

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM



Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

-Herrouel Nour El Yacine

-Sebihi Yacine Sami El Hak

THEME :

**Utilisation des systèmes multi-agents pour la
recommandation des Points d'intérêts**

Soutenu le :

Devant le jury composé de :

ADBELLAH BENSALLOUA C. MCA Université de Mostaganem Président

BENKERDAGH S. MCB Université de Mostaganem Examineur

MAGHNI SANDID ZOULIKHA MAA Université de Mostaganem Encadreur

Année Universitaire 2023-2024

Résumé

Dans un vaste environnement de données, les systèmes de recommandation sont un type de filtrage des données qui peut orienter l'utilisateur vers des informations pertinentes et attrayantes. En utilisant les données d'enregistrement, le système de recommandation des points d'intérêt (POI) prévoit les sites les plus probables. Ces systèmes ont été confrontés à un certain nombre de problèmes, notamment la rareté des données et la mise à l'échelle. Dans ce travail, nous avons abordé le problème en améliorant la suggestion de points d'intérêt (POI) dans notre application mobile à l'aide d'une série d'algorithmes. Nous avons appliqué nos propres algorithmes de pointe, tels que les systèmes implicites basés sur la confiance et les recommandations basées sur le contenu. En outre, nous avons intégré des algorithmes populaires et bien connus dans ce domaine, notamment l'algorithme du système de vote de Broda, les recommandations explicites basées sur la confiance et le filtrage collaboratif basé sur le voisinage. Toutes ces approches ont permis de rendre notre système de recommandation de points d'intérêt plus fiable et plus efficace.

Mots-clés : systèmes de recommandation, points d'intérêts, SMA.

Abstract

Within a vast data environment, recommendation systems are a type of data filtering that can direct the user to relevant and engaging information. Using the registration data, the Points of Interest (POI) recommendation system forecasts the most likely sites. These systems have faced a number of issues, including data and scaling. In this work, we have tackled the issue by improving the suggestion of points of interest (POI) within our mobile application by utilizing a range of algorithms. We have applied our own cutting-edge algorithms, such as implicit trust-based systems and content-based recommendations. Furthermore, we have integrated popular and well-known algorithms in this domain, including the Broda voting system algorithm, explicit trust-based recommendations, and neighborhood-based collaborative filtering. All of these different approaches have worked together to make our POI recommendation system more reliable and efficient.

Keywords: Recommendation system, point of interest, multi-agent-system

ملخص

ضمن بيئة البيانات الواسعة، تُعد أنظمة التوصية نوعًا من أنواع تصفية البيانات التي يمكنها توجيه المستخدم بالمواقع الأكثر (POI) إلى معلومات ذات صلة وجذابة. وباستخدام بيانات التسجيل، يتنبأ نظام التوصية بنقاط الاهتمام (POI) ببيانات سجلات نقاط الاهتمام) احتمالاً. وقد واجهت هذه الأنظمة عددًا من المشكلات، بما في ذلك ندرة البيانات والتوسع، والذي يشير إلى قدرة الخوارزمية على معالجة كميات كبيرة من البيانات بشكل جيد (غير شائعة للغاية).

في تطبيق الهاتف (POI) وقد عالجتنا في هذا العمل هذه المشكلة من خلال تحسين اقتراح نقاط الاهتمام المحمول الخاص بنا باستخدام مجموعة من الخوارزميات. لقد قمنا بتطبيق خوارزمياتنا المتطورة، مثل الأنظمة الضمنية القائمة على الثقة والتوصيات القائمة على المحتوى. علاوةً على ذلك، قمنا بدمج خوارزميات شائعة ومعروفة في هذا المجال، بما في ذلك خوارزمية نظام التصويت برودا والتوصيات الصريحة القائمة على الثقة والتصفية التعاونية القائمة على الجوار. وقد عملت جميع هذه الأساليب المختلفة معًا لجعل نظام التوصيات القائم على نقاط الاهتمام أكثر موثوقية وفعالية.

كلمات مفتاحية

نظام التوصية، نقطة الاهتمام، نظام متعدد العناصر

Dédicaces

This work is dedicated to:

Everyone who has helped me during my academic career. To my family, whose steadfast love and encouragement have always been my source of strength. To my buddies for their friendship and everlasting support. To my teachers, who have helped shape my path and motivated me to strive for perfection.

I dedicate this memoir to the memories of my grandmother, whose values and lessons continue to influence me. To my parents, for their sacrifices and opportunities that have served as the basis for my success.

Finally, I dedicate this to future students, hoping that it would ignite a flame of interest and desire for learning.

Sebihi Yacine sami El hak

Dédicaces

Ce travail est dédié à :

Tous ceux qui m'ont aidé tout au long de mon parcours académique. A ma famille, dont l'amour et les encouragements ont toujours été ma source de force. À mes amis, pour leur amitié et leur soutien indéfectible. À mes professeurs, qui m'ont aidée à tracer ma voie et m'ont motivée à viser la perfection.

Je dédie ce mémoire à ma mère, dont les valeurs et les leçons continuent de m'influencer. À mon père, pour son sacrifice et les opportunités qu'il m'a données, qui ont servi de base à ma réussite.

Enfin, je le dédie aux futurs étudiants, dans l'espoir qu'il allumera une flamme d'intérêt et de désir d'apprendre.

Herrouel Nour El Yacine

Remerciements

Nous souhaitons d'abord exprimer notre gratitude envers Dieu, le tout-puissant et miséricordieux, qui nous a accordé la force et la patience nécessaires pour mener à bien cette tâche.

N'hésitons pas à exprimer notre gratitude envers notre encadreur, Mme MAGHNI SANDID ZOULIKHA, pour son accompagnement, son écoute, ses explications, ses conseils, ses directives et ses encouragements.

N'oublions pas de remercier chaleureusement les membres du jury qui ont accepté de donner leur avis sur notre travail.

Nous souhaitons également exprimer notre gratitude envers tous ceux qui ont participé de manière directe ou indirecte à la réalisation de cette tâche.

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Exemple d'un système de recommandation (SR)	8
Figure 2	Interface Utilisateur de Netflix : Personnalisation et Recommandation.	8
Figure 3	Principales catégories de système de recommandation (SR).	11
Figure 4	Filtrage collaboratif.	12
Figure 5	Une illustration d'une matrice utilisateurs-items avec des évaluations claires.	13
Figure 6	Filtrage basé sur le contenu.	22
Figure 7	Filtrage hybride.	25
Figure 8	Filtrage basée sur la connaissance.	26
Figure 9	Graphes de connaissances.	27
Figure 10	La collecte des informations d'enregistrement dans Foursquare.	33
Figure 11	Proposition d'un cadre pour le système de recommandation de POI dans les LBSN Pour les individus et les groupes sociaux.	34
Figure 12	Le comportement séquentiel de mouvement de trois utilisateurs.	35
Figure 13	Un agent interagit avec son environnement grâce à ses capteurs et ses effecteurs.	39
Figure 14	Architecture des agents de notre système.	51
Figure 15	Diagramme de cas d'utilisation de notre système.	60
Figure 16	Diagramme de séquence « inscription »	65
Figure 17	Diagramme de séquence « Se connecter »	66
Figure 18	Diagramme de séquence « Consulter POI »	66
Figure 19	Diagramme de classe de notre Système	67
Figure 20	Diagramme de composant de notre Système	68
Figure 21	L'équation de similarité de Jaccard	76

Figure 22	Interface de démarrage	85
Figure 23	Interface d'inscription	86
Figure 24	Interface d'identification	87
Figure 25	Interface d'accueil	88
Figure 26	Interface de détail de POI	89
Figure 27	Interface de recherche ami	90
Figure 28	Interface de recherche POI	91
Figure 29	Interface POI catégories	92
Figure 30	Interface Profile d'utilisateur	93
Figure 31	Logo de l'application	97
Figure 32	Étapes de création une Base de données MongoDB	98

Liste des tableaux

TableauN°	Titre du tableau	Page
Tableau1	Exemple UBCF	16
Tableau 2	Exemple IBCF	18
Tableau 3	Tableau de synthèse	48
Tableau 4	Description textuelle de cas d'utilisation « Inscription ».	62
Tableau 5	Description textuelle du cas d'utilisation « se connecter ».	63
Tableau 6	Description textuelle de cas d'utilisation « consulter POI ».	64
Tableau 7	Matrice de confusion	78

Liste des abréviations

Abréviation	Expression Complète	Page
SR	Système de Recommandation.	7
FC	Filtrage Collaboratif.	7
FBC	Filtrage à base de contenu.	7
BC	L'approche basée sur la connaissance.	7
POI	Points d'Intérêt.	7
VPN	Virtual Private Network.	13
RGPD	Règlement Général sur la Protection des Données.	13
PCC	Coefficient de corrélation de Pearson.	15
UBCF	Filtrage collaboratif User-Based.	18
IBCF	Filtrage collaboratif item-Based.	18
IB	Item-Based.	19
SVD	Singular Value Decomposition.	19
RBM	Machines de Boltzmann restreintes.	20
IID	Indépendant et Identiquement Distribué.	20
RDF	Resource Description Framework.	27
LBSN	Les réseaux sociaux basés sur la localisation.	32
GPS	Global Positioning System.	32
DBSCAN	Density-Based Spatial Clustering of Applications with Noise.	44
SPs	StayPoints	44
Eps	Epsilon	44
ICA	Intelligent Central Agent	45

MAE	MeanAbsoluteError	46
SMA	Système Multi-Agents	37
GRSK	GeneralistRecommendation System Kernel.	46
API	Application Programming Interface.	50
UML	Unified Modeling Language.	58
RAM	Random Access Memory.	51
RSDPOI	Recommandation Système des Points d'Intérêt.	51
SDK	Software Development Kit.	70
iOS	iPhone Operating System.	70
VP	Vrai Positif.	78
TN	True Negative.	78
TP	True Positive.	78
FN	False Negative.	78
FP	False Positive.	78

Table des matières

Introduction Générale	5
Chapitre 1 Système de Recommandation & La Recommandation des Points d'intérêt.....	7
1.1 Introduction	7
1.2 Système de Recommandation:	7
1.2.1 Collection des données	9
1.3 Les Approches de Système de recommandation :	10
1.3.1 L'approche du filtrage collaboratif:.....	11
1.3.2 Filtrage basée sur le contenu:	21
1.3.3 L'approche Hybride :.....	24
1.3.4 L'approche basée sur la connaissance :	26
1.4 Points d'intérêt:.....	28
1.4.1 Composants de POI :	30
1.4.2 Types de points d'intérêt (POI) :	31
1.5 Recommandation des POIs :	32
1.5.1 Recommandation des POIs avec GPS Data :	32
1.5.2 Recommandation des POIs avec LBSN :	33
1.5.3 Facteurs d'influence :.....	34
1.6 Conclusion.....	36
Chapitre 2.....	37
2.1 Introduction	37
2.2 Agent.....	37
2.2.1 Définition Raffiné d'un Agent :	38
2.2.2 Distinction Entre Agents et Objets :	39
2.2.3 Vision Fonctionnelle :.....	40
2.2.4 Types d'Agents :.....	40
2.3 Un système multi-agents :	41

2.3.1	Défis et Perspectives dans l'Intégration des Systèmes Multi-Agents pour la Recommandation	42
2.4	Etat de l'art la recommandation et SMA.....	43
2.4.1	Architecture des SMA pour les Recommandations POI.....	43
2.4.2	SRMA pour l'Extraction des Intérêts Utilisateurs	44
2.4.3	Système de Recommandation Multi-Agents Intelligent	45
2.4.4	Architecture Multi-Agents pour l'e-Tourisme	46
2.5	Conclusion.....	49
Chapitre 3.....		50
3.1	Introduction	50
3.2	Système multi agent de recommandation des POI	50
3.2.1	Les Systèmes de Recommandation en Temps Réel : Flux de Données et Réactivité ...	57
3.3	Méthode d'Analyse et conception	58
3.3.1	UML.....	58
3.3.2	Spécification des besoins du système	58
3.4	Les Diagrammes UML.....	59
3.4.1	Diagramme de cas d'utilisation	59
3.4.2	Description textuelle	60
3.4.3	Diagramme de séquence.....	64
3.4.4	Diagramme de classe	67
3.4.5	Diagramme de composant.....	67
3.5	Conclusion.....	68
Chapitre 4.....		69
4.1	Introduction :	69
4.2	Environnement et développement	69
4.2.1	LucidChart.....	69
4.2.2	Creately	69
4.2.3	Flutter.....	70
4.2.4	Visual Studio Code	70
4.2.5	MongoDB.....	71

4.2.6	Canva	71
4.2.7	JSON.....	71
4.3	Langages de programmation	72
4.3.1	Python	72
4.3.2	JavaScript	72
4.3.3	Dart	72
4.4	Algorithmes utilisés :	72
4.4.1	L’algorithme de recommandation collaborative a base mémoire :	73
4.4.2	L’algorithme de recommandation a base contenue :	73
4.4.3	L’algorithme de recommandation a base confiance explicite :.....	74
4.4.4	L’algorithme de recommandation a base confiance implicite :	75
4.4.5	Cosinus similarité :	75
4.4.6	Jaccard similarité :	76
4.4.7	Broda technique de vote :.....	76
4.5	Evaluation et Analyse	77
4.5.1	Les métriques d’évaluation :	77
4.5.2	Analyse des résultats.....	83
4.6	Présentation de l’application	85
4.6.1	Interface de Démarrage.....	85
4.6.2	Interface d’inscription.....	86
4.6.3	Interface d’authentification	87
4.6.4	Interface de l’accueil.....	88
4.6.5	Interface de détaille de POI.....	89
4.6.6	Interface de recherche Ami	90
4.6.7	Interface de recherche POI.....	91
4.6.8	Interface des catégories.....	92
4.6.9	Interface Profile d’utilisateur.....	93
4.7	Conclusion.....	94
	Conclusion Générale	95
	Annexe A	97

Annexe B	98
Bibliographie	99

Introduction Générale

Dans une société où l'information est de plus en plus abondante et parfois écrasante, il est crucial de pouvoir filtrer et recommander des contenus pertinents et captivants. Les systèmes de recommandation (SR) jouent un rôle essentiel dans cette dynamique, en tant que boussoles numériques qui orientent les utilisateurs à travers un vaste environnement de données mondial.

Dans l'écosystème numérique actuel, où l'abondance d'informations peut s'avérer accablante, les systèmes de recommandation spécialisés dans les points d'intérêt (POI) émergent comme des outils essentiels. Ils servent de phares dans le paysage numérique, guidant les utilisateurs vers des expériences significatives en s'appuyant sur une analyse approfondie de leurs données d'enregistrement. Ces systèmes ne se contentent pas de filtrer le superflu, ils anticipent activement les préférences et les besoins des utilisateurs, leur proposant des lieux et des expériences qui résonnent avec leurs intérêts personnels et leurs comportements passés. Ainsi, les SR pour POI transforment la surcharge d'informations en une opportunité de découverte ciblée et enrichissante.

Toutefois, ces systèmes présentent des difficultés. Les données limitées et les difficultés de mise à l'échelle constituent des défis majeurs qui limitent l'efficacité des SR. Dans ce document, nous examinons ces difficultés en améliorant la suggestion de points d'intérêt dans notre application mobile. Nous avons inclus différents algorithmes sophistiqués, tels que des systèmes implicites basés sur la confiance et des recommandations basées sur le contenu, ainsi que des algorithmes couramment utilisés tels que l'algorithme du système de vote de Broda, les recommandations explicites basées sur la confiance et le filtrage collaboratif basé sur le voisinage.

Notre travail s'articule autour de quatre chapitres principaux, chacun abordant un aspect crucial des SR et de la recommandation des POI :

Chapitre 1 : Système de Recommandation et la Recommandation des POI

Ce chapitre pose les fondations théoriques des SR et explore les différentes approches utilisées pour la recommandation des POI, y compris le filtrage collaboratif, le filtrage basé sur le contenu, l'approche hybride et l'approche basée sur la connaissance.

Chapitre 2 : État de l'art de la recommandation et des Systèmes Multi-Agents (SMA)

Nous définissons ici le concept des SMA et leur intégration dans les SR, en mettant en lumière les défis et les perspectives de cette fusion. L'état de l'art de la recommandation et des SMA est également discuté, avec un accent sur leur application dans le domaine des POI.

Chapitre 3 : Conception et modélisation

Ce chapitre offre une vue conceptuelle de notre système, présentant les méthodes d'analyse et de conception utilisées, notamment l'UML, et détaillant les spécifications des besoins du système à travers divers diagrammes UML.

Chapitre 4 : Implémentation

Le dernier chapitre est dédié à l'implémentation de notre application mobile, "WanderWise". Nous y présentons les environnements de développement, les langages de programmation utilisés, et les algorithmes implémentés. De plus, nous décrivons les différentes interfaces qui composent notre application.

Chapitre 1

Système de Recommandation & La Recommandation des Points d'intérêt

1.1 Introduction

Au cours de ce chapitre, nous allons, tout d'abord, introduire les principales notions de base de recommandation. Nous commençons notre analyse en donnant une définition précise de ce qu'est un système de recommandation (SR) et en soulignant ses éléments clés. Par la suite, nous examinons les diverses méthodes de recommandation, telles que le filtrage collaboratif (FC), le filtrage à base de contenu (FBC), l'approche basée sur la connaissance (BC) et l'approche hybride. Nous étudions attentivement chacune de ces méthodes et les confrontons pour mieux appréhender leurs bénéfices et leurs limites.

Ensuite, nous examinons la spécification des Points d'Intérêt (POIs), en examinant les différentes méthodes et algorithmes employés pour suggérer des points d'intérêt pertinents aux utilisateurs. Finalement, nous terminons ce chapitre en élaborant une perspective sur l'avenir des systèmes de recommandation, en particulier en ce qui concerne les recommandations des POIs.

1.2 Système de Recommandation:

Un système de recommandation représente une méthode particulière de filtrage de l'information utilisée dans des situations où une multitude d'informations pourrait causer de la frustration chez l'utilisateur. La **Figure 1** illustre un exemple d'un system de recommandation.

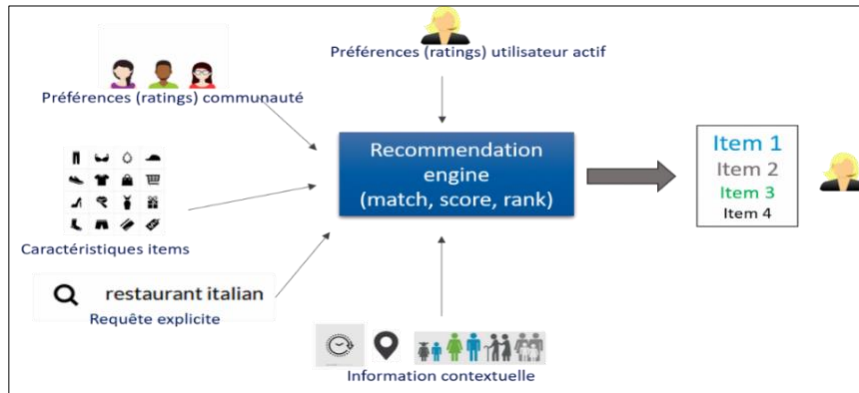


Figure -1- Exemple d'un système de recommandation (SR) [1].

Les systèmes de recommandation jouent un rôle important sur des sites Internet de premier plan tels qu'**Amazon**, **YouTube**, **Netflix**, **Yahoo**, **TripAdvisor**, **Last.fm**, entre autres. De plus, de nombreuses entreprises médiatiques développent et déploient actuellement des systèmes de recommandation dans le cadre des services qu'elles fournissent à leurs abonnés [2].

Par exemple, Netflix, le service de location de films en ligne, a attribué un prix d'un million de dollars à l'équipe qui a considérablement amélioré les performances de son système de recommandation. L'entreprise a intégré les propositions les plus pertinentes dans sa version de production du système de recommandation. Voici La **Figure 2** qui illustre l'interface Utilisateur d'un system de recommandation Netflix.

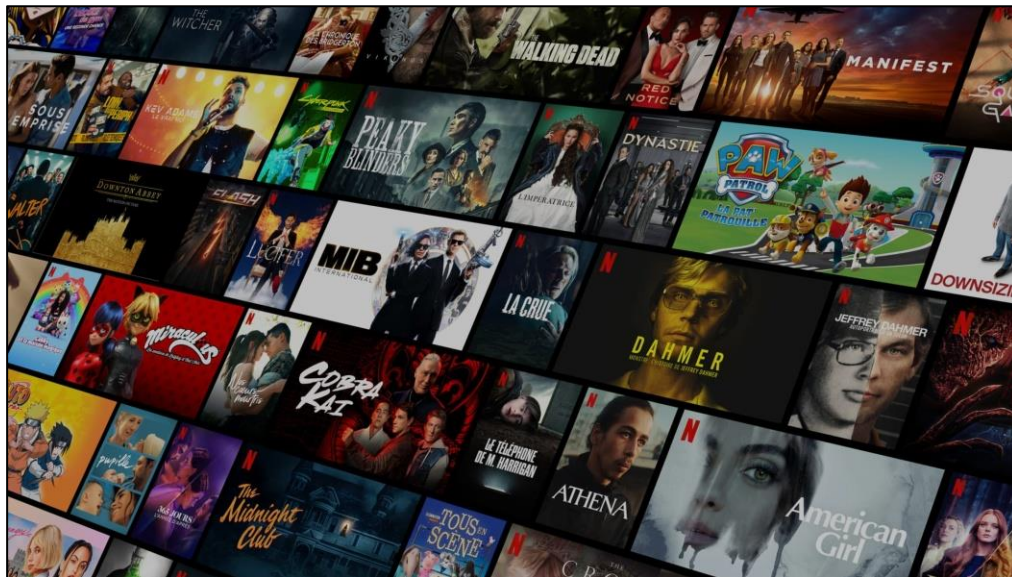


Figure -2- Interface Utilisateur de Netflix : Personnalisation et Recommandation [3]

Le principe essentiel d'un système de recommandation consiste à adapter les éléments ou le contenu à l'utilisateur en se basant sur ses interactions précédentes avec le système, tout en prenant en considération ses préférences. Il est possible que ces articles soient de différents types, comme des vidéos, de la musique, des vêtements, des applications, des documents, des articles, etc. [4]

Item : Les recommandations du système aux utilisateurs sont généralement désignées par le terme "Item". On peut parler de produits, de films, de restaurants, et ainsi de suite.

L'utilisateur : La personne qui utilise le système de recommandation est l'utilisateur. Il exprime son point de vue sur les divers éléments, puis il reçoit de nouvelles suggestions du système.

En général, la mise en place d'un système de recommandation se déroule en trois étapes :

- 1. Collecte de données concernant l'utilisateur** : Au cours de cette étape initiale, nous collectons des données concernant les préférences, les habitudes et les interactions de l'utilisateur avec les éléments du système. [5].
- 2. Mettre en place une Matrice d'Informations** : Nous exploitons ces informations afin de créer une matrice qui illustre les liens entre les utilisateurs et les éléments. Cette matrice renferme des données concernant les évaluations, les achats, les clics, et ainsi de suite. [5].
- 3. Recommandations extraites** : Finalement, nous avons la possibilité d'extraire une liste de recommandations sur mesure pour chaque utilisateur à partir de cette matrice. Ces suggestions reposent sur des méthodes qui étudient les liens entre les utilisateurs et les éléments.[5].

1.2.1 Collection des données :

Nous pouvons classer la collecte de données en deux types distincts :

1.2.1.1 Collecte explicite

Les utilisateurs sont encouragés à donner leur avis sur des éléments en utilisant différentes méthodes de rétroaction :

- **Un système de notation :**
Peut être utilisé pour donner une évaluation précise à un produit, un service ou un contenu, comme par exemple une grille de notation à 5 étoiles.
- **Questionnaires de satisfaction :**
Les questionnaires de satisfaction offrent la possibilité de collecter des avis plus précis et organisés. Par exemple, suite à un achat en ligne, il est possible de demander aux utilisateurs de donner leur avis sur leur expérience globale.
- **La communication sociale :**
Les utilisateurs ont la possibilité d'exprimer leur appréciation sur les réseaux sociaux en utilisant des fonctionnalités comme le bouton "commentaire".

1.2.1.1 Collecte Implicite :

Les interactions des utilisateurs sur le système sont à la base de la collecte implicite. Par exemple, le nombre de personnes qui visitent une page, le nombre de vues sur une vidéo, le temps consacré à une section spécifique.

1.3 Les Approches de Système de recommandation :

Il y a quatre principales catégories de systèmes de recommandation, chacune présentant ses propres bénéfices et difficultés :

- 1. L'approche du filtrage collaboratif :** Les recommandations sont élaborées en fonction des préférences des utilisateurs et de ses pairs.
- 2. Filtrage basée sur le contenu :** La méthode de filtrage basée sur le contenu consiste à fournir des recommandations en fonction du profil de l'utilisateur et des caractéristiques des éléments recommandés.
- 3. L'approche basée sur la connaissance :** les recommandations sont élaborées en fonction des spécifications de l'utilisateur et des caractéristiques des éléments suggérés.
- 4. L'approche hybride :** L'approche hybride combine les approches précédentes. voici La **Figure 3** qui illustre les Principales catégories de système de recommandation.

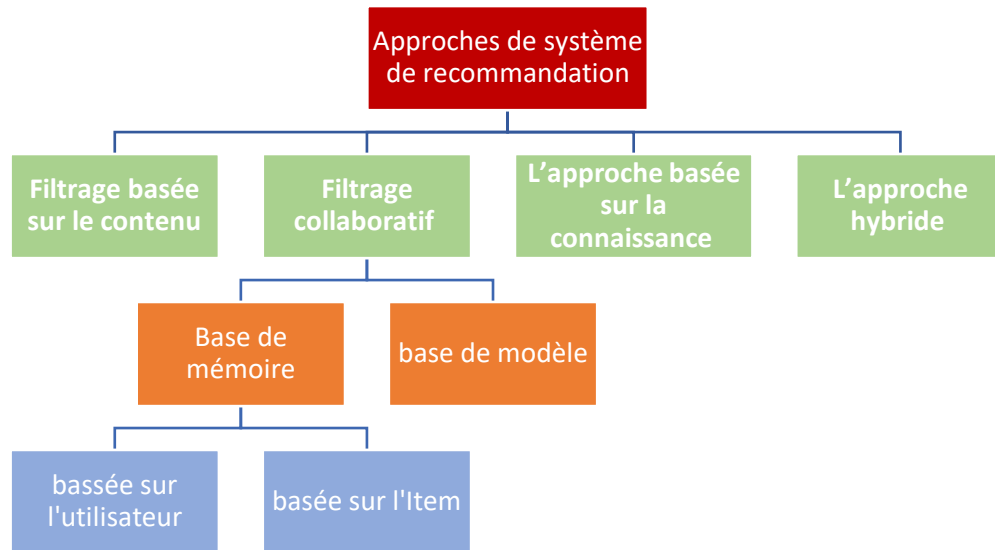


Figure -3- Principales catégories de système de recommandation (SR) [6]

1.3.1 L'approche du filtrage collaboratif:

La méthode de filtrage collaboratif se base sur la compréhension des préférences de l'utilisateur actif (d'une personne à qui on souhaite faire une recommandation), c'est-à-dire les éléments qu'il a le plus consultés par le passé, ainsi que les préférences d'une grande communauté d'utilisateurs.

Le filtrage collaboratif repose sur deux paradigmes :

- Des utilisateurs similaires expriment des préférences similaires et donnent des notes similaires (approche user-based). On prédit la préférence d'un utilisateur A pour un item X sur base des ratings donnés dans le passé par des utilisateurs ayant les mêmes intérêts qu'A à l'item X.
- Des items similaires reçoivent des ratings similaires. Pour prédire le rating que donnerait l'utilisateur A à l'item X, on se base sur les ratings donnés par cet utilisateur à des items similaires à X (approche item-based). Voici la **Figure 4** qui illustre le filtrage collaboratif.

