

Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

MEMOIRE DE FIN D'ÉTUDE

POUR L'OBTENTION DU DIPLOME DE MASTER EN
INFORMATIQUE

Option : Ingénierie des Systèmes d'Information

Présente par :

Khelifa Abdelkarim

Aimouche Abdessamad

THEME :

Vers un système de recommandation des POI basé
sur le contenu textuel et les catégories des POI

Soutenu le :25 /06/2023

Devant le jury composé de :

BESSAOUD KARIM	MCA	Université de Mostaganem	Président
DEDDOUCHE YAMINA	MAA	Université de Mostaganem	Examineur
MAGHNI SANDID ZOULIKHA	MAA	Université de Mostaganem	Encadreur

Année Universitaire 2022-2023

Résumé

Les systèmes de Points d'intérêt (point of interest) sont des algorithmes conçus pour fournir des recommandations personnalisées aux utilisateurs en fonction de leurs préférences, comportements et interactions passées. Ces systèmes utilisent des techniques d'apprentissage automatique pour analyser de grandes quantités de données qui peuvent varier considérablement entre contenu textuel ou catégorie des POI et doivent être prises en compte lors de la recommandation pour générer des suggestions. Notre système utilise des algorithmes et des techniques d'apprentissage automatique pour traiter des grands ensembles des données des traces d'utilisateurs telles que la catégorie de POI et le contenu textuel pour offrir une recommandation appropriée et pertinente. Nos résultats obtenus sont encourageants et nous pouvons conclure que notre système recommande avec succès des lieux qui satisfont les préférences des utilisateurs.

Mots-clés :

Système de recommandation SR, recommandation, apprentissage automatique, filtrage d'information, contenu textuel, catégories des POI.

Abstract

Point of interest systems are algorithms designed to provide personalized recommendations to users based on their preferences, behaviors, and past interactions. These systems use machine learning techniques to analyze large amounts of data which can vary greatly between textual content and category of POIs and must be considered when recommending and generating suggestions. Our system uses algorithms and machine learning techniques to process large sets of user trace data such as POI category and textual content to offer an appropriate and relevant recommendation. Our results are encouraging, and we can conclude that our system successfully recommends places that satisfy users' preferences.

Keywords:

Recommender system, recommendation, machine learning, information filtering, recommendation system, textual content, POI categories.

ملخص

أنظمة نقاط الاهتمام هي خوارزميات مصممة لتقديم توصيات مخصصة للمستخدمين بناءً على تفضيلاتهم وسلوكياتهم وتفاعلاتهم السابقة. تستخدم هذه الأنظمة تقنيات التعلم الآلي لتحليل كميات كبيرة من البيانات التي يمكن أن تختلف اختلافاً كبيراً بين المحتوى النصي أو فئة النقاط المهمة ويجب أن تؤخذ في الاعتبار عند التوصية بإنشاء اقتراحات . يستخدم نظامنا الخوارزميات وتقنيات التعلم الآلي لمعالجة مجموعات كبيرة من بيانات تتبع المستخدم مثل فئة POI والمحتوى النصي لتقديم توصية مناسبة وذات صلة. نتأجنا مشجعة ويمكننا أن نستنتج أن نظامنا يوصي بنجاح بالأماكن التي تلبى تفضيلات المستخدمين.

كلمات مفتاحية

نظام التوصية، التوصية، التعلم الآلي، تصفية المعلومات، نظام التوصية، المحتوى النصي.

Dédicaces

Au nom du dieu le miséricordieux louange à ALLAH le tout Puissant je dédie ce travail à mes chers parents Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

ABDELKARIM.

À MES CHERS PARENTS

Aucune dédicace ne saurait exprimer mon respect et ma considération pour vos sacrifices,

Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

ABDESSAMED.

Remerciements

Nous voudrions remercier et exprimer notre gratitude à notre encadrante de mémoire Mme MAGHNI SANDID ZOULIKHA, pour sa patience, sa disponibilité et conseils judicieux.

On remercie nos très chers parents pour leur soutien et nos familles pour leur encouragement inestimable.

Nos remerciements nos chers étudiants de la faculté des sciences exactes et informatique pour leur disponibilités et leur aide pour tester la qualité de notre travail.

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Approche basée sur contenu	5
Figure 2	Approche basée sur filtrage collaboratif	5
Figure 3	Approche hybride	6
Figure 4	Approche Basée sur utilisateur	7
Figure 5	Approche basée sur élément	9
Figure 6	Les différents types d'informations trouvées dans les LBSN.	15
Figure 7	Le fonctionnement de la LDA	17
Figure 14	Le fonctionnement de LSA	18
Figure 15	Méthode de classification d'opinion.	19
Figure 16	Exemple d'une table des tags d'un commentaire	23
Figure 17	Un exemple réel positif depuis la base de données	24
Figure 18	Un exemple réel négatif depuis la base de données	24
Figure 19	Fonctionnement de « i-Recommend »	26
Figure 20	Diagramme de cas d'utilisation de « i-Recommend »	27
Figure 21	Diagramme de classe de « i-Recommend »	30
Figure 22	Diagramme de séquence de « Log-in »	31
Figure 23	Diagramme de séquence de « Inscription »	32
Figure 24	Exemple d'une table des tags d'un commentaire	34

Figure 25	Exemple d'une table des tags d'un POI	35
Figure 26	Exemple d'une table des tags d'un POI utilisé dans les calculs de similarité	35
Figure 27	L'interface de l'espace Administrateur	39
Figure 28	L'interface de la base de données utilisée Firebase	40
Figure 29	Connexion au système	41
Figure 30	Formulaire d'inscription	42
Figure 31	La page d'accueil de l'application	43
Figure 32	La page de détails de la POI	44
Figure 33	La page des favoris	45
Figure 34	La section des commentaires	46

Liste d'abréviation

L'abréviation	Le mot
SR	Système de recommandation
POI	Point of interest (point d'intérêt)
SVD	Single value décomposition (Décomposition en Valeur singulière)
TF-IDF	Term Frequency-Inverse Document Frequency
LDA	L'allocation de Dirichlet latente
LSA	L'analyse sémantique latente
VADER	Valence Aware Dictionary and Sentiment Reasoner

Liste des Tableaux

Table N°	Titre de table	Page
Table 1	Comparaison entre le filtrage collaboratif et filtrage basé sur contenu	12
Table 2	Description textuelle du diagramme de cas d'utilisation	29
Table 3	Les données collectées d'après les utilisateurs qui ont testé l'application	47
Table 4	Un exemple d'évaluation d'un utilisateur qui a testé l'application	48

Liste des équations

Équation N°	Titre de table	Page
Équation 1	Similarité entre deux utilisateurs	8
Équation 2	Calcul de la note manquante	9
Équation 3	Similarité entre deux éléments	10
Équation 4	Calcul de la note d'un élément en utilisant la somme pondérée	10
Équation 5	Prédiction sur la correspondance entre les caractéristiques des utilisateurs et élément	11
Équation 6	La formule de calcul de l'indice de dice	12
Équation 7	La formule de calcul de comparative	26
Équation 8	La formule de calcul de la métrique de précision	49

Table des matières

INTRODUCTION GENERALE.....	3
CHAPITRE 1 : LES SYSTEME DE RECOMMANDATION	5
1.1 INTRODUCTION	5
1.2 LES TECHNIQUES DE RECOMMANDATION	5
1.2.1 <i>Les approches basées sur le contenu</i> :	5
1.2.2 <i>Les approches basées sur le filtrage collaboratif</i> :	6
1.2.3 <i>Les approches Hybrides</i> :	7
1.3 LES PRINCIPAUX ALGORITHMES DE RECOMMANDATION :	7
1.3.1 <i>Filtrage collaborative</i> :	7
1.3.2 <i>Approche basée sur le contenu</i> :	11
1.4 COMPARAISON ENTRE LE FILTRAGE COLLABORATIVE ET LE BASEES SUR LE CONTENU :	12
1.5 LES PROBLEMES DES SYSTEMES DE RECOMMANDATION :	13
1.5.1 <i>Le démarrage à froid</i> :	13
1.5.2 <i>Le manque de fraîcheur des données</i> :	13
1.5.3 <i>Le manque de diversité des données</i> :	14
1.6 CONCLUSION :	14
CHAPITRE 2 : LA RECOMMANDATION DES POIS.....	15
2.1 INTRODUCTION :	15
2.2 SOURCES D'INFORMATION ALTERNATIVES :	16
2.2.1 <i>Revue Textuelles</i> :	16
2.2.2 <i>Informations latérales riches sur les articles</i> :	20
2.3 SYSTEME DE RECOMMANDATION BASEE SUR LE CONTENU TEXTUELLE ET LES CATEGORIES :	21
2.4 CONCLUSION :	22
CHAPITRE 3 : CONCEPTION DE SYSTEME.....	23
3.1 INTRODUCTION :	23
3.2 METHODOLOGIE :	23
3.2.1 <i>Collection des données</i> :	23
3.2.2 <i>Analyse des sentiments</i> :	24
3.3 LE SYSTEME DE RECOMMANDATION :	26
3.4 LA CONCEPTION DE L'APPLICATION « I-RECOMMEND » :	28
3.4.1 <i>Diagramme de cas d'utilisation</i> :	28
3.4.2 <i>Diagramme de classe</i> :	30
3.4.3 <i>Diagramme de séquence</i> :	31
3.5 L'ALGORITHME DE RECOMMANDATION :	33
3.6 CONCLUSION :	37
CHAPITRE 4 : L'IMPLEMENTATION.....	38
4.1 INTRODUCTION :	38
4.2 ENVIRONNEMENT DE DEVELOPPEMENT ET LES LANGAGES UTILISEES :	38
4.2.1 <i>Dart</i> :	38
4.2.2 <i>Flutter</i> :	38
4.2.3 <i>Visual Studio Code</i> :	38
4.2.4 <i>Android studio</i> :	39
4.2.5 <i>FireBase</i> :	39

4.2.6	<i>Google Maps API</i> :.....	39
4.2.7	<i>Sentiment Dart</i> :.....	39
4.3	LES INTERFACES GRAPHIQUES DU SYSTEME :.....	39
4.3.1	<i>L'interface d'espace Administrateur</i> :.....	39
4.3.2	<i>La base de données (FireBase)</i> :.....	40
4.3.3	<i>L'interface d'espace Utilisateur</i> :.....	42
4.4	RESULTATS DES TESTS ET DISCUSSION :.....	48
4.5	CONCLUSION :	50
CONCLUSION GENERALE.....		51
BIBLIOGRAPHIE		52

Introduction Générale

Ces dernières années le développement technologique en particulier l'internet et les réseaux sociaux à encourager les chercheurs à explorer les systèmes de recommandation pour fournir des services personnalisés.

La diffusion générale d'internet en tant que plateforme commerciale a entraîné des changements fondamentaux dans les habitudes des consommateurs ce qui a généré une forte concurrence et une surabondance de l'information.

Pour faire ces défis, les systèmes de recommandation ont été développés comme un outil pour aider à la prise de décision en temps réel dans divers domaines tel que le commerce électronique le voyage et le divertissement.

Les systèmes de recommandation permettent de filtrer les informations et de fournir des recommandations personnalisées en fonction des préférences et des intérêts de chaque utilisateur. Les sources d'information varient, ce qui complique le développement d'algorithmes fiables et pertinents pour les utilisateurs. Les systèmes de recommandation de points d'intérêt POI basés sur les données de géolocalisation continuent de faire l'objet d'études approfondies pour résoudre les problèmes actuels et améliorer leur performance.

Ce projet consiste à développer une application mobile appelée "i-Recommend", qui recommande des lieux en fonction des préférences des utilisateurs, des commentaires et des catégories de points d'intérêt

Le présent mémoire est composé de quatre chapitres :

Premier chapitre : présente la définition d'un système de recommandation et les différentes approches utilisées avec leurs algorithmes les plus courants.

Deuxième chapitre : définit les systèmes de recommandation des points d'intérêt contextuels ainsi il présente un aperçu des différents algorithmes qui sont réalisés dans ce domaine.

Troisième chapitre : présente les différentes étapes de conception du système : collecte de données, la modélisation de système et le test de l'algorithme.

Quatrième chapitre : présente les détails de l'implémentation et la réalisation de l'application et les résultats obtenus.

Chapitre 1 : les système de recommandation

1.1 Introduction

Un système de recommandation pourrait utiliser diverses méthodes pour recommander des destinations ou des activités en fonction de préférences et d'intérêts. Certaines techniques courantes incluent le filtrage collaboratif qui se base sur le comportement et les préférences d'utilisateurs similaires, le filtrage basé sur le contenu qui recommande des lieux qui correspondent aux préférences et aux intérêts antérieurs de l'utilisateur. Et la recommandation hybride qui combine le filtrage collaboratif et le filtrage basé sur le contenu pour des recommandations améliorées.

Le choix de la méthode dépend des exigences et des objectifs spécifiques du système de recommandation.

1.2 Les techniques de recommandation

1.2.1 Les approches basées sur le contenu :

Le but de la tâche est de trouver les articles du catalogue qui correspondent le mieux aux goûts de l'utilisateur. Ce type d'approche ne nécessite pas un grand nombre d'utilisateur ni une importante quantité d'historique d'utilisation du système.

La figure suivante représente le principe des approches basées sur le contenu.

