



République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et d'Informatique

Domaine Mathématiques - Informatique (MI)
Filière Informatique

Mémoire de Licence

Ingénierie des Systèmes d'Information et des Logiciels (ISIL)

Thème

**Conception et réalisation d'un système de recommandation
de restaurants.**

Encadré par :

Mme BENSAOU Nacera

Co-encadré par :

M.N. Djouada

Présenté par :

Baroud Yasmine

Hammouda Khadidja

Devant le jury composé de :

M Y. HAMMAL

M M. HAMDAR

Binôme n° : ISIL 019/21

Remerciements

Ce mémoire est le fruit d'un travail réalisé dans le cadre de la préparation du diplôme de licence en « Ingénierie des systèmes d'information et des logiciels » à la Faculté d'électronique et d'informatique de l'université des sciences et de la technologie Houari Boumediene. Nous adressons nos sincères remerciements à :

En premier temps : Allah, le tout puissant, pour toute la volonté, la patience et le courage qu'il nous a donné, afin de pouvoir réaliser ce projet. Aux différents membres du jury que nous remercions de nous avoir honoré en acceptant d'examiner et juger notre travail. Veuillez trouver ici l'expression de notre reconnaissance et de notre respect les plus sincères.

Notre promotrice Mme Bensaou Nacera, pour nous avoir soutenu tout au long de la durée du projet. Nous la remercions aussi pour son orientation et surtout ses judicieux conseils.

Notre Co-encadreur Mr Djouada Moussa, qui s'est toujours montré à l'écoute et a toujours été disponible tout au long de la réalisation de ce mémoire, ainsi que les conseils qu'il nous a donné durant cette expérience enrichissante.

Nous tenons à remercier tous les professeurs et enseignants de l'Université des Sciences et de la Technologie Houari Boumediene qui ont permis d'assurer notre formation jusqu'à ce point.

Finalement nous remercions nos parents pour leur précieux soutien moral durant nos études. Ils nous ont toujours encouragé à réaliser nos rêves et à nous surpasser c'est pour ça que nous leur exprimons nos profond vœux de gratitude .

Dédicace

Avec mes merveilleux et sincères sentiments,
Je dédie ce modeste travail à mes très chers parents qui n'ont jamais cessé de m'encourager et de me motiver. Je vous dis merci et que Allah vous garde.

Mes deux frères et ma soeur. Je vous souhaite un avenir radieux plein de réussite .

Ma binome "Hammouda Khadidja" et tous mes amis qui ont su m'apporter un soutien moral et conseil tout au long de ce travail.

Yasmine

Avec l'expression de ma reconnaissance,
Je dédie ce modeste travail à mes chère parents, à qui je leur exprime mon amour le plus s'insère.

l'homme, mon précieux père,
à qui je dois ma vie, ma réussite et que je lui loue mon plus grand respect : mon père "Kamel".

À la femme qui a souffert sans me laisser souffrir,
qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse :
mon adorable mère.

À mes frères et ma soeur , ma grand-mère, mes oncles et mes tantes. Que Dieu leur donne une
longue et joyeuse vie.

À tous les cousins, les voisins et les amis que j'ai connu jusqu'à maintenant.

Merci pour leur amour et leurs encouragements.
Sans oublier ma binôme "Baroud Yasmine" pour son soutien moral, sa patience et sa
compréhension tout au long de ce projet.

Khadidja

Table des figures

Liste des tableaux

1	État de l'art	2
1.1	Les systèmes de recommandation	2
1.1.1	Histoire	2
1.1.2	Définition	3
1.1.3	Domaines d'application des systèmes de recommandation	3
1.1.4	Différentes approches des systèmes de recommandation	4
1.1.4.1	Filtrage Collaboratif	5
1.1.4.2	Filtrage Basé Contenu	6
1.1.4.3	Filtrage Hybride	8
1.2	Apprentissage automatique	9
1.2.1	Définition	9
1.2.2	Apprentissage supervisé	9
1.2.3	Apprentissage non supervisé	9
1.2.4	Apprentissage par renforcement	10
1.2.5	Apprentissage en profondeur	10
1.2.6	Réseaux de neurones	10
1.2.7	Traitement automatique du langage naturel	11
1.3	Conclusion	11
2	Conception	12
2.1	Conception du système	12
2.1.1	L'architecture du système de recommandation	12
2.1.2	Dataset	13
2.1.2.1	Récolte de données	13
2.1.2.2	Traitement de données	16
2.1.3	L'approche du système de recommandation	16
2.1.3.1	Création des profils d'utilisateur et d'item	16
2.1.3.2	La mesure TF-IDF	17
2.1.3.3	Calcul de la similarité	18

2.1.3.4	Filtrage par localistaion	19
2.2	Conception de l'application	20
2.2.1	Spécification des besoins	21
2.2.1.1	Besoins fonctionnels	21
2.2.1.2	Besoins non-fonctionnels	21
2.2.2	Diagramme de cas d'utilisation	21
2.2.3	Diagramme de classes	23
2.2.4	Modèle relationnel du système	24
2.2.5	Diagramme de séquence	24
2.3	Conclusion	26
3	Implémentation et Résultats	27
3.1	Les outils de programmation	27
3.2	Les langages de programmation	28
3.3	Implémentation du système de collecte de données	29
3.3.1	La collecte de données	29
3.3.1.1	Statistiques	31
3.4	Implémentation du système de recommandation	32
3.4.1	Représentation des résultats du système	32
3.4.2	Evaluation du Filtrage basé contenu	32
3.5	Implémentation de l'application	33
3.5.1	Les interfaces de l'application	33
3.6	Conclusion	36
	Bibliographie	39

TABLE DES FIGURES

1.1	Le système de recommandation vu comme une boîte noire	3
1.2	Un système de recommandation collaboratif	5
1.3	Interface de recommandation de site Amazon.com	6
1.4	Le système de recommandation hybride	9
1.5	Exemple sur un réseau de neurone	11
2.1	Schéma descriptif de l'architecture du système de recommandation des restaurants	13
2.2	Représentation graphique de données dans le dataset Restaurant	14
2.3	Interface du site TripAdvisor	15
2.4	Représentation graphique de données dans le dataset utilisateur	15
2.5	Profil d'utilisateur	16
2.6	Profil d'item	17
2.7	Appliaction du TF-Idf sur le profil d'utilisateur	18
2.8	Appliaction du TF-Idf sur le profil de restaurant	18
2.9	Résultat de recommandation pour un utilisateur	20
2.10	Diagramme de cas d'utilisation du client	22
2.11	Diagramme de cas d'utilisation du restaurateur	22
2.12	Diagramme de classes	23
2.13	Diagramme de séquence de connexion	25
2.14	Diagramme de séquence de recommandation	26
3.1	Exemple du code utilisé pour l'extraction des données	30
3.2	Exemple du résultat obtenu en utilisant le web scraping sur TripAdvisor	31
3.3	Exemple d'un résultat du recommandation des restaurants pour un utilisateur	32
3.4	Interface de connexion et d'inscription de RestoFinder	33
3.5	L'interface des choix de préférences	34
3.6	L'interface de la liste de recommandations et l'affichage des informations du restaurant	34
3.7	L'interface de gestion de compte du client	35
3.8	L'interface de la liste des restaurants ajoutés et l'ajout.	36

LISTE DES TABLEAUX

2.1	Score obtenu par l'application du cosinus similarité sur l'exemple	19
3.1	Statistiques de la base de données restaurant	31
3.2	Statistiques de la base de données utilisateur	31
3.3	Les données de l'utilisateur X	32

Introduction générale

L'exploration de données est un processus de recherche d'informations pertinentes, au sein d'un ensemble de données, destiné à détecter des corrélations cachées ou des informations nouvelles. Or, les utilisateurs doivent faire face à un volume toujours plus important d'informations en raison de l'accroissement des capacités de calcul et de stockage. De sorte qu'il est de plus en plus difficile de savoir exactement quelles informations rechercher et ou les chercher.

Le processus de recommandation va guider l'utilisateur lors de son exploration de la quantité d'informations à sa disposition en cherchant pour lui, les informations qui paraissent pertinentes [16]. Il s'agit d'une forme particulière de filtrage de l'information visant à présenter les éléments d'informations (films, musiques, livres, articles, images, produits, plats de cuisine, etc) qui sont susceptibles d'intéresser l'utilisateur.

Ces dernières années, avec le développement des technologies de communication sans fil et des appareils mobiles, les gens peuvent obtenir toutes les informations qu'ils veulent n'importe quand et en utilisant leur appareils mobiles. Le milieu de la restauration connaît un grand essor ou l'agrandissement du nombre de restaurant, ainsi que leurs offres variées représente une opportunité de développement économique et un moyen de promotion du tourisme dans notre pays l'Algérie.

Nous proposons dans ce projet une application mobile qui utilise un système de recommandation pour des restaurants algériens en appliquant le filtrage basé sur le contenu et ce afin de satisfaire les besoins des clients potentiels et d'augmenter les chances et la visibilité des restaurants ainsi que les services qu'ils proposent. Nous travaillons avec des données réelles. Nous proposons que le filtrage des données soit basé sur les préférences des utilisateurs par rapport à plusieurs critères comme le type de cuisine, la qualité de service, les prix et nous utilisons les informations de localisation de l'utilisateur et des restaurants pour générer dynamiquement les résultats de la recommandation. Ceci est basé sur l'hypothèse que les gens ont souvent tendance à fréquenter les restaurants les plus proches. Toutes ces critères contribuent à accroître la popularité d'un lieu par rapport à un autre.

En résumé, ce mémoire est organisé en trois chapitres :

Le premier est une présentation générale sur les systèmes de recommandation en commençant par leur histoire, leur définition et leur domaines d'application, puis en présentent leurs différentes approches, ainsi que les concepts de base de l'apprentissage automatique.

Le deuxième, il se devise en deux partie : Une description du processus de conception de notre solution pour le système de recommandation et la conception de notre application RestoFinder pour des restaurants algériens.

Le troisième est une présentation des données collectées, le résultat et l'évaluation du système de recommandation déployé dans l'application RestoFinder, ainsi que la réalisation de l'application avec ces différentes interfaces.

Enfin, ce mémoire se conclut par une conclusion générale dans laquelle nous abordons quelques perspectives de développement et les possibles améliorations de ce projet.

Un système de recommandation est une sous-catégorie d'un système d'extraction d'informations, qui aide à trouver le classement ou la préférence de l'utilisateur pour un item. Les systèmes de recommandation sont des méthodes dédiées qui donnent des idées sur des éléments utilisés par différents utilisateurs. De nombreuses décisions peuvent être prises en considérant les recommandations comme des produits à acheter, des restaurants pour manger, les types de musique à écouter, etc [12]. Nous allons traiter dans ce chapitre la notion du système de recommandation. Nous présenterons l'histoire, la définition, les domaines d'application des systèmes de recommandation et leur différents types, ainsi que les notions de l'apprentissage automatique nécessaires dans la recommandation automatique.