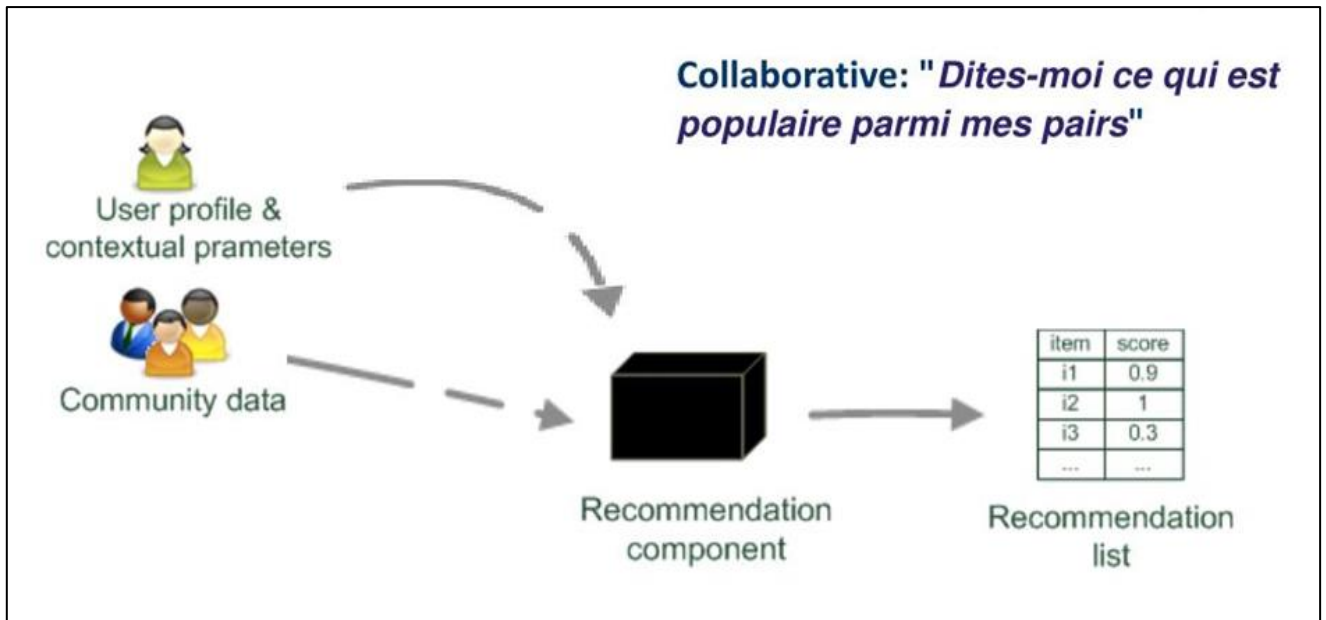


Figure -4- Filtrage collaboratif [7]

1. Quels sont les types de données ?

- **Les évaluations claires** : On demande à l'utilisateur de donner une évaluation à un élément (une notation allant de 1 à 5 étoiles est fréquemment employée). Il est particulièrement difficile d'obtenir ces évaluations explicites.
- **Les évaluations indirectes** : La note des éléments n'est pas attribuée par l'utilisateur, mais elle est calculée en fonction de certaines observations telles que le nombre de clics, le temps de consultation de la page, etc. Dans cette situation, la note obtenue est une estimation approximative de l'intérêt de l'utilisateur pour un objet.
- **Les notes en binaire** : Il est évident que la préférence positive ou négative est exprimée. Par exemple, j'aime/je n'aime pas.
- **Les évaluations unies** : Seule la préférence favorable est manifestée. Par exemple, l'achat d'un livre {Oui, Non}. Voici La **Figure 5** qui illustre Une illustration d'une matrice utilisateurs-items avec des évaluations claires.

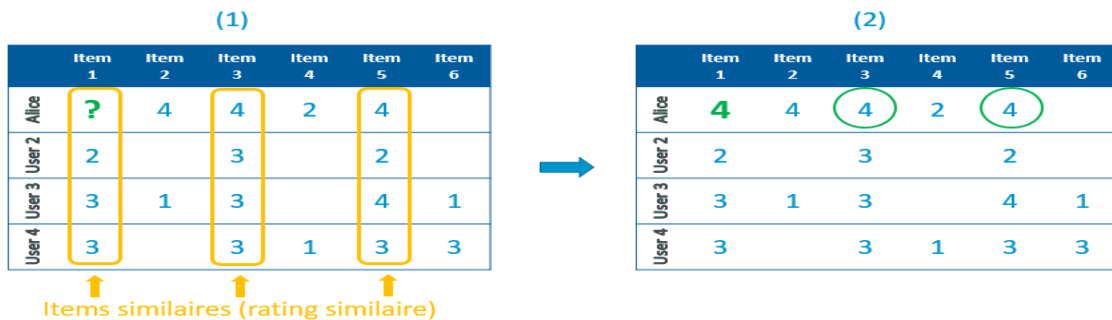


Figure -5- Une illustration d'une matrice utilisateurs-items avec des évaluations claires. [8]

2. Comment recueillir ces informations ?

Il y a deux situations possibles :

- L'utilisateur a accès à son compte en ligne. On peut donc facilement connecter les observations recueillies lors de l'utilisation du site internet à son compte et établir un profil utilisateur.
- Le compte de l'utilisateur n'est pas connecté. Les cookies ont la capacité d'identifier indirectement l'utilisateur et de le connecter à son compte, ou de créer un profil anonyme qui suit l'utilisateur lors de ses interactions futures. Toutefois, ces informations ne sont pas totalement fiables : le dispositif utilisé peut être utilisé par plusieurs individus, l'utilisateur peut se connecter à plusieurs appareils, l'utilisateur peut se connecter à un **VPN**, sans oublier les problèmes de respect du **RGPD** (Règlement Général sur la Protection des Données) que de telles pratiques peuvent engendrer.

Afin de faire une recommandation d'articles à un utilisateur actif (à qui on souhaite faire la recommandation), il est nécessaire de prévoir son niveau d'intérêt pour les articles pour lesquels il n'a pas d'évaluations. On présente à l'utilisateur les éléments qui ont obtenu les scores les plus élevés. Il existe deux méthodes, qui sont :

- **Filtrage collaboratif par le voisinage (neighbor-based collaborative filtering) :**

Cette méthode utilise un système de notation pour prédire les préférences d'un utilisateur en tenant compte des préférences d'un utilisateur similaire, ou du « voisin » [9].

Il existe deux façons de calculer les préférences ici, le FC basé sur l'utilisateur et le FC basé sur l'item.

Le FC à base de mémoire (voisin) souffre essentiellement de deux problèmes :

- a. Le passage à l'échelle et le manque de données. Le passage à l'échelle fait que le temps de calcul est énormément long lors de son application sur des données nombreuses.
- b. Le manque de données vient du fait qu'il n'est pas toujours facile de trouver assez de notes communes entre les utilisateurs, particulièrement sur des grandes listes d'items [10].

- **User-based :**

- Les utilisateurs similaires à A sont recherchés.
- Des éléments à A sont recommandés en effectuant une moyenne pondérée des notes de ces utilisateurs.
- En général, on se restreint à k utilisateurs.
- Les fonctions de similarité sont à calculer sur les lignes de la matrice des notes.

- **Item-based :**

- Afin de déterminer la note attribuée par A à l'élément I, on recherche un groupe d'éléments similaires à I, que A a évalué, et on calcule une moyenne pondérée.
- On calcule les similarités en se basant sur les colonnes.

- **Filtrage collaboratif User-Based :**

Les utilisateurs similaires partagent des évaluations similaires pour des éléments spécifiques.

Nous recherchons un groupe d'utilisateurs similaires à l'utilisateur cible, nous calculons la similarité entre cet utilisateur et tous les autres, et nous en conservons quelques-uns.

Notation pour la matrice :

- Soit $R = r_{uj}$ La matrice $m \times n$

- l_u Regroupe tous les indices des éléments enregistrés par u .
- L'ensemble noté par u et v est : $l_u \cap l_v$.
- Exemple : $l_u = \{1,3,5\}$, $l_v = \{1,2,3,4\}$, $l_u \cap l_v = \{1,3\}$.

Étape 01 : Similarité (coefficient de corrélation de Pearson)

- On calcule d'abord la moyenne des notes de u , μ_u :

$$\mu_u = \frac{\sum_{k \in I_u} r_{uk}}{|I_u|} \quad \forall u \in \{1 \dots m\} \quad (1.1)$$

- Puis la similarité elle-même :

$$\text{Sim}(u, v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u) \cdot (r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}} \quad \forall (u, v) \in \{1 \dots m\}^2 \quad (1.2)$$

En général, le PCC est calculé avec des μ uniquement sur les éléments de $I_u \cap I_v$.

- Une simplification informatique et programmatique est envisageable.
- Il est possible de maintenir le top-k des utilisateurs avec le PCC le plus performant.
- Cependant, le nombre de ratings pour un élément donné peut varier considérablement.
- Les top-k sont recherchés pour chaque élément.

Étape 2 : calcul de la note

De manière simple, on effectue une moyenne pondérée des notes des utilisateurs.

Éventuellement, les k les plus proches sont pris en compte. Le poids est attribué aux **PCC** entre les utilisateurs. Le problème réside dans les échelles de notes individuelles. La solution consiste à réaliser des évaluations centrées.

$$s_{uj} = r_{uj} - \mu_u \quad \forall u \in \{1 \dots m\} \quad (1.3)$$

Une évaluation pondérée est calculée en fonction de ces notes (positives ou négatives).

- La note finale prédite \hat{r}_{ij} est calculée en ajoutant la moyenne :

$$\hat{r}_{ij} = \mu_u + \frac{\sum_{v \in P(j)} \text{Sim}(u, v) \cdot s_{vj}}{\sum_{v \in P(j)} |\text{Sim}(u, v)|} \quad (1.4)$$

On cherche \hat{r}_{31} et \hat{r}_{36} dans l'exemple suivant :

Item-Id→	1	2	3	4	5	6	Mean Rating	Cosine (i, 3) (user-user)	Pearson (i, 3) (user-user)
User-Id ↓									
1	7	6	7	4	5	4	5.5	0.956	0.894
2	6	7	?	4	3	4	4.8	0.981	0.939
3	?	3	3	1	?	?	2	1.0	1.0
4	1	2	2	3	4	4	2.5	0.789	-1.0
5	1	?	1	2	3	3	2	0.645	-0.817

Tableau -1- Exemple UBCF

$$\text{Cosine}(1,3) = \frac{6 * 3 + 7 * 3 + 4 * 1 + 5 * 1}{\sqrt{6^2 + 7^2 + 4^2 + 5^2} \times \sqrt{3^2 + 3^2 + 1 + 1}} = 0.956$$

$$\text{Pearson}(1,3) == \frac{(6 - 5.5) \times (3 - 2) + (7 - 5.5) \times (3 - 2) + (4 - 5.5) \times (1 - 2) + (5 - 5.5) \times (1 - 2)}{\sqrt{1.5^2 + 1.5^2 + (-1.5)^2 + (-0.5)^2} \times \sqrt{(-1)^2 + (-1)^2 + 1 + 1}} = 0.894$$

- Le signe peut être informatif : (dis)similarité
- Top-2 utilisateurs proches de u3 : u1 et u2

$$\hat{r}_{31} = \frac{7 \times 0.894 + 3 \times 0.939}{0.894 + 0.939} \approx 6.49$$

$$\hat{r}_{31} = \frac{4 \times 0.894 + 4 \times 0.939}{0.894 + 0.939} \approx 4$$

- PCC plus discriminant que Cosinus

Il serait donc préférable de privilégier l'item 1, plutôt que le 6. Les prédictions indiquent que u3 appréciera davantage 1 et 6 que tous les autres.

○ **Filtrage collaboratif item-Based :**

On construit les voisinages en utilisant des éléments plutôt que des utilisateurs [10].

Comme précédemment, on effectue des évaluations centrées S_{ij} .

- On définit l'ensemble U_i des utilisateurs qui ont évalué i .
 - On calcule une similarité cosinus :

$$\text{cosA}(i, j) = \frac{\sum_{u \in U_i \cap U_j} s_{ui} \cdot s_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} s_{ui}^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} s_{uj}^2}} \quad (1.5)$$

Il est également possible d'utiliser un PCC entre colonnes, cependant les résultats sont généralement moins satisfaisants.

- Pour anticiper le score \hat{r}_{ut} de l'utilisateur u sur l'objet t , on effectue le calcul suivant :
 - Un ensemble $Q_t(u)$ contenant les top- k éléments, puis :

$$\hat{r}_{ut} = \frac{\sum_{j \in Q_t(u)} \text{cosA}(j, t) \cdot r_{uj}}{\sum_{j \in Q_t(u)} |\text{cosA}(j, t)|} \quad (1.6)$$

La prédiction est basée sur les propres notes de l'utilisateur.

- En ce qui concerne les films, les films les plus populaires seront souvent des films du même genre.
- On cherche \hat{r}_{31} et \hat{r}_{36} :

Item-Id →	1	2	3	4	5	6
User-Id ↓						
1	1.5	0.5	1.5	-1.5	-0.5	-1.5
2	1.2	2.2	?	-0.8	-1.8	-0.8

3	?	1	1	-1	-1	?
4	-1.5	-0.5	-0.5	0.5	-0.5	1.5
5	-1	?	-1	0	1	1
Cosine (1, j) (item-item)	1	0.735	0.912	-0.848	-0.813	-0.990
Cosine (6, j) (item-item)	-0.990	-0.622	-0.912	0.829	0.730	1

Tableau -2- Exemple IBCF

$$AdjustedCosine(1,3) = \frac{1.5 \times 1.5 + (-1.5) \times (-0.5) + (-1) \times (-1)}{\sqrt{1.5^2 + (-1.5)^2 + (-1)^2} \times \sqrt{1.5^2 + (-0.5)^2 + (-1)^2}} = 0.912 \quad (1.7)$$

- Les items 2 et 3 sont les plus proches de 1
- Les items 4 et 5 sont les plus proches de 6
- On reçoit:

$$r_{31}^{\wedge} = \frac{3 \times 0.735 + 3 \times 0.912}{0.735 + 0.912} = 3$$

$$r_{36}^{\wedge} = \frac{1 \times 0.829 + 3 \times 0.730}{0.829 + 0.730} = 1$$

- ❖ En général, les évaluations sont plus en accord avec les autres notes de l'utilisateur et sont plus en accord avec les listes de recommandations **UBCF** et **IBCF**.

6 Comparaison entre la CF basée sur les items et la CF basée sur les utilisateurs

- Il est souvent préférable d'utiliser les propres notes de l'utilisateur plutôt que les notes des autres utilisateurs similaires. Cependant, cela peut entraîner une préférence pour

des éléments trop (ou toujours) similaires à ceux déjà observés : moins de variété et d'émotion !

- Une stabilité accrue en **IB**. Plus d'utilisateurs que d'éléments, donc les éléments auront des ensembles d'utilisateurs en commun plus étendus. Et l'arrivée de plus d'utilisateurs dans le système entraîne une diminution des calculs de voisinage.
- Explicabilité meilleure en item-based :

"Because you watched "Secrets of the Wings," " [the recommendations are] <list> [11]

- En user-based, on ne connaît pas les utilisateurs similaires.

Les clients ayant acheté cet article ont également acheté



○ Filtrage collaboratif par modèle :

La méthode reposant sur l'acquisition d'un modèle. Son objectif est d'améliorer les performances observées avec la méthode des voisins. L'approche la plus fréquemment employée est la factorisation matricielle (**SVD** (Singular Value Decomposition), facteurs latents), qui permet de représenter la matrice initiale à l'aide de deux matrices de taille plus réduite, permettant ainsi de prédire les scores manquants.

Dans le cadre des approches basées modèles, la prédiction peut être faite de deux façons différentes :

- A partir de la prédiction fournie par le modèle lui-même, en construisant par exemple un modèle probabiliste pour l'estimation des valeurs de prédiction ou directement à partir du modèle.
- Ou bien, en regroupant les utilisateurs\ items par les méthodes de Clustering et par la suite, les méthodes basées mémoires (basées utilisateurs ou basées items) seront utilisées pour prédire les évaluations pour les items.

Dans cette méthode de filtrage collaboratif, différents algorithmes d'exploration de données et d'apprentissage automatique sont utilisés pour développer un modèle pour prédire l'évaluation d'un utilisateur d'un élément non estimé.

Quelques exemples de ces modèles sont les réseaux bayésiens [12],[13], les systèmes flous [14] [15] [16], les modèles de clustering [17], la factorisation matricielle [18] et les arbres de décision [19],[20] [21], les méthodes de la diagnostique de personnalité [22], ainsi que les méthodes de l'apprentissage automatique comme les algorithmes génétiques [23] [24], les Réseaux de Neurones Artificiels RNA (**Artificial Neural Networks ANN**) [25] , les Machines à Vecteur de Support (**Support Vector Machines SVM**) , les arbres de décision (**DecisionTree**) ,et récemment la tendance s'est tournée vers les méthodes de l'apprentissage en profondeur (Deep Learning).

L'apprentissage profond est également connu sous le nom d'apprentissage de caractéristiques en raison de sa capacité puissante à apprendre automatiquement les représentations de caractéristiques. De plus, les systèmes de recommandation peuvent utiliser les modèles en profondeur pour tirer parti des caractéristiques d'ordre élevé des données d'entrée.

En ce moment, grâce à l'évolution du concept d'apprentissage en profondeur, les chercheurs ont commencé à chercher à intégrer l'approche de l'apprentissage en profondeur dans les systèmes de recommandation basés sur le filtrage collaboratif.

Dans les systèmes de recommandation, divers algorithmes d'apprentissage en profondeur ont été employés, tels que l'approche basée sur les machines de Boltzmann restreintes (**RBM**).

Le modèle unifié **IID** (Indépendant et Identiquement Distribué), une extension de l'algorithme **RBM**, les machines de Boltzmann non restreintes utilisées pour la modélisation conjointe des utilisateurs et des éléments, ainsi que les Réseaux de Neurones Convolutionnels pour la recommandation de musique.

Les systèmes ont également adopté l'approche de l'apprentissage en profondeur pour être adaptés aux méthodes de classification des sentiments en se basant sur les commentaires écrits par les utilisateurs.

L'approche de l'apprentissage en profondeur permet d'acquérir une représentation de qualité supérieure pour chaque observation. Par la suite, cette représentation est utilisée

par les classificateurs de sentiments pour étudier les émotions des utilisateurs et améliorer la qualité des recommandations.

❖ **AVANTAGES ET INCONVÉNIENTS:**

- **Avantages du Filtrage Collaboratif :**
 - **Exploitation de la Sagesse de la Foule :** En utilisant les avis et les comportements de tous les utilisateurs, le filtrage collaboratif exploite la “sagesse de la foule” pour améliorer les recommandations [26].
 - **Adaptabilité aux Nouveaux Utilisateurs :** Contrairement à d’autres méthodes, le filtrage collaboratif peut fournir des recommandations même pour les nouveaux utilisateurs qui n’ont pas encore d’historique d’interaction [27].
- **Inconvénients du Filtrage Collaboratif :**
 - **Démarrage à Froid :** Lorsqu’un nouvel utilisateur rejoint le système, il n’a pas suffisamment d’historique pour des recommandations précises.
 - **Biais de Popularité :** Le filtrage collaboratif peut favoriser les articles populaires, car ils ont plus d’interactions et de données d’évaluation.

1.3.2 Filtrage basée sur le contenu:

Le fonctionnement des systèmes de recommandation basés sur le contenu se distingue du système collaboratif en utilisant exclusivement les préférences de l'utilisateur actif pour formuler les recommandations. Le but consiste à repérer des éléments similaires à ceux pour lesquels l'utilisateur a exprimé une préférence auparavant [28]. Chaque objet a des caractéristiques (contenu) qui sont utilisées pour formuler la recommandation [29].

Un système de filtrage basé sur le contenu suit deux fonctionnalités principales :

- La sélection des items et\ ou documents pertinents vis-à vis du profil de l'utilisateur.
- La mise à jour du profil de l'utilisateur en fonction du retour de pertinence fourni par l'utilisateur sur les items et\ ou documents qu'il a reçus. La mise à jour se fait par l'intégration des caractéristiques des items et\ ou des thèmes abordés dans les documents jugés pertinents. Ces caractéristiques fournissent

une description de l'objet, qu'elle soit exprimée de manière nominale (par exemple, le genre et l'auteur d'un livre) ou sous forme de texte libre. Voici la **Figure 6** qui illustre le filtrage basé sur le contenu.

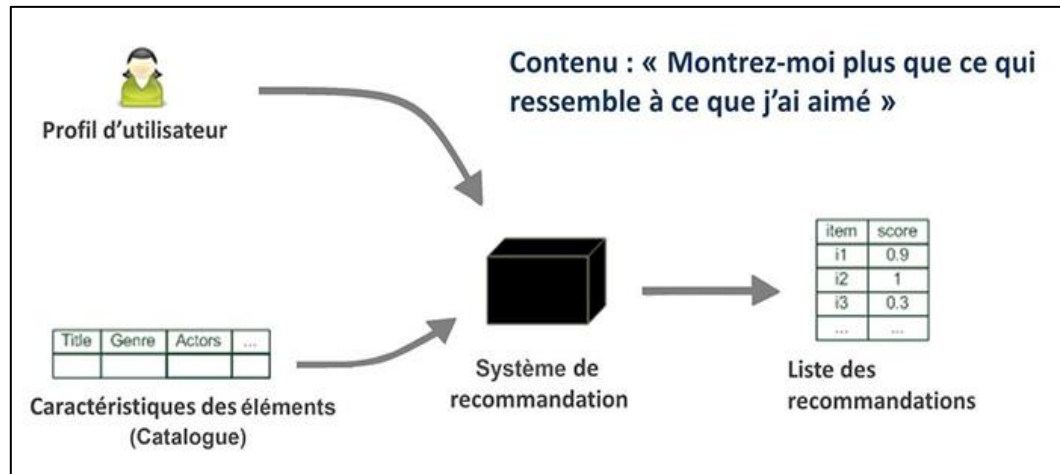


Figure -6- Filtrage basé sur le contenu. [30]

1.3.2.1 Types de données :

Le filtrage basé sur le contenu utilise des données spécifiques aux éléments (articles, produits, etc.) afin de suggérer des éléments similaires [31]. Voici les différentes catégories de données accessibles dans le cadre du filtrage basé sur le contenu :

1) Données Textuelles :

Les éléments sont évalués en utilisant des descriptions, des titres, des tags et des mots-clés qui leur sont associés [32]. Par exemple, lorsqu'on suggère des articles scientifiques, les mots-clés et les termes du résumé jouent un rôle essentiel en tant que données textuelles.

2) Données Structurées :

Les éléments sont caractérisés en utilisant des attributs structurés tels que les catégories, les genres, les caractéristiques techniques, etc. À titre d'exemple, lorsqu'il s'agit de recommander des films, les genres (action, comédie, drame) sont des informations organisées.

3) Données Multimédias :

Il est possible de faire appel à des images, des vidéos et des audios afin de décrire les éléments. Prenons par exemple la recommandation de musique, où les extraits audio ou les pochettes d'album sont des informations multimédias.

4) Données de Comportement Utilisateur :

La création d'un profil utilisateur repose sur les interactions passées d'un utilisateur avec des éléments (consultations, évaluations, achats).[33].

1.3.2.2 Etapes de la recommandation :

Les recommandations sont générées en utilisant les informations spécifiques des éléments (articles, produits, etc.) pour filtrer le contenu. Voici comment cela se fait :

a) Caractérisation du Contenu :

La caractérisation du contenu consiste à analyser les caractéristiques intrinsèques des éléments en utilisant le filtrage basé sur le contenu. Chaque élément est représenté en utilisant des données telles que les genres, les fonctionnalités, les descriptions, les mots-clés, etc. Par exemple, dans la recommandation de films, les genres (action, comédie, drame) et les mots-clés associés à un film sont utilisés pour caractériser son contenu.

b) Construction du Profil Utilisateur :

Le profil de chaque utilisateur est généré par le système en se basant sur ses interactions antérieures avec les éléments. Ces échanges peuvent englober des évaluations, des entretiens, des achats, et ainsi de suite. Par exemple, si un utilisateur a aimé des films d'action, le système construit un profil qui reflète cette préférence.

c) Comparaison entre le Profil Utilisateur et le Contenu des Éléments :

Le système établit une corrélation entre le profil de l'utilisateur et les caractéristiques des composantes. Il propose ensuite des éléments similaires à ceux que l'utilisateur a déjà appréciés. [34]

d) Génération des Recommandations :

Grâce à cette comparaison, le système élabore une liste de suggestions personnalisées pour l'utilisateur [35]. Par exemple, si un lecteur a apprécié un livre de science-fiction, le système suggérera d'autres ouvrages similaires.

❖ AVANTAGES ET INCONVÉNIENTS:

- **Avantages du Filtrage basé sur le contenu :**

- **Personnalisation :**

Le filtrage basé sur le contenu offre la possibilité de faire des recommandations sur mesure en exploitant les caractéristiques intrinsèques des éléments. Son attention est portée sur les préférences particulières de chaque utilisateur.

- **Indépendance des Données Utilisateur :**

À la différence de la sélection collaborative, la sélection basée sur le contenu ne repose pas sur les interactions entre les utilisateurs. Il est capable de proposer des conseils même aux nouveaux utilisateurs qui n'ont pas d'historique d'interaction.

- **Inconvénients du Filtrage Basé sur le Contenu :**

- **Biais de Contenu :** La possibilité de filtrer en fonction du contenu peut être restreinte par les caractéristiques disponibles pour décrire les éléments. Les recommandations peuvent manquer de diversité si les informations sont incomplètes ou biaisées.

- **Problème de Découverte de Nouveaux Contenus :**

Il peut être difficile de recommander des éléments complètement différents de ceux que l'utilisateur a déjà consommés grâce au filtrage basé sur le contenu. Il est possible que les suggestions manquent de sérendipité.[30]

1.3.3 L'approche Hybride :

Chaque approche collaborative et approche basée sur les contenus présente ses propres avantages et inconvénients. La mise en place d'un système de recommandation hybride afin de profiter des avantages de chacune de ces techniques tout en réduisant les désavantages.

Dans la réalité, On peut réaliser l'hybridation en combinant différentes techniques de différents types, où l'hybridation la plus courante est celle qui associe le filtrage collaboratif au filtrage basé sur le contenu. De plus, il est également envisageable de

combiner les diverses techniques similaires, comme par exemple en combinant les deux types de filtrage basé contenu avec l'algorithme Naïve de Bayes et l'algorithme des k-plus proches voisins, respectivement, ou les deux types de filtrage basé voisin et basé modèle, ou même le filtrage basé utilisateur avec le filtrage basé item [35].

Les problèmes rencontrés par les systèmes de recommandation classiques ont été abordés par l'approche hybride, qui sera exposé dans la section suivante.

Les chercheurs dans ce domaine ont largement exploré les problèmes clés tels que le démarrage à froid, la dispersion et l'évolutivité des données, ainsi que l'adaptabilité des systèmes aux changements de profil des utilisateurs (la stabilité par rapport aux problèmes de plasticité). Afin d'éviter les problèmes liés aux méthodes de recommandation, notamment le démarrage à froid pour un nouvel utilisateur et pour un nouvel item, on utilise généralement l'hybridation entre le **FC** et le **FBC** en utilisant le contenu des items, qui reflète le profil des utilisateurs, afin de repérer les similitudes entre eux et de créer les différentes communautés utilisées dans le filtrage collaboratif. Voici la **Figure 7** qui illustre le filtrage hybride.

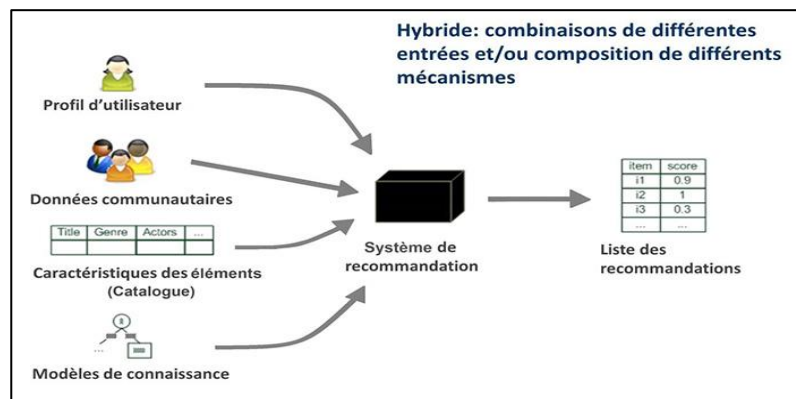


Figure -7- Filtrage hybride [7].

Ou encore, afin de repérer les similitudes entre les éléments les plus appréciés par les voisins de l'utilisateur actif, créés par un processus de filtrage collaboratif, et les éléments évalués par l'utilisateur pour lui fournir ceux qui lui sont intéressants.

Il est également possible de réaliser une hybridation puissante entre le **FC** et le filtrage basé sur la connaissance, principalement pour résoudre ce problème, car ce type de filtrage est puissant et ne présente pas les problèmes des autres méthodes.

Il existe différentes façons de combiner les méthodes de recommandation pour donner naissance à sept méthodes d'hybridation différentes proposées par Burke [36] :

1. Hybridation Pondéré (weighted).
2. Hybridation par Commutation (Switching).
3. Hybridation Mixte (Mixed).
4. Hybridation par Combinaison de caractéristiques (Feature Combination).
5. Hybridation en Cascade (Cascade).
6. Hybridation par Augmentation de Caractéristiques (Feature augmentation).
7. Hybridation de Méta-niveau (Meta-level).

