



ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET  
D'ANALYSE DES SYSTÈMES — RABAT

Projet de Fin d'Année – 2<sup>ème</sup> année (GL)

---

Mise en place d'un système de recommandation hybride  
pour les applications de santé en utilisant le NLP

---

**Date de soutenance : 08 / 06 / 2023**

Réalisé par :

EL OUARTI Mouad  
FIALI Mouad

Encadré par :

Pr. SABIRI BIHI

Jury :

Pr. EL ASRI Bouchra  
Pr. EL FAKER Abdellatif

Année scolaire : 2022/2023





# *Remerciements*

Nous tenons tout d'abord à exprimer notre profonde gratitude envers notre professeur EL ASRI BOUCHRA pour avoir proposé ce sujet captivant et pour sa confiance en notre capacité à mener à bien ce projet. Nous sommes également reconnaissants envers le Pr SABIRI BIHI pour son encadrement précieux tout au long de ce travail.

Nos remerciements sincères vont également aux membres du jury qui ont accepté de consacrer leur temps et leur expertise pour évaluer notre travail. Leur présence et leur contribution sont grandement appréciées.

Nous souhaitons également adresser nos remerciements chaleureux à l'ensemble du corps enseignant de l'ENSIAS. Leur dévouement et leurs efforts inlassables dans la transmission du savoir nous ont permis d'acquérir une formation de qualité.



# Résumé

Ce rapport constitue une synthèse de notre projet de fin d'année qui vise à approfondir et à améliorer nos connaissances dans le domaine des systèmes de recommandation, en mettant en avant l'utilisation du traitement automatique du langage naturel. Notre projet se concentre sur l'application de ces techniques dans le domaine de la santé. Nous avons développé un système de recommandation hybride en combinant le filtrage collaboratif, l'approche basée sur le contenu et le traitement automatique du langage naturel. En exploitant ces différentes approches, notre système prédit les maladies des utilisateurs en analysant leurs symptômes et leur propose des solutions personnalisées. Cette approche hybride, intégrant le traitement automatique du langage naturel, nous permet d'améliorer la précision et la pertinence des recommandations fournies.

# Abstract

This document provides a summary of our end-of-year project aimed at deepening and enhancing our knowledge in the field of recommendation systems. We specifically focused on applying these techniques in the healthcare domain. Our approach relied on a combination of several methods, including collaborative filtering and content-based approaches. By utilizing these approaches, we developed a hybrid recommendation system that predicts users' diseases based on their symptoms and offers personalized solutions. This hybrid approach allows us to leverage the advantages of both methods to provide more accurate and relevant recommendations. Furthermore, our system incorporates natural language processing techniques to further enhance its performance and effectiveness.

---

# TABLE DES MATIÈRES

---

TABLE DES FIGURES-----	8
INTRODUCTION-----	9
CHAPITRE 1 : CONTEXTE GÉNÉRAL DU PROJET -----	10
1. Contexte général -----	11
2. Problématique-----	11
3. Objectifs-----	11
4. Définitions et étude de l'état de l'art-----	12
4.1. Définitions -----	12
4.1.1. Systèmes de recommandation -----	12
4.1.2. Apprentissage automatique -----	13
4.2. État de l'art-----	13
4.2.1. DeepHealth -----	13
4.2.2. Clinical Decision Support Systems (CDSS) -----	14
5. Plannification du projet-----	14
CHAPITRE 2 : MATÉRIEL & MÉTHODOLOGIE -----	15
1. Jeu des données -----	16
1.1. Description de l'évidence -----	16
1.2. Description de la pathologie (maladies) -----	17
1.3. Description des patients-----	17
2. Approche et modèles choisis-----	18
2.1. Modèles de prédiction (Système de recommandation)-----	18
2.1.1. Filtrage collaboratif -----	18
2.1.2. Filtrage basé sur le contenu -----	19
2.2. Modèle d'extraction de symptômes (NLP) -----	19
3. Objectifs-----	20
3.1. Modèles de prédiction (Système de recommandation)-----	20
3.2. Modèle NER d'extraction des symptômes-----	20
4. Démarche et déroulement-----	20

4.1. Prédiction des maladies	20
4.2. NLP et extraction des symptômes	23
CHAPITRE 3 : RÉALISATION ET RÉSULTATS	26
1. Évaluation des performances	27
1.1. Extraction des évidences (NLP)	27
1.2. Prédiction des maladies	27
2. Interface graphique	28
CHAPITRE 4 : DISCUSSION ET PERSPECTIVES	31
1. Discussion des résultats	32
2. Améliorations et perspectives	33
CONCLUSION	35
BIBLIOGRAPHIE	36

---

# TABLE DES FIGURES

---

Figure 1: Illustration - système de recommandation -----	12
Figure 2: Illustration - Machine learning-----	13
Figure 3: Diagramme de gantt -----	14
Figure 4: Filtrage collaboratif - Preprocessing-----	21
Figure 5: Entrainement du système de filtrage collaboratif-----	22
Figure 6: Adaptation des données - NLP-----	23
Figure 7: Annotation manuelle des données -----	23
Figure 8: Annotation des entités -----	24
Figure 9: Entrainement du modèle - NER -----	24
Figure 10: Evaluation des résultats - NLP-----	27
Figure 11: Résultat du système de filtrage collaboratif -----	27
Figure 12: Probabilités des prédictions - filtrage basé sur le contenu -----	28
Figure 13: Interface - formulaire de collecte de données -----	28
Figure 14: Interface - chat du système de recommandation -----	29
Figure 15: Interface - interaction utilisateur et système-----	29
Figure 16: Interface - Réponse en cas d'échec -----	30
Figure 17: Interface - interaction utilisateur et système 2 -----	30



---

# INTRODUCTION

---

Au cours des dernières années, les systèmes de recommandation ont suscité un intérêt considérable en raison de leur capacité à fournir des suggestions personnalisées et à améliorer l'expérience des utilisateurs dans différents domaines. Un domaine où ces systèmes sont particulièrement prometteurs est celui des applications de santé. Avec la disponibilité croissante des données de santé numériques et la demande croissante de solutions de santé personnalisées, la mise en place d'un système de recommandation efficace devient essentielle.

Ce projet vise à relever ce défi en développant un système de recommandation hybride qui combine différentes approches pour fournir des recommandations personnalisées dans le domaine de la santé. Notre approche repose sur l'utilisation du traitement automatique du langage naturel (NLP) pour analyser les symptômes des utilisateurs et prédire les maladies potentielles. De plus, nous intégrons des techniques de filtrage collaboratif et de recommandation basée sur le contenu pour améliorer la pertinence des suggestions fournies.

