

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي



Mémoire de Master en informatique

Spécialité : *l'intelligente Artificielle*

Thème

AI Beauty Skin Advisor

■ Présenté par :

- Deghbadj Hibat El Rahmen Fatiha .
- Atig Chaimaa .

■ Dirigé par :

- Rahmani Mohamed El Hadi .

Année universitaire



2025-2026

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

Avant tout, nous remercions Dieu le Tout-Puissant de nous avoir accordé la santé, la patience et la volonté nécessaires pour accomplir ce travail .

Nous tenons à exprimer notre profonde reconnaissance et notre sincère gratitude à notre encadrant, **Monsieur Rahmani Mohamed El Hadi**, pour la qualité de son encadrement, ses conseils précieux, sa disponibilité constante ainsi que pour l'intérêt qu'il a porté à ce travail tout au long de sa réalisation.

Nous adressons également nos plus vifs remerciements à nos chers parents, pour leur soutien inconditionnel, leurs encouragements permanents et leurs prières, qui ont été une source de motivation tout au long de notre parcours.

Nous exprimons notre respectueuse gratitude aux membres du jury pour l'honneur qu'ils nous font en acceptant d'évaluer ce travail, ainsi que pour leurs remarques pertinentes et constructives.

Enfin, nous tenons à remercier toutes les personnes qui, de près ou de loin, ont contribué à la réalisation et à l'aboutissement de ce mémoire.

Dédicace

Je remercie Dieu Tout-Puissant de m'avoir donné la force, la patience et la volonté nécessaires pour mener à bien ce travail et atteindre cette étape importante de ma vie.

*Je dédie ce modeste travail à l'âme pure de mon père, feu **Mohamed**, qui aurait tant souhaité être présent pour partager avec moi ce moment de fierté. Ton absence est une douleur silencieuse, mais ta présence reste vivante dans mon cœur et dans chacun de mes pas. J'aurais tant aimé te voir sourire aujourd'hui. Je te promets, mon cher père, de continuer à avancer avec courage, dignité et détermination, en restant fidèle aux valeurs que tu m'as transmises. Qu'Allah t'accorde Sa miséricorde infinie, illumine ta tombe et t'accueille dans Son vaste paradis.*

*À ma chère **mère**, symbole de tendresse, de sacrifice et de patience, qui m'a entouré d'un amour inconditionnel et n'a jamais cessé de me soutenir. Que Dieu la protège et lui accorde une longue vie pleine de santé et de bonheur.*

*À mes chers frères et sœurs : **Kheira, Halima, Ilyas, Mhamed abd samed, Ishaq, Abdelillah, Mohamed, haithem abd el djalil** et **Mawlid Al-Saghir**, pour leur amour, leur soutien et les moments de joie partagés, qui ont illuminé mon parcours.*

À toute ma famille, ainsi qu'à toutes les personnes qui ont contribué de près ou de loin à ma réussite.

Deghbadj Hibat El rahmen Fatiha

dédicace

Avant toute chose, à mes très chers parents, pour leur amour inconditionnel, leur soutien indéfectible et tous les sacrifices qu'ils ont consentis afin de me permettre d'atteindre ce niveau. Que Dieu les protège et les garde.

*À ma sœur adorée, **Fatima**, pour son affection et son encouragement permanent.*

*À mes très chers frères, **Abdelkarim, Ahmed et Abdelhak**, pour leur soutien et leur présence à mes côtés.*

À toute ma famille et mes proches, pour leur amour et leurs encouragements.

À tous mes amis(es), qui m'ont soutenue et encouragée tout au long de cette année.

*À ma chère camarade **Hiba**, pour son aide précieuse et sa présence.*

Enfin, à toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

À tous ceux qui me sont chers.

Atig Chaimaa

Sommaire :

Remerciements	
Dédicace.....	
Liste Des Figure	
Liste des Tableaux	
introduction gènèrale.....	11
Chapitre 1 : Les systèmes de recommandation (Beauty & Skin care)	
Introduction	14
1.1 Définition d'un système de recommandation.....	14
1.2 Importance dans le domaine de la beauté.....	15
1.2.1 Les systèmes de recommandation permettent	15
1.3 Approches utilisées pour la recommandation	15
1.3.1 Le filtrage collaboratif (Collaborative Filtering).....	15
1.3.2 Filtrage basé sur le contenu (Content-Based Filtering).....	16
1.3.3 Filtrage hybride (Hybrid Filtering).....	17
1.4 Technologies utilisées.....	18
1.4.1 Machine Learning (ML) :	18
1.4.2 Natural Language Processing (NLP) :	19
1.5 Applications des systèmes de recommandation dans le domaine de la beauté :	19
1.5.1 Personnalisation des produits skincare :	19
1.5.2 Recommandation de maquillage :	19
1.5.3 Exemples réels :	19
1.6 Défis des systèmes de recommandation dans la beauté:.....	20
1.6.1 Problème de données :	20
1.6.2 Personnalisation complexe :	20
1.6.3 Évolution des besoins :	20
1.7 Importance des systèmes de recommandation :	20
Conclusion	21
Chapitre 2 : L'Intelligence Artificielle Générative	
Introduction.....	23
2.1 Définition de l'Intelligence Artificielle Générative.....	24
2.2 Différence entre l'IA classique et l'IA générative	24
2.3artificielle au service du diagnostic de peau.....	24
2.4 Les avantages de l'intelligence artificielle dans le diagnostic de peau.....	25
2.5 Limites et défis de l'Intelligence Artificielle Générative.....	26
2.6 Domaine de la beauté et du skincare.....	26
Conclusion.....	27
Chapitre 3: Conception (diagramme de votre application)	
Introduction.....	29
L'utilisation de diagramme.....	29
1.1 Définition d'un diagramme :	29
1.2 Caractéristiques des diagrammes UML :	29
1.3 Concepts d'UML :	29

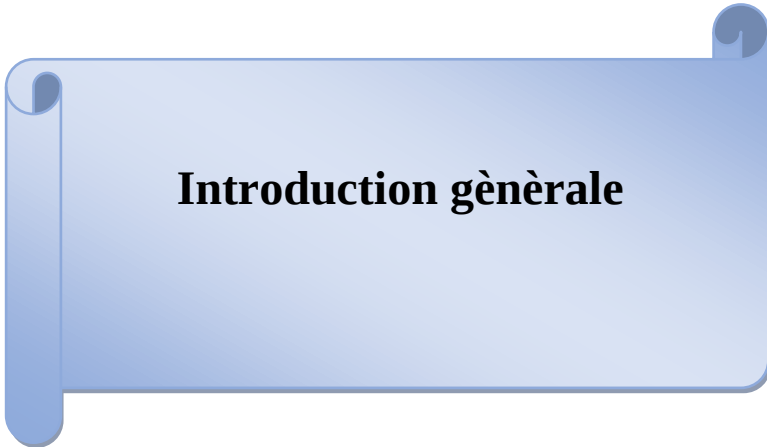
2 Architecture générale du système	29
2.1 Vue d'ensemble	29
2.2 Description des modules.....	30
2.2.1 Module Recommandation	30
2.2.2 Module Analyse IA.....	30
2.2.3 Module Chatbot.....	30
2.3 Schéma d'architecture.....	30
3 Diagrammes UML.....	30
3.1 Diagramme de cas d'utilisation.....	30
3.1.1 Acteurs identifiés.....	30
3.1.2 Cas d'utilisation.....	31
3.1.3 Schéma du diagramme.....	31
3.2 Diagramme de séquence.....	31
3.2.1 Scénario 1 : Recommandation.....	31
3.2.2 Scénario 2 : Analyse de peau.....	32
3.2.3 Scénario 3 : Chatbot.....	33
3.3 Diagramme de classes.....	34
3.3.1 Classes principales.....	34
3.3.2 Attributs et méthodes (exemple).....	34
3.4 Diagramme d'activité.....	35
3.4.1 Processus complet.....	35
4 Modélisation de la base de données.....	35
4.1 Dictionnaire de données.....	36
4.1.1 Table Utilisateurs.....	36
4.1.2 Table Produits.....	36
4.1.3 Table Messages.....	37
4.1.4 Table Analyses.....	37
4.2 Schéma relationnel (MCD/MLD).....	37
5 Conception des interfaces utilisateur.....	37.
5.1 Interface de recommandation.....	37
5.2 Interface d'analyse IA.....	37
5.3 Interface chatbot.....	37
5.4 Maquette globale.....	37
Conclusion.....	38

Chapitre 4: Implémentation (contient les outils que vous avez utilisé, le prompt que vous avez utilisé pour entrainer chatgpt avec explication comment avez vous fait, et les captures d'ecran de votre app)

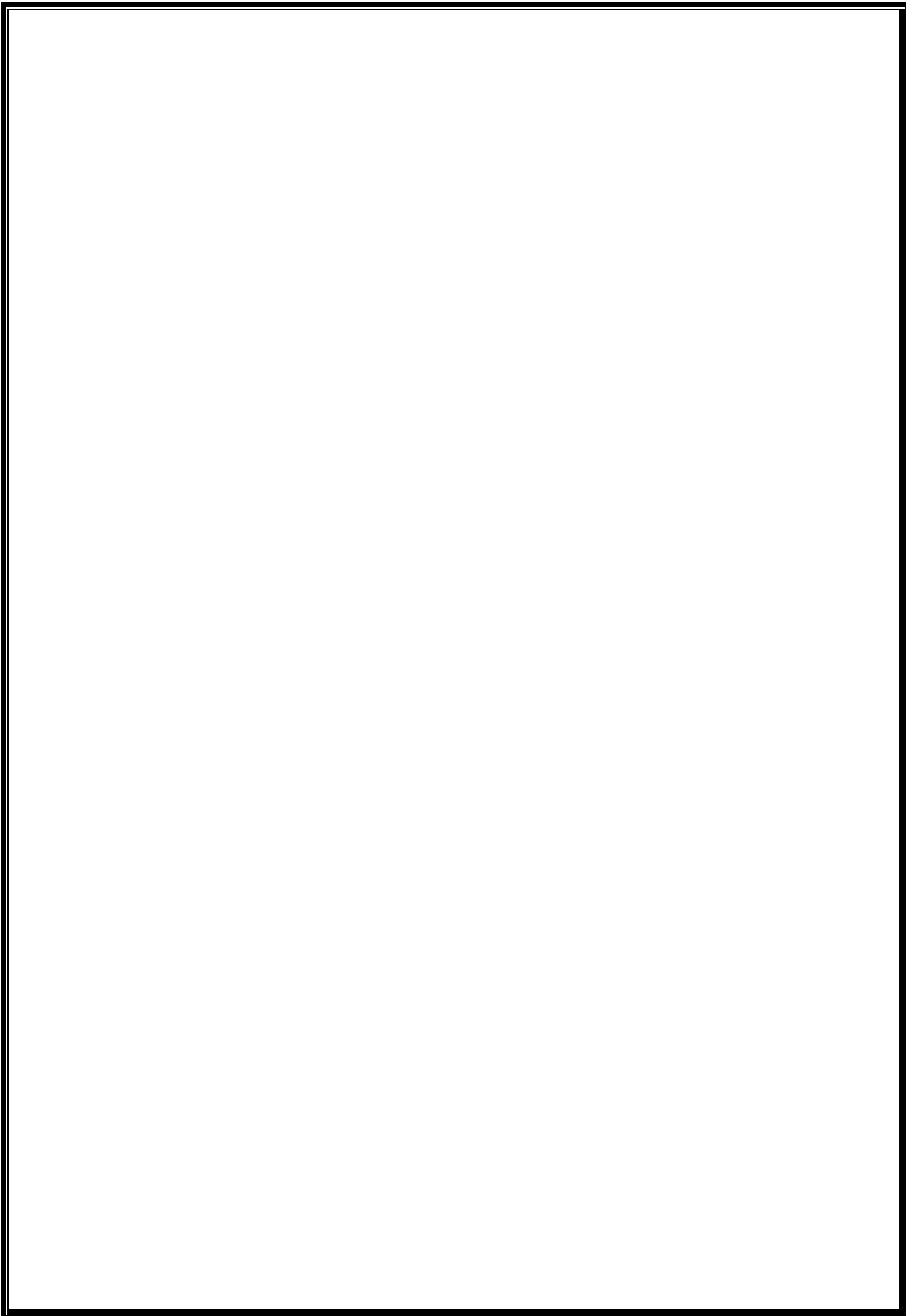
Introduction	40
1 Environnement de développement :	40
1.1 Le langage de programmation : Python.....	40
1.2 Le framework d'interface : Streamlit.....	41
1.3 La bibliothèque de gestion des données : Pandas.....	42
1.4 La bibliothèque de calcul numérique : NumPy.....	43
1.5 Le framework d'intelligence artificielle : TensorFlow / Keras.....	44

1.6 La bibliothèque de prétraitement : Pickle.....	44
1.7 Le module de réponses aléatoires : Random.....	45
1.8 Bibliothèque de prétraitement : Scikit-learn (LabelEncoder).....	45
1.9 Éditeur de code : Visual Studio Code (VS Code).....	46
1.10 Gestionnaire de versions : flutter.....	46
1. Tableau récapitulatif des outils (4.2) :.....	46
2 Structure du projet :.....	47
3 Les fichiers de données : CSV.....	47
Tableau 4.X : Dataset des utilisateurs (users.csv) :.....	48
Tableau 4.1 : Informations générales des utilisateurs :.....	48
Tableau 4.2 : Indicateurs des problèmes de peau :.....	49
3. Dataset des produits (products.csv):.....	49
4. Dataset des interactions (interactions.csv):.....	50
5.Dataset des solutions (Skin Analysis):.....	50
6 Architecture de l'application :.....	51
6.1 Interface utilisateur (Frontend) :.....	51
6.2 Logique de traitement (Backend).....	52
6.3 Données et modèle d'intelligence artificielle.....	53
6.4 Fonctionnement global de l'application.....	53
7 Organisation du projet et des fichiers :.....	53
7.1 Fonctionnement global :.....	54
7.2 Étape 1 : Saisie de l'utilisateur (User Input) :.....	54
7.3 Étape 2 : Traitement par le système :.....	54
7.4 Étape 3 : Affichage du résultat :.....	56
8 Schéma global du fonctionnement :.....	56
9 Description des fonctionnalités :.....	57
9.1 Fonctionnalité 1 : Système de recommandation de produits :.....	57
9.1.1 Comment ça fonctionne ?.....	57
9.1.2 Ce que l'utilisateur doit faire (Input) :.....	57
9.1.3 Ce que l'utilisateur obtient (Output) :.....	58
9.2 Fonctionnalité 2 : Analyse intelligente de la peau (AI Skin Analysis).....	59
9.2.1 Comment ça fonctionne ?.....	59
9.2.2 Ce que l'utilisateur doit faire (Input).....	60
9.2.3 Ce que l'utilisateur obtient (Output) :.....	60
9.3 Fonctionnalité 3 : Chatbot intelligent :.....	61
9.3.1 Comment ça fonctionne ?.....	61
9.3.2 Ce que l'utilisateur doit faire (Input) :.....	62
9.3.3 Ce que l'utilisateur obtient (Output) :.....	63
✓ 10 Implémentation du modèle IA :.....	64
10.1 Qu'est-ce qu'un réseau de neurones ?.....	64
10.2 L'entraînement du modèle (Training) :.....	66
10.3 Sauvegarde du modèle :.....	66
10.4 La prédiction (Prediction) :.....	67
10.5 Intégration du chatbot :.....	67

10.5 .1 Le fonctionnement général du chatbot :.....	67
10.5.2 La restriction du domaine (seulement les soins de la peau) :.....	68
10.5 .3 L'analyse des mots-clés (keywords) :.....	68
10.5.4 Les réponses aléatoires (random responses) :.....	68
10.5.5 La fonction fallback_answer() :.....	69
10.5.6 Le chatbot externe (Tawk.to) :.....	70
11 Tests et résultats :.....	70
11.1 Test du système de recommandation :.....	70
11.2 Test du module d'analyse IA (AI Skin Analysis) :.....	70
11.2.1 Comment nous avons testé :.....	70
11.3 Test du chatbot :.....	71
11.3.1 Comment nous avons testé.....	71
12.1 Récapitulatif de ce qui a été fait :.....	71
12.2 Les technologies utilisées :.....	72
12.3 Les points forts de l'application :.....	72.
12.4 Les limites actuelles.....	72
12.5 Perspectives d'amélioration (futur) :.....	73.
Conclusion.....	74
CONCLUSION GÉNÉRALE.....	76
Liste des Figures.....	
Figure 1.1 : Diagramme de cas d'utilisation Skincare.....	31
Figure1.2 : Diagramme de séquence Skincare.....	32
Figure 1.3 : Diagramme de séquence Analyse de peau.....	33
Figure 1.4 : Diagramme de séquence : Chatbot.....	34
Figure 1.5 : Diagramme des relations entre l'utilisateur, les analyses, recommandations et le Chatbot.....	35
Figure 1.6 : Diagramme d'activité du processus complet du système.....	36
Figure 4.1 : Interface du système de recommandation et analyse IA	52
Figure 4.2 : Interface du chatbot intelligent.....	52
Figure 4.1.1 : Interface du système de recommandation.....	59
La figure 4.1.2 : ci-dessous montre l'interface de l'analyse IA.	61
La figure 4.1.3 : ci-dessous montre l'interface du chatbot.....	64
Liste des Tableaux.....	
Tableau 4 : Tableau récapitulatif des outils (4.2)	48
Tableau 4.1 : Dataset des utilisateurs (users.csv)	48
Tableau 4.2 : Indicateurs des problèmes de peau	49
Tableau 3:. Dataset des produits (products.csv).....	49
Tableau 4: Dataset des interactions (interactions.csv).....	50
Tableau 5: Dataset des solutions (Skin Analysis).....	50
Rèfèrence	77
Abstract	79



Introduction générale



Introduction générale

Au cours des dernières années, l'évolution rapide des technologies numériques a profondément transformé plusieurs secteurs, notamment celui de la beauté et des soins de la peau [1]. L'Intelligence Artificielle (IA) joue aujourd'hui un rôle central dans cette transformation en offrant des solutions innovantes permettant d'améliorer l'expérience utilisateur et de répondre de manière plus précise aux besoins individuels [2]. Dans un contexte marqué par une grande diversité de produits cosmétiques et une augmentation constante de la demande en personnalisation, les consommateurs se retrouvent souvent confrontés à des difficultés dans le choix des produits les plus adaptés à leur type de peau et à leurs préférences [3].

En effet, le marché des cosmétiques propose une multitude de produits destinés à différents types de peau (grasse, sèche, mixte, sensible) ainsi qu'à divers problèmes cutanés tels que l'acné, les taches ou encore les rides. Cette diversité, bien qu'avantageuse, rend la prise de décision complexe pour les utilisateurs, surtout en l'absence de conseils personnalisés fiables [4]. Par conséquent, il devient nécessaire de développer des solutions intelligentes capables d'accompagner les utilisateurs dans leurs choix [5].

Dans ce cadre, les systèmes de recommandation apparaissent comme une solution efficace pour proposer des produits adaptés aux besoins spécifiques des utilisateurs [6]. Ces systèmes, largement utilisés dans le domaine du e-commerce, reposent sur l'analyse des données afin de suggérer des éléments pertinents [7]. Toutefois, les approches classiques présentent certaines limites, notamment en termes de personnalisation et d'adaptation en temps réel [8].

