

MINISTERE DEL'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'ETUDE

Pour l'Obtention du Diplôme de Master en Informatique Option :

**Ingénierie des Systèmes d'Information**

Présenté par:

**MEKKAOUI MERIEM  
BOUDENE KELTOUM**

THEME:

**Vers un système de recommandation des POI basé  
sur les préférences spatiales et temporelles**

Soutenu le: 24 / 06 / 2023 .

Devant le jury composé de :

Mr Mohamed HABIB ZAHMANI	MCA	Université de Mostaganem	Président
Mme Kheira BELKHEIR	MAA	Université de Mostaganem	Examineur
Mme MAGHNI SANDID Zoulikha	MAA	Université de Mostaganem	Encadreur

Année Universitaire 2022-2023

## **Résumé :**

Ce projet se positionne dans le cadre du domaine des systèmes de recommandation (SR), c'est un système automatique qui permet, par des algorithmes automatiques, de fournir à des utilisateurs des suggestions qui répondent à leurs exigences. Parmi les techniques de recommandation les plus utilisées, le filtrage collaboratif. Ce mémoire consiste en l'étude, la réalisation d'un guide touristique sur la plateforme Android pour la ville Mostaganem. Ce qui permet aux touristes de découvrir les principales destinations touristiques, restaurants, cafétérias en temps réel à partir de sa position actuelle dans le but de faciliter à l'utilisateur l'accès aux différents lieux et la consommation et l'expérience du touriste. En effet, Nous avons développé une application qui permet de classer les différents lieux selon leurs distances et horaires. Donc l'information deviendra accessible à toute heure et à quel que soit le lieu où l'utilisateur sera localisé.

**Mot clés :** systèmes de recommandation, points d'intérêts, filtrage collaboratif, temporelle, spatiales

## **Abstract:**

This project is positioned in the field of recommender systems (SR), it is an automatic system which, through automatic algorithms, provides users with suggestions that meet their requirements. Among the recommendation techniques the most used is collaborative filtering. This thesis consists of the study, the realization of a tourist guide on the Android platform for the city Mostaganem. This allows tourists to discover the main tourist destinations, restaurants, cafeterias in real time from their current position in order to facilitate the user's access to different places and the consumption and experience of the tourist. Indeed, we have developed an application that allows you to classify the different places according to their distances and times. So the information will become accessible at any time and regardless of where the user is located.

**Keywords:** recommendation systems, points of interest, collaborative filtering, temporal, spatial

## ملخص:

يقع هذا المشروع في مجال أنظمة التوصية، وهو نظام آلي يوفر ، من خلال خوارزميات تلقائية للمستخدمين الاقتراحات التي تلبى متطلباتهم. من بين تقنيات التوصية الأكثر استخدامًا في التصنيف التعاونية تتكون هذه الأطروحة من الدراسة ، وتحقيق دليل سياحي على منصة اندرويد لمدينة مستغانم . يتيح ذلك للسائحين اكتشاف الوجهات السياحية الرئيسية والمطاعم والكافيتريات في الوقت الفعلي من موقعهم الحالي من أجل تسهيل وصول المستخدم إلى الأماكن المختلفة واستهلاك وخبرة السائح. في الواقع ، لقد قمنا بتطوير تطبيق يسمح لك بتصنيف الأماكن المختلفة وفقًا للمسافات والأوقات. لذلك ستصبح المعلومات متاحة في أي وقت وبغض النظر عن مكان تواجد المستخدم.

## كلمات مفتاحية :

أنظمة التوصية ، نقاط الاهتمام ، التصنيف التعاونية ، الزمنية ، المكاني

## **Dédicaces:**

Je dédie ce mémoire à :

J'ai le plaisir de dédier ce modeste travail ;

Au meilleur des pères,

Tu me dirigeais toujours vers le bon chemin,

Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi

**A** ma très chère maman

Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement, tu n'as pas cessé de m'encourager et de prier pour moi.

Que vous trouvez en moi la source de leur fierté, Qu'allah le tout puissant vous préserve, vous accorde santé et bonheur

**A** mon cher frère à qui je souhaite un avenir plein de réussite.

**A** mon grand-père que dieu les protègent, une longue vie inchallah

**Keltoum**

## Dédicaces:

Tout d'abord , je remercie le bon dieu qui me faciliter monchemin jusqu'à l'arrivée à réaliser ce modeste travail et qui sans lui je ne peux rien faire.

Tous les mots ne sauraient exprimer la gratitude, l'amour, Le respect, la reconnaissance, c'est tout simplement que je dédie ce mémoire.

A mes parents car ils m'ont inculque un esprit de persévéranceet qu'ils m'ont toujours pousse et motive dans mes études.

Sans eux, certainement je ne serais pas à ce niveau.Ils ont tout fait pour mon bonheur et ma réussite.Nulle dédicace ne puisse exprimer ce que je leur dois.

Que dieu leur réserve la bonne santé et une longue vie.

A mon frère, qu'ils restent pour moi des exemples depersévérance, de courage et de générosité.Et à tous ceux qui me sont chers et qui ont contribue à mesencouragements.

**meriem**

## **Remerciements**

Nous remercions tout d'abord **ALLAH** le tout puissant de nous avoir données la santé, la patience, la puissance et la volonté pour réaliser ce mémoire.

Nous adressons, notre profonde gratitude et tout notre amour à nos parents, nos frères, amis qui ont su nous faire confiance et nous soutenir en toutes circonstances.

Nous tenons particulièrement à remercier notre promoteur, Dr Maghni Sandid Zoulikha , Docteur à l'université de Mostaganem, département - Informatique - pour avoir accepté la charge d'être rapporteur de ce mémoire, nous le remercions pour sa disponibilité, ses pertinents conseils sa patience et pour les efforts qu'il a consenti durant la réalisation de ce mémoire. Qu'il trouve ici l'expression de notre reconnaissance et de notre respect.

Nos vifs remerciements vont également aux membres du jury d'avoir honoré notre soutenance de Master Merci pour votre présence.

**« Merci »**

**Keltoum & Meriem**

## Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Les étapes d'un système de recommandation	12
Figure 2	Un système de recommandation basé sur le contenu	15
Figure 3	Un système de recommandation collaboratif	16
Figure 4	Les méthodes de filtrage collaboratif	17
Figure 5	Matrice Utilisateur x Item14	18
Figure 6	Le système de recommandation hybride	20
Figure 8	Conception d'hybridation monolithique	21
Figure 9	Conception d'hybridation tubulaire	21
Figure 10	LSBN typique	22
Figure 11	un exemple de réseau social	25
Figure 12	les 4 facteurs influents sur LBSN	33
Figure 13	Modèle de distribution de la loi de puissance	35
Figure 14	Distribution d'enregistrement dans les multicentres	36
Figure 15	Distributions des lieux d'enregistrement personnels	36
Figure 16	L'importance de l'influence sociale sur la recommandation du POI	37
Figure 17	Règle de transformation des préférences de sentiment	38
Figure 18	Diagramme de cas d'utilisation	41
Figure 19	Diagramme de séquence « Inscription »	43
Figure 20	Diagramme de séquence « Inscription »	44
Figure 21	Diagramme de classe	45
Figure 22	Interface d'accueil	50
Figure 23	Interface d'inscription	51
Figure 24	Interface d'identification	52
Figure 25	Interface d'accueil (après l'inscription)	53
Figure 26	Interface de détail de POI	54
Figure 27	Interface firebase	54

## Liste des tableaux

Figure N°	Titre de tableau	Page
Tableau 1	Exemple de matrice utilisateur	14
Tableau 2	Tableau Utilisateur x Item	19
Tableau 3	Les avantages et les inconvénients des techniques de recommandation	23
Tableau 4	Aperçu de quelques techniques récentes de recommandation de points d'intérêt	40
Tableau 5	Description textuelle du cas d'utilisation "inscription"	42
Tableau 6	Description textuelle du cas d'utilisation "login"	43

# Sommaire

Introduction Générale .....	10
Chapitre 1 .....	11
Les systèmes de recommandation.....	11
1.1 Introduction .....	11
1.2 Définitions des systèmes de recommandation .....	11
1.3 Objectif des systèmes de recommandations ... ..	12
1.4 Les étapes principales de la recommandation .....	13
1.4.1 Collecte d'information .....	13
1.4.2 Mise en œuvre d'une matrice utilisateur d'informations récoltées .....	14
1.4.3 Extraction de la liste de recommandations .....	15
1.5 Les principales approches de filtrage .....	15
1.5.1 Filtrage basé sur le contenu (content based filtering) .....	15
1.5.2 Filtrage collaboratif (collaborative filtering) .....	16
1.5.2.1 Méthode basée sur la mémoire .....	18
1.5.2.2 Méthode basée sur un modèle .....	20
1.5.3 Filtrage hybride (hybrid filtering) .....	20
1.6 Avantages et inconvénients des systèmes de recommandations .....	23
Conclusion .....	24
Etat de l'art sur la recommandation des POI .....	25
2.1 Introduction .....	25
2.2 Définition d'un point d'intérêts (POI) .....	25
2.3 Système de recommandation des points d'intérêts (POI) .....	25
2.4 Définition de problème .....	26
2.5 Différents problématiques de recommandation de POI .....	27
2.6 Modèles hybrides de filtrage collaboratif .....	28
2.7 Graph Based Approaches .....	29
2.8 Modèles de factorisation matricielle .....	30
2.9 Modèles pour plateformes sociales .....	32
2.9.1 Liens sociaux et leurs poids .....	33
2.9.2 Types de systèmes de recommandation sociale .....	34
2.10. Les caractéristiques de point d'intérêts.....	34
2.10.1 Influence géographique .....	35
2.10.2 Influence Sociale .....	37



2.10.3. Influence Temporelle .....	37
2.10.4. Indication du contenu .....	37
2.11. Aperçu des modèles importants .....	38
2.12. Conclusion .....	39
Conception et modélisation du système .....	40
3.1. Introduction .....	40
3.2. Méthode d'Analyse et conception .....	40
3.2.1. Définition UML .....	40
3.2.2. Les diagrammes UML .....	40
3.2.2.1. Diagramme de cas d'utilisation .....	40
3.2.2.2. Diagramme de séquence .....	43
3.2.2.3. Diagramme de classe .....	44
3.3. Algorithme de recommandation utilisé .....	45
Conclusion .....	47
Implémentation .....	48
4.1. Introduction.....	48
4.2. Environnement de développement .....	48
4.2.1. visual paradigm online .....	48
4.2.2. Flutter .....	48
4.2.3. Firebase .....	48
4.3. Langages de programmation .....	49
4.3.1.Dart .....	49
4.3.Présentation de l'application .....	49
4.4. Conclusion .....	53
CONCLUSION GÉNÉRALE .....	54
Références.....	55

# Introduction générale

Dans notre vie, on est souvent confronté à des situations qui nous poussent à prendre des décisions qui sont basées sur nos connaissances, nos préférences et nos informations. Pour ces raisons nous sommes obligés de créer des nouveaux domaines de recherche informatique.

En effet le système de recommandation est un système qui peut analyser et prédire les préférences des utilisateurs et les aider dans leurs choix. Il est basé sur les algorithmes de recommandation qui sont utilisés sur des sites de commerce électronique tels que les produits proposées sur Google.

Parmi les approches appliquées dans les systèmes de recommandation, il y a les méthodes de filtrage collaboratif basées sur la notation des utilisateurs pour trouver les personnes les plus similaires.

Les notes sont dans une matrice « n » par « m », où « n » est le nombre d'utilisateurs et « m » est le nombre d'éléments. Plus précisément, le filtrage collaboratif consiste à remplir une matrice de notes afin de prédire les notes qu'un utilisateur donnera à un élément en fonction des notes que d'autres utilisateurs ont déjà fournies.

Dans notre projet, nous intéressons aux systèmes de recommandation de points d'intérêt (systèmes de recommandation de POI), qui sont basés sur le filtrage collaboratif, et l'application initiale de ces méthodes est le réseau social qui s'appuie sur la localisation (livres sterling).

L'objectif de notre travail est d'abord, d'étudier et proposer de nouvelles méthodes pour traiter le problème de la recommandation de POI qui est basée sur les données LBSN (social network based on location ).

C'est pourquoi nous considérons que l'amélioration en termes de «qualité» des recommandations doit tenir compte de la façon dont les utilisateurs explorent leur environnement et de leurs préférences. Ce qui fait que la qualité nous oriente vers la prédiction qui vise à génère un score de la recommandation liées aux préférences d'un utilisateur et vers le classement qui vise à optimiser les d'intérêt par celui-ci.

Puis notre travail s'intéresse à aborder le problème de scalabilité de telles méthodes pour réaliser notre travail, nous allons concevoir une application mobile qui intègre un système (POI) liées aux recommandations des lieux touristiques de la ville de Mostaganem.

Pour cela nous abordant 4 chapitres :

**Premier chapitre** : définition de systèmes de recommandation avec leurs différentes approches, avantages et inconvénients.

**Deuxième chapitre** : définition de Système de recommandation des points d'intérêts (POI), avec leurs problèmes ainsi les algorithmes utilisés dans ces systèmes.

**Troisième chapitre** : une vue conceptuelle de notre système, il contient le diagramme UML (Unified Modeling Language).

**Quatrième chapitre** : présentation des environnements de développement et les langages de programmation avec notre application mobile.

# Chapitre 1

## Les systèmes de recommandation

### 1.1. Introduction

Dans le premier chapitre, nous présenterons d'abord les principaux concepts de base de la recommandation. Nous commençons par définir les systèmes de recommandation et leurs types, puis introduisons trois principales méthodes de filtrage et la comparaison entre elles, puis nous abordons les problèmes de ces systèmes. Enfin, nous résumerons les systèmes de recommandation collaboratifs et l'avenir de ces systèmes.

### 1.2. Définitions des systèmes de recommandation

Les systèmes de recommandation peuvent être définis de plusieurs façons, vue la diversité des classifications proposées pour ces systèmes qui peuvent se rapporter à différents types de données ou approches spécifiques, mais il existe une définition générale de Robin Burke que nous utiliserons dans ce mémoire et qui les définit comme suit : « Des systèmes capable de fournir des recommandations personnalisées permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important ».

Les deux entités de base qui apparaissent dans tous les systèmes de recommandations sont l'utilisateur et l'item. L'« usager » est la personne qui utilise un système de recommandation, donne son opinion sur diverses items et reçoit les nouvelles recommandations du système.