L'objectif principal de ce projet est de proposer un système de recommandation performant qui aide les utilisateurs à prendre des décisions éclairées en matière de santé. En fournissant des recommandations personnalisées basées sur les symptômes des utilisateurs et en proposant des solutions adaptées, nous visons à améliorer l'efficacité des soins de santé et à optimiser l'expérience des utilisateurs dans le domaine de la santé.

---

# CHAPITRE 1 :

## CONTEXTE GÉNÉRAL DU PROJET

---

### *Introduction du chapitre :*

Ce chapitre situe notre projet dans son contexte en identifiant la problématique à résoudre et en définissant notre objectif spécifique. Nous examinons les travaux et les systèmes de recommandation existants dans le domaine de la santé. Nous détaillons également la planification de notre projet, en décrivant les étapes et les ressources nécessaires.

# 1.Contexte général

Le projet se situe dans le domaine des applications de santé et vise à développer un système de recommandation hybride utilisant le traitement automatique du langage naturel (NLP). Ce système sera conçu pour prédire les maladies des utilisateurs en se basant sur leurs symptômes et leur fournir des recommandations personnalisées de solutions de santé.

## 2.Problématique

La disponibilité croissante de données de santé numériques et la demande croissante de solutions de santé personnalisées nécessitent la mise en place d'un système de recommandation efficace pour aider les professionnels dans le domaine de la santé. Cependant, les systèmes de recommandation traditionnels se basant uniquement sur le filtrage collaboratif ou l'approche basée sur le contenu peuvent présenter des limitations en termes de pertinence et de précision. Il est donc nécessaire de développer un système de recommandation hybride qui combine ces approches pour améliorer la pertinence des recommandations dans le domaine de la santé.

## 3.Objectifs

On pourra résumer les objectifs de notre projet dans ce qui suit :

- Développer un système de recommandation hybride : L'objectif principal est de concevoir un système de recommandation qui combine le filtrage collaboratif et l'approche basée sur le contenu pour fournir des recommandations personnalisées de solutions de santé aux utilisateurs.
- Utiliser le NLP pour l'analyse des symptômes : Le projet vise à intégrer des techniques de traitement automatique du langage naturel pour analyser les symptômes des utilisateurs et prédire les maladies potentielles.
- Améliorer l'efficacité des soins de santé : En fournissant des recommandations personnalisées et adaptées aux utilisateurs, le projet cherche à améliorer l'efficacité des soins de santé et à faciliter la prise de décision éclairée des utilisateurs.

- Optimiser l'expérience des utilisateurs : L'objectif est de créer un système convivial et intuitif qui offre une expérience utilisateur fluide et satisfaisante lors de l'obtention de recommandations et des prédictions dans ce domaine.

## 4. Définitions et étude de l'état de l'art

### 4.1. Définitions

#### 4.1.1. Systèmes de recommandation

Un moteur de recommandation, également appelé système de recommandation, est un outil ou un algorithme qui analyse les données sur les préférences, les intérêts ou les comportements d'un utilisateur pour lui fournir des recommandations personnalisées. Son objectif est d'aider les utilisateurs à découvrir de nouveaux produits, services, contenus ou informations qui correspondent à leurs goûts et à leurs besoins.

Le moteur de recommandation utilise des techniques d'apprentissage automatique, telles que le filtrage collaboratif, l'approche basée sur le contenu, le traitement automatique du langage naturel (NLP) ou d'autres méthodes analytiques, pour analyser les données disponibles. Il compare les informations de l'utilisateur avec celles d'autres utilisateurs similaires ou avec des caractéristiques similaires, afin de générer des recommandations personnalisées et pertinentes.

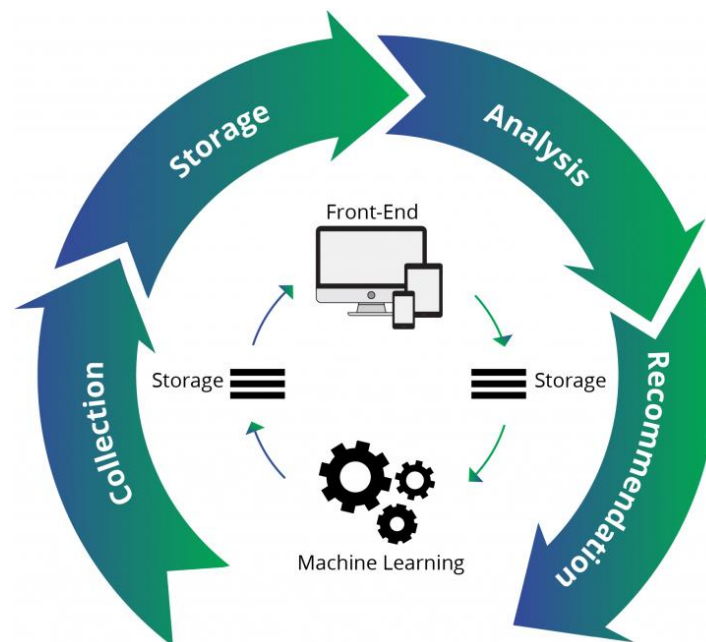


Figure 1: Illustration - système de recommandation

### 4.1.2. Apprentissage automatique

L'apprentissage automatique, également connu sous le nom de machine learning, est une discipline de l'intelligence artificielle (IA) et de l'informatique qui se focalise sur l'utilisation de données et d'algorithmes pour reproduire de manière progressive le processus d'apprentissage humain, en améliorant continuellement la précision des résultats obtenus.

De plus, l'apprentissage automatique joue un rôle prépondérant dans le domaine en plein essor de la science des données. En utilisant des méthodes statistiques, les algorithmes sont entraînés à réaliser des classifications ou des prédictions, permettant ainsi de mettre en évidence des informations cruciales dans le cadre de projets d'exploration de données. Ces informations servent ensuite de guide pour la prise de décisions au sein des applications et des entreprises.