Par ailleurs, l'émergence de l'Intelligence Artificielle Générative a ouvert de nouvelles perspectives dans le domaine des systèmes intelligents [9]. Grâce à des modèles avancés capables de comprendre et de générer du contenu, il est désormais possible de concevoir des systèmes plus interactifs, plus flexibles et mieux adaptés aux attentes des utilisateurs [10]. L'intégration de ces technologies dans le domaine de la beauté permet ainsi d'améliorer la qualité des recommandations et d'offrir une expérience plus personnalisée [11].

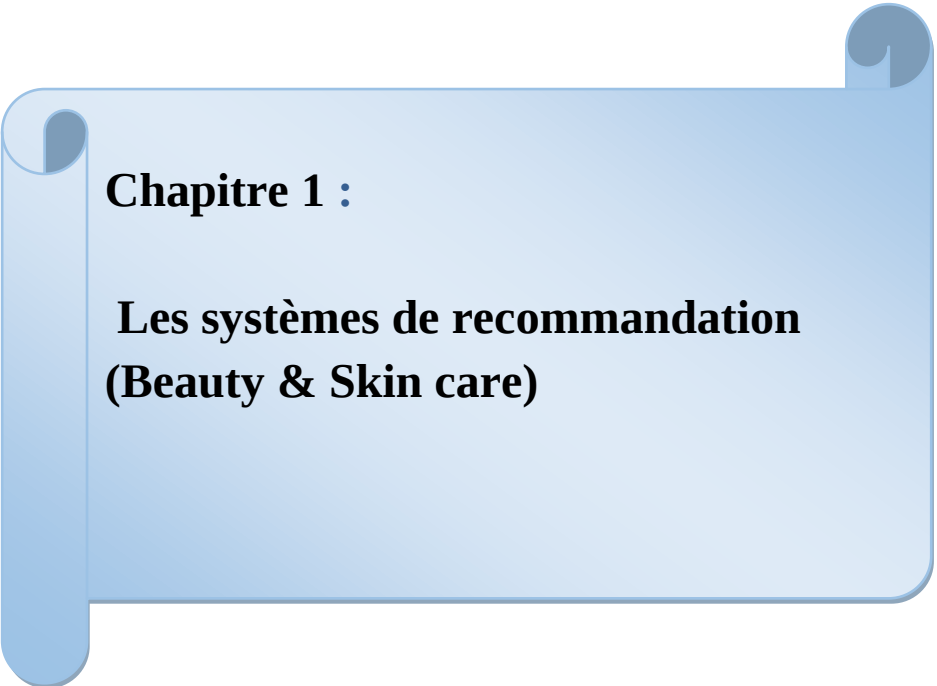
Ainsi, la problématique principale de ce travail peut être formulée comme suit : comment concevoir un système intelligent capable de recommander des produits cosmétiques de manière personnalisée en exploitant les capacités de l'Intelligence Artificielle Générative ?

Afin de répondre à cette problématique, l'objectif de ce mémoire est de concevoir et de développer un système de recommandation de produits cosmétiques basé sur l'Intelligence Artificielle Générative. Ce système vise à analyser les caractéristiques des utilisateurs, notamment leur type de peau, leurs préférences et leurs besoins spécifiques, afin de leur proposer des recommandations pertinentes et adaptées [12].

Pour atteindre cet objectif, une méthodologie basée sur l'étude des systèmes de recommandation, l'analyse des techniques d'Intelligence Artificielle Générative ainsi que la conception et l'implémentation d'un prototype fonctionnel a été adoptée [13]. L'utilisation

d'outils modernes et de modèles d'IA permet de mettre en place une solution performante et interactive [14].

Enfin, ce mémoire est structuré en quatre chapitres. Le premier chapitre est consacré à l'étude des systèmes de recommandation, en mettant l'accent sur leur application dans le domaine de la beauté et du skincare. Le deuxième chapitre présente les concepts fondamentaux de l'Intelligence Artificielle Générative ainsi que ses principales techniques. Le troisième chapitre est dédié à la conception du système proposé, incluant les différents diagrammes et l'architecture de l'application. Le quatrième chapitre porte sur l'implémentation du système, en décrivant les outils utilisés, les choix techniques effectués ainsi que les résultats obtenus à travers des captures d'écran de l'application développée.



Chapitre 1 :

Les systèmes de recommandation (Beauty & Skin care)

1. Introduction

Avec l'évolution rapide des technologies numériques et l'essor du commerce électronique, les systèmes de recommandation sont devenus des outils incontournables dans plusieurs domaines, notamment celui de la beauté et du skincare. Ces systèmes permettent d'analyser les préférences des utilisateurs afin de leur proposer des produits adaptés à leurs besoins spécifiques.

Dans l'industrie cosmétique, les consommateurs sont souvent confrontés à une grande variété de produits, ce qui rend le choix difficile. Ainsi, les systèmes de recommandation interviennent pour simplifier cette tâche en offrant des suggestions personnalisées basées sur différents types de données, comme les avis utilisateurs, les caractéristiques des produits et le type de peau .

1.1 Définition d'un système de recommandation

Un système de recommandation est un système intelligent d'aide à la décision conçu pour proposer à un utilisateur des éléments jugés pertinents en fonction de ses préférences explicites ou implicites [15]. Ces systèmes sont largement utilisés dans plusieurs domaines tels que le e-commerce, les réseaux sociaux, les plateformes de streaming ainsi que les applications de skincare et de beauté.

Les systèmes de recommandation reposent principalement sur des techniques d'intelligence artificielle permettant d'analyser les comportements des utilisateurs et de prédire leurs préférences futures [16]. Grâce à ces technologies, les recommandations proposées deviennent plus personnalisées et plus adaptées aux besoins spécifiques des utilisateurs.

Les principales techniques utilisées dans les systèmes de recommandation sont :

- Machine Learning (ML)
- Natural Language Processing (NLP)
- Data Mining

Le Machine Learning permet au système d'apprendre automatiquement à partir des données des utilisateurs afin d'améliorer la qualité des recommandations [17]. Le Natural Language Processing facilite l'analyse des avis et des descriptions textuelles des produits [18]. Quant au Data Mining, il permet d'extraire des informations utiles à partir de grandes quantités de données [19].

1.2 Importance dans le domaine de la beauté

Dans le domaine du skin care, chaque individu possède un type de peau différent (grasse, sèche, mixte, sensible). De ce fait, les recommandations doivent être précises et personnalisées



Les systèmes de recommandation permettent :

- D'améliorer l'expérience utilisateur .
- De réduire le temps de recherche .
- D'augmenter la satisfaction client .
- D'éviter les produits incompatibles avec la peau .

1.3 Approches utilisées pour la recommandation

Les systèmes de recommandation, également appelés systèmes de filtrage d'information, visent à aider les utilisateurs à faire face à l'abondance croissante de contenus disponibles, en sélectionnant et en proposant des informations pertinentes adaptées à leurs besoins spécifiques. Leur fonction principale est de filtrer un flux d'informations entrantes, en les collectant, en les analysant, puis en les triant pour ne suggérer que celles qui ont une forte probabilité d'intérêt pour un utilisateur donné [23]. Le principe de base repose sur une prédiction : le système évalue la probabilité qu'une information particulière soit jugée utile ou pertinente par l'utilisateur. Cette prédiction s'appuie généralement sur le profil de l'utilisateur, c'est-à-dire un ensemble de données représentant ses préférences, son comportement passé ou encore ses objectifs. En fonction de cette analyse, le système prend une décision : recommander ou non le contenu concerné .

1.3.1 Le filtrage collaboratif (Collaborative Filtering)

Le filtrage collaboratif est l'une des approches les plus utilisées dans les systèmes de recommandation. Il repose sur l'idée que les utilisateurs ayant des comportements ou des préférences similaires auront tendance à apprécier les mêmes items. Ainsi, le système tente de prédire l'utilité ou l'intérêt d'un produit pour un utilisateur donné en se basant sur les évaluations et les interactions d'autres utilisateurs présentant des profils similaires .

Cette approche s'inspire du principe du « bouche-à-oreille », où les opinions et expériences des autres utilisateurs jouent un rôle important dans la prise de décision. Bien que ce concept soit formalisé récemment dans le domaine de l'intelligence artificielle, il reproduit un comportement humain naturel consistant à se fier aux recommandations sociales pour évaluer un produit ou un service .

Dans la pratique, notamment dans les systèmes de e-commerce, le filtrage collaboratif analyse les notes et les avis attribués aux produits afin d'identifier des corrélations entre les préférences des utilisateurs. Par exemple, dans un système de recommandation de livres, l'algorithme recherche des utilisateurs similaires à un utilisateur cible et recommande uniquement les livres bien notés par ces profils proches.

Ce type de système recommande donc des produits en se basant sur les préférences d'autres utilisateurs similaires. Par exemple, si plusieurs utilisateurs ayant des besoins ou des caractéristiques similaires

➤ **Avantages :**

- Simple à mettre en œuvre .
- Performant lorsque les données utilisateur sont abondantes .
- Capable de découvrir des préférences implicites .

➤ **Inconvénients :**

- Problème de démarrage à froid (cold start) .
- Dépend fortement du nombre d'utilisateurs et d'évaluations disponibles .
- Moins efficace lorsque les données sont rares .

1.3.2 Filtrage basé sur le contenu (Content-Based Filtering)

Le filtrage basé sur le contenu est une approche des systèmes de recommandation qui consiste à proposer à l'utilisateur des produits ou des documents similaires à ceux qu'il a déjà appréciés ou consultés. Ce type de système repose sur l'analyse des caractéristiques des items, sans prendre en compte les interactions ou les avis des autres utilisateurs .

Dans cette approche, la similarité est calculée entre les préférences de l'utilisateur et les attributs descriptifs des produits (métadonnées, ingrédients, catégories, etc.). Par exemple, dans un système de recommandation de produits cosmétiques, si un utilisateur apprécie un produit contenant de l'acide hyaluronique, le système lui proposera d'autres produits ayant des ingrédients ou des propriétés similaires.

La mise en place de ce type de filtrage repose principalement sur deux étapes essentielles : la représentation des items et la construction du profil utilisateur. Le système analyse les éléments pertinents pour créer un profil qui reflète les intérêts de l'utilisateur. Ensuite, il sélectionne les produits correspondant le mieux à ce profil.

De plus, le profil utilisateur est continuellement mis à jour en fonction du retour d'expérience, c'est-à-dire selon les produits jugés pertinents ou non par l'utilisateur. Cette adaptation permet d'améliorer progressivement la qualité des recommandations.

Ce type de filtrage est largement utilisé dans différents domaines tels que la recommandation de pages web, de cours en ligne, ou encore de produits spécialisés comme les cosmétiques.

➤ **Avantages :**

- Personnalisation élevée des recommandations
- Ne dépend pas des autres utilisateurs
- Fonctionne même avec peu d'utilisateurs

➤ **Inconvénients :**

- Recommandations limitées aux mêmes types de contenu
- Manque de diversité dans les suggestions
- Peut enfermer l'utilisateur dans un seul type de produits (filter bubble)

1.3.3 Filtrage hybride (Hybrid Filtering)

Le filtrage hybride combine le filtrage collaboratif et le filtrage basé sur le contenu afin de tirer parti des avantages des deux approches et de réduire leurs limitations respectives. En effet, chaque méthode possède ses points forts et ses faiblesses : le filtrage collaboratif est efficace lorsque les données utilisateurs sont nombreuses mais souffre du problème de démarrage à froid et de la rareté des évaluations, tandis que le filtrage basé sur le contenu fonctionne indépendamment des autres utilisateurs mais peut limiter la diversité des recommandations .

Ainsi, le filtrage hybride vise à intégrer ces deux approches afin d'améliorer la qualité des recommandations et de rendre le système plus robuste [34]. Cette intégration peut être réalisée selon plusieurs stratégies :

- Implémenter les deux méthodes séparément puis combiner leurs résultats pour générer une recommandation finale.
- Ajouter certaines caractéristiques du filtrage collaboratif dans le filtrage basé sur le contenu.
- Intégrer certaines caractéristiques du filtrage basé sur le contenu dans le filtrage collaboratif.
- Construire un modèle unifié qui combine directement les deux approches dans un seul système.

Dans l'ensemble, les systèmes hybrides sont largement utilisés dans les applications modernes de recommandation, notamment dans le domaine du e-commerce et des produits cosmétiques, car ils permettent d'améliorer la précision des recommandations tout en réduisant les limitations des approches individuelles.

➤ **Avantages :**

- Meilleure précision des recommandations
- Réduction des limites du filtrage collaboratif et du contenu
- Système plus robuste et plus stable
- Les systèmes de recommandation dans le domaine de la beauté reposent sur plusieurs techniques issues de l'intelligence artificielle, permettant d'offrir des recommandations personnalisées adaptées aux besoins des utilisateurs. Parmi les principales techniques utilisées, on trouve le Machine Learning, le Deep Learning, la Computer Vision et le Natural Language Processing (NLP).

➤ **Machine Learning :**

- Le Machine Learning est largement utilisé pour analyser les données des utilisateurs telles que les préférences, les achats et les évaluations. Il permet de prédire les produits cosmétiques susceptibles de convenir à un utilisateur en fonction de son historique et de son comportement. Cette approche constitue la base des systèmes de recommandation modernes[20].

➤ **Deep Learning :**

- Le Deep Learning, basé sur les réseaux de neurones artificiels, permet de traiter des données complexes avec une grande précision. Dans les systèmes de beauté, il est utilisé pour améliorer la personnalisation des recommandations et pour analyser des données non structurées comme les images ou les textes.[21]

➤ **Computer Vision (analyse du visage) :**

- La Computer Vision permet l'analyse automatique des images du visage afin d'identifier des caractéristiques telles que le type de peau, l'acné, les rides ou les taches. Ces informations sont essentielles pour proposer des produits cosmétiques adaptés à l'état réel de la peau de l'utilisateur.[22]

➤ **NLP (Natural Language Processing):**

- Le NLP est utilisé pour analyser les avis et commentaires des utilisateurs concernant les produits cosmétiques. Il permet d'extraire les sentiments et les opinions afin de mieux comprendre la satisfaction des clients et d'améliorer la qualité des recommandations.[23]

1.4 Technologies utilisées

1.4.1 Machine Learning (ML) :

Le Machine Learning permet d'apprendre à partir des données et d'améliorer les recommandations avec le temps.

Exemples d'algorithmes :

- K-Nearest Neighbors (KNN)

- Random Forest
- Réseaux de neurones

1.4.2 Natural Language Processing (NLP) :

Le NLP est utilisé pour analyser les avis utilisateurs.

□ Exemple :

- Extraire les sentiments (positif / négatif)
- Identifier les mots-clés (hydratant, irritant...)

1.5 Applications des systèmes de recommandation dans le domaine de la beauté :

Le secteur de la beauté est l'un des domaines où les systèmes de recommandation jouent un rôle très important.

1.5.1 Personnalisation des produits skincare :

Les systèmes analysent :

- type de peau (grasse, sèche, mixte)
- problèmes cutanés (acné, rides...)
- sensibilité

Ensuite ils proposent des routines personnalisées.

1.5.2 Recommandation de maquillage :

Les recommandations peuvent être basées sur :

- carnation
- style
- occasion

1.5.3 Exemples réels :

Des entreprises comme Sephora et L'Oréal utilisent des systèmes intelligents pour :

- analyser la peau
- proposer des produits adaptés
- améliorer l'expérience utilisateur

1.6 Défis des systèmes de recommandation dans la beauté :

Malgré leurs avantages, ces systèmes présentent plusieurs défis :

1.6.1 Problème de données :

- Données insuffisantes
- Données incorrectes

1.6.2 Personnalisation complexe :

Chaque utilisateur est unique, ce qui rend la recommandation difficile.

1.6.3 Évolution des besoins :

Les préférences changent avec le temps.

1.7 Importance des systèmes de recommandation :

Les systèmes de recommandation permettent :

- ✓ Améliorer l'expérience utilisateur
- ✓ Gagner du temps
- ✓ Augmenter les ventes
- ✓ Offrir des solutions personnalisées

Dans le domaine du skincare, ils aident à éviter des erreurs pouvant nuire à la peau.

Conclusion

surcharge d'informations, notamment dans le domaine de la beauté et des soins de la peau. Ils permettent de réduire considérablement le temps de recherche des utilisateurs tout en leur proposant des produits pertinents et personnalisés, qu'ils n'auraient peut-être pas découverts autrement.

Au cours de ce chapitre, nous avons exploré les différentes techniques de recommandation, telles que les approches basées sur le contenu, le filtrage collaboratif et les systèmes hybrides. Nous avons également présenté d'autres formes de recommandation, notamment celles basées sur les interactions entre utilisateurs (people-to-people) ainsi que sur les tags.

Par ailleurs, l'intégration des techniques d'intelligence artificielle, telles que le Machine Learning, le Deep Learning, la Computer Vision et le Natural Language Processing, a permis d'améliorer significativement la précision et la qualité des recommandations, en exploitant à la fois les données utilisateurs, les images et les avis en ligne.

Cependant, malgré leurs performances, ces systèmes présentent encore certaines limites, notamment en termes de qualité des données, de démarrage à froid (cold start) et de complexité des modèles. Ces défis ouvrent la voie à l'émergence de nouvelles approches plus avancées, en particulier l'Intelligence Artificielle Générative, qui sera abordée dans le chapitre suivant.

Chapitre 2 : L'Intelligence Artificielle Générative

Introduction

L'Intelligence Artificielle (IA) désigne un ensemble de technologies capables de réaliser des tâches nécessitant généralement une intelligence humaine, telles que l'apprentissage automatique (Machine Learning), la reconnaissance de formes, la prise de décision et la compréhension du langage naturel. Grâce à l'exploitation des algorithmes et des données, ces systèmes sont capables de s'améliorer progressivement en fonction de l'expérience et des nouvelles informations.

On distingue généralement trois types d'intelligence artificielle selon leurs capacités : l'Intelligence Artificielle Faible (ANI – Artificial Narrow Intelligence), l'Intelligence Artificielle Générale (AGI) et l'Intelligence Artificielle Superintelligente (ASI). À l'heure actuelle, la majorité des applications utilisées, y compris les systèmes de recommandation et les intelligences artificielles génératives, appartiennent à la catégorie de l'IA Faible, spécialisée dans des tâches bien précises. Les deux autres catégories restent encore théoriques et font l'objet de recherches avancées.

Au cours des dernières années, l'IA a connu une évolution remarquable avec l'émergence de nouvelles approches capables non seulement d'analyser les données, mais aussi de générer du contenu. Cette nouvelle branche, appelée Intelligence Artificielle Générative, représente une avancée majeure dans le domaine de l'informatique.

Contrairement aux systèmes traditionnels, qui se limitent à l'analyse des données existantes, l'IA générative permet de produire de nouvelles données, telles que du texte, des images ou encore des recommandations personnalisées. Cette capacité ouvre de nouvelles perspectives dans plusieurs domaines, notamment celui de la beauté et du skincare, où la personnalisation joue un rôle essentiel.