1.1 Les systèmes de recommandation

1.1.1 Histoire

Les racines des systèmes de recommandation remontent aux travaux étendus dans les sciences cognitives, la théorie d'approximation, la recherche documentaire, la théorie de la prévoyance et ont également des liens avec la science de la gestion et le marketing, dans la modélisation des choix du consommateur [15]. Grundy [23], un système bibliothécaire, qui a ouvert la voie aux systèmes de recommandation. Ce système construit des modèles de ses utilisateurs, à l'aide de stéréotypes, puis exploite ces modèles pour le guider dans sa tâche, suggérant des romans que les gens peuvent trouver intéressants. Mais l'utilisation de ce système reste limitée. A l'apparition du filtrage collaboratif en 1990, plusieurs systèmes de recommandation ont vu le jour. L'année 1992 voit l'apparition de système de recommandation Tapestry. Tapestry est un système expérimental qui gère un flux entrant de documents électroniques, y compris le courrier électronique, les reportages et les articles NetNews. Tapestry utilise le filtrage collaboratif manuel pour gérer de gros volumes de documents entrants. Les utilisateurs de ce système sont encouragés à annoter les documents, et ces annotations peuvent ensuite être utilisées pour le filtrage. Le filtrage collaboratif automatique apparait ensuite en 1994 avec GroupLens, pour sa part, est un système de recommandation automatique de la filtration collaborative basée sur l'évaluation des utilisateurs. Il est utilisé pour recommander les films et les nouvelles. Actuellement, avec le développement rapide du domaine informatique, les systèmes de recommandation sont devenus une partie intégrante

de plusieurs sites électroniques.

1.1.2 Définition

Un système de recommandation calcule et fournit un contenu pertinent à l'utilisateur en fonction de la connaissance de ce dernier, du contenu et des interactions entre l'utilisateur et l'élément [16] (voir figure 1.1).

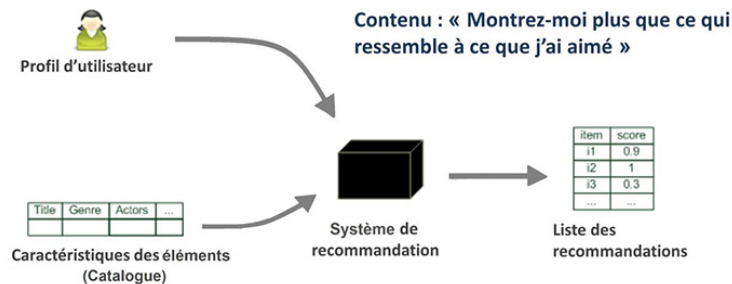


FIGURE 1.1 – Le système de recommandation vu comme une boîte noire

Généralement, un système de recommandation permet de comparer le profil d'un utilisateur à certaines caractéristiques de référence, et cherche à prédire l'« avis » que donnerait un utilisateur. Ces caractéristiques peuvent provenir de :

- L'objet lui-même, on parle « d'approche basée sur le contenu » ou content-based approach.
- L'utilisateur.
- L'environnement social, on parle d'approche de filtrage collaboratif ou collaborative filtering.

1.1.3 Domaines d'application des systèmes de recommandation

Système de recommandation de e-commerce

Les systèmes de recommandation sont très utilisés dans le domaine du e-commerce, où ils utilisent comme entrée les intérêts d'un client pour générer une liste des produits à recommander. Amazon.com utilise des algorithmes de recommandation pour personnaliser la boutique en ligne pour chaque client [16].

En introduisant des algorithmes sophistiqués, les boutiques en ligne devraient voir leurs ventes augmenter avec les systèmes de recommandation actuels. Le site le plus connu qui utilise ce système est Amazon.com.

Système de recommandation des films

Le système de Netflix propose des recommandations personnalisées afin d'aider les utilisateurs à découvrir des séries TV et des films susceptibles de les intéresser. Plus de 80% des contenus regardés sur Netflix sont découverts via le moteur de recommandation. L'importance de cet algorithme est donc primordial dans le modèle. Il repose sur trois éléments : les abonnés, l'indexation des contenus et un algorithme de l'apprentissage automatique.

Système de recommandation de musique

L'idée est simple : proposer une playlist personnalisée, adaptée au client et à ses préférences musicales. Ainsi, les goûts sont ciblés, l'offre est différente pour chacun. Cette personnalisation a pour but de fidéliser les clients de ces plateformes, en les séduisant avec des suggestions pertinentes pour eux. On peut citer parmi les services de lecture de musique qui utilisent les systèmes de recommandation : Spotify. L'algorithme de recommandation de ce service se base sur plusieurs critères pour établir un profil, tel que les chansons écoutées et les artistes, les styles de musique correspondants, l'acoustique, le tempo et la « dansabilité » des chansons sont prises en compte. Ces données sont comparées à celles des autres utilisateurs et donne un profil organisé par regroupement de genres musicaux les plus appréciés.

Système de recommandation des livres

Google Books et le système de recommandation de amazon.com ont été développés dans le cadre d'un site Web commercial, mais chacun d'eux génère les recommandations selon des approches bien différentes. Le premier emploie celle du filtrage basé sur le contenu, alors que le deuxième applique le filtrage collaboratif.

Système de recommandation des restaurants

Les systèmes de recommandation sont largement utilisés dans le domaine de la restauration, en raison de l'efficacité de ces systèmes pour augmenter les profits et aider les utilisateurs à trouver des restaurants qui correspondent à leurs besoins. Plusieurs plates-formes utilisent ces systèmes, dont les plus populaires sont :

Zomato : c'est une application de recherche et découverte de restaurants en ligne et mobile, qui fournit des informations fiables sur plus de 1,4 million de restaurants dans 23 pays. Zomato est utilisée par des consommateurs qui peuvent découvrir, tester et noter des restaurants mais aussi créer leur propre réseau pour des recommandations de confiance.

TripAdvisor : TripAdvisor est le site web le plus populaire dans la recommandation des restaurants dans le monde. Les recommandations en TripAdvisor se base principalement sur les notes et les commentaires recueillis par les utilisateurs.

1.1.4 Différentes approches des systèmes de recommandation

Les systèmes de recommandation peuvent être classés de manière générale en cinq catégories selon les informations qu'ils utilisent pour recommander des items. Ces cinq catégories sont les suivantes :

- Les systèmes de filtrage basés sur le contenu essaient de recommander des items qui correspondent au profil de l'utilisateur.
- Les systèmes basés sur le filtrage collaboratif produisent des recommandations en calculant la similarité entre les préférences d'un utilisateur et celles d'autres utilisateurs
- Les systèmes de filtrage démographique utilisent des informations démographiques telles que l'âge, le sexe, l'éducation, etc.

- Les systèmes basés sur la connaissance utilisent la fonction des connaissances pour générer des recommandations, c'est-à-dire des connaissances sur la façon dont un item particulier rencontre un utilisateur particulier et peut raisonner sur la relation entre un besoin et un produit.
- Les systèmes de recommandation hybrides utilisent à la fois des connaissances extérieures et les caractéristiques des éléments, combinant ainsi des approches collaboratifs et basés contenu.

Nous présentons plus en détail dans la suite l'approche basée sur le filtrage collaboratif et l'approche basée sur le contenu, et enfin l'approche hybride.

1.1.4.1 Filtrage Collaboratif

Le filtrage collaboratif (FC) exploite le pouvoir de la communauté pour fournir des recommandations. Il consiste à estimer l'appréciation d'un utilisateur par rapport à un item en se basant sur des évaluations (notes) et des attitudes passées. Les systèmes de filtrage collaboratifs sont l'un des modèles de recommandation les plus populaires utilisés dans l'industrie et ont rencontré un énorme succès pour des entreprises comme Amazon [1].

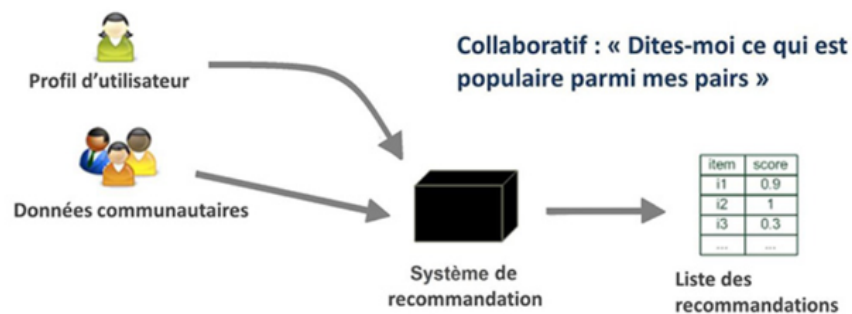


FIGURE 1.2 – Un système de recommandation collaboratif

On distingue deux grandes approches de filtrage collaboratif, à savoir : FC basé mémoire et FC basé modèle.

a- Filtrage collaboratif basé mémoire

Ce filtrage utilise l'ensemble de la base de données des évaluations des utilisateurs pour faire les prédictions, les évaluations de l'utilisateur actif sont prédites à partir d'informations d'utilisateur actif, et à partir de la base de données des évaluations des utilisateurs similaires.

b- Filtrage collaboratif basé modèle

Les algorithmes basés sur le modèle se basent sur les évaluations précédentes (les profils) des utilisateurs, sauf qu'on ne calcule pas directement les prédictions, mais on essaie de classer les utilisateurs suivant des groupes ou d'apprendre les modèles à partir de leurs données. Selon le FC basé modèle on distingue deux type :

Filtrage collaboratif basé sur l'utilisateur (user based) :

L'idée principale derrière le filtrage basé sur les utilisateurs est que si nous sommes en mesure de trouver des utilisateurs qui ont acheté et aimé des articles similaires dans le passé, ils sont plus susceptibles d'acheter des articles similaires dans le futur aussi. Par conséquent, ces modèles recommandent des articles à un utilisateur qui ont aimé par les utilisateurs similaires à l'utilisateur courant [1].

Filtrage basé sur les éléments (item based) :

Le FC basé sur les éléments est une branche de la recommandation qui prend en compte les informations sur les différents utilisateurs. Le mot « collaboratif » fait référence au fait que les utilisateurs collaborent entre eux pour recommander des éléments. En effet, les algorithmes prennent en compte les achats et les préférences des utilisateurs. Le point de départ est une matrice de notation dans laquelle les lignes correspondent aux utilisateurs et les colonnes aux éléments. Amazon a fait un bon usage de ce modèle en recommandant des produits en fonction de navigation et historique des achats des utilisateurs [1], comme illustré dans la figure suivante :

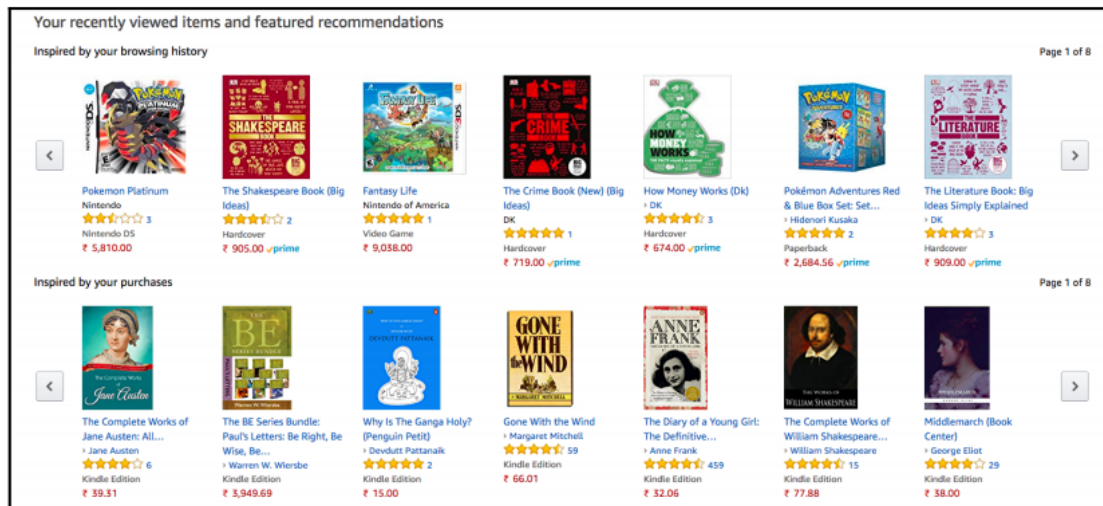


FIGURE 1.3 – Interface de recommandation de site Amazon.com

1.1.4.2 Filtrage Basé Contenu

Ce type de recommandation peut être considéré comme un système de recherche d'information exploitant le profil d'utilisateur. Ce type de système émet des recommandations en analysant la description des articles qui ont été évalués par l'utilisateur et la description des articles destinés à être recommandés. Cette technique est fondée sur l'analyse des similarités de contenu entre les items précédemment consultés par les utilisateurs et ceux qui n'ont pas été encore consultés [22]. Un système de recommandation basé sur le contenu fonctionne avec les données que l'utilisateur fournit, soit explicitement ou implicitement. Sur la base de ces données, un profil utilisateur est généré, qui est ensuite utilisé pour faire des suggestions à l'utilisateur. Par exemple, on construit un profil d'un utilisateur qui préfère des séries de genres actions et romance et on essaie de recommander des produits qui sont de la même section qu'il préfère.