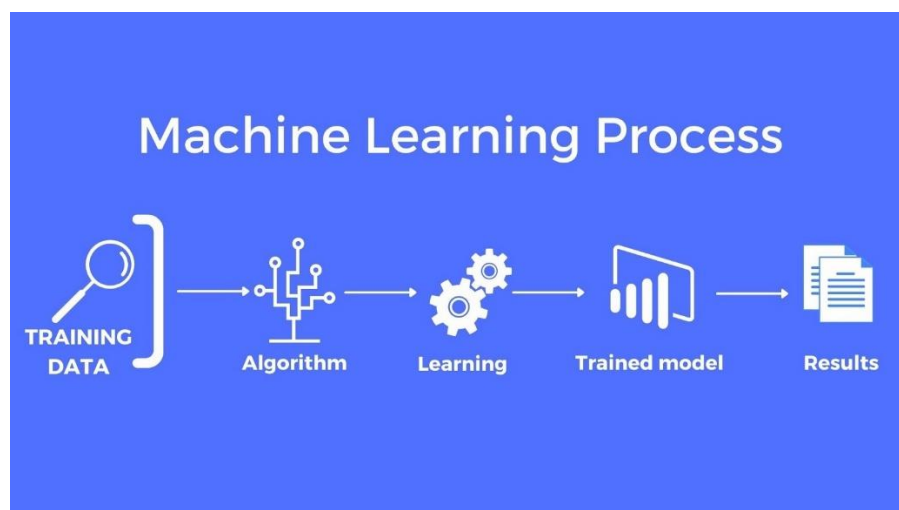


Figure 2: Illustration - Machine learning

## 4.2. État de l'art

### 4.2.1. DeepHealth

C'est une initiative de recherche qui se concentre sur l'utilisation de l'apprentissage en profondeur (Deep Learning) pour améliorer les systèmes de recommandation dans le domaine de la santé. Il explore des techniques avancées pour la prédiction de maladies, la recommandation de traitements et l'optimisation des soins de santé personnalisés. [\[1\]](#)

## 4.2.2. Clinical Decision Support Systems (CDSS)

Les systèmes de support à la décision clinique sont des outils informatiques conçus pour aider les professionnels de la santé dans leurs prises de décision en fournissant des recommandations basées sur des données cliniques, des protocoles médicaux et des directives de pratique. [2]

## 5.Plannification du projet

La planification du projet revêt une importance capitale dans la phase préliminaire. Elle englobe la prévision du déroulement du projet tout au long des différentes étapes qui composent le cycle de développement, dans notre cas, un cycle en cascade. À cet effet, nous avons élaboré un diagramme de Gantt afin de structurer les différentes phases du projet et d'établir leur échéancier respectif.



Figure 3: Diagramme de gantt

---

# CHAPITRE 2 :

## MATÉRIEL & MÉTHODOLOGIE

---

### *Introduction du chapitre :*

Ce chapitre se concentre sur la description de la collecte des données utilisées dans notre étude, afin de mieux comprendre les objectifs de leur utilisation. Ensuite, nous détaillerons la méthodologie que nous avons suivie et présenterons le déroulement étape par étape de notre projet. Cela nous permettra d'obtenir une vue d'ensemble complète de notre approche et des résultats que nous visons à atteindre.

# 1. Jeu des données

*Note : Nous utilisons le terme "évidence" de manière générale pour désigner un symptôme ou un antécédent. [\[3\] source de la dataset](#)*

Le jeu des données contient les fichiers suivants :

- **release\_evidences.json** : un fichier JSON décrivant toutes les évidences possibles prises en compte dans le jeu des données.
- **release\_conditions.json** : un fichier JSON décrivant toutes les pathologies prises en compte dans le jeu des données.
- **release\_train\_patients.csv** : un fichier CSV contenant les patients de l'ensemble d'entraînement.
- **release\_validate\_patients.csv** : un fichier CSV contenant les patients de l'ensemble de validation.
- **release\_test\_patients.csv** : un fichier CSV contenant les patients de l'ensemble de test.

## 1.1. Description de l'évidence

Chaque évidence dans le fichier "release\_evidences.json" est décrite en utilisant les entrées suivantes :

- **name** : nom de l'évidence.
- **code\_question** : un code permettant d'identifier les évidences liées. Les évidences ayant le même code\_question forment un groupe de symptômes connexes. La valeur du code\_question fait référence à l'évidence qui doit être simulée/activée pour que les autres membres du groupe soient éventuellement simulés.
- **question\_fr** : la requête associée à l'évidence en français.
- **question\_en** : la requête associée à l'évidence en anglais.
- **is\_antecedent** : un indicateur indiquant si l'évidence est un antécédent ou un symptôme.
- **data\_type** : le type d'évidence. Nous utilisons B pour binaire, C pour catégorique et M pour les évidences à choix multiple.
- **default\_value** : la valeur par défaut de l'évidence. Si cette valeur est utilisée pour caractériser l'évidence, c'est comme si l'évidence n'était pas synthétisée.
- **possible-values** : les valeurs possibles pour les évidences. Valide uniquement pour les évidences catégoriques et à choix multiple.



- **value\_meaning**: La signification, en français et en anglais, de chaque code qui fait partie du champ possible-values. Valide uniquement pour les évidences catégoriques et à choix multiple.

## 1.2. Description de la pathologie (maladies)

Le fichier "release\_conditions.json" contient des informations sur les pathologies dont les patients des ensembles de données peuvent souffrir. Chaque pathologie possède les attributs suivants :

- **condition\_name** : nom de la pathologie.
- **cond-name-fr** : nom de la pathologie en français.
- **cond-name-eng** : nom de la pathologie en anglais.
- **icd10-id** : code ICD-10 de la pathologie.
- **severity** : la gravité associée à la pathologie. Plus la valeur est basse, plus la pathologie est grave.
- **symptoms** : structure de données décrivant l'ensemble des symptômes caractérisant la pathologie. Chaque symptôme est représenté par son nom correspondant dans le fichier "release\_evidences.json".
- **antecedents** : structure de données décrivant l'ensemble des antécédents caractérisant la pathologie. Chaque antécédent est représenté par son nom correspondant dans le fichier "release\_evidences.json".

## 1.3. Description des patients

Chaque patient dans chacun des 3 ensembles de données possède les attributs suivants :

- **AGE** : l'âge du patient synthétisé.
- **SEX** : le sexe du patient synthétisé.
- **PATHOLOGY** : nom de la pathologie réelle (propriété "condition\_name" dans le fichier "release\_conditions.json") dont souffre le patient synthétisé.
- **EVIDENCES** : liste des évidences vécues par le patient. Une évidence peut être binaire, catégorique ou à choix multiple. Une évidence catégorique ou à choix multiple est représentée sous la forme [nom-de-l'évidence]\_@[valeur-de-l'évidence] où [nom-de-l'évidence] est le nom de l'évidence (entrée "name" dans le fichier "release\_evidences.json") et [valeur-de-l'évidence] est une valeur de l'entrée "possible-values". Notez que pour une évidence à choix multiple, il est possible

d'avoir plusieurs éléments [nom-de-l'évidence]\_[valeur-de-l'évidence] dans la liste des évidences, chaque élément étant associé à une valeur d'évidence différente. Une évidence binaire est représentée sous la forme [nom-de-l'évidence].