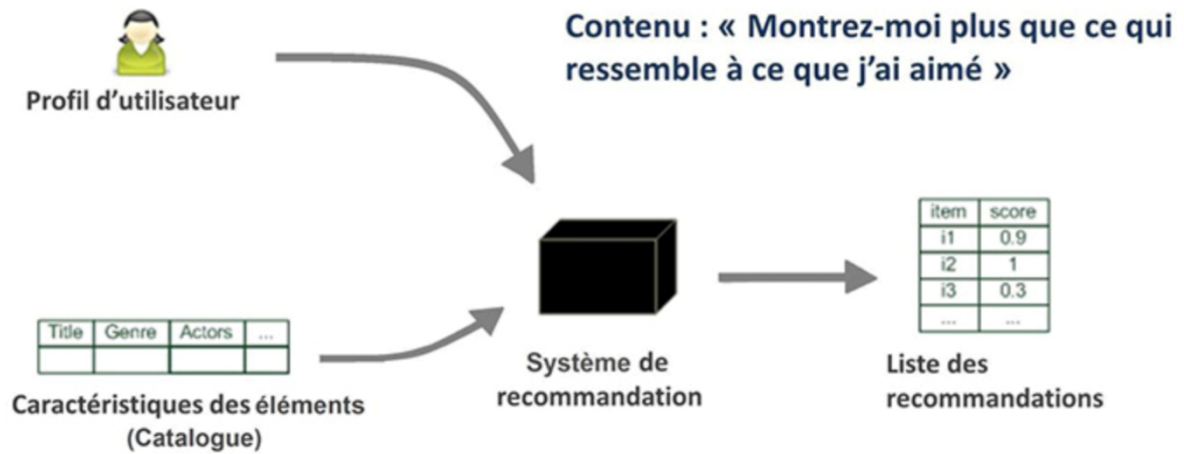


Figure 1 : Approche basé sur contenu [1]

1.2.2 Les approches basées sur le filtrage collaboratif :

Cette technique prévoit les préférences d'un utilisateur en recueillant les opinions de nombreux utilisateurs. Le principe sous-jacent de cette approche est que les personnes qui ont apprécié un article en particulier dans le passé seront plus susceptibles de l'apprécier à nouveau, ou un article similaire, à l'avenir.

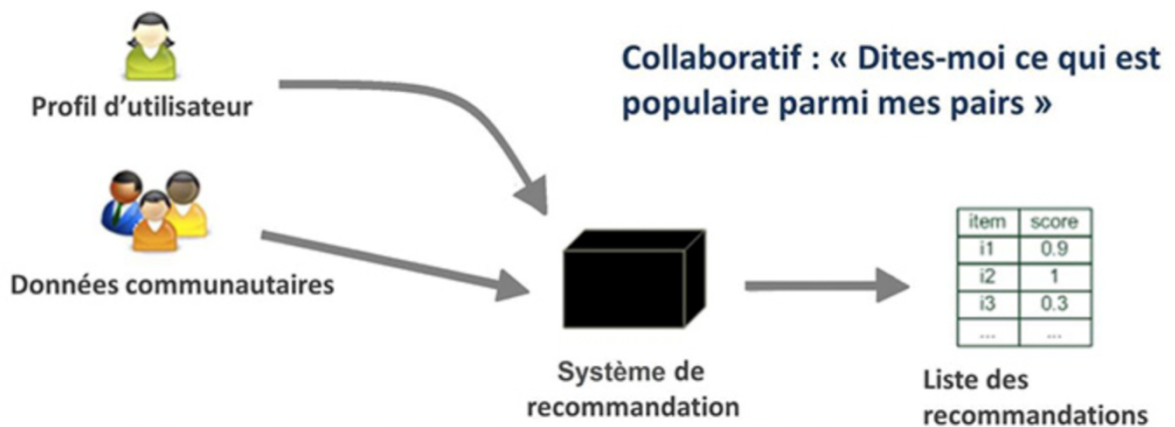


Figure 2 : Approche basé sur filtrage collaboratif [1]

1.2.3 Les approches Hybrides :

Un système de recommandation hybride combine des éléments de différentes approches de recommandation pour produire des recommandations. Il peut utiliser à la fois des données externes et les caractéristiques des éléments, combinant ainsi des méthodes collaboratives et basées sur le contenu.

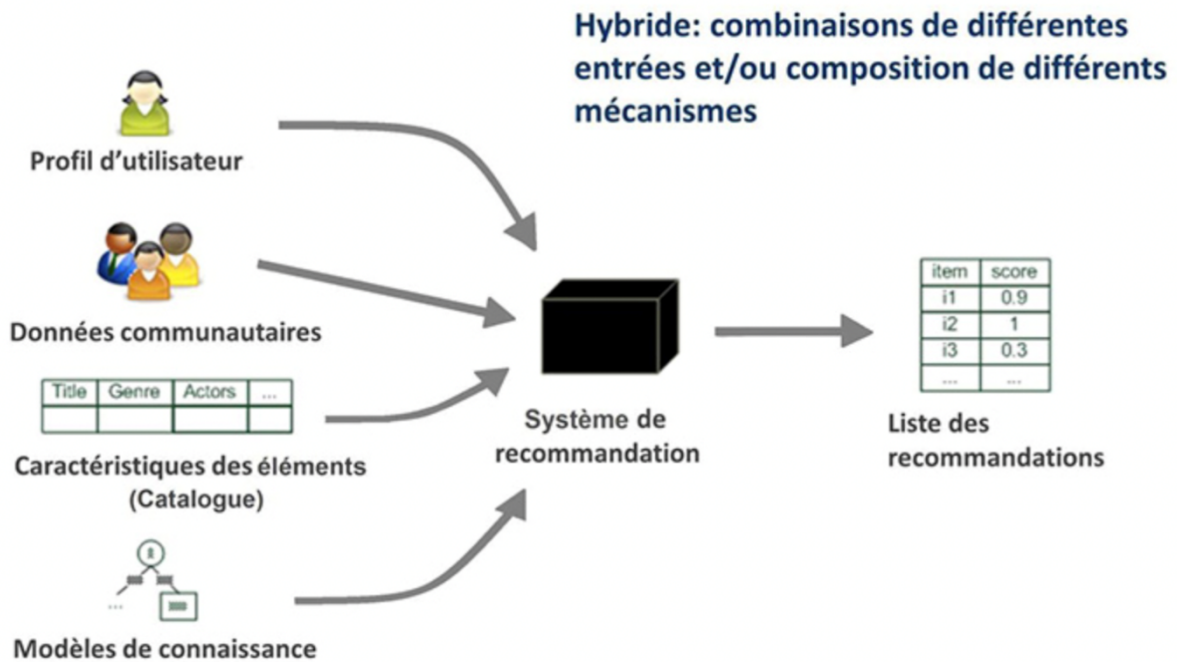


Figure 3 : Approche hybride [1]

1.3 Les principaux algorithmes de recommandation :

Ces principaux algorithmes sont des méthodes automatisées utilisées pour suggérer de contenus qui peut être intéressant pour un utilisateur en fonction de ses préférences ou de son comportement antérieur.

Il existe plusieurs algorithmes de recommandation qui se basent sur les approches citées dans la dernière section 1.1.

1.3.1 Filtrage collaboratif :

Il existe plusieurs manières de calculer les recommandations personnalisées basées sur le filtrage collaboratif [1]. Les algorithmes basés sur utilisateur se basent sur le profil des utilisateurs, tandis

que les algorithmes basés sur élément utilisent les profils de contenu. La factorisation de matrice est également un algorithme de filtrage collaboratif très utilisé. Chacun de ces algorithmes fonctionne différemment pour produire des recommandations.

1.3.1.1 Algorithme basé sur utilisateur :

L'algorithme repose sur le principe que les utilisateurs ayant des comportements et des préférences similaires au passé auront probablement des comportements et des préférences similaires à l'avenir. Cette approche peut être comparée au bouche-à-oreille qui est toujours très efficace pour certaines entreprises. Le filtrage collaboratif offre l'avantage d'être indépendant du contenu, se basant uniquement sur le profil de navigation des utilisateurs pour faire des recommandations personnalisées.

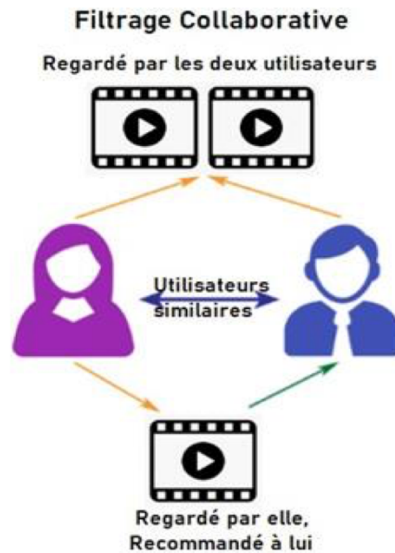


Figure 4 : Approche basé sur utilisateur [28]

Elle se compose de deux majeures étapes :

- **Première étape** : Rechercher les utilisateurs similaires avec l'utilisateur cible.

La similarité pour deux utilisateurs "a" et "b" peut être calculée à partir de la formule donnée,

$$Sim(a, b) = \frac{\sum_p (r_{ap} - \bar{r}_a)(r_{bp} - \bar{r}_b)}{\sqrt{(r_{ap} - \bar{r}_a)^2} \sqrt{(r_{bp} - \bar{r}_b)^2}}$$

r_{up} : l'évaluation de l'utilisateur u sur un élément p

p: les élément

Équation 1 : la formule de la similarité entre deux utilisateurs a et b

- **Deuxième étape** : Prédiction de la note manquante d'un élément.

Maintenant, l'utilisateur cible peut-être très similaire à certains utilisateurs et peut ne pas être très similaire aux autres. Par conséquent, les notes attribuées à un élément particulier par les utilisateurs les plus similaires doivent avoir plus de poids que celles données par les utilisateurs moins similaires, etc. Ce problème peut être résolu en utilisant une approche moyenne pondérée. Dans cette approche, vous multipliez la note de chaque utilisateur par un facteur de similarité calculé à l'aide de la formule mentionnée ci-dessus.

La note manquante peut être calculée comme suit :

$$r_{up} = \bar{r}_u + \frac{\sum_{i \in \text{utilisateurs}} \text{sim}(u, i) * r_{ip}}{\sum_{i \in \text{utilisateurs}} \text{sim}(u, i)}$$

Équation 2 : La formule de calcul de la note manquante

1.3.1.2 Algorithme basé sur élément :

L'algorithme de filtrage collaboratif basé sur élément s'appuie sur les similarités entre les profils d'articles pour faire des recommandations. Il identifie les similitudes entre les articles en utilisant un profil d'article qui est composé de la liste des utilisateurs ayant lu ou aimé cet article. Les articles considérés similaires à un article donné en raison du fait que les mêmes utilisateurs les ont appréciés sont recommandés aux personnes ayant lu cet article. Cette approche fonctionne de manière similaire à l'algorithme User-based, mais se concentre sur les similarités entre les articles plutôt que sur les similarités entre les utilisateurs.

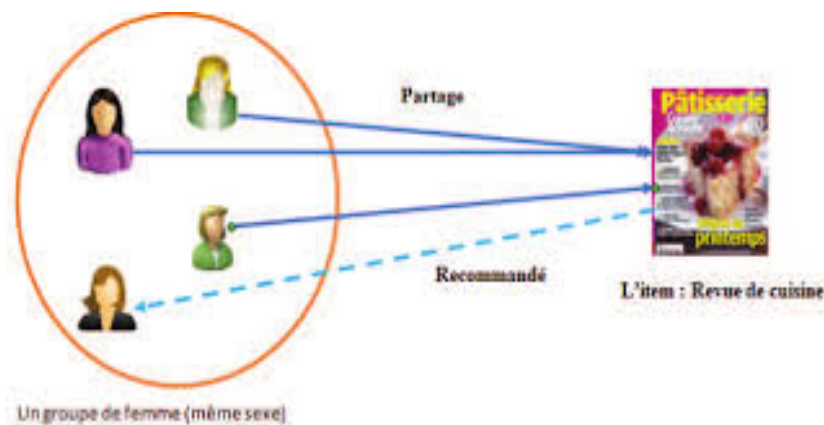


Figure 7 : Approche basé sur élément

Elle se compose de deux majeures étapes :

- **Première étape :** Similitude d'un élément à l'autre la toute première étape consiste à créer le modèle en trouvant la similitude entre toutes les paires d'éléments. La similitude entre les paires d'éléments peut être trouvée de différentes manières. L'une des méthodes les plus courantes consiste à utiliser la similarité cosinus.

$$Similarity(\vec{A}, \vec{B}) = \frac{\vec{A} * \vec{B}}{\|\vec{A}\| + \|\vec{B}\|}$$

Équation 3 : la formule de la similarité entre deux éléments A et B

- **Deuxième étape : Calcul de prédiction :** la deuxième étape consiste à exécuter un système de recommandation. Il utilise les éléments (déjà évalués par l'utilisateur) qui ressemblent le plus à l'élément manquant pour générer une évaluation. Nous essayons donc de générer des prédictions basées sur les notes de produits similaires. Nous calculons cela à l'aide d'une formule qui calcule la note d'un élément particulier en utilisant la somme pondérée des notes des autres produits similaires.

$$rating(U, I_i) = \frac{\sum_j rating(U, I_j) * s_{ij}}{\sum_j s_{ij}}$$

Équation 4 : la formule de calcul de la note d'un élément U en utilisant la somme pondérée.

1.3.1.3 Factorisation des matrices :

La décomposition de matrice est une méthode qui consiste à diviser une matrice en plusieurs autres matrices. Le but est d'approximer la matrice observée en calculant le produit des matrices obtenues. Ce produit donne une matrice prédite. En mathématiques, on représente chaque notation effectuée par un utilisateur U pour un item I par une matrice M, où la dimension de la matrice est $\|U\| \times \|I\|$.