L'« Item » est le terme général utilisé pour désigner ce que le système recommande aux usagers. Les données d'entrée pour un système de recommandation dépendent du type de l'algorithme de filtrage employé. Généralement, elles appartiennent à l'une des catégories suivantes :

- **Les estimations:**(également appelées les votes), expriment l'opinion des utilisateurs sur les articles (exemple : 1 mauvais à 5 excellent).
- **Les données démographiques :** se réfèrent à des informations telles que l'âge, le sexe, le pays et l'éducation des utilisateurs. Ce type de données est généralement

difficile à obtenir et est normalement collecté explicitement.

- **Les données de contenu** : qui sont fondées sur une analyse textuelle des documents liés aux éléments évalués par l'utilisateur. Les caractéristiques extraites de cette analyse sont utilisées comme entrées dans l'algorithme de filtrage afin d'en déduire un profil d'utilisateur.

### 1.3. Objectif des systèmes de recommandations

Les systèmes de recommandation visent à fournir aux utilisateurs des ressources liés à leurs préférences. Ce dernier voit donc son temps de recherche diminuer mais recevra également des suggestions du système, il n'aura pas faites involontairement attention. L'essor du Web et son omniprésence ont largement contribué à la mise en œuvre de tels systèmes dans le domaine du commerce électronique. Les recommandations peuvent initialement être considérées comme des réponses aux utilisateurs. Difficulté à prendre des décisions lors de l'utilisation du système. Recherche d'informations classique

### 1.4. Les étapes principales de la recommandation

Un système de recommandation requiert généralement trois (03) étapes, et qui sont illustré dans la figure suivante :



Figure 1 : Les étapes d'un système de recommandation

### 1.4.1. Collecte d'information

Pour être pertinent, un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. Une distinction peut être faite entre 2 formes de collecte de données :

- **Collecte de données explicites - Filtrage actif** : La collecte repose sur le fait que l'utilisateur indique explicitement au système ses intérêts. Exemple : Demander à un utilisateur de commenter, taguer/étiqueter, noter, liker ou encore ajouter comme favoris des contenus (objets, articles...) qui l'intéressent. On utilise souvent une échelle de ratings allant de 1 étoile (je n'aime pas du tout) à 5 étoiles (j'aime beaucoup) qui sont ensuite transformées en valeurs numériques afin de pouvoir être utilisées par les algorithmes de recommandation.
  - **Avantage** : Capacité à reconstruire l'historique d'un individu et capacité à éviter d'agrèger une information qui ne correspond pas à cet unique utilisateur (plusieurs personnes sur un même poste).
  - **Inconvénient** : Les informations recueillies peuvent contenir un biais dit de déclaration.
- **Collecte de données implicite - Filtrage passif** : Elle repose sur une observation et une analyse des comportements de l'utilisateur effectué de façon implicite dans l'application qui embarque le système de recommandation, le tout se fait en "arrière-plan" (en gros sans rien demander à l'utilisateur).

Exemple :

- Obtenir la liste des éléments que l'utilisateur a écoutés, regardés ou achetés en ligne;
- Analyser la fréquence de consultation d'un contenu par un utilisateur, le temps passé sur une page ;
- Monitore le comportement en ligne de l'utilisateur ;
- Analyser son réseau social ;
- **Avantage** : Aucune information n'est demandée aux utilisateurs, toutes les informations sont collectées automatiquement. Les données récupérées sont a priori justes et ne contiennent pas de biais de déclaration [1].

- **Inconvénient** : Les données récupérées sont plus difficilement attribuables à un utilisateur et peuvent donc contenir des biais d'attribution (utilisation commune d'un même compte par plusieurs utilisateurs). Un utilisateur peut ne pas aimer certains livres qu'il a achetés, ou il peut l'avoir acheté pour quelqu'un d'autre [1]

### 1.4.2. Mise en œuvre d'une matrice utilisateur d'informations récoltées

La mise en œuvre d'une matrice appelée " matrice utilisateur " ou bien " modèle utilisateur " incluant les informations concernant les utilisateurs récoltées durant la précédente étape qu'est la collecte d'informations. On peut la représenter comme un tableau qui contient des données recueillies sur l'utilisateur associées aux produits disponibles sur le site web.

	Item 1	Item 2	Item 3	Item 4	Item 5	...
User 1	😊			😊		...
User 2		😞	😊			...
User 3		😊				...
User 4			😞	😊	😞	...
...	...	...	...	...	...	...

**Tableau 1 : Exemple de matrice utilisateur [1]**

Le tableau ci-dessus présente un exemple fictif de matrice binaire contenant des informations de type " l'utilisateur u a apprécié/n'a pas apprécié l'item i ". Ces informations peuvent également être " a achète/n'a pas achète ", "a consulté/n'a pas consulté ", etc. Elles peuvent également se mesurer sur un nombre plus élevé de classes : " a mis 1/2/3/4/5 étoiles " etc. Un autre point important est comment le temps influence le profil de l'utilisateur. Les intérêts des utilisateurs, évoluent généralement au cours du temps. Les données du modèle utilisateurs devraient donc constamment être réajustées pour rester conformes aux nouveaux centres d'intérêts de l'utilisateur [1] [2].

### 1.4.3. Extraction de la liste de recommandations

Pour extraire une liste de suggestions c.-à-d. de recommandation à partir d'une matrice utilisateur, les algorithmes utilisent la notion de mesure de similarité entre objets ou personnes décrits par le modèle utilisateur. La similarité a pour but de donner une valeur ou un nombre (au sens mathématique du terme) à la ressemblance entre 2 choses. Plus la

ressemblance est forte, plus la valeur de la similarité sera grande. A l'inverse, plus la ressemblance est faible, et plus la valeur de la similarité sera petite [1][2].

## 1.5. Les principales approches de filtrage

Un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. Nous pouvons distinguer trois grandes classes d'approches selon la nature de ces données.

### 1.5.1. Filtrage basé sur le contenu (content based filtering)

Pour les recommandations basées sur le contenu [3], la tâche consiste à déterminer quels éléments du catalogue coïncident le mieux avec les préférences de l'utilisateur. Une telle approche ne requiert pas une grande communauté d'utilisateurs ou un gros historique d'utilisation du système. La figure 2 illustre ce processus :

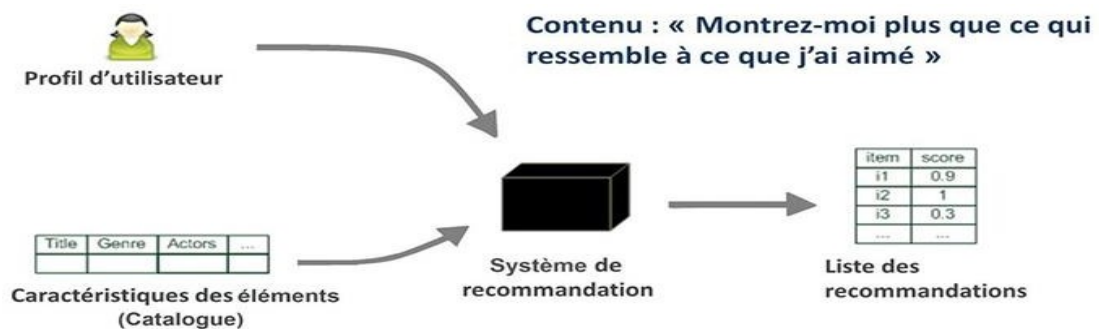


Figure 2 Un système de recommandation basé sur le contenu [4]

La manière la plus simple de décrire un catalogue d'éléments est d'avoir une liste explicite des caractéristiques de chaque élément (on parle aussi d'attributs, de profil d'élément, etc.). Pour un livre par exemple, on peut utiliser le genre, le nom des auteurs, l'éditeur ou toute autre information relative au livre, puis stocker ces caractéristiques (dans une base de données par exemple).

Le profil de l'utilisateur est exprimé sous forme d'une liste d'intérêts basée sur les mêmes caractéristiques. La coïncidence entre les caractéristiques des éléments et le profil de l'utilisateur peut être mesurée de différentes manières :

- l'indice de Dice ou d'autres mesures de similarité ;



- le TF-IDF (Term Frequency-Inverse Document Frequency) ;
- les techniques basées sur la similarité des espaces vectoriels (les approches bayésiennes [7], les arbres de décision, etc.) couplées avec des techniques statistiques, lorsqu'il y a trop de mots-clés.

### 1.5.2. Filtrage collaboratif (collaborative filtering)

Les systèmes basés sur le filtrage collaboratif [5] produisent des recommandations en calculant la similarité entre les préférences d'un utilisateur et celles d'autres utilisateurs. De tels systèmes ne tentent pas d'analyser ou de comprendre le contenu des éléments à recommander. La méthode consiste à faire des prévisions automatiques sur les intérêts d'un utilisateur en collectant des avis de nombreux utilisateurs. L'hypothèse sous-jacente de cette approche est que ceux qui ont aimé un élément spécifique dans le passé auront tendance à aimer cet élément spécifique, ou un autre très « proche », à nouveau dans l'avenir. La figure 3 illustre ce processus :

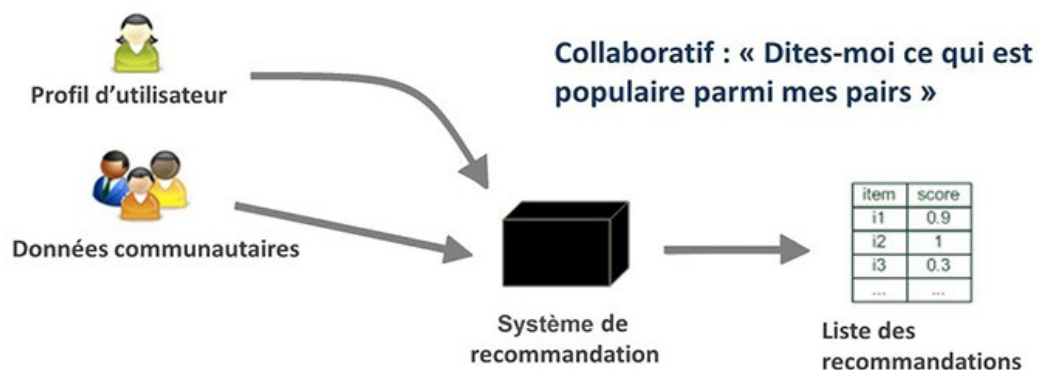


Figure 3 : Un système de recommandation collaboratif [4]

L'idée des approches collaboratives est d'essayer de prédire l'opinion d'un utilisateur sur les différents éléments. La recommandation est basée sur les goûts et avis précédents de l'utilisateur et sur une mesure de similarité avec d'autres utilisateurs. Les principales étapes de cette approche sont :

1. De nombreuses préférences d'utilisateurs sont enregistrées ;
2. Un sous-groupe d'utilisateurs est repéré dont les préférences sont similaires à celles de l'utilisateur qui cherche la recommandation ;
3. Une moyenne des préférences pour ce groupe est calculée ;
4. La fonction de préférence qui en résulte est utilisée pour recommander des

éléments à l'utilisateur qui cherche la recommandation.

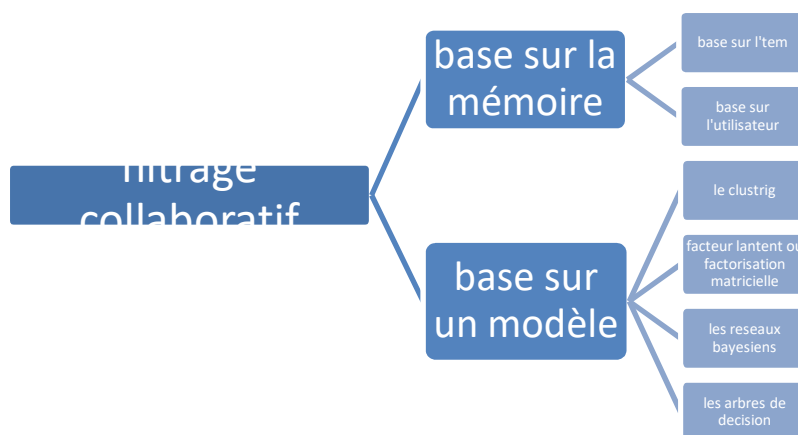
Il est possible de distinguer trois types d'approches pour définir la similitude ou la similarité :

- Les approches Item-to-Item basées sur la similarité entre les éléments (items). Notons que cette approche s'adapte à un nombre très important d'utilisateurs ou d'éléments.
- Les approches User-to-User basées sur la similarité entre les utilisateurs (users). Notons que cette approche n'est pas adaptée à un nombre très important d'utilisateurs.
- Et les autres approches.

Les systèmes de recommandation collaboratifs, par leur diversité, s'appuient donc sur des nombreuses techniques, qu'il s'agisse de :

1. similarité entre utilisateurs (coefficient de corrélation de Pearson [35], etc.) ou de sélection de voisinage (les algorithmes basés sur la recherche de voisinage) pour les approches User-to-Use ;
2. similarité entre éléments (la mesure de similarité cosinus [Qam10], etc.) pour les approches Item-to-Item ;
3. techniques de prédiction de scores (analyse en composantes principales ou ACP, factorisation de matrices, analyse sémantique latente, règles d'association, approches bayésiennes, etc.) pour les autres approches.

Le filtrage collaboratif est composé de deux méthodes



**Figure 4 : Les méthodes de filtrage collaboratif.**

### 1.5.2.1. Méthode basée sur la mémoire

Dans un système de filtrage collaboratif basé sur la mémoire, les données sont représentées sous forme d'une matrice "Utilisateur x Item" où les lignes représentent les utilisateurs  $U = \{u_1, \dots, u_m\}$  et les colonnes constituent les items  $I = \{i_1, \dots, i_j\}$ .

Les utilisateurs fournissent leurs opinions concernant les items sous forme de notes  $v$ . Pour un utilisateur actif  $U_a$  (par exemple Sarah) n'ayant pas exprimé son avis concernant un item  $i_k$  (le film "Les visiteurs"), le système recherche les utilisateurs voisins les plus proches notés  $U_a$  (parmi Imene, Hadi et Karim ayant noté le film "Les visiteurs" et qui ont déjà co-noté le film "Pulp Fiction" avec Sarah) et utilisent leurs opinions pour prédire la note manquante  $v(U_a, i_k)$  ( $v$ (Sarah, Les visiteurs)).