- **INITIAL\_EVIDENCE** : l'évidence fournie par le patient pour démarrer une interaction avec un système ASD/AD (Trouble du Spectre Autistique / Déficience Intellectuelle). Cela est utile lors de l'évaluation d'un modèle pour une comparaison équitable des systèmes ASD/AD, car ils commenceront tous une interaction avec un même patient à partir du même point de départ. L'évidence initiale est sélectionnée de manière aléatoire parmi les évidences binaires trouvées dans la liste des évidences mentionnée ci-dessus (c'est-à-dire EVIDENCES) et fait partie de cette liste.
- **DIFFERENTIAL\_DIAGNOSIS** : Le diagnostic différentiel réel pour le patient. Il est représenté sous la forme d'une liste de paires de la forme [[patho\_1, proba\_1], [patho\_2, proba\_2], ...] où patho\_i est le nom de la pathologie (entrée "condition\_name" dans le fichier "release\_conditions.json") et proba\_i est sa probabilité associée.

## 2.Approche et modèles choisis

### 2.1. Modèles de prédiction (Système de recommandation)

Afin d'implémenter un système de recommandation **hybride**, on a choisi de diviser la tâche des prédictions des maladies sur 2 systèmes de recommandation qui se base sur 2 techniques différentes :

#### 2.1.1. Filtrage collaboratif

Le système de recommandation repose sur la collecte d'informations concernant les préférences des utilisateurs et l'identification de similarités entre ces utilisateurs ou les éléments recommandés.

Dans notre cas, nous avons utilisé les données des patients comme base de données pour établir des similitudes entre leurs informations (âge, sexe et symptômes) et celles de l'utilisateur. Cela nous a permis de prédire la maladie dont l'utilisateur pourrait souffrir.

Les modèles sélectionnés pour cette partie sont des modèles de classification supervisée, à savoir :

- SGD Classifier (Stochastic Gradient Descent).
- MLP (Multi-Layer Perceptron)
- BernoulliNB (Naive Bayes)
- Perceptron

### 2.1.2. Filtrage basé sur le contenu

Dans la méthode de filtrage basée sur le contenu, nous avons décidé de prendre en compte les données spécifiques à chaque maladie.

Nous avons utilisé les descriptions de maladies qui contiennent des informations détaillées sur les symptômes associés à chaque maladie. Ces descriptions ont été utilisées comme contenu pour prédire la maladie à partir des symptômes fournis par l'utilisateur. Ainsi, en comparant les symptômes de l'utilisateur avec les informations contenues dans les descriptions de maladies, nous sommes en mesure de fournir une prédiction de la maladie correspondante.

Les modèles sélectionnés pour cette partie sont 3 modèles parmi ceux choisis dans le filtrage collaboratif, à savoir :

- SGD Classifier (Stochastic Gradient Descent).
- MLP (Multi-Layer Perceptron)
- BernoulliNB (Naive Bayes)

## 2.2. Modèle d'extraction de symptômes (NLP)

Le modèle choisi pour l'extraction des symptômes est un modèle **NER (Name Entity Recognition)** utilisant la bibliothèque de traitement automatique des langues **spaCy**. Le modèle a été entraîné sur une base de données de patients présentant différentes conditions médicales ainsi que le code médical auquel le symptôme est associé.

## 3.Objectifs

### 3.1. Modèles de prédiction (Système de recommandation)

L'objectif de cette partie est de développer un système de recommandation hybride visant à prédire la maladie dont un patient pourrait souffrir, tout en fournissant une liste des maladies possibles. Cette prédiction sera basée sur les symptômes présentés par le patient, ainsi que des informations telles que son âge et son genre (masculin ou féminin). En combinant les techniques du filtrage collaboratif et du filtrage basé sur le contenu, notre objectif est de fournir des recommandations précises et personnalisées pour améliorer le processus de diagnostic médical et faciliter la prise de décision clinique.

### 3.2. Modèle NER d'extraction des symptômes

L'objectif de cette étape était de développer un modèle **NLP** basé sur **spaCy NER** pour extraire avec précision les symptômes des textes médicaux des patients. Les symptômes extraits devraient être utilisés comme entrée pour notre modèle (modèle de prédiction des maladies), afin d'identifier la maladie correspondante et fournir d'autres informations cliniques et recommandations au patient.

## 4.Démarche et déroulement

### 4.1. Prédictions des maladies

#### ➤ Adaptation des données :

Dans le cadre du système de recommandation basé sur le contenu, une étape essentielle consiste à extraire les informations pertinentes à partir d'un fichier décrivant les maladies (fichier JSON) et à les préparer pour l'entraînement des modèles. Cette étape d'adaptation des données implique la transformation des informations extraites en un format lisible par les modèles d'apprentissage automatique utilisés dans notre projet. Elle vise à organiser et à structurer les données des maladies de manière à ce qu'elles puissent être utilisées efficacement par les algorithmes de recommandation.

## ➤ Prétraitement des données :

Pour le prétraitement des données, nous avons utilisé Dask, une bibliothèque de calcul parallèle, car notre ensemble de données était très volumineux (1 292 579 lignes) et nous ne disposions pas de ressources suffisantes pour le traiter en une seule fois. Avec Dask, nous avons divisé les données des patients en plusieurs parties, créant ainsi plusieurs fichiers CSV (chunks), ce qui nous a permis de les traiter de manière plus efficace avec la bibliothèque Pandas.

Nous avons ensuite lu ces fichiers en utilisant Pandas par chunks. Dans chaque chunk, nous avons effectué des opérations de prétraitement spécifiques. Tout d'abord, nous avons éliminé les colonnes non nécessaires telles que "initial\_evidence", qui n'étaient pas pertinentes pour notre analyse et on a transformé la colonne « SEX » en une colonne binaire. Ensuite, nous avons transformé la colonne "EVIDENCES", qui contenait une liste d'évidences, en plusieurs colonnes binaires (0 ou 1) en utilisant la technique MultiLabelBinarizer. Cette transformation nous a permis de représenter chaque évidence individuellement, ce qui était essentiel pour notre modèle de recommandation.

```
1 for chunk in dataset_df:
2     print('Processing chunk', chunk_number, '-----')
3
4     # Create a MultiLabelBinarizer object
5     mlb = MultiLabelBinarizer(classes=list(unique_values))
6
7     # Convert the evidence column into a list of lists
8     evidence_lists = chunk['EVIDENCES'].apply(ast.literal_eval)
9
10    # Apply one-hot encoding to the evidence lists
11    print('Creating encoded evidences dataframe using one-hot...')
12    evidence_encoded = pd.DataFrame(mlb.fit_transform(evidence_lists), columns=mlb.classes_)
13    print('Encoded evidences dataframe created.')
14
15    print('size of chunk: ', chunk.shape)
16    print('size of encoded: ', evidence_encoded.shape)
17
18    # drop indexes to avoid null values
19    chunk.reset_index(drop=True, inplace=True)
20    evidence_encoded.reset_index(drop=True, inplace=True)
21
22    # Concatenate the encoded evidence columns with the chunk
23    processed_chunk = pd.concat([chunk, evidence_encoded], axis=1)
24    print('Chunk processed as pandas dataframe')
25
26    processed_chunk.to_csv('/content/gdrive/MyDrive/Colab/Dataset chunks/chunk_'+ str(chunk_number) + '.csv', index=False)
27    print('Chunk added to csv.')
28    chunk_number+=1
```