1.3.4 L'approche basée sur la connaissance :

L'approche basée sur la connaissance dans les systèmes de recommandation exploite des informations sémantiques et des relations entre les éléments pour fournir des recommandations [37]. Voici la **Figure 8** qui illustre le filtrage basé sur la connaissance.

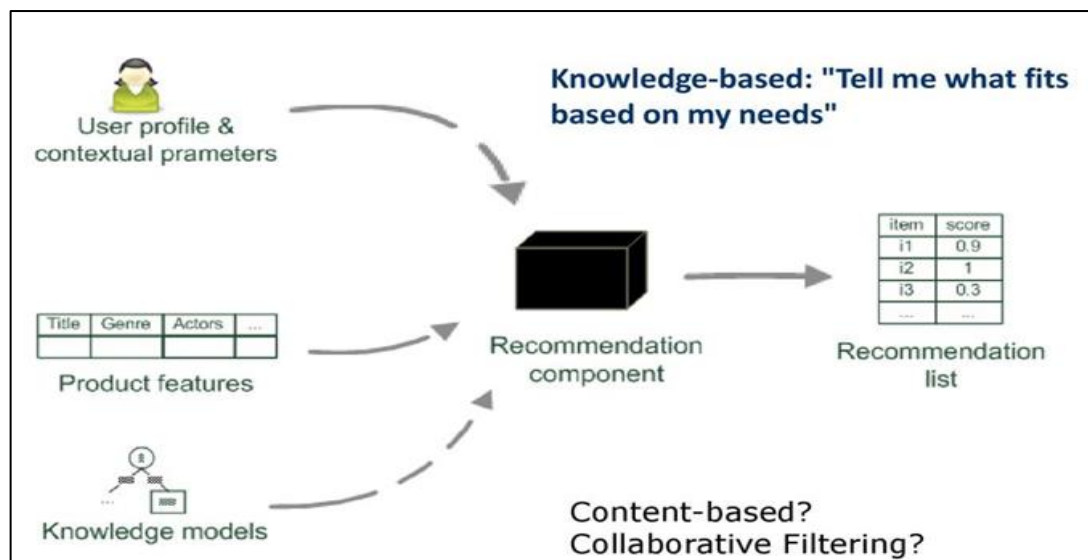


Figure -8- Filtrage basée sur la connaissance [38]

Voici quelques points clés à considérer :

i. Graphes de connaissances : Les graphes de connaissances, conçus en **RDF** (Resource Description Framework), offrent la possibilité de représenter des entités et leurs relations en utilisant des ontologies. Ils gagnent en popularité en matière de modélisation de l'information.[39] La **Figure 9** explique le graphe de connaissances.

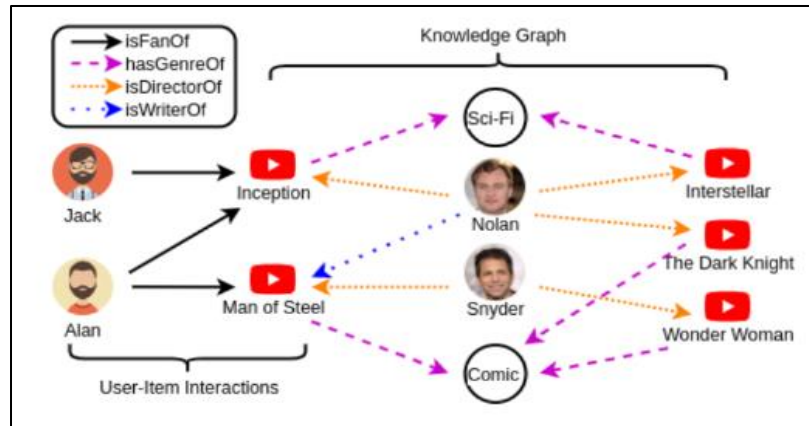


Figure -9- Graphes de connaissances [40]

ii. Les bénéfices des schémas de connaissance : Les liens et les relations entre les utilisateurs et les éléments sont plus représentés par ces éléments.

Les systèmes de recommandation basés sur des contraintes sont établis sur cette base solide.

iii. Systèmes de recommandation basés sur des contraintes : Ces systèmes font appel à une expertise approfondie afin de repérer des suggestions pertinentes. Quand ils sont associés à des graphiques de connaissances, ils présentent des bénéfices en ce qui concerne les ensembles de contraintes.[41]

Problèmes :

- Nécessite une ingénierie des connaissances importante.
- Savoir quelles caractéristiques sont importantes.
- Les caractéristiques doivent être accessibles.
- Pour les films : toujours le problème du contenu.

1.4 Points d'intérêt:

La cartographie et l'analyse spatiale font fréquemment appel à l'expression « Point d'Intérêt » (**POI**). Mais quelle est sa signification précise ? Examinons plus en profondeur cette définition. Un point d'intérêt désigne un endroit précis. Quelqu'un peut trouver bénéfique ou captivant. Les éléments de carte comprennent des titres détaillés, des coordonnées géographiques ainsi que des adresses. Il est possible que les Points d'Intérêt (POIs) soient des entreprises, des monuments ou des attractions qui sont pertinents pour un public spécifique. Les individus font appel aux services de cartographie et aux dispositifs de navigation afin d'obtenir davantage d'informations sur des endroits précis.

Le concept de **POI** ne se limite pas aux lieux physiques et tangibles. Dans le domaine numérique, ils peuvent également représenter des destinations en ligne. Par exemple, les liens d'un site web peuvent être considérés comme des points d'intérêt dans la structure du site. L'identification et l'utilisation des points d'intérêt sont essentielles pour de nombreux secteurs d'activité.

Du marketing au tourisme, en passant par marketing, le tourisme, l'urbanisme et l'immobilier, les points d'intérêt servent d'indicateurs fiables de l'intérêt et de l'activité humaine. Ils aident les entreprises à identifier les zones de forte activité de la clientèle, permettent aux touristes de découvrir des attractions populaires et aident les urbanistes à concevoir des villes. Par conséquent, la compréhension des points d'intérêt et de leur utilisation peut considérablement améliorer le processus de prise de décision dans ces secteurs. [42]

1. Quel est l'exemple de données POI ?

Supposons que vous utilisiez une application à navigation pour localiser un restaurant proche. Le **POI** contenu dans l'application peut regrouper des informations telles que le nom du restaurant, son adresse, son numéro de téléphone, le type de cuisine, les commentaires des utilisateurs, voire même des photos. Cette vaste compilation de données aide les utilisateurs à prendre des décisions informées sur leur destination.

2. La collecte d'informations sur les points d'intérêt (POI) ?

La collecte d'informations sur les points d'intérêt (POI) se fait à l'aide de plusieurs techniques, notamment [43] :

- **Web scraping** : Cette méthode consiste à extraire des informations à partir de sites web et de fichiers de référence. Les données pertinentes sont collectées en parcourant des pages web et en récupérant les détails des **POIs**.
- **Crowd-sourcing** : Le crowd-sourcing repose sur l'engagement des utilisateurs soumettre de nouveaux **POIs** ou à actualiser les existants. Les utilisateurs contribuent en partageant leurs connaissances sur des lieux spécifiques.
- **Collaboration avec des entreprises et des organisations** : Travailler en collaboration avec des entreprises et des organisations permet d'accéder à leurs données de localisation. Ces partenariats facilitent l'accès à des informations précises sur les **POIs**.
- **Saisie manuelle des données** : Dans certains cas, les informations sur les **POIs** sont saisies manuellement dans une base de données. Cela peut inclure des détails tels que les coordonnées géographiques, les heures d'ouverture et les descriptions.

❖ Exemples Point d'intérêt:

Pour mieux comprendre la notion de Point d'Intérêt (**POI**), considérez ces exemples spécifiques :

▪ **Restaurants et cafés** :

Les restaurants et les cafés sont très prisés pour les repas et les boissons. Les informations relatives à ces établissements comprennent leur nom, leur localisation, leurs horaires d'ouverture, les choix de menu et le style de cuisine.

▪ **Commerces de détail** :

Les **POIs** offrent également de nombreux avantages aux entreprises de vente au détail, en particulier aux centres commerciaux et aux magasins. L'attrait des consommateurs pour un centre commercial ou une boutique de marque peut varier en fonction de différents éléments tels que l'emplacement, les horaires d'ouverture et les promotions de produits.

▪ **Institutions publiques :**

Les établissements publics, tels que les écoles, les bibliothèques et les bureaux gouvernementaux, sont des destinations populaires pour les nouveaux résidents.

Les points d'intérêt incluent l'emplacement, les heures d'ouverture et les services disponibles.

1.4.1 Composants de POI :

Un point d'intérêt (**POI**) comprend généralement plusieurs composants clés :

1.4.1.1 Coordonnées géospatiales :

- Les coordonnées de latitude et de longitude du lieu, permettant un positionnement précis sur les cartes.
- Elles indiquent l'emplacement exact du **POI**.

1.4.1.2 Nom et description :

- Le nom du **POI** et une brève description qui fournit du contexte.
- Cela permet aux utilisateurs de comprendre ce que représente le **POI**.

1.4.1.3 Adresse :

- L'adresse textuelle physique du lieu, permettant aux utilisateurs de s'y rendre.
- Elle inclut des informations telles que le numéro de rue, la ville et le code postal.

1.4.1.4 Informations de contact :

- Les numéros de téléphone, les sites web et d'autres détails de communication.
- Cela permet aux utilisateurs de contacter le **POI** ou d'obtenir plus d'informations.

1.4.1.5 Catégorie et balises :

- La catégorisation du **POI**, telle que restaurants, hôtels, stations-service, etc.
- Cela aide à organiser et à rechercher les **POIs** en fonction de leur type.

1.4.1.6 Photos et avis :

- Des visuels et des avis générés par les utilisateurs qui offrent des informations aux utilisateurs.
- Cela permet aux utilisateurs d'avoir un aperçu visuel du lieu et de connaître l'expérience d'autres personnes.

1.4.1.7 Heures d'ouverture :

- Les heures d'ouverture, garantissant que les utilisateurs visitent le lieu lorsqu'il est ouvert.
- Cela évite les déplacements inutiles en dehors des heures d'ouverture.

1.4.2 Types de points d'intérêt (POI) :

Les POIs se déclinent en plusieurs types, notamment :

1.4.2.1 POIs commerciaux :

- Il s'agit d'entreprises telles que les restaurants, les hôtels, les centres commerciaux et les stations-service.
- Les utilisateurs peuvent être attirés par un centre commercial ou une station-service en fonction de facteurs tels que l'emplacement, les heures d'ouverture et les offres de produits.

1.4.2.2 POIs récréatifs :

- Ils comprennent des parcs, des musées, des théâtres et d'autres lieux de divertissement.
- Les utilisateurs peuvent explorer des lieux culturels, des événements et des attractions dans leur région.

1.4.2.3 POIs de transport :

- Ils incluent les aéroports, les gares, les arrêts de bus et les parkings.
- Les utilisateurs peuvent trouver des informations sur les transports publics et les services de mobilité.

1.4.2.4 Monuments :

- Il s'agit de sites historiques, de monuments et de structures architecturales uniques.
- Les utilisateurs peuvent découvrir des lieux emblématiques et en apprendre davantage sur l'histoire.

1.5 Recommandation des POIs :

Les progrès de la technologie et du tourisme ont entraîné l'essor des réseaux sociaux basés sur la localisation (**LBSN**) et des recommandations de points d'intérêt (**POI**).

Les réseaux sociaux basés sur la localisation (**LBSN**) et les recommandations de points d'intérêt ont connu un essor considérable. Les **LBSN** utilisent les informations de localisation pour faciliter la création de réseaux sociaux (Fusco, Abbas, Micheal et Aloudat, 2012) [44].

Les consommateurs peuvent partager leurs destinations préférées et en ajouter de nouvelles à leurs réseaux sociaux. Les **LBSN** sont utilisés par des applications de médias sociaux comme Foursquare et Facebook pour permettre aux utilisateurs d'explorer des lieux et de proposer des points d'intérêt (Gottapu&Monangi, 2017) [45].

Les systèmes de recommandation de points d'intérêt proposent des destinations basées sur le comportement et les activités de l'utilisateur. La combinaison des caractéristiques de comportement de vérification des applications **LBSN** avec les systèmes de recommandation de points d'intérêt améliore les services basés sur la localisation et aide à la fois les consommateurs et les entreprises.

Les utilisateurs peuvent découvrir de nouvelles régions, tandis que les entreprises peuvent cibler leurs clients et accroître leurs avantages grâce à la distribution commerciale. Les **LBSN** peuvent utiliser un système de recommandation de **POI** pour attirer des clients plus ciblés.

1.5.1 Recommandation des POIs avec GPS Data :

La suggestion de **POI** est étroitement liée au mouvement humain. Auparavant, les études se concentraient sur les données mobiles, à savoir les données **GPS** des téléphones

portables. Les téléphones portables sont couramment utilisés pour faciliter la vie des gens. Ils peuvent être considérés comme des capteurs mobiles de l'homme en général. Un utilisateur a presque toujours son téléphone portable sur lui. Les capteurs mobiles peuvent fournir des informations détaillées sur les mouvements humains. Les données **GPS** des téléphones portables comprennent souvent des points **GPS** horodatés et des activités mobiles telles que la musique et les films. La recherche sur la mobilité humaine a conduit au développement d'applications basées sur la localisation, telles que les recommandations de points d'intérêt.

Par exemple, **Figure 10** montre comment les informations d'enregistrements notamment le nom d'utilisateur, le POI, l'horodatage d'enregistrement et la position géographique sur la carte, sont collectés dans Foursquare.

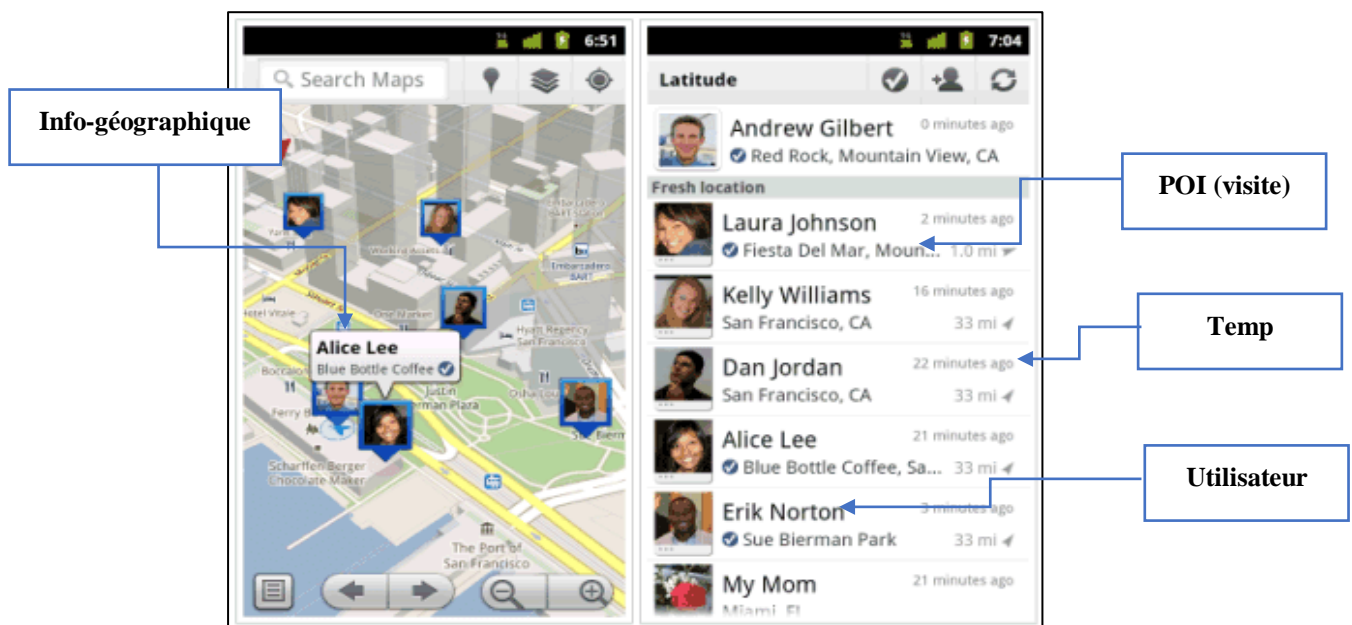


Figure -10- La collecte des informations d'enregistrement dans Foursquare.[46]

1.5.2 Recommandation des POIs avec LBSN :

En raison de la facilité d'accès à de nombreuses informations sur l'activité des utilisateurs dans les **LBSN**, un certain nombre d'études récentes ont utilisé les données des **LBSN** pour examiner les suggestions de points d'intérêt. En utilisant les informations spatiales, sociales, temporelles et de contenu des **POI**, la majorité des recherches antérieures ont tenté d'améliorer la recommandation de **POI** dans les **LBSN**.

Les **LBSN** ont été dotés d'une fonction de suggestion de **POI** grâce à Ye et al [47]. Étant donné les fortes corrélations entre les liens sociaux et la distance géographique démontrées dans les recherches antérieures, la majorité des travaux actuels sur la recommandation de **POI** dans les **LBSN** se concentrent sur l'utilisation des caractéristiques géographiques et sociales pour augmenter l'efficacité de la recommandation.

Ces recherches utilisent principalement les liens sociaux et les effets géographiques de manière indépendante, en combinant les résultats pour créer un modèle fusionné.

1.5.3 Facteurs d'influence :

Les **POIs** sont recommandés en fonction de divers facteurs tels que le facteur social, le facteur géographique, le facteur temporel et le facteur séquentiel. La **Figure 11** montre la Proposition d'un cadre pour le système de recommandation de **POI** dans les **LBSN**.

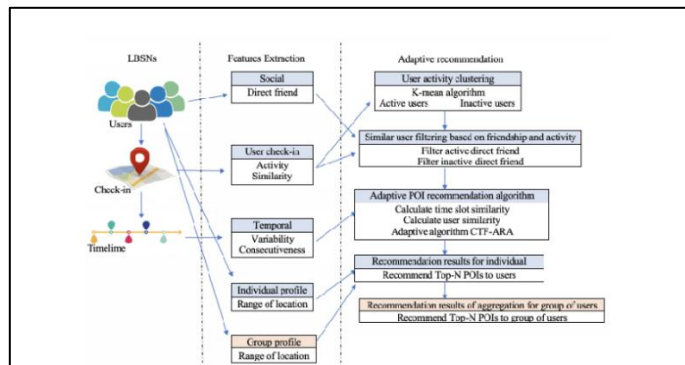


Figure -11- Proposition d'un cadre pour le système de recommandation de **POI** dans les **LBSN** Pour les individus et les groupes sociaux.[48]

Maintenant, nous allons vous donner une description approfondie de chaque élément.

1) L'influence sociale :

L'influence sociale joue un rôle crucial dans la recommandation des **POIs**. Cette influence repose principalement sur l'idée que les amis d'un utilisateur partagent des préférences et des intérêts similaires, ce qui leur permet de visiter les mêmes endroits. [50]

2) L'influence géographique :

Les comportements d'enregistrement check-in des utilisateurs envers les points d'intérêt sont grandement influencés par la proximité géographique des points

d'intérêt. On suppose que l'utilisateur a plus de chances de visiter des sites proches par rapport aux sites éloignés, donc l'impact des informations de géolocalisation sur les comportements de check-in a été largement utilisé et pris en compte dans les recommandations d'emplacement.[49]

3) L'influence temporelle :

Le facteur du temps joue un rôle crucial dans la proposition. Il mentionne que la destination d'un utilisateur est généralement liée au temps. Effectivement, l'impact du temps est crucial pour la proposition des **POIs**.

Les restrictions physiques sur l'activité de check-in entraînent l'émergence de schémas spécifiques qui représentent la corrélation entre les lieux et les informations de temps. Les utilisateurs ont l'habitude de fréquenter régulièrement certains lieux et de les quitter à une heure précise (par exemple, l'utilisateur se rend au travail ou au restaurant et fixe un emploi du temps). Ainsi, l'histoire de l'utilisateur est basée sur des séquences temporelles (séquences d'emplacement temporel) qui incluent l'heure dans chaque **POI**.[49]

4) L'influence séquentielle :

La régularité séquentielle du mouvement humain est observée [50]. C'est pourquoi des motifs séquentiels peuvent être extraits des séquences des lieux d'enregistrement des participants. Par exemple, la plupart des individus se rendent aux cinémas après les restaurants afin de se relaxer après le repas. L'importance de l'influence séquentielle sur l'activité d'enregistrement des utilisateurs a augmenté dans les recommandations visant à améliorer la recommandation des **POI**. La **Figure 12** illustre l'influence séquentielle.



Figure -12- Le comportement séquentiel de mouvement de trois utilisateurs.[51]

1.6 Conclusion

Dans ce chapitre, Nous avons débuté en fournissant une définition précise de ce qu'est un système de recommandation (**SR**) et en mettant en évidence ses éléments essentiels. Par la suite, nous avons étudié différentes approches de recommandation, comme le Filtrage collaboratif, le Filtrage à base de contenu, l'approche basée sur la connaissance et les approches hybride. Chaque méthode a été minutieusement examinée et confrontée afin de mieux comprendre leurs avantages et leurs limites. Par la suite, nous avons étudié les caractéristiques des Points d'Intérêt (**POIs**), en étudiant les diverses méthodes et algorithmes utilisés pour proposer des lieux d'intérêt pertinents aux utilisateurs.

Chapitre 2

Etat de l'art la recommandation & (SMA)

2.1 Introduction

Dans ce chapitre, nous allons examiner le domaine des Systèmes Multi-Agents (SMA) et leur utilisation dans les systèmes de recommandation. Nous commencerons par une présentation des SMA, en exposant en détail leurs principes essentiels et la façon dont ils peuvent être exploités afin d'améliorer les processus de recommandation. Nous serons étudiées les principales caractéristiques des SMA, telles que leur structure, leur communication et leur aptitude à résoudre des problèmes de manière collaborative. Cette partie soulignera les bénéfices des SMA et comment ils simplifient une approche de prise de décision distribuée et autonome.

En deuxième lieu, nous présentons un état de l'art sur les systèmes de recommandation qui utilisent les SMA. Cette section inclura une analyse critique des articles scientifiques pertinents, mettant en évidence les avancées accomplies et les difficultés rencontrées.

2.2 Agent

Avant de discuter de l'aspect des systèmes multi-agents, il est important de se focaliser sur les agents, qui constituent en quelque sorte les éléments essentiels de tout système multi-agents.

Tout d'abord, voici quelques définitions de ce qu'est un agent, d'après divers auteurs : Selon S. Russell et P. Norvig : « On appelle agent toute entité qui peut être considérée comme percevant son environnement grâce à des capteurs et qui agit sur cet environnement via des effecteurs. » [52].

Selon M. Wooldridge, « un agent est un système informatique situé dans un environnement donné, et qui est capable d'agir de manière autonome dans cet environnement en fonction des objectifs qui lui sont définis » [53].

Selon Jacques Ferber « Ferber propose une définition exhaustive de l'agent en tant qu'entité physique ou virtuelle :

- Capable d'agir dans son environnement,
- Pouvant communiquer directement avec d'autres agents.
- Mue par un ensemble de tendances qu'elle cherche à optimiser (objectifs individuels, fonction de satisfaction, voire de survie).
- Possédant des ressources propres.
- Capable de percevoir son environnement de manière limitée.
- Disposant d'une représentation partielle (voire aucune) de cet environnement.
- Dotée de compétences et d'offres de services.
- Éventuellement capable de se reproduire.
- Dont le comportement tend à satisfaire ses objectifs en tenant compte des ressources, compétences, perceptions, représentations et communications. » [54]

Dans la littérature, chacun a une définition d'un agent plus ou moins large. Selon la définition de M. Wooldridge, un agent est purement limité à une entité informatique [53]. Alors que la définition de S. Russell et P. Norvig est beaucoup plus ouverte et comprend les agents biologiques, contrairement à d'autres définitions restreintes au domaine de l'informatique ou du logiciel [52].

On peut également noter que l'on n'évoque pas les agents intelligents, mais uniquement les agents.

2.2.1 Définition Raffiné d'un Agent :

Un agent est une entité complexe, ancrée dans un environnement, dotée de la capacité de percevoir cet environnement à travers des capteurs sophistiqués. Il agit sur cet environnement via des effecteurs et poursuit activement un ou plusieurs objectifs définis.

Exemples Illustratifs d'Agents :

- Un robot d'exploration internet, méticuleusement indexant du contenu pour un moteur de recherche.
- Un virus biologique, cherchant activement à se reproduire dans son environnement (un animal ou un être humain) en utilisant les cellules et leurs ressources.
- Un être vivant complexe, avec des organes sensoriels avancés comme des capteurs, des membres supérieurs et inférieurs agissant comme des effecteurs, et ayant pour objectif fondamental la survie et la reproduction. Voici la **Figure 13** montre comment un agent interagit avec son environnement grâce à ses capteurs et ses effecteurs.

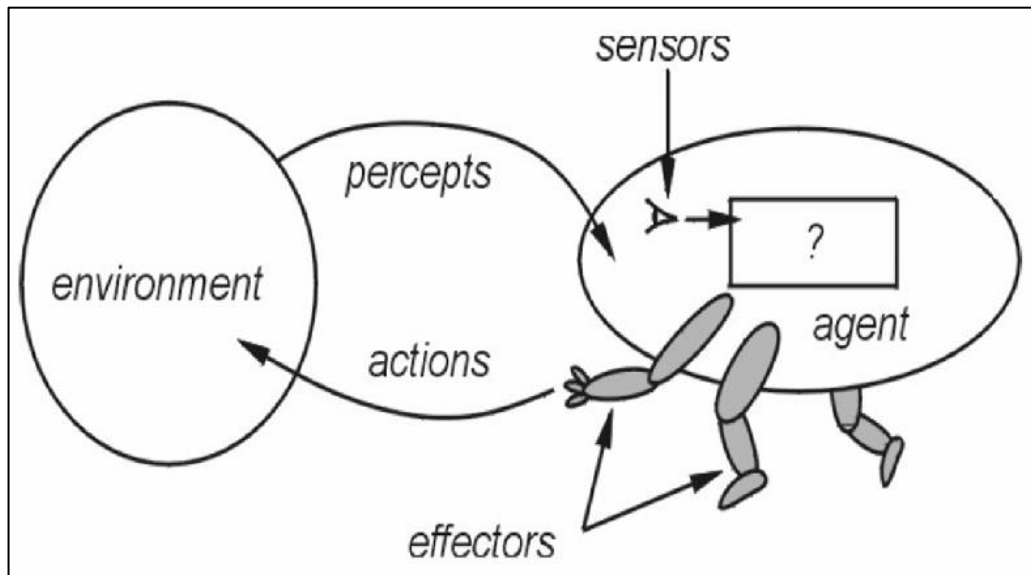


Figure -13-Un agent interagit avec son environnement grâce à ses capteurs et ses effecteurs. [55]

2.2.2 Distinction Entre Agents et Objets :

Il convient de noter qu'un agent ne peut pas être considéré comme un objet. La synthèse de l'exposé de M. Wooldridge se concentrera sur trois différences essentielles [53].

- La différence principale réside dans le fait qu'une entité externe a la capacité de manipuler directement l'état interne d'un objet (avec des attributs publics par exemple), tandis qu'on ne peut pas manipuler l'état interne d'un agent ni le contraindre à accomplir une fonction.
- La deuxième distinction réside dans le fait que le comportement d'un agent est autonome, contrairement à celui d'un objet qui est commandé.
- La troisième distinction porte sur la mise en œuvre des comportements. Pour un agent, la mise en œuvre ne dépend que de lui. Tandis que pour un objet, cela varie

en fonction du système auquel il appartient (par exemple, une plateforme logicielle) qui définit ses temps et ses durées d'exécution.

2.2.3 Vision Fonctionnelle :

➤ **Classification d'Agent:**

Évalue le degré d'intégration des facettes autonomie, environnement, interaction et organisation dans l'agent.

Types d'agents : autonomes, interactifs, sociaux.

➤ **Vision Décisionnelle :**

Évalue le degré de couplage à l'environnement.

Types d'agents : réactifs, hybrides, cognitifs.

Ces deux visions sont complémentaires, et les propriétés des agents sont étroitement liées à la manière dont ils représentent leur environnement. Une représentation élaborée de l'environnement augmente la capacité cognitive de l'agent, influençant ainsi ses actions et sa planification [53].