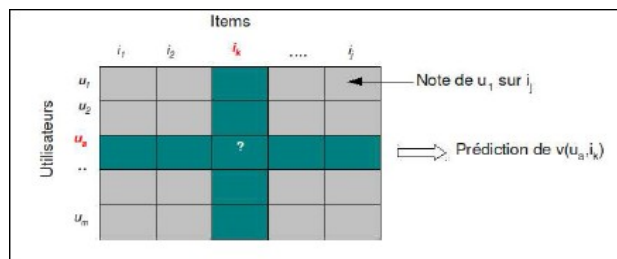


Figure 5 : Matrice Utilisateur x Item

	PulpF iction	Spiderman	Lesvis iteurs	Shrek
Sarah	1	5	?	3
Imene	4	2	4	?
Walid	3	?	?	5
Karim	4	?	5	?
Hadi	2	?	4	1

Tableau 2 : Tableau Utilisateur x Item

L'approche basée sur la mémoire exploite les appréciations des utilisateurs sur les items

(sous forme de notes par exemple), afin de générer les prédictions. Cette approche applique principalement des techniques statistiques dans le but d'identifier des utilisateurs voisins ayant, sur un même ensemble d'items, des appréciations similaires à celles de l'utilisateur actif.

Une fois les voisins identifiés, l'approche basée sur la mémoire utilise différents algorithmes afin de combiner les appréciations des voisins et générer des recommandations à l'utilisateur actif. Cette méthode utilise alors principalement les ratings (notations) (par ex : films). [3]

Le poids donné au rating de chaque utilisateur est déterminé par le degré de corrélation entre cet utilisateur et l'utilisateur pour qui nous désirons faire la recommandation. Les systèmes doivent généralement aussi gérer un grand nombre d'utilisateurs.

Faire des recommandations à partir des ratings de millions d'utilisateurs peut avoir de sérieuses implications en termes de performance. Ainsi, quand le nombre d'utilisateurs atteint un certain seuil, une sélection des "meilleurs" voisins doit être faite. Pour déterminer quels sont les voisins les plus pertinents à sélectionner et générer des recommandations fiables, nous utilisons généralement l'algorithme du k-nearestneighbor (k-NN) qui permet de sélectionner seulement les k meilleurs voisins ayant la plus haute valeur de corrélation.

Une autre approche (corrélation-thresholding) serait de sélectionner seulement les voisins possédant une corrélation plus grande qu'un certain seuil.

Nous pouvons distinguer deux méthodes de filtrage collaboratif basé sur la mémoire : la méthode basée sur la mémoire centrée sur l'item et la méthode basée sur la mémoire centrée sur l'utilisateur. [9]

### **1.5.2.2. Méthode basée sur un modèle**

Les méthodes basées sur un modèle ont été intégrées aux systèmes de recommandation pour améliorer et remédier aux problèmes des méthodes basées sur la mémoire.

Les algorithmes basés sur le modèle se basent aussi sur les évaluations précédentes (les profils) des usagers, mais cette méthode ne calcule pas directement les prédictions, elle classe les usagers suivant des groupes ou d'apprendre les modèles à partir de leurs données

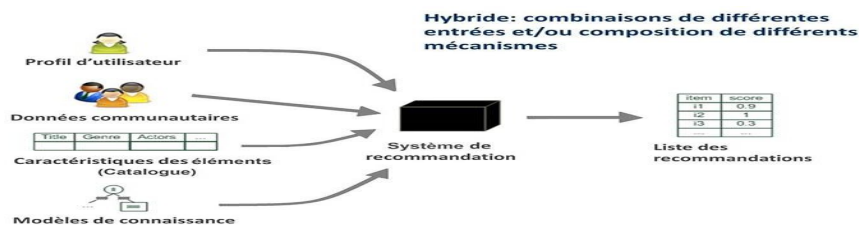
Pour la construction du modèle plusieurs méthodes sont utilisées. En général, les méthodes

basées sur le modèle utilisent les techniques d'apprentissage automatique, telles que le clustering, la factorisation matricielle, les réseaux bayésiens, les arbres de décision, etc. [9]

Dans notre travail, nous allons nous focaliser essentiellement sur la factorisation matricielle aussi dite décomposition matricielle. Elle consiste à décomposer une matrice en plusieurs autres matrices. Pour retrouver la matrice originale, il suffira de faire le produit de ces matrices entre elles.

### 1.5.3. Filtrage hybride (hybrid filtering)

Un système de recommandation hybride utilise des composants de différents types d'approches de recommandation ou s'appuie sur leur logique [6]. Par exemple, un tel système peut utiliser à la fois des connaissances extérieures et les caractéristiques des éléments, combinant ainsi des approches collaboratives et basées sur le contenu [4



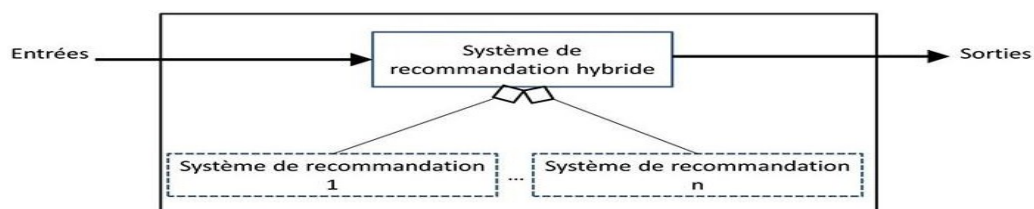
**Figure 6 Le système de recommandation hybride [4]**

Il est à noter que le terme « hybride » est un artefact de l'évolution historique des systèmes de recommandation où certaines sources de connaissances ont été exploitées en premier lieu, conduisant à des techniques bien établies qui ont ensuite été combinées. L'objectif est alors de s'appuyer sur des sources de connaissances multiples, en choisissant les plus appropriées à une tâche donnée afin de les utiliser le plus efficacement possible, la figure 6 illustre ce processus.

Il existe trois grandes catégories de combinaisons de systèmes de recommandation pour concevoir un système de recommandation hybride [6] [4] : la combinaison monolithique (monolithic hybridization design), la combinaison parallèle (parallelized hybridization design) et la combinaison tubulaire (pipelined hybridization design).

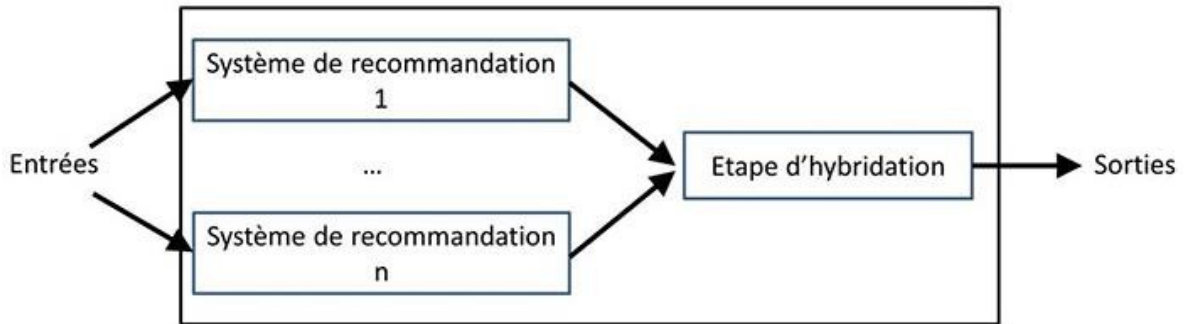
« Monolithique » décrit une conception d'hybridation qui intègre les aspects de différentes stratégies de recommandation en un seul algorithme. Comme illustré sur la figure 7 ci-dessous, différents systèmes de recommandation y contribuent puisque l'approche hybride utilise des données d'entrée additionnelles qui sont spécifiques à un autre algorithme de recommandation, ou bien les données d'entrée sont complétées par une technique et exploitées par une autre.

Par exemple, un système de recommandation basé sur le contenu qui exploite également des données communautaires pour déterminer



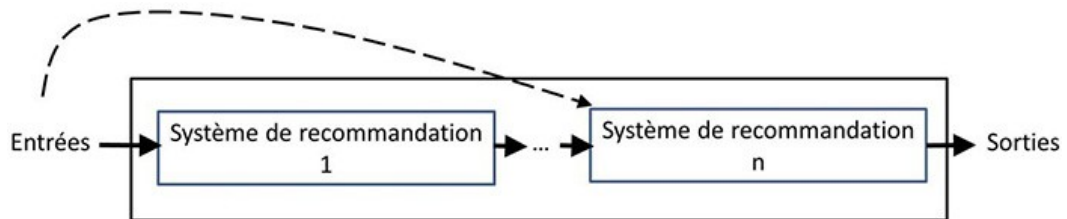
**Figure 7 : Conception d'hybridation monolithique [4]**

Les deux autres approches hybrides nécessitent au moins deux mises en œuvre de recommandations séparées qui sont combinées en conséquence. Sur la base de leurs données d'entrée, les systèmes hybrides de recommandation parallèles fonctionnent indépendamment l'un de l'autre et produisent des listes de recommandations distinctes, comme illustré sur la figure 8. Dans une étape ultérieure d'hybridation, leurs sorties sont combinées en un ensemble final de recommandations [6] [4].



**Figure 8 : Conception d'hybridation parallèle [4]**

Lorsque plusieurs systèmes de recommandation sont joints dans une architecture tubulaire, comme illustré par la figure 9, la sortie de l'un des systèmes de recommandation devient une partie des données d'entrée du système suivant [6] [4].



**Figure 9: Conception d'hybridation tubulaire [4]**

## **1.6. Avantages et inconvénients des systèmes derecommandations**

Malgré leur popularité croissante, les systèmes de recommandation ont subi quelques ratés. Nous identifions quelques problèmes dont ils souffrent [1] :

- **Adaptabilité** : Au fur et à mesure que la base de données des évaluations augmente, la recommandation devient plus précise ;
- **Nouvel utilisateur** : Un nouvel utilisateur qui n'a pas encore accumulé

suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes ;

- **Nouvel item** : Un item doit avoir suffisamment d'évaluations pour qu'il soit pris en considération dans le processus de recommandation ;
- **Démarrage à froid** : Souvent, on se retrouve confronté au problème qu'un utilisateur ne soit comparable avec aucun autre. Ce problème est dû au fait que peu ou pas d'utilisateurs ont évalué un article donné, ou qu'un utilisateur donné a évalué très peu ou pas d'articles. Généralement, ce problème survient quand un nouvel utilisateur ou une nouvelle ressource est ajouté à la base de recommandation ;
- **Le cas du système débutant** : provient lors du lancement d'un nouveau service de recommandation. Le système ne possède alors aucune information sur les utilisateurs et sur les items. Les méthodes de filtrage collaboratif ne peuvent pas fonctionner sur une matrice d'usages vide. La solution consiste en général à trouver des informations descriptives des items afin d'organiser le catalogue et inciter les utilisateurs à le parcourir jusqu'à ce que la matrice d'usages soit assez remplie et permette de passer en mode collaboratif.

Le tableau ci-dessous résume les avantages et inconvénients des méthodes traditionnelles utilisées par les systèmes de recommandation.

Approches	Avantages	Inconvénients
Filtrage à la base de contenu	<ul style="list-style-type: none"> <li>• Pas besoin d'une large communauté d'utilisateurs pour effectuer des recommandations.</li> <li>• Une liste de recommandation peut être générée même s'il n'y a qu'un seul utilisateur. La qualité croît avec le temps.</li> <li>• Pas besoin d'information sur les autres utilisateurs.</li> <li>• Prendre en considération les</li> </ul>	<ul style="list-style-type: none"> <li>• L'analyse du contenu est nécessaire pour faire une recommandation.</li> <li>• Problème de recommandation des images et de vidéos en absence de Métadonnées.</li> <li>• Nécessité du profil d'utilisateur.</li> </ul>



	gouts uniques des utilisateurs	
Filtrage collaboratif	<ul style="list-style-type: none"> <li>• Ne demande aucune connaissance sur le contenu de l'item ni sa sémantique.</li> <li>• La qualité de la recommandation peut être évaluée.</li> <li>• Plus le nombre d'utilisateurs est grand plus la recommandation est meilleure</li> </ul>	<ul style="list-style-type: none"> <li>• Démarrage à froid.</li> <li>• Nouvel Item.</li> <li>• Nouvel utilisateur.</li> <li>• Problème de confidentialité.</li> <li>• La complexité dans les systèmes avec un grand nombre d'items et d'utilisateurs le calcul croît linéairement.</li> </ul>

**Tableau 4: Les avantages et les inconvénients des techniques de recommandation**

## Conclusion

Ce chapitre avait pour objectif de faire un tour d'horizon, non exhaustif, sur les systèmes de recommandation. Nous avons évoqué leur historique, une définition théorique de ce qu'est un système de recommandation, Nous avons présenté les trois principaux types de systèmes de recommandation : ceux basés sur le contenu, ceux basés sur un filtrage collaboratif, et finalement les approches hybrides. Ensuite, nous avons présenté les avantages et les inconvénients de chaque type de système de recommandation. Aussi la manière d'évaluer ces systèmes.

Parmi les inconvénients majeur est le fait que ces systèmes restent limités car ils ne sont pas capables de s'adapter à leur environnement, autrement dit, ils ne prennent pas en compte le contexte (c'est-à-dire : l'ensemble des éléments qui peuvent influencer la compréhension d'une situation particulière).

## Chapitre 2

# Etat de l'art sur la recommandation des POI

### 2.1. Introduction :

De nos jours ,les systèmes recommandation sont devenus dominants dans la plupart des applications Web , ils peuvent être trouvés dans une grande variété d'industries, entreprises, service financier, musique / radio en ligne, tv et vidéos, les publications en ligne, et d'innombrables autres , le principal but de ces systèmes est de fournir des recommandations permettant de guider l'utilisateur vers des ressources intéressantes et utiles et augmenter ce chapitre porte : la définition des systèmes de recommandation des POI les méthodes et leurs algorithmes pour recommander les préférences temporelles ensuite spatiales ou bien les deux à la fois , les caractéristiques des POI, les modèles hybrides de filtrage collaboratif

### 2.2. Définition d'un point d'intérêts (POI) :

Un point d'intérêt (en Anglais Point Of Intrest) , est un point physique qui représente un lieu , ces point sont décrits par les cordonnées géographiques , il est peut être un hôtel ou bien une attraction touristique , certains logiciels de navigation et appareils GPS proposent des POI liées avec des informations enrichies comme : heures d'ouverture , numéro de téléphone , et même parfois liées avec des images comme le logo .