Figure 4: Filtrage collaboratif - Preprocessing

## ➤ Entraînement du modèle :

Pour l'entraînement des modèles (pour le filtrage collaboratif), nous avons également adopté une approche par chunks, étant donné la taille considérable de notre ensemble de données. Nous avons utilisé la méthode de lecture par chunks de Pandas pour charger et traiter les données par petits morceaux à la fois, afin de gérer efficacement la limitation des ressources.

```
1 def fit_by_chunks(model, features, targets):
2     if(len(features) != len(targets)):
3         raise Exception("features should be compatible with targets")
4
5     for i in range(len(features)):
6         print('Training model on chunk ', i+1)
7         # Train the model by parts
8         if i == 0: # pass the classes list in the 1st iteration
9             model.partial_fit(features[i], targets[i], classes=classes)
10        else:
11            model.partial_fit(features[i], targets[i])
12        print('Training on chunk ', i+1, ' finished')
13
14    y_pred_test = model.predict(X_test_global)
15    test_accuracy = accuracy_score(y_test_global, y_pred_test)
16    print('Test Accuracy:', test_accuracy)
```

Figure 5: Entraînement du système de filtrage collaboratif

Pour le filtrage basé sur le contenu on a entraîné les modèles directement sur les maladies existantes.

### ➤ Évaluation du modèle :

Pour évaluer les performances du filtrage collaboratif, nous avons divisé nos données en un ensemble d'entraînement (80%) et un ensemble de test (20%). Nous avons utilisé l'ensemble de test pour évaluer la précision de chaque modèle en calculant les scores correspondants. Ces scores nous ont permis de mesurer la qualité des recommandations générées par chaque modèle.

En ce qui concerne le filtrage basé sur le contenu, nous avons adopté une approche différente pour évaluer ses performances. Étant donné qu'il n'y avait pas de données de test disponibles, nous avons utilisé une méthode de génération aléatoire de symptômes pour simuler des scénarios d'utilisateurs. Nous avons ensuite utilisé les modèles basés sur le contenu pour prédire les maladies correspondantes à ces symptômes générés aléatoirement.

## 4.2. NLP et extraction des symptômes

### ➤ Adaptation des données :

Les textes médicaux présents dans la principale base de données ont été personnalisés pour un entraînement du modèle d'extraction des symptômes.

	A	B	C
1	Declaration patient	evidence_rep	
2	J'ai été infecté par un virus récemment.	B34.9	
3	Je suis actuellement traité ou j'ai été récemment traité avec un antibiotique pris par la bouche pour une infection de l'oreille ou une otite.	H6690	
4	Je suis infecté par le virus de l'immunodéficience humaine ou je suis séropositif.	HIV	
5	J'ai déjà eu une péricardite.	I30	
6	Moi ou un membre de ma famille avons déjà eu le croup.	J05.0	
7	J'ai déjà fait de l'eau sur les poumons.	J81	
8	Je souffre de pancréatite chronique.	K86.1	
9	J'ai une mauvaise alimentation.	Mauv_aliment	
10	Je suis dialysé.	Z99.2	
11	J'ai des ganglions enflés ou douloureux.	adp_dir	
12	J'ai pris ou j'ai déjà pris des anti-inflammatoires récemment.	ains	
13	J'ai allaité plus de 9 mois un de mes enfants.	allait_prol	
14	J'ai une allergie alimentaire connue et sévère.	allergie_sev	
15	Je trouve que mes symptômes sont pires depuis environ 2 semaines et que de moins en moins d'effort provoque les symptômes.	angor_accelere	
16	J'ai des douleurs au thorax même au repos.	angor_repos	
17	J'ai pris ou commencé un antipsychotique dans les derniers 7 jours.	antipsy_recent	
18	Je ressens une certaine anxiété, une fébrilité.	anxiete_s	
19	J'ai des origines asiatiques.	ap_asian	
20	Je suis atteint par la fibrose kystique.	ap_fk	
21	Je suis connu pour de l'hyperthyroïdie.	ap_hypert4	
22	Je souffre d'arthrite rhumatoïde.	ap_par	
23	J'ai déjà fait un pneumothorax spontané.	ap_pneumothorax	
24	Je suis connu pour un problème au niveau d'une valve cardiaque.	ap_valve	
25	Je fais des pauses respiratoires durant mon sommeil.	apnee	
26	Moi déjà connu pour avoir des problèmes de sommeil.	apnee	

Figure 6: Adaptation des données - NLP

### ➤ Prétraitement des données :

Les textes sont nettoyés pour supprimer les informations sensibles et les éléments non pertinents.

### ➤ Annotation des données :

Les symptômes ont été annotés manuellement dans les textes en utilisant des balises spécifiques.

```
{
  "annotations": [
    [
      {
        "J'ai été infecté par un virus récemment.",
        { "entities": [[24, 39, "B34.9"]] }
      },
      {
        "Je suis actuellement traité ou j'ai été récemment traité avec un antibiotique pris par la bouche pour une infection de l'oreille ou une otite.",
        {
          "entities": [
            [65, 128, "H6690"],
            [136, 141, "H6690"]
          ]
        }
      },
      {
        "Je suis infecté par le virus de l'immunodéficience humaine ou je suis séropositif.",
        {
          "entities": [
            [32, 58, "HIV"],
            [70, 81, "HIV"]
          ]
        }
      },
      {
        "J'ai déjà eu une péricardite.",
        { "entities": [[17, 28, "I30"]] }
      },
      {
        "Moi ou un membre de ma famille avons déjà eu le croup.",
        { "entities": [[45, 53, "J05.0"]] }
      },
      {
        "J'ai déjà fait de l'eau sur les poumons.",
        { "entities": [[18, 39, "J81"]] }
      }
    ]
  ]
}
```

Figure 7: Annotation manuelle des données

Pour ce faire, un outil d'annotation en ligne a été utilisé qui est le [NER Text Annotator](#). [4]