2.2.4 Types d'Agents :

- **Agent Cognitif :** Dispose d'une base de connaissances étendue liée à son domaine d'expertise et à la gestion des interactions. Intentionnel, avec des buts et des plans explicites pour atteindre ses objectifs [56].
- **Agent Réactif :** Perçoit l'environnement et agit de manière directe sans représentation symbolique. Répond au stimulus/action, sans mémoire ni capacité de prévision à long terme [57].
- **Agent Hybride :** Combine des capacités réactives et cognitives. Adapte son comportement en temps réel à l'évolution de l'environnement. Fusion de caractéristiques telles que la collaboration, l'autonomie et la capacité d'apprentissage [58].
- **Agent Intentionnel :** Les agents intentionnels, relevant de la catégorie cognitive, se démarquent par leur comportement téléonomique. Ils sont caractérisés par la possession d'objectifs clairs et d'une intention forte de les réaliser. Pour atteindre ces objectifs, ces agents mettent en œuvre des processus de planification, reflétant

ainsi une capacité à anticiper et à organiser leurs actions en fonction de leurs intentions.[59].

- **Agent Module** : Les agents modules, également de nature cognitive, se distinguent par leur comportement réflexe. Contrairement aux agents intentionnels, ils n'ont pas d'objectifs explicites. Cependant, leur utilité réside dans leur capacité à agir de manière réflexe, ce qui en fait des ressources d'information potentielles. Ces agents peuvent ainsi être sollicités par d'autres entités pour fournir des données contextuelles, participant ainsi à la dynamique d'interaction dans leur environnement.[60].
- **Agent Pulsionnel** : Les agents pulsionnels, appartenant à la catégorie réactive, sont spécifiquement adaptés au maintien d'objectifs définis. Leur comportement téléonomique les conduit à déclencher des réponses immédiates lorsque l'état de l'environnement s'écarte des objectifs préétablis. Cette réactivité caractéristique en fait des agents particulièrement aptes à maintenir une mission ou un objectif spécifique malgré les changements dans leur environnement.[55].
- **Agent Tropic** : Quant aux agents tropiques, faisant partie de la catégorie réactive, ils agissent de manière réflexe en réponse à des stimuli locaux sans posséder d'objectif prédéfini. Leur comportement automatique, dicté par les conditions immédiates de l'environnement, les place parmi les agents les plus basiques. Une évolution vers des agents tropiques dotés d'une mémoire, appelés agents hystérétiques, est envisageable pour renforcer leur capacité à tirer des enseignements de l'expérience passée.[61].

2.3 Un système multi-agents :

Un SMA est un ensemble d'agents qui interagissent entre eux dans un environnement commun. Il est possible que certains de ces agents soient des individus ou leurs avatars, ou même des machines mécaniques. Si trois agents sont présents, on parle plutôt d'interaction entre un homme et une machine ou d'une machine et une machine plutôt que de systèmes multi-agents.

La définition de J. Ferber [54] identifie les agents comme des entités artificielles capables de communication et d'action dans un environnement, et M. Woodbridge [53] souligne

l'importance fondamentale des interactions entre ces agents, les décrivant comme une sorte de société où la communication et l'influence sur l'environnement jouent un rôle central.

L'aspect distribué de l'intelligence, permettant des solutions locales à des problèmes globaux, est mis en évidence, notamment dans des domaines tels que le trafic urbain ou le développement de produits industriels. J. Ferber [54] explore également les théories sur l'émergence de structures au sein des SMA, soulignant l'influence des phénomènes chaotiques tels que l'effet papillon, qui, bien que rendant l'évolution des systèmes imprévisible, contribuent également à des phénomènes d'auto-organisation.

La répartition de la résolution de problèmes à travers l'environnement est discutée, mettant en lumière la nécessité d'une intelligence focalisée sur des aspects spécifiques. La relation dualiste entre les agents et l'environnement est abordée, distinguant les agents comme étant purement communicatifs ou situés, ces derniers possédant uniquement une représentation locale de l'environnement.

2.3.1 Défis et Perspectives dans l'Intégration des Systèmes Multi-Agents pour la Recommandation

Les difficultés mentionnées dans le contexte des systèmes de recommandation utilisant les systèmes multi-agents (SMA) peuvent se référer à divers défis rencontrés dans la recherche et l'application de ces technologies. Voici quelques exemples de difficultés potentielles :

- **Complexité de l'intégration** : L'intégration des SMA dans les systèmes de recommandation peut être complexe en raison de la nécessité de coordonner plusieurs agents autonomes ayant des objectifs différents.[62]
- **Gestion des données** : Les SMA doivent gérer et analyser de grandes quantités de données, ce qui peut poser des problèmes de performance et d'efficacité.[63]
- **Adaptabilité et apprentissage** : Les SMA doivent être capables de s'adapter et d'apprendre de nouveaux comportements en fonction des interactions avec les utilisateurs, ce qui peut être difficile à réaliser.[64]

- **Sécurité et confidentialité** : Assurer la sécurité des données et la confidentialité des utilisateurs dans les **SMA** est un défi majeur, surtout lorsque les recommandations sont personnalisées.[65]
- **Évaluation et validation** : Évaluer l'efficacité des recommandations fournies par les **SMA** et valider leur pertinence pour les utilisateurs finaux sont des tâches complexes. [66]

2.4 Etat de l'art la recommandation et SMA

Les systèmes de recommandation utilisant des systèmes multi-agents (**SMA**) constituent une avancée majeure dans le domaine de l'intelligence artificielle et de la personnalisation des services. Les **SMA**, composés d'individus libres qui peuvent communiquer et travailler ensemble, proposent une méthode dynamique et souple pour élaborer des recommandations pertinentes et adaptées aux besoins particuliers des utilisateurs. La gestion de la complexité et de la diversité des données utilisateur est améliorée grâce aux **SMA**, qui permettent de subdiviser les problèmes en sous-tâches gérées par divers agents. La modularité permet de s'adapter aux changements de comportement des utilisateurs ou aux nouvelles tendances des systèmes de recommandation.

Les **SMA** ont un intérêt pour la recommandation. Le principal intérêt réside dans la capacité des **SMA** à traiter efficacement les informations et à prendre des décisions. Les **SMA** ont la capacité d'améliorer la précision des recommandations en simulant des interactions sociales et en utilisant des stratégies de négociation. En outre, ils favorisent une expérience utilisateur améliorée en offrant des suggestions plus cohérentes et adaptées au contexte.

Dans qui suit nous détaillons quatre travaux qui ont utilisé les **SMA** pour la recommandation.

2.4.1 Architecture des SMA pour les Recommandations POI

L'architecture présentée dans le premier travail [67] repose sur un système multi-agents (**SMA**) pour les recommandations touristiques. Le processus commence par la transmission d'informations par le serveur aux agents concernant les Points d'intérêt

(**POI**) dans une ville. Ces **POIs** sont extraits d'une analyse automatique d'un ensemble de données réelles confirmé par Google.

Après avoir nettoyé les données des trajectoires en supprimant les points **GPS** incohérents et en regroupant les données en 6 intervalles de 4 heures. Ensuite, une étape de détection des StayPoints (**SPs**) en identifiant les régions où les utilisateurs ont passé du temps.

Enfin, l'utilisation de **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) pour regrouper les SPs en clusters, dans le but de permettre d'identifier les lieux fréquentés par plusieurs utilisateurs.

Chaque **POI** est précédemment déterminé par l'algorithme **DBSCAN** qui identifie des régions denses en fonction d'Epsilon (**Eps**) et de MinPts. Il crée des clusters de lieux significatifs (**POIs**) pour les utilisateurs, en sélectionnant les **POIs** les plus populaires. Les expériences montrent une similarité entre les utilisateurs en se déplaçant ensemble vers les mêmes **POIs** à différents moments. L'exécution de **DBSCAN** est rapide, prenant de 240 ms à 1,44 s, et chaque agent reçoit des détails tels que les créneaux horaires les plus visités, le nombre d'agents présents en temps réel, les commentaires des utilisateurs analysés par un algorithme de sentiment, et les évaluations basées sur les informations précédentes.

Les étapes suivantes incluent le choix d'un lieu d'intérêt par l'agent en fonction de la note, la transmission des coordonnées **GPS** au serveur central lorsque l'utilisateur est à moins d'un kilomètre du **POI**, la collecte des commentaires de l'utilisateur après la visite, et la réception d'une liste mise à jour de **POI** basée sur les expériences des utilisateurs précédents.

2.4.2 SRMA pour l'Extraction des Intérêts Utilisateurs

En revanche, le deuxième travail [68] explore les subtilités d'un système de recommandation multi-agents complet, spécifiquement conçu pour l'extraction des intérêts des utilisateurs. Malheureusement, ce système repose sur une infrastructure très complexe et vaste, composée de huit agents spécialisés, chacun responsable de différents aspects tels que l'âge, l'identité, les aspects sociaux et financiers.

Le système se compose de huit agents, chacun étant chargé de collecter des informations sur l'utilisateur en fonction de ses intérêts. Ces agents couvrent divers aspects tels que

l'âge, le statut social, l'identité, la personnalité, les besoins, l'emplacement et la situation financière de l'utilisateur. Les données collectées sont analysées par un agent central (ICA), qui propose ensuite des recommandations en fonction des intérêts de l'utilisateur. Ces recommandations peuvent inclure des vidéos, des achats ou des sites scientifiques, correspondant étroitement aux préférences de l'utilisateur.

La communication entre les agents est gérée par un gestionnaire de communication, qui dirige les commandes vers la machine administrative. Cette dernière interagit avec un ensemble de règles et un moteur de délibération pour déterminer les actions à entreprendre. Les règles, basées sur des expériences d'experts, guident le comportement du système. Un moteur d'inférence traite les événements et détermine les actions à prendre, tandis que l'agent central coordonne les interactions entre les agents pour garantir des recommandations optimales.

Enfin, le système utilise des vidéos pour évaluer les intérêts de l'utilisateur, et chaque agent est chargé de certaines caractéristiques ayant un impact sur le comportement humain.

2.4.3 Système de Recommandation Multi-Agents Intelligent

Par ailleurs, une contribution notable émane du troisième travail [69]. Les auteurs proposent un système de recommandation multi-agent comprenant un agent de correspondance et un agent de recommandation.

Les utilisateurs interagissent avec le système via une interface utilisateur pour accéder aux informations.

L'agent de correspondance utilise la méreologie approximative pour identifier des utilisateurs similaires, tandis que l'agent de recommandation suggère des valeurs basées sur les préférences des utilisateurs. L'agent de correspondance comprend trois modules : le module utilisateur, le module d'informations sur les données et le module de méreologie approximative, qui travaillent ensemble pour générer une liste de données correspondantes. L'agent de recommandation calcule les valeurs de décision pour les utilisateurs en fonction des valeurs de similarité fournies par l'agent de correspondance.

L'intégration avec l'interface utilisateur assure l'accessibilité et la facilité d'utilisation, permettant aux utilisateurs de modifier les profils et de visualiser les valeurs de décision recommandées.

L'évaluation se concentre sur des méthodes statistiques, en particulier l'erreur absolue moyenne (**MAE**), pour évaluer la qualité des recommandations, mettant en évidence le potentiel du système tout en reconnaissant les limites de l'utilisation de la météorologie approximative pour le classement. Un affinement supplémentaire est nécessaire pour optimiser les performances du système selon ce qui est mentionné.

2.4.4 Architecture Multi-Agents pour l'e-Tourisme

Le dernier travail [70] dévoile une architecture multi-agents sur mesure conçue spécifiquement pour les applications e-Tourisme. Cette architecture repose sur une structure complexe comprenant de différents rôles d'agents, tels que l'Utilisateur, le Noyau de Système de Recommandation Généraliste (**GRSK**), le Planificateur et le Trouveur, qui collaborent pour offrir des expériences touristiques personnalisées.

L'application e-Tourisme vise à fournir des recommandations personnalisées pour les activités de loisirs à Valence, en Espagne, en analysant les profils et les préférences des utilisateurs. Contrairement aux sites touristiques traditionnels, l'équipe e-Tourisme essaye d'adapter les recommandations en fonction des caractéristiques individuelles de chaque utilisateur, créant des agendas personnalisés pour chacun.

L'architecture du système comprend sept agents responsables de différents aspects de la collecte d'informations sur les utilisateurs, de la génération de recommandations et de la planification des agendas. Les utilisateurs peuvent s'inscrire, mettre à jour leurs profils, demander des plans de visite, recevoir des recommandations d'activités et planifier leurs agendas grâce à des interactions avec le système.

Le Noyau du Système de Recommandation Généraliste (**GRSK**) est au cœur du processus de recommandation, utilisant plusieurs techniques de recommandation telles que : Filtrage basé sur les préférences actuelles de l'utilisateur, Filtrage basé sur les préférences générales de l'utilisateur, données démographiques et enfin la recommandation basée sur les visites précédentes pour générer des listes d'activités personnalisées. L'agent de planification optimise ensuite l'agenda en fonction des préférences de l'utilisateur, du

temps disponible et des détails des activités. En utilisant une architecture de Système Multi-Agent (**SMA**), e-Tourisme parvient à la flexibilité, l'adaptabilité et l'évolutivité, offrant des recommandations et des agendas touristiques sur mesure pour les utilisateurs. Le tableau ci-dessus représente une synthèse sur les travaux cités auparavant :

Travaux	Objectif	Méthodes de recommandation	Les agents	Limites (faiblesses)
[67]	Proposer des recommandations touristiques personnalisées grâce à un SMA	L'algorithme DBSCAN est utilisé pour repérer les POIs populaires à partir de données GPS .	Les agents reçoivent des renseignements concernant les POIs , tels que les horaires en vogue, le nombre d'agents présents et les évaluations des utilisateurs.	L'utilisation d'un seul système de recommandation collaborative peut restreindre la pertinence pour les utilisateurs ayant des préférences différentes ou provenant de différentes cultures/régions/religions.
[68]	Analyser et extraire les intérêts des utilisateurs afin de proposer des recommandations personnalisées.	Recherche de données par huit agents spécialisés qui abordent différents aspects de l'utilisateur. Analyse approfondie de l'ICA afin de formuler des recommandations.	L'âge, le statut social, l'identité, la personnalité, les besoins, l'emplacement et la situation financière de l'utilisateur sont surveillés par huit agents spécialisés.	Infrastructure complexe et vaste. Il est essentiel de coordonner plusieurs agents ayant des responsabilités spécifiques.

[69]	Faciliter l'accès aux informations et offrir des recommandations personnalisées	Une méthode approximative de métrologie permet de repérer des utilisateurs similaires et de calculer des valeurs de décision.	Agent de correspondance comprenant trois modules et agent de recommandation qui propose des valeurs en fonction des préférences.	Seulement des valeurs numériques sont prises en compte, ce qui restreint la possibilité de capturer la diversité des préférences des utilisateurs. Les agents statiques restreignent la capacité d'adaptation et de réactivité du système.
[70]	Proposer des activités touristiques personnalisées à Valence, en Espagne.	Le filtrage repose sur les préférences actuelles et générales de l'utilisateur, les données démographiques et les recommandations basées sur les visites antérieures.	Sept agents chargés de recueillir des informations, de formuler des recommandations et de planifier les agendas.	- La gestion des agents est complexe ‘La coordination et la gestion peuvent devenir complexes lorsque plusieurs agents spécialisés sont impliqués, ce qui demande des mécanismes avancés pour garantir une communication et une collaboration efficaces entre les agents.’

Tableau -3- Tableau de synthèse

2.5 Conclusion

En conclusion, ce chapitre a présenté une vue d'ensemble sur les Systèmes Multi-Agents (**SMA**) et de leur application dans l'amélioration des systèmes de recommandation. Nous avons exploré les caractéristiques clés des **SMA**, notamment leur structure, leur communication et leur capacité à collaborer pour résoudre des problèmes. Nous avons également souligné comment les **SMA** facilitent une prise de décision distribuée et autonome, offrant ainsi des avantages significatifs dans le domaine des systèmes de recommandation.

Le prochain chapitre sera consacré à la partie conception et réalisation.

Chapitre 3

Conception et modélisation

3.1 Introduction

Au début de ce chapitre, nous exposons une architecture qui repose sur un système multi-agents pour WonderWise, dans le but de fournir des recommandations précises et efficaces. Elle incorpore des algorithmes de notre propre innovation ainsi que des méthodes connues dans le contexte des recommandations dans une API afin de fournir des décisions efficaces. Dans la deuxième partie, nous abordons la description de WonderWise en utilisant UML, en utilisant des diagrammes pour rendre sa conception plus visible et formalisée. On réalise des modélisations des cas d'utilisation, des séquences, des classes et diagramme de composant afin de pouvoir expliquer la structure et les opérations du système. Nous utilisons ces outils de modélisation pour expliquer comment notre recommandation de points d'intérêt fonctionne.

3.2 Système multi agent de recommandation des POI

Dans cette partie, nous présentons notre système de recommandation basé un système multi agents qui consistera le cœur de notre application « WonderWise ». Notre système repose sur un **SMA** qui englobe des agents qui implémente des méthodes de recommandation, dont qui sont plus utiliser dans ce contexte et d'autre sont nos propres propositions. Tout ce mécanisme sous format une **API**. L'objectif est de fournir des recommandations intelligentes et orientées vers l'intelligence humaine en orchestrant l'interaction entre sept agents distincts. Ces agents s'appuient sur les WebSockets pour permettre un échange bidirectionnel et en temps réel des données entre eux, assurant ainsi une interaction fluide et instantanée. Cette approche répond de manière optimale aux besoins des utilisateurs en

offrant une expérience utilisateur réactive et en temps réel. Aussi l'exécution de chaque agent dans son propre runtime garantit que chaque agent fonctionne dans un environnement spécialisé, avec son propre espace d'adressage dédié dans la mémoire **RAM**. Cette approche assure un déroulement non bloquant et asynchrone du système, améliorant ainsi la performance globale et la réactivité. L'identification de chaque agent par un port unique sur l'hôte renforce encore la séparation et l'individualité de chaque agent, facilitant ainsi la gestion et la communication entre eux. Cette approche permet une coordination efficace des tâches et des flux de données, renforçant la robustesse et la scalabilité du système dans son ensemble. Voici la **Figure 14** qui illustre notre Architecture des agents de notre système.

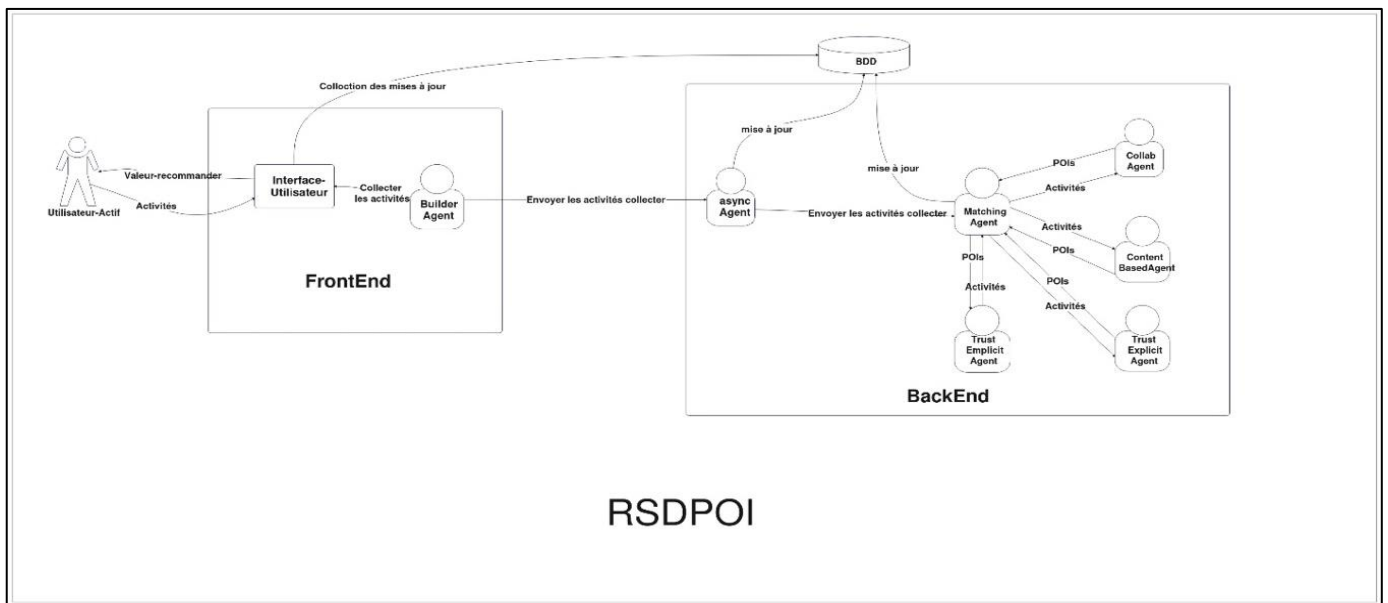


Figure -14- Architecture des agents de notre système.

Notre système est constitué de sept agents, qui sont :

1. **BuilderAgent :**

Au centre de l'expérience utilisateur, cet agent de collecte opère de manière proactive, capturant en temps réel une multitude d'activités de l'utilisateur. Il enregistre minutieusement les consultations des points d'intérêt, les évaluations, les

commentaires, ainsi que les interactions sociales telles que les likes et les favoris. De plus, il trace le temps passé sur chaque point d'intérêt consulté. Cette collecte exhaustive garantit une saisie complète de l'engagement de l'utilisateur avec la plateforme. Une fois les données recueillies, cet agent s'assure de leur transmission efficace vers l'agent **asynchroneAgent**, garantissant ainsi une intégration fluide et en temps réel des informations essentielles dans l'ensemble du système.

2. **MatchingAgent** :

Ce maître d'orchestre des recommandations joue un rôle central dans l'agrégation et l'évaluation des données. Il distribue les informations pertinentes aux agents **collab-agent**, **content-based-Agent**, **implicite-trust-Agent** et enfin **explicit-trust-Agent**, les résultats de leurs analyses pour produire des recommandations équilibrées et pertinentes. L'un de ces rôles aussi est de faire un vote dans chaque cycle de vie d'une recommandation. L'agent fusion et équilibrage des recommandations opère au sein du système de recommandation pour harmoniser et intégrer les recommandations provenant de différentes sources, garantissant ainsi une sortie équilibrée et cohérente pour l'utilisateur.

❖ **Responsabilité** :

- Partage et réception des données entre autres agents :

Une fois que l'agent reçoit des données à partir de l'agent **collector** concernant certain utilisateur sous format des activités, ce dernier a le job de garantir le partage des données en temps réel et de manière asynchrone avec les autres agents pour qu'il puisse commencer le processus de recommandation et aussi reste toujours à l'écoute de réception des résultats initiaux de recommandation par chaque agent à partir des quatre agents qui sont respectivement encore : **content-based-agent** , **collab-agent** , **implicite / explicit-trust-agents** , dès que les données sans prêt l'agent démarre directement le processus de vote .

- Processus de Vote (étapes) :

1) **Filtrage des Recommandations** : L'agent est chargé de filtrer les recommandations des agents : collaborative, confiance (explicite et implicite) et

contenu en fonction de leurs scores respectifs. Il trie les recommandations par ordre décroissant de score, garantissant que les plus pertinentes sont prioritaires.

2) Détermination de la source dominante : il identifie la source dominante des recommandations en déterminant celle avec la liste la plus longue de recommandations filtrées. Cela garantit que les recommandations de la source la plus influente sont prises en compte de manière prépondérante dans la sortie finale.

3) Processus de vote : l'agent orchestre un processus de vote pour calculer les scores finaux de chaque point d'intérêt (**POI**). Il itère à travers les recommandations filtrées, en commençant par la liste avec le plus d'entrées, et attribue des scores finaux à chaque **POI** en fonction de sa position relative dans les listes des autres sources.

- Mise à jour du profil utilisateur : une fois que les recommandations finales et les scores sont déterminés, l'agent met à jour le profil de l'utilisateur dans la base de données avec les nouvelles recommandations exactement dans le champ recommandations. Cela garantit que l'utilisateur reçoit des suggestions personnalisées et pertinentes pour ses interactions futures.

3. Content-based-Agent :

Cet agent joue un rôle crucial dans le système en fournissant des recommandations personnalisées de points d'intérêt (**POIs**) à l'utilisateur.

Voici comment il opère :

- **Connexion à la Base de Données** : L'agent se connecte à la base de données MongoDB au niveau de la collection userProfile en utilisant la clé userId pour récupérer le profil de l'utilisateur courant. Cette étape garantit l'accès aux informations pertinentes de l'utilisateur.
- **Mise à jour du profil utilisateur (fusion)** : Une fois le profil récupéré, l'agent fusionne les données de l'ancien profil avec celles nouvellement récupérées. Cela assure une mise à jour complète et précise du profil de l'utilisateur, en prenant en compte les dernières informations disponibles.

- **Analyse des activités utilisateur** : À partir des activités utilisateur telles que les clics, les likes, les commentaires sur les **POIs** et le temps passé sur chaque **POI**, l'agent calcule la fréquence de chaque catégorie de **POIs**. Cette analyse permet de comprendre les préférences et les intérêts de l'utilisateur.
- **Identification des catégories pertinentes** : Sur la base des fréquences calculées, l'agent identifie les catégories les plus pertinentes pour l'utilisateur. Ces catégories représentent les centres d'intérêt de l'utilisateur et guident les recommandations futures.
- **Calcul des scores de Pois** : Pour chaque catégorie identifiée, l'agent calcule les scores des **POIs** correspondants en fonction de leur pertinence pour l'utilisateur. Ces scores déterminent les **POIs** recommandés pour chaque catégorie, en mettant en avant les lieux les plus susceptibles d'intéresser l'utilisateur.

4. ColabAgent :

Cet agent, opère en récupérant le profil de l'utilisateur et en effectuant une analyse collaborative des évaluations des points d'intérêt (**POIs**). Voici un aperçu de son fonctionnement :

- **Construction de la matrice Utilisateur-POI** : L'agent construit une matrice utilisateur-poi où chaque élément représente l'évaluation (le rating) d'un utilisateur pour un **POI** donné. Cette matrice permet de visualiser les évaluations des utilisateurs pour chaque **POI**.
- **Identification des POIs non évalués** : En utilisant la matrice utilisateur-poi, l'agent identifie les **POIs** non évalués par l'utilisateur cible. Ces **POIs** représentent les éléments vides dans la matrice, ce qui indique que l'utilisateur n'a pas encore évalué ces lieux.
- **Calcul des évaluations prévues (prédiction)** : Pour chaque **POI** non évalué par l'utilisateur cible, l'agent calcule les prédictions des évaluations prévues en se basant sur les évaluations des utilisateurs similaires. Il utilise des techniques de filtrage collaboratif et calcul des voisinages via cosinus similarité (CosineSimilarity) pour estimer les évaluations manquantes. La fonction de similarité cosinus est un outil mathématique qui mesure à quel point deux ensembles de données sont similaires. Elle

compare la direction dans laquelle pointent les données, indépendamment de leur taille. Dans les recommandations ou les systèmes de filtrage, elle aide à trouver des éléments ou des utilisateurs similaires en analysant leurs préférences ou caractéristiques. Les évaluations prévues sont ensuite normalisées entre 0 et 1 pour faciliter la comparaison et garantir une cohérence dans les scores.

5. **Implicit-trust-Agent :**

Cet agent est chargé de recommander des Points d'intérêt (**POIs**) à un utilisateur en se basant sur la confiance entre cet utilisateur et d'autres utilisateurs. Voici comment il fonctionne en détail :

- **Calcul de la confiance :** L'agent commence par calculer la confiance entre l'utilisateur et chacun d'autres utilisateurs. Il compare les préférences de l'utilisateur avec celles de ses amis, en utilisant des critères tels que les catégories préférées et les **POIs** visités. La confiance est calculée en combinant la similarité entre les préférences de l'utilisateur et celles de ses amis, avec des poids ajustables pour chaque critère de similarité. Les amis sont ensuite triés par ordre de confiance décroissante.
- **Identification des POIs communs :** Pour chaque ami, l'agent identifie les **POIs** communs entre l'utilisateur et cet ami. Ces **POIs** sont ceux que l'ami a visités ou évalués et que l'utilisateur n'a pas encore explorés.
- **Calcul de la similarité d'activité :** Pour chaque **POI** commun, l'agent calcule la similarité entre l'activité de l'utilisateur et celle de l'ami pour ce **POI**. Cela permet de déterminer dans quelle mesure les activités de l'utilisateur et de l'ami se chevauchent pour ce **POI**.
- **Calcul de la similarité des évaluations :** De même, l'agent calcule la similarité entre les évaluations de l'utilisateur et celles de l'ami pour chaque **POI** commun. Cela permet de comparer les évaluations données par l'utilisateur et celles données par l'ami pour les mêmes **POIs**.
- **Combinaison des similarités :** Les similarités d'activité et d'évaluation sont combinées avec les scores de confiance pour chaque ami. Les scores combinés sont calculés en attribuant des poids aux similarités et à la confiance, puis en les moyennant. Cette étape permet de donner plus de poids aux amis en qui l'utilisateur a le plus confiance, tout en tenant compte de la similarité des activités et des évaluations.