Il existe plusieurs méthodes notamment :

- **La méthode SVD :**

Une méthode bien connue pour identifier les caractéristiques dans une matrice est la Décomposition en Valeurs Singulières (ou SVD). Or, la méthode SVD n'est pas recommandée dans le cas des matrices creuses, au risque de sur-considérer les données. Remplacer les données manquantes par des données générées n'est pas une bonne idée, étant donné que cela biaisera les données collectées et déduites et que cela consommera plus de mémoire de stockage en la

transformant en matrice de forte densité. Ainsi, ces éléments découragent l'utilisation de la méthode SVD pour les systèmes de recommandation.

- **La méthode des facteurs latents :**

Le modèle de facteurs latents représente les items et les utilisateurs par des vecteurs de caractéristiques de même taille. C'est la technique de « rétrocontrôle » (ou « feedback » dans la littérature scientifique) qui va fournir deux manières d'obtenir des informations sur les utilisateurs : explicite et implicite.

Grâce aux préférences déterminées avec le rétrocontrôle, on obtient les vecteurs de caractéristiques des utilisateurs. Chaque item possède un vecteur de caractéristiques qui lui est propre. Ainsi, plus la correspondance entre les caractéristiques d'un utilisateur avec celles d'un item est grande, plus la recommandation sera considérée comme pertinente. Mathématiquement, on détermine K facteurs latents, liés aux données de U et I

Le but étant de décomposer la matrice M en deux matrices P et Q de dimensions respectives $\|U\| \times K$ et $\|I\| \times K$.

Ainsi, la matrice initiale M pourra être approchée par le produit matriciel $P * Q^T$.

La prédiction sur la correspondance entre les caractéristiques de u et i se calcule de la façon suivante :

$$\widehat{r}_{ui} = q_i^T p_u$$

Équation 5 : prédiction sur la correspondance entre les caractéristiques de u et i

1.3.2 Approche basée sur le contenu :

L'algorithme basé sur le contenu [1] analyse un ensemble de contenu en se concentrant sur les items et en ignorant dans un premier temps les utilisateurs. La première étape consiste à transformer le profil d'un item en un vecteur de valeurs. Ensuite, le système calcule un vecteur de poids pour chaque utilisateur, où chaque poids représente l'importance accordée par l'utilisateur à une caractéristique spécifique de l'item. Ce vecteur de poids peut être déterminé à partir des commentaires et réactions des utilisateurs (par exemple, le bouton "j'aime/je n'aime pas"). La similitude entre les caractéristiques des éléments et le profil de l'utilisateur peut être mesurée de différentes façons :

- **L'indice de Dice** [2] ou d'autres mesures de similarité : L'indice de similarité de SÖRENSEN DICE est un indicateur statistique qui mesure la similarité de deux échantillons et s'écrit comme suit :

$$S_s = \frac{2(A \cap B)}{A + B}$$

Équation 6 : la formule de calcul de l'indice de dice

- **La méthode TF-IDF** [3] (Term Frequency-Inverse Document Frequency) évalue l'importance des mots dans un document par rapport à une collection ou un corpus. Elle augmente le poids proportionnellement au nombre d'occurrences du mot dans le document et en fonction de la fréquence du mot dans le corpus. Elle est souvent utilisée dans la recherche d'information et en particulier dans la fouille de texte.
- Les techniques basées **sur la similarité des espaces vectoriels** (les approches bayésiennes, les arbres de décision, etc.) couplées avec des techniques statistiques, lorsqu'il y a trop de mots-clés.

1.4 Comparaison entre le filtrage collaborative et le basées sur le contenu :

	Filtrage collaborative	Filtrage basé sur contenu
La quantité d'information	Le filtrage collaboratif n'a besoin de rien d'autre que la préférence historique de l'utilisateur sur un ensemble d'éléments à recommander.	L'approche basée sur le contenu nécessite une bonne quantité d'informations sur les fonctionnalités des éléments, plutôt que d'utiliser les interactions et les commentaires de l'utilisateur.
Le temps d traitement des données	La connaissance du domaine n'est pas nécessaire car les intégrations sont automatiquement apprises.	la représentation des éléments est dans une certaine mesure conçue à la main, cette technique nécessite beaucoup de temps.

La capacité de développement	Le modèle peut aider les utilisateurs à découvrir de nouveaux intérêts et bien que le système ne connaisse pas l'intérêt de l'utilisateur pour un élément donné, le modèle peut toujours le recommander car des utilisateurs similaires sont intéressés par cet élément.	Un modèle basé sur le contenu ne peut faire que des recommandations basées sur les intérêts existants de l'utilisateur et le modèle n'a donc qu'une capacité limitée à développer les intérêts existants des utilisateurs.
Le comportement d'algorithme	L'algorithme collaboratif utilise uniquement le comportement de l'utilisateur pour recommander des éléments.	Le filtrage basé sur le contenu, nous devons connaître le contenu de l'utilisateur et de l'élément.

Table 1 : Comparaison entre le filtrage collaborative et le basées sur le contenu

1.5 Les problèmes des systèmes de recommandation :

1.5.1 Le démarrage à froid :

Dans les systèmes de recommandation, le problème du "démarrage à froid" se produit lorsque le système rencontre un nouvel utilisateur pour la première fois et n'a pas suffisamment d'informations sur ses préférences pour fournir des recommandations fiables. Pour surmonter ce problème, des algorithmes comme (populaires ou random) peuvent être utilisés pour recommander les éléments les plus populaires ou les plus fréquemment consommés pour permettre au système de commencer à comprendre les préférences de l'utilisateur. Dès que le système a suffisamment d'informations, il peut commencer à utiliser des algorithmes plus avancés pour personnaliser les recommandations.

1.5.2 Le manque de fraîcheur des données :

En effet, les systèmes de recommandation doivent être régulièrement mis à jour pour tenir compte des nouveaux produits, clients, et informations contextuelles qui ont été ajoutées depuis

la dernière formation. Cela permet d'assurer la qualité et la pertinence des recommandations fournies. La fréquence de mise à jour dépend de la fréquence de mise à jour des données et de la complexité du modèle de recommandation utilisé.

1.5.3 Le manque de diversité des données :

Si l'algorithme de recommandation se base uniquement sur les produits les plus populaires ou les mieux notés, il y a un risque d'offrir toujours les mêmes produits aux utilisateurs, ce qui peut entraîner un appauvrissement de l'offre pour les utilisateurs. Il est donc important de prendre en compte des facteurs supplémentaires tels que les préférences personnelles de chaque utilisateur, les tendances actuelles et les nouveaux produits disponibles, pour offrir une expérience de recommandation plus variée et adaptée aux besoins individuels des utilisateurs.

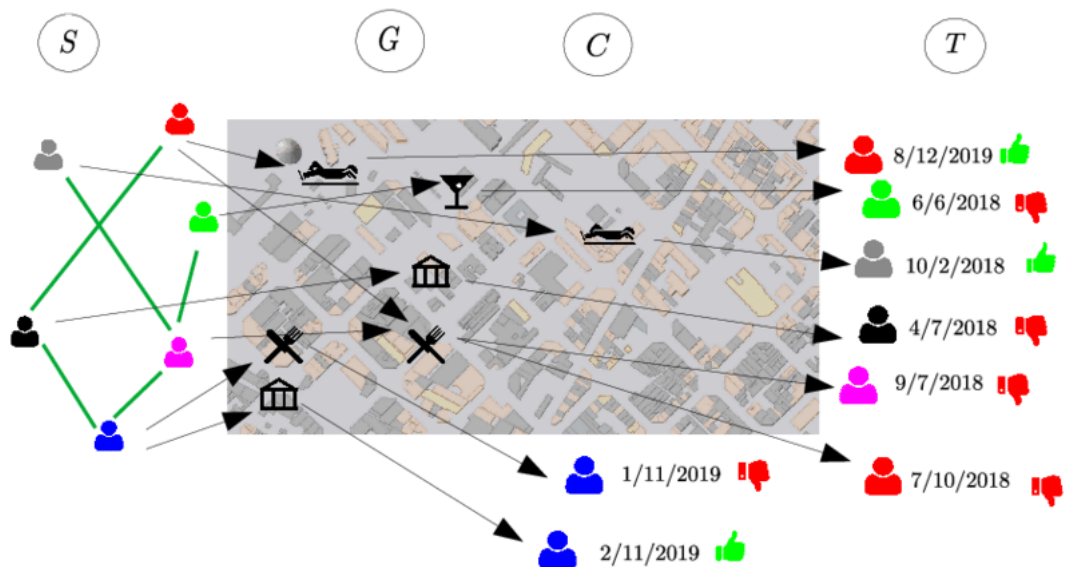
1.6 Conclusion :

Dans ce chapitre nous avons étudiées les systèmes de recommandation en générale et les approches utilisées, ainsi les défis qui peut causer des problèmes à telles approches. En effet, la qualité et la quantité des données utilisées sont déterminantes pour l'efficacité des systèmes de recommandation. De plus, une diversité adéquate des données permet d'éviter les biais dans les recommandations et d'améliorer la variété des offres proposées aux clients. Les algorithmes de recommandation peuvent être avancés, mais sans des données suffisantes et diversifiées, ils ne pourront pas fonctionner de manière optimale.

Chapitre 2 : la recommandation des POIs

2.1 Introduction :

Les systèmes de recommandation de points d'intérêt [4] basés sur les réseaux sociaux géolocalisés (LBSN) est influencée par divers facteurs tels que la géolocalisation, la démographie, les habitudes de déplacement, les réseaux sociaux et les préférences personnelles, ce qui complique les algorithmes de recommandation et nécessite une approche adaptée pour résoudre ce problème.



Différents types d'informations qui peuvent être stockées et collectées à partir des LBSN.

En raison de la grande richesse d'informations disponibles sur ces réseaux sociaux, plusieurs objectifs de recommandation ont été définis, notamment recommander des lieux, des voyages, des activités ou des amis. La recommandation de POI présente certaines particularités qui diffèrent de la recommandation traditionnelle. Ceux-ci incluent, mais ne sont pas limités à :

- La parcimonie désigne généralement la qualité d'être économique ou sobre dans l'utilisation des moyens. Dans le cas de la recommandation de POI, elle peut se référer à la relation entre les préférences observées et potentielles de l'utilisateur. En d'autres termes, la recommandation doit être simple, claire et directe, en ne considérant que les informations les plus pertinentes pour les recommandations.
- Influences externes : Ces influences externes sont souvent considérées comme des contextuelles qui peuvent offrir des informations supplémentaires pour améliorer les performances de recommandation. Il est important de prendre en compte ces influences pour une recommandation de POI plus précise et personnelle pour chaque utilisateur. Cependant, il est également important de considérer les problèmes de protection de la vie privée des utilisateurs, car la collecte et l'utilisation de ces informations peuvent poser des défis éthiques.

2.2 Sources d'information alternatives :

2.2.1 Revues Textuelles :

Pour aider à la recommandation de POI. La quantité et la qualité de ces données peuvent varier considérablement et doivent être prises en compte lors de la recommandation. En fin de compte, les informations textuelles [4] peuvent offrir une vision plus complète et plus précise des préférences des utilisateurs et aider à personnaliser davantage les recommandations soit des textes aussi longs et plus élaborés (comme dans Yelp¹) ou aussi courts et des textes concis (comme les soi-disant astuces dans Foursquare²). Cependant, il est important de prendre en compte les biais subjectifs dans les commentaires et d'adopter une approche équilibrée pour intégrer ces informations dans la recommandation afin d'assurer la qualité et l'exactitude de la recommandation produite. Ce type d'information textuelle peut être exploitée par des approches de recommandation et structurer cette information en utilisant des techniques de modélisation thématique comme l'allocation de Dirichlet latente (LDA) [5] ou l'analyse sémantique latente (LSA) [6] :

¹<https://www.yelp.com/>

²<https://foursquare.com/>

- L'allocation de Dirichlet latente (LDA) :** Le modèle LDA (Latent Dirichlet Allocation) est utilisé pour comprendre la structure sémantique des textes en identifiant les thèmes (ou les sujets) qui sont disséminés à travers un ensemble de documents. Le modèle fonctionne en considérant chaque document comme un mélange de différents thèmes et en assignant à chaque mot dans le document un thème particulier. Le modèle LDA est un modèle probabiliste qui utilise une approche générative pour expliquer les observations en utilisant des groupes de similarité de données. En utilisant LDA dans les systèmes de recommandation de POI, les textes d'évaluation peuvent être structurés en thèmes qui peuvent être utilisés pour faire des recommandations basées sur les intérêts des utilisateurs.

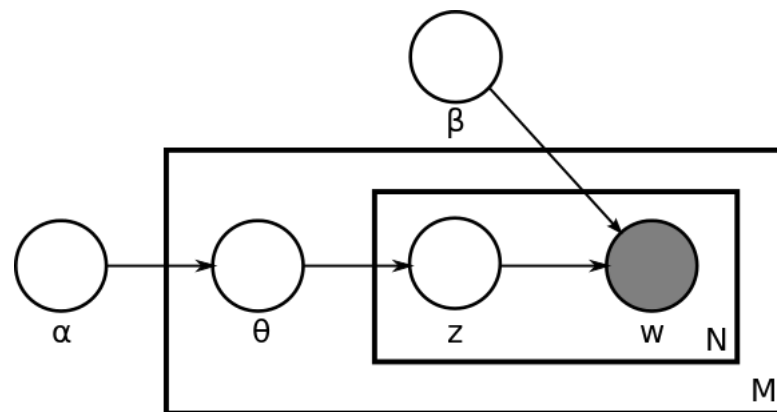


Figure 13 : Fonctionnement de la LDA

- L'analyse sémantique latente LSA :** L'analyse sémantique latente (LSI) est une méthode de traitement de la langue naturelle qui utilise la décomposition en valeurs singulières (SVD) pour simplifier les calculs. Il permet d'isoler les mots clés qui décrivent le "contenu" principal d'une phrase en enlevant les mots courants tels que les conjonctions, les pronoms et les verbes dits "stop words". Cette méthode aide à classer les documents en fonction de leur contenu sémantique, ce qui peut être utile dans la recherche d'information et la recommandation de contenu.