Figure 8: Annotation des entités

## ➤ Entraînement du modèle :

Le modèle **spaCy NER** a été entraîné sur les données annotées en utilisant un algorithme d'apprentissage automatique.

```
for train_example in spacy_data:
    text = train_example["text"]
    entities = train_example["entities"]
    doc = nlp.make_doc(text)
    example = Example.from_dict(doc, {"entities": entities})
    examples.append(example)

# Disable other pipeline components except NER
other_pipes = [pipe for pipe in nlp.pipe_names if pipe != "ner"]
with nlp.disable_pipes(*other_pipes):
    # Initialize the optimizer
    optimizer = nlp.initialize()

    # Training loop
    n_epochs = 100 # Number of training epochs

    for epoch in range(n_epochs):
        random.shuffle(examples)
        losses = {}
        nlp.update(examples, sg=optimizer, losses=losses)
        print(f"Epoch: {epoch+1} Loss: {losses}")

# Save the trained model
nlp.to_disk("new_new_model")
```

Epoch: 1 Loss: {'ner': 5651.268764972687}  
Epoch: 2 Loss: {'ner': 5650.381702065468}  
Epoch: 3 Loss: {'ner': 5648.788575708866}  
Epoch: 4 Loss: {'ner': 5645.891540884972}  
Epoch: 5 Loss: {'ner': 5639.979927182198}  
Epoch: 6 Loss: {'ner': 5625.948770284653}

Figure 9: Entraînement du modèle - NER



➤ **Évaluation du modèle :**

Le modèle entraîné a été évalué sur un ensemble de données de test distinct pour mesurer sa performance en termes de précision et de rappel.

➤ **Analyse de résultats :**

Les performances du modèle ont été analysées et interprétées pour évaluer son efficacité dans l'extraction des symptômes.

---

## CHAPITRE 3 : RÉALISATION ET RÉSULTATS

---

### *Introduction du chapitre :*

Ce chapitre met en évidence plusieurs aspects clés de notre système de recommandation hybride dans le domaine de la santé. Nous analyserons tout d'abord les performances du système en évaluant sa capacité à fournir des recommandations personnalisées et pertinentes. Ensuite, nous présenterons l'interface graphique conviviale développée pour faciliter l'interaction des utilisateurs avec le système.

# 1.Évaluation des performances

## 1.1. Extraction des évidences (NLP)

Pour évaluer les performances de notre modèle NLP, nous avons utilisé un ensemble de données de test distinct. Cet ensemble de données est composé d'exemples sur lesquels le modèle n'a pas été entraîné, ce qui permet d'évaluer son comportement sur des données inconnues.

```
#testing the model
my_model = spacy.load("new_new_model")

tester = "J'ai une douleur qui s'améliore lorsque je penche mon corps vers l'avant.J'ai un cancer actif.J'ai une di
tester = tester.split(".")

symptoms = []
for test in tester:
    doc = my_model(test)
    for ent in doc.ents:
        start = ent.start_char
        end = ent.end_char
        label = ent.label_
        text = ent.text
        symptoms.append(label)

        print(f"Entity: {text}, Label: {label}, Start: {start}, End: {end}")
    print(symptoms)
```

Python

Entity: douleur qui s'améliore lorsque je penche mon corps vers l'avant, Label: BW\_BENDING, Start: 9, End: 72  
['BW\_BENDING']  
Entity: cancer actif, Label: C00-D48, Start: 8, End: 20  
['BW\_BENDING', 'C00-D48']  
Entity: diminution de l'appétit ou des boissons, Label: BOIRE\_PED, Start: 9, End: 48  
['BW\_BENDING', 'C00-D48', 'BOIRE\_PED']

Figure 10: Evaluation des résultats - NLP

## 1.2. Prédiction des maladies

Pour évaluer les performances du filtrage collaboratif, L'ensemble de test a été utilisé pour calculer les scores de précision de chaque modèle et évaluer la qualité des recommandations générées. Voici les résultats obtenus pour chaque modèle :

	Model	Accuracy	Precision	Recall	F1-Score
0	SGDClassifier	0.959369	0.989798	0.959369	0.969244
1	Perceptron	0.982106	0.988327	0.982106	0.981558
2	BernoulliNB	0.997428	0.997577	0.997428	0.997396
3	MLPClassifier	0.997451	0.997631	0.997451	0.997417

Figure 11: Résultat du système de filtrage collaboratif

Concernant le filtrage basé sur le contenu, nous avons utilisé une approche différente pour l'évaluation. Étant donné l'absence de données de test, les modèles ont été évalués en simulant des scénarios d'utilisateurs en générant aléatoirement des symptômes ce qui permet de voir les probabilités des prédictions. Voici un exemple de prédictions et probabilités :

	model	condition	probability
0	SGDClassifier	VIH (Primo-infection)	0.452046
1	BernoulliNB	Laryngospasme	0.638086
2	MLPClassifier	Laryngospasme	0.074412

Figure 12: Probabilités des prédictions - filtrage basé sur le contenu

## 2. Interface graphique

Nous avons développé une interface graphique conviviale sous la forme d'un formulaire tout d'abord pour recueillir les données nécessaires du patient, telles que son âge et son genre. Ensuite, une interface de chat qui permet à l'utilisateur d'exprimer ses symptômes en langage naturel. Le message de l'utilisateur est ensuite traité par le modèle NER (Named Entity Recognition) afin d'extraire les évidences pertinentes qui correspondent aux symptômes dans les modèles du système de recommandation. Le système utilise ensuite ces évidences pour prédire la maladie correspondante. Enfin, une réponse claire est renvoyée à l'utilisateur, présentant les résultats obtenus par le système de recommandation.