### 2.3. Système de recommandation des points d'intérêts (POI) :

La recommandation au point d'intérêt est une branche importante dans des systèmes de recommandation dans les réseaux sociaux basés sur la localisation (LBSN) comme Facebook , ce dernier collecte les informations d'enregistrement des utilisateurs, y compris les lieux géographiques visités , La figure 10 montre : un LBSN typique, présentant les interactions (p. ex., l'activité d'enregistrement) entre utilisateurs et points d'intérêt, et interactions (amitié) entre les utilisateurs. Afin d'améliorer l'expérience d'utilisateur dans les LBSN, la recommandation de point d'intérêt (POI) est proposée qui suggère de nouveaux endroits à visiter pour les utilisateurs à partir des enregistrements des

utilisateurs miniers et les relations sociales.

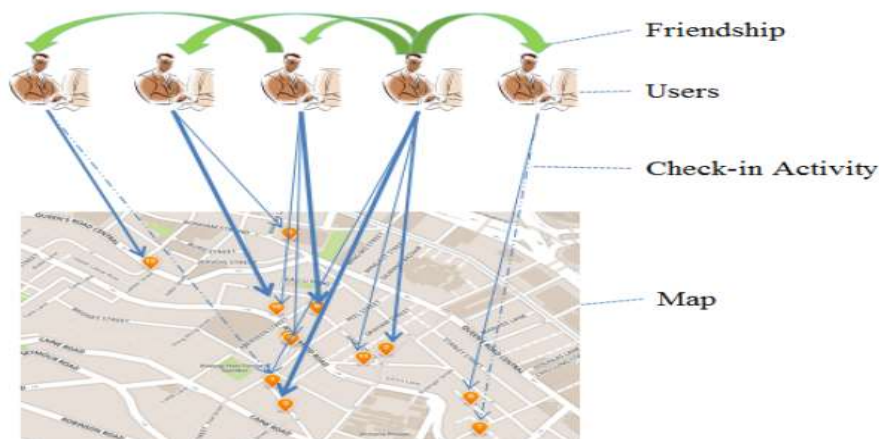


Figure 10 : LSBN typique

## 2.4. Définition de problème :

Chaque utilisateur  $u \in U$  est associé à un historique des lieux visités noté  $L_u$ . Nous utilisons ces ensembles  $L_u$  pour peupler la matrice  $X$  de vérification de l'utilisateur. Compte tenu de la matrice  $X$ , le problème de la recommandation de POI est de recommander pour chaque utilisateur  $u \in U$  une liste top- $k$  de nouveaux POI, c'est à dire des POI dans l'ensemble  $L \setminus L_u$ , qui ont le plus de chances de correspondre aux préférences de l'utilisateur, et donc d'être visités par  $u$ . Cette recommandation peut être faite par l'utilisateur. POI sans tenir compte de l'utilisateur, ou personnalisée si le résultat de la recommandation dépend de l'utilisateur. Résultat de la recommandation dépend de l'utilisateur. Contrairement aux défis traditionnels de la recommandation, la recommandation de POI s'accompagne d'autres défis spécifiques dus aux influences géographiques, temporelles et sociales.

- **L'influence géographique**

Selon la première loi de la géographie de Tobler [17], tout est lié à tout le reste, mais les choses proches sont plus liées que les choses éloignées. Cela signifie que la volonté de l'utilisateur d'enregistrer un POI est inversement proportionnelle à la distance qui le sépare de ce POI. En d'autres termes, plus le POI est éloigné, moins l'utilisateur aura tendance à le visiter. Ce phénomène, appelé phénomène de regroupement spatial (SCP) et a été largement exploité par le biais d'une assumption de loi de puissance dans la plupart des travaux existants. Dans la

plupart des travaux existants [18].

- **L'influence sociale.**

La plupart des travaux connexes ont établi que la plupart des amis ont un faible chevauchement de leurs check-in [18]. Cependant, le chevauchement est plus important que celui des non-amis, et il est donc toujours intéressant de l'exploiter.

- **Influence temporelle**

Les utilisateurs enregistrent généralement les restaurants à l'heure du déjeuner, tandis que les bars sont enregistrés vers minuit. Ainsi, différents utilisateurs peuvent se comporter de manière similaire ou différente en fonction du temps. On s'attend à ce que des POI réciproquement différents aient des heures d'ouverture différentes et une distribution non uniforme des enregistrements dans le temps. Ces deux informations ont été prises en compte par peu de travaux connexes, notamment [19].

## 2.5. Différents problématiques de recommandation de POI

Le problème de recommandation de centres d'intérêt décrit ci-dessus est le cas le plus général.

RS reçoit une requête  $Q(Lu)$  qui dépend uniquement de l'historique de l'utilisateur  $Lu$ . Cependant, il existe plusieurs sous-problèmes plus spécifiques qui exploitent généralement des informations secondaires pour effectuer une tâche similaire. Nous présentons ces tâches similaires dans la suite

- **Next POI Recommendation** : Dans ce cas, la demande de l'utilisateur dépend également de l'emplacement actuel de l'utilisateur :  $Q(Lu, lu_{actuelle})$ . L'objectif de ce problème est de faire des recommandations pour un emplacement donné et l'emplacement actuel de l'utilisateur. Cela C'est-à-dire que le système prendra en compte les séquences de visite [20]. Cependant, la plupart des travaux existants confrontés à ce problème exploitent des techniques et des méthodes utilisées dans la recommandation traditionnelle de POI.
- **Time-aware POI Recommendation** : Dans ce problème, la demande qui est reçue par le SR est  $Q(Lu, t_{courant})$ . Comme le problème précédent, ici la recommandation doit prendre en compte l'évolution des préférences de l'utilisateur dans le temps. Les auteurs de [21] proposent un cube user-time-POI pour modéliser

l'influence temporelle.

- **Recommandation d'itinéraires de POI** : De nombreuses approches ont été proposées pour recommander une liste de POI en fonction d'un budget en temps et/ou en argent. C'est ce que les auteurs de [22] ont étudié la question : ils ajoutent deux contraintes fortes au problème NP-hard de l'itinéraire optimal afin de proposer un système de recommandation. Deux contraintes fortes au problème NP-hard de l'itinéraire optimal afin de proposer une solution personnalisée. une solution personnalisée. Les auteurs de [23] proposent une approche de marche aléatoire pour maximiser l'expérience touristique des utilisateurs entre POI et la ville.

## 2.6. Modèles hybrides de filtrage collaboratif

Partant du constat que chaque famille de modèles a des avantages et des inconvénients, de nombreuses approches visent à combiner les avantages de méthodes distinctes, tout en minimisant leurs inconvénients. Ainsi, les modèles hybrides combinent plusieurs méthodes de recommandation. Dans cette partie, nous présentons brièvement quelques modèles hybrides célèbres.

- **iGSLR** : Geo-Social Location Recommendation : Ce modèle présenté dans [18] intègre les influences géographiques et sociales. L'influence sociale est calculée à l'aide d'une approche inspirée du filtrage collaboratif basé sur les amis proposé par [24] et par [25]. proposée par [24] et par [18]. Dans le iGSLR, la similarité sociale entre les utilisateurs  $u_i$  et  $u_j$  est calculée comme suit :

$$\text{SGSim}(U_i, u_j) = 1 - \frac{\text{distance}(U_i, u_j)}{\max_{u \in F(u_i)} \text{distance}(U_i, u_F)}$$

où  $F(u_i)$  correspond à l'ensemble des amis de l'utilisateur  $u_i$ . L'influence géographique est calculée à l'aide d'une estimation classique de la densité du noyau (kernel density estimation (KDE)) pour chaque utilisateur. l'historique des check-in tels que

$$\tilde{F}(d_{i,j}) = \frac{1}{|D|h} \sum_{d' \in \mathcal{L}^u} K\left(\frac{d_{i,j} - d'}{h}\right)$$

Sur la base de la distribution résultante, l'approche obtient alors une probabilité

qu'un utilisateur  $i$  visite un POI  $j$  en calculant les distances entre  $j$  et tous les POI visités par  $i$  comme suit :

$$p(l_j | \mathcal{L}^u) = \frac{1}{n} \sum_{i=1}^n \tilde{F}(d_{i,j})$$

Cette approche a deux limitations principales. Tout d'abord, KDE a besoin de savoir où se trouve le domicile de chaque utilisateur, alors que cette information n'est généralement pas affichée dans la plupart des LBSN. La solution à ce problème est de supposer l'emplacement du domicile en considérant les emplacements des enregistrements les plus fréquents. Cependant, cette hypothèse crée un biais important dans le modèle et, de plus, elle n'est pas pertinente dans le contexte des voyages à l'étranger. La deuxième limite de ce modèle est due à sa complexité.

En effet, le KDE nécessite de calculer la distance entre chaque paire de POI visités pour chaque utilisateur, ce qui est impossible pour les ensembles de données du monde réel.

## 2.7. Graph Based Approaches

Peu de travaux ont exploré les approches basées sur les graphes pour la recommandation de POI. Cependant, ces solutions sont intéressantes pour intégrer les influences géographiques et temporelles de manière naturelle. Habituellement, la principale limite de ces modèles est la quantité limitée d'informations secondaires qu'ils peuvent inclure. Dans cette partie, nous présentons GTAG, qui est représentatif de ces méthodes

- **GTAG: Geo-Temporal Aware Graph**

Cette approche [26] a proposé une approche basée sur un graphe tenant compte du temps pour intégrer les influences géographiques et temporelles. Le GTAG est inspiré du modèle STG proposé dans [27]. Dans ce modèle, il existe trois types de nœuds : Le nœud POI, le nœud session et le nœud utilisateur. Cette structure de graphe permet d'intégrer naturellement ces influences. Le modèle GTAG est basé sur quatre intuitions :

- les intérêts de l'utilisateur varient dans le temps, et son intérêt temporel à un moment donné est reflété par les POI qu'il a visités à ce moment-là,
- les check-in qui sont plus proches dans le temps de l'heure cible sont plus

importants,

- lorsque deux utilisateurs ont des modèles temporels similaires, ils ont tendance à visiter le même POI et
- les utilisateurs ont tendance à visiter des POI proches. Sur la base de ces intuitions, GTAG propose une approche de traversée de graphe qui exploite différents poids basés sur la largeur d'abord. différents poids basés sur une stratégie de propagation des préférences de type breadth-first. L'idée principale est d'injecter une préférence initiale pour le nœud d'utilisateur cible  $u$ , puis de propager cette valeur vers tous les POI candidats. La préférence propagée par chaque chemin  $p$  est la production de la préférence initiale  $r_u$  et des poids de toutes les arêtes du chemin comme suit

$$r_u^{(p)} = \prod_{\text{Edge}_{i,j} \in p} \omega_{i,j} \cdot r_u$$

## 2.8. Modèles de factorisation matriciel

L'objectif des méthodes de factorisation matricielle (MF) est de trouver une décomposition de la matrice  $X$  de vérification des utilisateurs avec 2 matrices : une matrice  $U \in \mathbb{R}^{M \times K}$  de la features latentes des utilisateurs et une matrice  $L \in \mathbb{R}^{(N \times K)}$  de caractéristiques latentes de POI où  $K \ll N$  est le nombre de facteurs latents. Généralement,  $K$  est défini entre 30 et 100. L'idée est de represent dans un "petit" espace caché (c'est-à-dire que  $K$  est petit par rapport à  $M$  et  $N$  :  $K \ll M, N$ ) les profils des utilisateurs et les descriptions des POI. Spécifiquement, chaque utilisateur  $i$  est représenté par une rangée  $u_i$  de  $U$  et chaque POI  $j$  est représenté par une rangée  $l_j$  de  $L$ . Ensuite, nous calculons le nombre de points d'intérêt.

$l_j$  de  $L$ . Ensuite, nous calculons le score de recommandation de l'utilisateur  $i$  pour le POI  $j$  avec le produit interne de leur produit latent correspondant. le produit interne de leurs vecteurs latents correspondants :  $\tilde{X}_{ij} = u_i l_j^T$ . Après une minimisation empirique du risque, nous obtenons le score de cal, nous obtenons la fonction objective la plus générale à minimiser :  $\min_{U,L} \|X - UL^T\|_F^2$ .

Cependant, cette fonction peut prendre en compte le problème d'overfitting problème de sur-ajustement en fonction de la catégorie du problème en ajoutant des termes de

régularisation comme suit :

$$\min_{U,L} \|X - UL^T\|_F^2 = +\lambda_1 \|U\|_F^2 + \lambda_2 \|L\|_F^2$$

où  $\lambda_1, \lambda_2 \in \mathbb{R}$  sont des scalaires de régularisation, et  $\|\cdot\|_F$  est la norme de Frobenius d'une matrice. Nous proposons dans la suite de cette section de présenter les modèles à base de MF les plus connus. Modèles à base de MF utilisés pour la recommandation de POI.

- **LRT: Recommandation d'emplacement avec effets temporels:**

Recommandation de localisation avec effets temporels (LRT) est un modèle de factorisation matricielle sensible au temps proposé récemment dans [19] pour traiter les influences temporelles des modèles de mobilité des utilisateurs dans les données LBSN. les influences temporelles des schémas de mobilité des utilisateurs dans les données LBSN. En effet, LRT propose (i) de modéliser la non-uniformité temporelle en définissant des créneaux temporels distinctes (ii) de modéliser la consécution temporelle en introduisant des termes de régularisation temporelle. Les auteurs observent que le comportement d'enregistrement des utilisateurs varie avec le temps, et Ils proposent de définir des vecteurs de caractéristiques latentes pour différentes tranches horaires de la journée.