Profil utilisateur : Dans le profil utilisateur, nous créons des vecteurs qui décrivent les préférences de l'utilisateur. Dans la création d'un profil d'utilisateur, nous utilisons la

matrice d'utilité qui décrit la relation entre l'utilisateur et l'article. Avec ces informations, la meilleure estimation que nous pouvons faire concernant l'item que l'utilisateur aime est une agrégation des profils de ces items.

Profil item : Dans les systèmes de recommandation basé sur le contenu, nous devons créer un profil pour chaque item, qui représentera les caractéristiques importantes de cet item. Par exemple, si nous créons un film en tant qu'item, ses acteurs, son réalisateur, l'année de sortie et le genre sont les caractéristiques les plus importantes du film. Nous pouvons également ajouter sa note à partir de l'IMDB (Internet Movie Database) dans le profil de l'objet.

Matrice d'utilité : La matrice d'utilité indique la préférence de l'utilisateur avec certains items. Dans les données recueillies auprès de l'utilisateur, nous devons trouver une relation entre les items qui sont appréciés par l'utilisateur et ceux qui ne le sont pas, à cette fin, nous utilisons la matrice d'utilité. Nous y attribuons une valeur particulière à chaque paire d'item utilisateur, cette valeur est connue sous le nom de degré de préférence. Ensuite, nous dessinons une matrice d'un utilisateur avec les éléments respectifs pour identifier leur relation de préférence.

Recommandation basé sur les vecteurs de mots-clés :

TF-IDF est une combinaison de deux mots différents, à savoir fréquence des termes et fréquence inverse des documents.

Tout d'abord, le terme "fréquence des termes" sera discuté. TF est utilisé pour mesurer le nombre de fois qu'un terme est présent dans un document. Supposons que nous ayons un document "T1" contenant 5000 mots et que le mot "Alpha" soit présent dans le document exactement 10 fois. Il est bien connu que la longueur totale des documents peut varier de très petite à grande, il est donc possible qu'un terme apparaisse plus fréquemment dans les grands documents que dans les petits. Pour remédier à ce problème, l'occurrence d'un terme dans un document est divisée par le nombre total de termes présents dans ce document, afin de trouver la fréquence des termes.

On comprend que l'occurrence la plus grande ou plus élevée d'un mot dans les documents donnera une fréquence de terme plus élevée et que l'occurrence moindre d'un mot dans les documents donnera une importance plus grande (IDF) pour ce mot clé recherché dans un document particulier. TF-IDF n'est rien d'autre que la multiplication de la fréquence des termes (TF) et de la fréquence inverse des documents (IDF) [21].

Recommandation basé sur l'analyse sémantique à l'aide d'ontologies :

L'analyse sémantique permet d'apprendre des profils plus précis contenant des références à des concepts définis dans des bases de connaissances externes. La motivation principale de cette approche est le défi de fournir à un système de recommandation les connaissances de base culturelles et linguistiques qui caractérisent la capacité d'interpréter les documents en langage naturel et de raisonner sur leur contenu. La description de ces stratégies est réalisée en tenant compte de plusieurs critères :

- Le type de source de connaissances impliquée (ex : lexicque, ontologie, etc.)
- Les techniques adoptées pour l'annotation ou la représentation des items.
- Le type de contenu inclus dans le profil d'utilisateur.
- La stratégie d'appariement item-profil.

Les avantages de filtrage basé contenu

L'adoption du paradigme de recommandation basé sur le contenu présente plusieurs avantages par rapport au modèle collaboratif :

Idépendance d'utilisateur : Les systèmes de recommandation basés sur le contenu exploitent uniquement les évaluations fournies par l'utilisateur actif pour construire son propre profil. Au lieu de cela, les méthodes de filtrage collaboratifs ont besoin d'être notées par d'autres utilisateurs afin de trouver les « voisins les plus proches » de l'utilisateur actif, c'est-à-dire les utilisateurs qui ont des goûts similaires puisqu'ils ont évalué les mêmes éléments de la même manière. Ensuite, seuls les éléments les plus appréciés par les voisins de l'utilisateur actif seront recommandés.

Transparance : Des explications sur le fonctionnement du système de recommandation peuvent être fournies en répertoriant explicitement les fonctionnalités de contenu ou les descriptions qui ont provoqué l'apparition d'un item dans la liste des recommandations. Ces caractéristiques sont des indicateurs à consulter pour décider de faire confiance à une recommandation. À l'inverse, les systèmes collaboratifs sont des boîtes noires car la seule explication d'une recommandation d'article est que des utilisateurs inconnus ayant des goûts similaires ont aimé cet article.

Nouvel article : Les systèmes de recommandation basés sur le contenu peuvent recommander des items qui n'ont pas encore été évalués par un utilisateur. En conséquence, ils ne souffrent pas du problème du premier évaluateur, qui affecte les recommandateurs collaboratifs qui se basent uniquement sur les préférences des utilisateurs pour faire des recommandations. Par conséquent, tant que le nouvel item ne sera pas évalué par un nombre important d'utilisateurs, le système ne pourra pas le recommander.

Les inconvénients du filtrage basé contenu

Nouvel utilisateur : Il faut recueillir suffisamment de notes avant qu'un système de recommandation basé sur le contenu puisse vraiment comprendre les préférences de l'utilisateur et fournir des recommandations précises. Par conséquent, lorsque peu de notes sont disponibles, comme pour un nouvel utilisateur, le système ne pourra pas fournir de recommandations fiables.

1.1.4.3 Filtrage Hybride

Il ne s'agit pas d'une approche indépendante, mais plutôt de combiner plusieurs approches de recommandation. L'hybridation est une solution qui consiste à proposer des systèmes, qui tirent parti des avantages des deux approches ou plus. Cette solution permet de combler les lacunes de l'une des approches sur des cas d'utilisation précis [9].

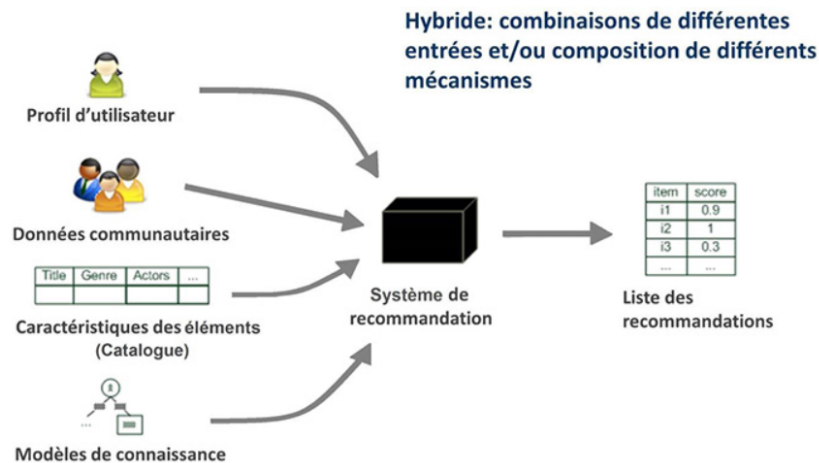


FIGURE 1.4 – Le système de recommandation hybride

1.2 Apprentissage automatique

1.2.1 Définition

L'apprentissage automatique (en anglais machine learning), également appelé apprentissage artificiel ou apprentissage machine, est un champ d'étude de l'intelligence artificielle (IA) qui permet à un système d'apprendre à partir des données et non à l'aide d'une programmation explicite. Cependant, l'apprentissage automatique n'est pas un processus simple. Au fur et à mesure que les algorithmes ingèrent les données de formation, il devient possible de créer des modèles plus précis basés sur ces données. Un modèle de machine learning est le résultat généré lorsque vous entraînez votre algorithme d'apprentissage automatique avec des données [14]. Lorsqu'un modèle a été formé à partir des données en entrée, vous recevez un résultat en sortie. Par exemple, un algorithme prédictif crée un modèle prédictif. Ensuite, lorsque vous fournissez des données au modèle prédictif, vous recevez une prévision qui est déterminée par les données qui ont servi à former le modèle [14].

1.2.2 Apprentissage supervisé

L'apprentissage supervisé commence généralement par un ensemble de données bien défini et une certaine compréhension de la façon dont ces données sont classifiées [13]. L'apprentissage supervisé a pour but de déceler des modèles au sein des données et de les appliquer à un processus analytique. Ces données comportent des caractéristiques associées à des libellés qui définissent leur signification. Par exemple, La création d'une application d'apprentissage automatique capable de faire la distinction entre plusieurs millions d'animaux, en se basant sur des descriptions écrites et des images. Ce type d'apprentissage peut traiter deux types de problèmes :

- La régression.
- La classification.

1.2.3 Apprentissage non supervisé

L'apprentissage non supervisé est utilisé lorsque le problème nécessite une quantité massive de données non étiquetées. Pour comprendre le sens de ces données, il est nécessaire d'utiliser des

algorithmes qui classifient les données en fonction des tendances ou des clusters qu'ils décèlent. L'apprentissage non supervisé mène un processus itératif, analysant les données sans intervention humaine. Il est utilisé avec la technologie de détection de spam envoyé par e-mail. Les e-mails normaux et les spams comportent un nombre de variables beaucoup trop élevé pour qu'un analyste puisse étiqueter les e-mails indésirables envoyés en masse. En revanche, les discriminants d'apprentissage automatique, basés sur la mise en cluster et l'association, sont appliqués pour identifier les courriers électroniques non désirés [14]. Il existe plusieurs algorithmes d'apprentissage non supervisé dont les plus populaires sont K-means et K-medoids.

1.2.4 Apprentissage par renforcement

L'apprentissage par renforcement est un modèle d'apprentissage comportemental. Il diffère des autres types d'apprentissage supervisé car le système n'est pas formé avec un ensemble de données exemple. Au lieu de cela, le système apprend plutôt par le biais d'une méthode d'essais et d'erreurs. Par conséquent, une séquence de décisions fructueuses aboutit au renforcement du processus, car c'est lui qui résout le problème posé plus efficacement [14].