2.1 Définition de l'Intelligence Artificielle Générative

L'Intelligence Artificielle Générative désigne un ensemble de techniques d'intelligence artificielle capables de produire du contenu nouveau à partir de données d'apprentissage. Contrairement aux approches traditionnelles qui se limitent à l'analyse ou à la classification des données, l'IA générative permet de créer des données originales telles que du texte, des images ou encore des recommandations personnalisées[24].

Ces systèmes reposent principalement sur des modèles avancés, notamment les réseaux de neurones profonds, capables d'apprendre des structures et des patterns à partir de grandes quantités de données. Une fois entraînés, ces modèles peuvent générer des résultats similaires aux données d'origine, tout en introduisant des variations nouvelles et pertinentes[25].

L'IA générative trouve aujourd'hui de nombreuses applications, notamment dans la génération de texte (chatbots intelligents), la création d'images, ainsi que dans les systèmes de recommandation avancés, où elle permet de proposer des solutions plus personnalisées et innovantes, en particulier dans des domaines comme la beauté et le skincare [26] .

2.2 Différence entre l'IA classique et l'IA générative

L'intelligence artificielle (IA) classique et l'intelligence artificielle générative représentent deux approches différentes dans le domaine de l'IA. L'IA classique est principalement conçue pour analyser des données existantes afin d'effectuer des tâches précises telles que la classification, la détection ou la prédiction. Elle se base sur des règles ou des modèles appris pour fournir des résultats généralement fixes et déterminés à l'avance[27].

En revanche, l'IA générative se distingue par sa capacité à produire de nouveaux contenus à partir des données sur lesquelles elle a été entraînée. Contrairement à l'IA classique, elle ne se limite pas à l'analyse, mais elle est capable de créer du texte, des images, ou encore des recommandations personnalisées[28]. Cette capacité de génération rend l'interaction avec l'utilisateur plus dynamique et plus flexible.

Ainsi, la principale différence entre ces deux types d'intelligence artificielle réside dans leur objectif : l'IA classique vise à comprendre et à exploiter les données existantes, tandis que l'IA générative cherche à créer de nouvelles données en s'inspirant de celles-ci[29]. Cette évolution permet de développer des systèmes plus intelligents, capables de s'adapter aux besoins spécifiques des utilisateurs en temps réel.

2.3 artificielle au service du diagnostic de peau

L'intégration de l'intelligence artificielle dans le domaine du diagnostic de la peau a considérablement amélioré la précision et l'efficacité des analyses dermatologiques. Grâce à des dispositifs avancés tels que les dermoscopes équipés de caméras haute définition, il est désormais possible de collecter une grande quantité de données relatives à l'état de la peau, incluant la texture, la pigmentation et les anomalies cutanées[30].

Les algorithmes d'intelligence artificielle, notamment ceux basés sur le *Machine Learning* et le *Deep Learning*, permettent d'analyser ces données de manière rapide et approfondie, facilitant ainsi la détection précoce de diverses affections cutanées telles que l'acné, les rides, les taches

pigmentaires ou encore certaines lésions suspectes. Ces systèmes contribuent également à proposer des recommandations personnalisées en matière de soins de la peau.

Cependant, malgré ses performances, l'intelligence artificielle demeure une technologie conçue par l'humain et n'est pas exempte d'erreurs liées à la conception des algorithmes ou à la qualité des données utilisées [31]. Par conséquent, elle ne peut pas se substituer à l'expertise d'un professionnel de santé, mais constitue plutôt un outil d'assistance visant à améliorer la qualité du diagnostic et la prise de décision.



2.4 Les avantages de l'intelligence artificielle dans le diagnostic de peau

L'utilisation de l'intelligence artificielle dans le domaine du diagnostic de peau présente de nombreux avantages, notamment en termes de précision, de personnalisation et d'efficacité. Grâce à des algorithmes avancés capables d'analyser des images cutanées avec une grande précision, l'IA permet d'identifier des anomalies ou des problèmes dermatologiques subtils que l'œil humain pourrait ne pas détecter. Cette capacité d'analyse fine contribue à l'établissement d'un diagnostic plus fiable et personnalisé [33].

Cette précision dans le diagnostic joue un rôle essentiel dans l'optimisation des soins proposés, en facilitant le choix de traitements et de produits cosmétiques mieux ciblés, ce qui augmente leur efficacité et réduit les risques d'erreurs ou d'inadéquation. Ainsi, l'IA améliore considérablement la qualité de la prise en charge en dermatologie et en cosmétique [33].

Par ailleurs, pour les professionnels du domaine, l'intégration de ces technologies permet un gain de temps significatif, une meilleure organisation du travail ainsi qu'une amélioration de la satisfaction client. Pour les utilisateurs, cela se traduit par une réduction des coûts liés à l'essai de produits inadaptés, ainsi qu'une prise de décision plus éclairée concernant leur routine de soins[34].

Enfin, les systèmes de diagnostic basés sur l'intelligence artificielle offrent également un suivi précis de l'évolution de la peau et de l'efficacité des traitements appliqués. Ce suivi continu permet aux professionnels d'ajuster leurs recommandations et d'améliorer constamment leurs pratiques, renforçant ainsi la qualité globale des soins esthétiques.

2.5 Limites et défis de l'Intelligence Artificielle Générative

- Malgré les progrès considérables réalisés dans le domaine de l'Intelligence Artificielle Générative, cette technologie présente encore plusieurs limites importantes qui doivent être prises en compte, notamment dans les applications liées à la beauté et aux soins de la peau.[35]
- Tout d'abord, les outils d'intelligence artificielle peuvent produire des résultats incorrects ou des recommandations erronées. En effet, l'IA peut identifier de faux problèmes ou, au contraire, ne pas détecter certaines anomalies pourtant visibles. Ces erreurs peuvent être causées par plusieurs facteurs, tels qu'une mauvaise qualité des données d'entrée (par exemple une mauvaise prise d'image ou une mauvaise utilisation des dispositifs d'analyse), ou encore des limites dans les algorithmes utilisés[36].
- De plus, même si les modèles d'IA sont entraînés sur de grandes bases de données contenant différents types de peau, la peau humaine reste extrêmement complexe et influencée par de nombreux facteurs internes et externes (alimentation, stress, environnement, hormones, etc.). Par conséquent, les recommandations générées par l'IA peuvent parfois être inadaptées et conduire à des choix de produits inefficaces, voire à des traitements inappropriés.
- pour cette raison que l'IA ne peut pas remplacer totalement l'expertise humaine. Elle doit être utilisée comme un outil d'aide à la décision, en complément des connaissances d'un professionnel de la peau, tel qu'un dermatologue ou une esthéticienne qualifiée. En effet, les professionnels possèdent une compréhension plus globale et nuancée des besoins des clients, en tenant compte non seulement des aspects physiques, mais aussi des facteurs émotionnels, psychologiques et contextuels, que l'IA ne peut pas totalement interpréter.
- Par ailleurs, une autre limite importante des systèmes d'intelligence artificielle dans le domaine de la beauté est le biais algorithmique et le biais de marque. Certains systèmes de diagnostic ou de recommandation peuvent favoriser les produits d'une marque spécifique, notamment celle qui a développé l'outil, ce qui limite l'objectivité des recommandations. Ce type de biais peut réduire la qualité des suggestions proposées aux utilisateurs et nuire à la confiance dans le système.
- Enfin, les systèmes d'IA présentent également une forte dépendance aux données. La qualité des recommandations dépend directement de la qualité et de la diversité des données utilisées lors de l'entraînement. Des données incomplètes ou biaisées peuvent donc entraîner des résultats peu fiables.
- Ainsi, bien que l'IA Générative offre des perspectives prometteuses dans le domaine de la beauté et des systèmes de recommandation, elle doit être utilisée avec prudence et en complément de l'expertise humaine afin de garantir des résultats fiables et adaptés

2.6 Domaine de la beauté et du skincare

Dans le domaine de la beauté, l'IA générative permet :

- d'analyser le type de peau
- de proposer des produits adaptés
- de créer des routines personnalisées

Conclusion

L'Intelligence Artificielle Générative représente aujourd'hui une avancée majeure dans le domaine de l'intelligence artificielle, en introduisant une nouvelle manière d'interagir avec les données et les utilisateurs. Contrairement aux approches traditionnelles, elle ne se limite pas à l'analyse ou à la prédiction, mais permet de générer du contenu original, ouvrant ainsi la voie à des applications plus intelligentes et plus personnalisées.

Au cours de ce chapitre, nous avons présenté les concepts fondamentaux de l'IA générative, ses principales techniques telles que les réseaux de neurones et les modèles de langage, ainsi que ses domaines d'application, notamment dans le secteur de la beauté et du skincare. Cette technologie offre des avantages significatifs, notamment en termes de personnalisation, d'interactivité et d'amélioration de l'expérience utilisateur.

Cependant, malgré ses performances, l'IA générative présente encore certaines limites, telles que sa dépendance aux données, les risques d'erreurs et les biais algorithmiques. Ces défis montrent que, bien qu'elle soit très puissante, cette technologie ne peut pas remplacer totalement l'expertise humaine, mais doit être utilisée comme un outil complémentaire.

Ainsi, l'intégration de l'IA générative dans les systèmes de recommandation constitue une solution prometteuse pour répondre aux besoins croissants de personnalisation, notamment dans le domaine des produits cosmétiques. Dans le chapitre suivant, nous allons nous intéresser à la conception du système proposé, en présentant les différents modèles et diagrammes nécessaires à sa réalisation.

chapitre 3 : conception

3.1 Introduction

Ce chapitre présente la phase de conception du système développé. Il décrit l'architecture globale de l'application, les différents modules fonctionnels, ainsi que la modélisation UML utilisée pour représenter les interactions entre les composants du système. L'objectif est de fournir une vision claire de l'organisation interne de la solution de recommandation de produits cosmétiques basée sur l'intelligence artificielle générative.

3.1 L'utilisation de diagramme

3.1.1 Définition d'un diagramme :

Un diagramme UML est une représentation graphique qui s'intéresse à un aspect précis du modèle. C'est une perspective du modèle, pas "le modèle". Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis). Un type de diagramme UML véhicule une sémantique précise (un type de diagramme offre toujours la même vue d'un système). Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système. Par extension et abus de langage, un diagramme UML est aussi un modèle (un diagramme modélise un aspect du modèle global)[37].

3.1.2 Caractéristiques des diagrammes UML :

diagrammes UML supportent l'abstraction. Leur niveau de détail caractérise le niveau d'abstraction du modèle. La structure des diagrammes UML et la notation graphique des éléments de modélisation est normalisée (document "UML notation guide"). 29 Rappel : la sémantique des éléments de modélisation et de leur utilisation est définie par le métamodèle UML (document "UML semantics"). [38]

3.1.3 Concepts d'UML :

UML se veut être une notation simple, précise et homogène, permettant. Un bon rendu visuel. Elle décrit le réalisé plutôt que le processus de réalisation. UML préconise de séparer les aspects fonctionnels, technologiques et architecturaux. Pour la compréhension entre les différents acteurs d'un projet UML propose des diagrammes qui vont permettre d'éclaircir les spécifications[39].

3.2 Architecture générale du système

3.2.1 Vue d'ensemble

Le système proposé repose sur une architecture modulaire composée de trois blocs principaux :

- Module de recommandation de produits
- Module d'analyse de peau basé sur l'IA
- Module chatbot intelligent

Ces modules interagissent avec une base de données centralisée et un moteur d'intelligence artificielle.[40]

3.2.2 Description des modules

3.2.2.1 Module Recommandation

Ce module est responsable de proposer des produits cosmétiques adaptés à chaque utilisateur. Il utilise :

- Les préférences utilisateur
- Les caractéristiques de la peau
- L'historique d'utilisation

3.2.2.2 Module Analyse IA

Ce module analyse la peau de l'utilisateur à partir d'une image ou d'un questionnaire.

Fonctions principales :

- Détection des types de peau (sèche, grasse, mixte)
- Détection des problèmes (acné, taches, rougeurs)
- Génération d'un profil cutané

3.2.2.3 Module Chatbot

Le chatbot sert d'interface conversationnelle entre l'utilisateur et le système. Fonctions :

- Répondre aux questions sur les produits
- Donner des conseils skincare
- Aider dans la navigation de l'application

Basé sur :

- NLP (Natural Language Processing)
- Modèles de langage (LLM)

3.2.3 Schéma d'architecture

Le système suit une architecture client-serveur :

Utilisateur → Interface mobile/web → API Backend → Modules IA → Base de données

3.3 Diagrammes UML

3.3.1 Diagramme de cas d'utilisation

3.3.1.1 Acteurs identifiés

- Utilisateur
- Administrateur
- Système IA

3.3.1.2 Cas d'utilisation

- S'inscrire / se connecter
- Recevoir une recommandation
- Analyser la peau
- Converser avec le chatbot
- Consulter les produits

3.3.1.3 Schéma du diagramme

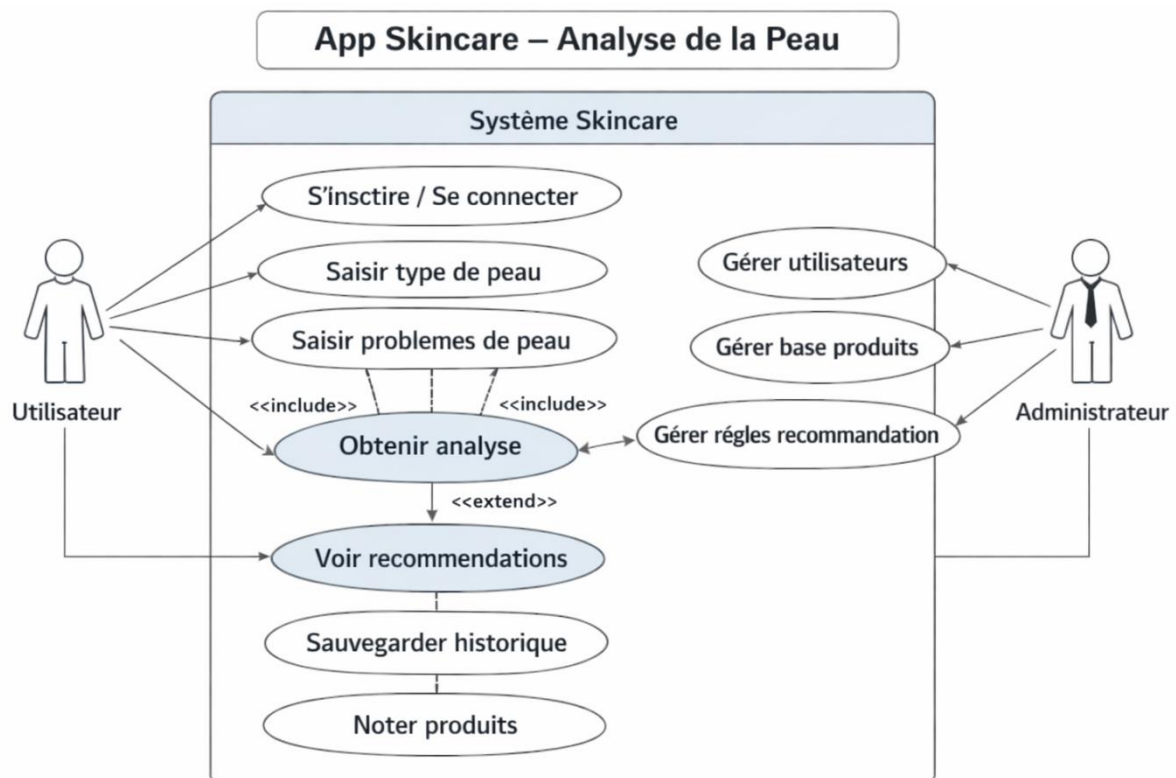


figure 1.1 : Diagramme de cas d'utilisation Skincare

3.3.2 Diagramme de séquence

3.3.2.1 Scénario 1 : Recommandation

Utilisateur → Interface → API → Module IA → Base de données → Résultat → Utilisateur

Scénario 1 : Recommandation

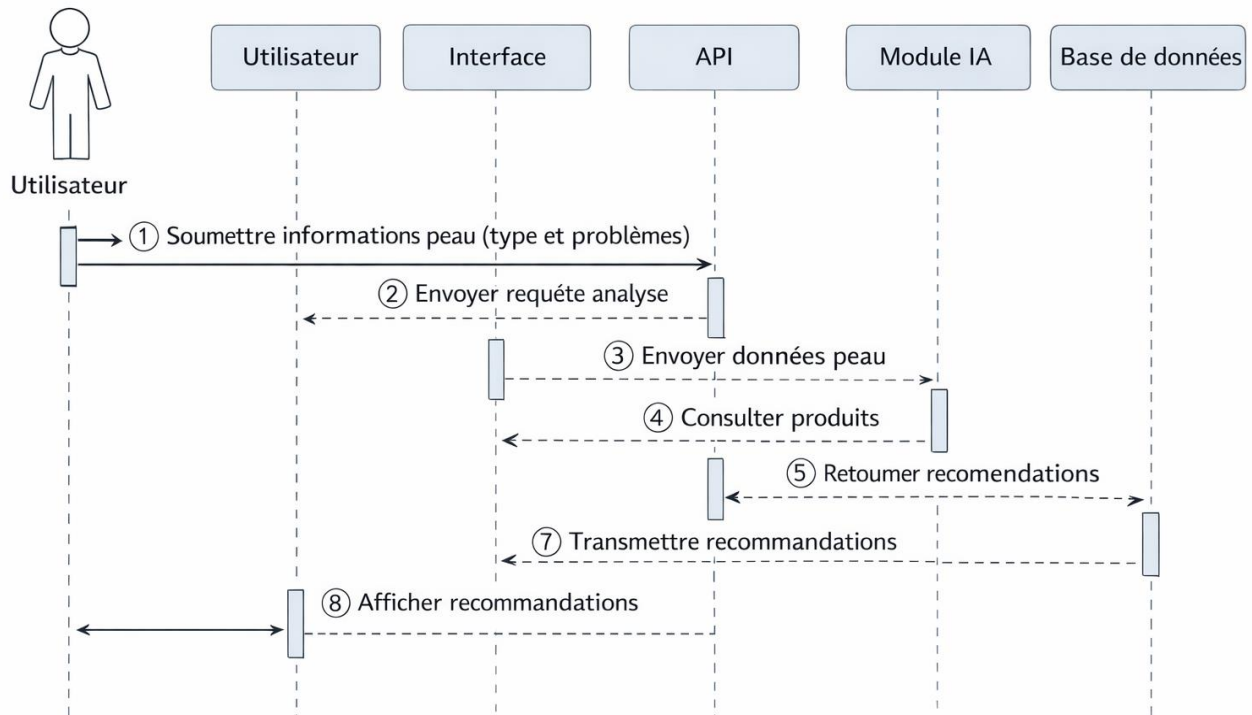


Diagramme de séquence - Scénario 1 : Recommandation

figure 1.2 : Diagramme de séquence Skincare

3.3.2.2 Scénario 2 : Analyse de peau

Utilisateur → Upload image → Module IA → Analyse CNN → Résultat → Profil utilisateur