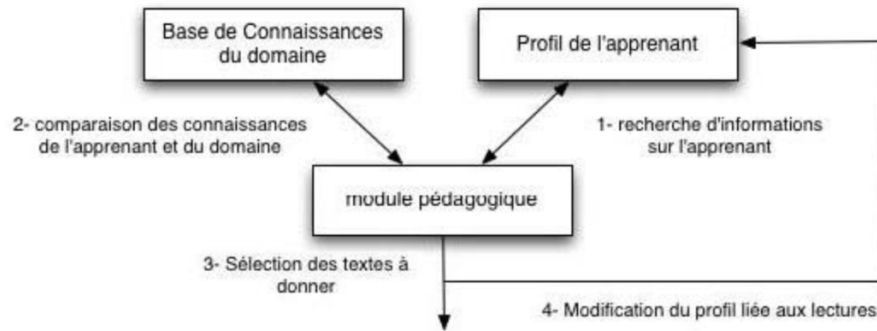


Figure14 : la figure illustre le fonctionnement de LSA [29]

L'analyse des commentaires textuels dans les avis d'utilisateurs peut fournir des informations plus utiles et de qualité supérieure sur les intérêts des utilisateurs. Cela permet de combiner des informations sur la fréquentation d'un lieu avec les sentiments de l'utilisateur à ce sujet. Il est important de faire une distinction entre les informations obtenues à partir des fonctionnalités POI et celles obtenues à partir des commentaires textuels, car les informations provenant des commentaires sont subjectives et les informations provenant des fonctionnalités POI sont plus stables. Les informations textuelles peuvent également fournir des informations de contenu qui peuvent être utilisées pour la classification. Ce type d'information peut être exploitée par des approches de recommandation et structurer cette information en utilisant des techniques d'analyses. Par exemple :

2.2.1.1 Analyse des sentiments :

L'analyse de sentiments [10] consiste à identifier les opinions, attitudes et émotions exprimées dans un document textuel, comme les commentaires d'utilisateurs sur les réseaux sociaux, les forums en ligne, les blogs, etc. L'objectif est d'attribuer une polarité positive, négative ou neutre à ces documents. Cela peut être utile pour comprendre les opinions et les attitudes des consommateurs sur les produits, les services, les entreprises, etc. Les sources de textes en ligne telles que les forums de discussion sur Internet sont très importantes pour les entreprises, car elles peuvent donner une vue à jour de la perception des consommateurs.

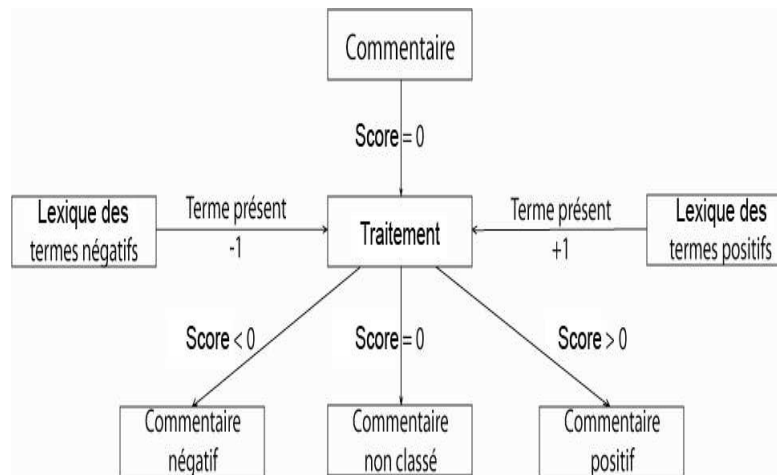


Figure15 : Méthode de classification d'opinion.

Nous présenterons des méthodes différentes pour effectuer la classification automatique de texte d'opinion :

- **Symbolique** : cette méthode [10] est une approche pour l'analyse de sentiments qui se base sur l'analyse syntaxique d'un texte pour extraire les relations sémantiques et émotionnelles présentes. Cette analyse est réalisée par un analyseur fonctionnel et relationnel [9] qui traite le texte phrase par phrase et identifie les relations syntaxiques. L'utilisateur peut personnaliser la grammaire utilisée pour l'analyse en ajoutant de nouvelles règles pour extraire les relations souhaitées. Par exemple, une phrase comme "Je déteste !!!" peut être analysée pour identifier la relation de sentiment négatif (détester).
- **Statistique** : cette méthode [10] de l'analyse de sentiments se base sur l'apprentissage automatique et a été développée par des travaux de Pang et al. (2002), (2004) et (2005). Elle propose deux axes de classification possibles : l'opposition subjectif-objectif ou la distinction des opinions subjectives en positif-négatif. Pour améliorer la classification sur l'axe positif-négatif, la méthode consiste à supprimer d'abord du texte les phrases objectives et à faire la classification uniquement sur la partie subjective du texte.
- **Hybride** : La méthode hybride combine les avantages des méthodes statistique et symbolique pour fournir une analyse de sentiments plus précise et efficace. Elle utilise les techniques d'apprentissage automatique pour construire les dictionnaires d'opinions

nécessaires à l'analyse symbolique, tout en profitant des avantages de l'analyse statistique pour traiter les données textuelles de manière plus efficace. Les méthodes hybrides peuvent améliorer la précision et la fiabilité de l'analyse de sentiments en intégrant les informations provenant de différentes sources et en utilisant des techniques de traitement du langage naturel avancées.

2.2.1.2 Vader (Valence Aware Dictionary and Sentiment Reasoner):

Valence Aware Dictionary and Sentiment Reasoner ou Vader est un outil d'analyse des sentiments développé par Hutto et ses collègues en 2014. Il utilise une approche hybride pour évaluer les sentiments dans un texte en combinant les méthodes statistique et symbolique. Il utilise des dictionnaires de mots ainsi que des règles pour attribuer un score de sentiment à chaque phrase d'un texte. Les scores sont ensuite normalisés et classés en quatre catégories : négatif, neutre, positif et composé. Cet outil est spécifiquement adapté aux sentiments exprimés dans les médias sociaux et est open-source sous la Licence MIT [17].

2.2.2 Informations latérales riches sur les articles :

Les systèmes basés sur les points d'intérêt (POI) sont des systèmes de recommandation qui permettent de recommander des lieux intéressants à visiter en fonction des préférences de l'utilisateur. Les éléments POI sont associés à une information plus détaillée [4], telles que les genres pour les films, livres ou musiques. Les lieux sont souvent liés à une catégorie de POI spécifique (comme les restaurants, hôtels, parcs, musées, etc.) qui peuvent avoir différents niveaux de spécificité, comme un restaurant vietnamien ou un restaurant asiatique. Ces informations sont utiles car elles permettent aux utilisateurs de se concentrer sur les types de POI qui les intéressent le plus. Ce type d'information peut être exploitée par des approches de recommandation qui suit :

2.2.2.1 Amazon.com Delivers :

Le système de livraison d'Amazon.com [7] permet aux clients de personnaliser les recommandations en choisissant des catégories/genres spécifiques dans lesquelles ils sont

intéressés. Les abonnés reçoivent régulièrement des mises à jour par courriel sur les dernières recommandations dans les catégories auxquelles ils se sont abonnés.

2.2.2.2 Style Finder :

Le système Style Finder de Levi StrausTM [7] utilise les préférences des clients pour la musique, les looks et le divertissement pour recommander des articles de vêtements. Les clients indiquent leur genre, puis évaluent différents termes ou sous-catégories dans les catégories de Musique, Looks et Amusement. En fonction de leurs évaluations, le système recommande des vignettes de 6 articles de vêtements.

2.2.2.3 MovieMap :

La fonction Movie Map [7] de Reel.com recommande des films aux clients en utilisant des algorithmes de recommandation en fonction de différents critères tels que le genre, le type de film, le format de visionnage et le prix. Les clients peuvent limiter les résultats à des "Sleepers" ou aux "Best of This genre". Les recommandations sont basées sur les préférences de l'éditeur et les critères de sélection des clients.

2.3 Système de recommandation basée sur le contenu textuelle et les catégories :

L'importance du contexte est cruciale pour améliorer la qualité des systèmes de recommandation touristiques [12]. La disponibilité de la localisation est une des informations contextuelles qui ont permis aux utilisateurs de partager leur position sur les sites Web et de donner leur avis sur les endroits visités, et cela a permis beaucoup de systèmes de recommandation de se baser sur l'analyse de ces commentaires [11], tandis que d'autres se baser sur la catégorie.

"Im feeling Loco" est une application pour smartphones qui propose une recommandation contextuelle en se basant sur la localisation [13]. Elle fournit des recommandations plus précises en utilisant des informations telles que les données temporelles, les catégories et la localisation. En basant sur le profil de l'utilisateur sur Foursquare. "My Eat !" est une application proposée qui permet de localiser les restaurants dans un environnement mobile [15]. Elle se sert des données de localisation pour renforcer l'expérience culinaire des utilisateurs.

Notre projet consiste à développer une application pour smartphones qui fournira des recommandations personnalisées à l'utilisateur. Nous combinerons une recommandation contextuelle basée sur la localisation [15] et la préférence catégorielle [13] pour une expérience de recommandation plus approfondie. Nous intégrerons également des modèles d'analyse lexicale [9] [10] et le modèle d'analyse de sentiments VADER [16] [17] pour une meilleure compréhension des préférences et des opinions des utilisateurs.

2.4 Conclusion :

Dans ce chapitre nous avons étudiées les systèmes de recommandation contextuels qui jouent un rôle clé dans l'amélioration de l'expérience utilisateur en fournissant des recommandations personnalisées en fonction du contexte dans lequel ils se trouvent. Ils utilisent une variété de techniques, notamment l'analyse de données, l'apprentissage automatique et les algorithmes de filtrage collaboratif, pour comprendre les préférences de l'utilisateur et lui suggérer des produits, des services ou du contenu en conséquence.

Chapitre 3 : conception de système

3.1 Introduction :

Le but de ce chapitre est de présenter les étapes de conception de notre système de recommandation. Par conséquent, les étapes suivantes sont décrites en détail : collecte de données, analyse des sentiments et principes de recommandation.

3.2 Méthodologie :

Dans cette section, nous présentons l'ensemble des méthodes et des techniques utilisées dans de notre système.

3.2.1 Collection des données :

Nous avons collecté les données de plusieurs points d'intérêt (restaurant, hôtel, plage, parc d'attraction, des loisirs commerciaux) à partir de Google Maps et les enregistrer en utilisant Firebase comme un système de gestion de base de données. Comme le montre la figure suivante :

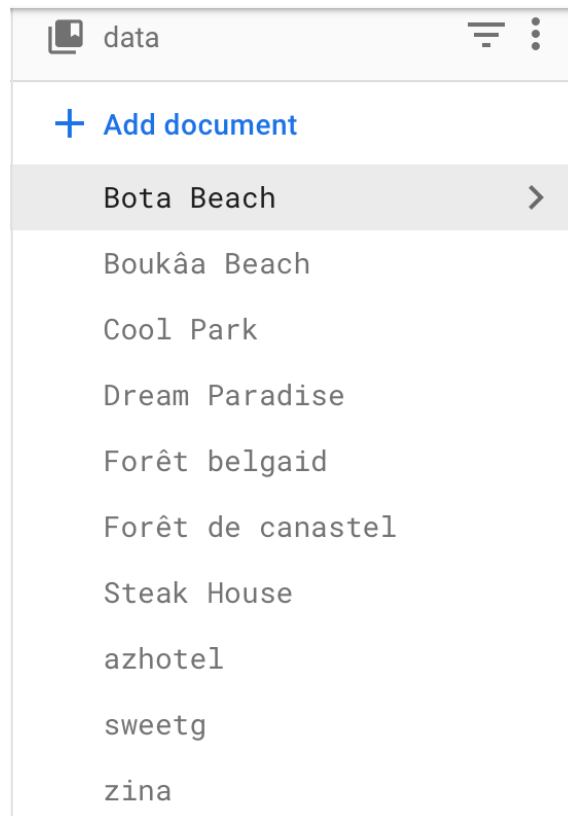


Figure 16 : Un exemple d'une partie de la collection des données qui concerne les points d'intérêt.

Chaque point est décrit par le nom, la catégorie, localisation, des tags, score des commentaires ainsi une autre collection des commentaires.

3.2.2 Analyse des sentiments :

Pour mieux représenter les préférences des utilisateurs, nous avons appliqué l'analyse des sentiments sur l'ensemble des commentaires. Le vecteur de caractéristique de chaque lieu contient une collection de commentaires des utilisateurs qui l'ont déjà visité. Nous avons calculé le score de chaque commentaire.

L'analyseur des sentiments Sentiment Dart [18] est un module de fléchettes qui utilise la liste de mots AFINN-165[29] et le classement des sentiments Emoji pour effectuer une analyse des sentiments sur des blocs arbitraires de texte d'entrée. Sentiment Dart est fortement inspiré de l'analyse des sentiments de Vader.