- **Filtrage des amis :** Les amis sont filtrés en fonction de leurs scores combinés. Seuls les amis avec des scores supérieurs à un seuil prédéfini sont conservés.
- **Génération de recommandations de POIs :** Enfin, l'agent génère des recommandations de **POIs** en se basant sur les **POIs** visités ou évalués par les amis filtrés. Il sélectionne les **POIs** que les amis ont appréciés et que l'utilisateur n'a pas encore découverts. Ce système sophistiqué offre une expérience de recommandation personnalisée et riche en nuances, tirant parti des interactions utilisateur et des dynamiques sociales pour fournir des suggestions de points d'intérêt pertinents et enrichissants.

6. **Explicit-Trust-Agent :**

Cet agent est chargé de recommander des Points d'intérêt (Pois) à un utilisateur en se basant sur la confiance explicite qui signifie que la recommandation va être basée sur la confiance donnée par l'utilisateur lui-même ou ses amis, cet agent base sur le paradigme ou bien la technique de filtrage collaborative à base mémoire sur le rating des amis pour la prédiction des pois. Voici comment il fonctionne en détail :

- **Récupération des amis avec confiance élevée :** L'agent focalise ses calculs de prédiction des pois non seulement sur les amis avec une confiance explicite exprimée déjà par l'utilisateur courant, mais aussi sur les amis avec une confiance très élevée dans notre système la confiance peut varier de 1 à 3, donc les amis seuls avec une confiance de 3 sont pris en considération.
- **Construction de la matrice Utilisateur-POI :** L'agent construit une matrice utilisateur-poi où chaque élément représente l'évaluation (le rating) d'un utilisateur pour un **POI** donné. Cette matrice permet de visualiser les évaluations des utilisateurs pour chaque **POI**.
- **Identification des POIs non évalués :** En utilisant la matrice utilisateur-poi, l'agent identifie les **POIs** non évalués par l'utilisateur cible. Ces **POIs** représentent les éléments vides dans la matrice, ce qui indique que l'utilisateur n'a pas encore évalué ces lieux.
- **Calcul des évaluations prévues (prédiction) :** Pour chaque **POI** non évalué par l'utilisateur cible, l'agent calcule les prédictions des évaluations prévues en se basant sur les évaluations des utilisateurs amis avec confiance. Il utilise des techniques de

filtrage collaboratif a base mémoire et calcul via cosinus similarité (cosine similarity) pour estimer les évaluations manquantes pour en moins différencier entre les amis même s'il possède la même confiance. Les évaluations prévues sont ensuite normalisées entre 0 et 1 pour faciliter la comparaison et garantir une cohérence dans les scores.

7. Agent Asynchrone:

Le rôle de l'Agent Asynchrone est essentiel dans l'organisation d'un système de recommandation. Il coordonne le flux de données entre le **BuilderAgent**, qui collecte activement les activités des utilisateurs, et la base de données, ainsi que le **MatchingAgent**, qui correspond aux activités des utilisateurs pour les recommandations.

Il assure principalement la continuité et l'efficacité de la communication des données, assurant ainsi des mises à jour instantanées et une coordination parfaite entre les agents, afin de fournir des recommandations toujours pertinentes et à jour.

3.2.1 Les Systèmes de Recommandation en Temps Réel : Flux de Données et Réactivité

Pour garantir le succès des recommandations de pois et assurer un flux de données en temps réel aux utilisateurs, plusieurs éléments sont essentiels. Premièrement, notre **API** se concentre sur la gestion des données dans deux collections principales de notre base de données. Elle utilise des fonctions **listener** agissant comme des **streams** pour détecter efficacement tout changement dans les documents de ces collections.

Cette fonctionnalité permet aux différents agents de recommandation, tels que les agents de recommandation collaborative, basée sur le contenu, ainsi que les agents de confiance implicite et explicite, d'être notifiés instantanément des modifications.

Cette notification est cruciale car elle leur permet d'ajuster leurs processus de recommandation en temps réel, assurant ainsi une pertinence optimale des recommandations fournies aux utilisateurs. Cette réactivité assure la pertinence et la précision des recommandations fournies, ce qui contribue à une expérience utilisateur améliorée.

3.3 Méthode d'Analyse et conception

3.3.1 UML

UML est un langage de modélisation unifié graphique et textuel. Son but est de saisir, décrire, préciser et établir des données sur les besoins, les systèmes, les architectures logicielles, ainsi que de développer des solutions et de communiquer les divers aspects d'un système info numérique. Autrement dit, « **UML** » est le langage de modélisation classique employé par les acteurs majeurs du marché afin de représenter et de structurer les concepts et les relations dans le domaine de la technologie objet [71].

3.3.2 Spécification des besoins du système

Au cours de cette étape, nous allons repérer les caractéristiques de notre système en identifiant les acteurs et les besoins fonctionnels et non fonctionnels du system.

3.3.2.1 Identification des acteurs

Dans notre système nous avons identifié un seul acteur qui est :

- ❖ **L'utilisateur** : L'acteur est responsable de l'utilisation du système et a la possibilité d'accéder à tous les éléments de l'application.

3.3.2.1.1 Identification des besoins

- **Les besoins fonctionnels**

Le système doit permettre à l'utilisateur de :

- S'authentifier
- Créer un compte
- Afficher les informations sur un POI
- Rechercher des POI
- Ajouter des POI aux favoris
- Avoir des recommandations personnalisées

Le système doit permettre au visiteur de :

- S'inscrire
- Rechercher des POI
- Afficher les informations sur un POI
- **Les besoins non fonctionnels**

Les exigences concernant les caractéristiques du système sont telles. Il concerne les critères de performance, le choix du matériel ou la conception. Toutes les modifications à réaliser doivent respecter les critères suivants :

- a) **Fiabilité** : Il est essentiel que l'application fonctionne de manière cohérente sans erreurs et qu'elle soit satisfaisante.
- b) **Les erreurs** : Des messages d'erreurs bien structurés doivent être utilisés pour signaler les ambiguïtés afin de guider l'utilisateur et de le familiariser avec notre application.
- c) **Ergonomie et bonne Interface** : Il est nécessaire de personnaliser l'application pour répondre aux besoins de l'utilisateur sans qu'il ne fasse aucun effort (utilisation claire et simple).
- d) **Sécurité** : Il est primordial que notre solution garantisse principalement la protection des données personnelles des clients, qui demeure l'une des contraintes les plus cruciales dans les applications mobiles.

3.4 Les Diagrammes UML

3.4.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation illustre le fonctionnement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur externe le perçoit [72]. La **Figure 16** présente le diagramme de cas d'utilisation de notre système.

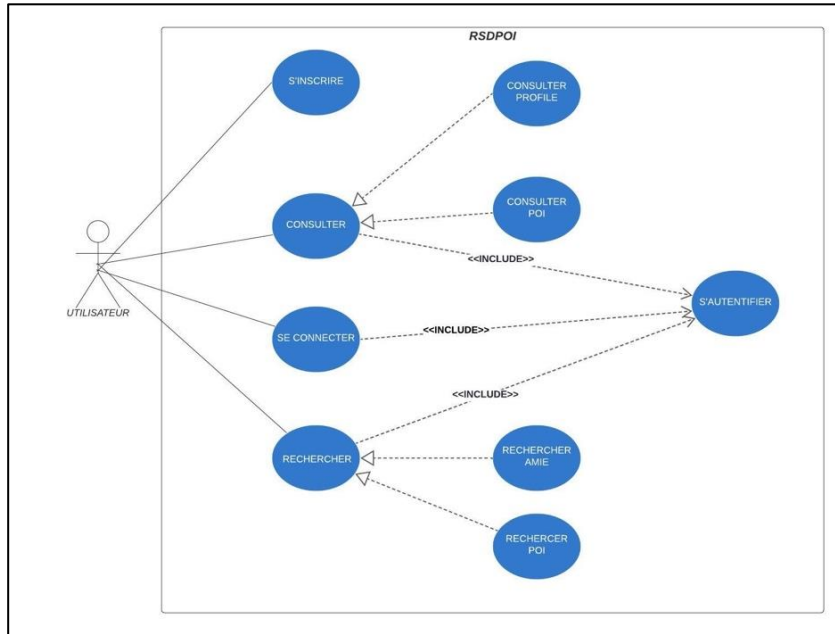


Figure -15- Diagramme de cas d'utilisation de notre système.

3.4.2 Description textuelle

3.4.2.1 Description textuelle de cas d'utilisation « Inscription »

Scénario1	Inscription
Objectif	Permettre à l'utilisateur de créer un compte dans le système.
Acteurs principaux	Utilisateur
Pré conditions	L'utilisateur doit avoir accès à l'interface de création de compte.
Postconditions	Le compte de l'utilisateur est créé et l'utilisateur est redirigé vers la page de connexion.
	<ol style="list-style-type: none"> 1. L'utilisateur initie l'accès à l'Interface de création de compte. 2. L'Interface de création de compte demande au Système de présenter l'interface. 3. Le Système affiche l'Interface de création de compte à l'Utilisateur. 4. L'Utilisateur remplit les champs requis sur l'Interface de création de

<p>Scénario nominal</p>	<p>compte.</p> <ol style="list-style-type: none"> 5. L'Interface de création de compte envoie les informations au Système pour examen. 6. Si les champs sont corrects, le système demande à l'API de vérifier si l'utilisateur existe déjà via la Base de données MongoDB. 7. Si l'utilisateur n'existe pas, le Système procède à la création du compte utilisateur. 8. L'API crée le compte dans la Base de données MongoDB et confirme la création au Système. 9. Le système informe l'Interface de création de compte de la réussite de la création. 10. L'Interface de création de compte redirige l'utilisateur vers la page de connexion.
<p>Scénario alternatif</p>	<p>A1 : Champs obligatoires vides ou format incorrect</p> <ol style="list-style-type: none"> 1. Le Système affiche un message d'erreur sur l'Interface de création de compte. 2. Le système retourne au scénario nominal pour permettre à l'utilisateur de corriger les erreurs. <p>A2 : Format de l'email incorrect</p> <ol style="list-style-type: none"> 1. Le Système affiche un message d'erreur "Format de l'email incorrect" sur l'Interface de création de compte. 2. Le système retourne au scénario nominal pour permettre à l'utilisateur de saisir un email correct. <p>A3 : Mot de passe invalide</p> <ol style="list-style-type: none"> 1. Le Système affiche un message d'erreur "Mot de passe invalide" sur l'Interface de création de compte.

	<p>2. Le système retourne au scénario nominal pour permettre à l'utilisateur de créer un mot de passe conforme.</p> <p>A4 : Utilisateur déjà existant dans le système</p> <p>1. Le Système affiche un message d'erreur "Utilisateur existe déjà" sur l'Interface de création de compte.</p> <p>2. Le système retourne au scénario nominal pour permettre à l'utilisateur d'utiliser une autre adresse email.</p> <p>A5 : Moins de trois catégories sélectionnées</p> <p>1. Le Système affiche un message d'erreur "Vous devez sélectionner au moins trois catégories" sur l'Interface de création de compte.</p> <p>2. Le système retourne au scénario nominal pour permettre à l'utilisateur de sélectionner les catégories nécessaires.</p> <p>A6 : Termes d'utilisation non acceptés</p> <p>1. Le Système affiche un message d'erreur "Vous devez accepter les termes d'utilisation" sur l'Interface de création de compte.</p> <p>2. Le système retourne au scénario nominal pour permettre à l'utilisateur d'accepter les termes et conditions.</p>
--	---

Tableau- 4- Description textuelle de cas d'utilisation « Inscription ».

3.4.2.2 Description textuelle de cas d'utilisation «se connecter »

Scénario1	S'authentifier
Objectif	Permettre à l'utilisateur de s'authentifier et d'accéder à son compte dans l'application.
Acteurs principaux	Utilisateur
Préconditions	L'utilisateur doit accéder à l'interface de connexion.
Postconditions	L'utilisateur est authentifié et accède à l'accueil de l'application.

<p>Scénario Nominal</p>	<ol style="list-style-type: none"> 1. L'Utilisateur accède à l'application via l'Interface de connexion. 2. L'Interface de connexion demande au Système d'afficher l'interface de connexion. 3. Le Système affiche l'interface de connexion. 4. L'Utilisateur introduit son email et son mot de passe. 5. L'Interface de connexion transmet les informations de connexion au Système. 6. Le Système valide les informations et, si elles sont correctes, demande à l'API de vérifier l'utilisateur dans la Base de données MongoDB. 7. L'API confirme l'authentification de l'utilisateur. 8. Le Système affiche l'accueil de l'application à l'Utilisateur.
<p>Scénario Alternatif</p>	<p>A1 : Champs obligatoires vides ou non valides</p> <ol style="list-style-type: none"> 1. Le Système affiche un message d'erreur sur l'Interface de connexion. 2. Le système retourne au scénario nominal pour permettre à l'utilisateur de corriger les erreurs. <p>A2 : Format de l'email incorrect ou Mot de passe invalide</p> <ol style="list-style-type: none"> 1. Le Système affiche un message d'erreur "Format de l'email incorrect" sur l'Interface de connexion. 2. Le système retourne au scénario nominal pour permettre à l'utilisateur de corriger les erreurs.

Tableau-5-Description textuelle du cas d'utilisation « se connecter ».

3.4.2.3 Description textuelle de cas d'utilisation « Consulter POI »

Scénario1	Consulter POI
Objectif	Permettre à l'utilisateur de rechercher des Points d'Intérêt (POI) en fonction de ses

	critères de recherche dans le système de recommandation.
Acteurs Principaux	Utilisateur
Préconditions	- L'utilisateur est authentifié dans le système.
Postconditions	Le Point d'Intérêt est affiché avec tous les détails et options, et le profil de l'utilisateur est mis à jour avec les nouvelles activités sélectionnées.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'Utilisateur accède à l'InterfacePOI. 2. InterfacePOI demande à Système d'afficher le Point d'Intérêt. 3. Système affiche le Point d'Intérêt avec détails, localisation sur carte, et option pour ajouter aux favoris.
Scénario Optionnel	<p>Op1 : L'utilisateur commence à effectuer des activités. L'Utilisateur sélectionne une activité sur InterfacePOI.</p> <ol style="list-style-type: none"> 1. InterfacePOI informe le système que l'ajout d'activité a été effectué. 2. Système transmet la demande de mise à jour à InterfaceAPI. 3. InterfaceAPI met à jour le profil de l'utilisateur avec la nouvelle activité dans MongoDB. 4. MongoDB confirme la mise à jour à InterfaceAPI. 5. InterfaceAPI confirme la réception de la confirmation au système.

Tableau-6- Description textuelle de cas d'utilisation « consulter POI ».

3.4.3 Diagramme de séquence

Un diagramme de séquence est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps. Le temps s'écoule au fur et à mesure que vous parcourez la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence de messages [73].

3.4.3.1 Diagramme de séquence « inscription »

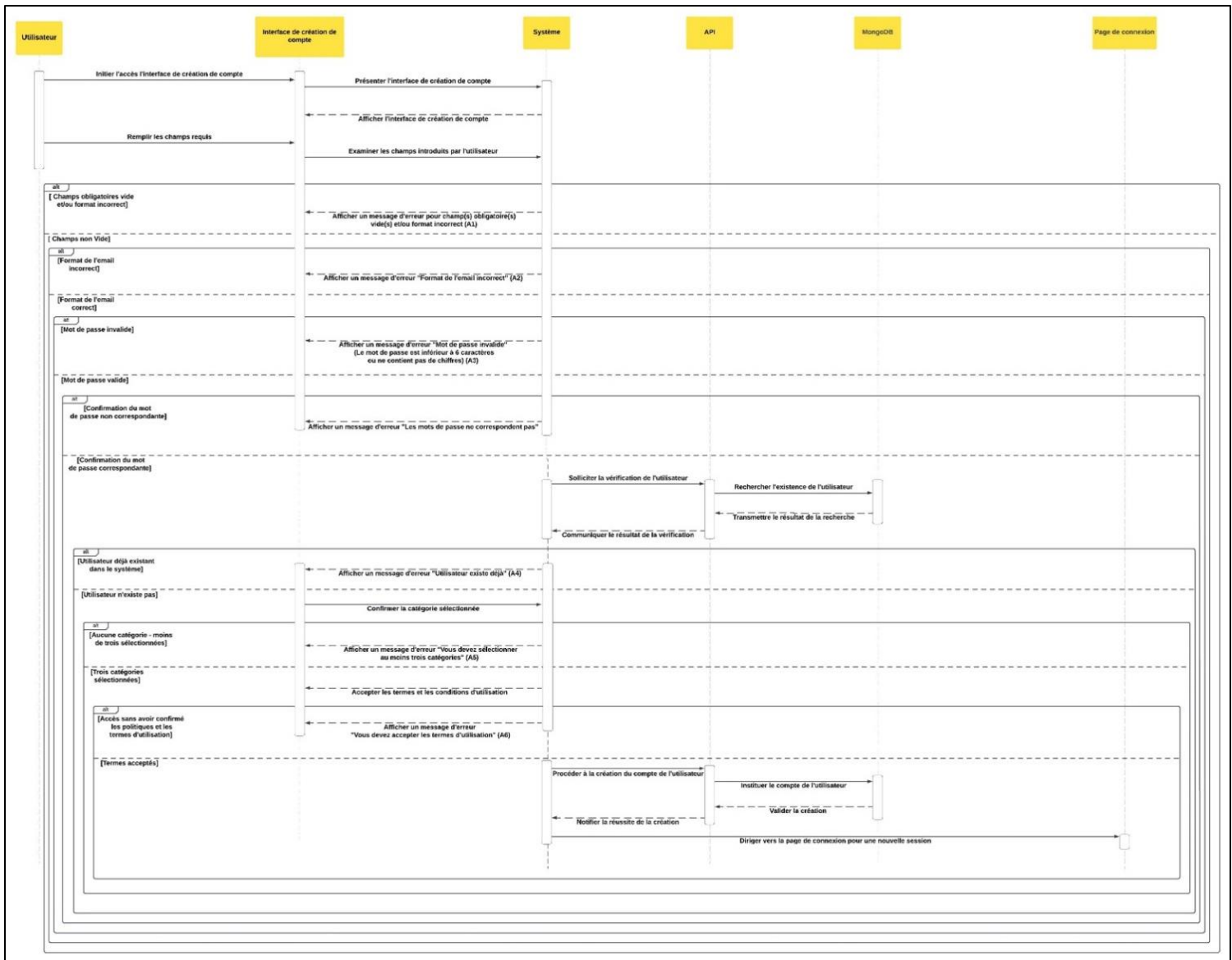


Figure -16- Diagramme de séquence « inscription »

3.4.3.2 Diagramme de séquence « Se connecter »

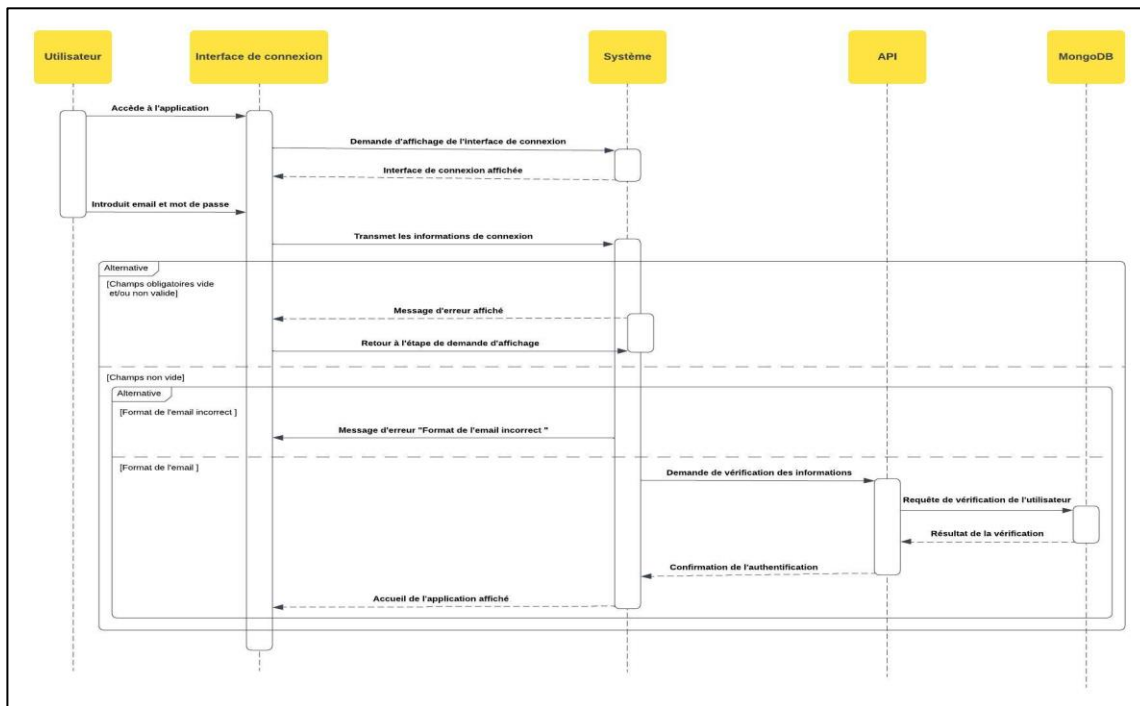


Figure-17 –Diagramme de séquence « Se connecter »

3.4.3.3 Diagramme de séquence « Consulter POI »

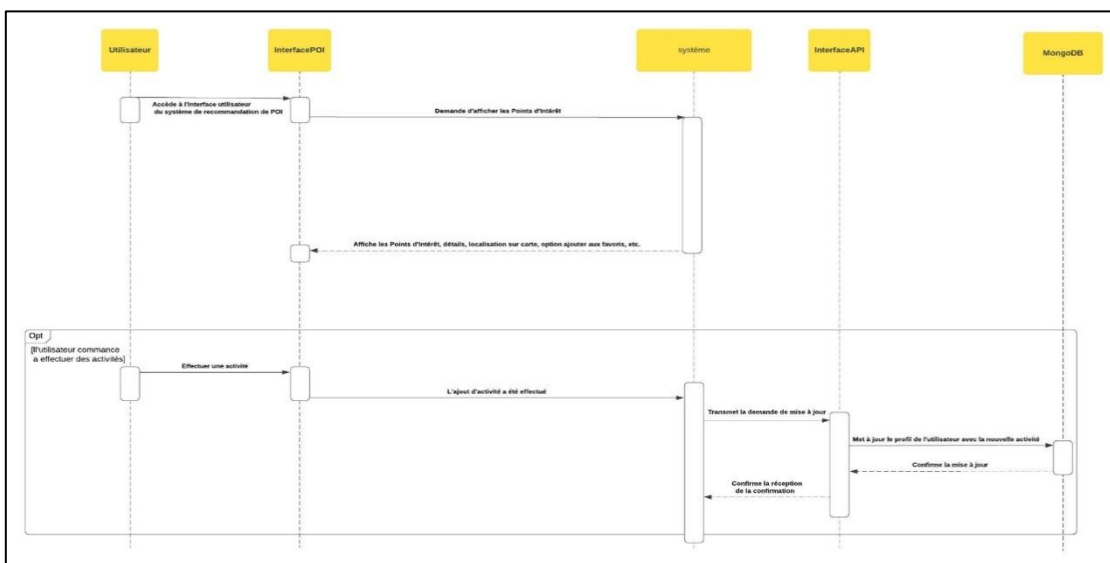


Figure -18- Diagramme de séquence « Consulter POI »

3.4.4 Diagramme de classe

Il est considéré que le diagramme de classes joue un rôle essentiel dans la modélisation orientée objet, et il est la seule étape nécessaire dans ce type de modélisation. Il incarne les catégories qui sont impliquées dans le système. Une représentation statique des éléments qui constituent le système et ses relations est un diagramme de classes. Chaque application qui mettra en place le système représente une représentation des diverses classes qui le constituent [74]. La **Figure 19** montre le diagramme de classe de notre système.

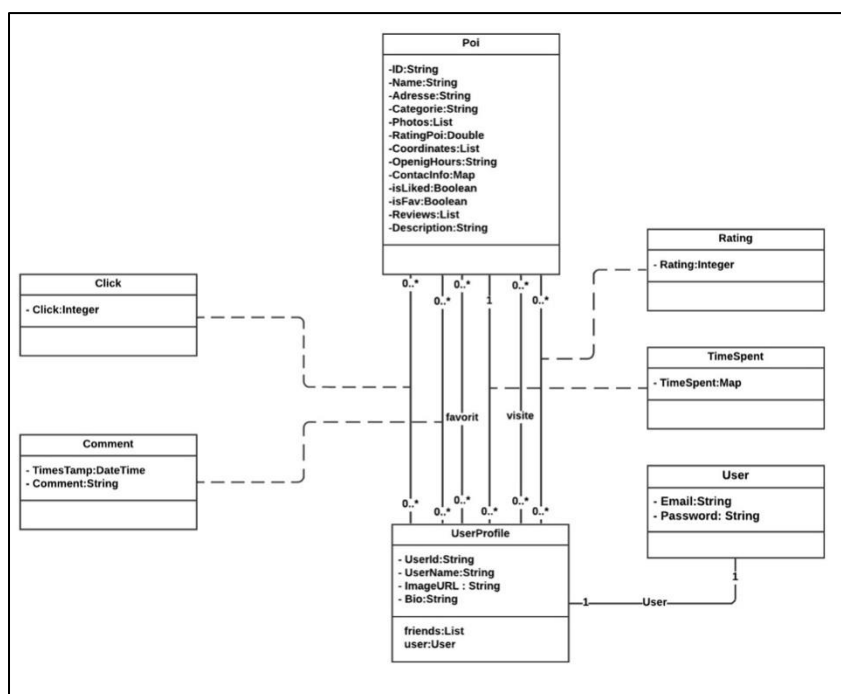


Figure -19- Diagramme de classe de notre Système

3.4.5 Diagramme de composant

Ils peuvent inclure des aspects de modélisation à la fois logiques et physiques. Dans le contexte de l'UML, les composants sont des éléments modulaires d'un système. Ils sont indépendants et peuvent être remplacés par des composants équivalents. Ces composants sont autonomes et encapsulent un certain nombre de structures complexes. Les éléments encapsulés entrent uniquement en contact avec d'autres composants via des interfaces.

Les composants peuvent mettre à disposition leurs propres interfaces ou utiliser les interfaces d'autres composants pour pouvoir accéder par exemple à leurs fonctions et leurs services. En parallèle, les interfaces documentent dans un diagramme de composants les relations et dépendances au sein d'une architecture logicielle. [75] La **Figure 20** présente le diagramme de composant de notre système.

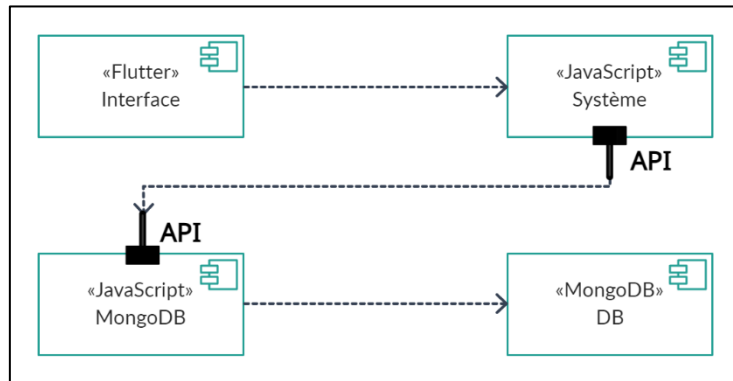


Figure -20- Diagramme de composant de notre Système.

3.5 Conclusion

Dans ce chapitre, nous avons exposé un système de recommandation multi-agents pour WonderWise, qui a été développé en se basant sur une analyse approfondie et les meilleures pratiques en architecture logicielle. Il intègre des recommandations précises et efficaces, des algorithmes spécifiques et la méthode de vote via une API, dans le but d'imiter l'intelligence humaine avec sept agents différents afin de proposer des recommandations précises sur les points d'intérêt.

Nous avons aussi exposé notre projet en utilisant UML, en optant pour des diagrammes de cas d'utilisation, de séquence, de classe et d'activité afin de refléter notre processus de conception. Grâce à ces outils de modélisation, nous avons pu observer, définir, élaborer et s'enregistrer la structure et les mécanismes de notre système de recommandation WonderWise. Le chapitre suivant représentera l'implémentation du système.

Chapitre 4

Implémentation

4.1 Introduction :

Dans ce dernier chapitre, nous allons entamer les outils et les environnements de programmation utilisés lors de l'implémentation de notre système sous une application mobile WanderWise, en suite, nous présenterons les différentes méthodes et algorithmes de recommandation pour qu'on puisse après discuter la partie évaluation et la présentation de notre application via les interfaces.

