République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

MEMOIRE

Présenté par

Boutassetta Messaouda

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Système de recommandation d'alimentation personnalisée intégrant la nutrition et la phytothérapie

Soutenu le : 12/ 06 / 2025 Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Présidente	Mme. Makhlouf Amina	MCB	Chadli Bendjedid El-Tarf
Rapporteur	**Mr. Chemam Chaouki**	**MCB**	**Chadli Bendjedid El-Tarf**
Examinatrice	Mme. Maâtallah Majda	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2024/2025

Remerciements

Je tiens à exprimer ma profonde gratitude à Dieu Tout-Puissant, qui m'a accordé la force, le courage et la persévérance nécessaires pour mener à bien ce travail.

J'adresse mes vifs remerciements à mon encadreur, Mr ***CHEMAM Chaouki****, pour ses conseils avisés, sa disponibilité et ses remarques constructives qui ont grandement enrichi ce mémoire.*

J'exprime ma reconnaissance aux membres du jury pour l'honneur qu'ils m'ont fait en acceptant d'évaluer ce travail, ainsi que pour le temps et l'attention qu'ils lui ont consacrés.

Je remercie les enseignants du département d'informatique pour leurs enseignements et leur transmission de savoir, qui ont posé les fondations de ce travail.

Merci à toutes les personnes qui, de près ou de loin, ont contribué et ont facilité l'aboutissement de ce projet.

Boutassetta Messaouda.

Dédicace

Aux êtres les plus chers à mon cœur :

À mes parents,

Que Dieu leur accorde une longue vie et une bonne santé.

À mon mari,

Pour sa patience, son soutien et ses encouragements.

À mes chers enfants,

Que Dieu les protège et leur accorde la droiture et la réussite.

À mes frères et sœurs et leurs enfants,

Qui sont toujours à mes côtés, prêts à m 'aider.

À l 'âme de ma très chère sœur Yasmina,

Qu 'Allah l 'accueille en Son vaste paradis et l 'entoure de Sa miséricorde.

À tous ceux qui m 'aiment et à tous ceux que j 'aime.

Je dédie ce travail avec tout mon amour.

Boutassetta Messaouda.

Table des matières

Liste des figures

Liste des tableaux

Introduction Générale

L'apprentissage automatique et les systèmes de recommandation sont devenus des outils incontournables dans notre monde connecté. Face à l'explosion des données et à la diversité des besoins individuels, les méthodes classiques de prise de décision ne suffisent plus à offrir des recommandations précises et adaptées. Grâce aux avancées en intelligence artificielle, il est aujourd'hui possible de concevoir des systèmes intelligents capables d'analyser des milliers de paramètres et d'optimiser les suggestions en temps réel. Que ce soit pour personnaliser l'alimentation, optimiser le bien-être, ou encore améliorer l'expérience utilisateur, ces technologies révolutionnent la façon dont nous interagissons avec l'information. En exploitant des algorithmes de machine learning, elles permettent de transformer des données brutes en connaissances exploitables, rendant chaque recommandation unique et parfaitement ajustée aux préférences et besoins de chacun. Cette capacité à anticiper et comprendre les attentes des utilisateurs fait des systèmes de recommandation un véritable moteur d'innovation, offrant des solutions personnalisées et adaptées à des domaines aussi variés que la santé, la nutrition et le commerce.

Dans un monde où la santé préventive et le bien-être sont devenus des priorités, la personnalisation de l'alimentation apparaît comme une approche incontournable. Chaque individu possède des besoins spécifiques en fonction de son âge, état de santé et objectifs nutritionnels, ce qui rend les recommandations standards souvent insuffisantes. La phytothérapie, avec ses propriétés naturelles et scientifiquement reconnues, joue un rôle essentiel dans l'optimisation de l'équilibre nutritionnel et le soutien de la santé. L'intégration d'un système de recommandation basé sur l'intelligence artificielle permet d'exploiter la puissance des données pour proposer des suggestions alimentaires adaptées, enrichies par l'usage des plantes. Grâce aux avancées en machine learning, ces systèmes peuvent apprendre des profils des utilisateurs et ajuster leurs recommandations en fonction des besoins spécifiques de chacun. Cette fusion entre technologie, nutrition et phytothérapie ouvre la voie à une nouvelle ère de santé personnalisée, où chaque choix alimentaire devient une opportunité d'améliorer le bien-être général de manière naturelle et ciblée.

Les projets antérieurs liés à ce domaine abordent la question de la nutrition sous l'angle de la valeur nutritionnelle (ex : la quantité de protéines et de vitamines dans les aliments), sans proposer de recette précise quant à la quantité des composants et à leur mode de préparation. Cependant, ces derniers n'intègrent pas la phytothérapie, que ce soit dans le traitement ou dans le cadre d'une perte ou d'une prise de poids.

Dans un monde en constante évolution, où la conscience de l'impact de l'alimentation sur la santé n'a jamais été aussi forte, les individus recherchent des solutions adaptées à leurs besoins spécifiques. La nutrition n'est plus perçue comme une simple nécessité biologique, mais comme un levier essentiel pour le bien-être et la prévention des maladies. Parallèlement, la phytothérapie, riche d'une tradition millénaire, gagne en popularité en tant qu'approche naturelle et complémentaire pour équilibrer l'organisme. Face à ces nouvelles attentes, il devient impératif de développer des **technologies intelligentes** capables d'accompagner cette transition vers une alimentation plus personnalisée et fonctionnelles. C'est dans ce contexte que les **systèmes de recommandation personnalisés** émergent, combinant **science des données et phytothérapie** pour offrir aux utilisateurs des suggestions adaptées à leur profil nutritionnel. Grâce aux avancées en intelligence artificielle et en apprentissage automatique, ces outils permettent non seulement de guider les choix alimentaires, mais aussi d'optimiser les bienfaits des plantes médicinales en fonction des besoins spécifiques de chacun, ouvrant ainsi la voie à une approche moderne et sur mesure de la santé par la nutrition.

Notre principal objectif dans cette étude est de développer un système de recommandation personnalisé pour une nutrition adaptée, reposant sur la phytothérapie. Ce système s'adresse aux utilisateurs d'une plateforme ou site Web, en tenant compte de leurs informations personnelles. Nous proposons des recommandations basées sur des données telles que le sexe, le poids, la tranche d'âge et l'état de santé. En cas de maladie spécifique, nous fournissons des suggestions incluant un ensemble d'ingrédients nutritionnels et à base de plantes, présentés sous forme de quantités et de méthode de préparation, c'est-à-dire sous forme de recettes, en fonction de leur objectif de prise ou de perte de poids.

Pour finir cette étude, nous avons préparé la structure suivante :

Chapitre 1 : L'apprentissage Automatique

Dans ce chapitre, nous aborderons plusieurs concepts fondamentaux. Nous commencerons par définir les termes clés et les concepts de base, incluant les algorithmes pertinents à notre domaine, tels que le classifieur bayésien, l'algorithme des k plus proches voisins (k-NN), les arbres de décision et les forêts aléatoires. Par la suite, nous examinerons les fouilles de données, en nous concentrant sur les motifs fréquents et les règles d'association. Enfin, nous décrirons les étapes d'un projet de machine learning, fournissant ainsi une vue d'ensemble complète de ce domaine fascinant et dynamique.

Chapitre 2 : Les systèmes de recommandation

Dans ce chapitre, nous explorerons les fondements des systèmes de recommandation, en commençant par leur définition et les différentes approches existantes, comme le filtrage

collaboratif et les méthodes basées sur le contenu. Nous analyserons également leur architecture, qui regroupe les mécanismes permettant la collecte, le traitement et l'exploitation des données. Enfin, nous aborderons les métriques d'évaluation essentielles pour mesurer leur efficacité, telles que la précision et le rappel, afin de mieux comprendre leur impact et leur fiabilité.

Chapitre 3 : Conception

Ce chapitre présente l'aspect conceptuel de notre étude. Il traitera du choix de l'ensemble de données utilisé pour entraîner le modèle, soulignant son rôle crucial dans l'efficacité des recommandations. Ensuite, il décrira l'architecture générale du système, en expliquant ses différentes composantes et leur interaction. Enfin, nous explorons les différents modèles de prédiction, en fournissant des explications sur leur fonctionnement et une évaluation de leur performance dans le cadre de la nutrition personnalisée.

Chapitre 4 : Implémentation

Ce chapitre pratique se concentre sur le développement d'un système de recommandation alimentaire, mettant en avant l'utilisation de Python et de bibliothèques comme Scikit-learn. Il décrit le processus sur une application démonstrative, illustrant comment générer des recommandations personnalisées. Les étapes clés incluent le prétraitement des données, le codage des variables, et la sélection des attributs. Enfin, il aborde l'exécution des algorithmes de recommandation pour fournir des suggestions pertinentes aux utilisateurs.

Chapitre 1 : L’apprentissage Automatique

1. Introduction

L'apprentissage automatique est devenu un domaine incontournable dans le traitement des données, révolutionnant la manière dont nous analysons et interprétons l'information. À l'ère du Big Data, la capacité à extraire des connaissances significatives à partir de vastes ensembles de données est essentielle pour prendre des décisions éclairées dans divers secteurs, allant de la finance à la santé. L’objectif de ce chapitre est d'explorer les fondements de l'apprentissage automatique, ses algorithmes clés et leur application dans notre domaine de recherche spécifique.

Dans ce chapitre, nous aborderons plusieurs concepts fondamentaux. Nous commencerons par définir les termes clés et les concepts de base, incluant les algorithmes pertinents à notre domaine, tels que le classifieur bayésien, l'algorithme des k plus proches voisins (k-NN), les arbres de décision et les forêts aléatoires. Par la suite, nous examinerons les fouilles de données, en nous concentrant sur les motifs fréquents et les règles d'association. Enfin, nous décrirons les étapes d'un projet de machine learning, fournissant ainsi une vue d'ensemble complète de ce domaine fascinant et dynamique.

2. Concepts de Base

A. Définition

L'apprentissage automatique (Machine Learning, ML) est une discipline de l'intelligence artificielle qui permet aux systèmes d'**apprendre à partir de données** sans être explicitement programmés pour une tâche spécifique. Contrairement aux approches traditionnelles, le ML utilise des algorithmes capables d'identifier des motifs complexes dans les données et de s'améliorer avec l'expérience. Ses applications couvrent des domaines variés, tels que la reconnaissance d'images, la prédiction financière ou les systèmes de recommandation. [7]

B. Différence entre Programmation Classique et ML

En **programmation classique**, les règles logiques sont codées manuellement par un développeur (ex : "si X alors Y"). En revanche, en **apprentissage automatique**, le modèle déduit ces règles à partir des données d'entraînement. Par exemple, un algorithme de ML pourra apprendre à reconnaître des chats dans des images sans qu'on lui ait explicitement défini les caractéristiques d'un chat [3]. Cette approche est particulièrement utile pour les problèmes complexes où les règles explicites sont difficiles à formaliser (ex : traitement du langage naturel). [8]

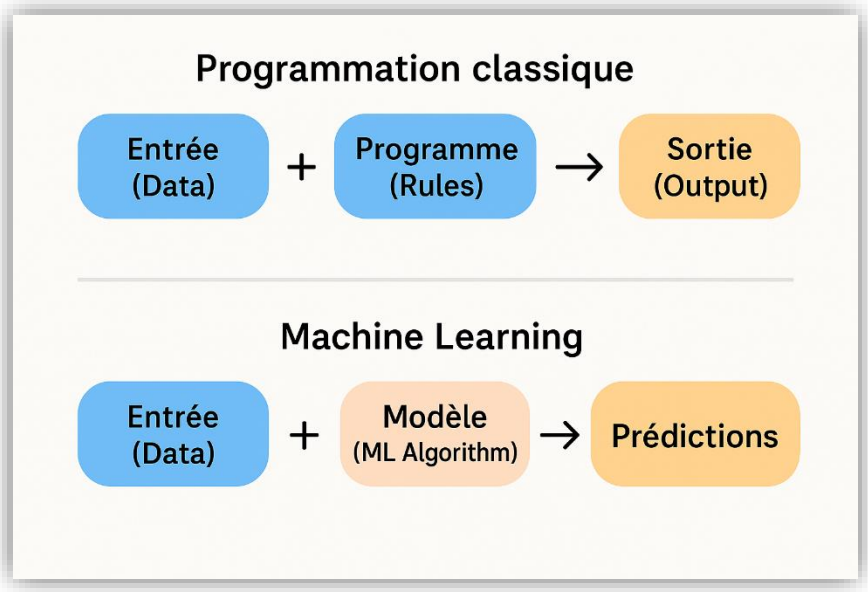
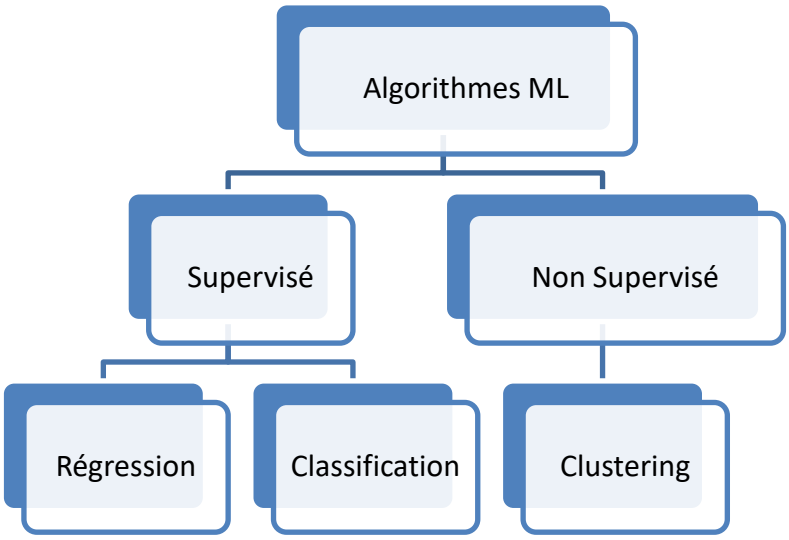


Figure 1. Programmation Classique vs. Machine Learning

C. Terminologie Clé

- **Features (Variables d'entrée)** : Données utilisées pour entraîner le modèle (ex : pixels d'une image).

- **Labels** : Résultats attendus en apprentissage supervisé (ex : "chat" ou "chien").
- **Modèle** : Fonction mathématique apprise à partir des données (ex : réseau de neurones).
- **Clustering** : Le clustering (ou regroupement) est une méthode non supervisée pour découvrir des groupes naturels dans les données.
- **Classification** : La classification consiste à attribuer une étiquette discrète (classe) à une observation. Contrairement à la régression, la sortie est une catégorie.
- **Régression** : La régression vise à prédire une valeur continue à partir des variables d'entrée. Elle est utilisée pour modéliser des relations entre des features et une cible numérique.