L'analyseur donne un score soit positif ou négatif :

Exemple de score positif :

```
content : "I liked the recommendation "  
date : May 24, 2023 at 10:35:34 AM UTC+1  
score : 0.5  
userName : "haganiwael"
```

Figure17 : Un exemple réel positif depuis la base.

Exemple de score négatif :

```
content : "but the service was bad "  
date : May 25, 2023 at 11:29:46 AM UTC+1  
score : -0.6  
userName : "ayoubsalah"
```

Figure18 : Un exemple réel négatif depuis la base.

Les Objets retournés par l'analyseur sont :

- Score : Score calculé en additionnant les valeurs de sentiment des mots reconnus.
- Comparatif : score comparatif de la chaîne d'entrée.
- Calcul : Un tableau de mots qui ont une valence négative ou positive avec leur score AFINN respectif.
- Token : Tous les jetons comme les mots ou les emojis trouvés dans la chaîne d'entrée.
- Mots : liste des mots de la chaîne d'entrée qui ont été trouvés dans la liste AFINN.

- Positif : liste des mots positifs dans la chaîne d'entrée qui ont été trouvés dans la liste AFINN.
- Négatif : liste des mots négatifs dans la chaîne d'entrée qui ont été trouvés dans la liste AFINN.

- Notre système ne s'intéresse qu'au comparatif, il est calculé à l'aide de l'équation suivante :

le score de liste des positifs + le score de liste des négatifs / le nombre des tokens

Équation 7 : Formule de calcul de comparative

Cette approche nous laisse avec un point médian de 0 et les limites supérieure et inférieure qui sont contraintes respectivement à 5 points positifs et 5 points négatifs, cela évite le problème où le commentaire ne contient que des mots positifs, ce qui peut entraîner un score positif exagéré.

Ensuite on rassemble toutes les notes comparatives de chaque commentaire puis on divise sur le nombre de commentaires. Afin d'avoir un score presque exact.

3.3 Le système de recommandation :

Les données et leurs caractéristiques sont exportées dans la console firebase qui est liée à une application que nous avons développée, nommée "i-Recommend".

L'application recommandera des endroits intéressants à l'utilisateur en utilisant différentes méthodes comme "les plus vues", "les plus appréciées" et la méthode la plus importante est la recommandation basée sur les commentaires des utilisateurs et les catégories visitées.

Par exemple, si un nouvel utilisateur utilise l'application, cela peut causer un problème de démarrage à froid du système, afin de ne pas avoir à faire face à ce problème majeur, nous avons utilisé la méthode "dernier clic". Le système enregistre la dernière catégorie sur laquelle l'utilisateur a cliqué et recommande un poi similaire qui peut être utilisé ultérieurement dans une autre recommandation plus améliorée.

Ensuite, l'utilisateur se retrouvera avec deux options, en ajoutant le lieu à sa collection favoris et en écrivant un commentaire expressif qui sera ajouté à la collection de commentaires d'un poi. Le système utilisera l'analyse des sentiments pour vérifier le score global de la collection, s'il est positif alors nous pouvons recommander cet endroit à d'autres utilisateurs qui sont intéressés par la même catégorie sinon l'endroit tombera en ordre par rapport à d'autres endroits mieux notés.

Pour que l'algorithme ait une meilleure recommandation, chaque point a sa propre table de tags.

Si l'utilisateur est intéressé par une catégorie particulière cela n'empêchera pas l'algorithme de recommander une autre catégorie similaire à sa préférence. Le système enregistrera les traces de l'utilisateur pour avoir une meilleure compréhension de son comportement. Par exemple si un utilisateur décide de visiter un complexe "Hôtel" touristique comme " Zina Beach " qui possède son propre restaurant et sa plage privée, trois catégories différentes, le système dans ce cas lui recommandera des endroits similaires aux dernières catégories mentionnées.

La figure suivante résume le fonctionnement de notre système :

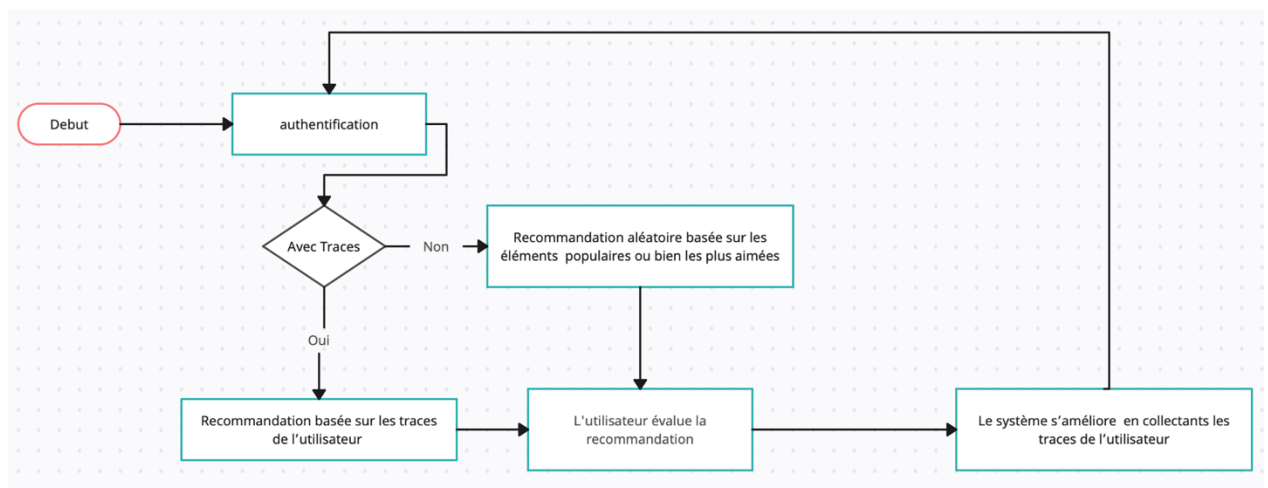


Figure19 : Fonctionnement de « i-Recommend »

3.4 La conception de l'application « i-Recommend » :

3.4.1 Diagramme de cas d'utilisation :

Un diagramme de cas d'utilisation est une représentation graphique des interactions possibles d'un utilisateur avec un système.

Alors qu'un cas d'utilisation lui-même peut approfondir de nombreux détails sur chaque possibilité, un diagramme de cas d'utilisation peut aider à fournir une vue de niveau supérieur du système, La figure ci-dessous représente le diagramme de cas d'utilisation de notre système.

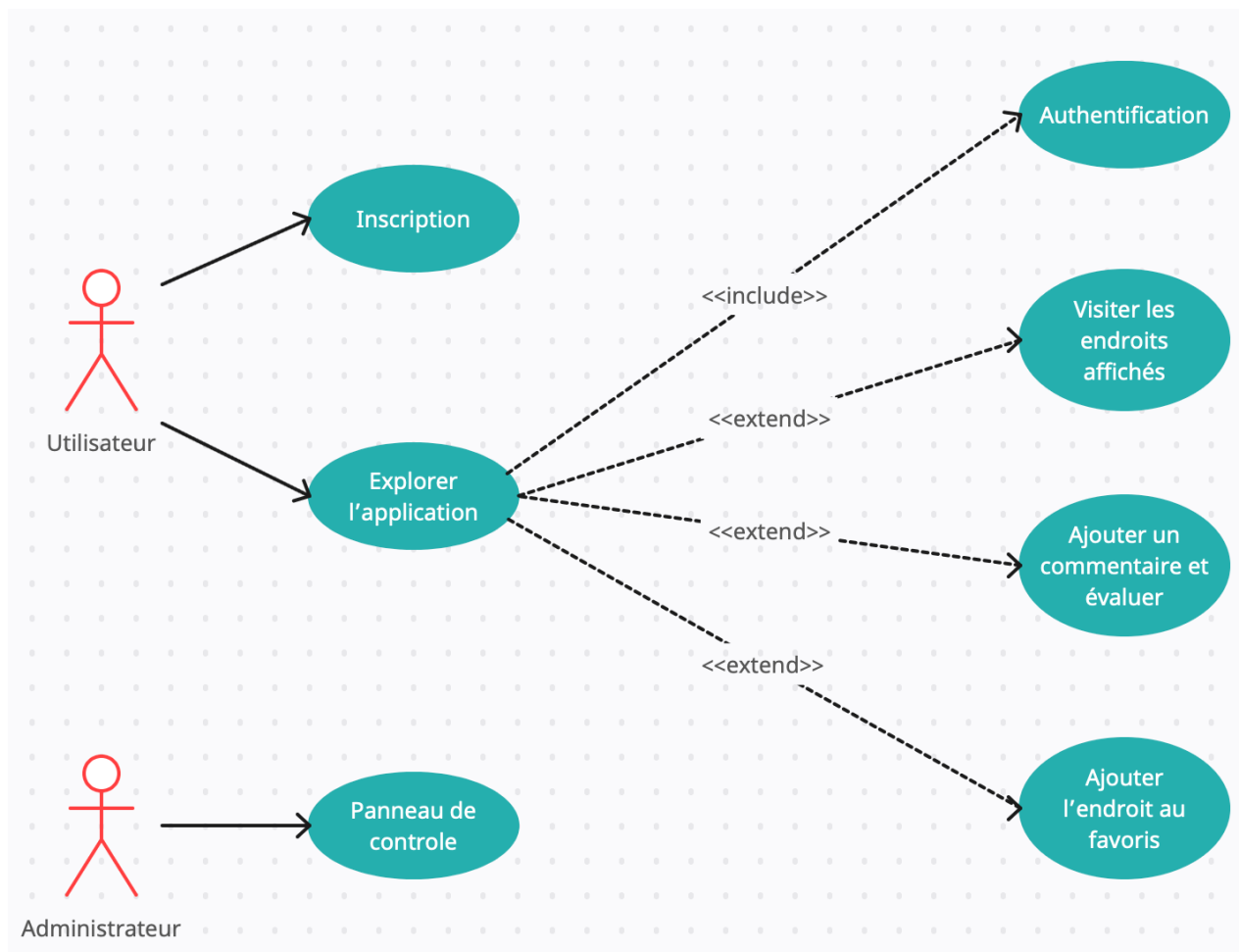


Figure20 : Diagramme de cas d'utilisation de « i-Recommend »

○ **Authentification :**

- **Nom du cas (Titre) :** Traiter l'utilisation de l'application.
- **Objectif et description :** un utilisateur veut profiter des fonctionnalités de l'application.
- **Acteurs :** Système (principal), l'utilisateur (secondaire).

○ **Séquencements :**

▪ **Préconditions :**

-le système doit être ouvert.

-la base de données est disponible.

▪ **Enchaînement nominal :**

Acteur	Système
1. Le cas d'utilisation début lorsqu'un l'utilisateur s'inscrit dans l'application.	2. Le système enregistre les informations de l'utilisateur dans la base des données.
3. L'utilisateur explore l'application.	4. Le système affiche pour l'utilisateur tous les lieux enregistrés dans la base de données en même temps enregistre ses traces pour une utilisation ultérieure
5. L'utilisateur jette un œil sur les endroits affichés par le système. L'utilisateur peut donner son avis sur ces endroits ou les ajouter à sa liste des favoris.	6. Le système enregistre le commentaire d'utilisateur dans la base de données, ainsi l'endroit dans sa collection des favoris.
	7. Le système recommande des endroits adaptés a les traces sauvegardées de l'utilisateur.

8. L'utilisateur exprime son avis sur la recommandation	
9. L'utilisateur quitte l'application.	

Table 2 : Description textuelle de diagramme de cas d'utilisation

▪ **Enchaînement Alternatif :**

-**A1** : l'utilisateur possède déjà un compte. L'enchaînement de A1 démarre au point 3 du scénario nominal :

3-L'utilisateur explore l'application.

▪ **Enchaînement d'erreur :**

-**E1** : courriel ou mot de passe incorrect.

-le cas d'utilisation se termine en échec.

3.4.2 Diagramme de classe :

En génie logiciel, un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant les classes du système, leurs attributs, opérations (ou méthodes) et les relations entre les objets.

Il est utilisé pour la modélisation conceptuelle générale de la structure de l'application et pour la modélisation détaillée, traduisant les modèles en code de programmation.