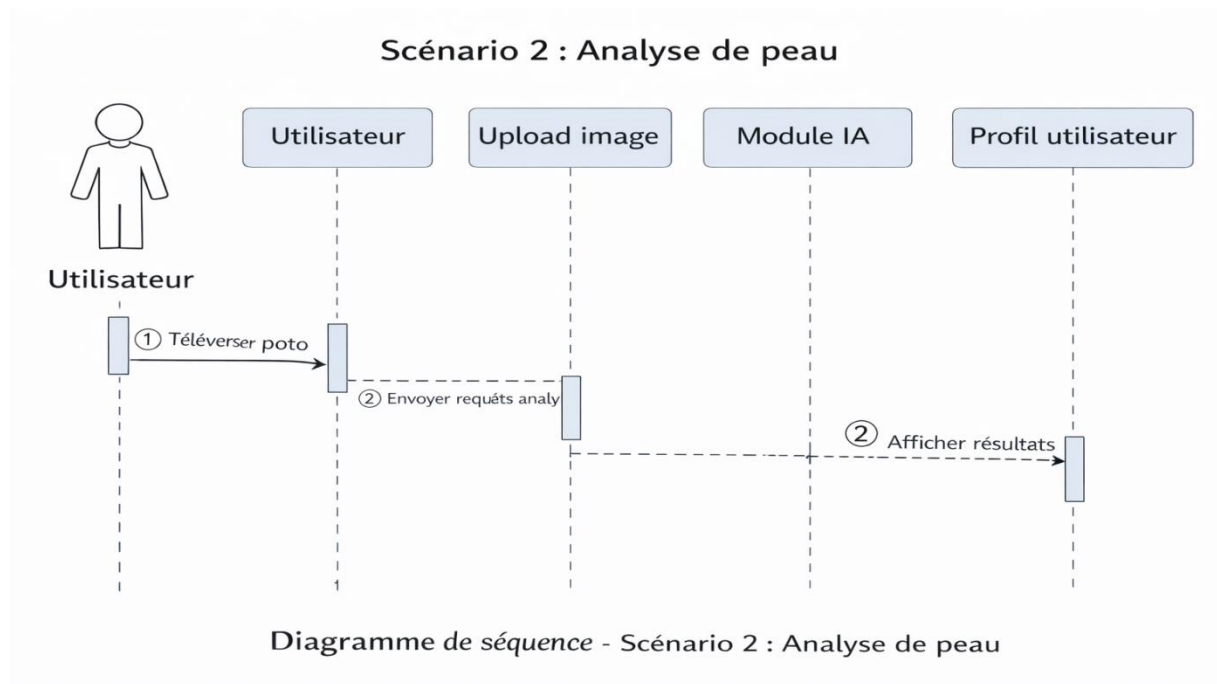


Figure 1.3 : Diagramme de séquence Analyse de peau

3.3.2.3 Scénario 3 : Chatbot

Utilisateur → Message → NLP Engine → Génération réponse → Retour utilisateur

figure 3 : Diagramme de séquence Analyse de peau

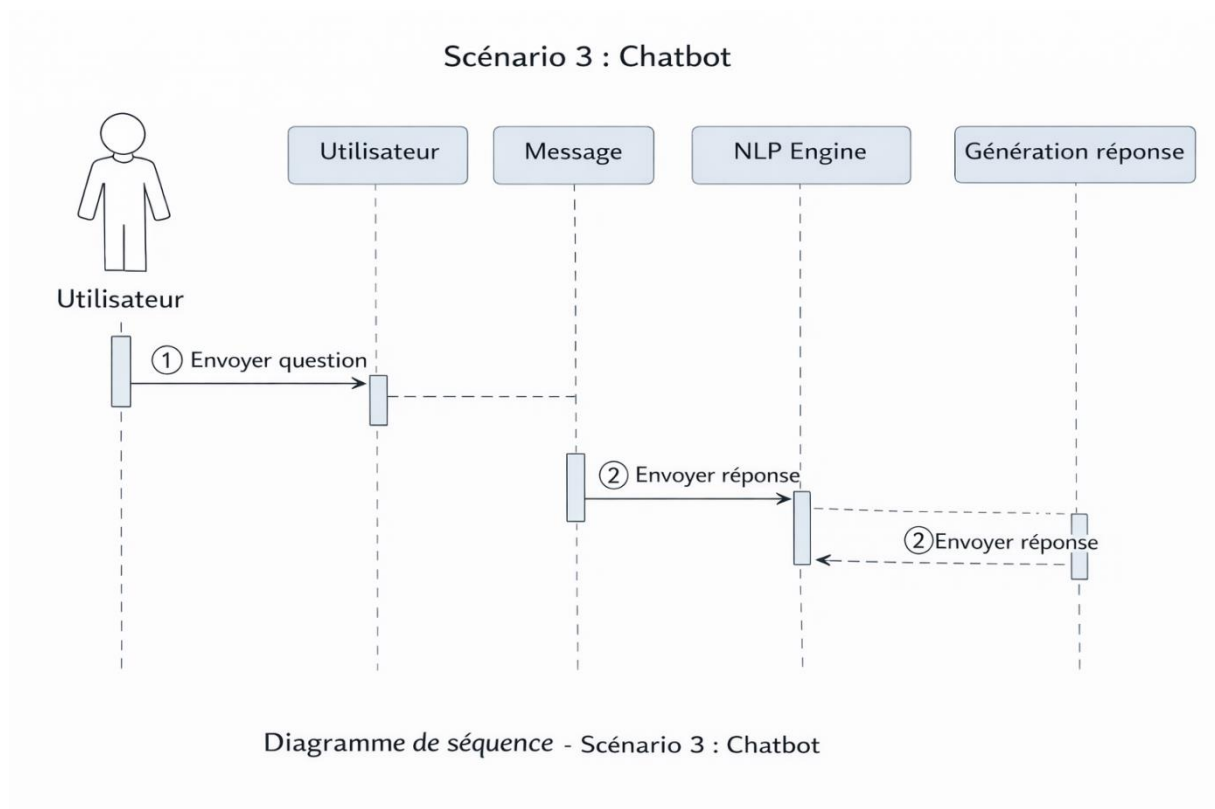


figure 1.4 : Diagramme de séquence : Chatbot

3.3.3 Diagramme de classes

3.3.3.1 Classes principales

- Utilisateur
- Produit
- AnalysePeau
- Message
- Recommandation

3.3.3.2 Attributs et méthodes (exemple)

Classe Utilisateur

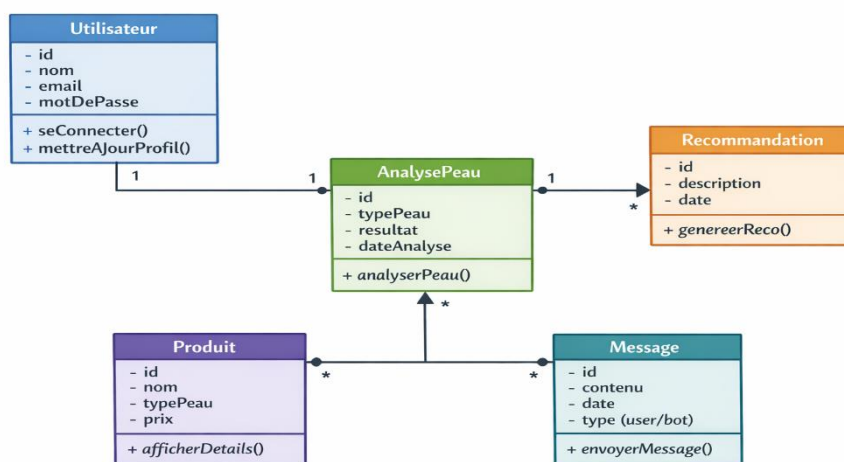
- id
- nom
- email
- mot de passe
- seConnecter()
- mettreAJourProfil()

Classe Produit

- id
- nom
- type peau
- prix
- afficherDetails()

3.3.3.3 Relations

- Un utilisateur possède plusieurs analyses
- Un utilisateur reçoit plusieurs recommandations
- Un chatbot échange plusieurs messages



- **Figure 1.5 : Diagramme des relations entre l'utilisateur, les analyses, recommandations et le chatbot**

3.3.4 Diagramme d'activité

3.3.4.1 Processus complet

1. Connexion utilisateur
2. Choix de service (analyse / recommandation / chatbot)
3. Traitement IA
4. Génération de résultat
5. Affichage à l'utilisateur

3.3.4.2 Schéma

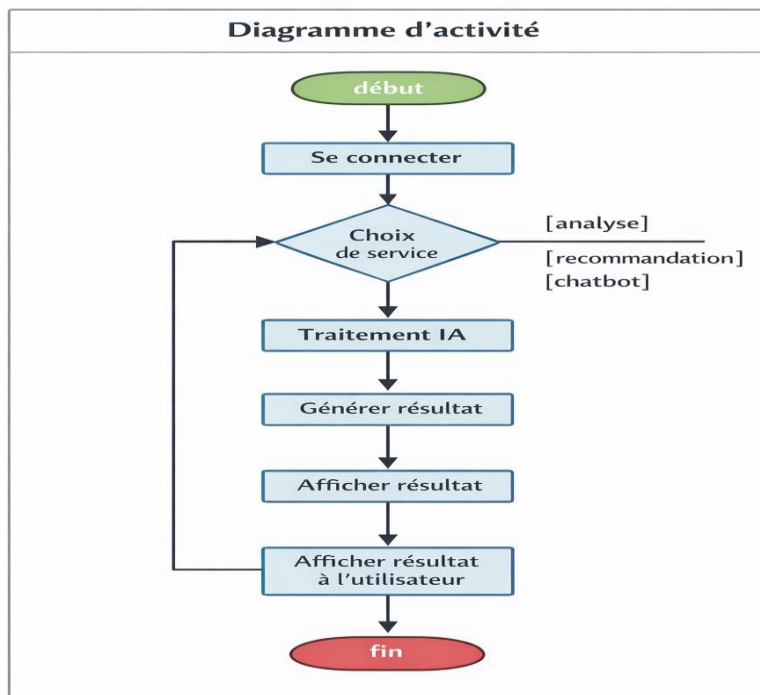


Figure 1.6 : Diagramme d'activité du processus complet du système

3.4 Modélisation de la base de données

3.4.1 Dictionnaire de données

3.4.1.1 Table Utilisateurs

- id_user (PK)
- nom
- email
- mot_de_passe
- type_peau

3.4.1.2 Table Produits

- id_produit (PK)
- nom
- marque
- type_peau
- prix

3.4.1.3 Table Messages

- id_message (PK)
- id_user (FK)
- contenu
- date

3.4.1.4 Table Analyses

- id_analyse (PK)
- id_user (FK)
- resultat
- image_path

3.4.2 Schéma relationnel (MCD/MLD)

Relations principales :

- Utilisateur 1 → N Messages
- Utilisateur 1 → N Analyses
- Utilisateur N ↔ N Produits (via Recommandation)

3.5 Conception des interfaces utilisateur

3.5.1 Interface de recommandation

- Affichage des produits adaptés
- Filtres (prix, type peau)
- Bouton “recommander”

3.5.2 Interface d’analyse IA

- Upload image peau
- Bouton analyser
- Résultat affiché (type peau + conseils)

3.5.3 Interface chatbot

- Zone de texte
- Historique conversation
- Réponses IA en temps réel

3.5.4 Maquette globale

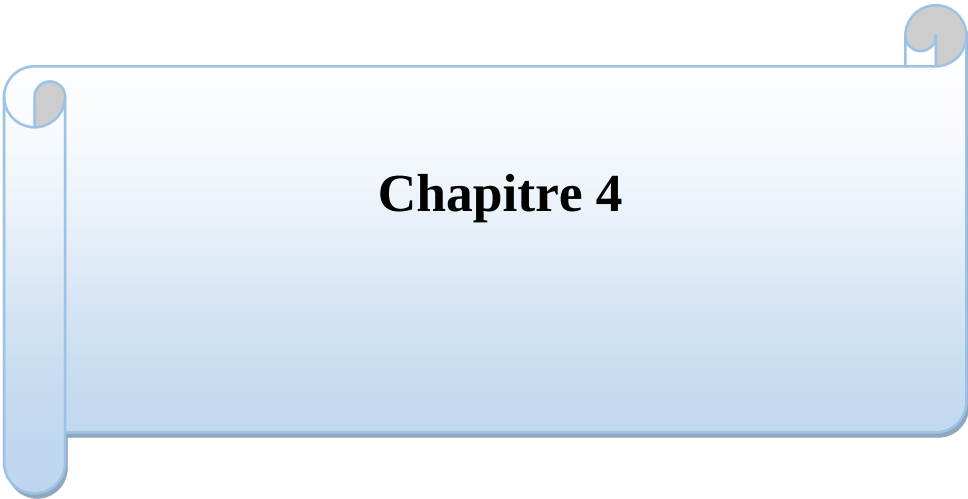
L’application est structurée de manière simple :

- Home
- Analyse

- Recommandation
- Chatbot
- Profil utilisateur

Conclusion

Ce chapitre a présenté la conception globale du système de recommandation de produits cosmétiques basé sur l'intelligence artificielle générative. Nous avons détaillé l'architecture du système, les différents modules fonctionnels ainsi que les diagrammes UML et la modélisation de la base de données. Cette étape est essentielle pour assurer une implémentation cohérente et structurée dans le chapitre suivant.



Chapitre 4

Introduction

chapitre présente la phase d'implémentation de notre application intelligente dédiée aux soins de la peau basée sur l'intelligence artificielle. Il s'agit de la partie pratique du projet, où toutes les idées et les modèles développés précédemment sont concrétisés sous forme d'une application web fonctionnelle.

Dans ce chapitre, nous détaillons l'environnement de développement utilisé, les outils et technologies adoptés, ainsi que la manière dont l'application a été construite et organisée. Nous expliquons également l'architecture générale du système et le rôle de chaque module dans le fonctionnement global de l'application.

Notre solution permet de proposer une expérience utilisateur complète et interactive, en offrant plusieurs fonctionnalités principales, notamment :

- la recommandation personnalisée de produits cosmétiques basée sur les données des utilisateurs,
- l'analyse du type de peau et des problèmes cutanés à l'aide d'un modèle d'intelligence artificielle,
- ainsi qu'un chatbot intelligent capable de répondre aux questions liées aux soins de la peau.

L'objectif de ce chapitre est également d'illustrer le fonctionnement de l'application à travers des captures d'écran et de montrer comment les différentes composantes interagissent pour fournir un système cohérent, moderne et facile à utiliser.

Enfin, cette phase d'implémentation permet de valider concrètement l'approche proposée et de démontrer l'efficacité de l'intégration de l'intelligence artificielle dans une application web destinée au domaine cosmétique.

1 Environnement de développement :

L'environnement de développement représente l'ensemble des outils, des langages et des technologies utilisés pour concevoir et réaliser notre application intelligente dédiée aux soins de la peau. Cet environnement joue un rôle essentiel car il nous permet de transformer une idée en une application web fonctionnelle, interactive et basée sur l'intelligence artificielle.

Dans notre projet, nous avons utilisé plusieurs outils complémentaires qui travaillent ensemble pour assurer le bon fonctionnement du système.

1.1 Le langage de programmation : Python

Python est un langage de programmation de haut niveau, interprété et polyvalent, largement reconnu aujourd'hui comme l'un des outils les plus importants dans le domaine de l'informatique moderne. Il est utilisé dans plusieurs secteurs tels que le développement web, l'intelligence artificielle, l'analyse de données, ainsi que le traitement automatique du langage naturel.

Créé au début des années 1990 par Guido van Rossum aux Pays-Bas, Python tire son nom de la célèbre émission humoristique “Monty Python”, ce qui reflète bien l’esprit du langage basé sur la simplicité et l’accessibilité. Depuis sa création, Python a été développé comme un projet open source et est aujourd’hui maintenu par la Python Software Foundation, ce qui a favorisé son évolution rapide et son adoption à grande échelle dans le monde entier.

Ce langage se distingue principalement par sa simplicité d’écriture et sa lisibilité, ce qui le rend particulièrement adapté aux débutants, tout en étant suffisamment puissant pour le développement de systèmes complexes. Python est également un langage multi-paradigme, supportant la programmation orientée objet, la programmation fonctionnelle ainsi que la programmation impérative.

Dans le domaine professionnel, Python occupe une place très importante et est utilisé par de grandes entreprises technologiques telles que Google, ce qui confirme sa fiabilité et sa performance. Sa popularité est également mesurée par des indices comme le TIOBE Index, qui le classe parmi les langages les plus utilisés et les plus recherchés dans le monde.

Pour exécuter un programme Python, il est possible d’utiliser un interpréteur interactif permettant de tester rapidement des instructions, ou bien de créer des scripts complets exécutés via la ligne de commande. Cette flexibilité rend Python adapté aussi bien aux petits tests qu’aux grands projets logiciels.

Python dispose également d’un large écosystème d’environnements de développement (IDE) tels que IDLE, PyCharm, Eclipse avec PyDev ou encore Emacs, qui facilitent l’écriture, le débogage et la gestion du code.

Dans le cadre de notre application, Python constitue le langage principal de développement. Il a été choisi pour sa simplicité, sa puissance et surtout son efficacité dans les domaines de l’intelligence artificielle et du machine learning. Nous avons utilisé Python pour développer l’ensemble des fonctionnalités de notre projet, notamment la logique de l’application, le système de recommandation, le module d’analyse de la peau basé sur l’intelligence artificielle, ainsi que le chatbot intelligent.

Ainsi, Python représente la base fondamentale de notre système, puisque l’ensemble du code de l’application est développé dans ce langage, ce qui garantit à la fois flexibilité, performance et évolutivité du projet.

1.2 Le framework d’interface : Streamlit

- Streamlit est un framework open-source basé sur le langage Python, utilisé pour la création rapide d’interfaces web interactives sans avoir besoin de recourir aux technologies classiques du développement web telles que HTML, CSS ou JavaScript. Il est particulièrement adapté aux projets basés sur la data science et l’intelligence artificielle, car il permet de transformer facilement un script Python en application web fonctionnelle.

- Dans le cadre de notre projet, Streamlit a été utilisé comme la couche Frontend de l'application. Il joue un rôle essentiel dans la présentation et l'interaction avec l'utilisateur, tout en restant directement connecté au Backend développé en Python.
- Grâce à Streamlit, nous avons pu concevoir une interface simple, claire et interactive permettant à l'utilisateur d'interagir facilement avec le système. Il est utilisé pour l'affichage des différents éléments graphiques de l'application, notamment les boutons, les champs de saisie et les zones de résultats.
- Streamlit permet également la communication en temps réel entre l'utilisateur et le modèle d'intelligence artificielle. En effet, chaque action effectuée sur l'interface (comme la saisie d'une information ou le clic sur un bouton) déclenche automatiquement un traitement côté Python, puis affiche immédiatement le résultat à l'écran. Cela rend l'application plus dynamique et plus fluide.
- Dans notre système, Streamlit est également utilisé pour afficher les résultats générés par le module d'analyse de la peau ainsi que les recommandations personnalisées. Il assure aussi l'intégration du chatbot, ce qui permet à l'utilisateur d'interagir directement avec le système de manière intuitive.
- Ainsi, Streamlit représente la partie visible de l'application (Frontend), tandis que toute la logique intelligente et le traitement des données sont assurés par Python dans le Backend. Cette séparation permet d'obtenir une architecture simple, efficace et facile à maintenir.
- Rôle dans notre application : Streamlit est responsable de toute l'interface graphique de l'application.