1.2.5 Apprentissage en profondeur

L'apprentissage en profondeur est une méthode spécifique d'apprentissage automatique qui intègre des réseaux neuronaux en couches successives afin d'apprendre des données de manière itérative. Il est particulièrement utile lorsque vous tentez de détecter des tendances à partir de données non structurées. Les réseaux neuronaux complexes d'apprentissage en profondeur sont conçus pour émuler le fonctionnement du cerveau humain, de sorte que les ordinateurs peuvent être entraînés pour faire face à des abstractions et des problèmes mal définis. Les réseaux de neurones et l'apprentissage en profondeur sont souvent utilisés dans les applications de reconnaissance d'image, de vision numérique, de communication orale et de recommandations personnalisées [14].

1.2.6 Réseaux de neurones

Les réseaux de neurones, également appelés réseaux de neurones artificiels (ANN) ou réseaux de neurones simulés (SNN), sont un sous-ensemble de l'apprentissage automatique et sont au cœur des algorithmes d'apprentissage en profondeur. Leur nom et leur structure sont inspirés du cerveau humain, imitant la façon dont les neurones biologiques se signalent les uns aux autres. Ils permettent aux programmes informatiques de reconnaître des modèles et de résoudre des problèmes courants dans les domaines de l'IA, de l'apprentissage automatique et de l'apprentissage en profondeur [17]. Les réseaux de neurones artificiels (ANN) sont constitués d'une couche de nœuds, contenant une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque nœud, ou neurone artificiel, se connecte à un autre et possède un poids et un seuil associés. Si la sortie d'un nœud individuel est supérieure à la valeur seuil spécifiée, ce nœud est activé, envoyant des données à la couche suivante du réseau. Sinon, aucune donnée n'est transmise à la couche suivante du réseau [8].

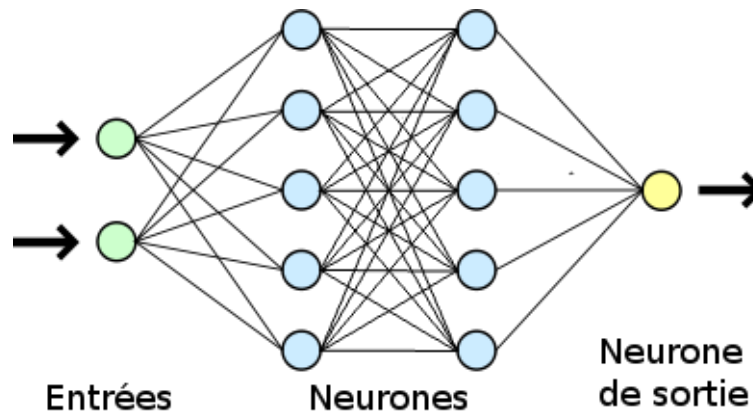


FIGURE 1.5 – Exemple sur un réseau de neurone

1.2.7 Traitement automatique du langage naturel

Le traitement automatique du langage naturel est une discipline s'appliquant au domaine de l'informatique et du langage. Il est utilisé par exemple pour les traductions, la reconnaissance vocale ou encore les réponses automatiques aux questions. Ces domaines représentent des défis majeurs, car les mots du langage sont souvent traités un à un par l'ordinateur. Or, de nombreux mots sont polysémiques et recouvrent différentes réalités, en s'inscrivant notamment dans des contextes ou expressions qui peuvent changer complètement leur sens d'origine. Par Exemple :

- le verbe « rimer » ne va pas avoir la même signification entre « ne pas rimer » et « ne rimer à rien », tout comme le mot « film » ne va pas se référer à la pellicule photographique dans l'expression « ne rien comprendre au film ».

Le traitement du langage se base sur 2 aspects essentiels : l'analyse syntaxique et l'analyse sémantique.

1.3 Conclusion

Dans cette partie, nous avons défini les systèmes de recommandation, les domaines d'application et leurs différentes approches. Ainsi, nous avons introduit les notions de base de l'apprentissage automatique . Le prochain chapitre sera consacré à l'étude conceptuelle du système de recommandation et de notre application mobile RestoFinder.

Ce chapitre présente une étude conceptuelle menée pour le développement de notre système de recommandation et l'application RestoFinder. Dans ce chapitre, nous allons décrire deux grandes parties, la première partie consiste à proposer un système de recommandation pour les restaurants algériens selon les préférences par rapport aux types de cuisine et la localisation des utilisateurs en présentant les datasets créés et l'architecture générale du système, la seconde partie décrit la conception de l'application mobile.

2.1 Conception du système

2.1.1 L'architecture du système de recommandation

Cette section décrit notre système de recommandation basé sur le contenu et la localisation.

Notre système de recommandation consiste à recommander des restaurants en Algérie aux utilisateurs selon les cuisines préférées et la localisation.

La figure suivante illustre notre approche appliquée :

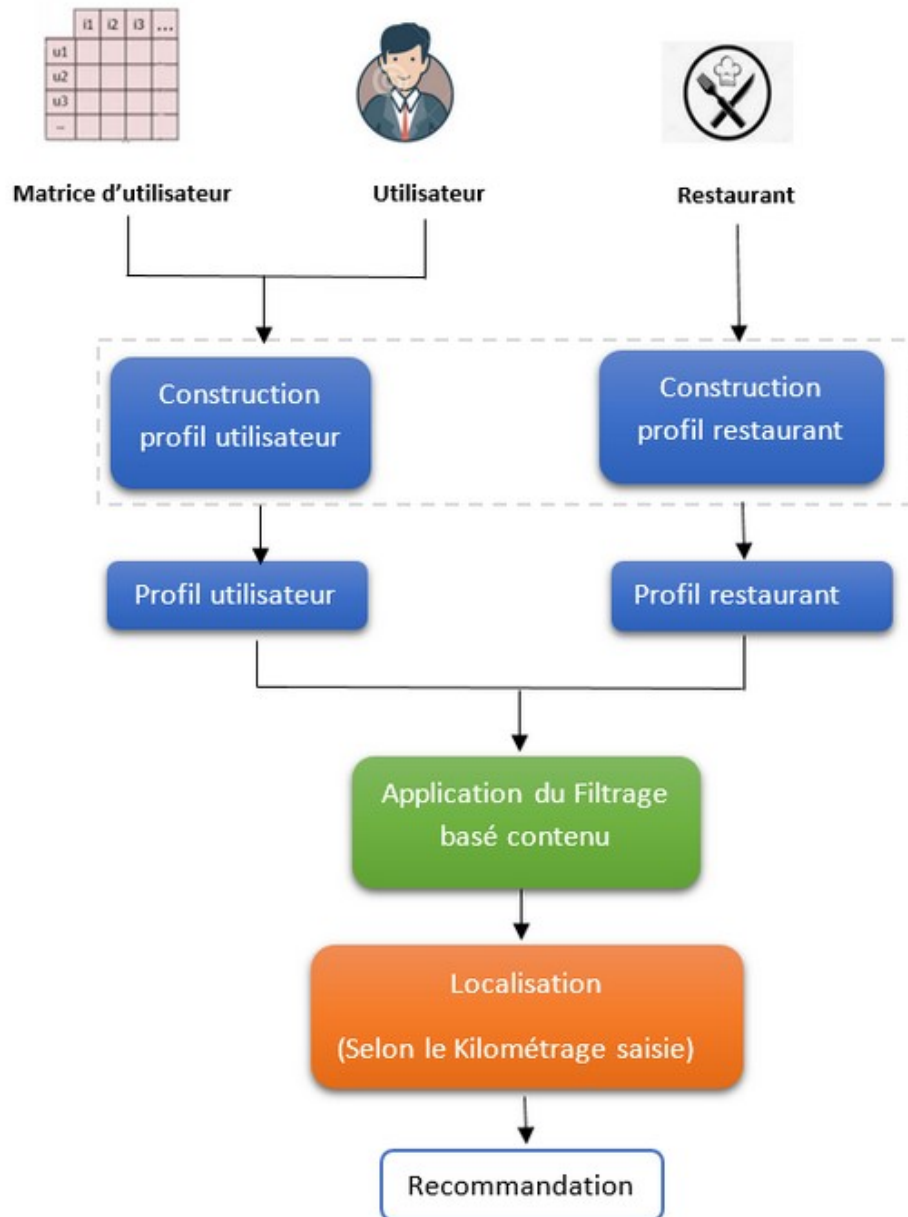


FIGURE 2.1 – Schéma descriptif de l'architecture du système de recommandation des restaurants

2.1.2 Dataset

2.1.2.1 Récolte de données

Il existe différents moyens pour la collecte de données : soit manuellement, ce qui peut nécessiter un temps considérable lorsqu'on cherche à disposer d'un volume important d'information, ou en utilisant des méthodes automatiques, via des logiciels permettant d'obtenir une quantité importante d'informations en un temps record. Parmi ces méthodes nous citons le web scraping.

a - dataset des restaurants

L'extraction de données des restaurants est réalisée à partir du site Web « Tripadvisor », nous avons créé un web scraper qui permet de scanner les restaurants algériens existants sur ce site. Lors du lancement du robot, ce dernier déclenche l'ouverture d'un browser dans le site tripadvisor¹, le scraper récupère une liste de tous les restaurants d'Algérie qui sont disponibles avec plusieurs données (nom, wilaya, numéro de téléphone, cuisines...). Le robot procède ainsi au filtrage de données en les structurant sous les formes adéquates à notre dataset. Elle contient les informations suivantes :

- ID : Identifiant du restaurant.
 - Nom : Nom du restaurant.
 - Wilaya : La wilaya dont laquelle se situe le restaurant.
 - Adresse : l'adresse du restaurant.
 - Num_Tél : Numéro de téléphone du restaurant.
 - Cuisines : (60 cuisines au total)
 - Rating : la note total de ce restaurant.
 - Qualité_Food : La qualité de food du restaurant.
 - Qualité_Service : La qualité de service du restaurant.
 - nbr_Reviews : Le nombre de personnes qui ont noté ce restaurant.
 - Latitude : Latitude du restaurant récupéré à partir du google map.
 - Longitude : Longitude du restaurant récupéré à partir du google map.
- La figure suivante illustre les informations incluses dans le dataset d'un restaurant.



FIGURE 2.2 – Représentation graphique de données dans le dataset Restaurant

1. https://www.tripadvisor.fr/Restaurants-g293718-Algiers_Algers_Province.html

Le site TripAdvisor

TripAdvisor est un site web américain qui offre des avis et des conseils touristiques émanant de consommateurs sur des hôtels, restaurants, villes et régions, lieux de loisirs, etc. Tripadvisor est la référence pour la visibilité des restaurants en ligne dans le monde entier. Des millions de personnes tous les jours utilisent Tripadvisor pour se renseigner, rechercher des restaurants de qualité.