La figure ci-dessous représente le diagramme de classe de notre système :

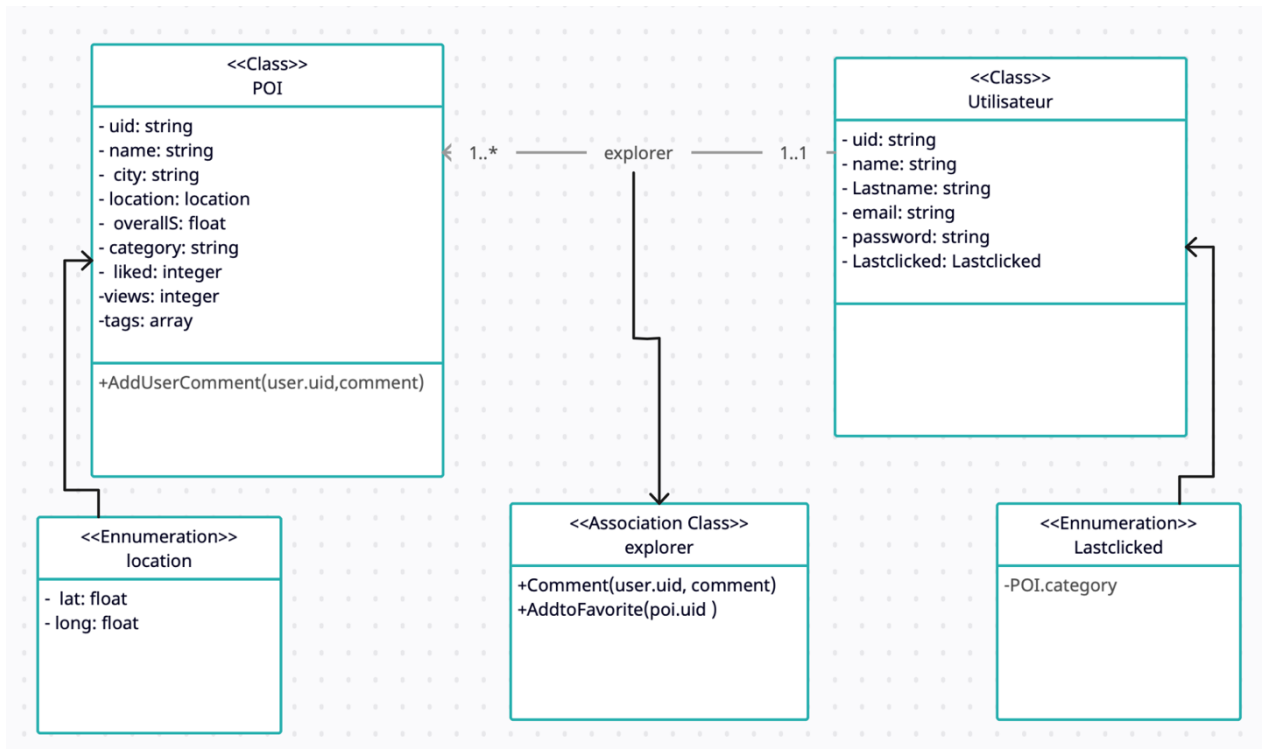


Figure21 : Diagramme de classe de « i-Recommend »

3.4.3 Diagramme de séquence :

Un diagramme de séquence ou diagramme de séquence de système (SSD) montre les interactions de processus organisées en séquence temporelle dans le domaine du génie logiciel. Il décrit les processus et les objets impliqués et la séquence de messages échangés entre les processus et les objets nécessaires pour exécuter la fonctionnalité.

Les figures ci-dessous représentent les interactions de l'utilisateur avec notre système :

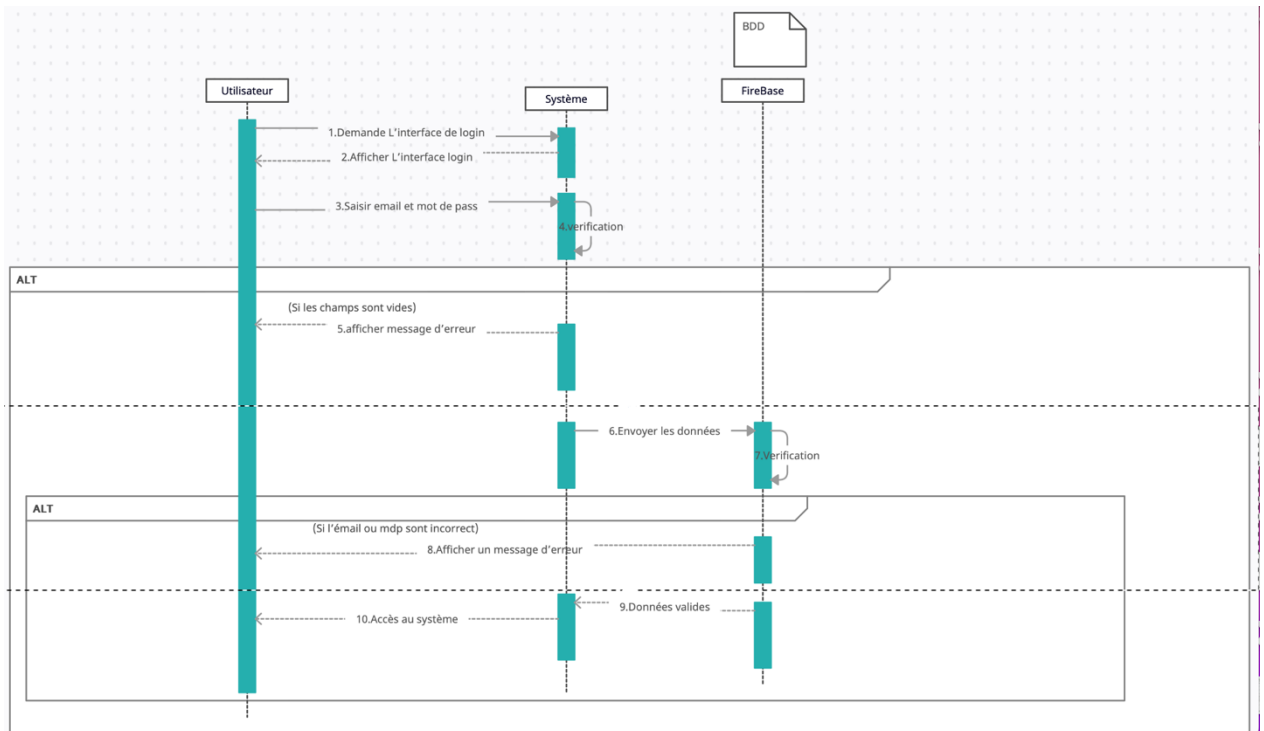


Figure22 : Diagramme de séquence « Log-In »

Chaque utilisateur doit remplir ses informations d'identification telles que le courriel et le mot de passe pour avoir accès à la page d'accueil, le système vérifiera si l'utilisateur existe dans la base de données. Si c'est le cas alors l'utilisateur peut accéder à la page d'accueil.

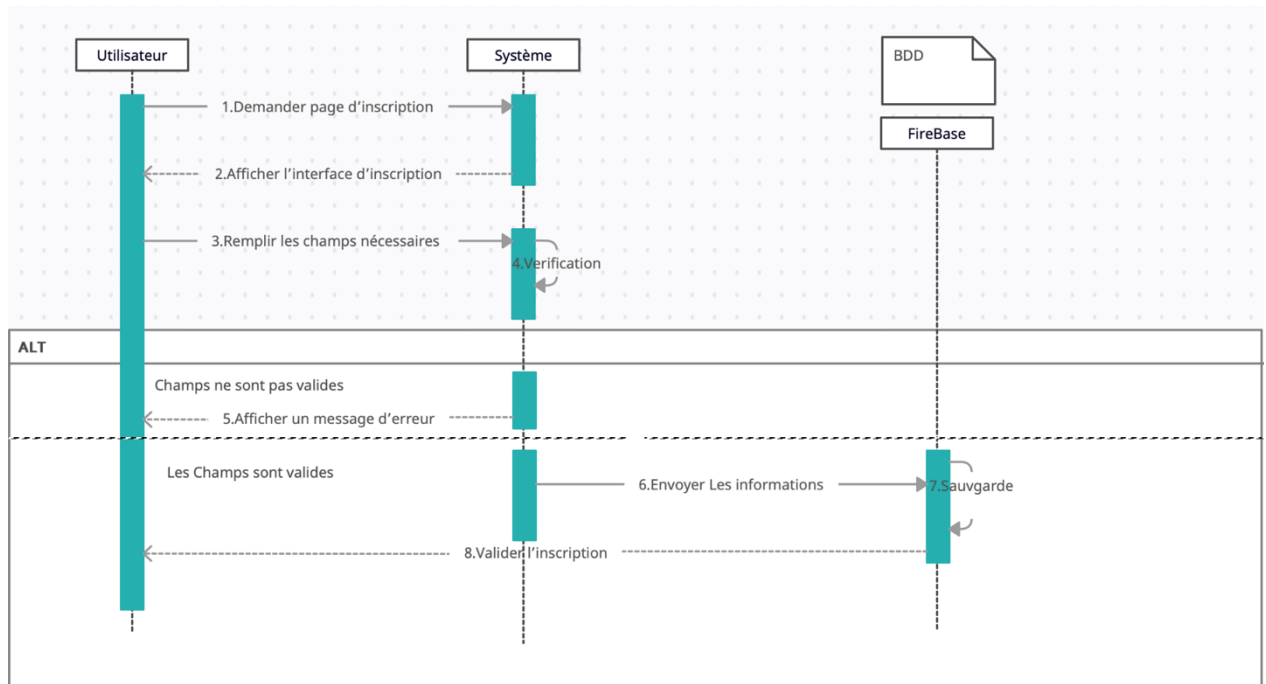


Figure23 : Diagramme de séquence « Inscription »

L'utilisateur doit d'abord remplir un formulaire d'inscription qui contient son nom complet, son adresse électronique, son mot de passe et son numéro de téléphone, le système vérifiera si les informations d'identification demandées ont été saisies correctement, puis les informations de l'utilisateur seront automatiquement stockées dans la base de données

3.5 L'algorithme de recommandation :

L'algorithme peut être décrit comme suite :

- Tout d'abord, l'algorithme affichera cinq endroits aléatoires à l'utilisateur et la possibilité de choisir une catégorie qui correspond à ses préférences.

Entrée : L'utilisateur explore les endroits.

Sortie : Les endroits les plus adaptés à la recommandation.

Début :

Il existe deux scénarios possibles, un nouvel utilisateur (un utilisateur qui utilise l'application pour la première fois) ou bien un ancien utilisateur qu'on a déjà ses traces.

1^{er} Cas : nouvel utilisateur sans traces dans le système (démarrage froid) :

L'algorithme utilise plusieurs méthodes pour gérer ce problème :

- Les plus vus (Most viewed) : il s'agit d'un algorithme prédéfini qui affiche les lieux qui ont le plus de vues de manière décrémente.
- Les plus aimés (Most Liked) : l'algorithme affichera les endroits qui ont le plus de j'aime (like), autrement dit, les endroits que les autres utilisateurs ont le plus aimés.
- Dernier clic (Last-click) : l'algorithme est configuré pour enregistrer la catégorie du dernier clic de l'utilisateur afin qu'il puisse recommander un lieu similaire à celui-ci en fonction du score global de l'analyse des sentiments

2^{ème} cas : utilisateur avec traces dans le système :

Le système recommande cinq endroits à l'utilisateur basé sur le score d'analyse des sentiments, les plus vus, les plus aimés et la similarité, autrement dit :

- Les lieux les mieux notés par l'utilisateur lui-même et les autres utilisateurs.
- Les lieux qui ont un meilleur nombre de vues, c'est-à-dire, les endroits les plus visités par les autres utilisateurs.
- Les endroits les plus aimés par les autres utilisateurs.
- La similarité, le système utilise un modèle basé sur les éléments ou chaque point d'intérêt possède une table des tags. L'algorithme enregistre les traces de chaque utilisateur comme les lieux consultés par ce dernier dans une collection, ensuite elle effectue une comparaison

entre les tables de tags de la collection enregistrée et les autres POIs afin de trouver et recommander un lieu similaire à l'utilisateur.

-La recommandation est toujours basée sur le score d'analyse des sentiments

-Le dernier clic de l'utilisateur ne fonctionne qu'avec les catégories, il est toujours mis à jour en fonction de ce que l'utilisateur souhaite.

-La similarité est basée sur le principe de l'indice de Dice [2][20], l'indice peut varier de 0 (quand les deux lieux sont disjoints) à 1 (quand les lieux sont similaires), par exemple :

Un utilisateur est intéressé par le complexe touristique Zina Beach, il a déjà visité l'endroit et a fait remarquer qu'il avait passé un bon moment en famille et que son séjour était amusant, le commentaire sera analysé pour vérifier s'il était positif ou négatif et pour vérifier les tags. Il sera stocké dans la base de données comme suit :

```
content : "I had a nice family time it was fun being there "  
date : June 12, 2023 at 1:52:22 PM UTC+1  
score : 0.6363636363636364  
▼ tags  
  0 "family"  
  1 "fun"  
userName : "KhelifaAbdelkarim"
```

Figure24 : Exemple d'une table des tags d'un commentaire à partir de la base de données

Ensuite, le système collectera tous les tags dans la collection de commentaires et les stockera dans la table de la POI comme suit :

```
▼ tags
  0 "restaurant"
  1 "beach"
  2 "fun"
  3 "park"
  4 "family"
timeofwork: "8am"
uid: "zina"
```

Figure25 : Exemple d'une table des tags d'un POI à partir de la base de données

Vient ensuite la similarité qui va comparer cette table avec toutes les autres tables des POIs existant pour trouver une recommandation adaptée à l'aide de cette formule :

$$S_s = \frac{2(A \cap B)}{A + B}$$

$(A \cap B)$: le nombre des éléments en commun.

$A + B$: le nombre total des éléments présents dans les deux vecteurs.

L'indice peut varier entre 0 et 1.

Après le calcul le système trouvera des endroits similaires par exemple :

```
▼ tags
  0 "fun"
  1 "family"
  2 "restaurant"
uid: "mostaland"
views: 7
```

Figure26 : Exemple d'une table des tags D'un POI utilisée dans les calculs de la similarité

La figure appartient à MostaLand parc une catégorie différente de Zina Beach mais après l'analyse de similarité l'indice était égal à 0,75 (75%), cela rend l'endroit plus adapté à la recommandation.

3.6 Conclusion :

La phase de conception consiste à obtenir une représentation claire, explicite, cohérente et condensée du système et de son fonctionnement, elle facilite grandement le travail des développeurs ce qui peut nous aider à réaliser le projet et fournir les outils dont nous avons besoin dans le processus de création de notre système.