1.3 La bibliothèque de gestion des données : Pandas

Pandas est une bibliothèque open-source du langage Python, largement utilisée dans les domaines de la data science, de l'intelligence artificielle et du traitement des données. Elle permet de manipuler, structurer et analyser des données sous forme de tableaux appelés DataFrames, ce qui facilite grandement le travail sur des données complexes.

Dans notre projet, Pandas a été utilisée comme un outil essentiel pour la gestion et le traitement des données. Elle nous a permis de structurer les informations collectées à partir de l'utilisateur ainsi que les résultats générés par les différents modules du système, notamment le module d'analyse et le système de recommandation.

Grâce à Pandas, nous avons pu organiser les données de manière claire et efficace, ce qui facilite leur exploitation par les algorithmes d'intelligence artificielle développés en Python. Elle a également joué un rôle important dans le nettoyage des données, la suppression des valeurs inutiles et la transformation des informations afin de les rendre compatibles avec les modèles utilisés.

En outre, Pandas a été utilisée dans la phase de prétraitement des données, qui constitue une étape essentielle dans tout système basé sur l'intelligence artificielle. Cette étape permet d'améliorer la qualité des résultats en garantissant que les données soient bien structurées et cohérentes avant leur traitement.

• Rôle dans notre application :

Dans notre application, Pandas joue un rôle central dans la gestion des données. Elle permet de structurer les informations utilisateur, de les organiser sous forme de tableaux, et de les préparer pour les différents modules du système. Grâce à Pandas, le traitement des données devient plus rapide, plus clair et plus fiable, ce qui contribue directement à la performance globale de l'application.

1.4 La bibliothèque de calcul numérique : NumPy

NumPy est une bibliothèque fondamentale du langage Python, largement utilisée dans les domaines du machine learning, de la data science et du calcul scientifique. Elle fournit une structure de données essentielle appelée "array multidimensionnel", qui permet de manipuler efficacement de grandes quantités de données numériques.

Dans notre projet, NumPy a été utilisée comme un outil central pour effectuer les opérations mathématiques et numériques nécessaires au traitement des données. Elle permet de représenter les données sous forme de vecteurs et de matrices, ce qui facilite leur manipulation par les algorithmes d'intelligence artificielle utilisés dans notre système.

Grâce à NumPy, les calculs sont réalisés de manière rapide et optimisée, ce qui améliore les performances globales de l'application. Elle est particulièrement utile dans les étapes de prétraitement des données ainsi que dans les calculs liés au système de recommandation et aux modèles d'analyse intelligente.

Dans les applications modernes de machine learning et de deep learning, NumPy constitue également une base importante sur laquelle reposent plusieurs autres bibliothèques comme Scikit-learn. Ces bibliothèques utilisent ses structures pour effectuer des traitements avancés sur les données.

Bien que NumPy soit très performant pour les calculs sur CPU, certaines applications plus avancées nécessitant des performances élevées utilisent des extensions compatibles avec GPU afin d'accélérer encore davantage les traitements. Cependant, dans notre projet, NumPy reste suffisante pour assurer les opérations numériques essentielles de manière stable et efficace.

• Rôle dans notre application :

Dans notre application, NumPy est utilisée pour la gestion des calculs numériques et la manipulation des données sous forme de tableaux. Elle joue un rôle important dans le traitement des données d'entrée et dans les calculs nécessaires aux modules d'intelligence artificielle. Grâce à NumPy, le système bénéficie d'une meilleure performance, d'une exécution plus rapide et d'une structure de données adaptée aux algorithmes utilisés.

1.5 Le framework d'intelligence artificielle : TensorFlow / Keras

TensorFlow et Keras sont des frameworks open-source développés pour la conception, l'entraînement et le déploiement de modèles d'intelligence artificielle et de machine learning. Ils sont aujourd'hui parmi les outils les plus utilisés dans le domaine de l'apprentissage profond (deep learning), grâce à leur puissance, leur flexibilité et leur facilité d'utilisation.

Keras est une interface de haut niveau intégrée à TensorFlow, qui permet de construire des modèles de réseaux de neurones de manière simple et rapide, sans avoir à gérer directement les opérations complexes de bas niveau. TensorFlow, quant à lui, fournit l'infrastructure principale pour l'exécution des calculs, l'optimisation des modèles et la gestion des données à grande échelle.

Dans notre projet, TensorFlow/Keras ont été utilisés pour développer les modèles d'intelligence artificielle responsables de l'analyse et du traitement des données. Ces modèles permettent de transformer les données d'entrée en résultats intelligents, notamment dans le cadre du système de recommandation et du module d'analyse basé sur l'IA.

Grâce à ces frameworks, il a été possible de concevoir des modèles capables d'apprendre à partir des données et de fournir des prédictions adaptées aux besoins de l'utilisateur. Cela améliore considérablement la précision et la qualité des résultats générés par l'application.

De plus, TensorFlow et Keras offrent une grande compatibilité avec les autres bibliothèques Python telles que NumPy et Pandas, ce qui facilite l'intégration de toutes les étapes du traitement des données, depuis le prétraitement jusqu'à la prédiction finale.

• Rôle dans notre application :

Dans notre application, TensorFlow et Keras constituent le cœur du module d'intelligence artificielle. Ils sont utilisés pour construire et exécuter les modèles de machine learning qui permettent l'analyse des données et la génération de recommandations intelligentes. Grâce à eux, le système est capable de fournir des résultats précis, automatisés et adaptés à chaque utilisateur.

1.6 La bibliothèque de prétraitement : Pickle

Pickle est une bibliothèque standard du langage Python utilisée pour la sérialisation et la désérialisation des objets. Elle permet de sauvegarder des structures de données ou des modèles déjà entraînés sous forme de fichiers, puis de les recharger facilement lors de l'exécution de l'application, sans avoir à répéter tout le processus de calcul ou d'entraînement.

Dans les projets d'intelligence artificielle et de machine learning, Pickle joue un rôle très important, car il permet de conserver les modèles après leur phase d'apprentissage. Une fois le modèle entraîné, il peut être enregistré dans un fichier et réutilisé directement dans le système, ce qui améliore considérablement la rapidité et l'efficacité de l'application.

Dans notre projet, Pickle a été utilisé pour sauvegarder les modèles d'intelligence artificielle ainsi que certains objets nécessaires au fonctionnement du système. Cela permet de charger rapidement ces éléments au moment de l'exécution de l'application, notamment dans les modules de recommandation et d'analyse intelligente.

Grâce à Pickle, nous évitons de refaire les calculs ou l'entraînement à chaque utilisation, ce qui réduit le temps de traitement et améliore les performances globales du système. Il facilite également le déploiement de l'application en permettant de réutiliser facilement les modèles déjà préparés.

• **Rôle dans notre application :**

Dans notre application, Pickle est utilisé pour sauvegarder et charger les modèles d'intelligence artificielle déjà entraînés. Il permet de stocker les résultats du training et de les réutiliser directement lors de l'exécution du système, garantissant ainsi un fonctionnement rapide, efficace et optimisé sans avoir besoin de réentraîner les modèles à chaque utilisation.

1.7 Le module de réponses aléatoires : Random

Le module Random est une bibliothèque intégrée en Python qui permet de générer des choix aléatoires à partir d'une liste d'éléments.

Dans notre chatbot, ce module est utilisé pour améliorer la qualité des réponses et rendre les interactions plus naturelles. En effet, au lieu de fournir toujours la même réponse pour une question identique, le système sélectionne aléatoirement une réponse parmi plusieurs propositions disponibles.

Cela permet de :

- varier les réponses données à l'utilisateur,
- éviter la répétition des mêmes phrases,
- rendre les conversations plus fluides et plus proches d'un échange humain réel.

• **Rôle dans notre application :**

Le module Random améliore l'expérience utilisateur du chatbot en rendant les réponses dynamiques, diversifiées et plus naturelles, ce qui augmente l'interactivité de l'application.

1.8 Bibliothèque de prétraitement : Scikit-learn (LabelEncoder)

Scikit-learn est une bibliothèque qui fournit des outils pour le machine learning. Nous avons utilisé spécifiquement la classe LabelEncoder.

Version utilisée : Scikit-learn 1.3 ou supérieure.

- **Rôle dans notre application :** LabelEncoder convertit les données textuelles (comme "oily", "dry") en nombres (0,1,2) car le réseau de neurones ne comprend que les nombres.

1.9 Éditeur de code : Visual Studio Code (VS Code)

Visual Studio Code est un éditeur de code gratuit développé par Microsoft.

- **Rôle dans notre application** : Nous avons écrit tout notre code (app.py, train_nn.py) dans VS Code.

1.10 Gestionnaire de versions : flutter

1. Définition de Flutter

Flutter est un framework open source développé par **Google** permettant de créer des applications mobiles, web et desktop à partir d'un seul code source. Il utilise le langage de programmation **Dart** et fournit un ensemble riche de widgets facilitant la conception d'interfaces utilisateur modernes, interactives et performantes.

2. Rôle de Flutter dans le projet

Dans ce projet, Flutter a été utilisé pour le développement de l'interface utilisateur de l'application **AI Beauty Skin Advisor**. Il a permis de concevoir les différentes pages de l'application, notamment l'interface d'accueil, les formulaires de saisie, les écrans d'analyse de la peau et l'affichage des résultats. Grâce à son approche multiplateforme, Flutter facilite le développement et la maintenance de l'application tout en offrant une expérience utilisateur fluide, rapide et cohérente sur les différents appareils mobiles.

1. Tableau récapitulatif des outils (4.2) :

Outil / Technologie	Version	Rôle
Python	3.10+	Langage de programmation principal
Streamlit	1.28+	Framework pour l'interface utilisateur
Pandas	2.0+	Manipulation des fichiers CSV
NumPy	1.24+	Calculs numériques
TensorFlow / Keras	2.15+	Création du réseau de neurones
Scikit-learn	1.3+	LabelEncoder
Pickle	-	Sauvegarde des encodeurs
Random	-	Réponses aléatoires du

		chatbot
Tawk.to	-	Chatbot externe
VS Code	-	Éditeur de code
Flutter	-	Développement de l'interface utilisateur (Frontend).

2 Structure du projet :

Notre projet est organisé selon l'arborescence suivante :

AI_Skincare_Pro/

```

|
|  └─ app.py          # Fichier principal (interface Streamlit)
|
|  └─ train_nn.py     # Script d'entraînement du réseau de neurones
|
|  └─ nn_model.h5      # Modèle Keras sauvegardé
|
|
|  └─ data/
|
|     └─ users (2).csv  # Données des utilisateurs
|
|     └─ products (2).csv  # Catalogue des produits
|
|     └─ interactions (1).csv  # Évaluations utilisateur-produit
|
|
|  └─ models/
|
|     └─ le_skin.pkl    # Encodeur pour les types de peau
|
|     └─ le_problem.pkl  # Encodeur pour les problèmes
|
|     └─ le_solution.pkl  # Encodeur pour les solutions
|
|
|  └─ .streamlit/      # Dossier de configuration

```

|
└─ requirements.txt # Dépendances Python

3 Les fichiers de données : CSV

Dans notre application, nous avons utilisé des fichiers au format CSV (Comma-Separated Values) pour stocker et organiser les données nécessaires au fonctionnement du système.

Un fichier CSV est un fichier texte simple dans lequel les données sont représentées sous forme de tableau, avec des lignes et des colonnes séparées par des virgules. Ce format est très utilisé en data science car il est léger, facile à lire et compatible avec de nombreux outils comme Python et Excel.

Dans notre projet, nous avons utilisé quatre fichiers CSV principaux :

- users.csv : contient les informations sur les utilisateurs (identifiants, âge, type de peau, etc.)
- products.csv : contient la liste des produits cosmétiques (nom, marque, prix, ingrédients)
- interactions.csv : contient l'historique des interactions entre les utilisateurs et les produits (évaluations)
- data.csv : contient les correspondances entre type de peau, problème cutané et solution recommandée (Skin Analysis)

- **Rôle dans notre application :**

Les fichiers CSV représentent la source principale des données de notre système. Ils sont chargés avec la bibliothèque Pandas afin de :

- lire les données sous forme de DataFrame,
- filtrer les informations selon l'utilisateur,
- fusionner les données utilisateurs et produits,
- alimenter le système de recommandation basé sur les interactions,
- entraîner le modèle d'intelligence artificielle (réseau de neurones) avec le fichier data.csv.

Grâce à ces fichiers, notre application peut fonctionner de manière dynamique et proposer des recommandations personnalisées en fonction des données disponibles.

📌 En résumé, les fichiers CSV jouent un rôle essentiel car ils constituent la base de données simple et accessible sur laquelle repose toute la logique de recommandation et d'analyse IA de notre application.

Tableau 4.X : Dataset des utilisateurs (users.csv) :

Tableau 4.1 : Informations générales des utilisateurs :

User_ID	Age	Skin_Type	Skin_Tone	Climate	Diet	Hormonal_Status	Budget_Level
0	43	Sensitive	Medium	Humid	Vegan	Pregnant	High
1	29	Sensitive	Deep	Cold	Vegan	Teen	High
2	22	Dry	Medium	Humid	High_Dairy	Pregnant	High
3	35	Normal	Light	Temperate	Omnivore	Normal	Medium
4	27	Oily	Tan	Humid	Low_Sugar	Normal	High

s

Description :

Ce tableau présente les informations générales des utilisateurs, telles que l'âge, le type de peau, le climat, le régime alimentaire et le statut hormonal. Ces données sont utilisées pour personnaliser les recommandations de soins.

Tableau 4.2 : Indicateurs des problèmes de peau :

User_ID	Acne_Severity	Dryness_Severity	Pigmentation_Severity	Aging_Severity	Sensitivity_Severity
0	0.0	0.0	0.0	2.4	6.34
1	4.54	3.59	4.33	0.0	7.98
2	0.0	8.52	0.0	0.0	0.0
3	2.10	1.50	1.20	3.10	2.50
4	5.20	0.50	2.10	0.80	1.90

Description :

Ce tableau regroupe les indicateurs numériques des différents problèmes de peau. Ces valeurs permettent au système d'évaluer la gravité de chaque condition et d'adapter les recommandations.

3. Dataset des produits (products.csv):

Product_ID	Brand	Price	Ingredients
0	LuxuryGlow	52	Retinol
1	DermaCare	134	Hyaluronic_Acid
2	LuxuryGlow	125	Ceramides Salicylic_Acid Retinol
3	BudgetBeauty	11	Salicylic_Acid
4	ClinicalSkin	22	Hyaluronic_Acid
5	ClinicalSkin	14	Niacinamide

4. Dataset des interactions (interactions.csv):

User_ID	Product_ID	Rating
0	114	2.38
0	63	3.97
0	560	3.1
1	586	4.34
1	549	3.96
2	532	2.89
2	452	2.45

5.Dataset des solutions (Skin Analysis):

skin_type	Problem	solution
Oily	Acne	Use salicylic acid cleanser
Dry	Sensitive	Use gentle moisturizer
Normal	Blackheads	Use clay mask weekly
Oily	Blackheads	Use exfoliating toner
Dry	Acne	Use hydrating cleanser
Normal	Sensitive	Use soothing cream

Oily	Sensitive	Use oil-free moisturizer
Dry	Blackheads	Use gentle exfoliating mask
Normal	Acne	Use mild cleanser
Oily	Acne	Use benzoyl peroxide treatment

6 Architecture de l'application :

L'architecture de notre application a été conçue de manière simple, claire et efficace afin de garantir une bonne organisation du code ainsi qu'une interaction fluide avec l'utilisateur. Elle repose principalement sur une application web développée avec Streamlit, intégrant des modules d'intelligence artificielle et de traitement de données.

De manière générale, notre système est structuré en trois parties principales : l'interface utilisateur (Frontend), la logique de traitement (Backend) et les données ainsi que le modèle d'intelligence artificielle.

L'utilisateur interagit directement avec l'interface web à travers différents composants comme les champs de saisie, les boutons et le chatbot. Une fois les données saisies, celles-ci sont envoyées vers les fonctions de traitement écrites en Python. Ensuite, les résultats sont calculés et affichés en temps réel sur l'interface, ce qui rend l'application dynamique et interactive.

6.1 Interface utilisateur (Frontend) :

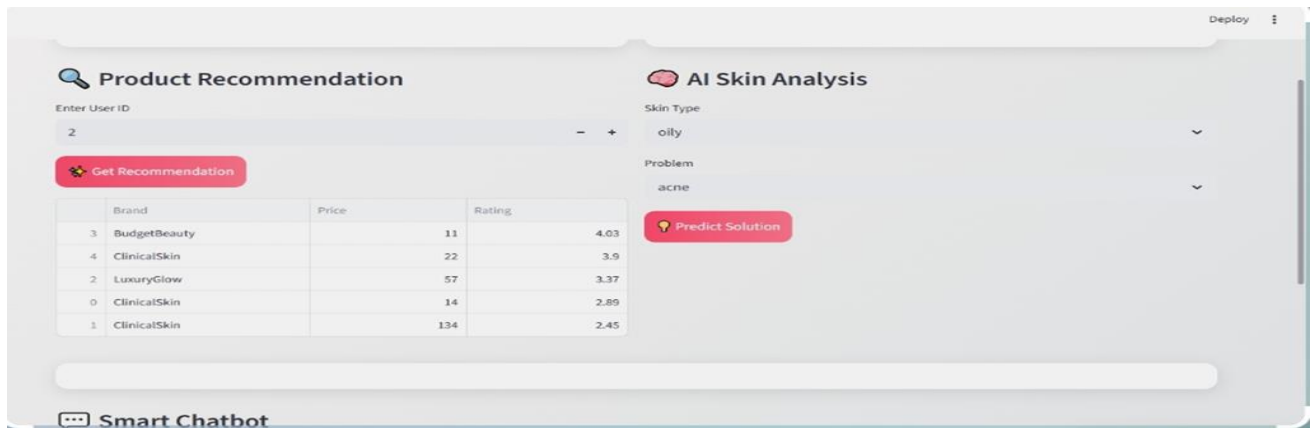
L'interface utilisateur a été développée à l'aide de Streamlit, qui permet de créer facilement des applications web interactives sans avoir besoin de technologies complexes comme HTML ou JavaScript.

Cette interface est organisée en plusieurs sections principales :

Tout d'abord, nous avons le module de recommandation de produits. L'utilisateur peut saisir son identifiant (User ID), puis cliquer sur un bouton pour obtenir une liste de produits recommandés. Les résultats sont affichés sous forme de tableau contenant les informations principales comme la marque, le prix et la note.

Ensuite, nous avons le module d'analyse intelligente de la peau. L'utilisateur sélectionne son type de peau (grasse, sèche ou normale) ainsi que le problème rencontré (acné, points noirs ou sensibilité). Après validation, l'application propose une solution adaptée basée sur une logique intelligente.

Enfin, l'application contient un chatbot interactif. Ce chatbot permet à l'utilisateur de poser



des questions liées aux soins de la peau. Il fournit des réponses automatiques et variées, ce qui améliore l'expérience utilisateur et rend l'application plus vivante.