Ensuite, tous les vecteurs latents sont exploités pour calculer le score final de recommandation. Par conséquent, une matrice de check-in  $X(t)$  est factorisée pour chaque tranche horaire  $t$  séparément, où  $t \in \{0, 1, \dots, 23\}$  correspond à une heure de la journée. Ensuite, des termes de régularisation sont ajoutés à la fonction objective comme suit :

$$\min_{U,L} \|X - UL^T\|_F^2 + \lambda_1 \|U\|_F^2 + \lambda_2 \|L\|_F^2 + \underbrace{\sum_{t=1}^T \sum_{i=1}^m \psi_i(t, t-1) \|u_i^{(t)} - u_i^{(t-1)}\|_F^2}_{\text{Temporal regularization terms.}}$$

où  $\psi_i(t, t-1)$  est la similarité entre  $X(t)_i$  et  $X^{(t-1)}_i$ . Ensuite, le score de recommandation est calculé comme la somme de tous les créneaux horaires :  $\tilde{x}_{i,j} = \sum_t U_i^t I_j^T$



- **GeoMF : Factorisation de matrices géographiques**

Les auteurs de [28] ont proposé une approche de factorisation matricielle géographique. GeoMF est basé sur WRMF [29] qui est une approche de factorisation matricielle régularisée qui donne des résultats de bonne qualité sur les images. Une approche de factorisation matricielle régularisée qui donne des résultats de bonne qualité sur des ensembles de données de rétroaction implicite. L'idée de GeoMF est de capturer le phénomène de regroupement spatial. Pour ce faire, GeoMF divise d'abord l'espace géographique entier en  $R$  grilles, chacune d'entre elles représentant une région géographique. Une région est un carré géographique de plus ou moins 500 mètres. Ensuite, GeoMF suppose que chaque POI propage son influence sur les régions environnantes. GeoMF modélise les régions d'activité des utilisateurs avec les vecteurs d'activité des utilisateurs et la région avec les vecteurs de propagation de l'influence. Les vecteurs latents géographiques des utilisateurs constituent la matrice  $G$  et les vecteurs de propagation des régions constituent la matrice  $Y$ . Le modèle GeoMF calcule ensuite le score de recommandation estimé comme suit :

$\tilde{X} = UL^T + GY^T$  GeoMF optimise finalement le problème suivant :

$$\min_{U,L,G} \left\| W \odot (X - UL^T - GY^T) \right\| + \gamma \left( \|U\|_F^2 + \|L\|_F^2 \right) + \lambda \|G\|_1$$

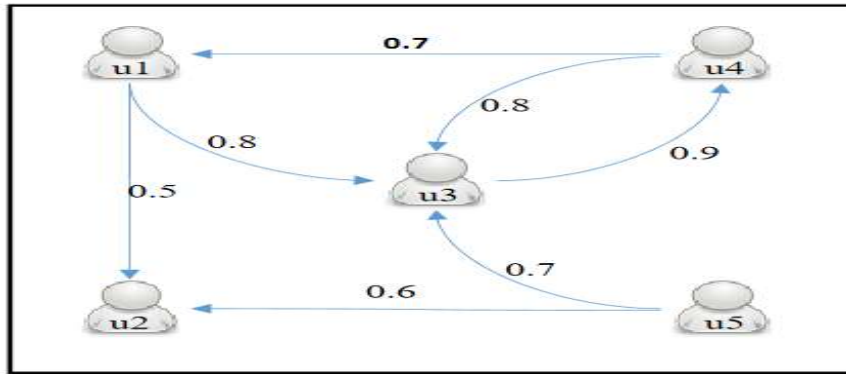
## 2.9. Modèles pour plateformes sociales

Avec l'arrivée des plateformes sociales sur internet, comme Facebook, Google plus, Twitter ou encore LinkedIn, les systèmes de recommandation ont pu ajouter une dimension à leur terrain de jeu : le réseau qui connecte les individus entre eux. Dans la vie on sait qu'il existe un lien entre les conseils de notre entourage et notre comportement, on peut être incité à acheter un produit ou à regarder un film, ou au contraire en être découragé à partir des goûts de nos proches. Cette dimension paraît donc évidente à prendre en compte afin de fournir des recommandations plus précises. En sociologie et en psychologie on connaît la tendance des individus à se connecter à des individus qui partagent des traits communs : CSP, âge, sexe, goûts etc... C'est le phénomène de l'homophilie [31]. Une recommandation de la part d'une personne fortement connectée dans le réseau aura peut-être ainsi plus de poids que celle d'une personne totalement étrangère. Cette notion peut

même se complexifier comme dans les travaux de Yang et al. [32], qui considèrent plusieurs cercles autour d'un utilisateur suivant la catégorie d'objets à recommander. Si un utilisateur peut faire confiance au goût d'un proche en matière de restaurant, il sera peut-être plus dubitatif sur ses conseils musicaux. A partir de chaque cercle d'amis liés à une catégorie, il est possible d'inférer la note manquante d'un utilisateur en prenant en compte ceux d'entre ses proches qui l'influencent dans cette même catégorie. Une autre méthode consiste à ajouter un terme lors de la prédiction d'une note, par exemple la somme pondérée des notes des amis de l'utilisateur cible comme dans [33] lors de la factorisation matricielle comme [36], [34],[35]. Si ajouter le réseau social comme biais de régularisation dans la recommandation tend en moyenne à baisser la précision des prédictions, la qualité des recommandations effectivement soumises à l'utilisateur semble augmenter paradoxalement [32]. De façon un peu similaire, les auteurs de [37] proposent d'utiliser une probabilité conditionnelle pour mesurer la similarité entre amis d'un réseau social, avec l'idée qu'un modèle probabiliste transcrit de façon plus réaliste les liens d'influence entre individus que la confiance. On peut donc construire un réseau Bayésien superposé au réseau des utilisateurs pour générer des recommandations.

### **2.9.1. Liens sociaux et leurs poids**

Selon Yang et al. (2014), les réseaux sociaux peuvent être généraux, tel Facebook ou spécifiques à un domaine, tel Netflix. Chaque utilisateur de ces réseaux a un ensemble de voisins directs qu'il suit ou en qui il a confiance. La relation sociale entre deux utilisateurs est une relation dirigée et pondérée (représentée par une valeur). Le poids social peut être interprété comme combien un utilisateur connaît ou fait confiance à un autre utilisateur dans un réseau social. Ce poids peut être basé sur un feedback explicite d'un utilisateur concernant un autre utilisateur (par exemple, l'évaluation d'un utilisateur par un autre), ou déduit d'un feedback implicite (par exemple, le degré d'interaction et de communication entre les deux utilisateurs). La figure 11 illustre un exemple de réseau social entre cinq utilisateurs où chacun d'entre eux a un groupe d'amis. Chaque lien d'amitié dirigé est pondéré par une valeur de confiance appartenant à l'intervalle  $[0, 1]$ .



**Figure 11 :un exemple de réseau social**

### 2.9.2. Types de systèmes de recommandation sociale

Bellogína et al. (2013) scindent les systèmes de recommandations sociales en quatre types que nous décrivons ci-dessous.

- **Recommandeur basé sur les amis**

Cette approche a été utilisée dans [30]. Elle incorpore l'information sociale dans le filtrage collaboratif basé sur les utilisateurs en utilisant la même formule de recommandation mais remplace l'ensemble des plus proches voisins de l'utilisateur actif par celui de ses amis.

- **Recommandeur basé sur la popularité sociale**

Cette approche, très simple, est basée sur le système de recommandation proposé par Barman et Dabeer (2010). Les systèmes de ce type recommandent les items les plus populaires chez les amis de l'utilisateur courant.

- **Recommandeur basé sur les distances**

L'approche de Ben Shimon et al. (2007) introduit explicitement les distances, entre les utilisateurs, dans le graphe social dans la formule du calcul de la recommandation. Cette approche utilise un algorithme qui calcule la distance entre deux nœuds dans un graphe, tel que l'algorithme de Dijkstra (1959)

### 2.10. Les caractéristiques de point d'intérêts :

Les préférences temporelles et spatiales résultent des : influences géographique, dynamiques temporelles, relations sociales et indications de contenu. La figure 12

démontre l'influence des ces 4 facteurs sur la recommandation des POI.

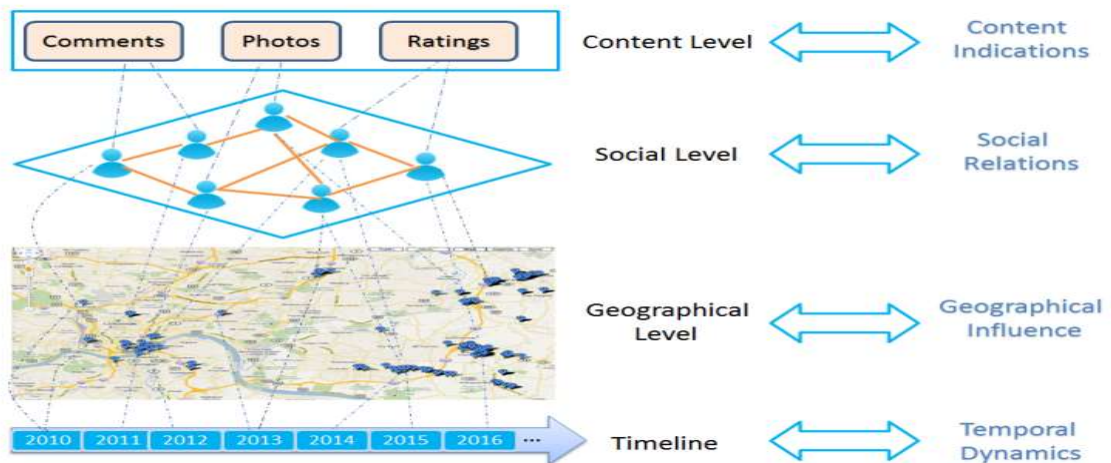


Figure 12 : les 4 facteurs influents sur LBSN

### 2.10.1. Influence géographique :

Est un facteur primordial car la recommandation dépend les caractéristiques géographiques, plusieurs études utilisent ce facteur pour améliorer les systèmes de recommandations des POI, En particulier trois modèles représentatifs à savoir: Modèle de distribution de la loi de puissance, modèle de distribution gaussien et densité du noyau, sont proposés pour saisir l'influence géographique dans la recommandation des POI.

- **Modèle de distribution de la loi de puissance :**

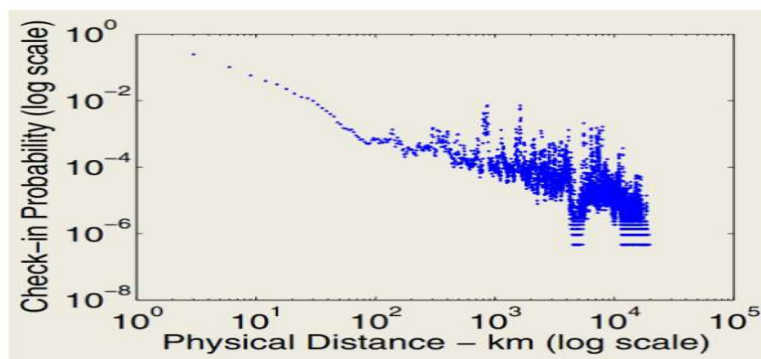
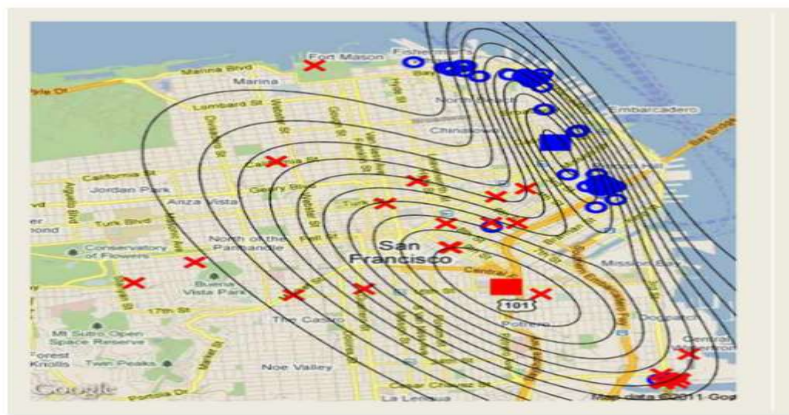


Figure 13 :Modèle de distribution de la loi de puissance

Selon la figure 13 on peut modéliser l'influence géographique comme suit :  $Y$  qui désigne la probabilité de cooccurrence de deux points d'intérêt par le même utilisateur tel que :  $y = a \cdot x^b$ , où le  $x$  est la distance entre deux POIs ensuite  $a$  et  $b$  sont les paramètres de distribution de la loi de puissance .

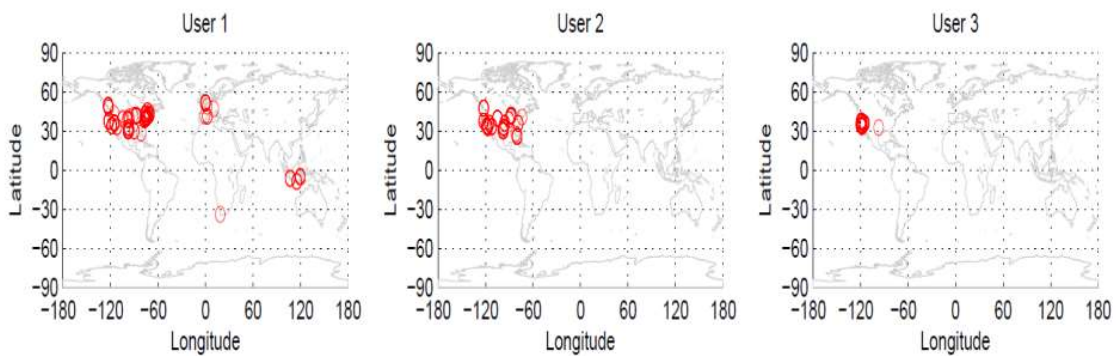
- **Modèle de distribution gaussien :**

Les chercheurs observent que les utilisateurs en LBSN régulièrement agir autour d'une activité comme le montre la figure 14 les chercheurs Cheng et al proposent un modèle gaussien multicentrique (MGM) pour saisir l'influence géographique pour la recommandation du POI.



**Figure 13 :Distribution d'enregistrement dans les multicentres**

- **Modèle KDE(Kernel Density Estimation) :**



**Figure 14 :Distributions des lieux d'enregistrement personnels**

D'après la figure 15 c'est difficile de modéliser différents utilisateurs en utilisant la même distribution, pour cela ils ont exploités l'estimation de la densité du noyau pour modéliser l'influence géographique en utilisant la distance de distribution de chaque utilisateur, KDE a deux étapes primordiales : Prélèvement d'échantillons à distance et estimation de la distribution des distances.

### 2.10.2. Influence Sociale :

L'influence sociale améliore les systèmes de recommandation mais d'une façon limitée comme on le montre dans la figure 16, on peut l'expliquer que les utilisateurs en LBSN se font des amis avec aucune limitation mais l'enregistrement oblige des interdépendances physiques entre les utilisateurs et POI.