: Implémentation

Chapitre 4 : l'implémentation

4.1 Introduction :

Dans ce chapitre nous aborderons la création de notre application, l'environnement et langages utilisé et tous les packages importés, puis nous présenterons les étapes détaillées du travail et nous discuterons des résultats.

4.2 Environnement de développement et les langages utilisés :

Un environnement de développement est un espace de travail permettant aux développeurs d'apporter des modifications à leur code. Dans certains cas, le terme environnement de développement est utilisé pour désigner un environnement de développement intégré (IDE). Selon la taille d'un projet plusieurs environnements peuvent être ajoutés à l'infrastructure, on va donc citer ceux qu'on a utilisés.

4.2.1 Dart :

Dart est un langage de programmation [21] conçu par Lars Bak et Kasper Lund [22] et développé par Google en 2011, il s'agit d'un langage orienté objet qui peut être utilisé pour développer des applications Web et mobiles ainsi que des applications serveur et de bureau.

4.2.2 Flutter :

Flutter est un kit de développement de logiciel d'interface utilisateur open source créé par Google [24]. Il est utilisé pour développer des applications multiplateformes [23] pour Android, iOS, Linux, macOS, Windows et le Web à partir d'une seule base de code.

4.2.3 Visual Studio Code :

Visual Studio Code combine la simplicité d'un éditeur de code source avec de puissants outils de développement, crée en 2015 par Microsoft [25], il prend en charge MacOS, Windows et Linux.

4.2.4 Android studio :

Android studio développé en 2014 par Google, est un environnement de développement d'applications mobiles Android [26]. Il prend en charge MacOS, Windows et Linux.

4.2.5 FireBase :

Firestore est un outil d'aide au développement d'applications web, Android, iOS, et Unity conçu en 2011 par Andrew Lee et par James Templin et rachetée par Google en octobre 2014. Il propose un hébergement en nuage et utilise NoSQL pour héberger des bases de données.

4.2.6 Google Maps API :

L'API Google Maps est un ensemble d'interfaces de programmation d'applications payant qui nous permet de communiquer avec ses services. Cela nous permettra de créer des applications simples ou des applications géolocalisées très sophistiquées pour le Web, iOS et Android.

4.2.7 Sentiment Dart :

Sentiment Dart [38] est un module de fléchettes qui utilise la liste de mots AFINN-165[49] et le classement des sentiments Emoji pour effectuer une analyse des sentiments sur des blocs arbitraires de texte d'entrée.

4.3 Les interfaces graphiques du système :

4.3.1 L'interface d'espace Administrateur :



Figure27 : L'interface de l'espace Administrateur

L'administrateur a accès à la base de données il peut voir tous les lieux les ajouter, les supprimer ou les modifier.

4.3.2 La base de données (FireBase) :

Comme nous l'avons expliqué précédemment, nous avons utilisé firebase comme base de données, Il permet une connexion à la base de données en temps réel, ce qui signifie que

plusieurs utilisateurs peuvent voir les modifications apportées aux données lorsque les données sont créées ou modifiées. Il fournit aux développeurs un ensemble d'outils et de services pour créer, mettre à l'échelle facilement, et maintenir leurs applications sans avoir à gérer une infrastructure backend traditionnelle.

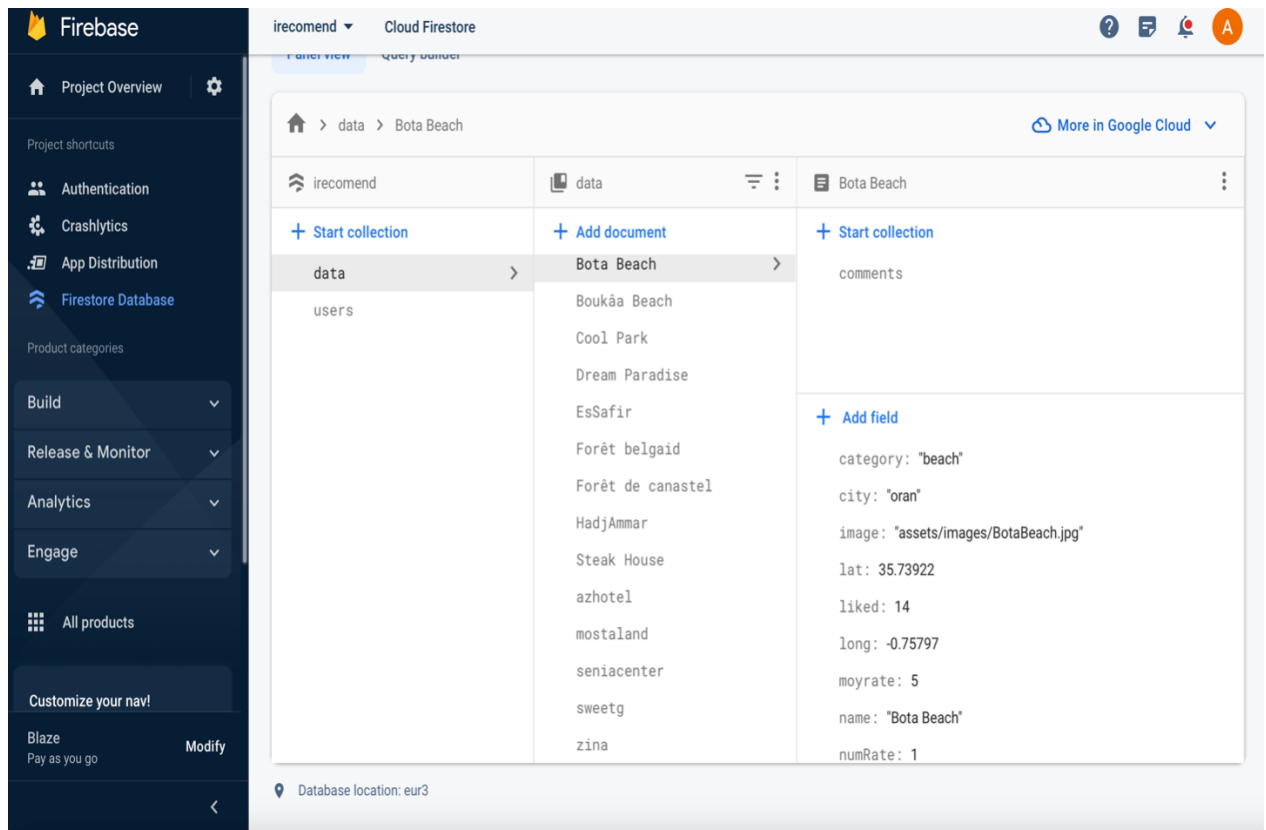


Figure28 : L'interface de la base de données utilisée Firebase

4.3.3 L'interface d'espace Utilisateur :

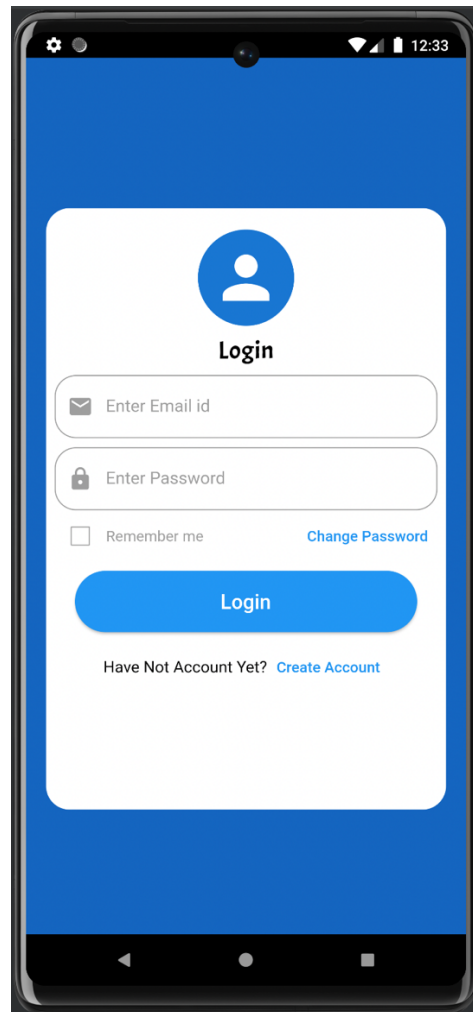


Figure 29 : Connexion au système

Cette interface permet aux utilisateurs de s'authentifier en utilisant un email et un mot de passe avec une option de créer un compte s'ils ne ont pas déjà inscrits dans le système.

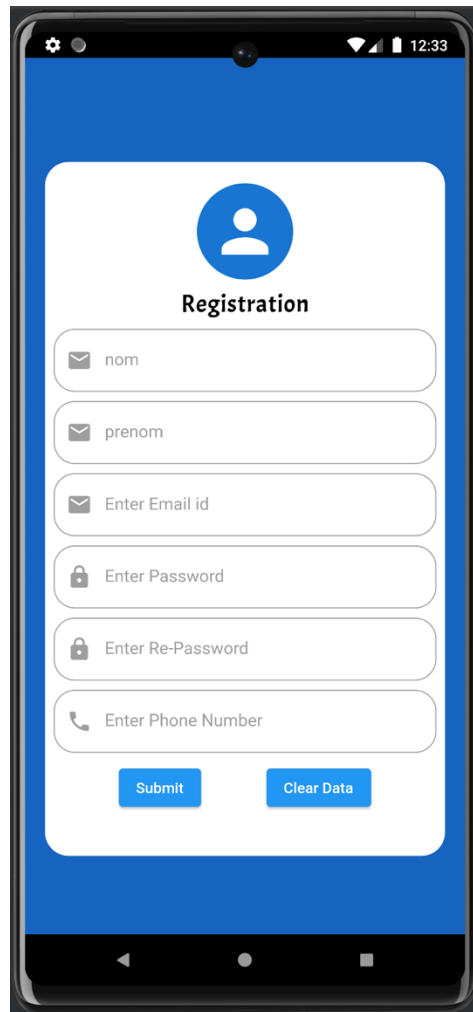


Figure 30 : Formulaire d'inscription

L'utilisateur doit remplir le formulaire d'inscription pour accéder aux différentes fonctionnalités de l'application.

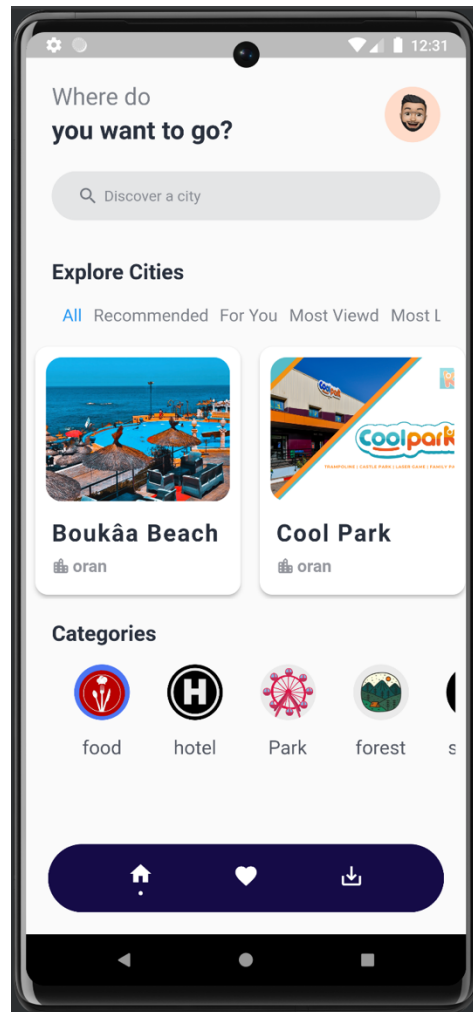


Figure 31 : La page d'accueil de l'application

Après une authentification réussie, les utilisateurs auront accès à la page d'accueil, plusieurs lieux seront affichés au choix avec plusieurs options de recommandation et catégories :

1. Catégorie : l'utilisateur peut choisir la catégorie qui lui convient le mieux.
2. Recommandation : quatre méthodes peuvent être utilisées :
 - Recommended : une recommandation basée sur le dernier clique de l'utilisateur.
 - For You : une recommandation basée sur la similarité des POIs parcouru par l'utilisateur.

- Most viewed : une recommandation basée sur le nombre de vues de chaque POI.
- Most Liked : une recommandation basée sur le nombre de ‘j’aime’ de chaque POI.

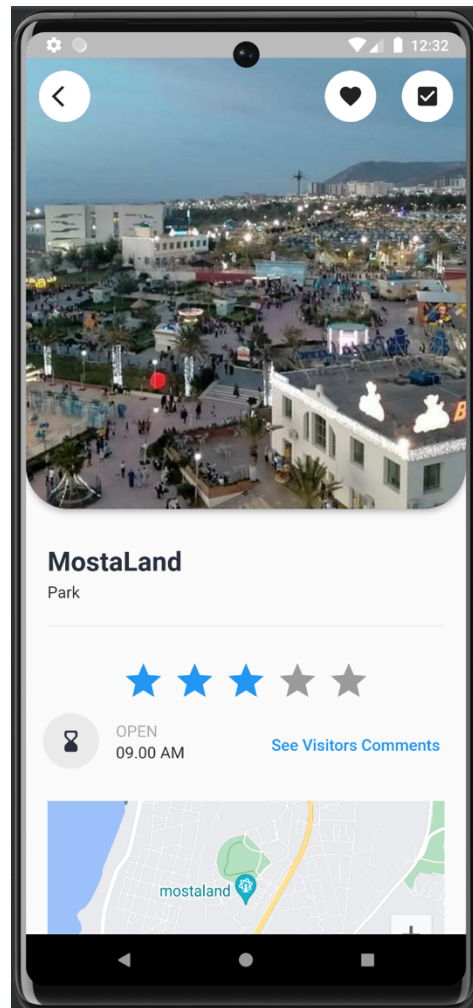


Figure 32 : La page de détails de la POI

Dans cette interface l'utilisateur peut voir une image du point d'intérêt, son nom, sa notation par étoiles et sa localisation exact avec une option pour ajouter la POI aux favoris et afficher les commentaires des autres utilisateurs et ajouter son propre avis.