- **Sur-apprentissage (Overfitting)** : Le **sur-apprentissage** (ou *overfitting*) se produit lorsqu'un modèle de Machine Learning **mémorise les données d'entraînement** au lieu d'apprendre des motifs généralisables. Il obtient alors d'excellentes performances sur les données d'entraînement mais des résultats médiocres sur de nouvelles données (test).
- **Sous-Apprentissage (Underfitting)** : Le sous-apprentissage (ou underfitting) se produit lorsqu'un modèle de Machine Learning est trop simple pour capturer les relations complexes dans les données. Il obtient alors de mauvaises performances aussi bien sur les données d'entraînement que sur les données de test. [9]

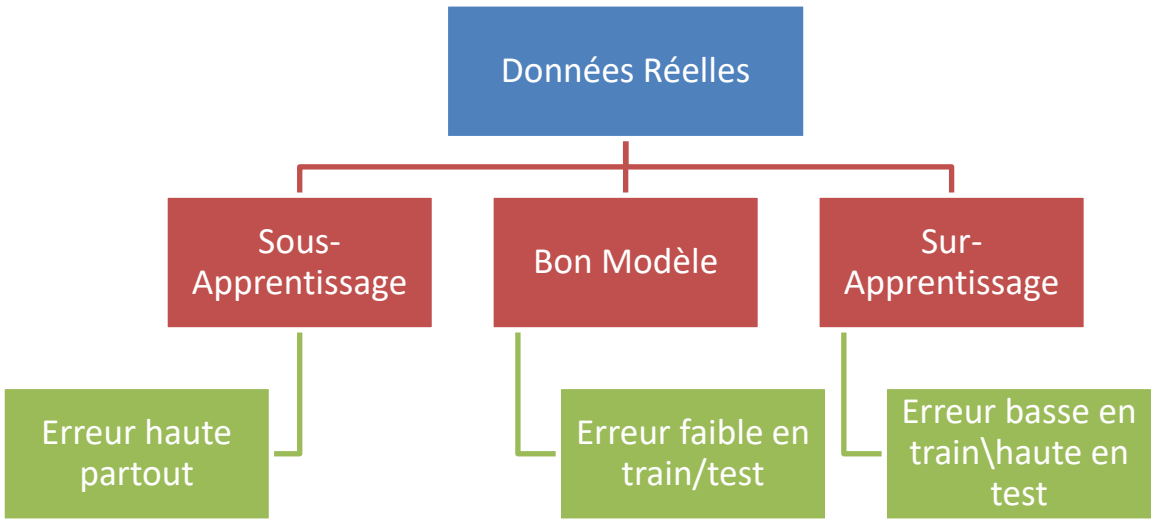


Figure 2. L'ajustement d'un modèle d'apprentissage automatique

D. Types d'Apprentissage Automatique

Apprentissage Supervisé (Supervised Learning) : est une approche de l'intelligence artificielle dans laquelle un modèle est entraîné à partir de données étiquetées. Chaque exemple du jeu de données est constitué d'une entrée et d'une sortie attendue, ce qui permet à l'algorithme d'apprendre les relations entre les variables. Ce type d'apprentissage est couramment utilisé pour la classification et la régression. Par exemple, il est utilisé pour prédire si un email est un spam ou non, ou pour estimer la valeur d'un bien immobilier en fonction de ses caractéristiques. [10]

Apprentissage Non Supervisé (Unsupervised Learning) : contrairement à l'apprentissage supervisé, n'utilise pas de données étiquetées. L'algorithme explore les données et identifie des motifs ou des structures sous-jacentes sans intervention humaine. Il est souvent employé pour la segmentation de données, comme le regroupement de clients en différents profils comportementaux ou la réduction de dimensions pour une meilleure visualisation. L'un des exemples les plus connus est l'algorithme K-Means, utilisé pour la classification automatique des données sans supervision explicite. [11]

Apprentissage par Renforcement (Reinforcement Learning)

L'apprentissage par renforcement est une méthode d'apprentissage où un agent interagit avec son environnement et ajuste ses actions en fonction des récompenses ou pénalités qu'il reçoit. Ce type d'apprentissage est inspiré de la psychologie comportementale et est souvent utilisé dans les jeux vidéo, la robotique et l'optimisation de systèmes complexes comme la gestion du trafic ou le trading algorithmique.[12]

3. Apprentissage Automatique

De nombreux algorithmes d'apprentissage automatique et des approches basées sur les fouilles de données existent. Dans cette section, nous nous concentrerons exclusivement sur ceux qui sont pertinents pour notre sujet de recherche. Nous examinerons certains algorithmes d'apprentissage automatique ainsi que les techniques d'extraction des motifs fréquents dans les fouilles de données, en fournissant des explications détaillées accompagnées d'exemples et de clarifications.

A. Classifieur Bayésien

Un classifieur bayésien (comme le Naive Bayes) est un modèle probabiliste qui utilise le théorème de Bayes pour prédire une classe.

1. Principe de Base

Le modèle calcule :

$$P(Classe \mid Features) = \frac{P(Features \mid Classe) \cdot P(Classe)}{P(Features)}$$

Où :

- P(Classe) = probabilité a priori
- P(Features|Classe) = probabilité des features pour une classe donnée.

Forces et Faiblesses

Avantages

- Rapide et efficace sur petits jeux de données.
- Gère bien les variables catégorielles.
- Interprétable (probabilités explicites).

Limites

- Hypothèse "naïve" : Suppose que toutes les features sont indépendantes, ce qui est rarement vrai (ex: *Health_Problem* et *Weight_Category* sont liés).
- Problème avec les classes rares : Si une combinaison n'existe pas dans les données (ex: Moderately_Obese + gain_weight), la probabilité sera 0.

B. Algorithme des k Plus Proches Voisins (k-NN)

L'algorithme k-NN (k-Nearest Neighbors) est une méthode d'apprentissage supervisé non paramétrique utilisée pour la classification et la régression. Il se base sur le principe que des individus similaires ont des comportements similaires.[11]

1. **Principe de Base**

- **Entrée** : Un jeu de données étiqueté (ex: patients avec leurs caractéristiques et recommandations).
- **Fonctionnement** :
    1. Pour un nouveau patient, calculer la **distance** (ex: Euclidienne, Manhattan) entre ses features.
    2. Sélectionner les **k** plus proches voisins.
    3. **Voter** pour la classe majoritaire parmi ces voisins (pour la classification).

2. **Forces et Faiblesses**

Avantages

- Simple à comprendre et à implémenter.
- Pas d'entraînement (mémoire des données).
- Adaptable (fonctionne avec des données catégorielles/numériques).

Limites

- Sensible au bruit (données aberrantes).
- Coûteux en mémoire (doit stocker tout le jeu de données).
- Choix de k critique :
    - Si k trop petit → Sensible au bruit.
    - Si k trop grand → Décision trop lisse.

3. Mesures de similarité (les Distances)[15]

- Distance Euclidienne : Espaces métriques standards (coordonnées, features normalisées).

$$D_{\text{Eucl}}(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Distance de Manhattan (L1) : Données avec outliers (moins sensible qu'Euclidienne).

$$D_{\text{Man}}(X,Y) = \sum_{i=1}^{n} |x_i - y_i|$$

- Distance de Minkowski : Généralisation d'Euclidienne (p=2) et Manhattan (p=1).

$$D_{\text{Min}}(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

- Similarité de Jaccard: Ensembles de catégories (ex: historique d'achats).

$$D_{\text{Jac}}(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$

- Distance Cosinus: Similarité entre textes (inclut de mots).

$$D_{Cos}(X,Y) = \frac{X \cdot Y}{\| X \| \| Y \|}$$

Chaque mesure est sélectionnée pour optimiser la prise en compte des spécificités des variables, qu'elles soient numériques, catégorielles ou textuelles, afin d'améliorer la pertinence des prédictions, on peut résumer l'ensemble de ce processus dans le tableau suivant :

Type de Données	Distance Recommandé	Cas d'Usage
Numériques	Euclidienne / Manhattan	Features continues (ex : poids, âge)
Catégorielles	Jaccard	Variables discrètes (ex : genre, catégorie d'âge)
Textuelles	Cosinus	Les données texte (des documents)

Choix des Mesures de Distance en Fonction du Type de Données

C. L'arbre de décision

L'arbre de décision est un algorithme d'apprentissage supervisé utilisé pour la classification et la régression. Il fonctionne en divisant les données en sous-ensembles selon des critères optimaux, souvent mesurés par des concepts tels que l'entropie et le gain d'information. Voici une explication détaillée.[13]

Principes essentiels de l'arbre de décision

L'algorithme suit une approche de partitionnement des données en fonction des caractéristiques les plus discriminantes. Voici l'algorithme sous les étapes suivantes :

a. Vérifier si tous les exemples appartiennent à la même classe.

b. Si non :

2.1 Calculer l'entropie initiale.

2.2 Calculer le gain d'information pour chaque attribut.

2.3 Sélectionner l'attribut avec le meilleur gain d'information.

2.4 Diviser les données en sous-ensembles.

2.5 Appliquer récursivement le processus.

1. **Entropie**

L'entropie mesure le niveau de désordre ou d'incertitude dans un ensemble de données. Une entropie élevée signifie que les classes sont mélangées et qu'il est difficile de prendre une décision. Une entropie faible signifie que les données sont bien séparées et qu'une décision est plus facile à prendre.

L'entropie d'un ensemble de données est définie par :

$$\text{Entropie}(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

où p_i est la proportion des éléments appartenant à chaque classe i.

2. **Gain d'information**

Le gain d'information est utilisé pour choisir la meilleure caractéristique permettant de diviser les données. Il est défini par :

$$\text{Gain}(S, A) = \text{Entropie}(S) - \sum_{v \in valeurs(A)} \frac{|S_v|}{|S|} \text{Entropie}(S_v))$$

Où A est une caractéristique utilisée pour diviser S, et S_v est le sous-ensemble des données après séparation.

Le critère est de choisir la caractéristique avec le plus grand gain d'information, car elle permet la meilleure réduction de l'incertitude.

Versions de l'algorithme : ID3, C4.5 et CART

- **ID3 (Iterative Dichotomiser 3)** : Utilise uniquement le gain d'information pour construire l'arbre. Il fonctionne avec des attributs catégoriques uniquement et est sensible aux données bruitées.
- **C4.5** : Amélioration de ID3, il peut gérer des attributs continus, utilise le gain d'information normalisé et prend en compte l'élagage de l'arbre pour supprimer les branches inutiles.
- **CART (Classification And Regression Trees)** : Peut être utilisé pour la classification et la régression. Il utilise l'impureté de Gini à la place de l'entropie et construit des arbres binaires uniquement.

D. Les forêts aléatoires

Les Random Forests (forêts aléatoires) sont une méthode d'apprentissage ensembliste (ensemble learning) qui combine plusieurs arbres de décision pour améliorer la performance et réduire le sur-apprentissage.[14]

- **Principe clé** :
  - **Bagging (Bootstrap Aggregating)** : Entraînement de plusieurs arbres sur des sous-ensembles aléatoires des données (avec remise).
  - **Feature Randomness** : Sélection aléatoire d'un sous-ensemble de caractéristiques (*features*) à chaque split.
- **Avantages** :
  - Réduction de la variance (moins de sur-apprentissage qu'un seul arbre).
  - Robustesse au bruit et aux données manquantes.
  - Pas besoin de normalisation des données.

Algorithmes et Fonctionnement

Étapes Clés :

1. **Bootstrap Sampling** : Création de N sous-ensembles (avec remise) à partir du jeu de données original.
2. **Construction des Arbres** :
   - Pour chaque sous-ensemble, un arbre de décision est entraîné.
   - À chaque split, seulement m features sont considérées (par défaut $m=\sqrt{p}$ pour la classification, $m=p/3$ pour la régression, où p = nombre total de features).

3. **Vote ou Moyenne** :
    - **Classification** : Vote majoritaire des arbres.
    - **Régression** : Moyenne des prédictions.

4. Fouilles des données

Les fouilles de données (Data Mining en anglais) se réfèrent à un ensemble de tâches, notamment la fouille des motifs et l'extraction de règles d'association. Dans notre étude, nous allons nous concentrer uniquement sur les aspects qui sont pertinents pour notre domaine de recherche.

A. Les motifs fréquents (*frequent itemsets*)

Le **motif fréquent** (ou en anglais frequent itemsets) désigne un ensemble d'items qui apparaissent ensemble dans un ensemble de données (par exemple, des transactions dans un panier d'achats) avec une fréquence supérieure à un seuil spécifié.

Un motif fréquent est un ensemble d'items X tel que le support de X est supérieur à un seuil minimum min_support. Le support d'un itemset est défini comme le ratio du nombre de transactions contenant cet itemset au nombre total de transactions.

Formule du Support

$$\textbf{Support}(\textbf{X}) = \frac{\textbf{Nombre de transactions contenant X}}{\textbf{Nombre total de transactions}}$$

Importance

Les motifs fréquents sont utilisés dans diverses applications telles que :

- L’analyse de panier d’achats pour comprendre les habitudes d'achat des consommateurs.
- La détection de modèles dans des données textuelles.
- L'analyse de données biomédicales et autres domaines.

Ces motifs aident à identifier des relations intéressantes et des tendances dans les données.

B. Type des motifs fréquents

Les motifs fréquents peuvent être classés en plusieurs types, selon leur structure et leur contexte d’utilisation. Voici les principaux types :

- **Motifs Fréquents Simples** : Ce sont des combinaisons d'items qui apparaissent ensemble dans les transactions. Par exemple, un motif fréquent pourrait être {pain, beurre}.

- **Motifs Fréquents Ordonnés** : Ces motifs tiennent compte de l'ordre dans lequel les items apparaissent. Par exemple, {pain} → {beurre} dans une séquence de transactions où le pain est acheté avant le beurre.
- **Motifs Fréquents Structurés** : Ces motifs représentent des relations complexes entre les items sous forme de graphes, permettant d'analyser des structures plus élaborées.
- **Motifs Fréquents avec Attributs** : Ces motifs incluent des informations supplémentaires sur les items, comme les caractéristiques des produits (ex. : couleur, taille).
- **Motifs Fréquents avec Contexte** : Ces motifs prennent en compte des informations contextuelles, comme le moment de l'achat (saisonnalité) ou le type de client (nouveau vs. régulier).
- **Motifs Fréquents Non-Redondants** : Ce sont des motifs qui ne contiennent pas de sous-ensembles fréquents. Par exemple, {pain, beurre} est minimal si ni {pain} ni {beurre} ne sont fréquents.
- **Motifs Fréquents avec Variabilité** : Ces motifs peuvent évoluer dans le temps, permettant de suivre les tendances d'achat et les changements de comportement des consommateurs.
- M**otifs maximaux :** sont un sous-ensemble des motifs fréquents qui ne sont pas inclus dans d'autres motifs fréquents plus grands. En d'autres termes, un motif maximal est un motif fréquent qui ne possède aucun super-ensemble fréquent.

Chacun de ces types de motifs fréquents peut être utilisé pour des analyses spécifiques, en fonction des objectifs de recherche et des caractéristiques des données.

C. Les règles d'association (association rules)

Une **règle d'association est** une implication qui décrit une relation entre deux ensembles d'items dans un ensemble de données. Elle est souvent utilisée dans le cadre de l'analyse de données pour découvrir des relations intéressantes entre des variables. [16]

Forme d'une Règle d'Association

Une règle d'association est généralement exprimée sous la forme :

$$\mathbf{A \Rightarrow B}$$

Où:

- A est l'antécédent (ou prémisse), un ensemble d'items.
- B est le conséquent (ou conclusion), un autre ensemble d'items.

Mesures Associées

Les règles d'association sont souvent évaluées à l'aide de deux mesures principales :

1. **Support** : Définit à quelle fréquence les items apparaissent ensemble dans les transactions.

$$\text{Support}(A \Rightarrow B) = \frac{\text{Nombre de transactions contenant } A \cup B}{\text{Nombre total de transactions}}$$

2. **Confiance** : Mesure la probabilité que B soit présent dans les transactions contenant A.

$$\text{Confiance}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

3. **Lift** : Évalue la force de l'association par rapport à une occurrence indépendante.

$$\text{Lift}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A) \times \text{Support}(B)}$$

Exemple

Considérons un exemple simple :

- Règle : **Si un client achète du pain (A), alors il achète également du beurre (B)**.
- Support : Mesure combien de fois le pain et le beurre sont achetés ensemble.
- Confiance : Mesure la proportion des clients qui achètent du beurre parmi ceux qui achètent du pain.

D. L'extraction des motifs fréquents

L'algorithme **Apriori** est un des algorithmes les plus populaires pour extraire des motifs fréquents d'un ensemble de données transactionnelles. Voici une description détaillée de son fonctionnement :

Algorithme Apriori

a. Principe de Base

L'algorithme repose sur le principe suivant : si un itemset est fréquent, alors tous ses sous-ensembles doivent également être fréquents. Cela permet d'éliminer les candidats non fréquents à chaque étape.

b. Étapes de l'Algorithme

1. **Initialisation** :
    - Définir un seuil de support minimum min_support.
    - Créer une liste de candidats d'items uniques (1-itemsets).
2. **Génération des Candidats** :
    - Pour chaque k-itemset trouvé, générer des candidats (k+1)-itemsets en combinant les k-itemsets fréquents.
3. **Évaluation des Candidats** :
    - Pour chaque candidat, calculer son support en comptant combien de transactions contient cet itemset.
    - Garder seulement ceux qui satisfont support ≥ min_support
4. **Itération** :
    - Répéter les étapes 2 et 3 jusqu'à ce qu'aucun nouveau motif fréquent ne soit trouvé.
5. **Extraction des Motifs Fréquents** :
    - Les motifs fréquents finaux sont ceux qui ont été retenus à la dernière itération.

c. Exemple d'Application

Supposons que nous ayons les transactions suivantes :

Transaction ID	Items
1	{A, B, C}
2	{A, B}
3	{B, C}
4	{A, C}
5	{A}

Si nous choisissons un seuil de support minimum de 2, l'algorithme procédera comme suit :

1. **1-itemsets** : Évaluer {A}, {B}, {C}.
    - Support : {A: 4, B: 3, C: 3} → Fréquents : {A, B, C}.
2. **Génération de 2-itemsets** :
    - Candidats : {A, B}, {A, C}, {B, C}.

- Support : {A, B: 2, A, C: 2, B, C: 2} → Tous sont fréquents.

3. **Génération de 3-itemsets** :
   - Candidat : {A, B, C}.
   - Support : {A, B, C: 2} → Fréquent.
4. **Fin de l'algorithme** :
   - Les motifs fréquents trouvés : {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}.