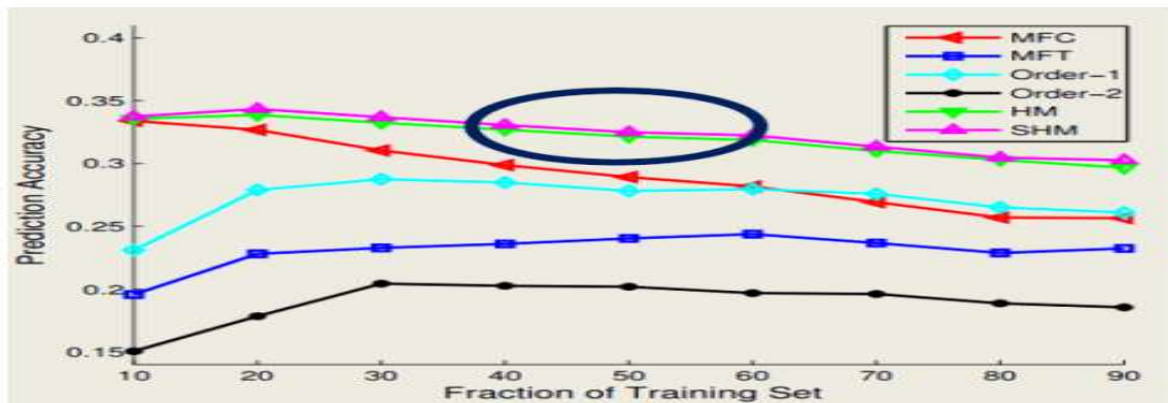


Figure 15 :L'importance de l'influence sociale sur la recommandation du POI

### 2.10.3. Influence Temporelle :

Est un facteur qui a une importance vitale pour la recommandation du POI car les contraintes physiques dans l'activité d'enregistrement résulte des modèles spécifiques, ce facteur a 3 aspects : périodicité, consécutive et non-uniformité.

### 2.10.4. Indication du contenu :

Les utilisateurs dans LBSN génèrent des évaluations qui sont convertis en matrice d'évaluation des éléments de l'utilisateur pour les POI, les notes sont des valeurs numériques donc la note la plus haute correspond à la meilleur satisfaction. Les chercheurs Yang et Al ont trouvés que les commentaires améliorent les systèmes de recommandation des POI, ainsi ils ont proposés une méthode de recommandation d'emplacement améliorée par sentiment, la figure 17 montre que : comment transformer les commentaires en des

phrases nominales comme : bad place...



Figure 16 : Règle de transformation des préférences de sentiment

## 2.11. Aperçu des modèles importants

Dans cette section, nous présentons brièvement certaines techniques récentes qui appartiennent à la famille des modèles qui ont inspiré nos recherches. Nous présentons d'abord une classification des méthodes existantes les plus récentes. Ensuite, nous décrivons la ligne conductrice des modèles présentés dans cette thèse.

- Modèles de cette thèse

Dans cette thèse, notre travail se concentre sur le problème le plus général de la recommandation de points d'intérêt.

Les approches que nous proposons dans la suite de cette thèse sont des méthodes basées sur la factorisation qui exploitent les influences géographiques et temporelles

Articles		Approche				Influence					
		Collab.Filt	Graph.-based	MAT?fact	Others	Geographique	Social	Temporal	Sequential	Categories	Textual
Lian et al 2014	GeoMF			✓		✓					
Gao,Tang,Hu,and Liu	LRT			✓				✓			

Liu and Xiong 2013	TL-PMF			✓		✓					✓
Cheng, Yang, King, and Lyu 2012	FMFMGM			✓		✓	✓				
Ye, Yin, Lee, and Lee 2011	USG	✓				✓					
Zhang and Chow 2013	iGSLR	✓				✓	✓			✓	
Liu, Fu, Yao, and Xiong 2013	GT-BNMF			✓		✓					
Li, Cong, et al. 2005	Rank- GeoFM			✓		✓		✓	✓		
Feng et al. 2015	PREM-G				✓	✓					
Zhang and Wang 2015	LTSCR		✓			✓	✓				
Ference, Ye, and Lee 2013	UPS-CF	✓				✓			✓	✓	
Zhang and Chow 2015	GeoSoCa	✓				✓	✓			✓	

**Tableau 5: Aperçu de quelques techniques récentes de recommandation de points d'intérêt.**

## 2.12. Conclusion :

L'objectif de ce chapitre est définir et citer les objectifs majeurs, caractéristiques des POI puis on a donné une définition des systèmes de recommandations des POI, ainsi expliquer brièvement les algorithmes et ces modèles en les illustrant à travers des exemples, finalement on a comparé les travaux par rapport aux problèmes, modèle et les influences.



# Chapitre 3

## Conception et modélisation du système

### 3.1. Introduction :

Dans ce chapitre nous allons présenter notre conception du projet en utilisant le langage de modélisation UML.

### 3.2. Méthode d'Analyse et conception :

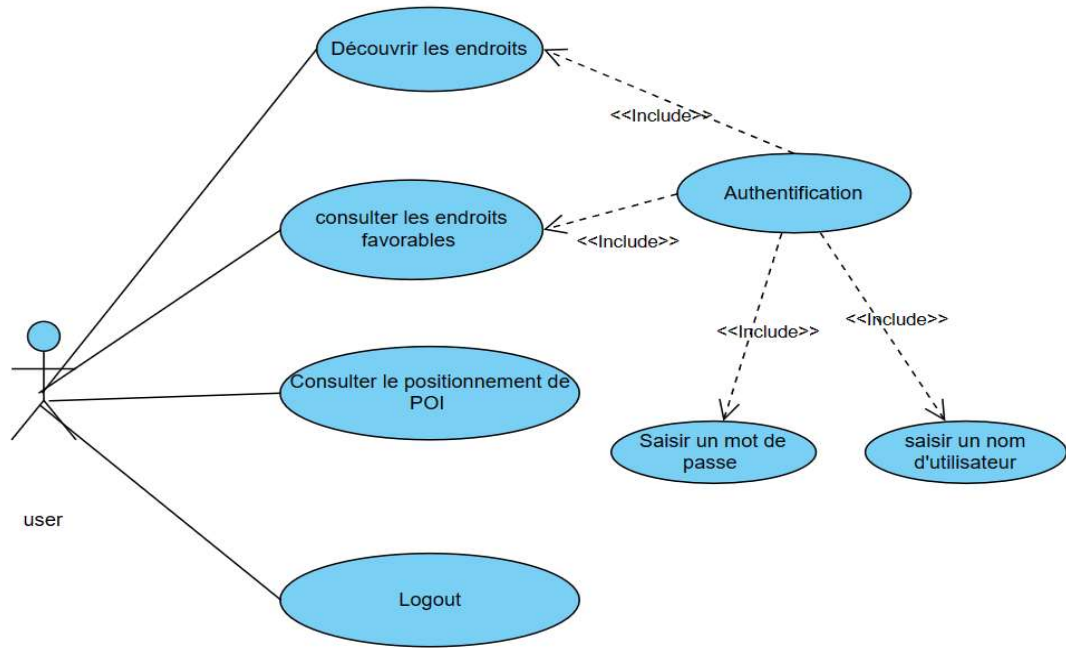
#### 3.2.1. Définition UML

UML, ou Unified Modeling Language (langage de modélisation unifié), est un langage standardisé utilisé pour représenter graphiquement la structure, le comportement et les interactions d'un système logiciel. Il fournit des diagrammes et des notations visuelles pour communiquer et documenter les différentes facettes d'un système logiciel, permettant ainsi de mieux comprendre, concevoir et développer des applications informatiques. UML est largement utilisé dans l'industrie du logiciel pour la modélisation, la conception et la communication entre les membres d'une équipe de développement..

#### 3.2.2. Les diagrammes UML

##### 3.2.2.1. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un type de diagramme utilisé dans UML pour représenter les interactions entre les utilisateurs (acteurs) et un système logiciel. Il permet de visualiser les fonctionnalités principales du système du point de vue des utilisateurs, en montrant comment ils interagissent avec le système pour atteindre leurs objectifs. Le diagramme met en évidence les acteurs, qui peuvent être des utilisateurs humains ou des entités externes, ainsi que les cas d'utilisation, qui décrivent les actions ou les scénarios spécifiques effectués par les acteurs. Les relations entre les acteurs et les cas d'utilisation sont représentées par des associations. Le diagramme de cas d'utilisation facilite la communication entre les membres de l'équipe de développement et les parties prenantes, en fournissant une vue claire des fonctionnalités attendues du système[38]. La Figure 18 montre le diagramme de cas d'utilisation de notre système



**Figure 17 : Diagramme de cas d'utilisation**

- **Description textuelle**

- Description textuelle du cas d'utilisation « Inscription »

Scénario	Inscription
Objectif	Crée un compte sur l'application
acteur	Utilisateur
Pré conditions	Disposition d'une connexion Interne
Post conditions	Création d'un compte
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le visiteur demande l'interface de création de compte</li> <li>2. Le système affiche l'interface de création de compte</li> <li>3. Le visiteur remplit les champs requis</li> <li>4. Le système vérifie les champs introduits par le visiteur</li> <li>5. Le visiteur accède à la page d'accueil</li> </ol>
	A1 : Champs obligatoires vides et/ou format incorrect/court <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> </ol>

Scénario alternatif	2. Le système réaffiche l'interface de création de compte
---------------------	---

**Tableau 3 ; Description textuelle du cas d'utilisation "inscription"**

- Description textuelle cas d'utilisation « Login »

Scénario	authentification
Objectif	Authentification dans l'application
acteur	Utilisateur
Pré conditions	Avoir un compte
Post conditions	Accès à son compte
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur accède au site</li> <li>2. Le système affiche l'interface d'authentification</li> <li>3. L'utilisateur introduit son email et son mot de passe</li> <li>4. Le système vérifie les champs introduit par l'utilisateur</li> <li>5. Le système vérifie l'existence de l'utilisateur</li> </ol> <p>Si les informations introduites sont correctes, le système affiche l'accueil</p>
Scénario alternatif	<p>A1 : Champs obligatoires vide et/ou non valide/ court</p> <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> <li>2. Le système reprend à l'étape 2 du scénarionominal</li> </ol> <p>A2 : Email et/ ou mot de passe incorrect</p> <ol style="list-style-type: none"> <li>3. Le système affiche un message d'erreur</li> <li>4. Le système réaffiche l'interface de login.</li> </ol>

**Tableau 4 ; Description textuelle du cas d'utilisation "login"**

### 3.2.2.2. Diagramme de séquence

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.

- Diagramme de séquence « Inscription »

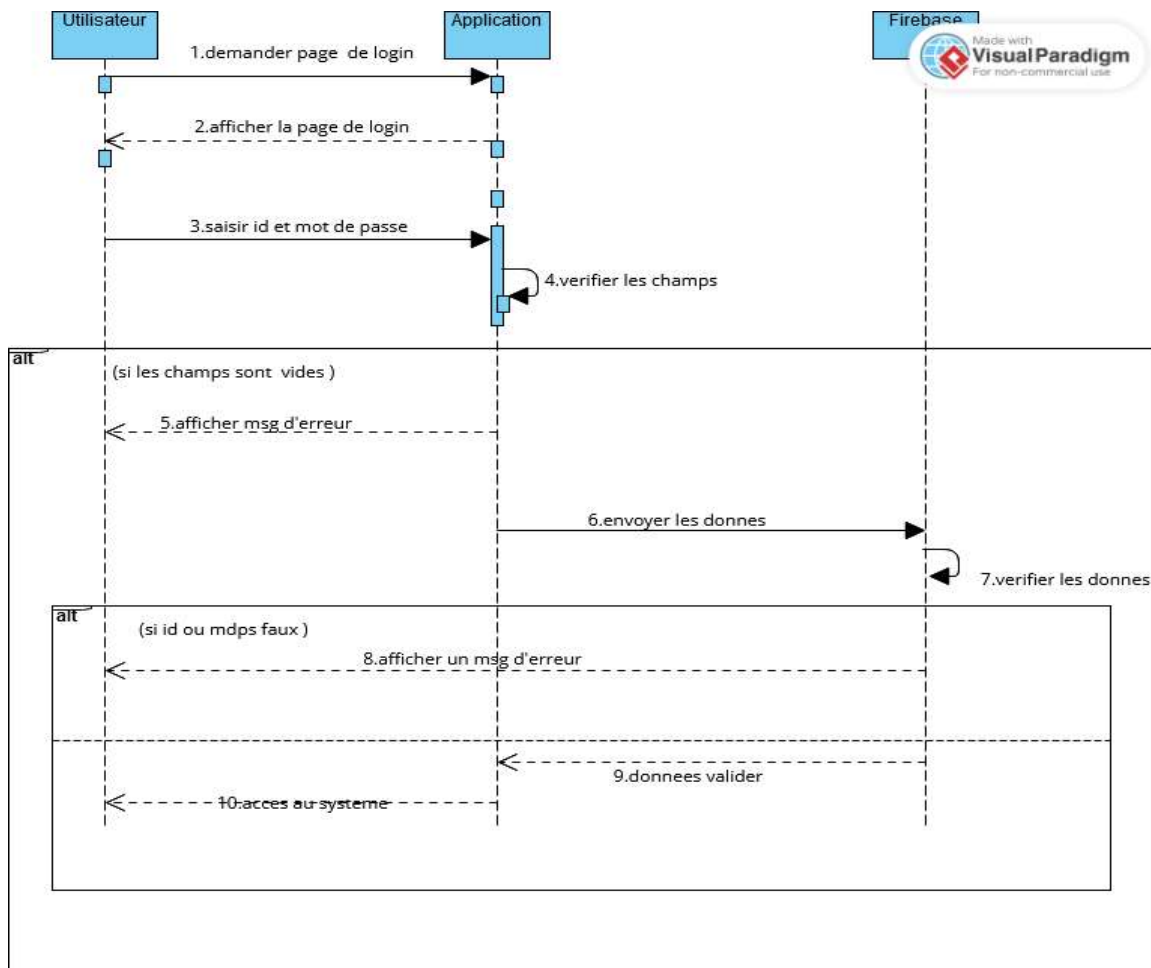


Figure 18 : Diagramme de séquence « Inscription »

- Diagramme de séquence « login »