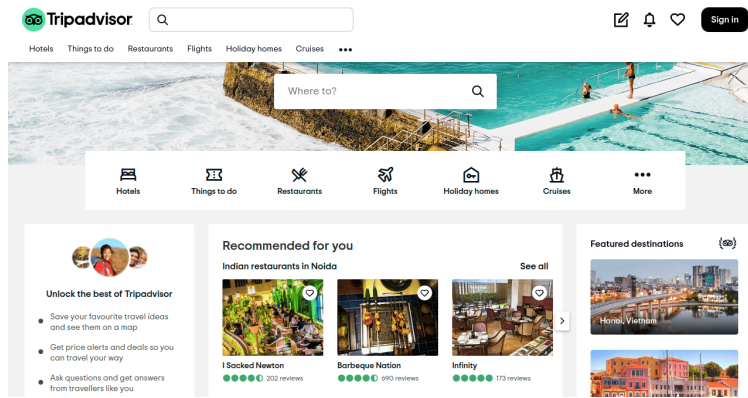


FIGURE 2.3 – Interface du site TripAdvisor

b - dataset des utilisateurs

À l'absence d'un dataset sur les profils utilisateurs adéquate aux types de cuisines préférés pour les restaurants en Algérie ainsi que leurs localisations, nous avons procédé à la collecte de données à partir de formulaire² que nous avons créé. Ce formulaire permet de récupérer les informations suivantes sur les utilisateurs : le nom, localisation(à partir de la localisation de chaque utilisateur, on a extrait latitude et longitude), les types de cuisines préférés. À partir de ces informations, nous avons initialisé notre base de profil.

La figure suivante illustre les informations incluses dans le dataset d'un utilisateur.

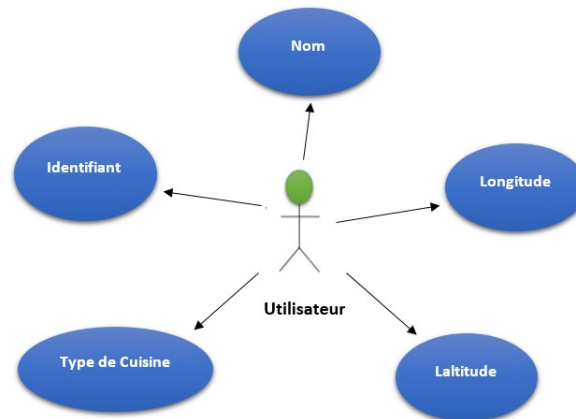


FIGURE 2.4 – Représentation graphique de données dans le dataset utilisateur

2. https://docs.google.com/forms/d/e/1FAIpQLScig3vMkXA9ngOFXiZZofzs72J1jCyZQ0_nJ_t4Vr51eV5g/viewform

2.1.2.2 Traitement de données

Les informations comme les types de cuisines nécessiteront des notions de Traitement Automatique du langage (NLP), Nous allons donc présenter les différents prétraitements appliqués à ces données :

Suppression des signes de ponctuations : Nous commençons notre traitement en éliminant les signes de ponctuations ".,;?!".

Suppression des mots vides et des Noms propres : Les mots vides (stop words) comme les prépositions(après, avant) ou les pronoms(le, cet, il, elle) seront retirés pour ne garder que l'essentiel de l'information.

Tokenization : cette étape consiste à vectoriser un corpus de texte, c'est à dire découper notre structure textuelle en un vecteur ou chaque case représentera un mot.

Lemmatisation et stemming : Le stemming est le processus de réduction d'un mot à son radicale via ses flexions ou à la racine des mots connus sous le nom de lemme.

2.1.3 L'approche du système de recommandation

2.1.3.1 Création des profils d'utilisateur et d'item

Les profils utilisateurs et d'items jouent un rôle important dans les processus de recommandation. La plupart des systèmes de personnalisation doivent créer un profil d'utilisateur ou un modèle de préférences d'utilisateur afin d'identifier les besoins de l'individu utilisateur. Ainsi, la qualité des informations disponibles en entrée sur l'utilisateur est une condition pour l'efficacité des systèmes de recommandation. Le profil utilisateur dans notre cas est alimenté par les préférences que l'utilisateur introduit explicitement (liste des intérêts). Le profil d'item doit contenir les caractéristiques du restaurant, ces critères sont également utilisés pour créer le profil de l'utilisateur, mais avant d'utiliser ces profils dans notre algorithme il faut d'abord appliquer des traitement de langage suivants :

- Supprimer la ponctuation.
- Enlever les majuscules.
- Eliminer les stop words.

Les figures suivantes illustrent les profils avant et après le traitement.

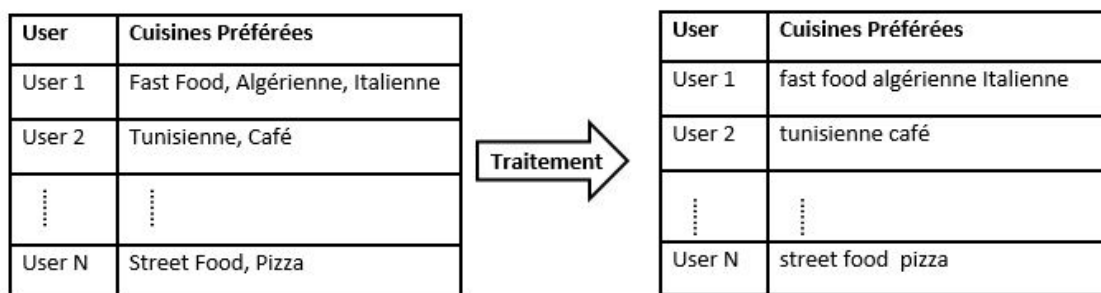


FIGURE 2.5 – Profil d'utilisateur

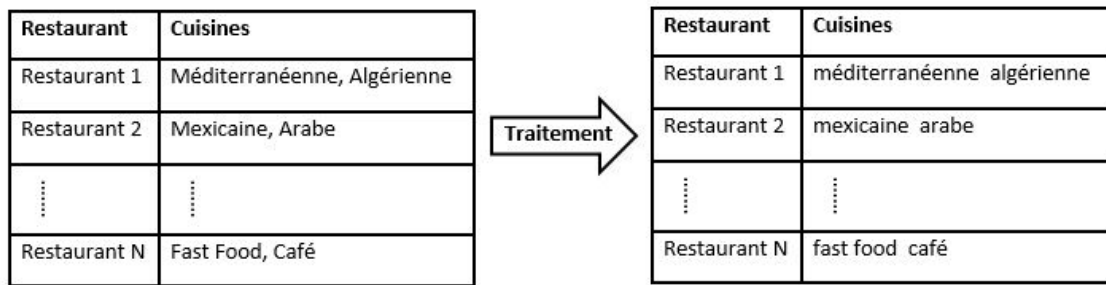


FIGURE 2.6 – Profil d’item

2.1.3.2 La mesure TF-IDF

Pour calculer le poids des termes des documents, nous avons choisi la méthode de pondération TF-IDF qui permet d’évaluer l’importance d’un terme contenu dans un document, relativement à une collection ou un corpus [28]. Dans notre cas, l’ensemble des termes est les types de cuisines existe dans les deux profils utilisateur et restaurant. Ces termes ont plus de chance de représenter le sujet du restaurant que d’autres termes fréquents dans le corpus.

La méthode TF-IDF est utile aux utilisateurs quand il s’agit de créer des textes le plus unique possible. Cela permet de viser bons classements dans les pages de résultats du moteur de recherche pour la recherche de certains mots. Pendant longtemps, la densité des mots-clés était utilisée comme référence pour les textes optimisés. Aujourd’hui, la formule TF-IDF représente une manière beaucoup plus précise d’optimiser ses contenus.

La méthode TF-IDF est composée de deux parties : TF (Term Frequency) et IDF (Inverse Document Frequency).

TF : permet de mesurer l’importance relative d’un mot dans un document. Elle est définie par la formule suivante :

$$TF(i, j) = \frac{\log(1 + Freq(i, j))}{\log(L_j)} \quad (2.1)$$

Où :

i : le mot.

j : le document.

L_j : le nombre total de mot dans le document.

$Freq(i, j)$: la fréquence du mot.

IDF : mesure la signification d’un terme non pas en fonction de sa fréquence dans un document particulier, mais en fonction de sa distribution et de son utilisation dans l’ensemble des documents. Plus un terme a de potentiel, plus l’Inverse Document Frequency est élevée.

On obtient le score par la formule suivante :

$$IDF(i) = \log\left(\frac{N_D}{f_i} + 1\right) \quad (2.2)$$

Où :

N_D : le nombre total de documents dans le corpus.

f_i : nombre de documents où le mot i apparaît.

Une fois les deux termes calculés, on calcule sa valeur par la formule suivante :

$$TF - IDF(i, j) = TF(i, j) * IDF(i) \quad (2.3)$$

Après l'application de cette mesure (TF-IDF) sur le profil utilisateur, chaque utilisateur aura un vecteur. La figure suivante présente le principe de construction de la matrice de caractéristique des utilisateurs avec TF-IDF.

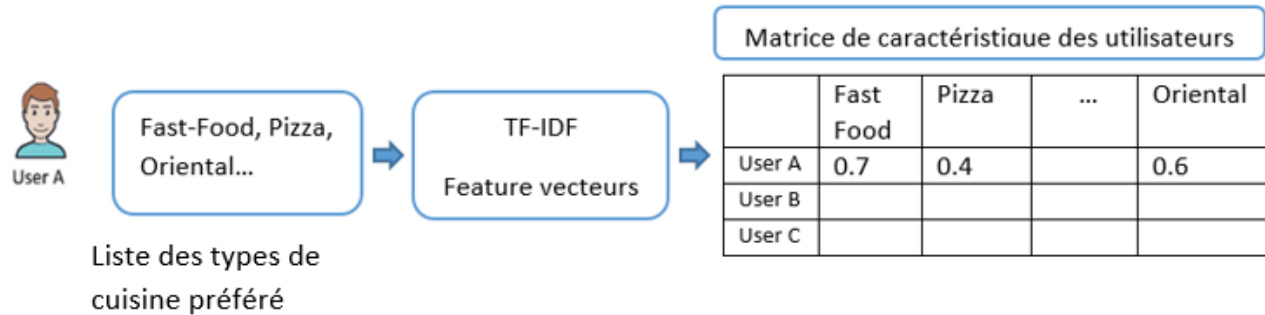


FIGURE 2.7 – Appliation du TF-Idf sur le profil d'utilisateur

Nous avons aussi appliqué la mesure (TF-IDF) sur le profil restaurant pour représenter les données sous forme d'un vecteur. La figure suivante présente le principe de construction de la matrice de caractéristique des restaurants avec TF-IDF.

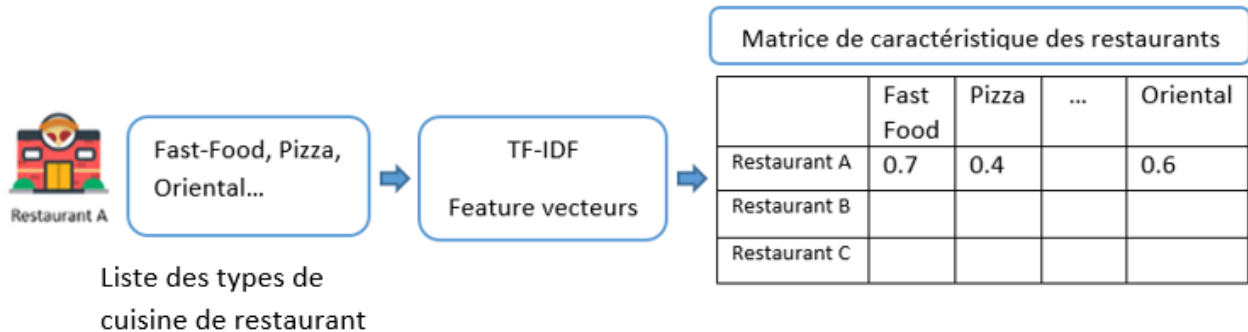


FIGURE 2.8 – Appliation du TF-Idf sur le profil de restaurant

Une fois que nous avons les deux matrices de l'utilisateur et le restaurant qui contient les scores TF-IDF, nous pouvons calculer directement la valeur de similarité entre eux. Les détails de son fonctionnement seront abordés dans la partie ci-dessous.