Autres algorithmes

Il existe de nombreux algorithmes dans ce domaine, et la recherche continue d'aboutir à l'émergence de méthodes plus efficaces. Pour des raisons de concision, nous ne détaillerons pas tous les algorithmes liés à l'extraction de motifs récurrents. Nous nous limiterons à présenter les plus populaires dans un tableau comparatif, qui mettra en évidence leurs principes, avantages et inconvénients, cas d'utilisation et complexité. Nous expliquons cela en détail ci-dessous :

Algorithme	Principe	Avantages	Limites	Cas d'Usage	Complexité
Apriori (Agrawal & Srikant, 1994)	Génération itérative de candidats et élimination basée sur le support minimal.	- Simple à implémenter. - Compatible avec tout type de données discrètes.	- Coûteux en mémoire (génération de candidats exponentielle). - Lent pour grands datasets.	- Analyse de paniers d'achat. - Données avec peu d'items uniques.	$O(2^N)$ (N = nombre d'items uniques)
FP-Growth (Han et al., 2000)	Construction d'un arbre compressé (FP-Tree) sans génération de candidats.	- Rapide (2x à 10x plus vite qu'Apriori). - Économe en mémoire.	- Complexité de construction de l'arbre. - Moins intuitif à déboguer.	- Grands datasets (ex: logs web, transactions retail).	$O(N \times T)$ (T = nombre de transactions)
Eclat (Zaki,	Exploration en	- Efficace sur données denses.	- Nécessite un pré-	- Données biomédicale	$O(T \times L)$ (L =

2000)	profondeur avec intersections d'ensembles de transactions.	- Meilleure localisation mémoire.	traitement coûteux (matrice verticale).	s (gènes-patients). - Réseaux sociaux.	taille moyenne des transactions)
LCM (Uno et al., 2004)	Utilisation de fermetures pour éviter les calculs redondants.	- Optimal pour motifs fermés/maximaux. - Évite les répétitions.	- Spécialisé (peu flexible pour d'autres types de motifs).	- Extraction de règles non redondantes (ex: diagnostics médicaux).	$O(N \times T)$
SAX-VSM (Senin & Malinchik, 2013)	Combinaison de Symbolique Agrégat approximation (SAX) et Vector Space Model.	- Adapté aux séries temporelles. - Réduction de dimension efficace.	- Perte d'information due à la discrétisation.	- Données temporelles (capteurs, ECG). - Détection d'anomalies .	

Tableau 1. Les algorithmes célèbres pour l'extraction des motifs fréquents dans littérature

5. Étapes d'un Projet de Machine Learning

La création d'un modèle de machine learning repose sur plusieurs étapes essentielles, allant de la collecte des données à leur déploiement. Voici une synthèse des étapes fondamentales :[w3]

A. Collecte des données

La qualité des données est primordiale pour la performance du modèle. Cette phase commence par la définition du problème et l'identification des sources de données pertinentes (bases de données, APIs, web scraping). Les données doivent être précises et représentatives afin d'améliorer la capacité de généralisation du modèle.

B. Prétraitement et nettoyage des données

Les données brutes doivent être transformées pour être exploitables. Cette étape comprend :

- La suppression des valeurs manquantes et aberrantes.
- L'encodage des variables catégoriques.
- La normalisation des données numériques.
- L'ingénierie des caractéristiques pour optimiser les performances du modèle.

C. Choix du modèle de machine learning

La sélection du modèle dépend de la nature du problème :

- **Classification** (ex. prédiction d'étiquettes).
- **Régression** (ex. estimation d'une valeur numérique).
- **Clustering** (ex. regroupement d'entités similaires). Il est essentiel d'évaluer la complexité et l'interopérabilité des algorithmes disponibles.

D. Entraînement du modèle

Le modèle est entraîné en utilisant les données prétraitées. L'algorithme ajuste ses paramètres pour minimiser l'écart entre les prédictions et les valeurs réelles. Des méthodes comme la descente de gradient permettent d'optimiser cet apprentissage et améliorer la capacité de généralisation du modèle.

E. Évaluation des performances

La performance du modèle est mesurée à l'aide de métriques adaptées :

- **Régression** : Erreur absolue moyenne (MAE), erreur quadratique moyenne (MSE), coefficient R^2.
- **Classification** : Précision, rappel, F1-score, matrice de confusion. L'évaluation permet d'identifier les faiblesses du modèle et d'apporter des ajustements.

F. Optimisation et réglage des hyperparamètres

Pour améliorer la performance du modèle, il est nécessaire d'ajuster ses hyperparamètres (ex. taux d'apprentissage, régularisation). Des méthodes comme **Grid Search** et **Cross-Validation** aident à identifier les meilleures configurations.

G. Déploiement et prédiction

Une fois le modèle optimisé, il est intégré dans un environnement de production où il peut effectuer des prédictions sur de nouvelles données. Des outils comme **Docker** et **Kubernetes** facilitent cette mise en production en assurant l'efficacité et l'évolutivité du modèle.

La construction d’un modèle de machine learning suit un processus structuré : collecte et nettoyage des données, sélection et entraînement du modèle, optimisation et déploiement. En appliquant ces étapes rigoureusement, il est possible de développer des systèmes de prédiction efficaces et adaptés aux besoins réels.

6. Conclusion

En conclusion, ce chapitre a mis en lumière les concepts essentiels de l'apprentissage automatique, soulignant son importance croissante dans l'analyse de données. Nous avons exploré les définitions et la terminologie de base, ainsi que plusieurs algorithmes clés, tels que le classifieur bayésien, l'algorithme des k plus proches voisins, les arbres de décision et les forêts aléatoires. De plus, nous avons examiné les fouilles de données, en mettant l'accent sur les motifs fréquents et les règles d'association, qui sont cruciaux pour extraire des informations pertinentes.

Nous avons vu comment ces techniques permettent d'améliorer la prise de décision et d'optimiser les processus dans divers domaines. En comprenant ces algorithmes et concepts, nous sommes mieux équipés pour aborder les défis posés par des ensembles de données complexes et variés.

Dans le chapitre suivant, nous explorerons les fondements des systèmes de recommandation. Nous commencerons par définir ce qu'ils sont et examinerons les différentes approches existantes, telles que le filtrage collaboratif et les méthodes basées sur le contenu. Nous analyserons également leur architecture, incluant les mécanismes de collecte, de traitement et d'exploitation des données. Enfin, nous aborderons les métriques d'évaluation essentielles, telles que la précision et le rappel, pour mieux comprendre leur impact et leur fiabilité.

Chapitre 2 : Les systèmes de recommandation

1. Introduction

Avec l'explosion des données numériques et la diversification des choix offerts aux utilisateurs, les systèmes de recommandation sont devenus indispensables dans de nombreux domaines tels que le commerce, la culture et la santé. Ces systèmes facilitent la prise de décision en proposant des suggestions personnalisées basées sur les préférences et les comportements des utilisateurs. Qu'il s'agisse de recommander un film, un livre ou un régime alimentaire adapté, leur rôle est de rendre l'accès à l'information plus efficace et pertinente.

L'apprentissage automatique offre une variété de techniques adaptées à des besoins spécifiques et à différents domaines de recherche. Ces techniques permettent de développer des modèles d'intelligence artificielle performants, optimisant ainsi leur capacité à traiter des données complexes et à produire des résultats pertinents. À mesure que la précision de ces modèles s'améliore, il est essentiel de réévaluer les retours d'information générés par ces systèmes, car ils influencent directement le processus d'apprentissage. De plus, les méthodes de formation doivent être adaptées pour tenir compte des nouvelles dynamiques introduites par des modèles plus précis, afin d'assurer une amélioration continue et une efficacité optimale. Ainsi, l'évolution de l'apprentissage automatique nécessite une approche flexible et proactive pour intégrer ces avancées et maximiser leur impact dans divers contextes d'application.

Dans ce chapitre, nous explorerons les fondements des systèmes de recommandation, en commençant par leur définition et les différentes approches existantes, comme le filtrage collaboratif et les méthodes basées sur le contenu. Nous analyserons également leur architecture, qui regroupe les mécanismes permettant la collecte, le traitement et l'exploitation des données. Enfin, nous aborderons les métriques d'évaluation essentielles pour mesurer leur efficacité, telles que la précision et le rappel, afin de mieux comprendre leur impact et leur fiabilité.

2. L’architecture d’un système de recommandation

A. Définition d’un système de recommandation

Un moteur de recommandation, ou système de recommandation, est une solution d'intelligence artificielle (IA) conçue pour faire des suggestions aux utilisateurs. Ces systèmes utilisent l'analyse de big data et des algorithmes de machine learning (ML) pour identifier des modèles dans les données de comportement des utilisateurs et formuler des recommandations pertinentes basées sur ces observations.

Les moteurs de recommandation aident les utilisateurs à découvrir des contenus, des produits ou des services qu'ils n'auraient pas forcément trouvés par eux-mêmes. Ces systèmes sont largement adoptés par de nombreuses entreprises en ligne pour stimuler les ventes et favoriser l'engagement, notamment sur des sites de e-commerce, des plateformes de streaming, des moteurs de recherche et des réseaux sociaux. [w1]

B. Modèle Utilisateur

Le modèle utilisateur est généralement représenté sous forme de matrice, semblable à un tableau contenant des données collectées sur l'utilisateur, liées aux produits disponibles sur le site web.

Un aspect essentiel est l'impact du temps sur le profil de l'utilisateur. Les intérêts des utilisateurs évoluent souvent avec le temps, ce qui nécessite un réajustement constant des données du modèle utilisateur afin de rester en phase avec leurs nouvelles préférences.

C. Liste de recommandations

Pour générer une liste de suggestions à partir d'un modèle utilisateur, les algorithmes s'appuient sur la notion de mesure de similarité entre objets ou personnes décrits par ce modèle. La similarité attribue une valeur numérique à la ressemblance entre deux éléments ; plus cette ressemblance est forte, plus la valeur de similarité est élevée. Inversement, une faible ressemblance se traduit par une valeur de similarité plus basse. Des exemples seront présentés ultérieurement.

D. L’architecture d’un système de recommandation

L’acquisition et le traitement des données sont des étapes fondamentales dans la conception d’un **système de recommandation**. Ces processus permettent de collecter, nettoyer et structurer les informations afin d’améliorer la pertinence des suggestions.

1. Acquisition des données

L'acquisition des données repose sur plusieurs sources (expliquer on détaille dans Collecte d'Information)

- **Données utilisateur** : Historique de navigation, préférences, interactions avec le système.
- **Données externes** : Bases de connaissances, informations nutritionnelles, propriétés des plantes médicinales.
- **Données biométriques** : Informations physiologiques et comportementales pour une personnalisation avancée.

2. Traitement des données

Une fois collectées, les données doivent être **nettoyées, transformées et analysées** pour garantir leur qualité et leur pertinence. Les étapes clés incluent :

- **Nettoyage des données** : Suppression des valeurs aberrantes et des incohérences.
- **Structuration** : Organisation des données sous forme de tables SQL ou de graphes sémantiques.
- **Analyse et prétraitement** : Normalisation, réduction de dimension et extraction des caractéristiques.

3. Intégration dans un système de recommandation

L'utilisation de l'un des modèles de systèmes de recommandation compatibles connus dans la littérature est expliquée en détail dans : Types de système de recommandation

3. Collecte d'Information

Pour qu'un système de recommandation soit efficace, il doit être capable de prédire les préférences des utilisateurs. Cela nécessite la collecte de données variées pour établir un profil individuel pour chaque utilisateur.

On peut distinguer deux types de collecte de données :[1]

Collecte de données explicite - Filtrage actif : Cette méthode repose sur le fait que l'utilisateur communique directement ses intérêts au système.

Exemple : Inviter l'utilisateur à commenter, étiqueter, noter, aimer ou ajouter en favoris des contenus qui l'intéressent. On utilise souvent une échelle de notation de 1 étoile (pas du tout aimé) à 5 étoiles (très aimé), que l'on convertit ensuite en valeurs numériques pour les algorithmes de recommandation.

- **Avantage :** Permet de reconstruire l'historique d'un utilisateur et d'éviter l'agrégation d'informations non pertinentes pour cet utilisateur spécifique (par exemple, plusieurs personnes utilisant le même compte).
- **Inconvénient :** Les données recueillies peuvent être sujettes à un biais de déclaration.

Collecte de données implicite - Filtrage passif : Cette méthode s'appuie sur l'observation et l'analyse des comportements des utilisateurs, sans qu'ils aient à fournir d'informations explicitement.

Exemples :

- Accéder à la liste des éléments écoutés, regardés ou achetés par l'utilisateur en ligne.
- Analyser la fréquence de consultation d'un contenu et le temps passé sur une page.
- Surveiller le comportement en ligne de l'utilisateur.
- Étudier son réseau social.
- **Avantage :** Aucune information n'est requise de la part des utilisateurs, toutes les données sont collectées automatiquement. Ces données sont généralement précises et exemptes de biais de déclaration.
- **Inconvénient :** Les données collectées sont plus difficiles à relier à un utilisateur spécifique et peuvent présenter des biais d'attribution (par exemple, un compte partagé entre plusieurs utilisateurs). Un utilisateur peut avoir acheté un livre sans l'apprécier ou l'avoir acheté pour une autre personne. [1]

4. Types de système de recommandation

Il existe cinq approches couramment utilisées pour les systèmes de recommandation :

1. Recommandation sans contexte (Popularité)
2. Recommandation personnalisée
3. Recommandation par contenu (Content-Based Filtering)
4. Recommandation sociale (Collaborative Filtering - Context Aware)
5. Recommandation hybride

A. Recommandation par Popularité

Cette approche consiste à recommander des objets en fonction de leur niveau de popularité. Les articles les plus en vogue sont proposés de manière identique à tous les utilisateurs, sans

personnalisation ni contexte. Les éléments sont classés selon leur popularité, puis recommandés en conséquence.

B. Recommandation Personnalisée

Cette méthode se base sur le comportement antérieur de l'utilisateur pour recommander des objets.

Exemples :

- Produits achetés ou sélectionnés sur une plateforme de e-commerce, ainsi que diverses actions effectuées par l'utilisateur, qui permettent de prédire de nouveaux articles qui pourraient l'intéresser.
- Les publicités, telles que celles d'AdSense de Google, sont des systèmes de recommandation personnalisée qui s'appuient sur les comportements passés de l'utilisateur (navigation, clics, historique de recherche).

C. Recommandation par contenu (Content-Based Filtering)

Les **systèmes de recommandation basés sur le contenu** fonctionnent en analysant les caractéristiques des éléments et en les comparant aux préférences des utilisateurs. Contrairement aux méthodes basées sur le filtrage collaboratif, cette approche ne nécessite pas une large communauté d'utilisateurs ni un historique important de leurs interactions.

Principe du filtrage basé sur le contenu

Chaque élément du catalogue possède des caractéristiques précises (attributs), telles que le genre, l'auteur, ou des mots-clés pour un livre. Ces informations sont stockées et exploitées pour établir un **profil utilisateur**, qui recense ses préférences sous les mêmes catégories.

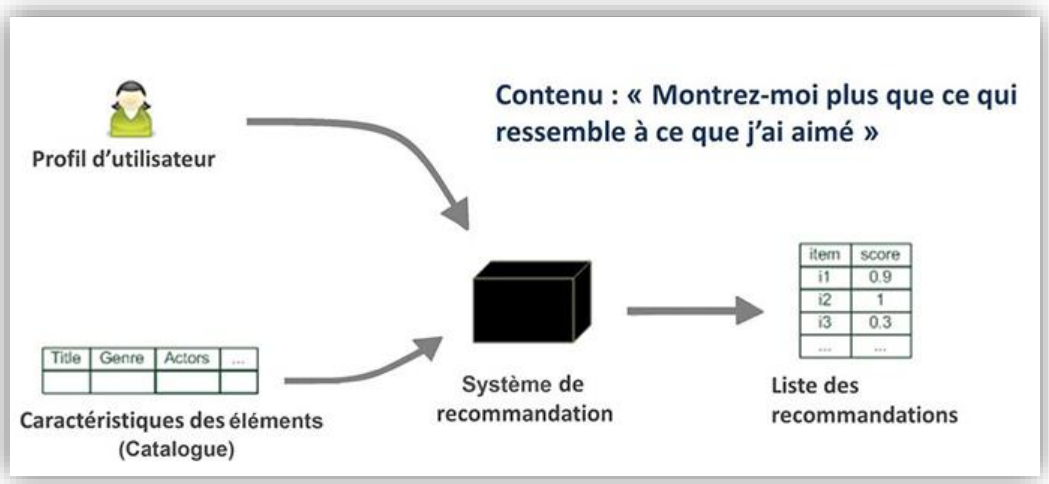


Figure 3. Un système de recommandation basé sur le contenu

La correspondance entre les préférences utilisateur et les caractéristiques des éléments peut être mesurée via des techniques comme :

- **Indices de similarité** (ex. indice de Dice) pour comparer les profils,
- **TF-IDF** (Term Frequency-Inverse Document Frequency) pour analyser les mots-clés pertinents,
- **Modèles bayésiens et arbres de décision** pour gérer de grands ensembles de données.

Avantages des recommandations basées sur le contenu

- Proposition d'éléments similaires à ceux déjà appréciés par l'utilisateur,
- Adaptation aux préférences individuelles sans dépendre des interactions des autres utilisateurs,
- Fonctionnement sur divers types de données (textuelles, numériques),
- Possibilité de recommander des objets **nouveaux** ou **peu populaires**,
- Absence de problèmes liés au **démarrage à froid**, car chaque élément peut être recommandé indépendamment de l'activité des autres utilisateurs.

