qu'il fonctionne mal avec des entrées de données de grande dimension. Cette dernière est aussi parfois appelée le [phénomène de peaking](https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.418.6517&rep=rep1&type=pdf) (PDF, 340 Mo) (lien externe à ibm.com), où une fois que l'algorithme a atteint le nombre optimal de fonctions, des fonctionnalités supplémentaires augmentent le nombre d'erreurs de classification, en particulier lorsque la taille d'échantillon est plus petite.
  + **Enclin au surajustement** : en raison de la « malédiction de la dimensionnalité », KNN est également plus enclin au surajustement. Bien que les techniques de sélection des fonctions et de réduction de la dimensionnalité soient utilisées pour éviter que cela ne se produise, la valeur de k peut également avoir un impact sur le comportement du modèle. Des valeurs inférieures de k peuvent sur ajuster les données, tandis que des valeurs plus élevées de k ont tendance à « lisser » les valeurs de prédiction, car il calcule la moyenne des valeurs sur une plus grande zone ou sur le voisinage. Cependant, si la valeur de k est trop élevée, il peut sous-ajuster les données.

# LDA (linear Discriminant Analysis) :

# Définition :

En anticipant sur la présentation des méthodes de régularisation, nous avons choisi de présenter

ici l’Analyse Discriminante Linéaire (LDA en anglais) car elle est généralement considérée comme

une méthode d’analyse discriminante à part entière plutôt que comme une méthode de régularisation

de QDA. Pourtant, LDA est une régularisation de QDA car elle fait, par rapport à QDA, l’hypothèse

supplémentaire d’égalité des matrices de variances . L’analyse discriminante linéaire doit également son nom au fait qu’elle réalise des séparations linéaires entre les classes

# Hypothèses et Formules :

Nous disposons d’un échantillon de observations réparties dans groupes d’effectifs .

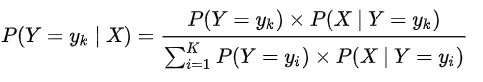
Notons la variable à prédire, elle prend ses valeurs dans l’ensemble .Des classes. Nous disposons de variables prédictives .

Nous notons  {\displaystyle \mu \_{k}} les centres de gravité des nuages de points conditionnels et  {\displaystyle W\_{k}} leurs [matrice de variance-covariance](https://fr.wikipedia.org/wiki/Matrice_de_variance-covariance).

# La règle bayésienne :

L’objectif est de produire une règle d’affectation {\displaystyle X(\omega )\mapsto Y(\omega )} qui permet de prédire, pour une observation {\displaystyle \omega } donnée, sa valeur associée de Y à partir des valeurs prises par X.

La règle bayésienne consiste à produire une estimation de la probabilité *a posteriori* d’affectation

) est la probabilité *a priori* d’appartenance à une classe.) représente la fonction de densité des conditionnellement à la classe .

La règle d’affectation pour un individu  à classer devient alors)

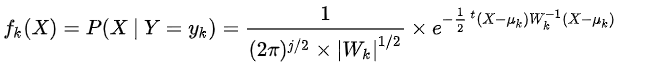
 Toute la problématique de l’analyse discriminante revient alors à proposer une estimation de la quantité ).{\displaystyle P(X~|~Y=y\_{k})}

# L'analyse discriminante paramétrique - L'hypothèse de multi normalité

On distingue principalement deux approches pour estimer correctement la distribution ).

* L’approche non-paramétrique n’effectue aucune hypothèse sur cette distribution mais propose une procédure d’estimation locale des probabilités, au voisinage de l’observation à classer. Les procédures les plus connues sont la méthode d'estimation par noyau et la méthode des plus proches voisins. La principale difficulté est de définir de manière adéquate le voisinage.
* La seconde approche effectue une hypothèse sur la distribution des nuages de points conditionnels, on parle dans ce cas **d’analyse discriminante paramétrique.** L’hypothèse la plus communément utilisée est sans aucun doute l’hypothèse de multi normalité.

Dans le cas de la loi normale multidimensionnelle, la distribution des nuages de points conditionnels s’écrit :



Où représente le déterminant de la matrice de variance covariance conditionnellement à .

L’objectif étant de déterminer le maximum de la probabilité a posteriori d’affectation, nous pouvons négliger tout ce qui ne dépend pas de . En appliquant le logarithme à la relation de Bayes, nous obtenons le score discriminant proportionnel à .



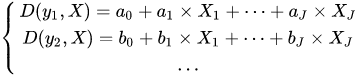
La règle d’affectation devient donc .

Si l’on développe complètement le score discriminant, nous constatons qu’il s’exprime en fonction du carré et du produit croisé entre les variables prédictives. On parle alors d’analyse discriminante quadratique. Très utilisée en recherche car elle se comporte très bien, en matière de performances, par rapport aux autres méthodes, elle est moins répandue auprès des praticiens. En effet, l’expression du score discriminant étant assez complexe, il est difficile de discerner clairement le sens de la causalité entre les variables prédictives et la classe d’appartenance. Il est notamment malaisé de distinguer les variables réellement déterminantes dans le classement, l’interprétation des résultats est assez périlleuse.