Voici les interfaces de notre application :



**Bienvenue !**  
Veuillez entrer les informations suivantes avant de commencer!

Nom  
Mouad FIALI

Genre  
Homme

Age  
22

Continuer

Figure 13: Interface - formulaire de collecte de données

L'utilisateur est ensuite dirigé vers l'interface de chat :

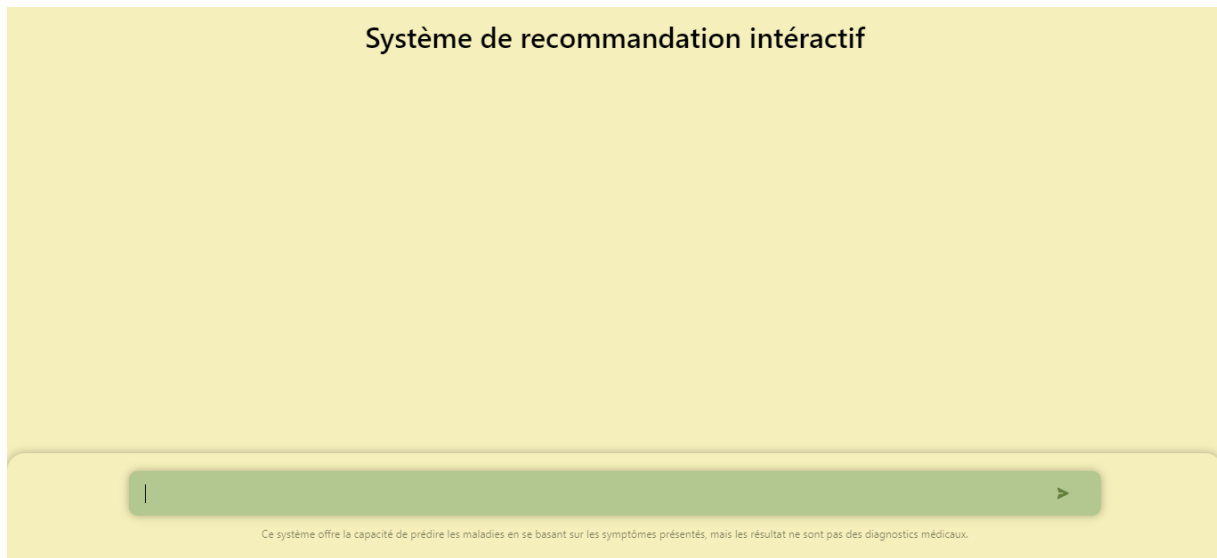


Figure 14: Interface - chat du système de recommandation

Exemple de réponses du système :

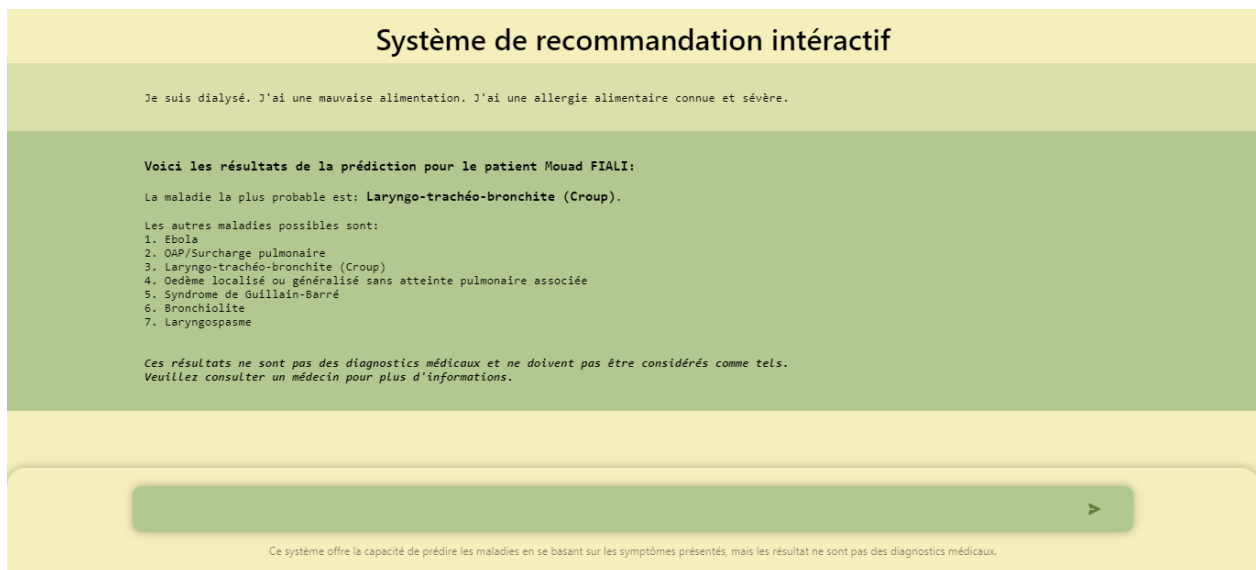


Figure 15: Interface - interaction utilisateur et système

En cas d'échec, c'est-à-dire que le modèle NLP ne détecte aucune évidence ou symptôme, il renvoie la réponse suivante :

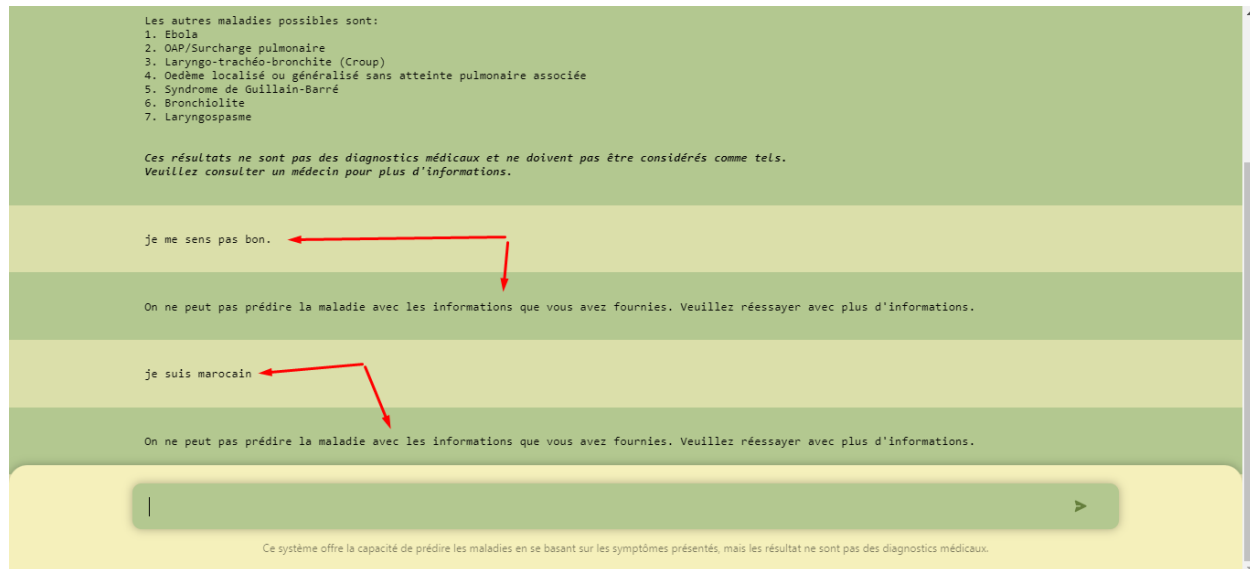


Figure 16: Interface - Réponse en cas d'échec



Figure 17: Interface - interaction utilisateur et système 2

---

# CHAPITRE 4 :

## DISCUSSION ET PERSPECTIVES

---

### *Introduction du chapitre :*

Ce chapitre offre une analyse approfondie et une interprétation des principales conclusions et aspects de notre système de recommandation hybride dans le domaine de la santé (Soit en ce qui concerne les performances des modèles entraînés pour le NLP et les prédictions ou aussi l'interface graphique mise en place).