□ **Figure 4.1 : Interface du système de recommandation et analyse IA**

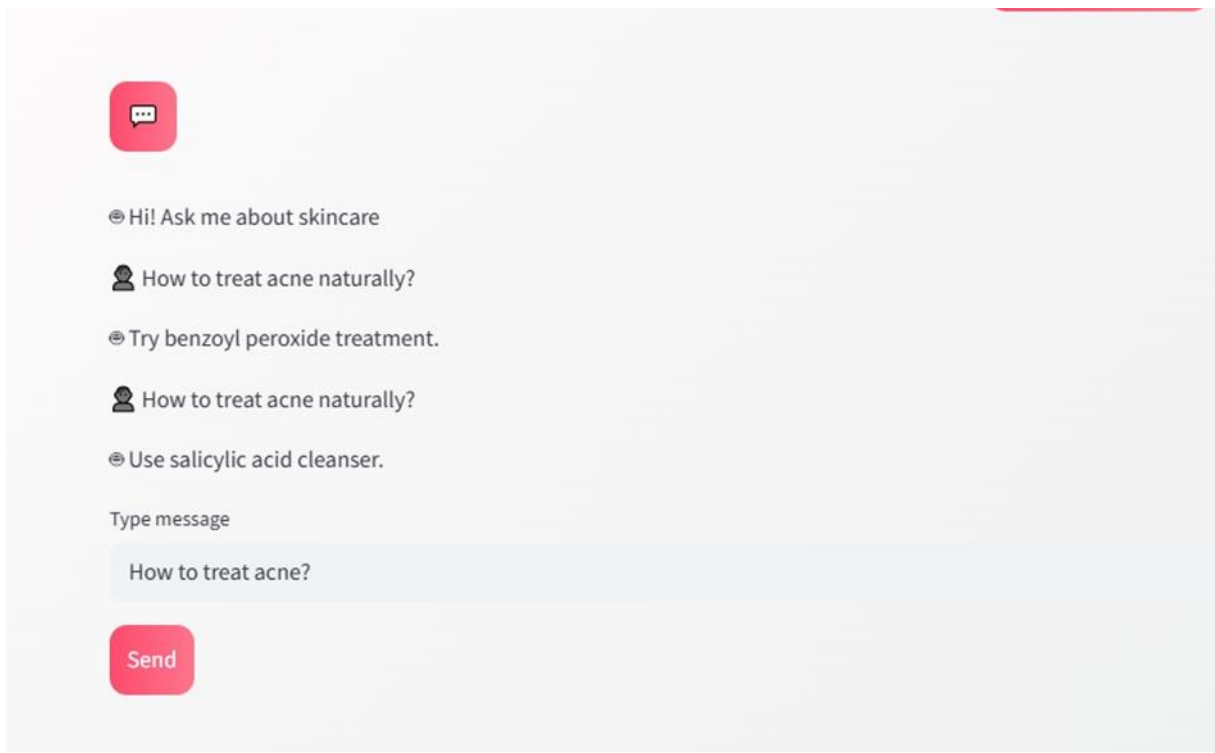


Figure 4.2 : Interface du chatbot intelligent

6.2 Logique de traitement (Backend)

- Le backend constitue le cœur de l'application. Il est responsable du traitement des données et de l'exécution des différentes fonctionnalités.
- Dans notre projet, le backend est entièrement développé en Python. Il contient plusieurs fonctions principales :

- La première fonction concerne le système de recommandation. Elle utilise les données des utilisateurs ainsi que leurs interactions avec les produits afin de proposer les meilleures recommandations. Cette fonction repose sur des opérations de filtrage, de fusion et de tri des données.
- La deuxième fonction est liée à l'intelligence artificielle. Elle permet d'analyser les informations fournies par l'utilisateur (type de peau et problème) afin de proposer une solution adaptée. Cette partie représente le côté intelligent de l'application.
- La troisième fonction correspond au chatbot. Elle analyse les questions posées par l'utilisateur et génère des réponses en se basant sur des mots-clés liés au domaine des soins de la peau.
- Grâce à cette organisation, le backend assure un traitement rapide et efficace des demandes.

6.3 Données et modèle d'intelligence artificielle

- L'application repose sur plusieurs sources de données stockées sous forme de fichiers CSV, notamment :
 - les données des utilisateurs
 - les informations des produits
 - les interactions entre utilisateurs et produits
- Ces données sont utilisées pour alimenter le système de recommandation.
- En parallèle, un modèle d'intelligence artificielle a été développé et sauvegardé sous forme de fichier. Ce modèle permet d'analyser les caractéristiques de la peau et de proposer des solutions adaptées.
- Des fichiers supplémentaires (comme les encodeurs) sont également utilisés pour transformer les données textuelles en valeurs numériques compréhensibles par le modèle.

6.4 Fonctionnement global de l'application

- Le fonctionnement global de l'application peut être résumé en عدة étapes simples :
- Dans un premier temps, l'utilisateur interagit avec l'interface en saisissant ses informations ou en posant une question.
- Ensuite, ces informations sont envoyées vers le backend où elles sont traitées à l'aide des fonctions Python et du modèle d'intelligence artificielle.
- Après traitement, les résultats sont générés puis affichés directement sur l'interface sous forme de recommandations, de solutions ou de réponses du chatbot.
- Ce processus se fait en temps réel, ce qui permet d'offrir une expérience utilisateur rapide et fluide.
- Comme illustré dans la figure ci-dessus, le projet est organisé en plusieurs composants essentiels. Le fichier **app.py** représente le point d'entrée principal de l'application, tandis que les fichiers CSV contiennent les données nécessaires au fonctionnement du système.
- Les fichiers de type **.pkl** sont utilisés pour charger les encodeurs déjà entraînés, ce qui permet d'assurer la cohérence entre la phase d'entraînement et la phase d'utilisation du modèle.
- Cette structure rend l'application plus claire, plus organisée et plus facile à gérer.

7 Organisation du projet et des fichiers :

- Afin d'assurer une bonne structuration du code et de faciliter le développement, notre application est organisée sous forme de plusieurs fichiers et dossiers, chacun ayant un rôle spécifique.
- Le fichier principal de l'application est **app.py**, qui contient l'ensemble du code nécessaire à l'exécution de l'interface Streamlit ainsi que les différentes fonctionnalités (recommandation, analyse IA et chatbot).
- En plus de cela, nous utilisons plusieurs fichiers de données au format CSV, tels que les fichiers des utilisateurs, des produits et des interactions. Ces fichiers sont essentiels pour alimenter le système de recommandation.
- Par ailleurs, certains fichiers spécifiques sont utilisés pour le modèle d'intelligence artificielle, notamment les fichiers **.pkl**, qui contiennent les encodeurs nécessaires à la transformation des données.
- Cette organisation permet de séparer clairement les différentes composantes de l'application et facilite sa maintenance et son évolution

7.1 Fonctionnement global :

- Dans cette section, nous décrivons le fonctionnement interne de l'application. À chaque interaction utilisateur, le système suit trois étapes principales :
- La saisie de l'utilisateur (input)
- Le traitement par le système
- L'affichage du résultat (output)
- Ces étapes sont appliquées à chacune des trois fonctionnalités principales de l'application : la recommandation, l'analyse intelligente (IA) et le chatbot.

7.2 Étape 1 : Saisie de l'utilisateur (User Input) :

L'utilisateur ouvre l'application et choisit ce qu'il veut faire. Selon la fonctionnalité qu'il utilise, il fait des choses différentes.

- Pour la recommandation : L'utilisateur tape son numéro (User ID) dans le champ "Enter User ID", puis clique sur le bouton "Get Recommendation".
- Pour l'analyse IA : L'utilisateur choisit son type de peau (oily, dry, normal) et son problème (acne, blackheads, sensitive) dans des listes déroulantes, puis clique sur "Predict Solution".
- Pour le chatbot : L'utilisateur écrit sa question (par exemple "What causes acne?") dans la zone de chat, puis clique sur "Send".

Résumé de l'étape 1 :

Fonctionnalité	Ce que fait l'utilisateur
Recommandation	Tape son User ID + clique sur "Get Recommendation"

Analyse IA	Choisit Skin Type + Problem + clique sur "Predict Solution"
Chatbot	Écrit une question + clique sur "Send"

7.3 Étape 2 : Traitement par le système :

Après que l'utilisateur a cliqué, l'ordinateur se met à travailler. Chaque bouton déclenche une fonction différente.

◆ Pour la recommandation (recommend(user_id)) :

L'ordinateur fait les choses suivantes :

1. Il lit le fichier "interactions.csv" qui contient toutes les notes des utilisateurs.
2. Il garde seulement les lignes qui concernent l'utilisateur qui a tapé son ID.
3. Il va chercher dans le fichier "products.csv" les informations de chaque produit (marque, prix).
4. Il classe les produits par note (la plus haute en premier).
5. Il garde seulement les 5 meilleurs produits.

◆ Pour l'analyse IA (predict_solution(skin, problem)) :

L'ordinateur regarde ce que l'utilisateur a choisi et applique des règles simples :

- Si peau grasse + acné → "Use salicylic acid cleanser"
- Si peau sèche + acné → "Use gentle cleanser + moisturizer"
- Si peau normale + points noirs → "Use exfoliating scrub"
- Si peau grasse + points noirs → "Use clay mask"
- Si peau sensible → "Use soothing products like aloe vera"
- Sinon → "Use sunscreen + basic skincare routine"

◆ Pour le chatbot (fallback_answer(question)) :

L'ordinateur analyse la question posée :

1. Il transforme la question en minuscules (pour mieux comparer).
2. Il cherche si la question contient des mots-clés comme "acne", "routine", "sunscreen", etc.
3. Si la question ne contient aucun mot-clé lié aux soins de la peau, le chatbot répond qu'il ne peut pas répondre.
4. Si la question contient des mots-clés, le chatbot choisit une réponse au hasard parmi plusieurs réponses possibles.

Résumé de l'étape 2 :

Fonctionnalité	Fonction appelée	Ce qu'elle fait
----------------	------------------	-----------------

Recommandation	recommend(user_id)	Lit les fichiers, filtre, fusionne, trie, retourne les 5 meilleurs produits
Analyse IA	predict_solution(skin, problem)	Teste les conditions (if/else) et retourne la solution
Chatbot	fallback_answer(question)	Analyse les mots-clés, choisit une réponse au hasard

7.4 Étape 3 : Affichage du résultat :

Après le traitement, l'ordinateur affiche le résultat à l'écran.

- Pour la recommandation : L'ordinateur affiche un tableau avec trois colonnes : la marque (Brand), le prix (Price) et la note (Rating). L'utilisateur voit les 5 produits qu'il a le mieux notés.
- Pour l'analyse IA : L'ordinateur affiche un message vert avec la solution recommandée. Par exemple : "Use salicylic acid cleanser".
- Pour le chatbot : L'ordinateur affiche la question de l'utilisateur avec une icône de visage (👤) et la réponse du chatbot avec une icône de robot (🤖). Tous les messages restent affichés pour que l'utilisateur puisse voir toute la conversation.

Résumé de l'étape 3 :

Fonctionnalité	Type d'affichage	Exemple
Recommandation	Tableau	Tableau avec Brand, Price, Rating
Analyse IA	Message vert	"Use salicylic acid cleanser"
Chatbot	Zone de chat	"👤 question" et "🤖 réponse"

8 Schéma global du fonctionnement :

Ce tableau résume les trois étapes pour chaque fonctionnalité :

Étape	Recommandation	Analyse IA	Chatbot
1. Input	Saisir User ID + cliquer sur "Get Recommendation"	Choisir Skin Type + Problem + cliquer sur "Predict Solution"	Écrire une question + cliquer sur "Send"
2. Traitement	recommend(user_id) lit, filtre, fusionne, trie	predict_solution(skin, problem) teste les conditions	fallback_answer(question) analyse les mots-clés
3. Output	Tableau (Brand, Price, Rating)	Message vert avec la solution	Zone de chat avec 👤 et 🤖

Exemple global :

Pour mieux comprendre, voici un exemple simple de chaque fonctionnalité :

- Recommandation : L'utilisateur tape son numéro 2 → le système regarde quels produits il a bien notés → le système affiche ses 5 produits préférés.
- Analyse IA : L'utilisateur choisit "oily" (peau grasse) et "acne" → le système lui dit "Use salicylic acid cleanser".
- Chatbot : L'utilisateur demande "What causes acne?" → le système détecte les mots "cause" et "acne" → le système répond "Acne is caused by clogged pores, excess oil production, bacteria, and dead skin cells."

9 Description des fonctionnalités :

Dans cette section, nous décrivons en détail les trois fonctionnalités principales de notre application. Pour chaque fonctionnalité, nous expliquons comment elle fonctionne, ce que l'utilisateur doit faire, et ce qu'il obtient comme résultat. Des captures d'écran illustrent chaque fonctionnalité.

9.1 Fonctionnalité 1 : Système de recommandation de produits :

9.1.1 Comment ça fonctionne ?

Le système de recommandation aide l'utilisateur à trouver les produits cosmétiques qui pourraient lui plaire. Il utilise une méthode simple mais efficace : il regarde les produits que l'utilisateur a déjà notés dans le passé, et il lui montre ceux qu'il a le mieux notés.

Concrètement, voici ce qui se passe à l'intérieur de l'ordinateur :

- L'application lit un fichier CSV appelé "interactions (1).csv". Ce fichier contient toutes les notes que tous les utilisateurs ont données à tous les produits. Chaque ligne du fichier contient trois informations : l'identifiant de l'utilisateur (User_ID), l'identifiant du produit (Product_ID), et la note (Rating) que l'utilisateur a donnée à ce produit.

L'application lit aussi un fichier CSV appelé "products (2).csv". Ce fichier contient la liste de tous les produits avec leur marque (Brand), leur prix (Price), et leurs ingrédients (Ingredients).

- Quand l'utilisateur saisit son identifiant et clique sur le bouton, l'application cherche dans le fichier des interactions toutes les lignes où l'identifiant de l'utilisateur correspond à celui saisi.

- Ensuite, elle fusionne ces résultats avec le fichier des produits pour obtenir les détails de chaque produit (marque, prix).

- Ensuite, elle trie les produits par ordre décroissant de la note (les meilleures notes en premier).

- Ensuite, elle ne garde que les 5 premiers produits (les mieux notés).

- Enfin, elle affiche ces 5 produits dans un tableau à l'écran.

9.1.2 Ce que l'utilisateur doit faire (Input) :

L'utilisateur doit suivre deux étapes très simples :

- **Étape 1** : Il tape son numéro d'identifiant (User ID) dans le champ de texte. Ce numéro peut être n'importe quel nombre. Dans notre exemple, l'utilisateur tape le chiffre 2.

- **Étape 2** : Il clique sur le bouton "Get Recommendation" (Obtenir une recommandation).

C'est tout. L'utilisateur n'a pas besoin de faire autre chose.

9.1.3 Ce que l'utilisateur obtient (Output) :

Après avoir cliqué sur le bouton, l'utilisateur voit apparaître un tableau avec trois colonnes :

Brand : La marque du produit (par exemple "BudgetBeauty", "ClinicalSkin", "LuxuryGlow")

- Price : Le prix du produit en euros- Rating : La note que l'utilisateur a donnée à ce produit (sur une échelle de 1 à 5)

Le tableau contient exactement 5 lignes (les 5 produits les mieux notés par cet utilisateur).4.4.1.4 Exemple concretPrenons un utilisateur qui a l'identifiant 2. Voici ce qu'il voit après avoir tapé "2" et cliqué sur "Get Recommendation" :

Cet utilisateur peut maintenant voir ses produits préférés et décider lequel acheter.

Exemple concret :

Prenons un utilisateur qui a l'identifiant 2. Voici ce qu'il voit après avoir tapé "2" et cliqué sur "Get Recommendation" :

Brand	Price	Rating
BudgetBeauty	11	4.03
ClinicalSkin	22	3.9
LuxuryGlow	57	3.37
ClinicalSkin	14	2.89
ClinicalSkin	134	2.45

Cet utilisateur peut maintenant voir ses produits préférés et décider lequel acheter.

4.4.1.5 Capture d'écran :

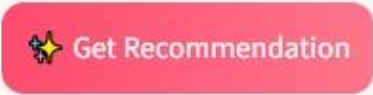


Product Recommendation

Enter User ID

2

- +



	Brand	Price	Rating
3	BudgetBeauty	11	4.03
4	ClinicalSkin	22	3.9
2	LuxuryGlow	57	3.37
0	ClinicalSkin	14	2.89
1	ClinicalSkin	134	2.45

Figure 4.1 : Interface du système de recommandation

Cette capture montre le champ de saisie de l'User ID (avec la valeur 2), le bouton "Get Recommendation", et le tableau des résultats avec les colonnes Brand, Price, et Rating

9.2 Fonctionnalité 2 : Analyse intelligente de la peau (AI Skin Analysis)

9.2.1 Comment ça fonctionne ?

L'analyse intelligente de la peau aide l'utilisateur à savoir quel produit utiliser en fonction de son type de peau et de son problème cutané.

L'application utilise une logique simple basée sur des règles "si... alors..." (if/else). Voici comment cela fonctionne :

L'application propose deux menus déroulants à l'utilisateur : un pour le type de peau, et un pour le problème.

L'utilisateur choisit son type de peau parmi trois options : "oily" (peau grasse), "dry" (peau sèche), ou "normal" (peau normale).

- L'utilisateur choisit son problème parmi trois options : "acne" (acné), "blackheads" (points noirs), ou "sensitive" (peau sensible).

- Quand l'utilisateur clique sur le bouton "Predict Solution", l'application vérifie la combinaison choisie et retourne la solution correspondante.

Voici la logique complète :

Si l'utilisateur a la peau grasse (oily) ET qu'il a de l'acné (acne), alors la solution est : "Use salicylic acid cleanser" (Utilise un nettoyant à l'acide salicylique).

- Sinon, si l'utilisateur a la peau sèche (dry) ET qu'il a de l'acné (acne), alors la solution est : "Use gentle cleanser + moisturizer" (Utilise un nettoyant doux et une crème hydratante).

Sinon, si l'utilisateur a la peau normale (normal) ET qu'il a des points noirs (blackheads), alors la solution est : "Use exfoliating scrub" (Utilise un gommage exfoliant).

- Sinon, si l'utilisateur a la peau grasse (oily) ET qu'il a des points noirs (blackheads), alors la solution est : "Use clay mask" (Utilise un masque d'argile).

- Sinon, si l'utilisateur a la peau sensible (sensitive) (quel que soit le type de peau), alors la solution est : "Use soothing products like aloe vera" (Utilise des produits apaisants comme l'aloe vera).

Sinon (pour tous les autres cas non couverts), la solution par défaut est : "Use sunscreen + basic skincare routine" (Utilise une crème solaire et une routine de soin de base).

9.2.2 Ce que l'utilisateur doit faire (Input)

L'utilisateur doit suivre trois étapes très simples :

- **Étape 1** : Dans le premier menu déroulant (Skin Type), il choisit son type de peau. Les options sont : "oily", "dry", ou "normal".

- **Étape 2** : Dans le deuxième menu déroulant (Problem), il choisit son problème cutané. Les options sont : "acne", "blackheads", ou "sensitive".

- **Étape 3** : Il clique sur le bouton "Predict Solution" (Prédire la solution).