Inconvénients et limites

- **Difficulté à gérer certains types de contenu** (ex. images, vidéos non accompagnées de texte descriptif),
- **Manque de diversité** : risque de sur-spécialisation dans les suggestions,
- **Complexité de gestion des profils utilisateurs** et de leur évolution dans le temps,
- **Système dépendant des scores d'intérêt** : les recommandations sont limitées si peu de données sont disponibles,
- **Nécessité de retours utilisateur** pour améliorer les suggestions, ce qui peut être contraignant.[w1]

D. Recommandation sociale (Collaborative Filtering)

Les **systèmes de recommandation collaboratifs** exploitent les comportements des utilisateurs ayant des goûts similaires pour proposer des suggestions pertinentes. L'idée centrale repose sur le fait que **si un utilisateur A partage des préférences avec un utilisateur B pour un élément X, alors il est probable qu'il ait aussi des préférences similaires pour un élément Y**.[3]

Ces systèmes collectent et analysent **les activités, préférences et historiques des utilisateurs** pour identifier des **corrélations** et anticiper les choix futurs. Ils sont largement utilisés dans des domaines comme **le streaming, l'e-commerce et les recommandations alimentaires**.[4]

Catégories de filtrage collaboratif

Le **filtrage collaboratif** peut être divisé en deux grandes approches :

1. **Mémoire (Memory-based)**
    - Fonctionne en temps réel en exploitant les données des utilisateurs précédents.
    - Exemple : Un site de streaming qui recommande un film basé sur les avis d'utilisateurs ayant aimé des films similaires.
2. **Modèle (Model-based)**
    - Utilise des algorithmes avancés comme les **réseaux neuronaux** ou les **modèles bayésiens** pour anticiper les préférences.
    - Exemple : Un système de recommandation de nutrition qui utilise des algorithmes prédictifs pour suggérer un régime alimentaire adapté aux utilisateurs ayant des profils physiologiques similaires.

Approche basée sur le voisinage (User-based nearest neighbor)

L'algorithme **User-Based Nearest Neighbor** identifie un groupe d'utilisateurs partageant des intérêts proches avec un utilisateur actif et s'appuie sur leurs évaluations pour effectuer des recommandations. [5]

Fonctionnement :

1. Sélection des utilisateurs voisins en analysant leurs préférences et interactions.
2. Calcul de la **similarité** entre ces utilisateurs et l'utilisateur actif.
3. Utilisation des notes attribuées par ces utilisateurs pour prédire les préférences de l'utilisateur actif.
4. Proposition des éléments les plus susceptibles de lui plaire.

Approche basée sur les éléments (Item-Based Nearest Neighbor)

L'approche **item-centrique** (ou item-to-item) propose une inversion du filtrage collaboratif user-centrique. Plutôt que de mesurer la similarité entre les utilisateurs, cette méthode analyse les corrélations entre les contenus eux-mêmes, en s'appuyant sur les évaluations attribuées par les utilisateurs.

1. Principe de fonctionnement

Les évaluations des utilisateurs sont exploitées pour identifier des éléments fortement corrélés. Si deux films, comme *Fargo* et *Pulp Fiction*, ont reçu des notes similaires de la part d'un groupe d'utilisateurs, le système peut prédire qu'un utilisateur qui a apprécié *Pulp Fiction* aimera probablement *Fargo*.

Contrairement à l'approche user-centrique, cette méthode effectue un traitement préalable des données et stocke les relations de similarité entre les éléments dans une **matrice de similarité**. Ce prétraitement permet de **réduire la charge computationnelle** et d'effectuer des recommandations en temps réel.

2. Construction d'une matrice de similarité des éléments

Le système détermine les éléments similaires avant de générer une recommandation. Cette approche est plus efficace en mémoire et permet des prédictions plus rapides que l'approche user-based .

Avantages et limites du filtrage collaboratif

Capacité d'adaptation aux préférences évolutives des utilisateurs Recommandations **précises** sans nécessiter de descriptions détaillées des éléments Applicable **à divers domaines** (films, nutrition, e-commerce, jeux vidéo…)

Démarrage à froid : Un nouvel utilisateur sans historique de préférences reçoit peu de recommandations.

Dépendance aux données : Si le nombre d'utilisateurs actifs est faible, les recommandations sont limitées.

Biais d'influence : Si un groupe d'utilisateurs a des goûts très spécifiques, les recommandations peuvent manquer de diversité.

E. Recommandation hybride

Les systèmes de recommandation hybrides combinent plusieurs approches pour améliorer la qualité des suggestions. Ils intègrent généralement des méthodes collaboratives et basées sur le contenu, exploitant ainsi les caractéristiques des éléments et les connaissances extérieures.

L'hybridation est née de l'évolution des systèmes de recommandation, cherchant à minimiser les limites des approches individuelles tout en capitalisant sur leurs avantages. L'idée principale est d'optimiser les recommandations en s'appuyant sur plusieurs sources d'information adaptées à chaque utilisateur ou contexte.

Il existe trois principales conceptions hybrides :

Monolithique – Fusion des techniques dans un seul algorithme.

Parallèle – Plusieurs systèmes fonctionnent indépendamment avant d'unifier leurs recommandations.[w1]

Tubulaire – La sortie d'un système devient une donnée d'entrée pour un autre.[w2]

1. Conception monolithique

Cette approche regroupe différents modèles en un seul algorithme. L'information peut être enrichie par des données externes ou une autre méthode de recommandation avant d'être exploitée.

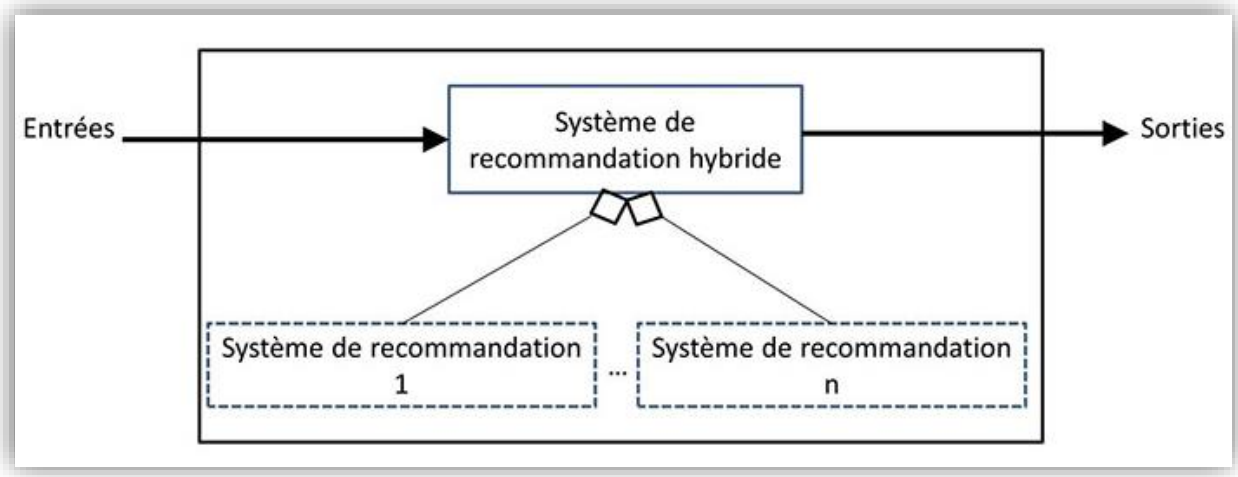


Figure 4. Conception d'hybridation monolithique

Exemple : Un système basé sur le contenu qui utilise aussi des données communautaires pour identifier des similitudes entre éléments.

2. Conception parallèle

Deux systèmes indépendants produisent séparément des listes de recommandations avant qu'elles soient fusionnées dans une phase finale.

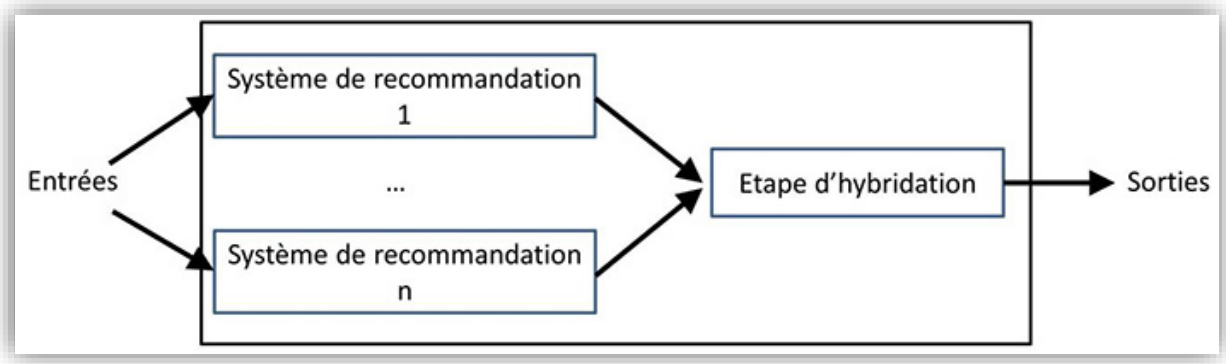


Figure 5. Conception d'hybridation parallèle

Exemple : Un algorithme basé sur les préférences utilisateurs et un autre basé sur des tendances générales proposent chacun une liste, qui est ensuite combinée.

3. **Conception tubulaire**

Un système de recommandation alimente un autre, où la sortie du premier devient l'entrée du second.

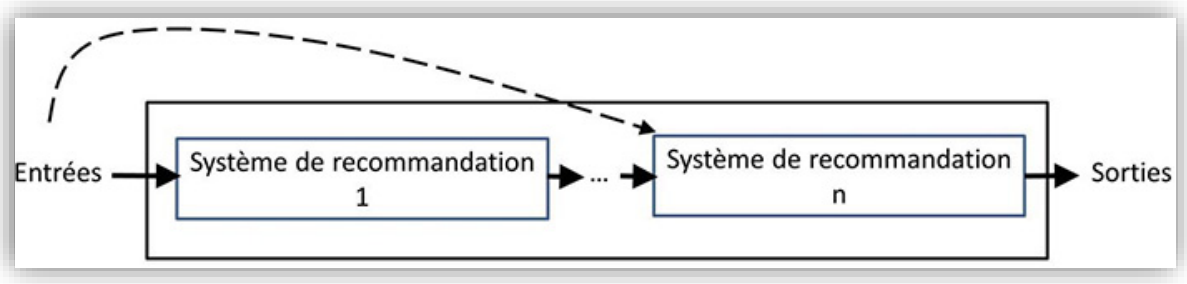


Figure 6. Conception d'hybridation tubulaire

Exemple : Un premier module filtre les préférences utilisateur, puis un second module ajuste les recommandations en fonction des données historiques.

Avantages des systèmes hybrides

Meilleure précision en combinant plusieurs sources d'information Réduction des limites propres à chaque technique Adaptabilité aux différentes situations et préférences

Inconvénients et défis

Complexité accrue dans la mise en œuvre Besoin de plus de données pour assurer une cohérence optimale Coût computationnel élevé selon les modèles utilisés

5. Évaluation d'un système de recommandation

L'efficacité d'un **système de recommandation** repose sur plusieurs métriques permettant de mesurer la pertinence des suggestions fournies aux utilisateurs. Deux indicateurs fondamentaux sont utilisés : **la précision** et **le rappel**. [6]

1. Précision

La précision représente la **qualité des recommandations**, c'est-à-dire la proportion des suggestions réellement pertinentes par rapport à l'ensemble des recommandations effectuées.

$$\text{Précision} = \frac{\text{nombre de suggestions pertinentes}}{\text{nombre total de suggestions}}$$

Une haute précision signifie que les éléments recommandés sont généralement bien alignés avec les attentes des utilisateurs.

2. Rappel

Le **Rappel** mesure la capacité du système à recommander un maximum d'éléments pertinents parmi tous les contenus potentiellement intéressants pour l'utilisateur.

$$\text{Recall} = \frac{nombre\ de\ suggestions\ pertinentes\ propose}{nombre\ total\ d'elements\ pertinents\ disponibles}$$

Un recall (rappel) élevé signifie que l'algorithme parvient à suggérer une grande partie des éléments pertinents présents dans la base de données.

3. Évaluation statistique de l'algorithme

Outre la précision et le rappel, on peut utiliser des **outils statistiques** pour affiner l'évaluation des recommandations. Parmi ces méthodes, l'**Erreur Moyenne Absolue (MAE)** est largement employée.

La MAE permet de comparer les **prévisions du système** avec les choix réels des utilisateurs. Elle mesure la **différence moyenne** entre les notes prédites et les notes réelles attribuées par les utilisateurs.

$$MAE\ = \frac{1}{n}\sum_{i}^{n} |prediction - reelle|$$

Plus la MAE est **faible**, plus les prédictions sont précises et proches des préférences réelles des utilisateurs.

Importance de ces métriques

Ces indicateurs permettent aux concepteurs de **systèmes de recommandation** d'ajuster les modèles afin d'optimiser leur **pertinence** et leur **efficacité**. Un **équilibre** entre précision et recall est essentiel pour garantir une expérience utilisateur satisfaisante.

6. Conclusion

Dans ce chapitre, nous avons ainsi présenté les principes fondamentaux des systèmes de recommandation, leur fonctionnement et les différentes approches qui permettent d'optimiser la pertinence des suggestions. Nous avons également exploré l'architecture de ces systèmes, qui repose sur des modèles analytiques et statistiques, ainsi que les techniques d'évaluation qui garantissent leur fiabilité.

À travers cette analyse, nous constatons que les **systèmes de recommandation** offrent une solution efficace pour filtrer la surcharge d'informations et améliorer l'expérience utilisateur. Ils représentent un levier clé pour la personnalisation des services numériques et leur optimisation, mais leur mise en œuvre requiert une gestion rigoureuse des données, une protection de la vie privée et une adaptation constante aux besoins des utilisateurs.

Dans le chapitre suivant, nous appliquerons ces concepts à un **système de recommandation d'alimentation personnalisée intégrant la nutrition et la phytothérapie**. Cette approche combinera des modèles algorithmiques et des connaissances scientifiques pour proposer des régimes adaptés aux besoins individuels, en tenant compte des préférences et des objectifs de santé de chaque utilisateur.

Chapitre 3 : Conception

1. Introduction

Dans un monde en évolution, la prise de conscience de l'impact de l'alimentation sur la santé est croissante. La nutrition est désormais considérée comme un levier pour le bien-être et la prévention des maladies. La phytothérapie, avec sa tradition millénaire, devient populaire comme approche naturelle pour équilibrer l'organisme. Face à ces nouvelles attentes, il est crucial de développer des technologies intelligentes pour une alimentation plus personnalisée. Les systèmes de recommandation personnalisés, alliant science des données et phytothérapie, émergent pour offrir des suggestions adaptées aux profils nutritionnels des utilisateurs. Grâce à l'intelligence artificielle et à l'apprentissage automatique, ces outils optimisent les choix alimentaires et les bienfaits des plantes médicinales, favorisant une approche moderne et sur mesure de la santé.

Ce chapitre présente l'aspect conceptuel de notre étude. Il traitera du choix de l'ensemble de données utilisé pour entraîner le modèle, soulignant son rôle crucial dans l'efficacité des recommandations. Ensuite, il décrira l'architecture générale du système, en expliquant ses différentes composantes et leur interaction. Enfin, nous explorons les différents modèles de prédiction, en fournissant des explications sur leur fonctionnement et une évaluation de leur performance dans le cadre de la nutrition personnalisée.

2. Objectif

Le moteur de recommandation repose sur une approche data-driven, exploitant des paramètres clés du profil utilisateur : Le désir, Poids, Tranche d'âge, État de santé

Lorsqu'un utilisateur présente des conditions spécifiques, le système génère des recommandations personnalisées sous forme de recettes, comprenant : Ingrédients nutritionnels et phytothérapeutiques adaptés à ses besoins Quantités optimisées pour atteindre l'objectif de prise ou de perte de poids Méthode de préparation afin de maximiser l'efficacité des ingrédients, le graphique ci-dessous illustre ce processus.

Figure 7. L'objectif de recommandation personnalisée à base d'alimentation et phytothérapie

3. Choix de data set

Comme pour toute initiative d'apprentissage automatique ou d'exploration de données, il est essentiel de disposer d'un ensemble de données pour entraîner notre modèle à formuler des prédictions ou des recommandations. Cet ensemble de données représente la matière première nécessaire au développement du système.

Il peut être constitué à partir d'une seule activité ciblée ou, au contraire, être agrégé à partir d'un ensemble diversifié d'activités. Par exemple, des données peuvent être collectées à partir de questionnaires sur les préférences alimentaires, d'enquêtes de santé ou même d'interactions sur la plateforme. En combinant ces différentes sources, nous pouvons enrichir notre base de données, ce qui permettra de mieux comprendre les besoins des utilisateurs et d'améliorer la précision des recommandations fournies.

Ainsi, la qualité et la diversité des données sont cruciales pour le succès du modèle, car elles influencent directement sa capacité à générer des résultats pertinents et personnalisés.