4.2 Environnement et développement

4.2.1 LucidChart

LucidChart est un logiciel de création de diagrammes en ligne qui offre aux utilisateurs la possibilité de travailler ensemble et de partager visuellement des diagrammes et des graphiques. Il s'agit d'une plateforme en ligne qui permet la collaboration en temps réel, ce qui implique que plusieurs individus peuvent collaborer sur le même schéma en même temps, avec des modifications visibles instantanément pour tous les participants. On utilise fréquemment LucidChart pour concevoir des organigrammes, des cartes mentales, des diagrammes UML, des wireframes pour des sites web, ainsi que d'autres formes de diagrammes visuels. Son interface utilisateur intuitive et sa compatibilité avec différents appareils et systèmes d'exploitation lui valent d'être prisé.[76]

4.2.2 Creately

Creately est une solution de gestion du travail basée sur un modèle visuel intelligent. Des équipes l'emploient pour le remue-méninge, la planification, la gestion de projets et la

collecte de connaissances, tout cela dans un même espace. Creately facilite la collaboration et l'organisation en connectant et en centralisant les éléments de travail provenant de diverses applications, documents et personnes en un seul endroit. C'est une édition de diagrammes et de conception qui est à la fois disponible en ligne sur le cloud et téléchargeable hors ligne pour les ordinateurs de bureau, compatible avec Windows, Mac et Linux. En outre, Creately propose des caractéristiques comme des tableaux blancs infinis, une puissante conception de diagrammes, une connexion de données et une intégration avec d'autres outils, ce qui en fait un outil polyvalent pour la collaboration visuelle. [77]

4.2.3 Flutter

Flutter est un Framework qui a été développé par Google pour le développement d'applications multiplateforme. Il propose de nombreuses bibliothèques d'éléments d'interface utilisateur standard pour Android et iOS. Le langage de programmation Dart, également développé par Google, est utilisé comme base du Flutter SDK. Il se présente comme le remplaçant contemporain du langage JavaScript traditionnel et, comme celui-ci, il fonctionne directement sur les navigateurs, sous forme d'application web.

Il est également possible d'exécuter les programmes Dart directement sur un serveur. [78]. Le but est de donner aux développeurs la possibilité de proposer des applications performantes qui semblent intuitives sur diverses plateformes. Le code existant de Flutter est utilisé par les développeurs et les organisations à travers le monde, et il est gratuit et open source.[79]

4.2.4 Visual Studio Code

Visual Studio Code, également connu sous le nom de VS, est l'un des éditeurs de code source les plus en vogue parmi les programmeurs. Il est aussi rapide, léger et puissant ! La fonctionnalité d'extension, extrêmement personnalisable, est sans aucun doute la caractéristique la plus pratique.

VS a été développé par Microsoft en tant qu'éditeur de code multiplateforme destiné à l'écriture d'applications Web et cloud.[80]

4.2.5 MongoDB

MongoDB, une plateforme de données destinée aux programmeurs. C'est une base de données NoSQL axée sur les documents, ce qui implique que les données sont stockées dans des documents JSON avec un schéma dynamique, plutôt que dans des tables et des lignes comme dans les bases de données relationnelles classiques. Grâce à sa structure flexible, cela facilite et accélère l'intégration des données, favorise l'évolutivité horizontale et permet de construire des applications plus rapidement.[81]

4.2.6 Canva

Entre les outils professionnels du dessin graphique nous allons trouver Canva comme un des plus utilisés. Canva offre aux utilisateurs la possibilité de concevoir différents types de contenus visuels tels que des publications sur les réseaux sociaux, des présentations, des affiches, des vidéos et des logos. Elle a été développée pour être conviviale, proposant une interface glisser-déposer intuitive et une large sélection de modèles, d'images et de polices. Canva convient à la fois aux experts en design et aux personnes sans expérience en design, ce qui rend la création de contenu visuel accessible à n'importe qui.[82]

4.2.7 JSON

JSON, abréviation de JavaScript Object Notation, est un format de fichier ouvert standard et un format d'échange de données qui utilise du texte lisible par l'homme pour stocker et transmettre des objets de données constitués de paires attribut-valeur et de tableaux (ou d'autres valeurs sérialisables). Le format JSON est basé sur la syntaxe de notation des objets JavaScript, mais il est exclusivement textuel. On utilise fréquemment JSON pour échanger des données électroniques, en particulier dans les applications web avec des serveurs [83]. Les règles de syntaxe JSON suivantes sont :

- a. Les informations sont regroupées en paires nom/valeur.
- b. Les informations sont entourées de virgules.
- c. Des objets sont stockés dans les accolades.
- d. Les crochets renferment des peintures.

4.3 Langages de programmation

4.3.1 Python

Python est un langage de programmation de haut niveau, facile à comprendre et polyvalent, courant dans le domaine du développement général et de la programmation à des fins variées. Sa popularité repose sur sa lisibilité et sa syntaxe claire, ce qui permet aux développeurs de se concentrer davantage sur la résolution de problèmes plutôt que sur la syntaxe elle-même. Plusieurs paradigmes de programmation sont supportés par Python, tels que la programmation orientée objet, impérative et fonctionnelle. [84]

4.3.2 JavaScript

JavaScript est un langage dynamique de programmation qui joue un rôle crucial dans la création de sites web interactifs. Il offre la possibilité d'intégrer des fonctionnalités avancées sur des pages web, comme l'affichage de mises à jour de contenu en temps réel, des cartes interactives, des graphiques animés en 2D/3D, des juke-box vidéo défilants, et bien d'autres encore. [85]

4.3.3 Dart

Dart est un langage de programmation optimisé pour le développement d'applications rapides sur toutes les plateformes. Il est conçu pour être un langage de programmation productif pour le développement multiplateforme, associé à une plateforme d'exécution flexible pour les Framework d'applications. Dart est souvent utilisé avec le Framework Flutter pour développer des applications mobiles, web et de bureau.[86]

4.4 Algorithmes utilisés :

Pour notre système puisse atteindre le but de la recommandation des points d'intérêts on a choisi l'utilisation des algorithmes qui sont de notre proposition (basé sur le contenu, confiance implicite) ainsi que d'autre algorithmes connu et bien populaire dans ce contexte (algorithme broda comme une solution de vote, recommandation basée sur la confiance explicite, filtrage collaborative basé sur le voisinage)

4.4.1 L'algorithme de recommandation collaborative a base mémoire :

Cet algorithme a pour un but de prédire des évaluations des pois non découverte déjà par l'utilisateur, il commence par se connecter à notre base de données MongoDB à l'aide de module Mongoose. Elle récupère ensuite tous les profils d'utilisateurs et tous les POI's depuis la base de données. En utilisant ces données, il crée la matrice utilisateur-élément. Ensuite, il récupère la ligne correspondant à l'utilisateur cible à depuis de cette matrice. Pour qu'il puisse ensuite parcourir chaque élément non évalué pour l'utilisateur courant. Pour chaque poi, elle calcule une évaluation prévue en utilisant la similarité cosinus entre l'utilisateur cible et les autres utilisateurs voisins qui ont évalué les mêmes POI. Elle normalise ensuite cette évaluation prévue et l'ajoute à une liste des évaluations prévues pour chaque pois associer à un score. L'implémentation de cette étape en langage JavaScript est la suivante :

```
for (let poiIndex = 1; poiIndex < targetUserRow.length; poiIndex++) {
  if (targetUserRow[poiIndex] === null) {
    let numerator = 0;
    let denominator = 0;

    for (let neighborRow of userItemMatrix) {
      if (neighborRow[0] !== userId && neighborRow[poiIndex] !== null) {
        const similarity = cosineSimilarity(targetUserRow.slice(1), neighborRow.slice(1)); // Exclude userId from similarity calculation
        numerator += similarity * neighborRow[poiIndex];
        denominator += similarity;
      }
    }

    const predictedRating = denominator === 0 ? null : numerator / denominator;
    const normalizedRating = predictedRating !== null ? predictedRating / 5 : null;
    if (normalizedRating !== null && !isNaN(normalizedRating)) {
      predictedRatings.push({ poiId: pois[poiIndex - 1].id, score: normalizedRating });
    }

    // Adjust poiIndex to map to correct POI
  }
}
```

4.4.2 L'algorithme de recommandation a base contenu :

L'utilisation de ce genre d'algorithme assure toujours l'actualisation des recommandations des pois pour notre utilisateur vu qu'il base sur un calcul des nouveaux poids pour chaque catégorie en basant sur les activités d'utilisateur.

Il commence tout d'abord par la combinaison des activités de l'utilisateur courant à partir du paramètre **userProfile** avec celles déjà présentes dans le profil récupéré depuis la base de données. Ces activités comprennent les clics, les likes, les commentaires et le temps passé sur chaque POI.

Ensuite, elle calcule la fréquence de chaque catégorie en se basant sur les activités de l'utilisateur et le temps passé sur chaque POI. Ces informations sont utilisées pour

déterminer les préférences de l'utilisateur en termes de catégories de POI. Une mise à jour du profil utilisateur nécessaire dans la base de données avec les catégories triées par fréquence. Comme une dernière étape, il recommande des POI pour chaque catégorie en calculant les scores pour chaque POI dans la catégorie en fonction de la position de la catégorie dans la liste triée des catégories.

Les POI recommandés sont retournés sous forme d'une liste associée avec leur score.

L'implémentation de cette étape en langage JavaScript :

```
const calculateScoreBySingleCategory = async (category, totalCategoriesLength, index) =>{
  // Calculate weight for each category
  const categoryWeight = 1 - (index / totalCategoriesLength);

  try {
    // Fetch POIs for the specified category
    const poisForCategory = await POI.find({ category });

    // Calculate score for each POI in the category
    const poisWithScore = poisForCategory.map((poi, index) => ({
      poiId: poi._id.toString(), // Convert Mongoose ObjectId to a String
      score: categoryWeight // Calculate score based on position in sorted categories
    }));

    return poisWithScore;
  } catch (error) {
    console.error("Error fetching POIs for category:", error);
    throw error;
  }
}
```

4.4.3 L'algorithme de recommandation a base confiance explicite :

Cet algorithme a le job de recommander des pois a base confiance explicite qui signifie une confiance fournie par l'utilisateur, il propose une solution de recommandation collaborative a base mémoire presque identique au premier algorithme proposé, sauf que seulement la considération des amis explicites avec une confiance élevée, et aussi aucun calcul de similarité est nécessaire grâce à l'explicité. L'implémentation de cette étape en langage JavaScript :

```
async recommendPOIsBasedOnTrustImplicit(userId) {
  async function calculateTrust(userId) {
    try {
      // Fetch user profile
      const userProfile = await Profile.findOne({ "userId": userId }).exec();
      // Get user's preferences
      const userPreferences = userProfile.preferences;

      // Get the list of all users except the current user
      const allUsersExceptCurrent = await Profile.find({ "userId": { $ne: userId } });
      // Extract the user IDs of all users except the current user
      const userIdsExceptCurrent = allUsersExceptCurrent.map(user => user.userId);
      // Now, set the 'friends' variable to contain all user IDs except the current user's user ID
      const friends = userIdsExceptCurrent;
      // Fetch profiles of friends
      const friendProfiles = await Profile.find({ "userId": { $in: friends } }).exec();
      // Calculate trust for each user
      const trustScores = [];
      for (const user of friendProfiles) {
        // Get friend's preferences
        const friendPreferences = user.preferences;
        // Calculate similarity between user and friend preferences
        const categorySimilarity = calculateCategorySimilarity(userPreferences.categories, friendPreferences.categories);
        const favoritePoisSimilarity = calculatePoisSimilarity(userPreferences.favoritePois, friendPreferences.favoritePois);
        const visitedPoisSimilarity = calculatePoisSimilarity(userPreferences.visitedPois, friendPreferences.visitedPois);
        // Combine similarity scores (you can adjust weights as needed)
        const trustScore = (categorySimilarity + favoritePoisSimilarity + visitedPoisSimilarity) / 3;
        trustScores.push({ friendId: user.userId, trustScore });
        console.log(user.userId, trustScore);
      }
      // Sort users by trust score
      trustScores.sort((a, b) => b.trustScore - a.trustScore);
      return trustScores;
    }
  }
}
```

4.4.4 L'algorithme de recommandation a base confiance implicite :

Cet algorithme prend le rôle de recommandé des pois d'une manière totalement implicite .sans aucune information fournit par l'utilisateur ce algo est capable à faire des recommandation en basant sur des différent calcul et merge de similarité , voici un aperçu détaillé à propos le calcul des similarités : pour le calcul de la similarité entre les activités de l'utilisateur et celles de ses amis en l'algorithme vide deux grands aspects qui sont respectivement : la similarité des activités et la similarité des évaluations.

- Pour la similarité des activités, il utilise la mesure de similarité de Jaccard pour identifier les activités communes, telles que les clics, les likes et les commentaires sur les mêmes points d'intérêt (POIs). La similarité est ensuite calculée comme le rapport entre la taille de l'intersection de ces ensembles et la taille de leur union.

Pour le calcul de la similarité des évaluations, il compare les évaluations des POIs communs entre l'utilisateur courant et chaque autre utilisateur, attribuant une similarité en fonction de la différence entre les évaluations. Ces calculs sont effectués pour chaque utilisateur, la confiance implicite est ensuite calculée comme une moyenne pondérée de ces similarités. Plus la similarité est élevée, plus la confiance est grande dans les recommandations de POIs de cet utilisateur. Enfin, ces scores de confiance sont utilisés pour pondérer les scores de similarité des activités et des évaluations lors de la recommandation de POIs. L'implémentation de cette étape en langage JavaScript :

```
// Helper function to calculate similarity between arrays of categories or POIs
function calculateCategorySimilarity(categories1, categories2) {
  // Calculate Jaccard similarity between categories
  const intersection = categories1.filter(category => categories2.includes(category));
  const union = [...new Set([...categories1, ...categories2])];
  return intersection.length / union.length;
}

function calculatePoisSimilarity(pois1, pois2) {
  // Calculate Jaccard similarity between points of interest
  const intersection = pois1.filter(poi => pois2.includes(poi));
  const union = [...new Set([...pois1, ...pois2])];
  return intersection.length / union.length;
}
```

4.4.5 Cosinus similarité :

Calcule la similarité cosinus entre deux vecteurs. Elle le fait en effectuant trois opérations principales : le produit scalaire des vecteurs, le calcul de leurs magnitudes respectives,

puis en divisant le produit scalaire par le produit des magnitudes pour obtenir le score de similarité cosinus. Ce score varie entre -1 et 1, où 1 représente une similitude parfaite, 0 indique aucune similitude, et -1 signifie une dissimilarité totale. L'implémentation de cette étape en langage JavaScript :

```
// Cosine similarity function
function cosineSimilarity(vec1, vec2) {
  const dotProduct = vec1.reduce((acc, val, i) => acc + val * vec2[i], 0);
  const magnitudeVec1 = Math.sqrt(vec1.reduce((acc, val) => acc + val * val, 0));
  const magnitudeVec2 = Math.sqrt(vec2.reduce((acc, val) => acc + val * val, 0));
  return dotProduct / (magnitudeVec1 * magnitudeVec2);
}
```

4.4.6 Jaccard similarité :

La similarité de Jaccard utilisée dans les deux fonctions

CalculateCategorySimilarity même aussi **calculateCategorySimilarity**, son rôle joue sur la comparaison et la similarité entre deux ensembles avec les mêmes attributs dans notre cas des pois ou bien des catégories en examinant les éléments communs et en les comparant à l'ensemble total des éléments uniques. Le coefficient de similarité obtenu, toujours variant entre 0 et 1 plus le coefficient approche du 1 plus la similarité est forte, aussi pour le 0, plus le coefficient varie vers 0 plus l'indication de la différence entre les éléments.[87]. La **Figure 21** présente L'équation de similarité de Jaccard qui en a utilisé dans le système.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Figure -21- L'équation de similarité de Jaccard [88].

4.4.7 Broda technique de vote :

Pour chaque recommandation, le score initial est égal à la longueur de la liste moins l'index de la recommandation. Ensuite, pour chaque autre liste de recommandations, il vérifie si le POI est également présent dans cette liste. Si oui, il ajoute la position du POI dans cette autre liste au score final, ajusté par la longueur de la liste. Le score final est ensuite enregistré dans une carte avec le POI comme clé. Cette approche vise à pondérer les

recommandations en fonction de leur position dans les listes de recommandations, favorisant les Pois recommandés à des positions inférieures dans d'autres listes. L'implémentation de cette étape en langage JavaScript :

```
const allRecommendations = [
  filteredCollabRecommendations,
  filteredTrustRecommendations,
  filteredAdditionalRecommendations,
  filteredImplicitTrustRecommendations
];

allRecommendations.forEach((recommendations, listIndex) => {
  recommendations.forEach((recommendation, index) => {
    const poiId = recommendation.poiId;
    const score = recommendations.length - index; // Calculate score based on index
    let finalScore = score; // Initial final score

    allRecommendations.forEach((otherRecommendations, otherListIndex) => {
      // Skip current list
      if (otherListIndex === listIndex) return;

      const indexInOtherList = otherRecommendations.findIndex(otherRecommendation => otherRecommendation.poiId === poiId);
      if (indexInOtherList !== -1) {
        // POI found in other list, add its position to final score
        finalScore += otherRecommendations.length - indexInOtherList + (6 - otherRecommendations.length); // Adjust score based on index
      }
    });

    console.log(`Final score for POI with poiId ${poiId}: ${finalScore}`);
    votes.set(poiId, finalScore);
  });
});
```

4.5 Evaluation et Analyse

Dans cette phase on va essayer de comprendre le comportement du système on analyse ses résultats de recommandation pour des différents utilisateurs ont basé sur des différentes techniques de mesure, d'où le système a été testé avec 15 utilisateurs, et environ 30 poi avec un échantillon de 15 catégories pour les différents poi. L'évaluation du système de recommandation wanderwise va être dirigée via deux modes de vision, d'où la première va être une visualisation et évaluation des résultats dans un niveau le plus abstrait possible comme étant une vision en boîte noire, pour un but de envisager le comportement de système d'une manière générale, ensuite une nécessité a une étude de ce dernier d'une manière plus approfondie d'où on va discuter et détailler les différents piliers qui jouent un rôle dans la recommandation comme étant une évaluation boîte blanche.

4.5.1 Les métriques d'évaluation :

Pour l'évaluation, nous avons calculé de F-Mesure. La mesure F (F-score ou F1-score) est un indicateur de précision d'un test. Cette valeur rassemble à la fois la précision et le rappel. La capacité du modèle à reconnaître tous les éléments positifs est évaluée par le

rappel. Cela implique que la proportion d'éléments positifs réels (vrais positifs) correctement identifiés parmi tous les éléments positifs réels est évaluée. En divisant le nombre de vrais positifs par la somme des vrais positifs et des faux négatifs, on obtient le rappel. La précision est évaluée en fonction de la proportion d'éléments positifs identifiés par le modèle qui sont effectivement favorables. En divisant le nombre de vrais positifs par la somme des vrais positifs (VP) et des faux positifs, on obtient le calcul exact.

		Prédit	
		0	1
Réel	0	TN	FP
	1	FN	TP

Tableau-7-Matrice de confusion [89]

Ces deux mesures sont combinées dans la mesure F en une seule fonction. Elle est calculée en utilisant la formule suivante :

La mesure F permet de donner une vue plus globale de la performance d'un modèle, en tenant compte à la fois de sa capacité à identifier correctement les éléments positifs (rappel) et à minimiser les fausses alarmes (précision).

4.5.1.1 Evaluations

Dans cette partie, nous exposerons une vision globale du système en prenant en considération toutes les valeurs de recommandation reçues, sans expliquer leur méthode de calcul. Par la suite, nous effectuerons des mesures d'évaluation comme la précision, le rappel et la mesure F, dans le but d'évaluer les performances globales de notre système de recommandation.

4.5.1.1.1 Top-k=6

Voici un aperçu sur des résultats qui englobe et qui prend en considération tous les utilisateurs qui font partie de cette évaluation. Les deux figures rapportent les résultats retournés par le système.

```

recommended poi's :

65e4799f70bc22440e30e690,29
65d4bfd1669a6f74936210b5,25
65e480c270bc22440e30e691,19
6625717748bbd95bc96384ea,13
662c024d3306b92963adbbf2,13
662570010549be8a7e2e8711,11
662c00033306b92963adbbe6,11
6625714b48bbd95bc96384e7,10
6625712c48bbd95bc96384e6,9
662c01633306b92963adbbe9,9
662c010e3306b92963adbbe8,8
662c00763306b92963adbbe9,7
6625716248bbd95bc96384e8,6
662c02b23306b92963adbbf4,6
662571a748bbd95bc96384ec,5
662c00ee3306b92963adbbec,4
6625716a48bbd95bc96384e9,3

662c01ef3306b92963adbbf0,2
662c00033306b92963adbbe6,1
6625711248bbd95bc96384e5,1

```

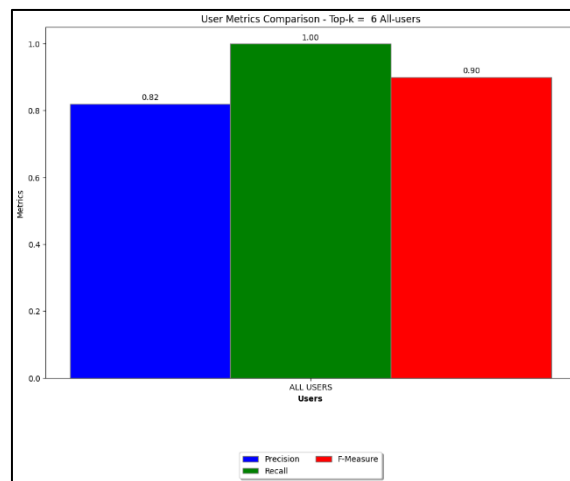
```

evaluated :

65e480c270bc22440e30e691,19
6625717748bbd95bc96384ea,13
662c024d3306b92963adbbf2,13
662570010549be8a7e2e8711,11
65d4bfd1669a6f74936210b5,11
662c00033306b92963adbbe6,11
6625711248bbd95bc96384e5,10
65e4799f70bc22440e30e690,10
662571a748bbd95bc96384ec,10
662c00763306b92963adbbe9,7
662c010e3306b92963adbbe8,7
6625714b48bbd95bc96384e7,7
6625716248bbd95bc96384e8,4
662c01ef3306b92963adbbf0,3

```

Ces résultats englobent tous les pois des tous utilisateurs, comme on peut remarquer que le système a retourné 20 pois des 28 pois dans la base de données, 17 POI recommandé est pertinent qui veut signifier un score supérieur ou égale au seuil qui est 3 avec 3 pois recommandés mais non pertinent vu que le score est strictement inférieur au seuil. Aussi vous pouvez remarquer que 14 pois des 20 POI recommandés ont été évalué par les utilisateurs, voici les résultats obtenus dans la figure qui suite.



On peut remarquer que généralement le système a retourné des scores plutôt élevés, pour les 3 métriques : précision, rappel et f-Mesure. Selon la formule déjà discuter des métriques, ce qu'on peut conclure pour le moment ce que seule la valeur de la précision, joue un rôle dans le calcul de f-mesure vu que le rappel est pour 1, qui veut signifier qu'aucun bruit ou sensibilité ont été détectés.

4.5.1.1.2 Top-k = 4

Dans ce cas, nous avons réduit le nombre de POIs retournés à 4. Les résultats sont presque similaires, seulement cette fois les pois recommandés et pertinent était 18, fasse au non pertinent avec 2 pois. Pour les évaluations, les résultats vont rester sans changement comme les deux figures si dessous les montres.

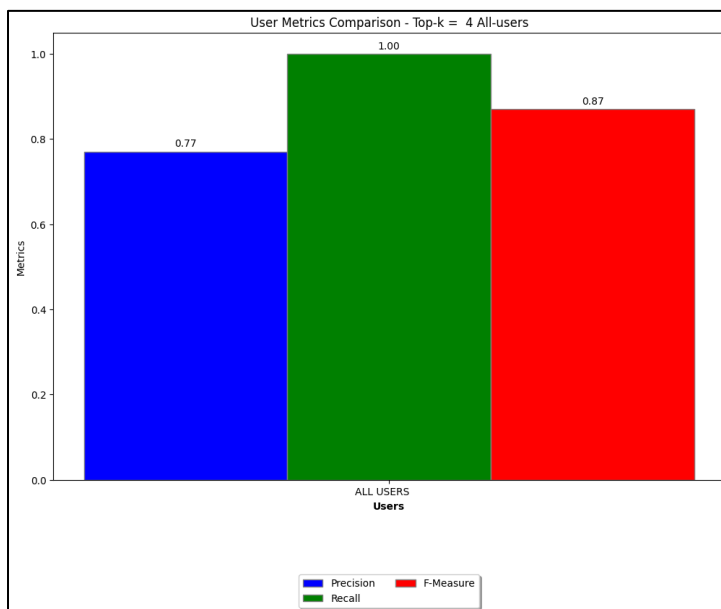
```
recommended :
65e4799f70bc22440e30e690,29
65d4bfd1669a6f74936210b5,25
65e480c270bc22440e30e691,19
6625717748bbd95bc96384ea,13
662c024d3306b92963adbbf2,13
662570010549be8a7e2e8711,11
662c00033306b92963adbbe6,11
6625714b48bbd95bc96384e7,10
6625712c48bbd95bc96384e6,9
662c01633306b92963adbbe,9
662c010e3306b92963adbbe,8
662c00763306b92963adbbe,7
6625716248bbd95bc96384e8,6
662c02b23306b92963adbbf4,6
662571a748bbd95bc96384ec,5
662c00ee3306b92963adbbec,4
6625716a48bbd95bc96384e9,3
662c01ef3306b92963adbbf0,2

662c00033306b92963adbbe6,1
6625711248bbd95bc96384e5,1
2

evaluated :
65e480c270bc22440e30e691,19
6625717748bbd95bc96384ea,13
662c024d3306b92963adbbf2,13
662570010549be8a7e2e8711,11
65d4bfd1669a6f74936210b5,11
662c00033306b92963adbbe6,11
6625711248bbd95bc96384e5,10
65e4799f70bc22440e30e690,10
662571a748bbd95bc96384ec,10
662c00763306b92963adbbe9,7
662c010e3306b92963adbbe,7
6625714b48bbd95bc96384e7,7
6625716248bbd95bc96384e8,4
662c01ef3306b92963adbbf0,3

14
```

Il est possible de constater à travers la figure qu'il n'y a généralement pas de modification significative par rapport aux résultats précédents, même avec un seuil différent, en raison d'un échantillon simple des pois et pour les trois métriques : précision, rappel et f-Test. Cependant, il est évident que la précision cette fois-ci est inférieure à celle de top-k = 6, ce qui suggère que le choix des bonnes mesures est essentiel lors de la phase d'évaluation afin de maîtriser pleinement le potentiel du système. L'impact de cette décision sera particulièrement important avec un grand nombre de pois.