9.2.3 Ce que l'utilisateur obtient (Output) :

Après avoir cliqué sur le bouton, l'utilisateur voit apparaître un message de couleur verte. Ce message contient la solution recommandée en anglais.

Par exemple :

- Si l'utilisateur a choisi "oily" et "acne", le message affiche : "Use salicylic acid cleanser"

- Si l'utilisateur a choisi "dry" et "acne", le message affiche : "Use gentle cleanser + moisturizer"

- Si l'utilisateur a choisi "normal" et "blackheads", le message affiche : "Use exfoliating scrub"

Exemple concret :

Prenons un utilisateur qui a la peau grasse (oily) et qui souffre d'acné (acne).

- Il choisit "oily" dans le menu Skin Type.

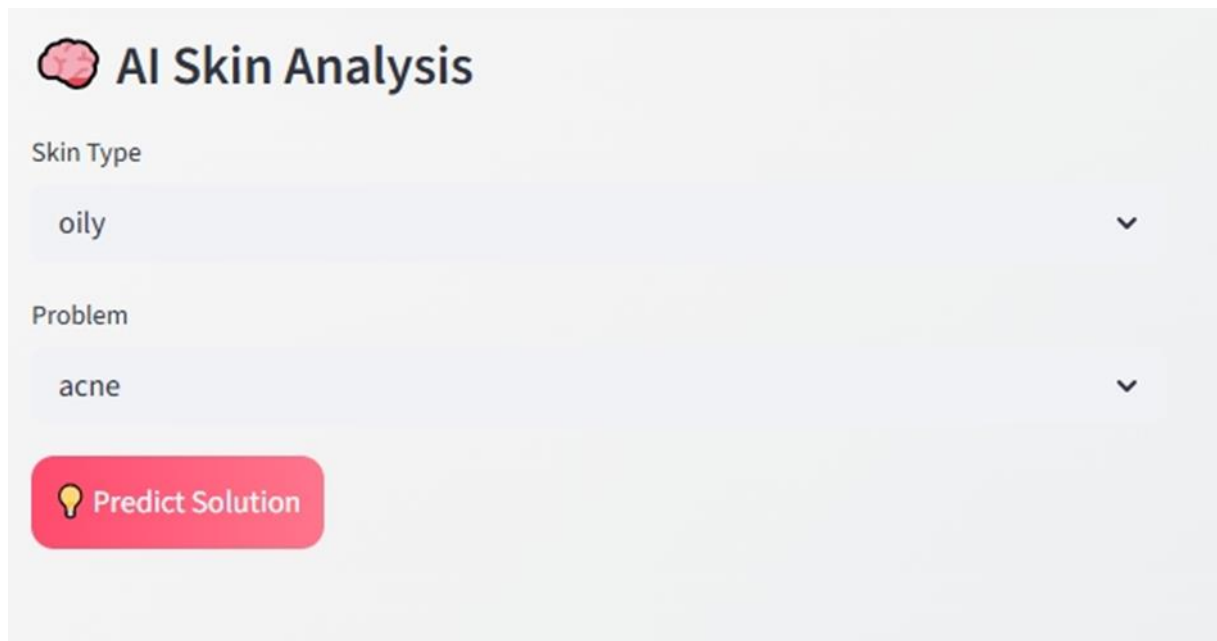
- Il choisit "acne" dans le menu Problem.

- Il clique sur "Predict Solution".

affiche alors un message vert :

"Use salicylic acid cleanser"

L'utilisateur sait maintenant qu'il doit utiliser un nettoyant à l'acide salicylique.



La figure 4.2 : ci-dessous montre l'interface de l'analyse IA.

Cette capture montre les deux menus déroulants (Skin Type et Problem) avec les valeurs "oily" et "acne" sélectionnées, ainsi que le bouton "Predict Solution". Le résultat "Use salicylic acid cleanser" apparaît dans un message vert.

9.3 Fonctionnalité 3 : Chatbot intelligent :

9.3.1 Comment ça fonctionne ?

Le chatbot intelligent permet à l'utilisateur de poser des questions sur les soins de la peau et d'obtenir des réponses automatiques.

Notre application contient en fait deux chatbots :

- Un chatbot externe (Tawk.to) : C'est un service en ligne gratuit. Il ajoute une fenêtre de chat flottante sur l'application. Cette fenêtre permet aux utilisateurs de discuter avec un opérateur humain en temps réel.
- Un chatbot interne (fallback) : C'est un chatbot automatique qui répond aux questions sans intervention humaine. Il utilise une logique basée sur des règles et des réponses aléatoires.

Nous allons détailler ici le chatbot interne (fallback), car c'est lui que nous avons entièrement développé.

Voici comment fonctionne le chatbot interne, étape par étape :

Étape 1 : L'utilisateur tape une question dans le champ de saisie et clique sur "Send".

Étape 2 : L'ordinateur convertit la question en minuscules. Par exemple, "What causes acne?" devient "what causes acne?". Cela permet de comparer les mots plus facilement.

Étape 3 : L'ordinateur vérifie si la question contient des mots-clés liés aux soins de la peau. La liste des mots-clés est : "skin", "acne", "face", "beauty", "pimple", "skincare", "oily", "dry", "cause", "why", "sunscreen", "glowing", "ingredients", "routine", "moisturizer", "cleanse".

Si la question ne contient AUCUN de ces mots, le chatbot répond : " ✕ Sorry, I can only answer skincare and beauty-related questions." (Désolé, je ne peux répondre qu'aux questions sur les soins de la peau.)

- Étape 4 : Si la question contient des mots-clés, l'ordinateur cherche quel type de question c'est. Il regarde si la question contient certains mots spécifiques.

- Étape 5 : Selon le type de question, l'ordinateur choisit une réponse au hasard parmi plusieurs variantes (généralement 3 variantes). Par exemple, pour une question sur les causes de l'acné, il y a 3 réponses possibles. L'ordinateur en choisit une au hasard. Cela rend les conversations moins répétitives.

- Étape 6 : L'ordinateur affiche la question de l'utilisateur avec une icône de visage (👤) et la réponse du chatbot avec une icône de robot (🤖). L'historique des messages est conservé, donc l'utilisateur peut voir toute la conversation.

9.3.2 Ce que l'utilisateur doit faire (Input) :

L'utilisateur doit suivre trois étapes très simples :

- Étape 1 : Il clique sur le bouton "Open Chat" (Ouvrir le chat) pour afficher la zone de chat. Ce bouton se trouve en bas de l'interface.

- Étape 2 : Il tape sa question dans le champ de texte. Par exemple, il peut taper "What causes acne?" ou "What is a good skincare routine?".

- Étape 3 : Il clique sur le bouton "Send" (Envoyer) à côté du champ de texte.

Le chatbot peut aussi être utilisé sans ouvrir le chat interne : le chatbot Tawk.to (externe) est toujours disponible sous forme de fenêtre flottante.

9.3.3 Ce que l'utilisateur obtient (Output) :

Après avoir cliqué sur "Send", l'utilisateur voit apparaître dans la zone de chat :

- Un message avec l'icône  contenant sa question.
- Un message avec l'icône  contenant la réponse du chatbot.

L'historique de tous les messages est conservé, donc l'utilisateur peut voir toute la conversation du début à la fin.

Exemples concrets :

Exemple 1 : Question sur les causes de l'acné

L'utilisateur tape : "What causes acne?"

- Le chatbot répond (par exemple) : "Acne is caused by clogged pores, excess oil production, bacteria, and dead skin cells."

Exemple 2 : Question sur la routine de soin

- L'utilisateur tape : "What is a good skincare routine?"

- Le chatbot répond (par exemple) : "A good skincare routine includes: cleansing, moisturizing, and applying sunscreen daily."

Exemple 3 : Question sur l'importance de la crème solaire

- L'utilisateur tape : "Why is sunscreen important?"

- Le chatbot répond (par exemple) : "Sunscreen protects your skin from harmful UV rays, prevents premature aging, and reduces the risk of skin cancer."

Exemple 4 : Question hors sujet

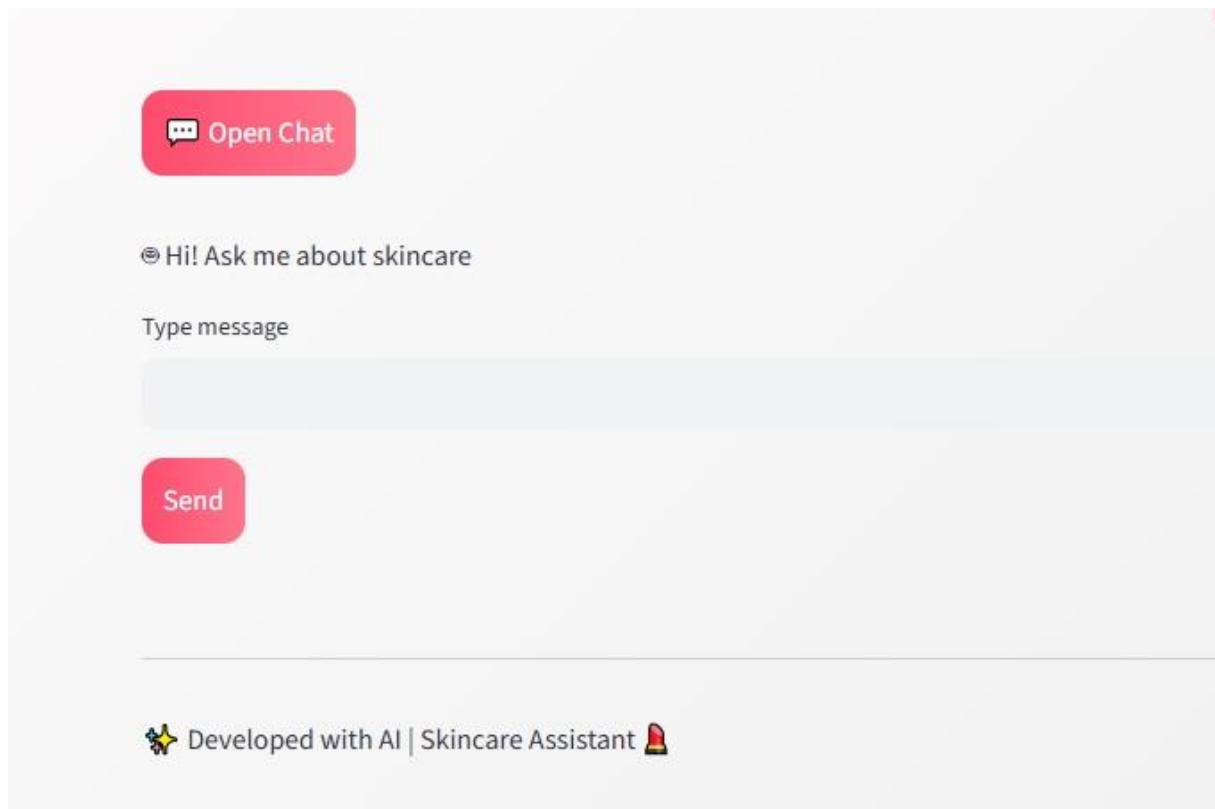
- L'utilisateur tape : "What is the weather today?"

- Le chatbot répond : " ❌ Sorry, I can only answer skincare and beauty-related questions."

Exemple 5 : Question sur la peau qui brille

- L'utilisateur tape : "How to get glowing skin?"

- Le chatbot répond (par exemple) : "To get glowing skin: drink water, sleep well, use vitamin C serum, and exfoliate weekly."



La figure 4 .3 : ci-dessous montre l'interface du chatbot.

Cette capture montre la zone de chat avec l'historique des messages. On voit des questions comme "What causes acne?", "What is a good skincare routine?", "Why is sunscreen important?", et les réponses correspondantes. Le champ de saisie "Type message" et le bouton "Send" sont visibles en bas.

✓ 10 Implémentation du modèle IA :

Dans cette section, nous expliquons comment nous avons construit et utilisé le modèle d'intelligence artificielle pour l'analyse de la peau. Ce modèle est un réseau de neurones artificiels (Neural Network).

10 .1 Qu'est-ce qu'un réseau de neurones ?

Un réseau de neurones est un programme informatique qui imite le fonctionnement du cerveau humain. Il est composé de petites unités appelées "neurones" qui sont connectées entre elles. Le réseau apprend à partir d'exemples. Plus on lui donne d'exemples, mieux il apprend.

Dans notre cas, nous avons donné au réseau 10 exemples. Chaque exemple associe un type de peau et un problème à une solution. Après avoir appris à partir de ces exemples, le réseau peut

prédire la solution pour de nouvelles combinaisons (type de peau, problème) qu'il n'a jamais vues.

4.6.2 Les outils utilisés : TensorFlow et Keras

Pour construire notre réseau de neurones, nous avons utilisé deux bibliothèques Python :

- TensorFlow : C'est une bibliothèque développée par Google. Elle est très puissante pour le machine learning et le deep learning.
- Keras : C'est une interface (API) qui simplifie l'utilisation de TensorFlow. Avec Keras, nous pouvons construire un réseau de neurones en quelques lignes de code.

Nous avons utilisé TensorFlow version 2.15 ou supérieure, et Keras qui est intégré à l'intérieur de TensorFlow.

4.6.3 L'encodage des données : LabelEncoder

Le réseau de neurones ne comprend pas le texte. Il ne comprend que les nombres. Nous devons donc convertir (encoder) les données textuelles en nombres.

Par exemple :

- "oily" (peau grasse) doit devenir un nombre, par exemple 2
- "dry" (peau sèche) doit devenir un nombre, par exemple 0
- "normal" (peau normale) doit devenir un nombre, par exemple 1

Pour faire cette conversion, nous avons utilisé LabelEncoder de la bibliothèque scikit-learn.

Voici comment nous avons procédé :

- Nous avons créé trois encodeurs : un pour les types de peau (le_skin), un pour les problèmes (le_problem), et un pour les solutions (le_solution).
- Nous avons demandé à chaque encodeur d'apprendre la correspondance entre les textes et les nombres.
- Ensuite, nous avons remplacé chaque texte par son nombre correspondant dans nos données. Les encodeurs ont été sauvegardés dans des fichiers pour être réutilisés plus tard. Ces fichiers sont :
 - le_skin.pkl : contient l'encodeur pour les types de peau
 - le_problem.pkl : contient l'encodeur pour les problèmes
 - le_solution.pkl : contient l'encodeur pour les solutions

10.2 L'entraînement du modèle (Training) :

L'entraînement est la phase où le réseau de neurones apprend à partir des exemples.

Voici les étapes de l'entraînement :

Étape 1 : Préparation des données

Nous avons préparé deux ensembles de données :

- X (les entrées) : contient le type de peau et le problème (encodés en nombres)
- y (la sortie) : contient la solution (encodée en nombre)

Étape 2 : Construction du modèle

Nous avons construit un réseau de neurones avec trois couches :

- Couche d'entrée (implicite) : 2 neurones (skin_type et problem)
- Première couche cachée : 8 neurones avec la fonction d'activation ReLU
- Deuxième couche cachée : 4 neurones avec la fonction d'activation ReLU
- Couche de sortie : 1 neurone (la solution prédite)

Étape 3 : Compilation du modèle

Nous avons compilé le modèle avec :

- loss='mse' (Mean Squared Error) : c'est la fonction qui mesure l'erreur du modèle. Plus l'erreur est faible, meilleur est le modèle.
- optimizer='adam' : c'est l'algorithme qui ajuste les poids du réseau pour réduire l'erreur.

Étape 4 : Entraînement

Nous avons entraîné le modèle pendant 200 époques (epochs). Une époque est un passage complet de toutes les données d'entraînement dans le réseau. À chaque époque, le réseau calcule l'erreur et ajuste ses poids pour s'améliorer.

Après 200 époques, le réseau a appris à mapper chaque couple (type de peau, problème) à la solution correspondante.

10.3 Sauvegarde du modèle :

Après l'entraînement, nous avons sauvegardé le modèle dans un fichier appelé nn_model.h5. Ce fichier contient :

- L'architecture du réseau (nombre de couches, nombre de neurones)

- Les poids du réseau (les valeurs que le réseau a apprises)

Nous avons également sauvegardé les trois encodeurs dans des fichiers .pkl.

10.4 La prédiction (Prediction) :

Dans l'application principale, nous chargeons le modèle sauvegardé et les encodeurs. Ensuite, nous utilisons une fonction simplifiée appelée `predict_solution()` pour faire les prédictions.

Cette fonction n'utilise pas encore le modèle de deep learning. Elle utilise une logique basée sur des règles (if/else) parce que c'est plus simple pour la démonstration. Mais le modèle `nn_model.h5` est prêt à être utilisé pour des prédictions plus avancées.

Voici comment fonctionne la fonction `predict_solution()` :

Elle reçoit le type de peau et le problème.

- Elle teste plusieurs conditions "si... alors..." (if/else).
- Elle retourne la solution correspondante.

Par exemple :

- Si peau grasse (oily) ET acné (acne) → retourne "Use salicylic acid cleanser"
- Si peau sèche (dry) ET acné (acne) → retourne "Use gentle cleanser + moisturizer"
- Si peau normale (normal) ET points noirs (blackheads) → retourne "Use exfoliating scrub"

10.5 Intégration du chatbot :

Dans cette section, nous expliquons comment nous avons intégré le chatbot dans notre application. Notre application contient deux chatbots : un chatbot externe (Tawk.to) et un chatbot interne (fallback). Nous détaillons ici le chatbot interne que nous avons entièrement développé.

10.5 .1 Le fonctionnement général du chatbot :

Le chatbot interne est un programme qui répond automatiquement aux questions des utilisateurs. Il ne nécessite pas d'intervention humaine. Il fonctionne selon un principe simple : il analyse la question de l'utilisateur, cherche des mots-clés, et retourne une réponse appropriée.

Voici les étapes que le chatbot suit à chaque question :

1. L'utilisateur tape une question dans le champ de saisie et clique sur "Send".
2. Le chatbot convertit la question en minuscules (pour faciliter la comparaison).

3. Le chatbot vérifie si la question concerne les soins de la peau.
4. Si la question est hors sujet, le chatbot refuse poliment de répondre.
5. Si la question est dans le sujet, le chatbot cherche quel type de question c'est.
6. Le chatbot choisit une réponse au hasard parmi plusieurs variantes.
7. Le chatbot affiche la question et la réponse dans la zone de chat.

10.5.2 La restriction du domaine (seulement les soins de la peau) :

Le chatbot ne doit répondre qu'aux questions sur les soins de la peau. Pour cela, nous avons défini une liste de mots-clés autorisés.

Voici la liste des mots-clés :

["skin", "acne", "face", "beauty", "pimple", "skincare", "oily", "dry", "cause", "why", "sunscreen", "glowing", "ingredients", "routine", "moisturizer", "cleanse"]

Si la question de l'utilisateur ne contient AUCUN de ces mots, le chatbot répond :

" ✕ Sorry, I can only answer skincare and beauty-related questions."

Cela signifie : "Désolé, je ne peux répondre qu'aux questions sur les soins de la peau et la beauté."

Exemple de question hors sujet : "What is the weather today ?" (Quel temps fait-il aujourd'hui ?) Le chatbot refusera de répondre.