A. Data set disponibles

Au cours de nos recherches, nous avons identifié les ensembles de données suivants :

Nutrition Datasets sur Kaggle

Ce dataset contient des informations sur les valeurs nutritionnelles des aliments, y compris les macronutriments tels que les calories, les protéines, les glucides et les lipides. Ces données sont particulièrement utiles pour la planification diététique et l'analyse nutritionnelle.[w4]

Food Nutritional Values Dataset

Cette base de données fournit des détails sur les valeurs nutritionnelles des aliments, incluant des informations sur les vitamines, minéraux et antioxydants. Elle est conçue pour les professionnels de la santé et les chercheurs.

Nutritional Food Dataset

Ce dataset regroupe des données sur les macronutriments et micronutriments des aliments, ainsi que des informations sur les bienfaits pour la santé et les indices glycémiques. Il est adapté aux nutritionnistes et aux chercheurs en alimentation.

Cependant, nous avons constaté que ces ensembles de données ne correspondaient pas tout à fait à un système de recommandation pour une nutrition personnalisée axée sur la phytothérapie. Par conséquent, nous avons décidé qu'il serait plus approprié de créer un ensemble de données spécifiquement conçu pour cet objectif.

B. La génération de data set

Pour entraîner un modèle capable de fournir des recommandations nutritionnelles adaptées au profil des utilisateurs, nous avons élaboré un ensemble de données, comme illustre le tableau suivant :

id	Age_Group	Gender	Weight_Category	target_goal	Health_Problem	Foods	Herbs	ingredients	preparation_method

1	Teens	Female	Overweight	lose_weight	Obesity	Spinach + Chicken Breast + Quinoa	Green Tea + Garcinia Cambogia + Dandelion	2 cups spinach, 1 chicken breast, 1/2 cup quinoa, 1 green tea bag, 500mg Garcinia Cambogia, 1 tsp Dandelion root	1. Steam spinach. 2. Grill chicken breast. 3. Cook quinoa. 4. Brew herbal tea mixture. 5. Combine all ingredients.
103	Early Adulthood	Male	Underweight	gain_weight	Nutrient Deficiency	Chicken + Eggs + Rice	Ashwagandha + Fenugreek + Maca	2 chicken breasts, 3 eggs, 1 cup rice, 1 tsp Ashwagandha, 1 tsp Fenugreek, 1 tsp Maca	1. Cook chicken thoroughly. 2. Boil eggs. 3. Prepare rice. 4. Mix herbs into warm water and let steep for 10 minutes. 5. Combine all ingredients and serve.
….	…..	…..	…..	…..	…..	…..	…..	…..	…..

Tableau 2. Exemple d'ensemble de données sur les recommandations nutritionnelles et phytothérapeutiques

Exprimer les données sous forme de catégories présente de nombreux avantages. Tout d'abord, cela améliore la clarté et la compréhension des informations, en permettant de résumer des données complexes en groupes significatifs. Cela aide à visualiser des tendances et à rendre les résultats plus accessibles. Ensuite, l'analyse devient simplifiée, car les données catégorisées permettent d'identifier rapidement des patterns et des corrélations, ce qui améliore la prise de décision.

Voici une explication des colonnes de catégories dans le dataset : voir l’annexe ci-dessous

1. **Catégorie d'Âge**

Cette colonne classe les utilisateurs en fonction de leur tranche d'âge. Chaque catégorie représente une période de la vie, la tranche d’âge d’une personne influence significativement ses objectifs de prise ou de perte de poids en raison de facteurs biologiques, métaboliques,

hormonaux et psychosociaux. Voir l'annexe pour plus de détails sur les catégories définies et la tranche d'Âge.

2. **Classification du Poids**

Cette colonne classe les utilisateurs selon leur poids. Chaque catégorie définit une plage de poids, Cette colonne joue un rôle crucial dans la détermination du désir de l'utilisateur de perdre ou de prendre du poids. Nous pouvons conclure que la catégorie de poids faible voudra prendre du poids, tandis que la catégorie de poids moyen maintiendra son poids et n'apparaîtra pas dans notre ensemble de données. La catégorie de poids élevé est supérieure à la moyenne et son désir de perdre du poids peut être déterminé. Voir l'annexe pour plus de détails sur les catégories définies et les plages de Poids.

3. **Problèmes de santé**

En ce qui concerne cette colonne, nous nous sommes concentrés sur certaines des maladies les plus courantes. Lors de la distribution, il faut tenir compte du conflit de certains composants nutritionnels avec les composants à base de plantes et des composants à base de plantes avec l'état de santé de l'utilisateur (par exemple, suggérer des composants nutritionnels contenant des sucres pour un patient diabétique), ainsi que du conflit de leur exposition avec la catégorie de poids, par exemple, vous ne trouverez pas une personne souffrant d'obésité et souffrant d'un problème de santé tel qu'une mauvaise croissance.

4. **Suggestions nutrition et phytothérapie**

Dans cette partie de l'ensemble de données, nous avons réalisé des recherches sur les ingrédients à utiliser pour ceux qui souhaitent perdre ou prendre du poids. Chaque colonne présente un maximum de trois ingrédients, permettant à l'utilisateur d'exclure ceux qui ne sont pas disponibles. Nous sommes conscients que certains ingrédients, en particulier les plantes, peuvent être rares. Selon la saison et les ingrédients nutritionnels et végétaux indiqués dans la colonne correspondante, les quantités et la méthode de préparation de cette combinaison seront présentées sous forme de recette, les colonnes dans le data set « Foods», « Herbs», « ingredients», « preparation_method» correspondes respectivement a les aliments, les herbes, les ingrédients utilisés et la méthode de préparation. Voir l'annexe pour plus de détails.

Pour créer un ensemble d'exemples pour les modèles d'apprentissage et les étiqueter comme perte de poids ou gain de poids, il suffit de prendre les valeurs uniques des classifications catégorielles de poids, de sexe, d'âge et de problème de santé et de faire une jointeur entre ces colonnes comme dans le langage SQL ou un langage de programmation spécifique en utilisant une fonction préparée ou définie comme certaines fonctions aléatoires pour obtenir toutes les

combinaisons possibles, avec la suppression des exemples contenant des valeurs contradictoires pour le poids, le désir et l'état de santé.

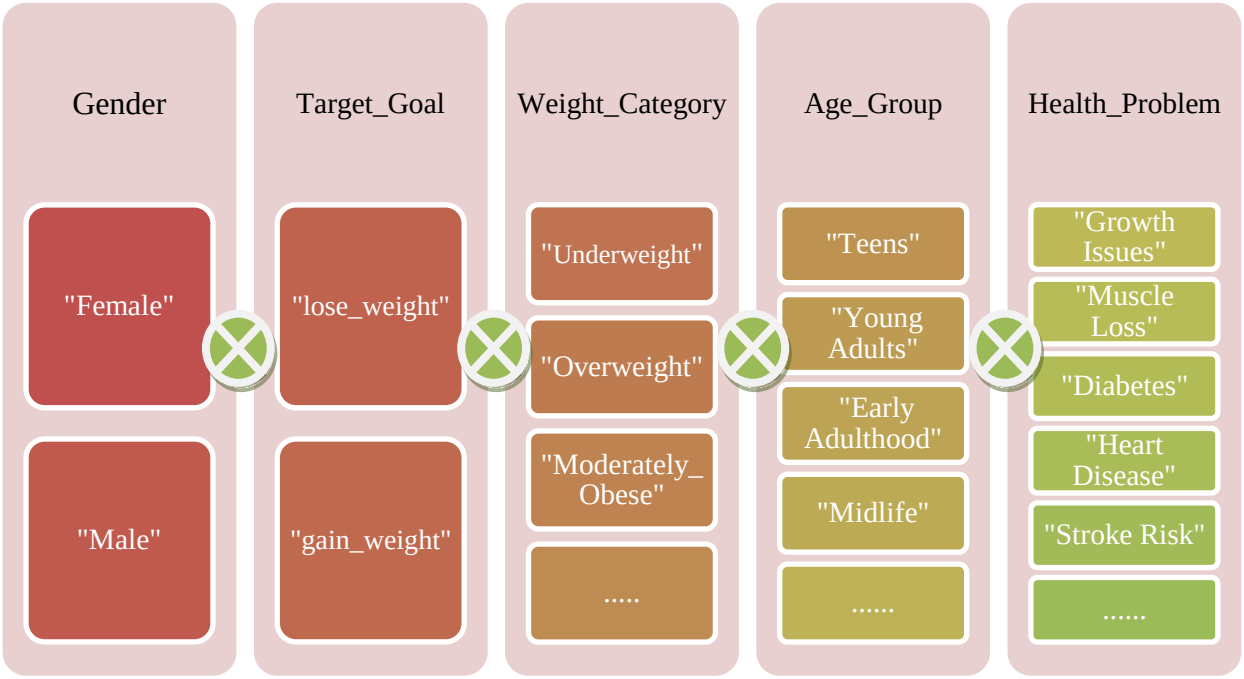


Figure 8. modèle de construction de l'ensemble d'apprentissage

4. Choix de modèle

Dans cette section, nous élaborons l'architecture de notre système de recommandation et des algorithmes d'apprentissage automatique et de fouille des données appropriés.

A. L'architecture du système

Notre modèle devrait recommander deux types d'ingrédients : des ingrédients à base de plantes et des ingrédients nutritionnels, situés dans différentes colonnes de l'ensemble de données. Étant donné qu'il n'est pas possible d'entraîner un seul modèle pour suggérer les deux en même temps, nous préférons former un modèle distinct pour la recommandation d'ingrédients nutritionnels et un autre pour les ingrédients à base de plantes. Ensuite, nous combinerons les résultats obtenus, comme illustré dans la figure suivante.

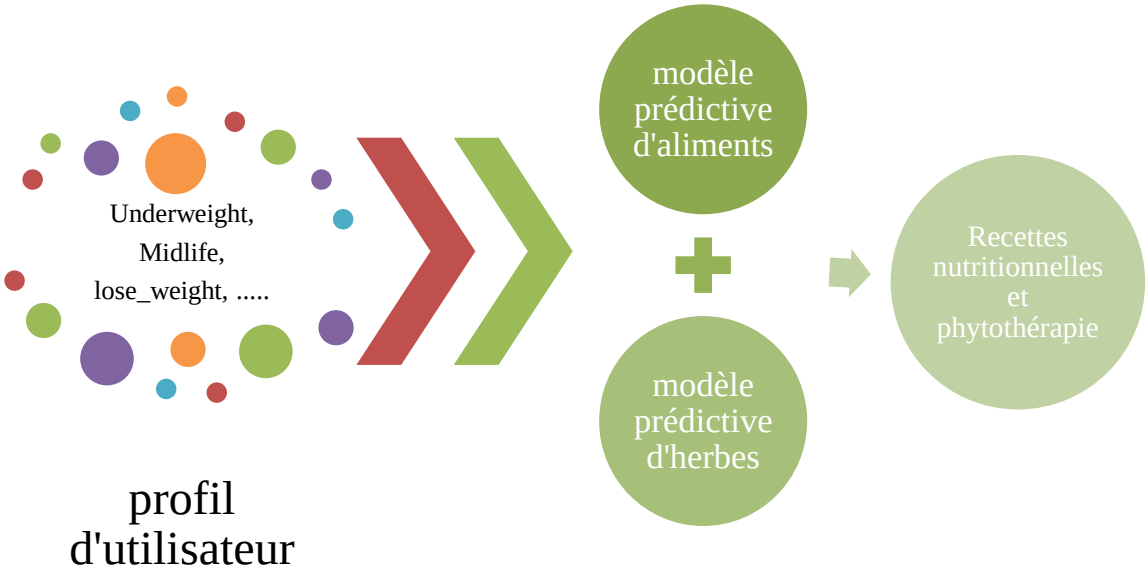


Figure 9. L'architecteur du système de recommandation personnalisée pour la nutrition basée sur la phytothérapie

B. Le choix d'algorithme

Ci-dessous, nous examinerons l'ensemble des algorithmes pertinents pour notre système de recommandation et la possibilité de les mettre en œuvre dans notre cas, tout en discutant de leurs avantages et inconvénients.

Avant d'appliquer l'un des algorithmes suivants, il est nécessaire d'identifier les variables, la variable d'entrée x=[Age_Group , Gender , Weight_Category , Target_Goal , Health_Problem] et la variable de sortie y= [Foods] ou y= [Herbs], puis l'encodage des données à l'aide de l'un des modèles d'encodage expliqués par la suite .

1. Classificateur Bayésien (Naïve Bayes)

Pour classer un nouvel utilisateur U(Teens, Male, Underweight, gain_weight, Growth Issues), on calcule la probabilité qu'il ait besoin d'une herbe spécifique comme Fenugreek ou Green Tea :

$$P(\text{Fenugreek} \mid U) = \frac{P(U \mid \text{Fenugreek}) \times P(\text{Fenugreek})}{P(U)}$$

Ici, on choisit la classe ayant la plus haute probabilité.

🔹 Exemple : calcul des probabilités P (Fenugreek | U) = 0.75, P (Green Tea | U) = 0.20, P (Garcinia Cambogia | U) = 0.05, L'algorithme prédit " Fenugreek " car il a la probabilité la plus élevée.

Mais le défaut de ce modèle est qu'il ne suppose aucune corrélation entre les variables, et c'est le contraire que nous avons. En fait, l'ensemble des variables, telles que le poids, la tranche

d'âge et le problème de santé, détermine le désir d'une personne de prendre ou de perdre du poids.

2. K plus proche voisin

Pour prédire la valeur dans cette approche il faut :

Calcul des Distances : Pour un nouvel utilisateur U (Teens, Male, Underweight, gain_weight, Growth Issues), on calcule la distance avec tous les autres

$$D_{\text{Eucl}}(U, Q) = \sqrt{\sum \left(\text{Age}_u - \text{Age}_q\right)^2 + \left(\text{Weight}_u - \text{Weight}_q\right)^2 + \cdots + \left(\text{Gender}_u - \text{Gender}_q\right)^2}$$

Sélection des k Voisins (on a sélectionné Herbes comme variable de sortie)

- Si **k=3**, on prend les 3 patients les plus proches.
- Exemple :
    - Voisin 1 : [Teens, Male, Underweight, gain_weight, Growth Issues] → (Ashwagandha + Fenugreek + Maca)
    - Voisin 2 : [Teens, Female, Underweight, gain_weight, Iron Deficiency] → (Ashwagandha + Fenugreek + Maca)
    - Voisin 3 : [Young Adults, Male, Underweight, gain_weight, Nutrient Deficiency] → (Green Tea + Garcinia Cambogia + Guarana)

Donc **Vote Majoritaire**: 2 votes pour (Ashwagandha + Fenugreek + Maca)→ Résultat.

Nous avons exécuté un scénario de test de l'algorithme du K plus proche voisin sur notre ensemble de données. Chaque fois que nous avons modifié la valeur de K, nous avons enregistré la précision de prédiction pour les divisions d'entraînement et de test. Nous avons obtenu la courbe suivante :