4.5.1.2 Résultats des tests :

- **Phase 1** : Le système fournira des recommandations basées sur le contenu selon les catégories partagées par l'utilisateur. Comme illustré dans la figure ci-dessous, le système a pu recommander 6 pois pour les différentes catégories de restaurants, de jardins publics et de centres commerciaux.

```
third user : oum Yacine
date : 26/04/2024 20:15 - 21:22
email : "oumyacine@gmail.com"
categorie sélectionne au départ : Restaurants , Jardins Publics , Centres Commerciaux

phase01 :

first POIS recommendation : 662571a748bd95bc96384ec , 65d4bfd1669a6f74936210b5 , 662570010549be8a7e2e8711 , 662c00033306b92963adbb6
662c008e3306b92963adbbea , 662c02b23306b92963adbbf4

expected pois : 662571a748bd95bc96384ec , 65d4bfd1669a6f74936210b5 , 662570010549be8a7e2e8711 , 662c00033306b92963adbb6
662c008e3306b92963adbbea , 662c02b23306b92963adbbf4
```

- **Phase 2** : Comme indiqué dans la figure ci-dessus, la deuxième étape de l'évaluation consiste à recueillir les activités des utilisateurs telles que les clics, les likes, les commentaires, les évaluations et le TimeSpent. Ensuite, le système fournira des recommandations en tant que résultat de partage d'activités. Les différents agents du système sont utilisés, à l'exception de l'agent qui repose sur la recommandation explicite, car dans cette section, l'utilisateur ne découvre pas encore la partie où il va choisir ses propres amis (phase 03), les résultats sont présentés sous la forme d'une liste contenant des cartes d'objets, chaque carte contenant les poiId provenant de la base de données, et chaque poi est également attribué un score de 0 à 1. Tous les résultats de l'agent sont présentés dans des formats similaires.

```
phase 02 :

collected data from user Agent : collected Data:
//Flutter (26200): clicks: [662570010549be8a7e2e8711: 2, 662571a748bd95bc96384ec: 1]
//Flutter (26200): likes: [662570010549be8a7e2e8711, 662571a748bd95bc96384ec]
//Flutter (26200): comments: {}
//Flutter (26200): Rating: [662570010549be8a7e2e8711: 5.0]
//Flutter (26200): Time Spent: [662570010549be8a7e2e8711: 20, 662571a748bd95bc96384ec: 6]

sorted Recommendations for content-based: [
  { poiId: '662570010549be8a7e2e8711', score: 1 },
  { poiId: '662c008e3306b92963adbbea', score: 1 },
  { poiId: '662571a748bd95bc96384ec', score: 0.666666666666667 },
  { poiId: '662c02b23306b92963adbbf4', score: 0.666666666666667 },
  { poiId: '65d4bfd1669a6f74936210b5', score: 0.333333333333333 },
  { poiId: '662c00033306b92963adbb6', score: 0.333333333333333 }
]

sorted Recommendations for Implicit-trust-Agent: [
  { poiId: '65d4bfd1669a6f74936210b5', score: 0.4 },
  { poiId: '65e4799f70bc22440e30e690', score: 0.4 },
  { poiId: '6625716248bd95bc96384e8', score: 0.333333333333333 },
  { poiId: '65e480c270bc22440e30e691', score: 0.2 }
]

final recommendations: [
  { poiId: '662570010549be8a7e2e8711', finalScore: 6 },
  { poiId: '65d4bfd1669a6f74936210b5', finalScore: 8 },
  { poiId: '662c008e3306b92963adbbea', finalScore: 5 },
  { poiId: '662571a748bd95bc96384ec', finalScore: 4 },
  { poiId: '662c02b23306b92963adbbf4', finalScore: 3 },
  { poiId: '65e4799f70bc22440e30e690', finalScore: 5 },
  { poiId: '6625716248bd95bc96384e8', finalScore: 4 },
  { poiId: '662c00033306b92963adbb6', finalScore: 1 },
  { poiId: '65e480c270bc22440e30e691', finalScore: 3 }
],
```

- **Phase 3** : maintenant pour la dernière phase d'où le niveau de la recommandation va être le plus abstrait possible vu que le système va proposer des recommandations des pois après avoir garanti l'orchestration des différents résultats à partir des différents agents, vous allez remarquer que 3 agents, seulement, ont retourné des résultats de recommandation, sauf pour le collab-agent vu qu'il manque des évaluations(rating) des pois par les autres utilisateurs.
- **Finale recommandation** : pour les recommandations finales pour les phases 2 et 3, nous avons utilisé la technique de vote broda déjà entamer dans la définition des algorithmes utilisés, cette technique consiste à pondérer les recommandations en fonction de leur position dans les listes de recommandations, favorisant les Pois recommandés à des positions inférieures dans d'autres listes.

```
Sorted Recommendations for Trust-Implicit agent : [
  { poiId: '65e4799f70bc22440e30e690', score: 0.0666666666666667 },
  { poiId: '65e480c270bc22440e30e691', score: 0.0666666666666667 },
  { poiId: '6625716248bbd95bc96384e8', score: 0.0666666666666667 }
]

Sorted Recommendations for collab-agent: []

Sorted Recommendations for trust-Explicit agent: [
  { poiId: '65d4bfd1669a6f74936210b5', score: 0.4 },
  { poiId: '65e4799f70bc22440e30e690', score: 0.4 },
  { poiId: '6625716248bbd95bc96384e8', score: 0.3333333333333333 },
  { poiId: '65e480c270bc22440e30e691', score: 0.2 }
]

Sorted Recommendations for content-based agent: [
  { poiId: '662570010549be8a7e2e8711', score: 1 },
  { poiId: '662c008e3306b92963adbbca', score: 1 },
  { poiId: '662571a748bbd95bc96384ec', score: 0.6666666666666667 },
  { poiId: '662c02b23306b92963adbbf4', score: 0.6666666666666667 },
  { poiId: '65d4bfd1669a6f74936210b5', score: 0.3333333333333337 },
  { poiId: '662c0033306b92963adbbe6', score: 0.3333333333333337 }
]

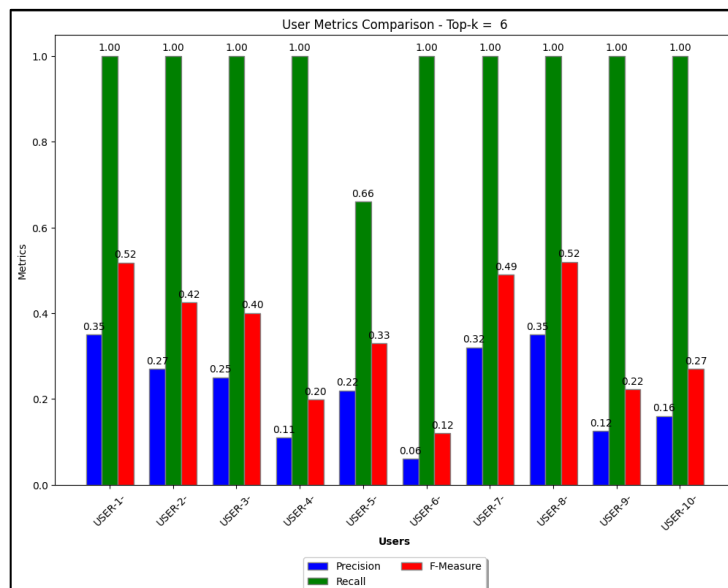
final recommendation :[
  { poiId: '65e4799f70bc22440e30e690', finalScore: 9 },
  { poiId: '662570010549be8a7e2e8711', finalScore: 6 },
  { poiId: '65d4bfd1669a6f74936210b5', finalScore: 8 },
  { poiId: '662c008e3306b92963adbbca', finalScore: 5 },
  { poiId: '662571a748bbd95bc96384ec', finalScore: 4 },
  { poiId: '65e480c270bc22440e30e691', finalScore: 8 },
  { poiId: '6625716248bbd95bc96384e8', finalScore: 8 },
  { poiId: '662c02b23306b92963adbbf4', finalScore: 3 },
  { poiId: '662c0033306b92963adbbe6', finalScore: 1 }
]
```

4.5.2 Analyse des résultats

4.5.2.1 Top-K = 6

Les résultats fournis ci-dessous représentent les résultats des recommandations finaux pour nos 10 utilisateurs sous format d'un diagramme représenté par le f mesure, précision et le rappel pour chaque utilisateur dans une échelle de 0 à 1. Entamant ensemble le détail des résultats fournis.

- **Rappel Mesure :** Alors pour le rappel mesure, nous pouvons remarquer que les résultats ont été presque identiques pour une métrique de 1.0 qui signifie une sensibilité nulle sauf pour l'utilisateur user-5 qui a un résultat de 0.6 qui signifie une perte légère lors de la phase de calcul des recommandations par le système.
- **Précision :** On peut remarquer que les résultats de l'évaluation varient considérablement en fonction du comportement de chaque utilisateur. Lorsque le résultat approche de 1, cela signifie que l'utilisateur a aimé la majorité des points d'intérêt recommandés dans les trois phases. À l'inverse, un résultat proche de 0, comme pour l'utilisateur User-6, indique que cet utilisateur a peu aimé la majorité des points d'intérêt recommandés.
- **F-mesure :** les résultats des recommandations finaux pour chaque utilisateur n'ont pas dépassé les 60%, cela signifie que la recommandation a été pas assez mauvaise mais aussi non assez bonne. Pour des raisons déjà connues d'où le comportement de l'utilisateur envers des posts via des likes joue un rôle crucial dans les résultats de l'évaluation.

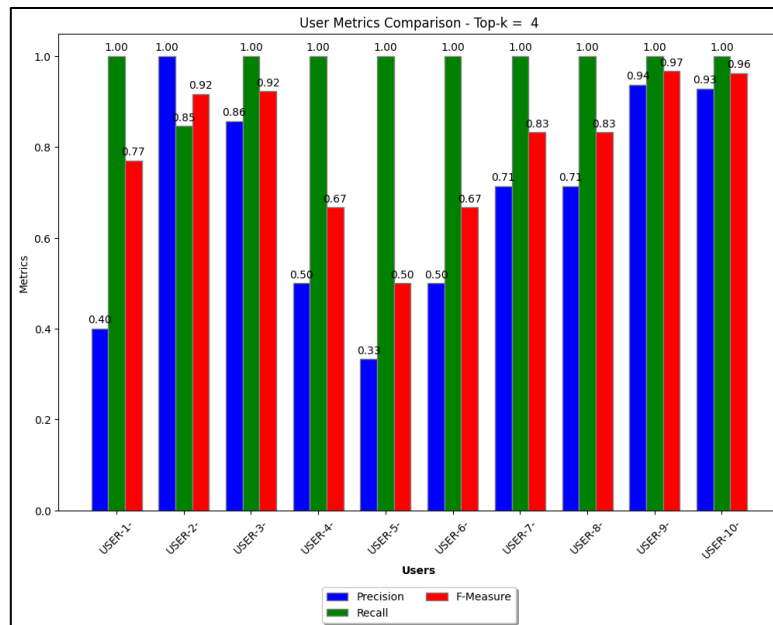


Même avec ces résultats, nous avons remarqué lors de l'analyse que le rappel était assez bon, ce qui signifie un faible taux d'erreur ou une sensibilité élevée. Le problème se situait donc probablement au niveau de la précision. Nous ne pouvons pas valider nos résultats avec une précision basse en raison de l'utilisation inadéquate des attributs d'évaluation. C'est pourquoi nous avons décidé de faire une autre évaluation avec les mêmes résultats, mais en utilisant un top-k différent égal à 4, afin de mieux comprendre le système et de proposer des solutions d'optimisation. Voici les résultats que nous avons constatés.

4.5.2.2 Top-k = 4

Alors comme une conclusion finale on peut remarquer que le même système avec les mêmes données et utilisateurs et lorsque nous avons changé le top-k vers 4, nous pouvons constater qu'à l'aide les résultats si dessous sous format d'un diagramme que les mesures ont montré un changement significatif.

Pour le rappel, nous pouvons constater que nous avons un léger changement. Par contre pour la précision, nous pouvons remarquer qu'il existe un changement remarquable grâce à la sélection des pois recommandés et pertinents avec un score final supérieur ou égale au seuil, qui est le $k/2 = 4/2 = 2$.



Ces observations sont confirmées par l'utilisation de la mesure F, dont la performance peut atteindre jusqu'à 90% pour certains utilisateurs.

Pour résumer, la modification du paramètre top-K a un effet considérable sur les performances du système, notamment en ce qui concerne la précision. Cela souligne l'importance de choisir ce paramètre avec soin en fonction des exigences particulières de l'application et des préférences des utilisateurs.

4.6 Présentation de l'application

4.6.1 Interface de Démarrage

L'interface représente le lancement de notre application mobile pour la découverte des POIs, avec un bouton "Commencer maintenant" en bas

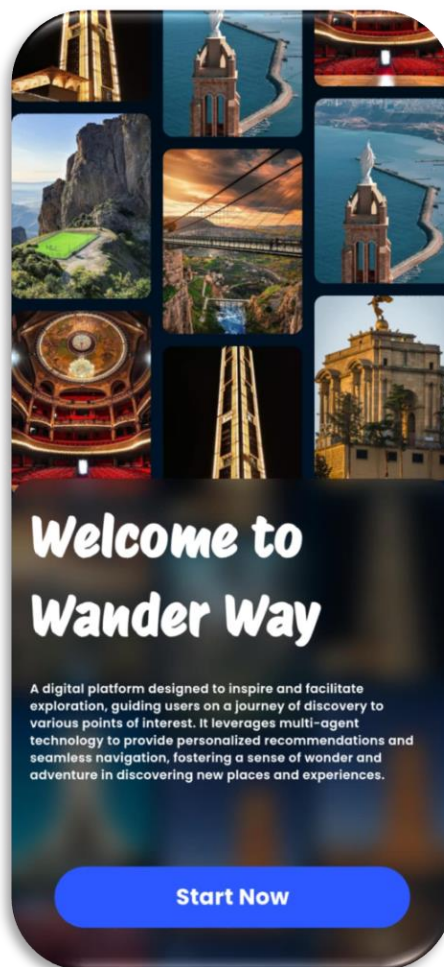
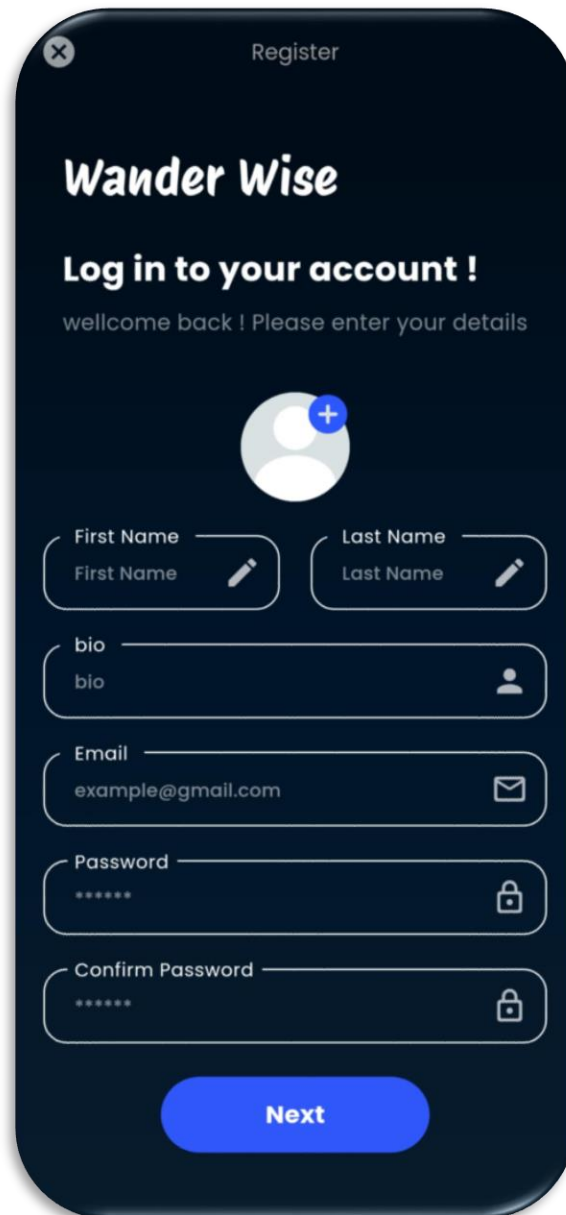


Figure -22- Interface de démarrage

4.6.2 Interface d'inscription

Une interface utilisateur pour l'inscription à un compte WanderWise.



The image shows a mobile application registration screen for 'WanderWise'. The screen has a dark blue background with white text and icons. At the top, there is a close button (X) and the title 'Register'. Below the title is the 'WanderWise' logo. The main heading is 'Log in to your account !' followed by the subtext 'welcome back ! Please enter your details'. In the center, there is a placeholder for a profile picture, represented by a grey circle with a white plus sign. Below this are several input fields: 'First Name' and 'Last Name' (each with a pencil icon for editing), 'bio' (with a person icon), 'Email' (with an envelope icon and the example 'example@gmail.com'), 'Password' (with a lock icon and masked characters '*****'), and 'Confirm Password' (with a lock icon and masked characters '*****'). At the bottom, there is a prominent blue button labeled 'Next'.

Figure -23- Interface d'inscription

4.6.3 Interface d'authentification

Une interface utilisateur pour la connexion.

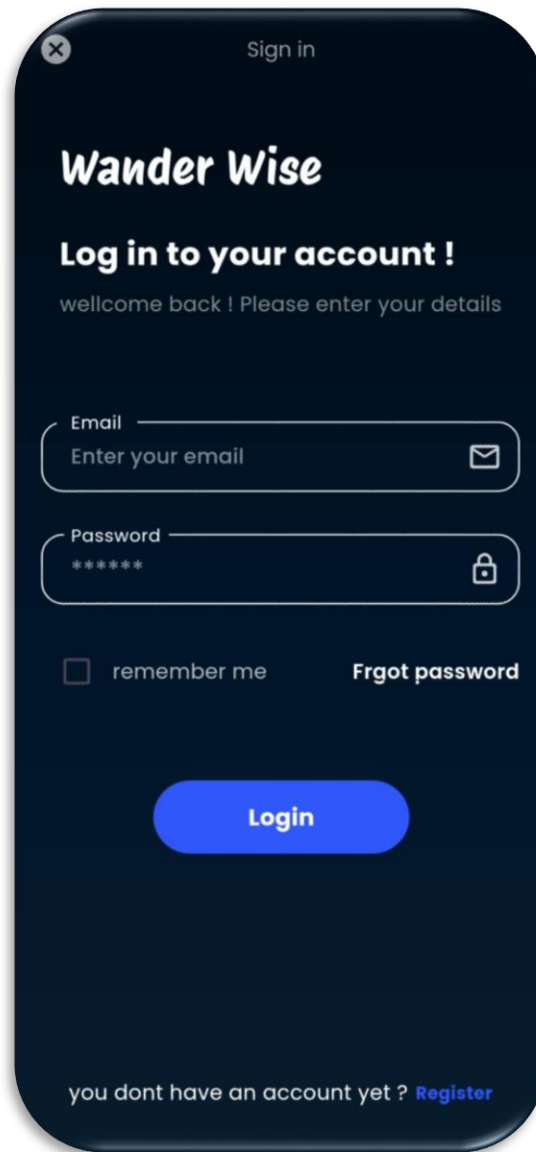


Figure -24- Interface d'authentification

4.6.4 Interface de l'accueil

L'interface montre à l'utilisateur des lieux touristiques pour explorer après la connexion.

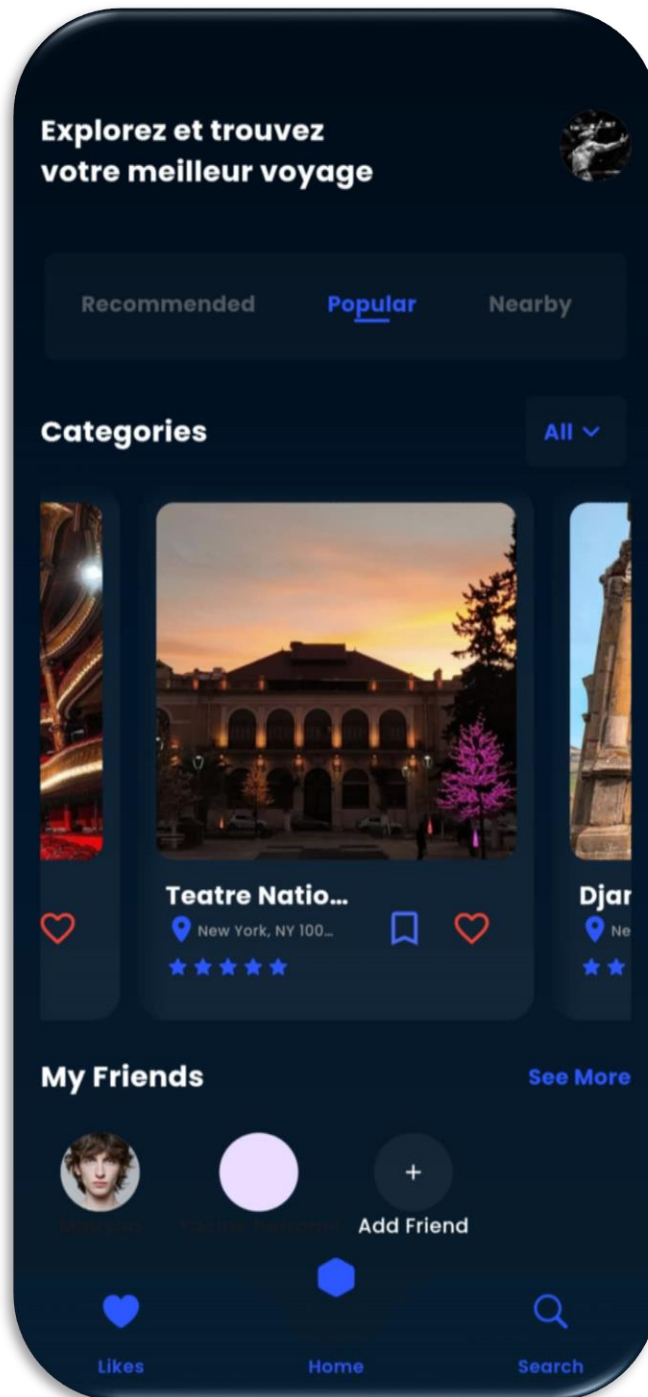


Figure -25- Interface d'accueil

4.6.5 Interface de détail de POI

L'interface présente les détails d'un point d'intérêt.

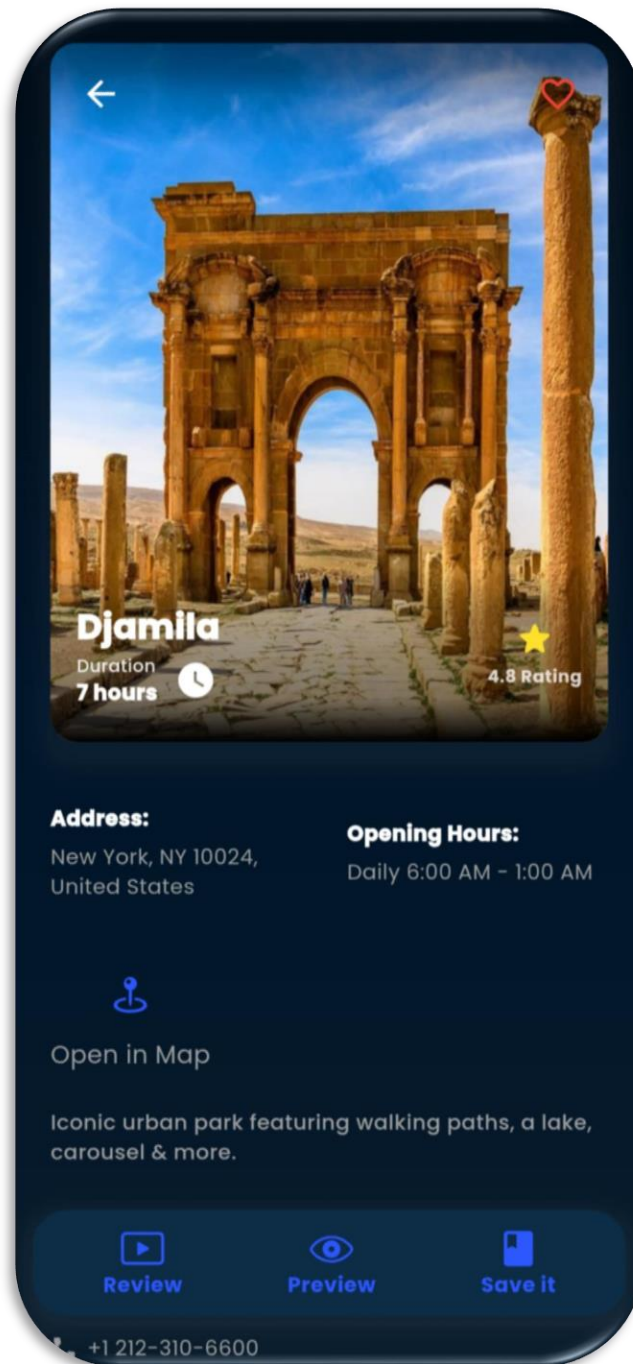


Figure -26- Interface de détail de POI

4.6.6 Interface de recherche Ami

L'interface montre la recherche de contacts d'un ami dans notre WanderWise app.

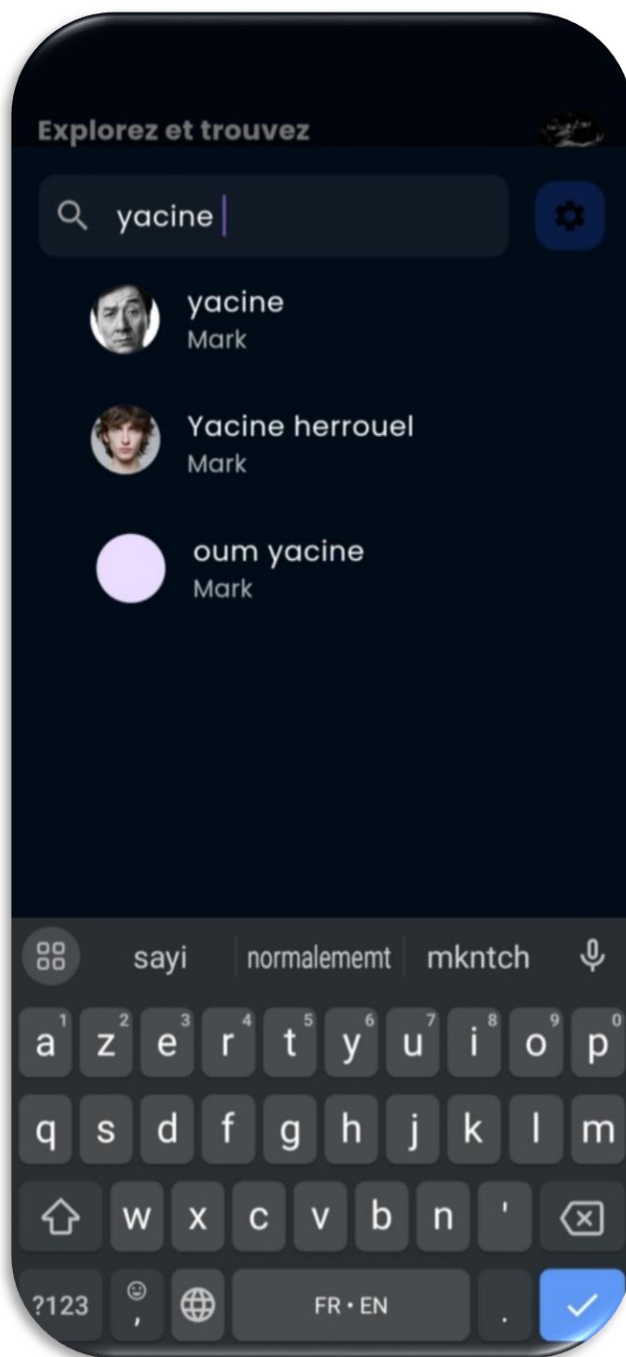


Figure -27- Interface de recherche ami

4.6.7 Interface de recherche POI

L'interface montre la recherche de point d'intérêt dans notre WanderWise app.

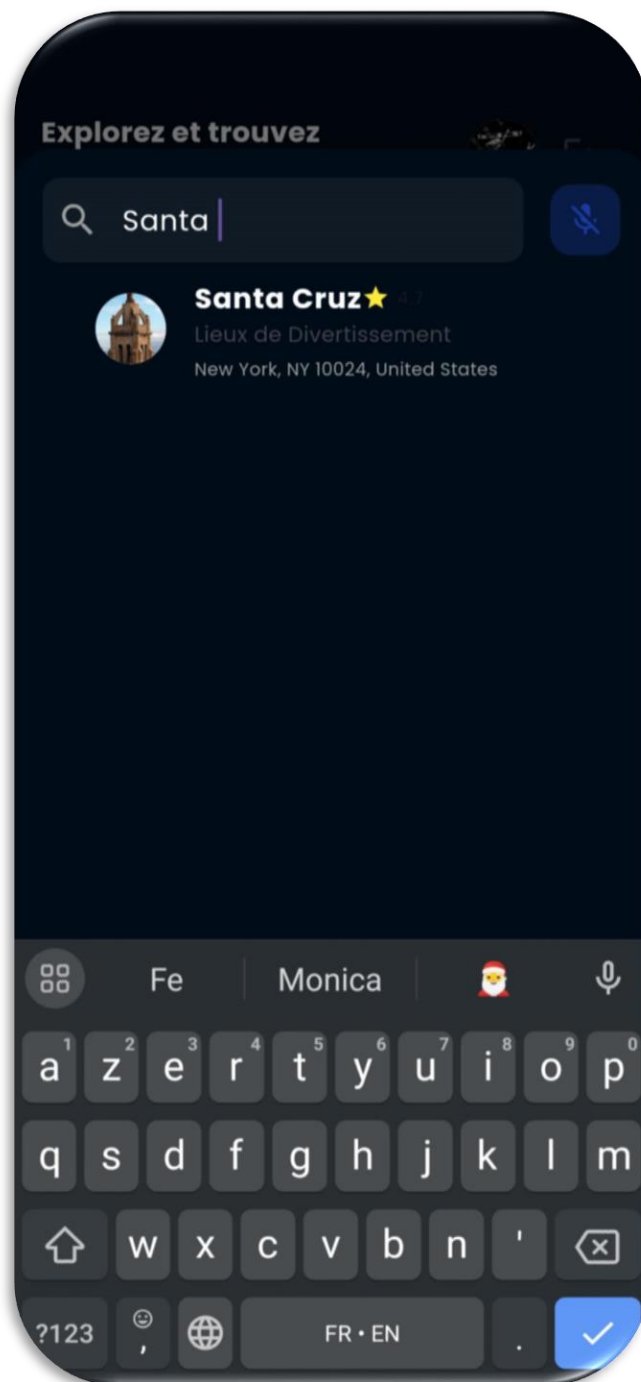


Figure -28- Interface de recherche POI

4.6.8 Interface des catégories

L'interface montre la découverte de services et attractions dans notre WanderWise app.