2.1.3.3 Calcul de la similarité

Plusieurs formules de calcul de similarité ont été proposés pour calculer la similarité entre un profil d'utilisateur et un profil d'item (restaurant). Nous présenterons ci-dessous la formule que nous avons utilisé dans notre algorithme de recommandation.

Similarité cosinus

La similarité cosinus est la méthode la plus utilisée en tant que mesure de ressemblance entre deux documents. Elle est utilisée dans la recherche d'information (un document vectorisé est constitué par les mots de la requête et est comparé par mesure de cosinus de l'angle avec des vecteurs correspondant à tous les documents présents dans le corpus. On évalue ainsi lesquels sont les plus proches) [16].

Le score de similarité cosinus calcule le cosinus de l'angle entre deux vecteurs dans un espace à n dimensions [2].

Dans cette méthode le profil utilisateur et le profil restaurant sont considérés comme deux vecteurs de même origine dans un espace de n dimensions, n est égale au nombre de catégories par les deux profils utilisateur et restaurant.

Par exemple, étant donné un utilisateur u , dont le vecteur caractéristique de préférence est $P_u = (p_1, p_2, \dots, p_n)$, et un restaurant r , dont le vecteur caractéristique est $F_r = (f_1, f_2, \dots, f_n)$. Afin de déterminer si l'utilisateur u préfère le restaurant r ou non, l'algorithme cosinus similarité est appliqué pour calculer la similarité entre u et r [30]. La formule suivante montre la similarité de cosinus détaillée entre u et r :

$$\text{Similar}(u, r) = \cos(u, r) = \frac{\sum_{i=1}^n p_i f_i}{\sqrt{\sum_{i=1}^n p_i^2} \times \sqrt{\sum_{i=1}^n f_i^2}}$$

La valeur du $\text{Similar}(u, r)$ est toujours entre $[0, 1]$. Si $\text{Similar}(u, r)$ est proche de 1, u et r sont très similaires. Si l'utilisateur u visite le restaurant r plusieurs fois, la similarité entre u et r augmentera. Par conséquent, la cosinus similarité peut refléter efficacement le changement des préférences d'utilisateur.

	Restaurateur r
Utilisateur u	0.8221

TABLE 2.1 – Score obtenu par l'application du cosinus similarité sur l'exemple

2.1.3.4 Filtrage par localistaion

Dans notre contexte, nous recommandons des restaurants, l'utilisateur sera donc obligé de se déplacer physiquement vers ce restaurant. C'est pour cela que la distance entre l'utilisateur et le restaurant est un facteur très important, qui peut directement influencer le résultat de la recommandation. Pour cela nous avons raffiné le résultat de recommandation en considérant la distance séparent l'utilisateur et les restaurants. Le GPS est utilisé pour localiser la position des utilisateurs ainsi que les restaurants afin de calculer la distance entre deux points en utilisant leur latitude et longitude obtenu [7].

Étant donné deux points : Utilisateur(X_1, Y_1) et Restaurant(X_2, Y_2), tel que :

X1 : Latitude de l'utilisateur.

Y1 : Longitude de l'utilisateur.

X2 : Latitude de restaurant.

Y2 : Longitude de restaurant.

La distance entre l'utilisateur et le restaurant est calculée par les formules suivantes :

$$a = \sin^2\left(\frac{X2 - X1}{2}\right) + \cos(X1) * \cos(X2) + \sin^2\left(\frac{Y2 - Y1}{2}\right) \quad (2.4)$$

$$c = 2 * \arctan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \quad (2.5)$$

$$Dist(Utilisateur, Restaurant) = R * c \quad (2.6)$$

R : Rayon = 6371

Lorsque l'utilisateur envoie une demande de recommandation, le système applique les méthodes précédentes pour obtenir une liste des restaurants trier par la similarité de ces restaurants par rapport au profil de cet utilisateur, après le système va calculer la distance entre ce dernier et chaque restaurant par les formules (2.4) (2.5) (2.6) et il recommande les restaurants les plus similaires dont la distance est moins ou égale à la distance désirée par l'utilisateur.

La figure suivante montre la liste de recommandation pour un utilisateur :

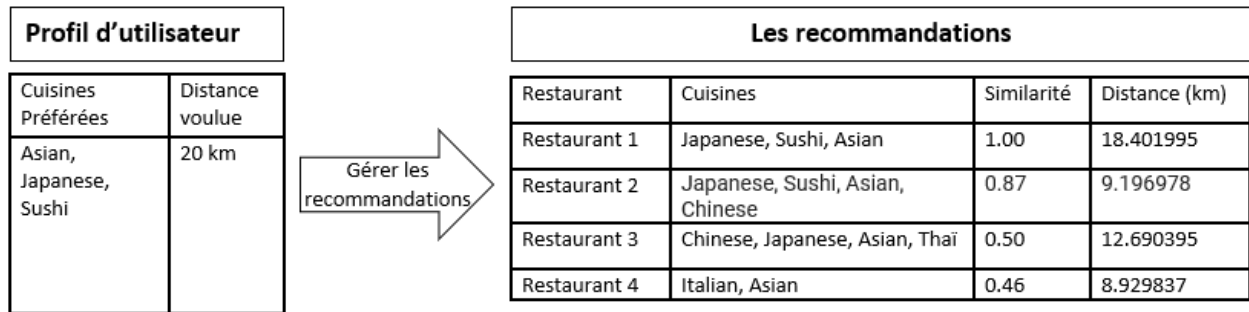


FIGURE 2.9 – Résultat de recommandation pour un utilisateur

2.2 Conception de l'application

Cette partie représente la vue générale de notre solution où sont identifiés en premier les besoins fonctionnels et non-fonctionnels, les acteurs principaux, et enfin les interactions entre acteurs et composants. Pour cela nous avons opté pour une approche orientée objet basée sur le langage UML.

Le langage de modélisation unifié (UML) est un langage de modélisation qui a été pensé pour permettre de modéliser, selon une approche orientée objet, les éléments et les comportements des systèmes [29]. C'est une simplification de la réalité qui permet de mieux comprendre, documenter

et faire comprendre (communiquer les documents aux équipes de développement) le système à développer. Il comporte plusieurs types de diagrammes qui apportent chacun une description d'un certain aspect du système.

2.2.1 Spécification des besoins

Cette section identifie les besoins fonctionnels puis non fonctionnels que notre application doit satisfaire.

2.2.1.1 Besoins fonctionnels

Les besoins fonctionnels représentent les fonctionnalités principales de notre solution et doivent permettre aux :

— Client :

1. Créer un compte client.
2. Consulter la liste des résultats de la recommandation des restaurants.
3. Consulter les informations des restaurants comme l'adresse, le numéro de téléphone, les types de cuisines, la qualité de service et la qualité de food, etc.
4. Gérer les choix de préférences.
5. Gérer son compte.

— Restaurateur :

1. Créer un compte restaurateur.
2. Ajouter un restaurant.
3. Consulter les restaurants ajoutés.

2.2.1.2 Besoins non-fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis l'application pour son bon fonctionnement. Notre application devra répondre aux besoins suivants :

1. La disponibilité : l'application doit être disponible à l'utilisation par n'importe quel utilisateur.
2. La sécurité : l'application doit sécuriser l'accès aux informations critiques et assurer la confidentialité.
3. La fiabilité : les données fournies par l'application doivent être fiables.
4. La convivialité de l'interface graphique : l'application doit fournir une interface conviviale et simple.
5. L'adaptabilité : permettre à l'utilisateur d'adapter l'application par rapport à sa propre expérience.

2.2.2 Diagramme de cas d'utilisation

Celui-ci est utilisé afin de donner une vision globale du comportement fonctionnel du système en spécifiant les fonctionnalités principales et les interactions entre les utilisateurs et le système.

Identification des acteurs

- **Client** : c'est l'acteur principal de l'application et la cible de recommandation, il peut consulter la liste des restaurants recommandés selon ses préférences (les cuisines préférées, la distance voulue pour une recommandation plus proche à sa localisation). La figure 2.10 illustre le diagramme de cas d'utilisation de cet acteur.

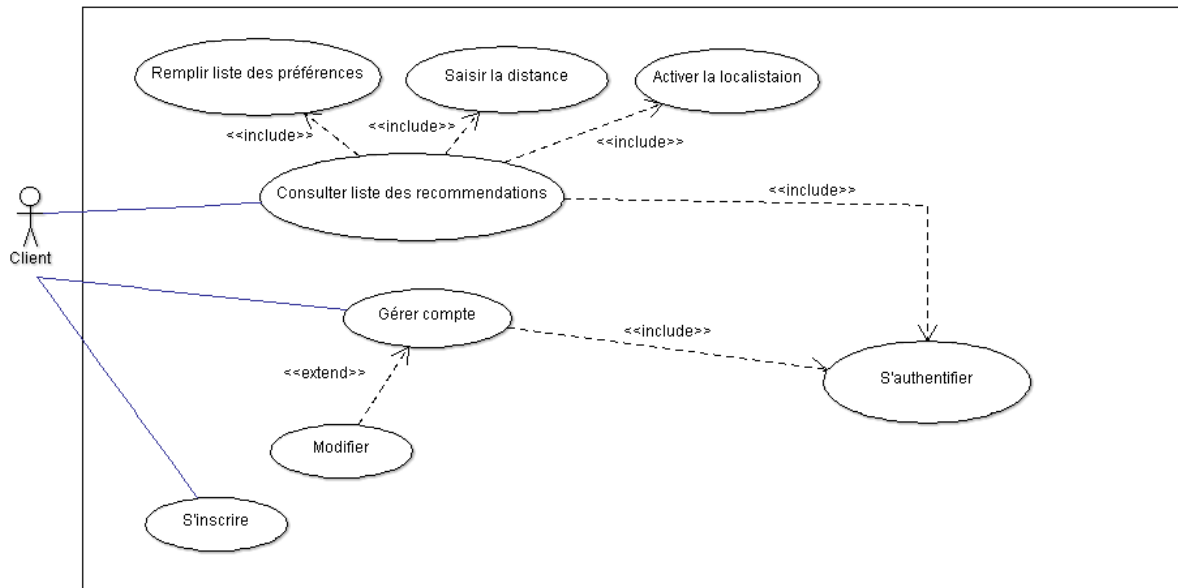


FIGURE 2.10 – Diagramme de cas d'utilisation du client

- **Restaurateur** : Il peut ajouter ses restaurants et les consulter. La figure 2.11 illustre le diagramme de cas d'utilisation de cet acteur.