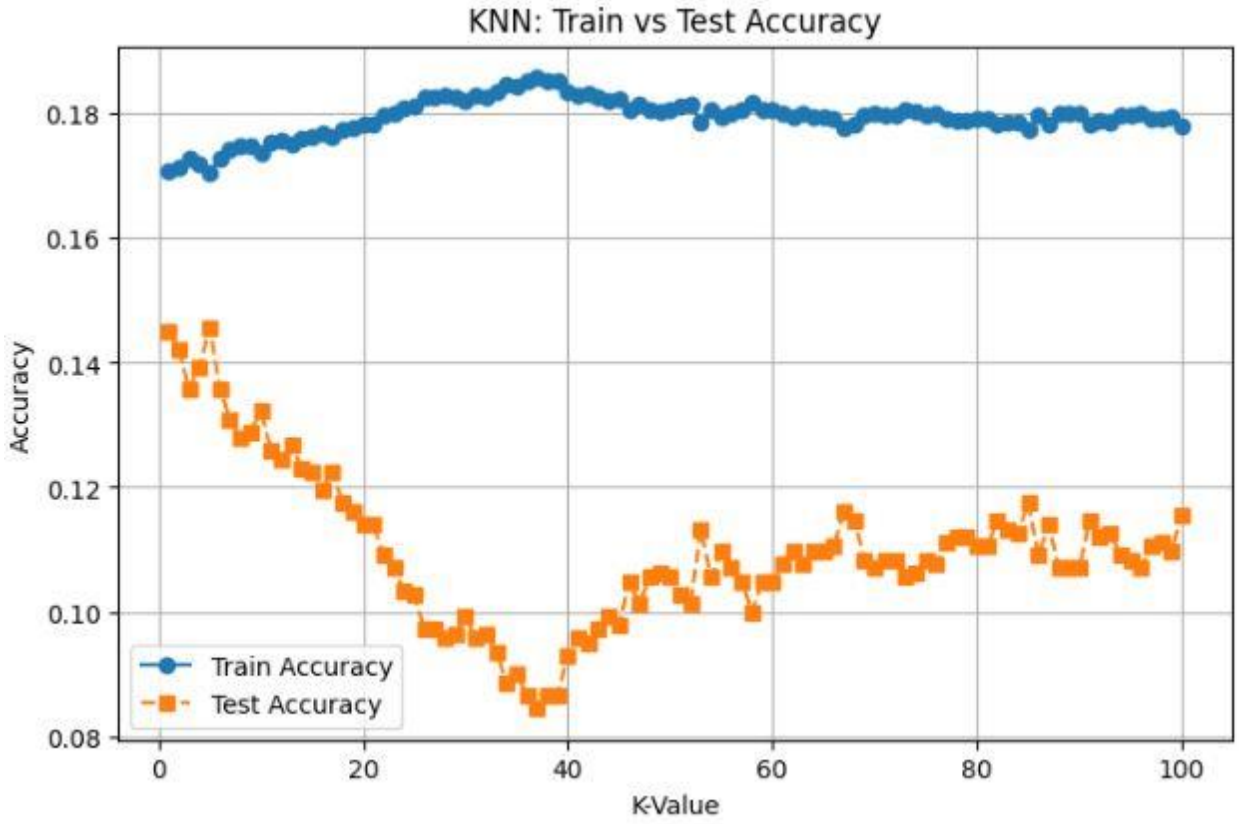
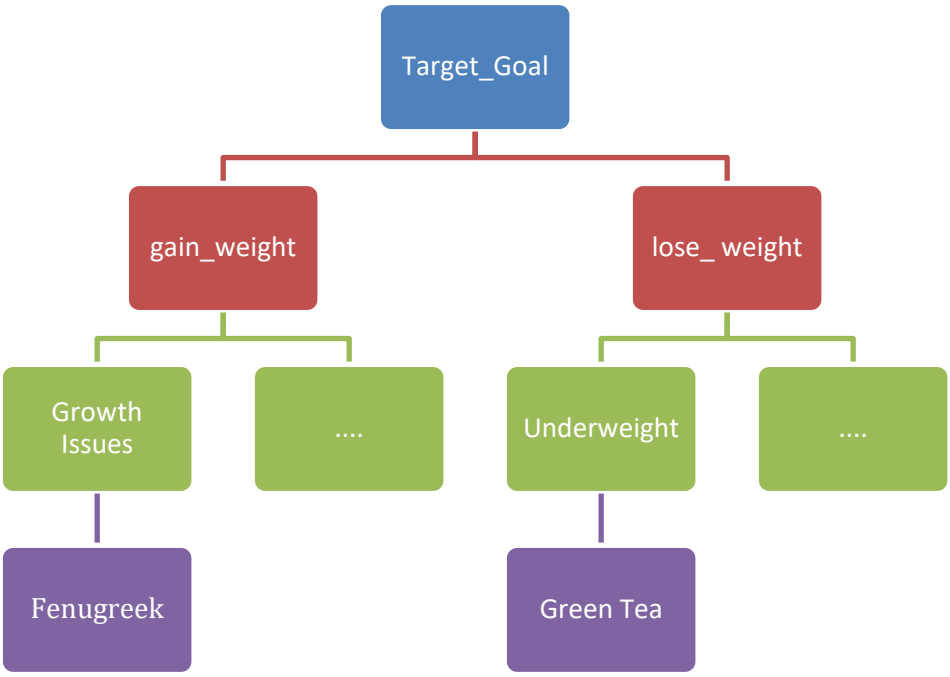


Figure 10. La performance de l'algorithme de kNN test et entraînement

Mauvaise performance d'algorithme en précision ne dépasse pas 20% dans les deux divisions pour être acceptable il doit dépasser 50% cela est dû à la nature des colonnes dans l'ensemble de données qui contiennent des valeurs avec une relation d'ordre comme le poids et le groupe d'âge, et d'autres sans relation d'ordre comme les composants qu'ils soient alimentaires ou à base de plantes ou les problèmes de santé sont des étiquettes, ici le calcul de la distance ne fait pas une grande différence

3. L’arbre de décision

Dans cet algorithme, l’arbre de décision est créé en choisissant l'attribut le plus précieux en termes de gain d’informations. L'ensemble de données est divisé en fonction des différentes valeurs de cet attribut. Par exemple, si nous choisissons l’attribut de désir de poids, nous aurons deux divisions homogènes de l'ensemble de données sous la forme de deux enfants (gain_weight et lose_ weight) pour le nœud (Target_Goal). Nous appliquons le même processus aux feuilles obtenues jusqu'à épuiser toutes les attributs, l'arbre sera comme le suivant :



Les feuilles seraient le composant suggéré à l'utilisateur, mais ce modèle ne répond pas aux besoins de notre système car nous voulons suggérer à l'utilisateur une recette composée de plusieurs composants.

4. L'algorithme apriori et les règle d'association

Imaginant que nous avons trois transactions (utilisateurs) avec leurs caractéristiques et les herbes associées :

- ✓ u1: [Teens, Male, Underweight, gain_weight, Growth Issues] → (Ashwagandha + Fenugreek + Maca)
- ✓ u2: [Teens, Female, Underweight, lose_weight, Growth Issues] → (Guarana + Fenugreek + Green Tea)
- ✓ u3: [Young Adults, Male, Underweight, gain_weight, Growth Issues] → (Green Tea + Garcinia Cambogia + Guarana)

Étape 1 : Formation des motifs fréquents

Pour trouver les motifs fréquents (combinaisons d'herbes qui apparaissent souvent ensemble), nous utilisons un algorithme comme Apriori.

Définir un seuil de support minimal: Disons que nous voulons des itemsets qui apparaissent dans au moins 2 transactions (support $\geq 2/3 \approx 66\%$), La séquence des étapes est illustrée dans la figure suivante :

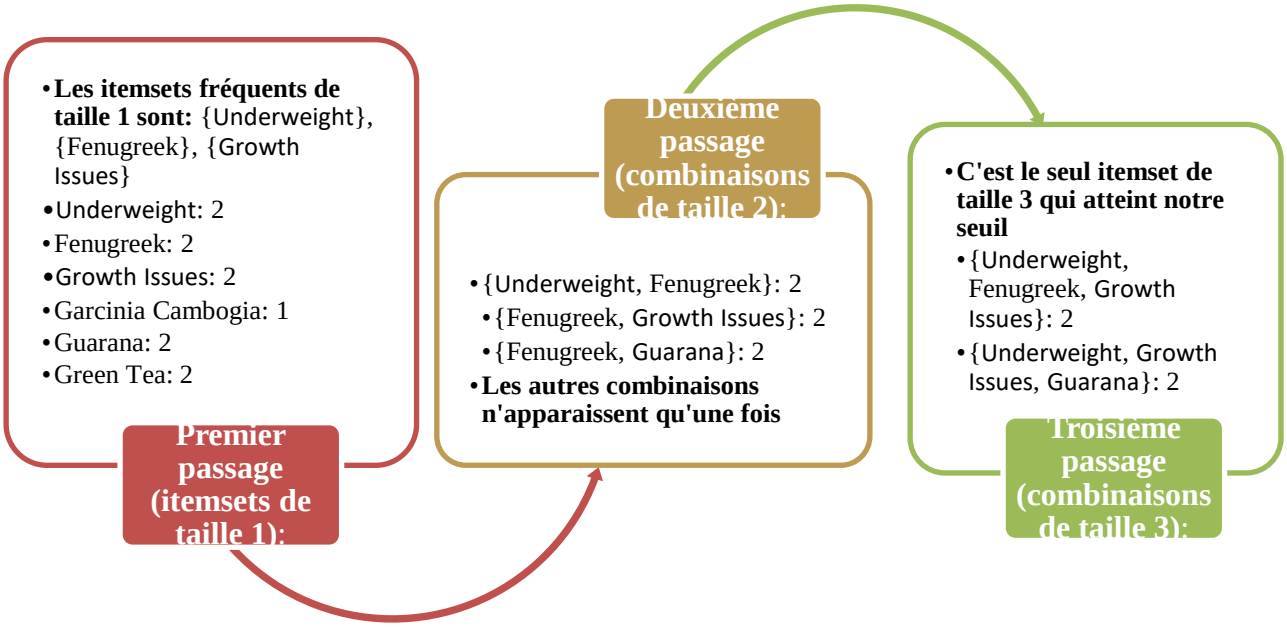


Figure 11. Exemple de génération des motifs fréquents d'herbes

Étape 2 : Génération des règles d'association

Nous voulons extraire des règles de la forme :

[Underweight ou gain_weight ou Growth Issues] → (Herbe)

Voici les règles extraites présentées sous forme de tableau :

Règle	Ingrédient	Support	Règle Extraite et Confiance
Règle 1	Fenugreek	2/3	P(Fenugreek | Underweight + gain_weight + Growth_Issues) = 66.67%
Règle 2	Guarana	2/3	P(Guarana | Underweight + gain_weight + Growth_Issues) = 66.67%
Règle 3	Green Tea	1/3	P(Green Tea | Underweight + lose_weight + Growth_Issues) = 33.33%

Tableau 3. Exemple de génération des règles d'association d'herbes

5. Conclusion

Ce chapitre a permis d'explorer en profondeur les éléments clés d'un système de recommandation personnalisé intégrant la phytothérapie. Nous avons d'abord analysé les ensembles de données pouvant être utilisés pour entraîner le modèle, en mettant en évidence leur structure et leur pertinence pour une approche nutritionnelle adaptée. Ensuite, nous avons détaillé l'architecture du système, en décrivant les étapes essentielles, du prétraitement des données à la mise en œuvre des recommandations. Enfin, nous avons examiné différents modèles de prédiction, comparant leurs performances et leur applicabilité à ce type de problématique. Cette réflexion nous a permis de comprendre les forces et limites des approches existantes et d'ouvrir la voie à des améliorations futures, notamment en matière d'optimisation des recommandations et d'intégration de données plus riches pour une personnalisation encore plus fine.

Chapitre 4 : Implémentation

1. Introduction

Ce chapitre pratique se concentre sur le développement d'un système de recommandation alimentaire, en mettant en avant les techniques et outils utilisés tout au long du processus. L'objectif est de démontrer comment ces technologies peuvent être appliquées pour créer un système efficace capable de fournir des recommandations personnalisées aux utilisateurs.

Nous commencerons par explorer le choix du langage de programmation, en privilégiant Python en raison de sa simplicité et de sa richesse en bibliothèques adaptées à la science des données et à l'apprentissage automatique. Parmi celles-ci, des bibliothèques comme Scikit-learn sont essentielles pour la mise en œuvre des algorithmes de machine learning, permettant ainsi d'analyser et de traiter les données de manière efficace.

Le chapitre détaillera également le processus sur une application démonstrative. Cette application servira de cas d'utilisation concret, illustrant comment les recommandations alimentaires peuvent être générées et présentées aux utilisateurs. Nous expliquerons les différentes étapes de développement, y compris :

Prétraitement des Données : Nous aborderons les méthodes de nettoyage et de préparation des données, cruciales pour garantir la qualité des informations utilisées par le modèle.

- **Codage des Variables :** Le chapitre expliquera comment les variables catégorielles sont codées, ce qui est essentiel pour que le modèle puisse les interpréter correctement.
- **Sélection des Attributs :** Une attention particulière sera portée à la sélection des attributs pertinents, afin d'optimiser les performances du modèle tout en minimisant la complexité.
- **Exécution des Algorithmes :** Nous détaillerons les différentes phases d'exécution des algorithmes de recommandation, en expliquant comment les règles d'association peuvent être appliquées pour générer des recommandations pertinentes.

2. Techniques utilisées

A. Prétraitement de données

Avant d'exécuter un algorithme d'apprentissage automatique ou de fouille des données, tout d'abord, nous devons définir les variables d'entrée et de sortie pour chaque modèle, en tant que notre système est divisé en deux modèles.

```
#  Sélection des colonnes importantes
features = ["Age_Group", "Gender", "Weight_Category", "target_goal", "Health_Problem"]

#  modèle prédictive
taget=["Herbs"] #Foods or Herbs
```

Aussi, il est essentiel de coder les données correctement pour qu'elles soient exploitables par l'algorithme. Voici les principaux types de codage utilisés et leur utilité :

1. **Label Encoding (Encodage de Label)**

- **Principe** : Associe une valeur numérique à chaque catégorie unique d'une variable catégorielle.
- **Pourquoi l'utiliser ?** : Simple et efficace pour les modèles basés sur les arbres, mais peut créer une hiérarchie artificielle si mal utilisé.

🔻 Exemple :

Gender → ["Male", "Female"]

Label Encoding → Male = 0, Female = 1

2. **Ordinal Encoding (Encodage Ordinal)**

- **Principe :** Comme le Label Encoding, mais conserve une relation d'ordre entre les catégories.
- **Pourquoi l'utiliser ?** : Utile quand les valeurs ont une signification ordinale, comme Age_Group ou Weight_Category.

🔻 Exemple :

Weight_Category → ["Underweight", "Normal", "Overweight", "Obese"]

Ordinal Encoding → Underweight = 0, Normal = 1, Overweight = 2, Obese = 3

Meilleur choix si les catégories ont une progression naturelle.

3. **One-Hot Encoding**

- **Principe** : Crée une colonne binaire pour chaque catégorie.
- **Pourquoi l'utiliser ?** : Évite la hiérarchie artificielle de Label Encoding, parfait pour les modèles linéaires et réseaux neuronaux.

🔻 Exemple :

Color → ["Red", "Blue", "Green"]

One-Hot Encoding →

Red	*Blue*	*Green*
1	**0**	**0**
0	**1**	**0**
0	**0**	**1**

B. Environnement de développement

Que ce soit pour le prétraitement de l'ensemble de données, la formation ou l'évaluation des modèles, nous utilisons l'environnement de développement en ligne Kaggle.

Kaggle est une plateforme en ligne dédiée à la science des données et à l'apprentissage automatique. Elle offre une multitude de ressources et d'outils pour les utilisateurs, qu'ils soient débutants ou experts. Voici une description des principales fonctionnalités et ressources fournies par Kaggle :

1. **Fonctionnalités**

- **Compétitions de Science des Données :** Kaggle organise régulièrement des compétitions où les utilisateurs peuvent soumettre leurs modèles pour résoudre des problèmes réels. Ces compétitions offrent des prix et permettent aux participants de se mesurer les uns aux autres tout en apprenant.
- **Datasets :** Kaggle héberge une vaste collection de jeux de données provenant de diverses sources. Les utilisateurs peuvent explorer, télécharger et utiliser ces données pour leurs projets d'apprentissage automatique.
- **Notebooks :** Kaggle propose un environnement de développement intégré appelé "Notebooks" où les utilisateurs peuvent écrire et exécuter du code Python. Ces notebooks

supportent des bibliothèques populaires comme NumPy, pandas, TensorFlow et PyTorch. Ils permettent aussi de partager des analyses et des résultats avec la communauté.

- **Kaggle Kernels :** Les utilisateurs peuvent créer des "Kernels" (notebooks partagés) pour montrer leur travail, collaborer avec d'autres et recevoir des commentaires. Cela favorise l'apprentissage communautaire et l'échange d'idées.
- **Éducation et Tutoriels :** Kaggle propose des cours gratuits sur divers sujets liés à la science des données et à l'apprentissage automatique. Ces cours couvrent des thèmes comme la manipulation de données, l'apprentissage supervisé et non supervisé, et la visualisation des données.
- **Forums et Communauté :** Kaggle dispose de forums où les utilisateurs peuvent poser des questions, partager des conseils et interagir avec d'autres passionnés de science des données. La communauté est active et très utile pour résoudre des problèmes ou échanger des idées.
- **API Kaggle :** Kaggle offre une API qui permet aux utilisateurs de télécharger des jeux de données et de soumettre des prédictions directement depuis leurs environnements de développement locaux.
- **Publications :** Les utilisateurs peuvent publier des articles et des blogs sur des analyses, des projets ou des résultats de compétitions, ce qui contribue à la diffusion des connaissances au sein de la communauté.

2. Ressources de Calcul

Kaggle fournit des environnements de calcul dans ses notebooks, permettant aux utilisateurs d'exécuter du code Python avec accès à des ressources de calcul gratuites. Cela inclut :

- ✓ **CPU :** Les utilisateurs peuvent exécuter leurs notebooks sur des processeurs puissants, adaptés pour des tâches de traitement de données et d'entraînement de modèles.
- ✓ **GPU :** Kaggle offre également l'accès à des unités de traitement graphique (GPU) pour les utilisateurs qui souhaitent exécuter des modèles de deep learning. Cela accélère considérablement l'entraînement des modèles complexes.
- ✓ **TPU :** Pour les utilisateurs de TensorFlow, Kaggle propose des unités de traitement tensoriel (TPU), optimisées pour les opérations d'apprentissage automatique.
- ✓ **Mémoire :** Les notebooks Kaggle disposent d'une mémoire suffisante pour gérer des jeux de données de taille moyenne à grande. Les capacités de mémoire peuvent varier, mais

généralement, elles permettent de charger et de manipuler des datasets allant jusqu'à plusieurs gigaoctets.

- ✓ **Limites :** Bien que Kaggle offre des ressources de calcul puissantes, il existe des limites sur le temps d'exécution des notebooks et la taille des fichiers. Ces limites sont mises en place pour garantir une utilisation équitable des ressources par tous les utilisateurs.