# 1. Discussion des résultats

Le système de recommandation hybride que nous avons développé présente des performances prometteuses dans la prédiction des maladies en se basant sur les symptômes des patients. L'évaluation des performances du système a révélé plusieurs aspects clés de son fonctionnement.

Tout d'abord, en ce qui concerne l'extraction des évidences à partir des textes médicaux, notre modèle NLP basé sur spaCy NER a démontré une précision satisfaisante lors de l'extraction des symptômes. L'évaluation sur un ensemble de données de test distinct a montré que le modèle était capable d'identifier avec précision les symptômes pertinents, ce qui est crucial pour la prédiction précise des maladies.

En ce qui concerne le filtrage collaboratif, les modèles de classification supervisée que nous avons utilisés ont donné des résultats encourageants. Les modèles SGD Classifier, MLP et BernoulliNB ont tous montré des performances satisfaisantes en termes de précision des recommandations. Cependant, il convient de noter que des améliorations pourraient être apportées en explorant d'autres techniques de modélisation et en affinant les paramètres des modèles existants.

Concernant le filtrage basé sur le contenu, nous avons adopté une approche différente pour évaluer ses performances, étant donné l'absence de données de test. En simulant des scénarios d'utilisateurs en générant aléatoirement des symptômes, nous avons pu observer les probabilités de prédictions pour chaque maladie. Cela nous a permis d'obtenir une idée de l'efficacité du système dans la prédiction des maladies à partir des symptômes fournis. Les résultats obtenus ont montré des prédictions cohérentes avec des probabilités variables, ce qui indique que le système est capable de fournir des recommandations potentiellement utiles aux utilisateurs.

L'interface graphique conviviale que nous avons développée a facilité l'interaction des utilisateurs avec le système. Le formulaire de collecte des données a permis de recueillir les informations nécessaires du patient, tandis que l'interface de chat a offert aux utilisateurs la possibilité d'exprimer leurs



symptômes de manière naturelle. Les réponses claires et précises du système ont fourni aux utilisateurs les résultats de prédiction des maladies, ce qui peut les aider dans leur processus de diagnostic.

Cependant, il est important de souligner que notre système de recommandation présente certaines limites. Les performances du modèle NLP et des modèles de recommandation pourraient être améliorées en utilisant des ensembles de données plus larges et plus diversifiés. De plus, il est essentiel de continuer à améliorer les modèles et les algorithmes utilisés, ainsi que de mettre à jour régulièrement les informations médicales afin de garantir la pertinence des recommandations.

## 2. Améliorations et perspectives

Malgré les résultats satisfaisants qu'on a obtenus, il faudra améliorer notre système de recommandation hybride. Voici quelques améliorations qu'on pourra ajouter :

- Ajouter des suggestions de précautions : nous pourrions envisager d'ajouter des fonctionnalités telles que la suggestion de précautions et de médicaments basés sur les maladies identifiées chez les utilisateurs. Cela permettrait d'accroître l'utilité du système en fournissant des informations supplémentaires pour la gestion et le traitement des maladies.
- Utiliser l'apprentissage actif : Plutôt que de simplement demander aux utilisateurs de saisir leurs symptômes, nous pourrions concevoir un système qui pose des questions supplémentaires pour collecter des informations plus détaillées. Cela permettrait d'affiner davantage les recommandations en obtenant des données plus précises sur les symptômes et les antécédents médicaux des utilisateurs.
- Intégrer des données en temps réel : Actuellement, notre système utilise une base de données statique pour les symptômes et les maladies. En intégrant des données médicales en temps réel provenant de sources fiables, telles que des études cliniques et des

bases de données médicales mises à jour, nous pourrions garantir la pertinence et l'actualité des recommandations.

- Implémenter un système de feedback utilisateur : En permettant aux utilisateurs de fournir un feedback sur la pertinence des recommandations, nous pourrions améliorer en continu notre système en tenant compte des retours des utilisateurs réels. Cela nous aiderait à affiner les modèles de recommandation et à les adapter davantage aux besoins individuels des utilisateurs.

---

# CONCLUSION

---

En conclusion, ce rapport met en évidence l'importance des systèmes de recommandation dans le domaine de la santé et présente le développement d'un système de recommandation hybride qui intègre différentes approches pour fournir des suggestions personnalisées. Grâce à l'utilisation du traitement automatique du langage naturel (NLP) pour analyser les symptômes des utilisateurs et prédire les maladies potentielles, ainsi que l'incorporation de techniques de filtrage collaboratif et de recommandation basée sur le contenu, notre système vise à améliorer l'efficacité des prédictions des maladies des utilisateurs.

Les résultats obtenus lors de la mise en œuvre de notre système de recommandation ont démontré sa capacité à fournir des suggestions pertinentes et adaptées aux utilisateurs, en se basant sur leurs symptômes. Cela peut contribuer à améliorer la prise de décision en matière de santé, en aidant les utilisateurs à obtenir des informations précises et à prendre des décisions éclairées.

Cependant, il convient de souligner que notre système de recommandation n'est qu'un outil d'assistance et ne remplace pas les professionnels de la santé. Il est essentiel de rappeler que toute recommandation ou diagnostic médical doit être confirmé par un professionnel qualifié. Notre système vise à compléter le travail des experts en fournissant des informations supplémentaires et des suggestions, mais il ne peut en aucun cas se substituer à une consultation médicale.

---

# BIBLIOGRAPHIE

---

- [1] <https://deephealth.com/>
  - [2] <https://www.techtarget.com/searchhealthit/definition/clinical-decision-support-system-CDSS>
  - [3] [https://figshare.com/articles/dataset/DDXPlus\\_Dataset/20043374](https://figshare.com/articles/dataset/DDXPlus_Dataset/20043374)
  - [4] <https://tecoholic.github.io/ner-annotator/>
- Autres sources utilisées :
- [5] <https://www.geeksforgeeks.org/implement-chatgpt-in-a-flask-application/>
  - [6] <https://levelup.gitconnected.com/build-your-own-chatgpt-in-5-minutes-fd1cd369d681>
  - [7] <https://scikit-learn.org/stable/>
  - [8] <https://docs.dask.org/en/stable/>