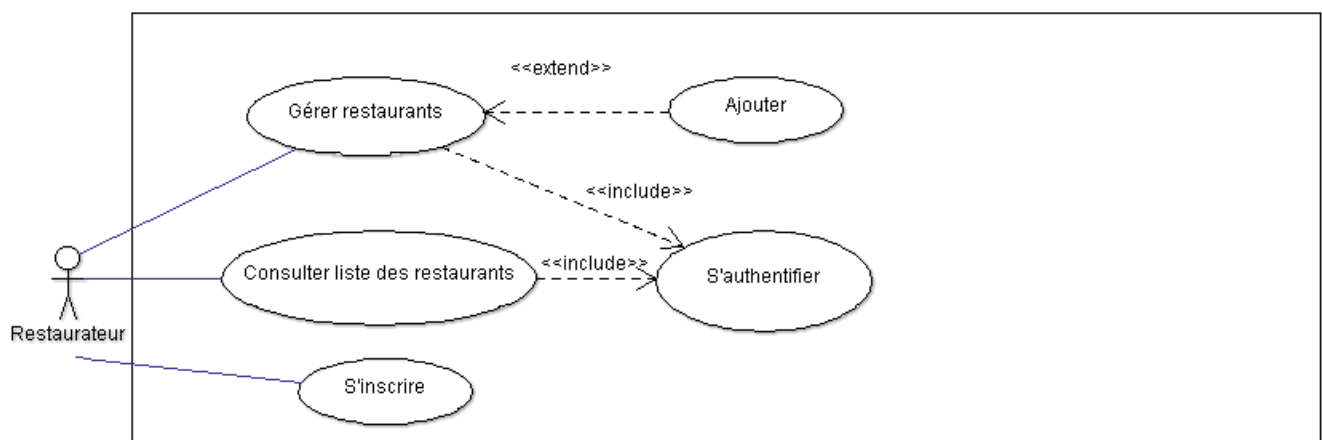


FIGURE 2.11 – Diagramme de cas d'utilisation du restaurateur

2.2.3 Diagramme de classes

Le diagramme de classes est une vue statique de la structure interne du système, des éléments qui le composent et leurs relations.

La figure suivante montre le diagramme de classes de l'application :

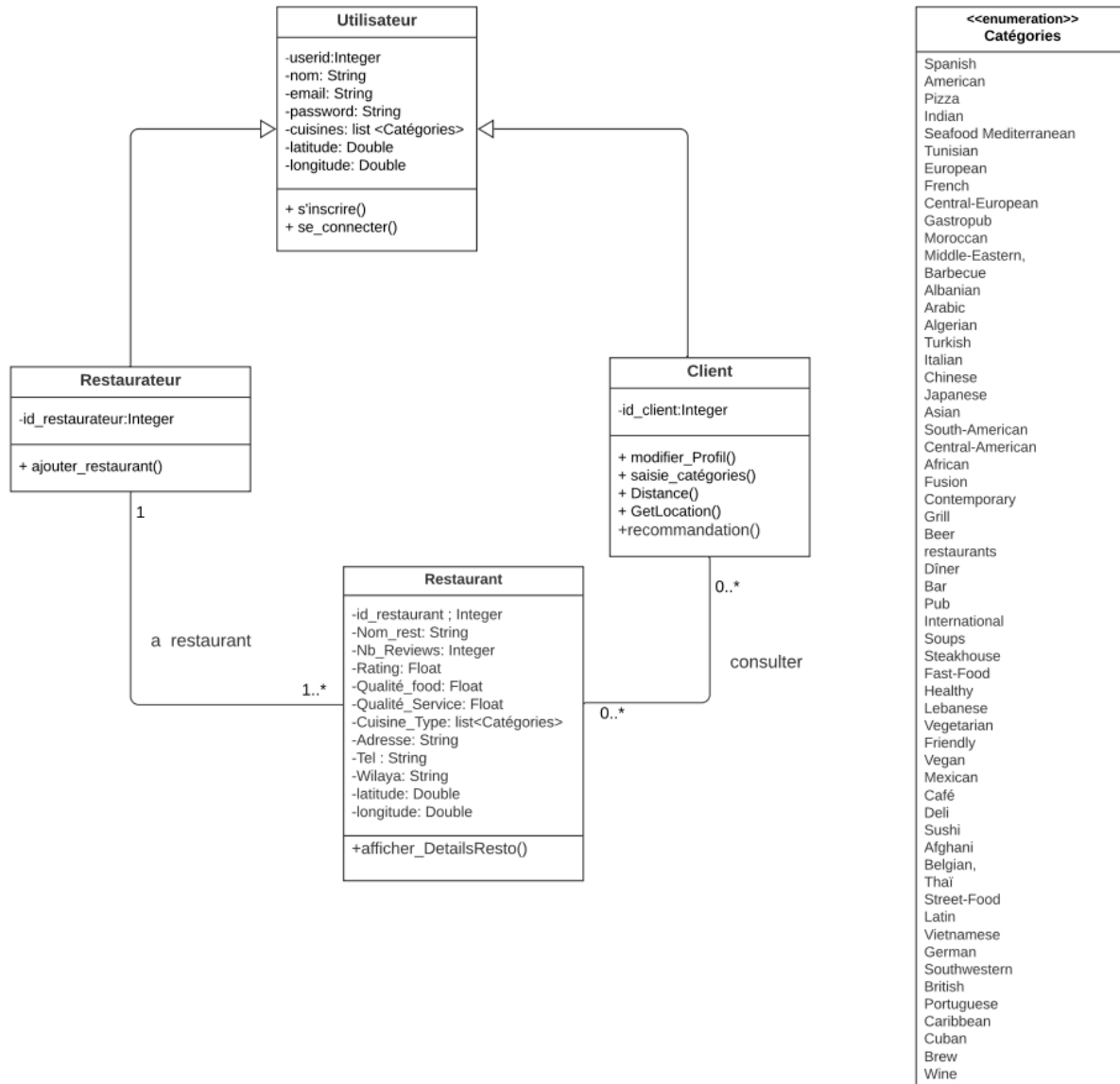


FIGURE 2.12 – Diagramme de classes

Explication du diagramme :

- Un restaurant a une liste de types de Cuisines.
- Un client peut consulter un ou plusieurs restaurants recommander.
- Un restaurant peut être consulter par zéro ou plusieurs utilisateurs.
- Un restaurateur peut ajouter un restaurant ou plus.

2.2.4 Modèle relationnel du système

A partir du diagramme de classes nous avons déduit le schéma relationnel qui représente la structure de notre base de données. Pour cela nous avons suivi les règles de passage au modèle relationnel.

règle 1 : Présence de la cardinalité (1) d'un côté de l'association.

- Chaque classe se transforme en une relation.
- L'identifiant de la classe qui est associée à la cardinalité (1) devient la clé étrangère de l'autre classe.

règle 2 : Présence de la cardinalité (0..*) ou (1..*) des deux côtes de l'association.

- Chaque classe se transforme en une relation.
- La classe-association se transforme en une relation. Cette relation a comme attributs l'identifiant de chacune des deux classes, ainsi que d'éventuels autres attributs.

Schéma relationnel du système :

Voici le modèle déduit par l'application des règles de passage :

Utilisateur (userid, nom, email, password, cuisines, latitude, longitude)

Client (id_client*)

Restaurateur (id_restaurateur*)

Restaurant (id_restaurant, Nom_rest, Nb_Reviews, Rating, Qualité_food, Qualité_Service, Cuisine_Type, Adresse, Tel, Wilaya, latitude, longitude, id_restaurant*)

Client_Restaurant (id_client*, id_restaurant*)

2.2.5 Diagramme de séquence

Le diagramme de séquence décrit les messages échangés ou les interactions entre les utilisateurs et le système en fonction du temps. Nous avons choisi de représenter les scénarios suivants : Se connecter et recommandation d'un restaurant.

Diagramme de séquence « Se connecter »

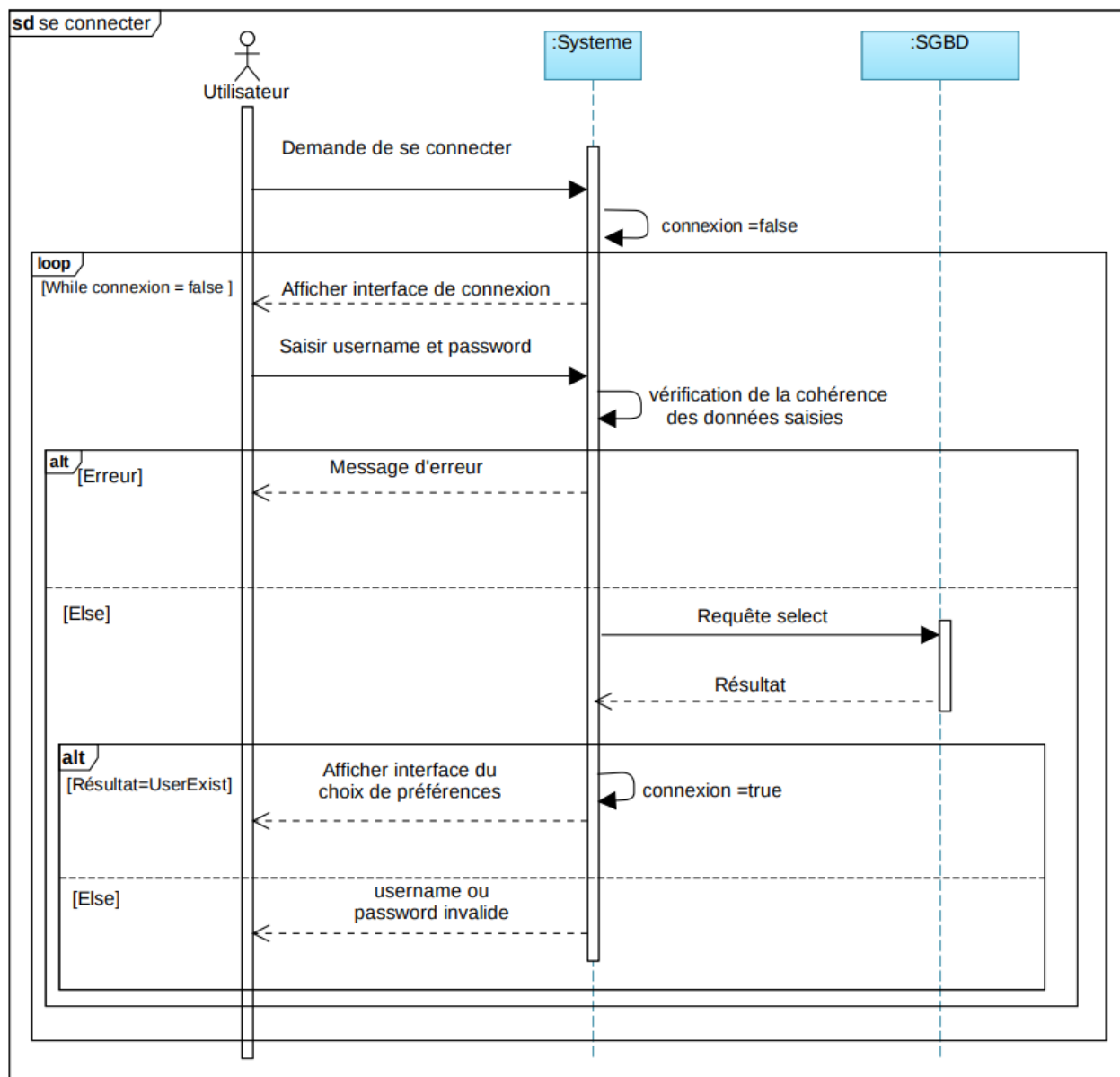


FIGURE 2.13 – Diagramme de séquence de connexion

Diagramme de séquence «recommandation»