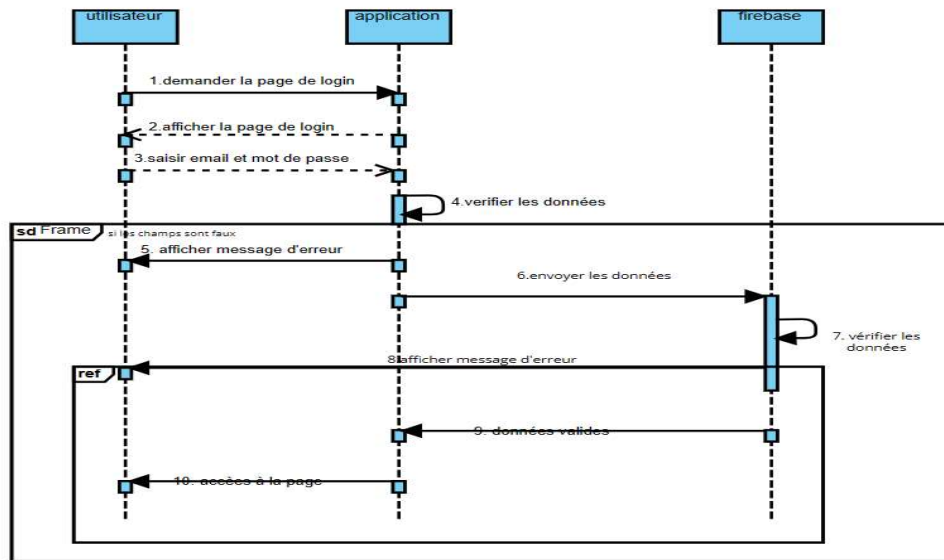


Figure 19 : Diagramme de séquence « login »

### 3.2.2.3. Diagramme de classe

Un diagramme de classe est un outil de modélisation utilisé dans UML pour représenter la structure statique d'un système logiciel. Il montre les classes, les interfaces, les attributs et les méthodes, ainsi que les relations entre ces éléments. Chaque classe est représentée par un rectangle avec son nom, et les attributs et les méthodes sont listés à l'intérieur. Les relations entre les classes sont illustrées par des lignes et des symboles spécifiques, tels que l'association, l'agrégation et l'héritage. Le diagramme de classe permet de visualiser la structure du système, de comprendre les relations entre les différentes classes et de faciliter la communication entre les membres de l'équipe de développement. Il est utilisé pour la conception, la documentation et la compréhension des systèmes logiciels.[38]

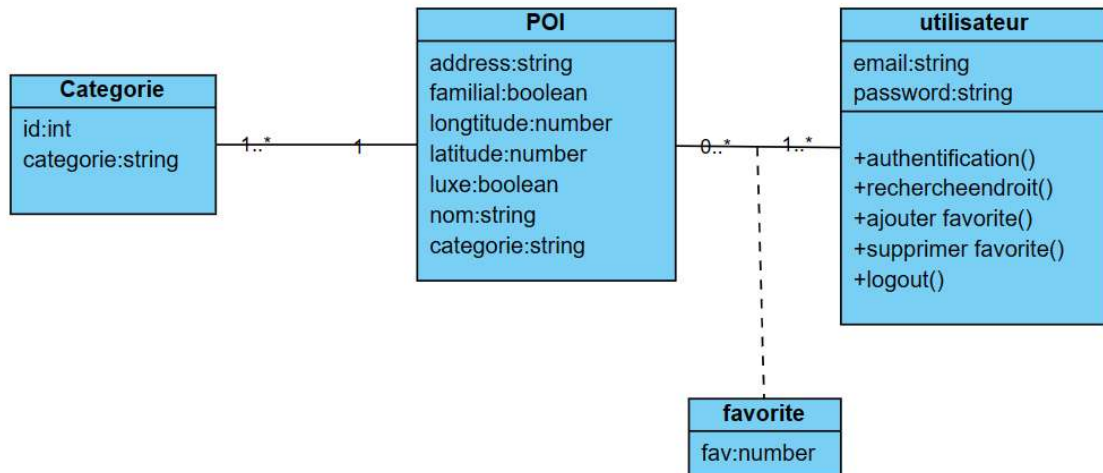


Figure 20 : Diagramme de classe

### 3.3. Algorithme de recommandation utilisé :

Nous avons opté pour Le filtrage collaboratif basé sur les utilisateurs a été introduit pour la première fois dans le système GroupLens [41] , Il s’agit de déterminer les utilisateurs qui sont similaires à l’utilisateur courant, ensuite calculer une valeur de prédiction pour chaque Item candidat à la recommandation en analysant les évaluations que les voisins de L’utilisateur courant ont exprimées sur cet item.

- **La similarité de cosinus :**

Cosinus est une mesure de similarité entre deux objets a et b de manière globale, très pratiquée en recherche d’informations [43] , qui consiste à représenter les deux objets par deux vecteurs  $V_u, V_{u'}$  et de mesurer le cosinus de l’angle formé par les deux vecteurs

$$\text{Sim}(\mu', u) = \cos(v_u, v_{\mu'}) = \frac{\sum_{i \in I_{u, \mu'}} r_{u,i} r_{\mu',i}}{\sqrt{\sum_{i \in I_{u, \mu'}} r_{u,i}^2}}$$

Où  $v_u$  et  $v_{\mu'}$  représentent les vecteurs des utilisateurs  $u$  et  $\mu'$  respectivement et il s’agit de :  $r_{u,i}$  et  $r_{\mu',i}$  respectivement  $I_{u, \mu'}$  consiste en l’ensemble des items notés à la fois par l’utilisateur  $u$  et l’utilisateur  $\mu'$

- **la corrélation de pearson :**

Ce coefficient est spécialement utilisé pour calculer la similarité entre deux utilisateurs u et v par les auteurs du système GroupLens [45]. Le coefficient de corrélation de Pearson mesure la relation entre la covariance et le produit de l'écart type des scores donnés par deux utilisateurs, de ce fait il utilise les items notés à la fois par u et v pour mesurer la similarité.

$$\text{Sim}(\mu', u) = \text{pearson}(\mu', u) \frac{\sum_{i \in I_{u, u'}} (r_{u,i} - \bar{r}_u)(r_{i, u'} - \bar{r}_{u'})}{\sqrt{\sum_{i \in I_{u, u'}} (r_{u,i} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_{u, u'}} (r_{i, u'} - \bar{r}_{u'})^2}}$$

Où  $\bar{r}_u$  et  $\bar{r}_{u'}$  sont les moyennes des notes données par les utilisateurs u et u' respectivement sur les items qu'ils ont notés

- **calcul de la prédiction :**

La prédiction p de note qu'un utilisateur u peut donner à un item i est donnée par la formule suivante :

$$P = \bar{r}_u + \frac{\sum_{k \in N(u) \cap U_i} \text{sim}(k, u) \cdot (r_{k,i} - \bar{r}_k)}{\sum_{k \in N(u) \cap U_i} |\text{sim}(k, u)|}$$

Où  $N(u)$  est l'ensemble des voisins de l'utilisateur u et  $U_i$  est l'ensemble des utilisateurs ayant notés l'item i.

### Exemple

Afin d'établir des recommandations, les algorithmes se baseront sur une matrice binaire du même type que ce que nous venons de présenter, à savoir avec une valeur de 1 lorsqu'un utilisateur a aimé le film et de 0 lorsqu'il ne l'a pas aimé. Dans cette matrice, les lignes représentent les utilisateurs et les colonnes correspondent aux films.

	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8
Utilisateur 1	1	1	0	0	1	0	0	1
Utilisateur 2	0	0	1	1	0	0	1	0
Utilisateur 3	1	1	0	1	0	0	1	0
Utilisateur 4	0	0	1	0	1	1	0	1

**Utilisateur 1 aime le film 2**

Nous utiliserons également la mesure de similarité cosinus afin de calculer les similarités entre toutes paires d'utilisateurs. Pour ce faire, la formule sera appliquée:

$$\text{Cosine sim}(u, v) = \frac{\sum_{i \in I_{u,v}} r_{u_i} r_{v_i}}{\sqrt{\sum_{i \in I_u} r_{u_i}^2} \sqrt{\sum_{i \in I_v} r_{v_i}^2}}$$

Il est possible de déterminer les similarités cosinus entre toutes paires d'utilisateurs. Ces similarités sont reprises

	Utilisateur 1	Utilisateur 2	Utilisateur 3	Utilisateur 4
Utilisateur 1	1	0,2236	0,4	0,4472
Utilisateur 2	0,2236	1	0,4472	0,25
Utilisateur 3	0,4	0,4472	1	0
Utilisateur 4	0,4472	0,25	0	1
Utilisateur 5	0,6	0,2236	0,4	0,4472

## Conclusion :

Dans ce chapitre, nous avons expliqué la conception du système qui repose à recommander un POI à un utilisateur donné, puis la modélisation des besoins en se basant sur le langage UML,



# Chapitre 4

## Implémentation

### 4.1. Introduction

Dans ce dernier chapitre nous présentons l'ensemble des outils ainsi que les environnements de programmation utilisées pour l'implémentation de notre application. Enfin nous présentons les divers interfaces de notre application.

### 4.2. Environnement de développement

#### 4.2.1. Visual Paradigm Online

Visual Paradigm Online est une plateforme de modélisation visuelle basée sur le cloud qui offre des fonctionnalités avancées de création de diagrammes et de modèles, ainsi que des options de collaboration en temps réel. C'est un outil pratique pour les équipes travaillant à distance ou souhaitant partager et collaborer sur des modèles et des diagrammes en ligne.[38]

#### 4.2.2. Flutter

Flutter est un Framework de développement d'applications multiplateforme, conçu par Google. Flutter met à disposition une grande variété de bibliothèques d'éléments d'IU standard pour Android et iOS. Le Flutter SDK se base sur le langage de programmation Dart également développé par Google. Il se veut le successeur moderne du langage JavaScript classique et, tout comme ce dernier, il s'exécute directement sous forme d'application mobile. Les programmes Dart peuvent aussi être exécutés directement sur un serveur. [39]

L'objectif est de permettre aux développeurs de fournir des applications hautes performances qui semblent naturelles sur différentes plates-formes. Flutter fonctionne avec le code existant, il est utilisé par les développeurs et les organisations du monde entier et est gratuit et open source.[40]

#### 4.2.3. Firebase

Firebase est une base de données No SQL hébergée dans le Cloud, c'est-à-dire qu'elle s'exécute sur un cloud et que l'accès à l'utilisateur est fourni en tant que service. Il stocke les données dans le format JSON (Javascript Object Notation), un format pour stocker ou

transporter des données. Tous les utilisateurs qui y sont connectés peuvent accéder aux données en temps réel. Firebase fournit plusieurs services [41] :

- **Firebase Authentication** : Il s'agit d'un service d'authentification qui permet aux développeurs de gérer l'authentification des utilisateurs de leur application à l'aide de méthodes telles que l'inscription par e-mail et mot de passe, l'authentification via des fournisseurs tiers (Google, Facebook, etc.), l'authentification anonyme, etc.
- **Cloud Firestore** : C'est une base de données NoSQL en temps réel et évolutive offerte par Firebase. Elle permet de stocker et de synchroniser les données en temps réel entre les clients de l'application. Firestore est souvent utilisé pour la gestion des données en temps réel, la création de chat en direct, la synchronisation des données hors ligne, etc.
- **Cloud Storage** : Il s'agit d'un service de stockage d'objets qui permet aux développeurs de stocker et de récupérer des fichiers tels que des images, des vidéos, des fichiers audio, etc. Il est généralement utilisé pour le stockage de fichiers statiques dans les applications web et mobiles.
- **Firebase Cloud Messaging (FCM)** : C'est un service de messagerie qui permet aux développeurs d'envoyer des notifications push à leurs utilisateurs sur les appareils Android, iOS et Web. Il facilite la communication en temps réel avec les utilisateurs de l'application.
- **Firebase Hosting** : Il s'agit d'un service d'hébergement web qui permet de déployer rapidement des applications web statiques. Firebase Hosting propose un déploiement simple et sécurisé, ainsi qu'un contenu mis en cache sur un réseau mondial de diffusion de contenu (CDN) pour des performances optimales.
- **Firebase Functions** : Il s'agit d'un service de calcul sans serveur qui permet d'exécuter du code côté serveur en réponse à des événements déclenchés par Firebase ou des appels HTTP. Les Firebase Functions permettent d'ajouter des fonctionnalités côté serveur à une application sans avoir à gérer une infrastructure serveur

## 4.3. Langages de programmation

### 4.3.1. Dart

Un langage open source, purement orienté objet, typé en option et un langage basé sur les classes qui offre un excellent support pour la programmation fonctionnelle et réactive. Contrairement à C# ou Java. En fait, c'est une langue relativement simple, moderne et très efficace pour travailler avec, Dart est assez facile à apprendre et il rend l'expérience de développement très fluide. [42]

## 4.4. Présentation de l'application

- **Interface de l'accueil**

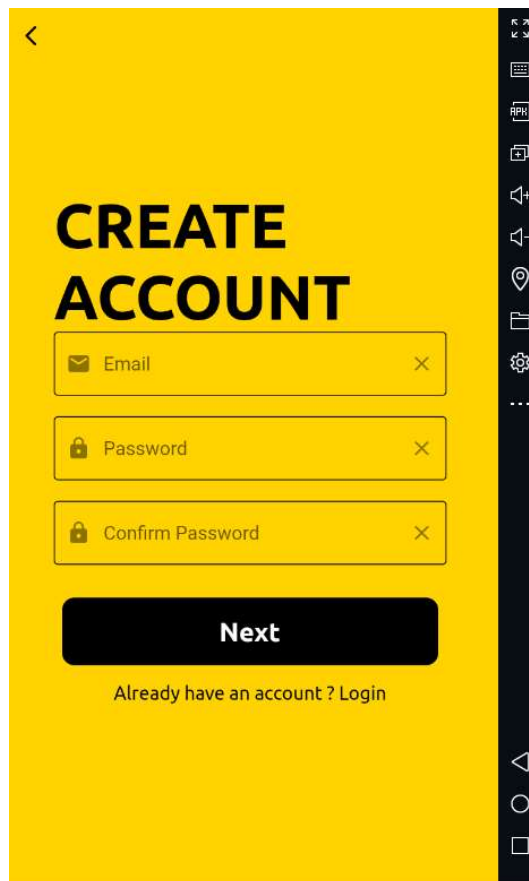
La figure 22 présente la page d'accueil, c'est la première page qui va être affichée au visiteur du site. Cette interface contient une section qui contient le logo du site, le bouton de commencer pour accéder à l'application



Figure 212 : Interface de l'application

- **Interface d'inscription**

Sur cette interface l'utilisateur doit remplir le formulaire d'inscription pour créer un compte (Email et mot de passe et la confirmation de mot de passe).