C. Language utilisées

One utilisé le Language Python, il a été créé par Guido van Rossum et publié pour la première fois en 1991, est un langage de programmation **interprété, multiparadigme et multiplateforme**. Il favorise la programmation **impérative structurée, fonctionnelle et orientée objet**. Doté d'un **typage dynamique fort**, il est largement utilisé dans des domaines tels que l'**intelligence artificielle, le traitement des données et le développement web.**[w8]

Python est largement utilisé dans le développement de systèmes de recommandation pour plusieurs raisons :

1. **Bibliothèques Riches** :
    - Python dispose de bibliothèques puissantes comme **Pandas** pour la manipulation de données, **NumPy** pour les calculs numériques, et **Scikit-learn** pour les algorithmes de machine learning. Ces outils facilitent le traitement et l'analyse des données.
2. **Simplicité et Lisibilité** :
    - La syntaxe claire de Python permet de développer rapidement des prototypes et de collaborer efficacement au sein d'équipes. Cela réduit le temps de développement et facilite la maintenance.
3. **Support du Machine Learning** :
    - Python offre des frameworks comme **TensorFlow** et **PyTorch**, qui sont essentiels pour construire des modèles de deep learning, souvent utilisés dans des systèmes de recommandation avancés.
4. **Intégration Facile** :
    - Python s'intègre facilement avec d'autres technologies, bases de données et systèmes, ce qui permet de déployer des modèles de recommandation dans des environnements de production variés.
5. **Communauté Active** :

- La vaste communauté Python fournit une multitude de ressources, de tutoriels et de forums, ce qui facilite le développement et la résolution de problèmes.

6. **Visualisation des Données** :
    - Des bibliothèques comme **Matplotlib** et **Seaborn** permettent de visualiser les données et les résultats, ce qui aide à comprendre le comportement des utilisateurs et l'efficacité des recommandations.

En résumé, Python est un outil idéal pour le développement de systèmes de recommandation grâce à sa flexibilité, sa richesse en bibliothèques, et sa facilité d'utilisation.

D. Bibliothèques utilisées

Ci-dessous, nous parlerons des bibliothèques associées utilisées à la fois dans l'entraînement et l'évaluation du modèle.

a. NumPy

NumPy (Numerical Python) est une bibliothèque fondamentale pour le calcul numérique en Python. Elle fournit des objets de tableaux multidimensionnels efficaces (appelés ndarray) et un ensemble de fonctions mathématiques pour opérer sur ces tableaux. [w9]

- **Alias :** np est une convention courante pour importer NumPy, permettant d'utiliser des fonctions NumPy avec le préfixe np (par exemple, np.array()).

b. Pandas

Pandas est une bibliothèque Python utilisée pour la manipulation et l'analyse de données. Elle offre des structures de données et des opérations pour manipuler des tableaux numériques et des séries temporelles.

```
file_path = "/kaggle/input/userfrequante/userf.csv"
data = pd.read_csv(file_path)
```

- **Alias :** pd est une convention courante pour importer pandas (par exemple, pd.DataFrame()).

c. Scikit-Learn (sklearn)

Scikit-learn est une bibliothèque d'apprentissage automatique en Python. Elle fournit des outils simples et efficaces pour l'analyse prédictive des données, notamment des algorithmes de classification, de régression, de clustering et de réduction de dimensionnalité. [w10]

1. **sklearn.preprocessing :** Le package sklearn.preprocessing fournit des fonctions utilitaires et des classes de transformateur pour modifier les vecteurs de caractéristiques brutes en une représentation plus appropriée pour les estimateurs en aval. Cela inclut la standardisation, la normalisation, la mise à l'échelle et l'encodage des caractéristiques catégorielles.
2. **sklearn.preprocessing.OrdinalEncoder:** OrdinalEncoder encode les caractéristiques catégorielles sous forme de tableau d'entiers. Cette transformation convertit les caractéristiques en entiers ordinaux, résultant en une seule colonne d'entiers (de 0 à n_categories - 1) par caractéristique. Il est utilisé pour encoder des données catégorielles ordinales, où il existe une relation d'ordre entre les catégories.

```
#  Sélection des colonnes importantes
features = ["Age_Group", "Gender", "Weight_Category", "target_goal", "Health_Problem"]
target = "Herbs"

#  Encodage des variables catégoriques
encoders = {}
for col in features:
    encoders[col] = preprocessing.LabelEncoder()
    data[col] = encoders[col].fit_transform(data[col])
```

- **Paramètres importants :**
    - categories : Les catégories (valeurs uniques) par caractéristique.
    - handle_unknown : Indique comment gérer les caractéristiques catégorielles inconnues lors de la transformation.

3. **sklearn.model_selection.train_test_split :** La fonction train_test_split divise les tableaux ou matrices en sous-ensembles d'apprentissage et de test aléatoires. C'est un utilitaire rapide pour valider les entrées et appliquer la division des données en une seule fonction.

```
#  Division en ensemble d'entraînement et de test
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2, random_state=42)
```

- **Paramètres importants :**
    - *arrays : Séquence de tableaux à diviser.
    - test_size : La proportion du jeu de données à inclure dans la division test.

- train_size : La proportion du jeu de données à inclure dans la division d'apprentissage.
- random_state : Graine utilisée par le générateur de nombres aléatoires.
- shuffle : Indique s'il faut mélanger les données avant de les diviser.
- stratify : Si non nul, les données sont divisées de manière stratifiée, en utilisant ce tableau comme étiquettes de classe.

4. **sklearn.neighbors.KNeighborsClassifier:** KNeighborsClassifier est un classificateur qui implémente le vote des k plus proches voisins. Il classe les points de données en fonction des classes de leurs voisins les plus proches dans l'ensemble d'apprentissage.

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
```

- **Paramètres importants :**
  - n_neighbors : Nombre de voisins à utiliser par défaut.
  - weights : Fonction de pondération utilisée dans la prédiction (uniform ou distance).
  - algorithm : Algorithme utilisé pour calculer les plus proches voisins (auto, ball_tree, kd_tree, brute).
  - metric : Métrique de distance à utiliser.

5. **sklearn.metrics.accuracy_score** : La fonction accuracy_score calcule le score de précision de la classification. En classification multi-étiquettes, elle calcule la précision du sous-ensemble, où l'ensemble des étiquettes prédites pour un échantillon doit correspondre exactement à l'ensemble correspondant d'étiquettes dans y_true.

```
y_pred_test = knn.predict(X_test)
test_accuracy = accuracy_score(y_test, y_pred_test)
```

- **Paramètres importants :**
  - y_true : Les vraies étiquettes.
  - y_pred : Les étiquettes prédites.

- normalize : Si True, renvoie la fraction d'échantillons correctement classés ; sinon, renvoie le nombre d'échantillons correctement classés.
- sample_weight : Poids d'échantillon.

d. **Matplotlib.Pyplot**

matplotlib.pyplot est un module de Matplotlib qui fournit une interface de type MATLAB pour tracer des graphiques. Il est couramment utilisé pour la visualisation de données en Python.

Nous avons utilisé le code suivant pour évaluer les performances de l'algorithme du voisin le plus proche et créer la figure : ci-dessus

```
#  Plot accuracy trends
plt.figure(figsize=(8, 5))
plt.plot(k_values, train_accuracies, marker='o',
         label="Train Accuracy", linestyle='-')
plt.plot(k_values, test_accuracies, marker='s',
         label="Test Accuracy", linestyle='--')

#  Customize the plot
plt.xlabel("K-Value")
plt.ylabel("Accuracy")
plt.title("KNN: Train vs Test Accuracy")
plt.legend()
plt.grid(True)
plt.show()
```

e. **MLxtend**

MLxtend (machine learning extensions) est une bibliothèque Python d'outils utiles pour les tâches quotidiennes de science des données Elle fournit des algorithmes et des utilitaires pour l'apprentissage automatique et l'exploration de données.

1. **mlxtend.frequent_patterns.apriori:** apriori est une fonction de la bibliothèque MLxtend qui implémente l'algorithme Apriori pour l'extraction d'ensembles d'éléments fréquents à partir de données transactionnelles.

```
#  Création des transactions (One-Hot Encoding)
encoded_data = pd.get_dummies(new_data)

#  Application de l'algorithme Apriori
frequent_itemsets = apriori(encoded_data, min_support=0.01, use_colnames=True)
```

- **Paramètres importants :**
    - df : DataFrame pandas des données transactionnelles.
    - min_support : Seuil minimal de support pour les ensembles d'éléments.
    - use_colnames : Indique s'il faut utiliser les noms de colonnes au lieu des indices.

2. **mlxtend.frequent_patterns.association_rules:** association_rules est une fonction de la bibliothèque MLxtend qui génère des règles d'association à partir d'ensembles d'éléments fréquents. Elle permet d'identifier les relations entre différents éléments dans un ensemble de données.

```
#  Extraction des règles d'association
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=0.1)
```

- **Paramètres importants :**
    - df : DataFrame pandas des ensembles d'éléments fréquents.
    - metric : Métrique pour évaluer si une règle est intéressante (par exemple, confidence, lift).
    - min_threshold : Seuil minimal pour la métrique d'évaluation.

3. Présentation du système

Dans la section précédente de la recherche, nous avons développé l'architecture du système de recommandation et examiné les performances des algorithmes qui pourraient être mis en œuvre, ainsi essayé des exemples expérimentaux de la procédure de prédiction dans chacun de ces modèles. D'après ce qui précède, il nous apparaît clairement que le modèle de règles d'association sera le plus adapté à notre système actuel et qu'il est cohérent avec l'ensemble de données disponibles. En général, les règles d'association sont largement utilisées dans les

systèmes de recommandation en raison de leur philosophie de base consistant à sélectionner des motifs en fonction de la fréquence de co-occurrence des éléments.

Il convient de noter que l'ensemble des règles d'association extraites par l'algorithme apriori constitue un ensemble de règles brutes qui doivent être soumises à un filtre. Ce filtre permet de supprimer toutes les règles qui ne contiennent pas d'antécédent valide correspondant à l'ensemble des variables marquées comme variables d'entrée. Inversement, toutes les variables qui n'appartiennent pas aux variables cibles doivent également être éliminées. Dans notre cas, nous avons convenu de nous concentrer uniquement sur le maintien des règles qui contiennent trois variables ou plus en entrée.

```
#  Filtrer les règles où tous les antécédents appartiennent à valid_antecedents
valid_antecedents = {"Age_Group", "Gender", "Weight_Category", "target_goal", "Health_Problem"}
valid_consequents = {"Foods", "Herbs"}