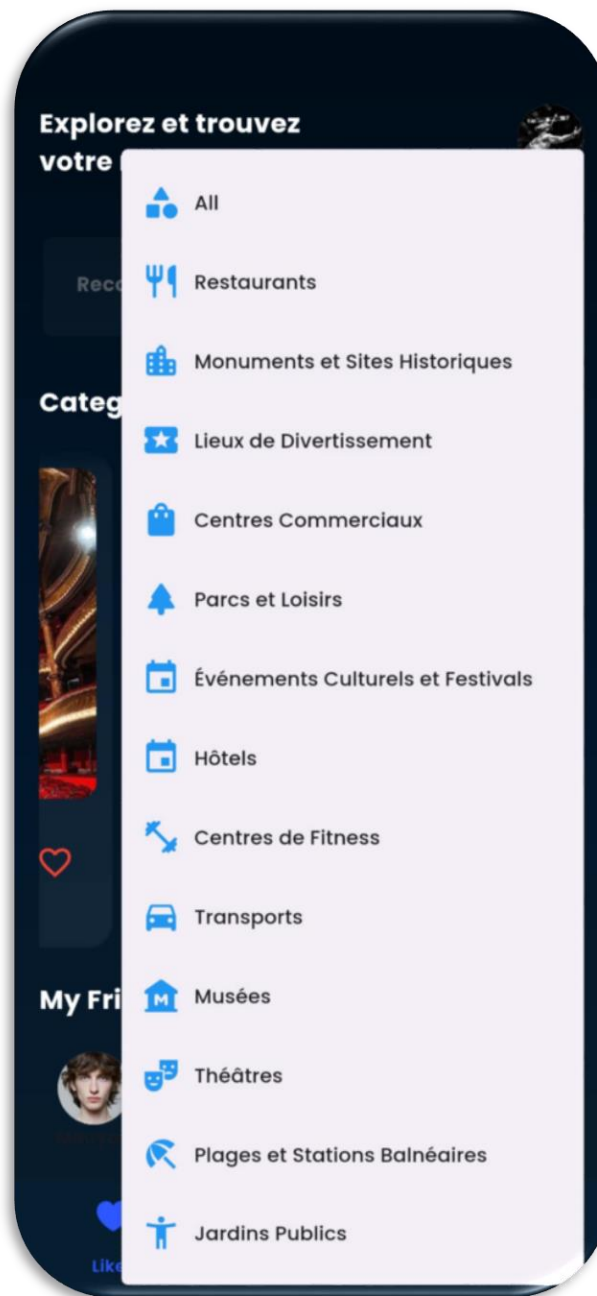


Figure -29- Interface POI catégories.

4.6.9 Interface Profile d'utilisateur

L'interface utilisateur avec des statistiques et des favoris, catégories de profil.

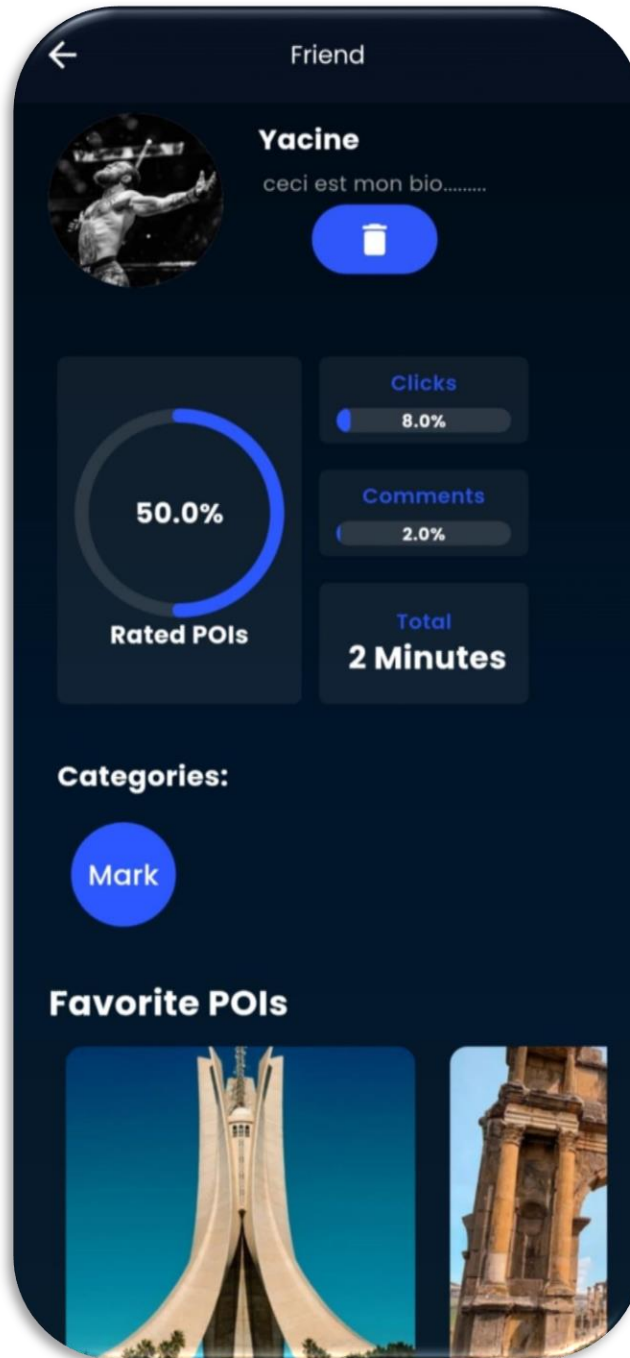


Figure -30- Interface Profile d'utilisateur.

4.7 Conclusion

Au cours de ce chapitre, nous avons exposé les différentes technologies et langages et logiciels employés pour mettre en place notre application mobile WanderWise. Cette application, qui repose sur des algorithmes de recommandation innovants et un écosystème de développement solide, a prouvé son efficacité en proposant des recommandations sur mesure et pertinentes. L'analyse approfondie des résultats a mis en évidence l'importance essentielle des paramètres d'algorithmes, tels que le top-K, pour la précision et le rappel des recommandations. La capacité du système à s'adapter aux préférences des utilisateurs tout en maintenant une grande sensibilité a été confirmée par nos résultats. Dans le futur, d'autres améliorations pourraient encore renforcer la précision et enrichir l'expérience utilisateur.

Conclusion Générale

Ce travail de recherche constitue une analyse complète de la conception d'un système de recommandation de points d'intérêt (POI) avancé, en se basant sur une analyse approfondie des méthodes et des technologies actuelles « SMA ». En quatre chapitres, une étude approfondie des principes des systèmes de recommandation, des nuances des méthodes de filtrage collaboratif, basé sur le contenu, à base de connaissances et hybride a été réalisée.

Notre système a été conçu et modélisé avec minutie, en prenant en considération les exigences en temps réel et la réactivité requise pour une expérience utilisateur optimale. La mise en place a été effectuée en utilisant des outils modernes et des langages de programmation courants, assurant ainsi une plateforme solide et flexible.

La précision et l'efficacité de notre système ont été grandement améliorées grâce aux algorithmes de recommandation que nous avons développés et intégrés, tels que ceux qui reposent sur la confiance implicite et explicite, ainsi que les techniques de similarité et de vote. Avec ces progrès technologiques, il est prévu que l'expérience des utilisateurs soit améliorée en leur offrant des recommandations pertinentes et sur mesure.

En conclusion, notre recherche joue un rôle important dans l'évolution des systèmes de recommandation de POI en proposant une solution novatrice qui utilise les dernières avancées en matière d'algorithmes et de systèmes multi-agents. Les opportunités d'optimisation et de développement de notre travail permettent de créer des applications encore plus performantes et intuitives à l'avenir.

Comme perspectives nous envisageons d'apporter quelques améliorations à savoir :

- ✓ Adaptabilité et personnalisation accrues : Les SMA peuvent gérer des environnements dynamiques et complexes, permettant une personnalisation plus fine des recommandations de POI en fonction des comportements et préférences changeants des utilisateurs.
- ✓ Amélioration de la prise de décision : Grâce à la communication et à la collaboration entre agents, les SMA peuvent combiner des méthodes de recommandation pour offrir des suggestions plus précises et contextuelles.

- ✓ Scalabilité et performance : Les SMA sont particulièrement adaptés pour évoluer et gérer de grandes quantités de données, ce qui est essentiel pour les systèmes de recommandation qui doivent traiter les informations de nombreux utilisateurs et POI.
- ✓ Interaction en temps réel : L'intégration de SMA permettrait une interaction en temps réel avec les utilisateurs, offrant des recommandations instantanées basées sur leur activité actuelle.

Annexe A



Figure -31- Logo de l'application

Annexe B

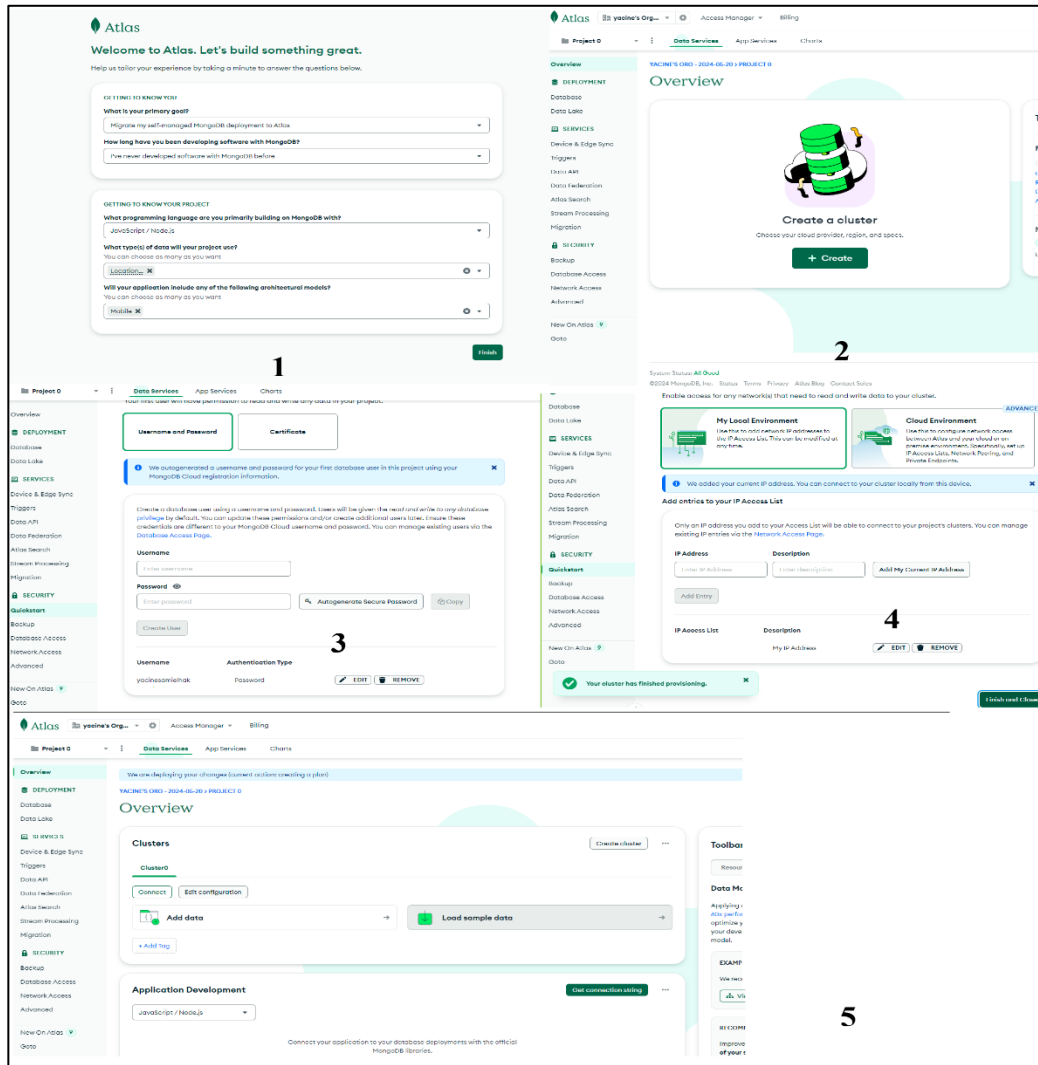


Figure -32- étapes de création une Base de données MongoDB

Obtenir votre base de données MongoDB et accéder aux données de votre application. La création d'un compte MongoDB est très simple, il vous suffit d'accéder au site de MongoDB Atlas (mongodb.com) et de vous connecter avec vos identifiants. Ensuite, suivez les étapes et félicitations tu as une base de données MongoDB.

Bibliographie

- [1] Fokou, K. : Spécialisation en intelligence artificielle. Smals Research <http://www.smalsresearch.be>.
- [2] Taouli, S., Benachenhou, W. : Utilisation de factorisation matricielle sans les systèmes de recommandation sensible au contexte. Mémoire de master, Université Abou Bakr Belkaid – Tlemcen, 2017.
- [3] Dubois, C. : Ne ratez pas les nouveautés Netflix de la semaine du 16 au 22 octobre 2023. PlayTV, Modifié le 3 novembre 2023 à 8h48. Disponible sur : PlayTV. Consulté le [04/17/2024].
- [4] Picot-Clemente, A. : Une architecture générique de Systèmes de recommandation de combinaison d'items. Application au domaine du tourisme. Recherche d'information [cs.IR]. Université de Bourgogne, 2011. Français. [Tel-00688994v1].
- [5] La matrice de factorisation pour la recommandation des POIs. Disponible sur : Université Abdelhamid ibn Badis Mostaganem, e-biblio.univ-mosta.dz. Consulté le [04/17/2024].
- [6] Kuanr, M., Mohapatra, P., Choudhury, S. S.: TSARS: A Tree-Similarity Algorithm-Based Agricultural Recommender System. Dans "Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries", Scrivener Publishing LLC, 2020, pp. 387-400.
- [7] Hadj Henni, M., Dennouni, N., Slama, Z. : Schéma explicatif de SR basé sur le Filtrage collaboratif (FC). Dans "SIG et la recommandation contextuelle des POI", GISDAY proceedings, 29 août 2020.
- [8] Fokou, K. : Introduction aux systèmes de recommandation. Smals Research, 28 juin 2022.
- [9] Tony, A.: All You Need to Know About Collaborative Filtering. Digital Vidya. (Digitalvidya.com).
- [10] Charif ALCHIEKH HAYDAR. Les systèmes de recommandation à base de confiance. Université de Lorraine .2014. Thèse de doctorat.
- [11] Fournier-S'niehotta, R. : Systèmes de recommandation 1e partie, CNAM Paris, 2020-2021.
- [12] Aussem, A. : Introduction aux Réseaux Bayésiens. CNRS.

- [13] Breese, J. S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI), 1998, pp. 43-52.
- [14] Yager, R. R. (2003). Fuzzy logic methods in recommender systems. *Fuzzy Sets and Systems*, 136(2), pp.133-149.Consulté le [04/17/2024].
- [15]M.Maâtallah and H. Seridi, (2010). A Fuzzy Hybrid Recommender System. In the Proc. Of the 1st Inter. Conf. on Machine and Web Intelligence (ICMWI'10), pp. 258-263.Consulté le [04/17/2024].
- [16] Porcel, C., López-Herrera, A. G., & Herrera-Viedma, E. (2009). A recommender system for research resources based on fuzzy linguistic modeling. *Expert Systems with Applications*, 36(3), pp. 5173-5183
- [17] Denoyer, L., Soulier, L. : Clustering et Classification, BI = Business Intelligence Master Data-Science Cours 8, Université Pierre et Marie Curie, lip6.fr. Consulté le [04/17/2024].
- [18] PTC Mathcad : Exemple de factorisation de matrice QR. Utilisation de la fonction QR pour la décomposition de matrices en facteurs Q et R. Disponible sur : support.ptc.com
- [19] CNAM : Arbres de décision. Cours RCP209 — Apprentissage, réseaux de neurones et modèles graphiques.
- [20] Paatero P. and Tapper U., (1994). Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. *Environmetrics*, Vol. 5, pp. 111–126.
- [21] Lee D. and Seung H., (2001). Algorithms for Non-Negative Matrix Factorization. *Advances in Neural Information Processing Systems*, Vol. 13, pp.556–562.
- [22] Pennock, D. M., Horvitz, E., Lawrence, S., & Giles, C. L. (2000, June). Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In Proc. of the Sixteenth conference on Uncertainty in artificial intelligence, pp. 473-480.
- [23] Kim H. and Park H., (2008). Nonnegative Matrix Factorization Based on Alternating Non-negativity-constrained Least Squares and the Active Set Method. *SIAM J. on Matrix Analysis and Applications (SIMAX)*, 30(2), pp.713-730.
- [24] Bobadilla, J., Ortega, F., Hernando, A., & Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *KBS*, 24(8), pp.1310-1316.

- [25] Christakou, C., Vrettos, S., & Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *Inter. Journal on Artificial Intelligence Tools*, 16(05), 771-792.
- [26] Kermarrec, A.-M. : L'algorithme de filtrage collaboratif en cinq atouts. *Mediego*.
- [27] Mise en place d'un système de communication multi-agents. Disponible sur : Université Ibn Khaldoun de Tiaret, dspace.univ-tiaret.dz, 2016. Consulté le [04/17/2024].
- [28] Montaner M., López B., De La Rosa J. L., (2003). A Taxonomy of Recommender Agents on the Internet, *Artificial Intelligence Review*, Vol. 19, pp.285-330. Consulté le [04/17/2024].
- [29] Burke R., (2002): Hybrid recommender systems: Survey and experiments. In *User Modeling and User Adapted Interaction*, 12(4), pp. 331-370. Consulté le [04/17/2024].
- [30] Negre, E. : Les systèmes de recommandation : une catégorisation. *Interstices*, Publié le: 20/09/2018. Disponible sur : Interstices. Consulté le [04/17/2024].
- [31] Schafer J.B., Konstan J., and Reidl J., (1999). Recommender Systems in Ecommerce. *Proc. of the 1st ACM Conf. on Electronic Commerce*, Denver, CO, pp. 158 -166. Consulté le [04/17/2024].
- [32] Adomavicius G. & Tuzhilin A., (2005). Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. In *IEEE Trans. On KDE*, 17(6), pp. 734-749. Consulté le [04/17/2024].
- [33] Filtrage basé sur le contenu dans un système de recommandation utilisant JupyterColab Notebook. *ICHI.PRO*. Disponible sur : (ichi.pro). Consulté le [04/17/2024].
- [34] Comment choisir entre filtrage collaboratif et filtrage basé sur le contenu. *LinkedIn*. Disponible sur : ([LinkedIn](https://www.linkedin.com)). Consulté le [04/17/2024].
- [35] Burke, R.: Hybrid Web Recommender Systems, in "The Adaptive Web", Brusilovsky, P.; Kobsa, A.; Nejdl, W. (Eds.), *Lecture Notes in Computer Science 4321*, Springer-Verlag, 2007, pp. 377-408. Consulté le [04/17/2024].
- [36] Lemdani, R. : Système hybride d'adaptation dans les systèmes de recommandation. Thèse de Doctorat, Université Paris-Saclay, 2016. Disponible sur : theses.hal.science. Consulté le [04/17/2024].
- [37] Du, Y., Ranwez, S., Sutton-Charani, N., Ranwez, V. : Apports des ontologies aux systèmes de recommandation : état de l'art et perspectives, dans les actes des 30es Journées Francophones d'Ingénierie des Connaissances (IC 2019), AFIA, Juillet 2019, Toulouse, France, pp. 64-77. Consulté le [04/17/2024].

- [38] Knowledge-based recommendation, SlideServe, disponible sur:SlideServe.Consulté le [04/17/2024].
- [39] Le, N. L., Abel, M.-H., Gouspillou, P. : Construction d'un système de recommandation basé sur des contraintes via des graphes de connaissances. Rapport technique, Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of ComplexSystems), 20231, arXiv :2306.03247.
- [40] Dadoun, A.: Introduction to Knowledge Graph-Based Recommender Systems. Towards Data Science. Disponible sur :(Towards Data Science). Publié le 1 avril 2023. Consulté le [04/17/2024].
- [41] Le, N. L., Zhong, J., Negre, E., Abel, M.-H. : Système de recommandations basé sur les contraintes pour les simulations de gestion de crise. Rapport technique, Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of ComplexSystems), 20232, arXiv :2306.01504.Consulté le [04/17/2024].
- [42]Korem : Qu'est-ce qu'un point d'intérêt (POI) ?.Disponible sur :Korem.Consulté le [04/17/2024].
- [43] Outscrapr: Base de données des points d'intérêt (POI), publié par Yunus le 22 février 2022, disponible sur :Outscrapr.Consulté le [04/17/2024].
- [44] Braga, R. : LIDU : une approche basée sur la localisation pour l'identification de similarités d'intérêts entre utilisateurs dans les réseaux sociaux. Thèse de Doctorat, Université de Grenoble, 2012. Disponible sur :theses.hal.science.Consulté le [04/17/2024].
- [45] Gottapu, R.D., Monangi, L.V. : Point-of-interestrecommender system for social groups, Procedia Computer Science, Volume 114, 2017, Pages 159-164.Consulté le [04/17/2024].
- [46] Google Latitude Check-ins. Google System Blog. Disponible sur :(Google System Blog). Publié le 1 février 2011. Consulté le [04/17/2024].
- [47]Ye, M., Yin, P., Lee, W.-C., & Lee, D.-L. : Exploitinggeographical influence for collaborative point-of-interestrecommender system, dans les actes de la 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM, 2011, pages 325-334.Consulté le [04/17/2024].
- [48] Ngamsa-ard, S., Razavi, M., Prasad, P.W.C., Elchouemi, A.: Point-of-interest (POI) Recommender Systems for Social Groups in Location Based Social Networks (LBSNs) –

- Proposition of an Improved Model. *IAENG International Journal of Computer Science*, Vol. 47, No. 3, Septembre 2020, pp. 331-3421. Consulté le [04/17/2024].
- [49] Raous, A., Belhachani, N. : Titre du document non spécifié. Université KasdiMerbah Ouargla. Disponible sur : DSpace at KasdiMerbahUniversity Ouargla. Consulté le [04/17/2024].
- [50] W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, et X. Zhou, « SPORE: A sequential personalized spatial item recommender system », in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 2016, p. 954–965. Consulté le [04/17/2024].
- [51] Zhao, S., Chen, X., King, I., Lyu, M. R.: *Personalized Sequential Check-in Prediction: Beyond Geographical and Temporal Contexts*. Disponible sur : ResearchGate. Consulté le [04/17/2024].
- [53] Russell, S., Norvig, P. : *Artificial Intelligence : A Modern Approach*. Prentice Hall, 2020.
- [54] Wooldridge, M.: *An Introduction to MultiAgent Systems*. John Wiley& Sons, 2009.
- [55] Ferber, J. : *Dans Les Systèmes Multi-Agents : Vers une intelligence collective*. InterEditions, 1995.
- [56] Tranier, J. : *Vers une vision intégrale des systèmes multi-agents : Contribution à l'intégration des concepts d'agent, d'environnement, d'organisation et d'institution*. Thèse de doctorat, Université Montpellier II - Sciences et Techniques du Languedoc, 2007.
- [57] Georgeff, M. P., Pell, B., Pollack, M. E., Tambe, M., Wooldridge, M.: *The Belief-Desire-Intention Model of Agency*. Dans *Intelligent Agents V: Agent Theories, Architectures, and Languages*, ATAL 1998.
- [58] Brooks, R. A.: *A Robust Layered Control System for a Mobile Robot*. *IEEE Journal of Robotics and Automation*, RA-2(1), 1986, pp. 14-23.
- [59] Müller, J. P.: *The Design of Intelligent Agents: A Layered Approach*. *Lecture Notes in Computer Science*, vol 1177. Springer, Berlin, Heidelberg, 1996.
- [60] Dörner, D.: *The Logic of Failure: Recognizing and Avoiding Error in Complex Situations*. Metropolitan Books, 1996.
- [61] Simon, H. A.: *The Sciences of the Artificial*. MIT Press, 1996.
- [62] Weiss, G. (Ed.): *Multiagent Systems*. MIT Press, 2013.
- [63] Kambhampati, S.: *Data Management in Multi-Agent Systems*. *The Knowledge Engineering Review*, Vol. 16, No. 3, 2001, pp. 187-210.

- [64] Stone, P., Veloso, M.: Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, Vol. 8, No. 3, 2000, pp. 345-383.
- [65] Jiang, S., Fang, X.: Security and Privacy in Multi-Agent Systems. Proceedings of the 4th International Conference on Agents and Artificial Intelligence, 2012.
- [66] Sycara, K.: MultiagentSystems. *AI Magazine*, Vol. 19, No. 2, 1998, pp. 79-92.
- [67] Cavallaro, C., Verga, G., Tramontana, E., Muscato, O.: Multi-Agent Architecture for Point of Interest Detection and Recommendation, in Proceedings of the 21st Workshop from Objects to Agents (WOA), Bologna, Italy, September 14–16, 2020, CEUR Workshop Proceedings Vol. 2706, pp 237-246.
- [68] Mansour, A. M., Obeidat, M. A., Hawashin, B.: A Novel Multi Agent Recommender System for User Interests Extraction, in Springer Science+Business Media, LLC, part of Springer Nature, received: 18 January 2022, revised: 5 April 2022, accepted: 17 June 2022, Published online: 3 August 2022.
- [69] Mahmood, A. A., El-Bendary, N., Platoš, J., Hassanien, A. E., & Hefny, H. A.: An Intelligent Multi-Agent Recommender System, in Innovations in Bio-inspired Computing and Applications, Advances in Intelligent Systems and Computing 237, Springer, 2014, pp 201-213
- [70] Sebastia, L., Giret, A., García, I.: A Multi Agent Architecture for Tourism Recommendation, in Advances in Practical Applications of Agents and Multiagent Systems, Demazeau, Y.; Dignum, F.; Corchado, J.M.; Bajo, J. (Eds.), Advances in Intelligent and Soft Computing 70, Springer, January 2010, pp 421-430
- [71] Futura-Sciences. UML (Unified Modeling Language) : définition, news et bonnes pratiques, Futura-Sciences, Disponible sur: futura-sciences.com. Consulté le [04/17/2024].
- [72] Siloged : SysML - Diagramme de cas d'utilisation. Disponible sur SysML - Siloged. Consulté le [04/17/2024].
- [73] Mansouri, K. : UML Part 3 - Diagramme de Séquences. Disponible sur SlideShare. Consulté le [04/17/2024].
- [74] Audibert, L. : UML 2 - De l'apprentissage à la pratique, Diagramme de classes. Disponible sur Developpez.com. Consulté le [04/17/2024].
- [75] SparxSystems : UML 2 Tutoriel - Diagrammes de Composants. Disponible sur SparxSystems.
- [76] Lucidchart : Page de destination. Disponible sur Lucidchart. Consulté le [04/17/2024].

- [77] Creately, disponible sur :[Creately](https://creately.com) .Consulté le [04/17/2024].
- [78] Flutter disponible sur :ionos.fr.Consulté le [04/17/2024].
- [79] Flutter disponible sur:flutter.dev.Consulté le [04/17/2024].
- [80] Visuel Studio Code disponible sur : code.visualstudio.com. Consulté le [04/17/2024].
- [81] MongoDB disponible sur : MongoDB.com. Consulté le [04/17/2024].
- [82] Canva disponible sur : Canva.com. Consulté le [04/17/2024].
- [83] JSON disponible sur : json.org. Consulté le [04/17/2024].
- [84] Python disponible sur : python.org. Consulté le [04/17/2024].
- [85] JavaScript disponible sur : developer.mozilla.org. Consulté le [04/17/2024].
- [86] Dart disponible sur : dart.dev. Consulté le [04/17/2024].
- [87]Bhayani, A.: Quantifying set similarity using Jaccard similarity coefficient and MinHash. Disponible sur : arpitbhayani.me.Consulté le [04/17/2024].
- [88]Snowflake Inc. Estimation de la similarité de deux ensembles ou plus, Snowflake Documentation. Disponible sur : docs.snowflake.com.Consulté le [04/17/2024].
- [89] Tremblay, C. C., Charles, C. : Matrice de confusion pour le Machine Learning, Kobia. Disponible sur : kobia.fr.Consulté le [04/17/2024].