# Fonction de classement linéaire

En développant l’expression du score discriminant après introduction de l’hypothèse d’homoscédasticité, on constate qu’elle s’exprime linéairement par rapport aux variables prédictives.

Nous disposons donc d’autant de fonctions de classement que de modalités de la variable à prédire, ce sont des combinaisons linéaires de la forme suivante :



Cette présentation est séduisante à plus d’un titre. Il est possible, en étudiant la valeur et le signe des coefficients, de déterminer le sens des causalités dans le classement. De même, il devient possible, comme nous le verrons plus loin, d’évaluer le rôle significatif des variables dans la prédiction.

# Robustesse :

Les hypothèses de multi normalité et d’homoscédasticité peuvent sembler trop contraignantes, restreignant la portée de l’analyse discriminante linéaire dans la pratique.

La notion clé qu’il faut retenir en statistique est la notion de robustesse. Même si les hypothèses de départ ne sont pas trop respectées, une méthode peut quand même s’appliquer. C’est le cas de l’analyse discriminante linéaire. Le plus important est de le considérer comme un séparateur linéaire. Dans ce cas, si les nuages de points sont séparables linéairement dans l’espace de représentation, elle peut fonctionner correctement.

Par rapport aux autres techniques linéaires telles que la régression logistique, l’analyse discriminante présente des performances comparables. Elle peut être lésée néanmoins lorsque l’hypothèse d’homoscédasticité est très fortement violée.

# Taux d’erreur :

De manière classique en apprentissage supervisé, pour évaluer les performances d'une fonction de classement, nous confrontons ses prédictions avec les vraies valeurs de la variable à prédire sur un fichier de données. Le tableau croisé qui en résulte s’appelle une matrice de confusion avec : en ligne les vraies classes d’appartenance, en colonne les classes d’appartenance prédites. Le taux d’erreur ou taux de mauvais classement est tout simplement le nombre de mauvais classement, lorsque la prédiction ne coïncide pas avec la vraie valeur, rapporté à l’effectif du fichier de données.

Le taux d’erreur a de séduisant qu’il est d’interprétation aisée, il s’agit d’un estimateur de la probabilité de se tromper si l’on applique la fonction de classement dans la population.

Attention cependant, on parle de taux biaisé ou taux d'erreur en resubstituions, le taux d’erreur mesuré sur les données qui ont servi à construire la fonction de classement. Tout simplement parce que les données sont juges et parties dans ce schéma. La bonne procédure serait de construire la fonction de classement sur une fraction des données, dites d'apprentissage ; puis de l’évaluer sur une autre fraction de données, dite de test. Le taux d’erreur en test ainsi mesuré est un indicateur digne de foi.

La pratique veut que la répartition des données en apprentissage et test soit de 2/3 – 1/3. Mais en réalité, il n’y a pas de règle véritable. Le plus important est de concilier deux exigences contradictoires : en avoir suffisamment en test pour obtenir une estimation stable de l’erreur, tout en réservant suffisamment en apprentissage pour ne pas pénaliser la méthode d’apprentissage.

Lorsque les effectifs sont faibles, et que le partage apprentissage-test des données n’est pas possible, il existe des méthodes de rééchantillonnage telles que la validation croisée ou le Bootstrap pour évaluer l’erreur de classement.

# Séparabilité - Évaluation globale :

Le taux d’erreur permet d’évaluer et de comparer des méthodes, quelles que soient leurs hypothèses sous-jacentes. Dans le cas de l’analyse discriminante linéaire, nous pouvons exploiter le modèle probabiliste pour réaliser des tests d’hypothèses.

Un premier test permet de répondre à la question suivante : est-il possible de discerner les nuages de points dans l’espace de représentation. Rapporté dans le cadre multinormal, cela revient à vérifier si les centres de gravité conditionnels sont confondus (hypothèse nulle) ou si un au moins de ces centres de gravité s’écarte significativement des autres (hypothèse alternative).

La statistique du test est le de Wilks, son expression est la suivante :



où représente le déterminant de la matrice de variance covariance intra-classes, le déterminant de la matrice de variance covariance globale.

La table des valeurs critiques de la loi de Wilks étant rarement disponible dans les logiciels, on utilise couramment les transformations de Bartlett et de Rao qui suivent respectivement une loi du KHI-2 et de Fisher.

Avec un prisme différent, nous constatons que ce test peut s’exprimer comme une généralisation multidimensionnelle de l’analyse de variance à un facteur (**ANOVA**), on parle dans ce cas de **MANOVA** (Multidimensionnel Analysis of Variance).