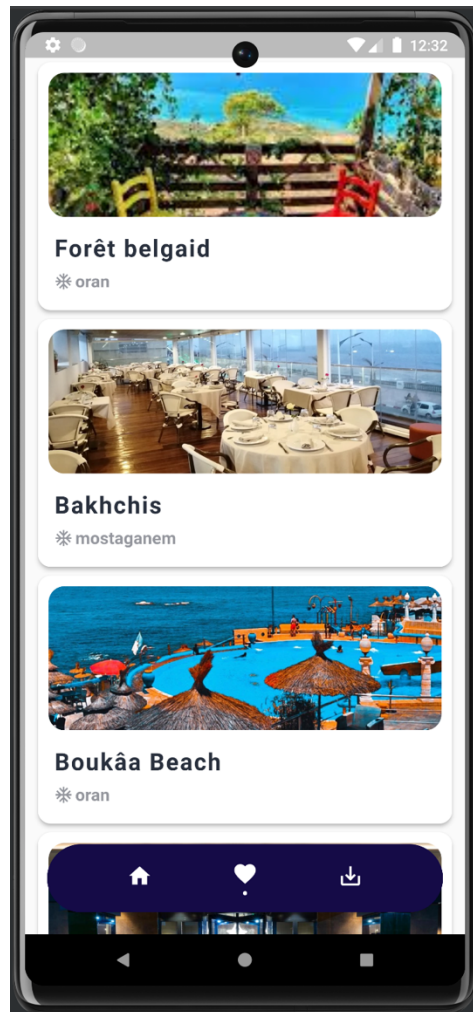


Figure 33 : La page des favoris

Cette interface représente la collection de favoris des utilisateurs.

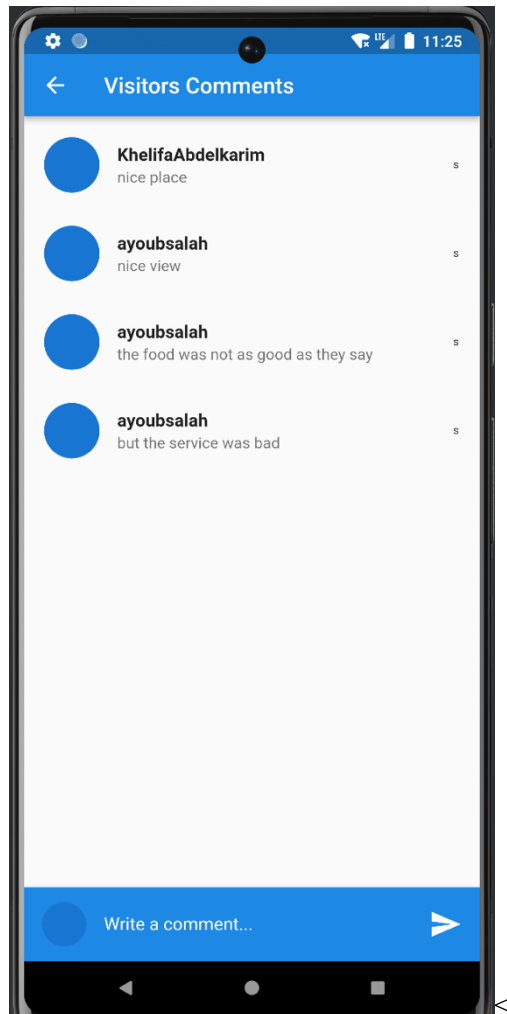


Figure 34 : La section des commentaires

Dans cette page l'utilisateur peut voir les commentaires des autres utilisateurs et il peut ajouter sa propre critique.

4.4 Résultats des tests et discussion :

Pour évaluer la qualité de notre système et de nos algorithmes, nous avons proposé notre application à nos chers étudiants de la faculté des sciences exactes et de l'informatique pour la tester librement comme ils le souhaitent.

Chaque étudiant a créé un compte dans notre application et a consulté des endroits qui l'intéressaient. Chacun d'eux a laissé des traces sur le système telles que des commentaires et des endroits favoris afin que le système puisse trouver une recommandation appropriée pour eux.

Après l'étape de recommandation, l'utilisateur exprime son degré de satisfaction du service présenté par le système sous forme d'un formulaire. Toutes les données ont été rassemblées dans le tableau suivant :

User Id	Moyenne d'évaluation	Précision
1	0,767	1
2	0,625	1
3	0,625	1
4	0,8	1
5	0,7	1
6	0,73	1
7	0,475	0,25
8	0,725	1
9	0,7	1
10	0,7	1
11	0,5	1
12	0,75	1
13	0,525	0,5
14	0,75	1
15	0,7	0,75

Table 3 : Les données collectées d'après les utilisateurs qui ont testé l'application

Nous avons utilisé la mesure de précision [27] qui se définit comme 'la qualité d'être exact', elle permet de mesurer une qualité d'un système donné et d'évaluer ses performances. La précision est calculée par la formule ci-dessous :

$$Précision = \frac{\text{nbr d'endroits jugés intéressant par l'utilisateur}}{\text{nbr d'endroits recommandés par le système}}$$

Équation 8 : Formule de la metrique de la précision

Par exemple :

L'utilisateur avec le numéro d'identification 7 s'est vu présenter 4 recommandations différentes jugées intéressantes par le système auxquelles il leur a attribué une seule note d'évaluation positive (supérieure ou égale à 0,5) et le reste c'était inférieur à 0,5. comme le montre la table 4.

La précision de la recommandation est calculée comme suit :

$$Précision = \frac{1}{4} = 0,25$$

User Id	Recommandation	Évaluation
7	Recommended	0,8
7	For You	0,3
7	Most Liked	0,4
7	Most Viewed	0,4

Table 4 : Un exemple d'évaluation d'un utilisateur qui a testé l'application

Comme indiqué dans la table 3, à partir de 15 utilisateurs qui ont testé l'application, la moyenne d'évaluation est supérieure à 0,6, et la précision était majoritairement égale à 1, et la précision moyenne était égale à 0,9. C'est considéré comme un bon résultat.

Nous pouvons conclure que le système a recommandé avec succès des endroits appropriés aux utilisateurs en fonction de leur intérêt.

4.5 Conclusion :

Dans ce chapitre nous avons présenté les différents outils utilisés pour le développement, les étapes et méthodes d'implémentation, les différentes interfaces créées et les résultats d'évaluation acquis après les tests.

L'objectif principal de ce chapitre est d'évaluer la qualité de notre système de recommandation proposé afin de vérifier s'il a des défauts et des vulnérabilités afin que nous puissions les corriger et améliorer le système à un meilleur état.

Conclusion générale

Au cours de la dernière décennie, les systèmes de recommandation intelligents fournissant des services publics ont été appliqués dans tous les domaines, entraînant un changement significatif du cadre de vie.

Dans ce mémoire, nous nous intéressons au domaine de la recommandation de lieux en fonction des préférences des différents utilisateurs, nous avons d'abord présenté les systèmes de recommandations à deux dimensions qui se basent sur le contenu, le filtrage collaboratif ou une hybridation entre les deux, puis nous avons exploité leurs algorithmes et approches pour choisir l'algorithme qui convient le mieux à nos objectifs

En effet, dans ce projet nous avons élaboré un système de recommandation qui peut recommander des lieux aux utilisateurs, aussi différents soient-ils. Des paramètres contextuels ont été pris en considération, à savoir la catégorie préférée des utilisateurs, le score d'analyse des sentiments, les mots-clés trouvés dans les commentaires des utilisateurs, le nombre de likes et de vues de chaque élément, qui nous a permis de mettre en place des divers algorithmes de recommandations, par exemple :

- La recommandation des plus populaires
- La recommandation des plus aimées
- La recommandation par similarité

Les résultats que nous avons obtenus de ce travail sont encourageants, cependant, aucune application n'est parfaite. Cela ouvre la porte à d'autres améliorations telles que l'ajout d'une autre dimension comme le temps, les algorithmes basés sur la localisation, le filtrage de l'âge des utilisateurs et l'utilisation de l'hybridation

Nous cherchons toujours à améliorer notre application pour garantir une meilleure expérience utilisateur tout dans le cadre du tourisme intelligent.

Bibliographie

- [1] Elsa Negre. « *Les systèmes de recommandation : une catégorisation* », 2018.
- [2] Lee R. Dice, « *Measures of the Amount of Ecologic Association Between Species* »
- [3] Akiko Aizawa. *An information-theoretic perspective of tf-idf measures*.
- [4] PABLO SÁNCHEZ, ALEJANDRO BELLOGÍN. *Point-of-Interest Recommender Systems based on Location-Based Social Networks: A Survey from an Experimental Perspective*.
- [5] David M. Blei, Andrew Y. Ng et Michael I. Jordan, « *Latent Dirichlet allocation* », *Journal of machine Learning research*, John Lafferty, vol. 3, 2003, p. 993-1022
- [6] Thomas K. Landauer, Peter W. Foltz et Darrell Laham, « *Introduction to Latent Semantic Analysis* », *Discourse Processes*, vol. 25, 1998
- [7] J. Ben Schafer, Joseph Konstan, John Riedl. *Recommender Systems in E-Commerce*
- [8] Don Peppers and Martha Rogers. *The One-to-One Future: Building Relationships One Customer at a Time*. Bantam Doubleday Dell Publishing, 1997.
- [9] Aït-Mokhtar et al, Dini et al, Basili et al *les travaux sur l'analyse syntaxique et sémantique* de 1999, 2001, 2002, 2007.

- [10] Sigrid Maurel, Paolo Curtoni, Luca Dini. *L'analyse des sentiments dans les forums*, (2014).
- [11] Xiaoyao ZHENG et al. « *A tourism destination recommender system using users' sentiment and temporal dynamics* ». In : *Journal of Intelligent Information Systems* 51.3 (2018), p. 557-578.
- [12] Zahra ABBASI-MOUD, Hamed VAHDAT-NEJAD et Javad SADRI. « *Tourism recommendation system based on semantic clustering and sentiment analysis* ». In: *Expert Systems with Applications* 167 (2021), p. 114324.
- [13] Norma Saiph Savage, Maciej Baranski, Norma Elva Chavez, and Tobias Höllerer. I'm feeling loco: A location-based context aware recommendation system. In *Advances in Location-Based Services*, Springer, 2012, p 37–54.
- [14] Daniele Quercia, Neal Lathia, Francesco Calabrese, Giusy Di Lorenzo, and Jon Crowcroft. *Recommending social events from mobile phone location data*. In 2010 IEEE international conference on data mining, pages 971–976. IEEE, 2010.
- [15] Anant Gupta and Kuldeep Singh. *Location based personalized restaurant recommendation system for mobile environments*. In 2013 International Conference on Advances in Computing, Communications, and Informatics (ICACCI), pages 507–511. IEEE, 2013.
- [16] Shihab Elbagir and Jing Yang, *Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment*. IMECS 2019, March 13-15, 2019, Hong Kong.
- [17] Clayton HUTTO et Eric GILBERT. « *Vader: A parsimonious rule-based model for sentiment analysis of social media text* ». In: *Proceedings of the international AAAI conference on web and social media*. T. 8. 1. 2014, p. 216-225.

- [18] https://pub.dev/documentation/sentiment_dart/latest/
- [19] Finn Årup Nielsen A new ANEW: *Evaluation of a word list for sentiment analysis in microblogs*. Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': *Big things come in small packages* 718 in CEUR Workshop Proceedings 93-98. 2011 May.
- [20] Thorvald Sørensen, « *A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons* », Biologiske Skrifter/Kongelige Danske Videnskabernes Selskab, vol. 5, no 4, 1948, p. 1–34
- [21] "Dart, a new programming language for structured web programming", GOTO conference (presentation) (opening keynote), Århus conference, 2011-10-10
- [22] Ladd, Seth. "What is Dart". *What is Dart?.* O'Reilly. Retrieved August 16, 2014.
- [23] «*With Flutter, Google Aims Dart to Mobile App Cross-Development* ». InfoQ. Retrieved 2022-03-17.
- [24] "Google announces Flutter 1.0, the first stable release of its cross-platform mobile development toolkit". Android Police. 2018-12-05. Retrieved 2022-03-17.
- [25] Lardinois, Frederic. "Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows". TechCrunch. Retrieved April 15, 2018, (April 29, 2015).
- [26] « *Google annonce un nouvel EDI pour Android : Android Studio au Google I/O, il s'appuiera sur IntelliJ* » [archive] (consulté le 23 juin 2015).

- [27] « *La précision, le rappel et de la matrice de confusion dans l'apprentissage automatique* », sur Bigdata.ma (version du 16 mai 2021 sur Internet Archive).
- [28] « *Content based recommended* ». (s.d.). Récupéré sur kdnuggets : <https://www.kdnuggets.com/2019/11/content-based-recommender-using-natural-language-processing-nlp.html> »
- [29] Virginie Zampa. « *Utilisation de l'analyse sémantique latente pour tenter d'optimiser l'acquisition par exposition à une langue étrangère de spécialité* » spécial Atala VOL. 8, N° 2 | 2005 p. 135-146