#  Filtrer les règles où tous les antécédents appartiennent à valid_antecedents
filtered_rules = rules[
    rules['consequents'].apply(lambda x: taget[0] in str(x)) &
    rules['antecedents'].apply(lambda x: all(not any(feature in i for i in x)
                                        for feature in valid_consequents)  and len(x) > 2)
]
```

Voici un échantillon des règles d'association extraites par l'algorithme de la bibliothèque **mlxtend** sur **python :** voir l'annexe ci-dessous

- Si frozenset({'Age_Group_Early Adulthood', 'Gender_Female', 'Weight_Category_Overweight'}) → Alors frozenset({'Herbs_Dandelion + Orthosiphon + Cinnamon'}) (Support: 0.0125, Confiance: 0.125, Lift: 1.0)
- Si frozenset({'Age_Group_Early Adulthood', 'Gender_Female', 'Weight_Category_Overweight'}) → Alors frozenset({'Herbs_Green Tea + Garcinia Cambogia + Dandelion'}) (Support: 0.0125, Confiance: 0.125, Lift: 1.0)

La recommandation avec le type de règles ci-dessus en temps réel sur un site Web, ou sur application phone peut être effectuée en chargeant l'ensemble des règles (l'antécédent et la conséquence ainsi que les critères de performance tels que le support et la confidence) dans une table de base de données et en effectuant la correspondance à l'aide d'une requête Sql lorsqu'une tentative se produit.

Afin de démontrer un cas d'utilisation réel du système de recommandation développé, nous avons créé une application de démonstration avec une interface utilisateur pour saisir des données d'entrée (données personnelles) et récupérer des résultats (recettes personnalisées), ou gérer les règles d'association utilisées, comme suit :

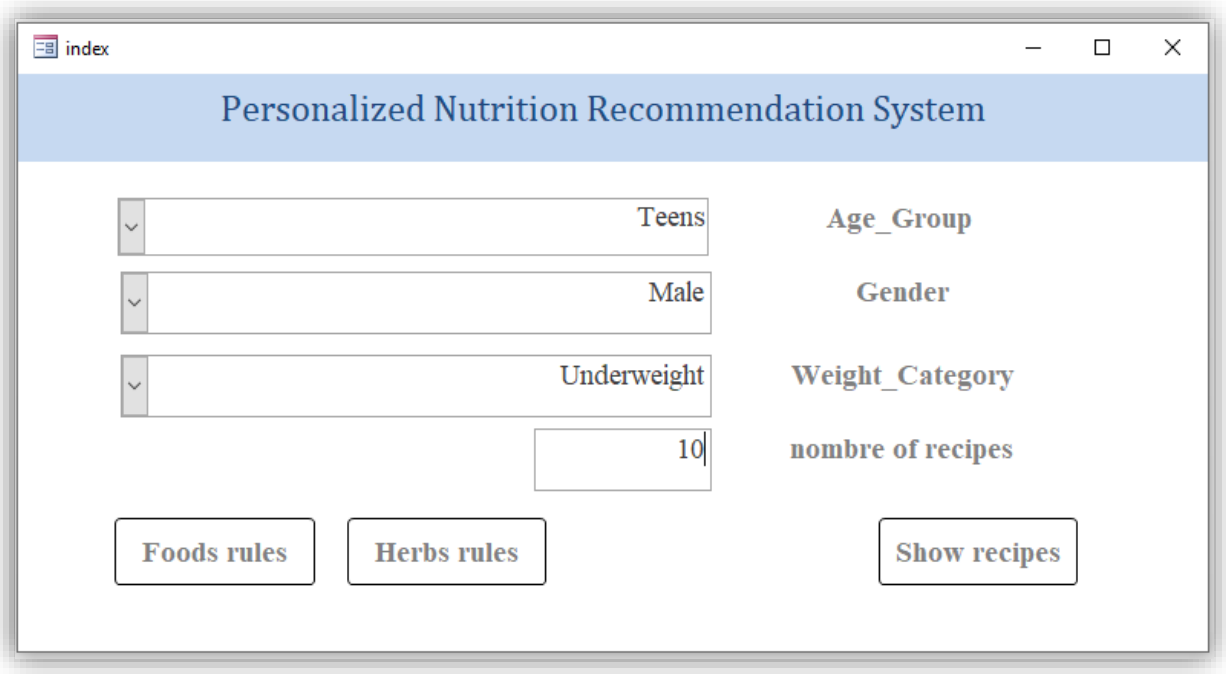


Figure 12. La fenêtre principale de l'application démonstratif

Tout d'abord, cette fenêtre permet à l'utilisateur de prévisualiser les règles d'association chargées, qu'elles concernent l'inférence de recommandations alimentaires ou herbels, leur suppression ou ajout peut se faire manuel. Nous avons utilisé le langage SQL et la programmation procédurale en Visual Basic.

Figure 13. La fenêtre pour gérer les règles de recommandions

Aussi cette fenêtre, permet à l'utilisateur de saisir des informations sur l'âge, le poids et le sexe, qui sont nécessaires au modèle après l'entraînement. Dans ce cas, l'objectif n'est pas spécifié ; le système le déduira de l'entraînement précédent. Il est également possible d'indiquer le nombre de recettes requises, car le système, en fonction de la présence de plusieurs règles d'association, affichera un ensemble de recommandations. La plupart de ces recommandations seront uniques, mais certaines recettes peuvent présenter une légère répétition, car les règles d'association incluent d'autres règles définies précédemment. Voici les résultats :

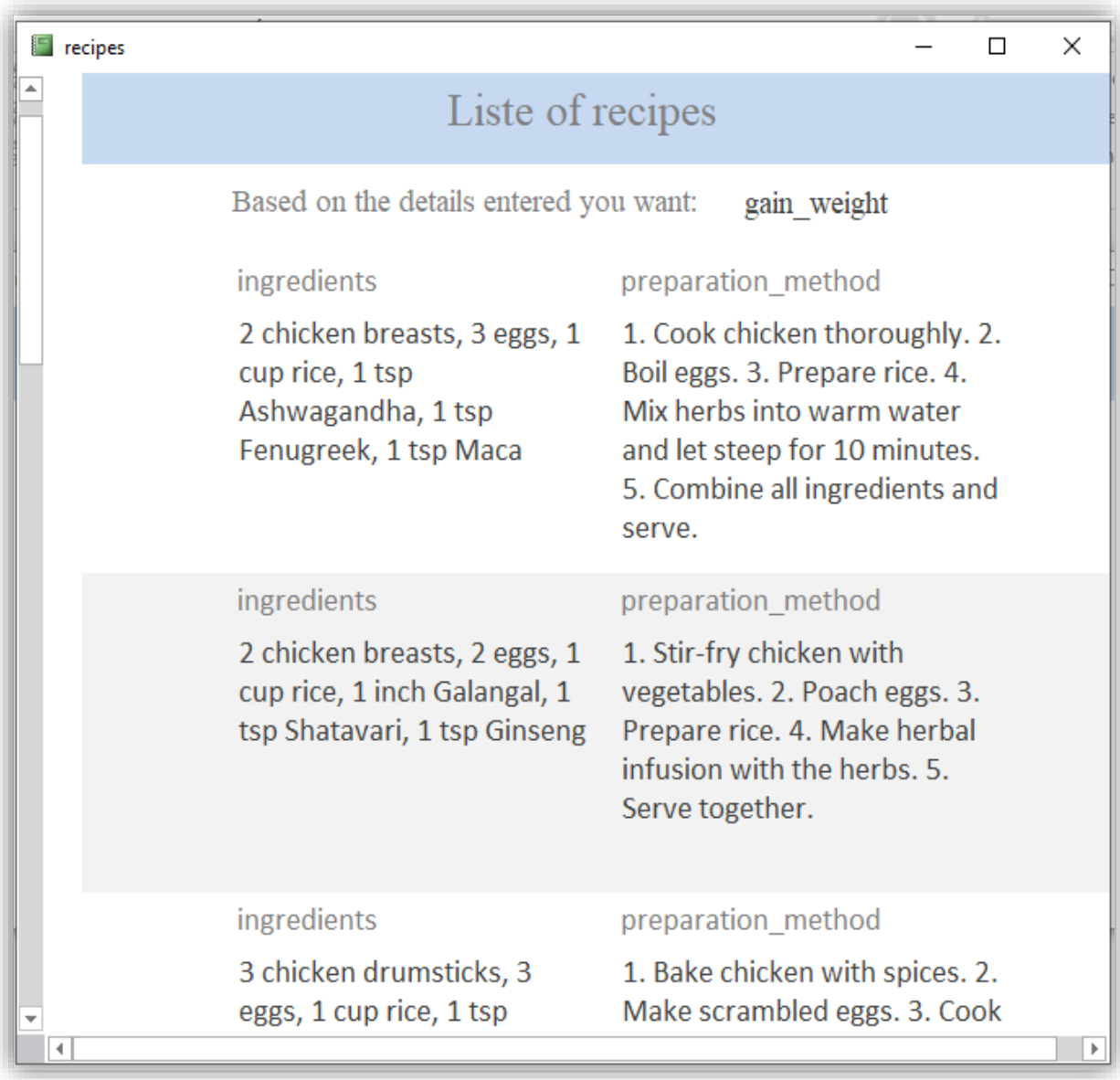


Figure 14. La forme de liste recommandation

L'utilisateur est libre de choisir la recette qui lui convient ou d'exclure certains ingrédients.

4. Discussion

Le travailler avec des règles d'association semble plus intuitif et peut être facilement expliqué aux humains, L'extraction des règles d'association peut être facilement convertie en des instructions conditionnels (Si... Alors... Sinon). Pour notre système de recommandation, si nous voulons suggérer plus d'une recette à l'utilisateur, nous pouvons le faire en choisissant les règles d'association avec le moins de précision en termes de confiance. Si nous voulons suggérer plus d'une recette, nous pouvons le faire en choisissant les règles d'association avec la valeur la plus élevée en termes de confiance.

La flexibilité de travailler avec ce type de modèle est démontrée par le fait que vous pouvez modifier ou mettre à jour les règles d'association sans arrêter le système. Cela signifie que vous pouvez supprimer certaines règles, en ajouter d'autres et gérer la fouille pour des autres en parallèle du travail du système de recommandation sans le redémarrer.

L'évaluation des règles d'association dépend la valeur de confidence spécifiée lors de l'extraction des règles dans la deuxième étape et du support minimum lors de la fouille des motifs fréquents dans la première étape. Notez que nous avons utilisé une valeur faible pour le support minimum en raison du petit ensemble de données. Ce modèle nécessite une grande quantité de données pour rendre les règles plus robustes, mais cela donnera un coup de pouce de départ à la recommandation. Lorsque nous disposons d’une partie significative des transactions, nous refaire la fouille avec des conditions idéales.

Les faiblesses de ce type de modèle sont qu'il repose sur la répétition de certains éléments, même s'ils sont efficaces dans le problème de prise ou de perte de poids, et qu'ils peuvent ne pas apparaître dans les règles d'association car ils sont consommés par un petit groupe d'utilisateurs. De plus, le facteur temps joue un rôle dans la limitation de l'apparition de composants nutritionnels ou d'herbes, ce qui n'est pas pris en compte.

5. Conclusion

Ce chapitre a permis d’explorer en profondeur le développement d’un système de recommandation alimentaire, en soulignant l’importance des techniques et outils utilisés tout au long du processus. Grâce à Python et à ses bibliothèques, telles que Scikit-learn, nous avons pu mettre en œuvre des algorithmes performants pour analyser les données et générer des recommandations pertinentes. L'application démonstrative que nous avons créée illustre concrètement comment ces techniques peuvent être appliquées pour répondre aux besoins des utilisateurs en matière de recommandations alimentaires.

En examinant les différentes étapes, du prétraitement des données à la sélection des attributs, nous avons mis en lumière l'importance de chaque phase dans le développement d'un modèle efficace. Cela démontre que la qualité des données et la rigueur méthodologique sont essentielles pour garantir des résultats fiables. Ce travail ouvre la voie à des explorations futures, où des améliorations et des ajustements pourraient être réalisés pour affiner encore davantage les recommandations et enrichir l'expérience utilisateur.

Conclusion et Perspectives

Ce mémoire porte sur la conception d'un système de recommandation personnalisé intégrant la nutrition et la phytothérapie, dans une optique de santé préventive et de bien-être individualisé. Dans un contexte où la personnalisation des conseils alimentaires devient une exigence croissante, ce travail ambitionne de combiner les apports des aliments classiques aux vertus thérapeutiques des plantes médicinales, particulièrement adaptées aux modes de vie méditerranéens. L'étude s'inscrit dans un double objectif : proposer des recommandations nutritionnelles adaptées au profil de l'utilisateur, tout en valorisant l'usage raisonné des plantes dans l'accompagnement de certains objectifs comme la perte ou la prise de poids.

La réalisation de ce travail a été marquée par plusieurs difficultés, principalement liées à la rareté de bases de données structurées en phytothérapie, à la diversité non standardisée des effets des plantes et à la complexité d'intégrer des profils individuels variés. Malgré ces obstacles, l'avancement des techniques en intelligence artificielle et en science des données a permis de proposer une solution fonctionnelle. Le développement de l'application, basé sur l'algorithme Apriori pour l'extraction de règles d'association, a permis de générer automatiquement des suggestions sous forme de recettes personnalisées, prenant en compte les caractéristiques de l'utilisateur telles que l'âge, le poids, les objectifs, les éventuelles restrictions ou allergies. Ces recommandations combinent aliments et plantes médicinales avec indication des quantités et modes de préparation, dans une interface conviviale et adaptable.

Ce projet a été l'occasion de renforcer mes compétences en programmation, en fouille de motifs fréquents, en conception de systèmes de recommandation, et de me confronter à des enjeux concrets liés à la structuration de données hétérogènes. Les résultats obtenus sont encourageants, avec un système flexible, capable de s'adapter à différentes contraintes utilisateurs. Cependant, certaines limites subsistent, notamment l'absence de standardisation des effets des plantes, le manque de données validées sur leurs interactions avec les profils de santé, et l'absence de prise en compte du facteur temporel, comme la saisonnalité ou la durée de consommation des plantes.

Partant de ces limites, plusieurs pistes d'amélioration ont été identifiées. Il serait pertinent de constituer une base de connaissances unifiée regroupant des sources scientifiques, traditionnelles et empiriques validées, permettant une modélisation plus fiable des effets phytothérapeutiques. L'intégration d'approches hybrides, combinant règles d'association et logique experte à l'aide d'ontologies, permettrait une meilleure prise en compte du contexte médical individuel. De plus, l'ajout d'une dimension temporelle dans les modèles, à travers des techniques d'analyse de

séquences comme les réseaux LSTM, permettrait d’adapter les recommandations aux effets différés ou saisonniers de certaines plantes.

Enfin, ce travail ouvre la voie à de futures recherches dans le domaine de la médecine personnalisée appuyée par l’intelligence artificielle, en envisageant par exemple l'intégration de données en temps réel issues de capteurs physiologiques, ou encore la validation clinique des recommandations générées par le système. Ces perspectives visent à enrichir, sans le dépasser, le travail accompli, tout en mettant en lumière son potentiel d’évolution dans des applications concrètes de santé préventive et nutritionnelle

1. Références Bibliographiques

[1] Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook*. Springer.

[2] Adomavicius, G., & Tuzhilin, A. (2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Future Directions." *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.

[3] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331-370.

[4] Negre, E. (2015). Recommender systems and knowledge representation. *European Journal of Information Systems*, 24(2), 125-147.

[5] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.

[6] **Herlocker, J., Konstan, J., Terveen, L., & Riedl, J. (2004).** Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5-53

[7] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

[8] Domingos, P. (2012). "A Few Useful Things to Know About Machine Learning". Communications of the ACM, 55(10), 78–87.

[9] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[10] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[11] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[12] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.4. Algorithmes Courants en Machine Learning

[13] Breiman, L., et al. (1984). Classification and Regression Trees. Wadsworth.

[14] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

[15] **James, G. et al. (2021)**. *An Introduction to Statistical Learning*. Springer.

[16] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast discovery of association rules. Advances in knowledge discovery and data mining, 12(1).

2. Références Web (Techniques)

[w1] Qu'est-ce qu'un mot,eur de recommandation : une catégorisation (consulté le 13/05/2025), https://www.ibm.com/fr-fr/think/topics/recommendation-engine

[w2] Les systèmes de recommandation : une catégorisation (consulté le 12/05/2025), https://interstices.info/les-systemes-de-recommandation-categorisation/

[w3] Steps to Build a Machine Learning Model (consulté le 20/05/2025), https://www.geeksforgeeks.org/steps-to-build-a-machine-learning-model/

[w4] Kaggle nutirition data set (25/06/2025) https://www.kaggle.com/datasets?tags=4306-Nutrition

[w5] Machine Learning,(consulté le 10/04/2025), https://www.w3schools.com/python/python_ml_getting_started.asp

[w6] Machine Learning Algorithms, (consulté le 12/05/2025), https://www.geeksforgeeks.org/machine-learning-algorithms/

[w7] Data Mining Techniques, (consulté le 14/05/2025), https://www.geeksforgeeks.org/data-mining-techniques/

[w8] Python uses and Applications, (consulté le 02/05/2025), https://www.geeksforgeeks.org/what-is-python/

[w9] NumPy Introduction, (consulté le 03/05/2025), https://www.geeksforgeeks.org/introduction-to-numpy/

[w10] User Guide sklearn, (consulté le 04/05/2025), https://scikit-learn.org/stable/modules/preprocessing.html

Annexe : les règles d'association générée

A. Pour prendre du poids

Antécédents	Conséquences	Support	Confiance	Lift
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Herbs_Ashwagandha + Fenugreek + Maca	0.0417	0.25	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Herbs_Astragalus + Turmeric + Burdock	0.0417	0.25	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Herbs_Galangal + Shatavari + Ginseng	0.0417	0.25	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Herbs_Rhodiola + Shilajit + Spirulina	0.0417	0.25	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Herbs_Ashwagandha + Fenugreek + Maca	0.0208	0.25	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Herbs_Astragalus + Turmeric + Burdock	0.0208	0.25	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Herbs_Galangal + Shatavari + Ginseng	0.0208	0.25	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Herbs_Rhodiola + Shilajit + Spirulina	0.0208	0.25	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Foods_Chicken + Eggs + Rice	0.0278	0.1667	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Foods_Legumes + Whole Grains + Nuts	0.0278	0.1667	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Foods_Nuts + Dark Chocolate + Honey	0.0278	0.1667	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Foods_Potatoes + Cheese + Seeds	0.0278	0.1667	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Foods_Salmon + Avocado + Olive Oil	0.0278	0.1667	1.0
Weight_Category_Underweight, Age_Group_Early Adulthood, Gender_Male	Foods_Whole Milk + Smoothies + Dried Fruits	0.0278	0.1667	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Foods_Chicken + Eggs + Rice	0.0139	0.1667	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Foods_Legumes + Whole Grains + Nuts	0.0139	0.1667	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Foods_Nuts + Dark Chocolate + Honey	0.0139	0.1667	1.0

Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Foods_Potatoes + Cheese + Seeds	0.0139	0.1667	1.0
Health_Problem_Bone Density Loss, Age_Group_Early Adulthood, Gender_Male	Foods_Salmon + Avocado + Olive Oil	0.0139	0.1667	1.0

B. Pour perdre du poids

Antécédents	Conséquences	Support	Confiance	Lift
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Dandelion + Orthosiphon + Cinnamon	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Garcinia Cambogia + Green Coffee + Cinnamon	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Green Tea + Garcinia Cambogia + Dandelion	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Green Tea + Milk Thistle + Yerba Mate	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Guarana + Fucus Vesiculosus + Artichoke	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Guarana + Yerba Mate + Green Coffee	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Milk Thistle + Artichoke + Fucus Vesiculosus	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Herbs_Yerba Mate + Dandelion + Orthosiphon	0.0125	0.125	1.0
Age_Group_Late Midlife, Weight_Category_Overweight, Gender_Female	Herbs_Dandelion + Orthosiphon + Cinnamon	0.0125	0.125	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Moderately_Obese	Foods_Avocado + Greek Yogurt + Cucumber	0.01	0.1667	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Moderately_Obese	Foods_Broccoli + Mushrooms + Watermelon	0.01	0.1667	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Moderately_Obese	Foods_Eggs + Almonds + Chia Seeds	0.01	0.1667	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Moderately_Obese	Foods_Salmon + Green Tea + Berries	0.01	0.1667	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Moderately_Obese	Foods_Spinach + Chicken Breast + Quinoa	0.01	0.1667	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Moderately_Obese	Foods_Sweet Potatoes + Apple Cider Vinegar + Dark Chocolate	0.01	0.1667	1.0
Age_Group_Early Adulthood,	Foods_Avocado + Greek	0.0167	0.1667	1.0

Gender_Female, Weight_Category_Overweight	Yogurt + Cucumber			
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Foods_Broccoli + Mushrooms + Watermelon	0.0167	0.1667	1.0
Age_Group_Early Adulthood, Gender_Female, Weight_Category_Overweight	Foods_Eggs + Almonds + Chia Seeds	0.0167	0.1667	1.0

Annexe : Description des valeurs catégorielles d'ensemble de données

A. Le poids

Catégorie	Plage de poids	Implications pour la santé
Category	40–49 kg	Risque de carences nutritionnelles, fatigue, affaiblissement du système immunitaire.
Underweight	50–64 kg	Poids optimal pour la plupart des tailles (IMC ~18,5–24,9), risque minimal de maladies.
Healthy Weight	65–74 kg	Risque modéré de troubles métaboliques (si sédentaire) : hypertension, pré-diabète.
Overweight	75–80 kg	Risque élevé de diabète/maladies cardiaques (si IMC ≥30), pression articulaire accrue.

B. L'âge

Catégorie	Tranche d'âge	Caractéristiques clés
Teens	15–19	Phase de croissance, besoins nutritionnels élevés.
Young Adults	20–29	Pic métabolique, formation des habitudes de vie (alimentation, exercice).
Early Adulthood	30–39	Période de stabilité, début des changements métaboliques (ralentissement).
Midlife	40–49	Début de perte musculaire (sarcopénie), changements hormonaux (préménopause, etc.).
Late Midlife	50–59	Risques accrus de maladies chroniques (diabète, hypertension).
Senior Adults	60–70	Baisse d’absorption des nutriments, priorité à la mobilité et santé osseuse.

Résumé

Cette étude vise à développer un système de recommandation nutritionnelle personnalisé intégrant la phytothérapie pour les utilisateurs, en fonction de leur souhait de prendre ou de perdre du poids. Nous avons créé un jeu de données spécialement à cet effet et comparé un ensemble de modèles d'apprentissage automatique et de fouille de données applicables à ce problème. Les règles d'association ont démontré des performances généralement acceptables du système.

Mots clés : apprentissage automatique, fouille de données, nutrition, phytothérapie, système recommandation.

Abstract

This study aims to develop a personalized nutritional recommendation system integrating herbal medicine for users based on their desire to gain or lose weight. We created a dataset specifically for this purpose and compared a set of machine learning and data mining models applicable to this problem. The association rules demonstrated generally acceptable system performance.

Keywords: machine learning, data mining, nutrition, herbal medicine, recommendation system.

ملخّص

تهدف هذه الدراسة إلى تطوير نظام توصيات غذائية مُخصص يدمج الأدوية العشبية، موجه للمستخدمين الذين يرغبون في زيادة الوزن أو إنقاصه. أنشأنا مجموعة بيانات خصيصًا لهذا الغرض، وقارنا مجموعة من نماذج التعلم الآلي والتنقيب عن البيانات المُطبقة على هذه المشكلة. أظهرت قواعد الارتباط أداءً مقبولًا للنظام بشكل عام.

الكلمات المفتاحية: التعلم الآلي، استخراج البيانات، التغذية، الأدوية العشبية، نظام التوصيات.