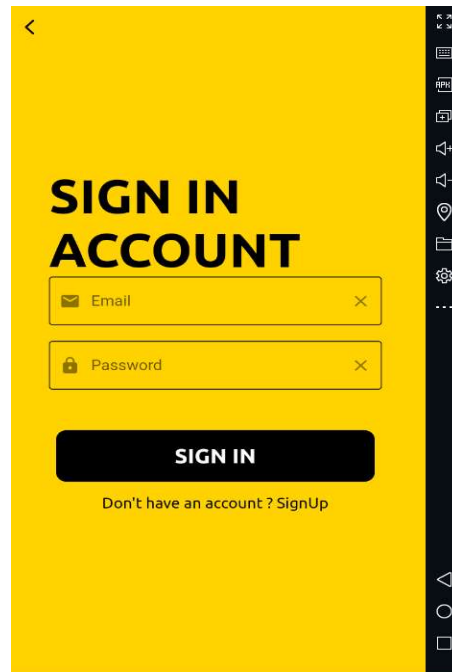


The image shows a mobile application interface for account creation. The background is a solid yellow color. At the top left, there is a white back arrow icon. The main heading 'CREATE ACCOUNT' is centered in a large, bold, black sans-serif font. Below the heading are three vertically stacked input fields, each with a white border and a small 'x' icon on the right side. The first field is labeled 'Email' and has an envelope icon on the left. The second field is labeled 'Password' and has a lock icon on the left. The third field is labeled 'Confirm Password' and has a lock icon on the left. Below these fields is a prominent black button with the word 'Next' in white, bold, sans-serif font. Underneath the button, the text 'Already have an account ? Login' is displayed in a smaller, black, sans-serif font. On the right side of the screen, there is a dark vertical navigation bar containing several white icons: a keyboard, a printer, a document, a speaker with a plus sign, a speaker with a minus sign, a magnifying glass, a folder, a gear, and a vertical ellipsis. At the bottom right corner, there are three white navigation icons: a square, a circle, and a triangle.

**Figure 223 : Interface d'inscription**

- **Interface d'identification**

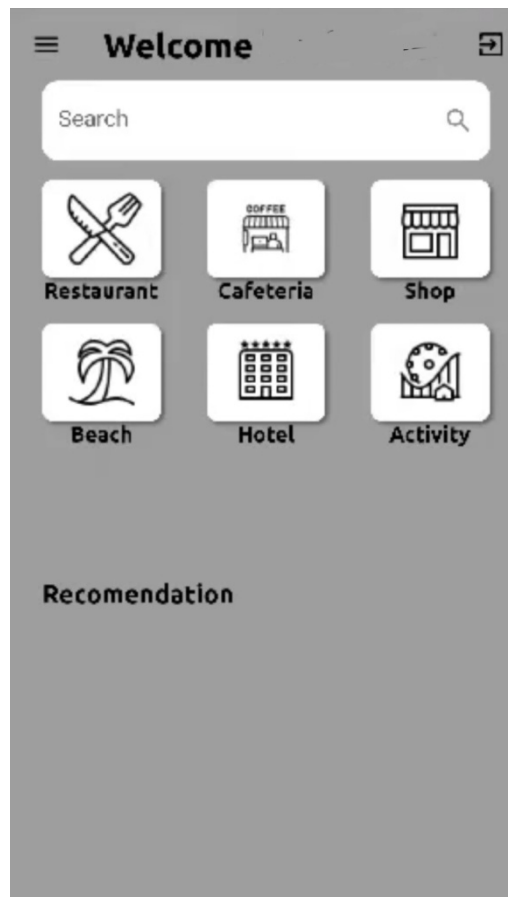
Cette interface permet au visiteur déjà inscrit dans application d'accéder à leur espace personnel.



**Figure 234 : Interface d'identification**

- **Interface d'accueil (après inscription)**

Après l'inscription, l'interface de l'accueil sera réaffichée mais avec plusieurs autres fonctionnalités. La section Just pour vous contient les POI (catégorie) les POI recommandé à l'utilisateur.



**Figure 245 : Interface d'accueil**

- **Interface de détail de POI**

Cette interface (figure 26) représente la page de détail de POI dans laquelle nous avons fournis les informations nécessaires pour l'utilisateur.

L'utilisateur peut ajouter le POI à ses favoris en cliquant sur le cœur en haut de l'image.

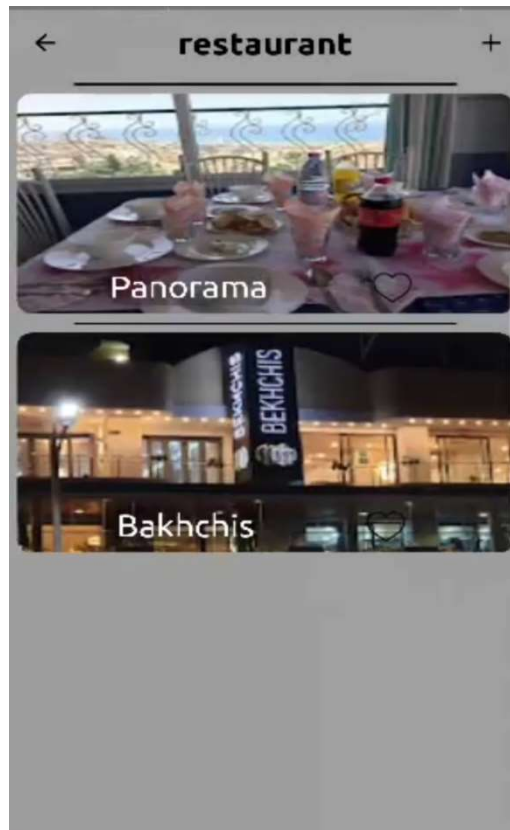


Figure 256 : Interface de détail de POI

- **Base de données (Firestore)**

Comme nous avons expliqué précédemment que nous avons utilisé Firestore comme base de données pour notre application, cette base de données en ligne va nous assurer que si un utilisateur change de dispositif, son compte sera encore sauvegardé dans la base de données, voici une capture d'une page de Firestore

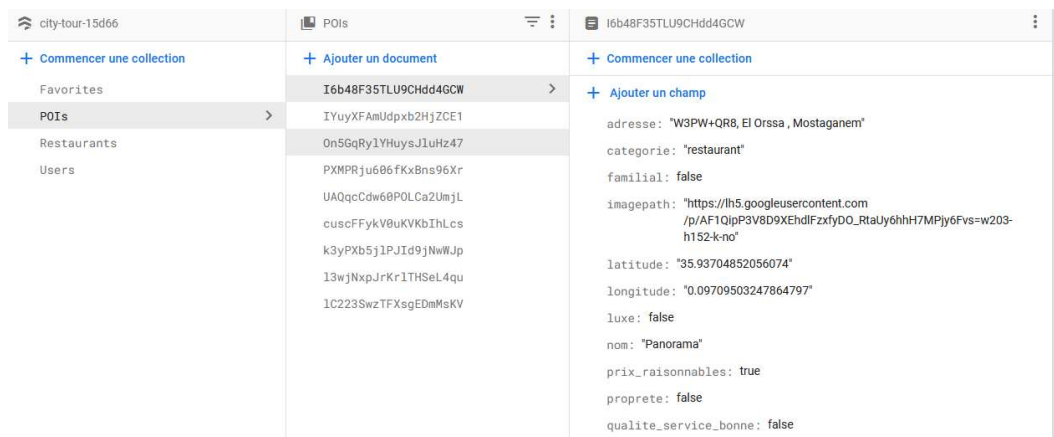


Figure 267 : Interface de firestore

## **4.5. Conclusion :**

Dans ce chapitre , nous avons présentés les différents logiciels et langages utilisés , les étapes d'implémentation et les différentes interfaces de notre système.

Le but principal de cette implémentation est d'évaluer notre proposition de système de recommandation de POI et de valider et justifier les différentes phases proposées afin d'améliorer la qualité de recommandation



## CONCLUSION GÉNÉRALE

Les Systèmes de Recommandation des poi sont des outils de recherche d'information et de filtrage qui aident les utilisateurs à explorer de nouvelles endroits mais encore ils aident les propriétaires des POI d'augmenter leurs revenus en attirant des utilisateurs.

Dans ce mémoire, nous sommes intéressés au domaine de la recommandation des poi en basant sur les préférences spatiales et temporelles des utilisateurs, ce travail est composé de 4 chapitres, nous avons vu dans le premier chapitre la définition du système de recommandation et nous avons détaillé types le filtrage, puis dans le deuxième chapitre nous avons parlé sur les points d'intérêt et ses caractéristiques, puis les algorithmes qui sont utilisés dans les systèmes de recommandations.

Le troisième chapitre est consacré pour la conception de notre système et finalement nous allons présenter les outils de l'implémentation et les interfaces de notre application.

Dans ce cadre, ce travail consiste sur le développement d'une application mobile de l'attractivité commerciale et touristique de la ville Mostaganem, qui repose sur le système de recommandation de POI.

En fin, ce mémoire nous a permis d'explorer de nouvelles méthodes et techniques dans ce domaine de recherche et de lire plusieurs articles intéressants. (fournit un système de recommandation basé sur l'algorithme de filtrage collaboratif sur les préférences spatiale et temporelles

## Références

- [1] Laouar, A. Korichi, K. Un système de recommandation pour l'assistance à la navigation sur Internet. Mémoire MASTER ACADEMIQUE Spécialité : Informatique industrielle. UNIVERSITE KASDI MERBAH OUARGLA. 2016.
- [2] Basu, C., Hirsh, H., Cohen, W. and Manning, N. C. Technical paper recommendation : A study in combining multiple information sources, 2001.
- [3] Margaritis, K. and Vozalis, E. Analysis of recommender systems' algorithms. In 6th Hellenic European Conference on Computer Mathematics its Applications (HERCMA), Athens, Greece, 2003.
- [4] Jannach, Zanker, Felfernig and Friedrich, Recommender Systems –2010
- [5] Bettini C., Brdiczka O., Henricksen K., Indulska J., Nicklas D., Ranganathan A., and Riboni D., A survey of context modelling and reasoning techniques. Journal of Pervasive and Mobile Computing, 2010.]
- [6] Burke, R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4) :331–370, 2002.
- [7] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325-341. Springer Verlag, Berlin, Heidelberg, 2007.
- [8] Jun Zeng Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing University, Chongqing, China
- [9] Yinghua Li Graduate School of Software Engineering, Chongqing University, Chongqing, China
- [10] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu (2012). “Fused Matrix Factorization with Geographical and Social Influence in Location-based Social Networks”.
- [11] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King (2013). “Where You Like to Go Next: Successive Point-of-interest Recommendation
- [12] Eunjoon Cho, Seth A. Myers, and Jure Leskovec (2011). “Friendship and Mobility: User Movement in Location-based Social Networks”.
- [13] Christian Desrosiers and George Karypis (2011). “A Comprehensive Survey of Neighborhood-based Recommendation Methods.
- [14] Hanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan (2015). “Personalized Ranking Metric Embedding for Next New POI Recommendation”.
- [15] Jean-Benoît Griesner, T.alel Abdesslem, and Hubert Naacke (2015). “POI Recommendation: Towards Fused Matrix Factorization with Geographical and Temporal Influences”.
- [16] Yehuda Koren, Robert Bell, and Chris Volinsky (2009). “Matrix Factorization Techniques for Recommender Systems”.

- [17]Harvey J. Miller (2004). “Tobler’s first law and spatial analysis”. English. In: Annals of the American Association of Geographers
- [18]Jia-Dong Zhang and Chi-Yin Chow (2013). “iGSLR: Personalized Geo-social Location Recommendation: A Kernel Density Estimation Approach”
- [19]WHuiji Gao, Jiliang Tang, Xia Hu, and Huan Liu (2013). “Exploring Temporal Effects for Location Recommendation on Location-based Social Networks”.
- [20]Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan (2015). “Personalized Ranking Metric Embedding for Next New POI Recommendation”
- [22]Chenyi Zhang, Hongwei Liang, Ke Wang, and Jianling Sun (2015). “Personalized Trip Recommendation with POI Availability and Uncertain Traveling Time”
- [23] Claudio Lucchese, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini (2012). “How Random Walks Can Help Tourism”
- [24]Hao Ma, Irwin King, and Michael R. Lyu (2009). “Learning to Recommend with Social Trust Ensemble
- [25]Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee (2011). “Exploiting Geographical Influence for Collaborative Point-of-interest Recommendation”
- [26]Quan Yuan, Gao Cong, and Aixin Sun (2014). “Graph-based Point-of-interest Recommendation with Geographical and Temporal Influences
- [27]Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun (2010). “Temporal Recommendation on Graphs via Long- and Short-term Preference Fusion
- [28]Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui (2014). “GeoMF: Joint Geographical Modeling and Matrix Factorization for Point-of-interest Recommendation”.
- [29] Yifan Hu, Yehuda Koren, and Chris Volinsky (2008). “Collaborative Filtering for Implicit Feedback Datasets

- [30]Liu, F., & Lee, H. J. (2010). Use of social network information to enhance collaborative filtering performance. *Expert systems with applications*
- [31]Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather : Homophily  
in social networks. *Annual review of sociology*, 27(1) :415–444, 2001
- [32]Xiwang Yang, Harald Steck, and Yong Liu. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1267–1275. ACM, 2012
- [33]Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.
- [34]Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec : social recommendation  
using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008
- [35]Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation  
for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010
- [36]Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems  
with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011
- [37]Xiwang Yang, Yang Guo, and Yong Liu. Bayesian-inference-based recommendation in  
online social networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(4) :  
642–651, 2013
- [38]Définition de Flutter disponible à l’adresse : <https://flutter.dev/docs>  
(consulté le 26/05/2021)
- [39]**  
Définition de Visuel Studio Code disponible à l’adresse : <https://code.visualstudio.com/docs>

[40]Définition de FireBase, disponible à

l'adresse :<https://www.geeksforgeeks.org/firebaserealtimedatabase-with-operations-in-android-with>

[41]10 good reasons to learn Dart. If you already know C++, C#, or Java... | by NafisFuad |

HackerNoon.com | Medium (consulté le 26/05/2021)