10.5 .3 L'analyse des mots-clés (keywords) :

Si la question est dans le sujet, le chatbot cherche des mots-clés spécifiques pour déterminer le type de question.

10.5.4 Les réponses aléatoires (random responses) :

Pour rendre le chatbot moins monotone et plus naturel, nous utilisons le module random de Python. Ce module permet de choisir un élément au hasard dans une liste.

Par exemple, pour une question sur les causes de l'acné, le chatbot a trois réponses possibles :

1. "Acne is caused by clogged pores, excess oil production, bacteria, and dead skin cells."
2. "Hormonal changes, stress, and certain cosmetics can also cause acne."
3. "Acne happens when hair follicles become clogged with oil and dead skin cells."

Le chatbot utilise `random.choice(liste)` pour choisir une de ces trois réponses au hasard. Ainsi, si l'utilisateur pose la même question plusieurs fois, il peut obtenir des réponses différentes.

10.5.5 La fonction `fallback_answer()` :

Toute la logique du chatbot est contenue dans une fonction appelée `fallback_answer()`. Cette fonction :

- Reçoit la question de l'utilisateur en paramètre.
- Convertit la question en minuscules.
- Vérifie la présence de mots-clés.
- Retourne la réponse appropriée.

Voici un résumé de ce que fait la fonction :

```
```python
def fallback_answer(question):

 q = question.lower()

 # 1. Vérification du domaine :

 if not any(mot in q for mot in mots_cles_skincare):

 return "✗ Sorry, I can only answer skincare questions."

 # 2. Détection du type de question:

 if "cause" in q and ("acne" in q or "pimple" in q):

 return random.choice(reponses_causes_acne)

 if "routine" in q:

 return random.choice(reponses_routine)

 if "sunscreen" in q:

 return random.choice(reponses_sunscreen)

 # ... autres conditions ...

 # 3. Réponse par défaut

 return random.choice(reponses_default)
```

### 10.5.6 Le chatbot externe ([Tawk.to](https://tawk.to)) :

En plus du chatbot interne, nous avons intégré un chatbot externe appelé [Tawk.to](https://tawk.to). C'est un service en ligne gratuit qui offre une fenêtre de chat flottante.

Les avantages de [Tawk.to](https://tawk.to) sont :

- Il permet aux utilisateurs de discuter avec un opérateur humain en temps réel.
- Il est facile à intégrer : il suffit d'ajouter un script JavaScript dans la page.
- Il est gratuit pour un usage basique.
- Il offre une interface professionnelle.

L'intégration se fait avec une seule ligne de code dans Streamlit :

```
python
```

```
from streamlit.components.v1 import html
```

```
html(script_tawk_to, height=0)
```

Le script [Tawk.to](https://tawk.to) charge automatiquement la fenêtre de chat qui apparaît en bas à droite de l'écran.

## 11 Tests et résultats :

Dans cette section, nous présentons les tests que nous avons effectués pour vérifier le bon fonctionnement de chaque fonctionnalité de notre application. Nous montrons également les résultats obtenus.

### 11.1 Test du système de recommandation :

#### 4.8.1.1 Comment nous avons testé

Pour tester le système de recommandation, nous avons suivi ces étapes :

- Étape 1 : Nous avons lancé l'application.
- Étape 2 : Nous avons saisi différents User ID dans le champ de saisie.
- Étape 3 : Nous avons cliqué sur le bouton "Get Recommendation".
- Étape 4 : Nous avons vérifié que le tableau des produits s'affichait correctement.

### 11.2 Test du module d'analyse IA (AI Skin Analysis) :

#### 11.2.1 Comment nous avons testé :

Pour tester le module d'analyse IA, nous avons suivi ces étapes :

- Étape 1 : Nous avons lancé l'application.
- Étape 2 : Nous avons sélectionné différentes combinaisons de Skin Type et Problem.
- Étape 3 : Nous avons cliqué sur le bouton "Predict Solution".

- Étape 4 : Nous avons vérifié que la solution affichée correspondait à la combinaison choisie.

### **Exemple de résultat :**

Lorsque nous avons choisi Skin Type = "oily" et Problem = "acne", puis cliqué sur "Predict Solution", le système a affiché un message vert :

"Use salicylic acid cleanser"

Le test est validé.

## **11.3 Test du chatbot :**

### **11.3.1 Comment nous avons testé**

Pour tester le chatbot, nous avons suivi ces étapes :

- Étape 1 : Nous avons lancé l'application.
- Étape 2 : Nous avons cliqué sur "Open Chat" pour ouvrir la zone de chat.
- Étape 3 : Nous avons tapé différentes questions dans le champ de saisie.
- Étape 4 : Nous avons cliqué sur "Send".
- Étape 5 : Nous avons vérifié que la réponse du chatbot était appropriée.

### **Exemple de résultat :**

- Lorsque nous avons tapé la question "What causes acne?" et cliqué sur "Send", le chatbot a répondu (par exemple) :
- "Acne is caused by clogged pores, excess oil production, bacteria, and dead skin cells."
- La question et la réponse sont apparues dans la zone de chat avec les icônes  et .
- Le test est validé.

## **12.1 Récapitulatif de ce qui a été fait :**

Nous avons développé une application web fonctionnelle qui combine trois fonctionnalités principales :

Un système de recommandation de produits cosmétiques. L'utilisateur saisit son identifiant et obtient un tableau des 5 produits qu'il a le mieux notés, avec leur marque, leur prix et leur note.

Un module d'analyse de peau par intelligence artificielle. L'utilisateur choisit son type de peau (grasse, sèche, normale) et son problème (acné, points noirs, peau sensible), et l'application lui propose une solution adaptée.

- Un chatbot intelligent. L'utilisateur peut poser des questions sur les soins de la peau et obtenir des réponses automatiques. Le chatbot refuse poliment les questions hors sujet.

## 12.2 Les technologies utilisées :

Pour réaliser cette application, nous avons utilisé :

- Python comme langage de programmation principal.
- Streamlit pour créer l'interface utilisateur web. Streamlit nous a permis de transformer notre script Python en application web sans avoir à écrire de HTML, CSS ou JavaScript.
- TensorFlow et Keras pour construire et entraîner le réseau de neurones.
- Pandas pour manipuler les fichiers CSV (utilisateurs, produits, interactions).
- Tawk.to pour intégrer un chatbot externe.
- Pickle pour sauvegarder et charger les encodeurs.

## 12.3 Les points forts de l'application :

l'application présente plusieurs qualités :

- Elle est simple à utiliser. L'utilisateur n'a besoin d'aucune connaissance technique. Il suffit de cliquer sur des boutons et de choisir des options dans des menus déroulants.
- Elle est rapide. Les réponses sont affichées en quelques secondes.
- Elle combine trois fonctionnalités utiles dans une seule interface : recommandation, analyse de peau, et chatbot.
- Elle est modulaire. Chaque fonctionnalité est indépendante. On peut améliorer l'une sans affecter les autres.
- Elle est développée avec des technologies modernes et gratuites (Python, Streamlit, TensorFlow, Tawk.to).

## 12.4 Les limites actuelles :

Notre application a aussi quelques limites :

- Le système de recommandation ne fonctionne que pour les utilisateurs qui ont déjà évalué des produits. Un nouvel utilisateur (sans historique) n'obtient aucune recommandation. C'est ce qu'on appelle le problème du "cold start".
- Le module d'analyse IA utilise actuellement une logique basée sur des règles (if/else), pas le vrai réseau de neurones. Le modèle nn\_model.h5 est prêt, mais nous ne l'avons pas encore intégré.
- Le chatbot ne comprend que les questions en anglais. Il ne reconnaît que les mots-clés anglais.

- Les données (utilisateurs, produits, interactions) sont des données de démonstration, pas des données réelles.

## 12.5 Perspectives d'amélioration (futur) :

Notre application peut être améliorée de plusieurs façons :

- Intégrer le vrai réseau de neurones (nn\_model.h5) pour les prédictions, au lieu de la logique if/else.

- Ajouter plus de données d'entraînement pour que le réseau de neurones apprenne mieux et plus de cas.

Ajouter un système de recommandation plus avancé, comme le filtrage collaboratif ou la factorisation matricielle.

Ajouter la possibilité pour l'utilisateur de créer un compte et de sauvegarder son historique.

- Ajouter une base de données (MySQL ou Firebase) au lieu des fichiers CSV.

- Ajouter plus de langues pour le chatbot (français, arabe, etc.).

- Déployer l'application sur le cloud (Streamlit Cloud, Heroku, ou AWS) pour qu'elle soit accessible en ligne.

## Discussion

Dans cette section, nous analysons les résultats obtenus après les tests de notre application.

Les résultats montrent que le système fonctionne correctement et répond aux objectifs fixés. Le système de recommandation fournit des résultats pertinents en se basant sur les interactions des utilisateurs. Le module d'analyse IA donne des solutions cohérentes pour les différents types de peau.

Le chatbot, quant à lui, offre une interaction simple et efficace avec l'utilisateur, même si son fonctionnement reste basé sur des règles simples.

Cependant, certaines limites ont été observées :

- Le modèle IA n'est pas encore totalement exploité (logique if/else)
- Le chatbot reste limité aux mots-clés
- Les données utilisées sont simulées

Malgré cela, l'application reste fonctionnelle et démontre la faisabilité du projet.

## Conclusion

Ce chapitre a présenté l'implémentation complète de notre application intelligente dédiée aux soins de la peau.

Nous avons détaillé l'environnement de développement, l'architecture du système ainsi que les différentes fonctionnalités proposées. L'intégration de l'intelligence artificielle a permis de créer une application interactive capable de fournir des recommandations personnalisées, d'analyser les besoins de l'utilisateur et de répondre à ses questions.

Les résultats obtenus montrent que l'application est fonctionnelle, efficace et facile à utiliser. Elle constitue une première étape vers le développement d'un système plus avancé dans le domaine des soins cosmétiques intelligents.

Enfin, ce travail ouvre la voie à plusieurs améliorations futures, notamment l'intégration complète du modèle de deep learning, l'enrichissement des données et l'optimisation du chatbot



## **CONCLUSION GÉNÉRALE**

## conclusion générale

Dans ce mémoire, nous avons travaillé sur la conception et la réalisation d'un système de recommandation de produits cosmétiques basé sur l'intelligence artificielle générative. Notre objectif était de créer une application web qui aide les utilisateurs à choisir des produits adaptés à leur peau, à analyser leur type de peau et leurs problèmes, et à poser des questions à un chatbot intelligent.

Pour arriver à ce résultat, nous avons organisé notre travail en quatre chapitres.

Dans le premier chapitre, nous avons étudié les systèmes de recommandation. Nous avons vu les différentes méthodes comme le filtrage collaboratif et le filtrage basé sur le contenu. Nous avons aussi cherché des exemples d'applications dans le domaine de la beauté.

Dans le deuxième chapitre, nous avons exploré l'intelligence artificielle générative. Nous avons expliqué ce que sont les grands modèles de langage comme GPT, et nous avons appris les bases du prompt engineering pour orienter les réponses d'un chatbot.

Dans le troisième chapitre, nous avons conçu notre application. Nous avons dessiné des diagrammes UML pour montrer l'architecture du système, les interactions entre l'utilisateur et l'application, et la structure de la base de données.

Dans le quatrième chapitre, nous avons implémenté l'application. Nous avons utilisé Python avec Streamlit pour l'interface, TensorFlow pour le réseau de neurones, et Pandas pour les données. L'application propose trois fonctionnalités : la recommandation de produits, l'analyse de la peau, et un chatbot.

Après avoir testé notre application, nous avons constaté qu'elle fonctionne correctement. Le système de recommandation affiche les 5 meilleurs produits pour chaque utilisateur. L'analyse de la peau donne des solutions adaptées comme "Use salicylic acid cleanser" pour une peau grasse avec de l'acné. Le chatbot répond aux questions sur les soins de la peau et refuse poliment les questions hors sujet.

Cependant, notre application a quelques limites. Le système de recommandation ne fonctionne pas pour un nouvel utilisateur sans historique. L'analyse de la peau utilise encore une logique simple (if/else) au lieu du vrai réseau de neurones. Le chatbot ne comprend que l'anglais et seulement quelques mots-clés.

Nous pouvons améliorer notre travail à l'avenir. Par exemple, nous pourrions intégrer le vrai modèle de deep learning, ajouter plus de données d'entraînement, ajouter une vraie base de données comme MySQL, et déployer l'application sur Internet avec Streamlit Cloud.

En conclusion, nous avons réussi à créer une application fonctionnelle qui montre comment l'intelligence artificielle peut aider les gens à prendre soin de leur peau. Ce projet nous a appris à utiliser Python, Streamlit, TensorFlow, et à organiser un travail de recherche. Nous espérons continuer à améliorer cette application dans le futur.

## Références

- [1] Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson. [2] Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine*. [3] Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook*. Springer. [4] Amisha et al. (2019). Artificial intelligence in healthcare: Ethical and general perspective. *Journal of Family Medicine and Primary Care*. [5] Burke, R. (2002). Hybrid recommender systems. *User Modeling and User-Adapted Interaction*. [6] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems. *IEEE Transactions on Knowledge and Data Engineering*. [7] Pazzani, M., & Billsus, D. (2007). Content-based recommendation systems. [8] Zhang, S., et al. (2019). Deep learning based recommender systems. *ACM Computing Surveys*. [9] OpenAI. (2020). Language models are few-shot learners. [10] Vaswani, A., et al. (2017). Attention is all you need. *NeurIPS*. [11] Esteva, A., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. [12] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. [13] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. [14] Chollet, F. (2018). *Deep Learning with Python*. Manning Publications. [15] Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender systems handbook*. Springer. [16] 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. [17] Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill. [18] Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Pearson. [19] Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann. [20] : Mitchell (1997) *Machine Learning*, McGraw Hill. [21] Goodfellow et al. (2016) *Deep Learning*, MIT Press. [22] : Szeliski (2010) *Computer Vision: Algorithms and Applications* Springer. [23] : Jurafsky & Martin (2023) *Speech and Language Processing*. [24] Goodfellow, I., et al. (2014). *Generative adversarial nets*. *NeurIPS*. <https://arxiv.org/abs/1406.2661> [25] LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. *Nature*. <https://doi.org/10.1038/nature14539> [26] Radford, A., et al. (2019). *Language models are unsupervised multitask learners*. OpenAI. <https://openai.com/research/language-unsupervised> [27] Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson. [28] Goodfellow, I., et al. (2014). *Generative adversarial nets*. <https://arxiv.org/abs/1406.2661> [29] LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. *Nature*. [30] Esteva, A., et al. (2017). *Dermatologist-level classification of skin cancer with deep neural networks*. *Nature*. [31] Topol, E. (2019). *High-performance medicine: AI in healthcare*. *Nature Medicine*. [32] Esteva, A., et al. (2017). *Dermatologist-level classification of skin cancer with deep neural networks*. *Nature*. <https://doi.org/10.1038/nature21056> [33] Goodfellow, I., et al. (2014). *Generative adversarial nets*. <https://arxiv.org/abs/1406.2661> [34] LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. *Nature*. [35] Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson. [36] Russell, S. & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson [37] Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook*. Springer. [38] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. [39] Zhang, S. et al. (2019).

“Deep Learning based Recommender Systems”, IEEE [40] Vaswani, A. et al. (2017).

“Attention is All You Need”, NeurIPS.

- Software Foundation. *Python Documentation*. Disponible sur (chapitre 4) : <https://docs.python.org/3/>
- Streamlit. *Streamlit Documentation*. Disponible sur : <https://docs.streamlit.io/>
- Pandas Development Team. *Pandas Documentation*. Disponible sur : <https://pandas.pydata.org/docs/>
- NumPy Developers. *NumPy Documentation*. Disponible sur : <https://numpy.org/doc/>
- TensorFlow. *TensorFlow Documentation*. Disponible sur : <https://www.tensorflow.org/>
- Keras. *Keras Documentation*. Disponible sur : <https://keras.io/>
- Scikit-learn. *Scikit-learn Documentation*. Disponible sur : <https://scikit-learn.org/stable/>
- Visual Studio Code. *Documentation officielle*. Disponible sur : <https://code.visualstudio.com/> Git. *Git Documentation*. Disponible sur : <https://git-scm.com/docs> Tawk.to. *Official Website*. Disponible sur : <https://www.tawk.to/>

## ملخص

### ملخص باللغة العربية

يندرج هذا البحث ضمن تطوير الأنظمة الذكية المطبقة في مجال الجمال والعناية بالبشرة. الهدف الرئيسي من هذا العمل هو تصميم وإنجاز نظام توصية لمنتجات التجميل يعتمد على الذكاء الاصطناعي التوليدي.

في ظل تنوع وغنى سوق مستحضرات التجميل، يواجه المستخدمون صعوبة في اختيار المنتجات المناسبة لاحتياجاتهم الخاصة. ولمعالجة هذه الإشكالية، اقترحنا حلاً ذكياً قادراً على تحليل خصائص بشرة المستخدم وتقديم توصيات مخصصة.

يعتمد هذا النظام على تقنيات التعلم الآلي والشبكات العصبية، مما يسمح بنمذجة تفضيلات المستخدمين وتحسين دقة الاقتراحات. كما تم تصميم معمارية كاملة تشمل جمع البيانات، معالجتها، ثم توليد التوصيات في الوقت الحقيقي.

تُظهر النتائج المحصل عليها أن هذا النهج يحسن بشكل كبير تجربة المستخدم من خلال تقديم توصيات دقيقة، مخصصة وديناميكية.

## Abstract

### Yoursummary in English

This thesis is part of the development of intelligent systems applied to the field of beauty and skincare. The main objective of this work is the design and implementation of a cosmetic product recommendation system based on generative artificial intelligence.

In a context where the cosmetics market is highly rich and diverse, users often face difficulties in selecting products that are suitable for their specific needs. To address this issue, we propose an intelligent solution capable of analyzing the user's skin characteristics and generating personalized recommendations.

The system is based on machine learning techniques and neural networks, allowing it to model user preferences and improve the relevance of the suggested products. A complete architecture has been designed, including data collection, data processing, and real-time recommendation generation.

The obtained results show that the proposed approach significantly enhances user experience by providing accurate, personalized, and dynamic recommendations.

## Résumé

### Votre résumé en français

Ce mémoire s'inscrit dans le cadre du développement des systèmes intelligents appliqués au domaine de la beauté et des soins de la peau. L'objectif principal de ce travail est la conception et la réalisation d'un système de recommandation de produits cosmétiques basé sur l'intelligence artificielle générative.

Dans un contexte où le marché cosmétique est riche et diversifié, les utilisateurs rencontrent souvent des difficultés à choisir des produits adaptés à leurs besoins spécifiques. Afin de répondre à cette problématique, nous avons proposé une solution intelligente capable d'analyser les caractéristiques de la peau de l'utilisateur et de générer des recommandations personnalisées.

Ce système repose sur des techniques d'apprentissage automatique et de réseaux de neurones, permettant de modéliser les préférences des utilisateurs et d'améliorer la pertinence des suggestions proposées. Une architecture complète a été conçue, incluant la collecte des données, leur traitement, ainsi que la génération de recommandations en temps réel.

Les résultats obtenus montrent que l'approche proposée permet d'améliorer significativement l'expérience utilisateur en offrant des recommandations adaptées, précises et dynamiques.