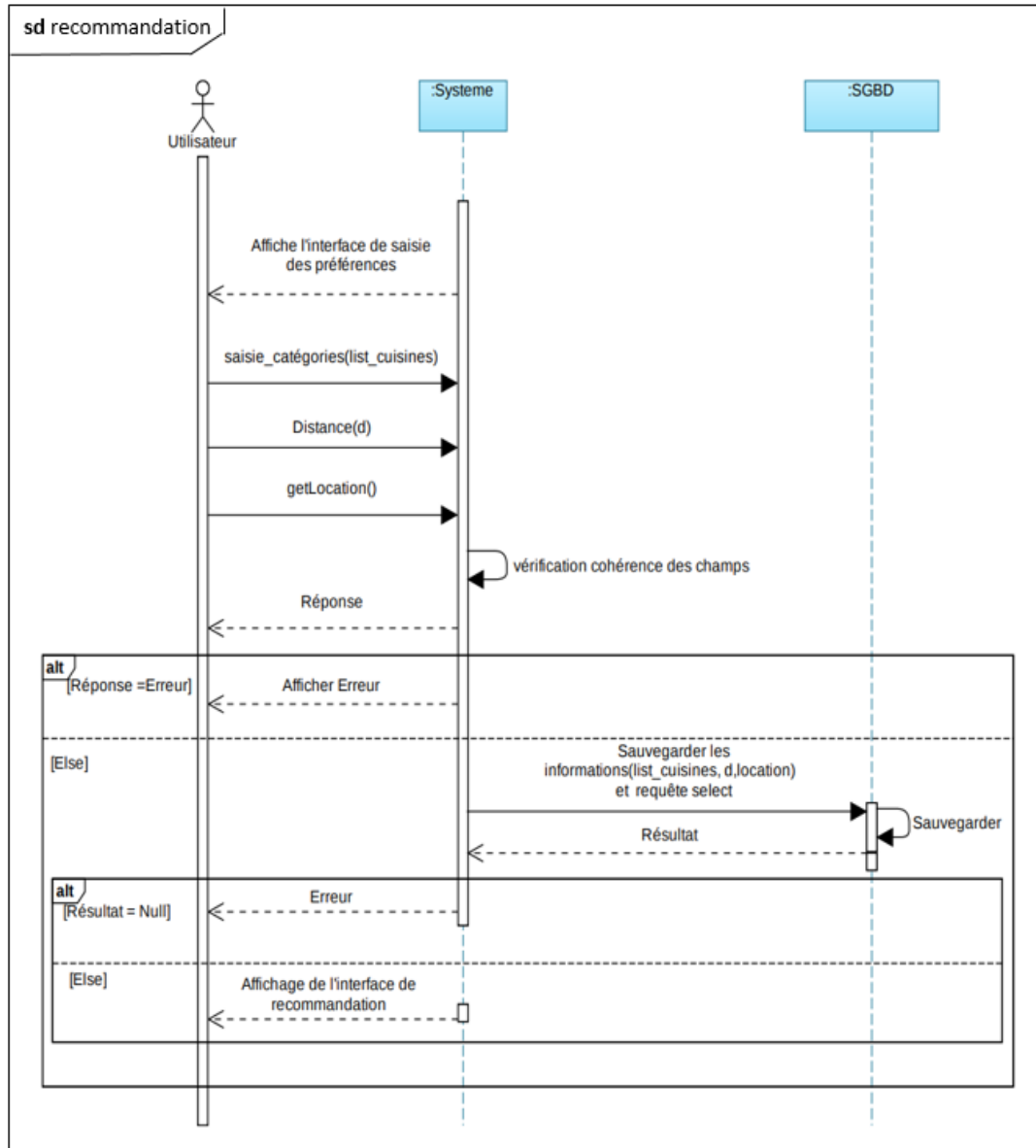


FIGURE 2.14 – Diagramme de séquence de recommandation

2.3 Conclusion

Dans ce chapitre, nous avons présenté une approche pour la recommandation des restaurants algériens à partir de notre application RestoFinder. Puis, nous avons fait une modélisation à l'aide du langage de modélisation UML afin de décrire l'aspect fonctionnel et non fonctionnel de l'application. Dans le prochain chapitre, nous présenterons les outils utilisés pour nos données, ainsi que le résultat de notre système de recommandation et les différentes interfaces de l'application RestoFinder.

Dans ce chapitre, nous allons présenter nos données avec les outils et les langages utilisés pour les traiter, par la suite nous allons présenter l'implémentation de la collecte de données des datasets, ainsi que l'implémentation de notre système de recommandation. Enfin, nous présenterons les différentes interfaces de notre application mobile RestoFinder.

3.1 Les outils de programmation

Android studio

Android studio est un environnement de développement d'applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle.

PyCharm

PyCharm est un environnement de développement intégré (IDE) utilisé en programmation informatique, spécifiquement pour le langage Python. Il est développé par la société tchèque JetBrains. Il fournit une analyse de code, un débogueur graphique, un testeur d'unités intégrées, une intégration avec les systèmes de contrôle de version (VCS) et prend en charge le développement Web avec flask. Nous avons utilisé cet outil pour développer notre système de recommandation avec le langage python.

SQLite

SQLite est une bibliothèque du langage C qui implémente un petit moteur de base de données. Il est rapide, autonome, hautement fiable et complet. SQLite est le moteur de base de données le plus utilisé au monde. SQLite est intégré à tous les téléphones mobiles et à la plupart des ordinateurs et il est intégré à d'innombrables autres applications que les gens utilisent chaque jour [27]. Nous l'avons utilisé pour la création de la base de données de notre projet.

Flutter

Flutter est un framework de développement d'applications permet à partir d'un seul code de développer des applications mobiles pour les deux grandes plateformes IOS et Android . Il est

développé par Google. La première version stable est arrivée en décembre 2018. Flutter s’appuie sur le langage Dart et met à disposition une grande variété de bibliothèques d’éléments d’IU standard pour Android et iOS [6]. Nous l’avons utilisé dans l’implémentation de notre application.

Flask

Flask est un framework open source Web en Python populaire, ce qui signifie qu’il s’agit d’une bibliothèque Python3 utilisée. Il est classé comme microframework car il est très léger [5]. Nous avons utilisé ce framework dans la réalisation du backend de notre application.

Json

JSON JavaScript Object Notation est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter l’information de façon structurée comme le permet XML par exemple [11].

Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développement logiciel, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s’appuyant principalement sur Java [4].

Selenium

Selenium est un framework web qui permet d’exécuter des tests multi-navigateurs. Cet outil est utilisé pour automatiser les tests d’applications Web pour vérifier qu’elles fonctionnent correctement. Nous l’avons utilisé pour créer un robot de scraping [25].

3.2 Les langages de programmation

Java

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d’être très largement utilisé pour le développement d’applications de types très disparates. Java est notamment largement utilisé pour le développement d’applications d’entreprises et mobiles. Android Studio nous permet principalement d’éditer les fichiers Java [10]. Nous avons utilisé le langage java pour la collecte de nos données.

Dart

Dart est un langage de programmation développé principalement par Google. Il est utilisé pour créer la programmation côté serveur, ainsi que le développement d’applications mobiles (via l’API Flutter). La programmation dans Dart doit être une alternative attrayante à JavaScript dans des navigateurs Web modernes [3].

SQL

Le langage de requête structuré (SQL) est un langage de base de données standard qui est utilisé pour créer, maintenir et récupérer la base de données relationnelle [26].

Python

Python est un langage de programmation orienté objet clair et puissant, comparable à Perl, Ruby, Scheme ou Java. Il utilise une syntaxe facile et simple à utiliser, facilitant la lecture des programmes. Python peut également être modifié et redistribué librement, car le langage est protégé par des droits d’auteur et disponible sous une licence open source [20]. Pour le développement de notre système, le langage de développement python a été utilisé avec les bibliothèques suivantes : Scikit-learn, NumPy, Pandas .

Numpy

Numpy est une bibliothèque pour le langage de programmation Python, ajoutant la prise en charge de grands tableaux et matrices multidimensionnels, ainsi qu’une grande collection de fonctions mathématiques de haut niveau pour opérer sur ces tableaux [18].

Pandas

Pandas est une bibliothèque open source sous licence BSD (Berkeley Software Distribution License) fournissant des structures de données et des outils d’analyse de données hautes performances et faciles à utiliser pour le langage de programmation Python. Elle est principalement utilisée pour l’analyse de données. Pandas permet d’importer des données à partir de divers formats de fichiers tels que des valeurs séparées par des virgules, JSON, SQL, Microsoft Excel. Pandas permet diverses opérations de manipulation de données telles que la fusion, le remodelage, la sélection, ainsi que le nettoyage des données et les fonctionnalités de gestion des données [19].

Scikit-learn

Scikit-learn est une bibliothèque libre Python destinée à l’apprentissage automatique. Elle propose dans son Framework de nombreuses bibliothèques d’algorithmes à implémenter, clé en main. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support [24].

3.3 Implémentation du système de collecte de données

3.3.1 La collecte de données

la collecte de données pour les restaurants algériens a été effectuée sur le site tripadvisor. L’extraction de données disponibles sur Tripadvisor pour les restaurants algériens est basée sur l’utilisation du web scraping avec le robot selenium programmé avec le langage java. La figure suivante montre une vision d’une partie du code utilisé pour l’extraction de données :

```

19 public class DataSetAuto {
20
21     /**
22      * @param args the command line arguments
23      */
24     public static WebDriver driver ;
25     public static FileWriter csvWriter ;
26     public static void main(String[] args) {
27         System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\Isil-Yas\\\\workspace\\\\data\\\\exefiles\\\\chromedriver.exe");
28
29         ChromeOptions options = new ChromeOptions();
30         options.addArguments("--disable-gpu", "--blink-settings=imagesEnabled=false");
31         driver = new ChromeDriver(options);
32
33
34
35         BrowserUtil.openNewTab(driver);
36         BrowserUtil.switchTo(driver,0);
37
38         TripAdv.init(driver);
39         TripAdv.goToResteauAlgeria();
40
41         try {
42             csvWriter = CSVUtils.create_CSV_file("Rest.csv","URL", "Nom","Nombre de Reviews","Review Note",
43             ,"food","service","valeur","Details","Adrs","Tel","RevDet");
44         } catch (Exception e) {
45             System.out.println("Fichier Pas Creer");
46             System.exit(1);
47         }
48
49         int page = 1 ;
50         boolean existeNextPage = true ;
51         while (existeNextPage)
52         {
53             System.out.println("Page : "+page);
54             ArrayList<Integer> indexes = TripAdv.getLinks();
55             System.out.println(indexes);
56             for (int i = 0 ; i < indexes.size() ; i++ )
57             {
58                 System.out.println("Page : "+page+" , i : "+i+" , Index : "+indexes.get(i));
59                 TripAdv.ScrapLink(indexes.get(i));
60                 System.out.println("-----");
61             }
62
63             existeNextPage = TripAdv.ClickNextButton(page);
64             page++;
65             BrowserUtil.sleep(10000);
66         }
67
68         CSVUtils.finish(csvWriter);
69     }
70 }
71 }
72

```

FIGURE 3.1 – Exemple du code utilisé pour l'extraction des données

La figure suivante démontre le résultat des données obtenu pour quelques restaurants avec l'utilisation du web scraping :

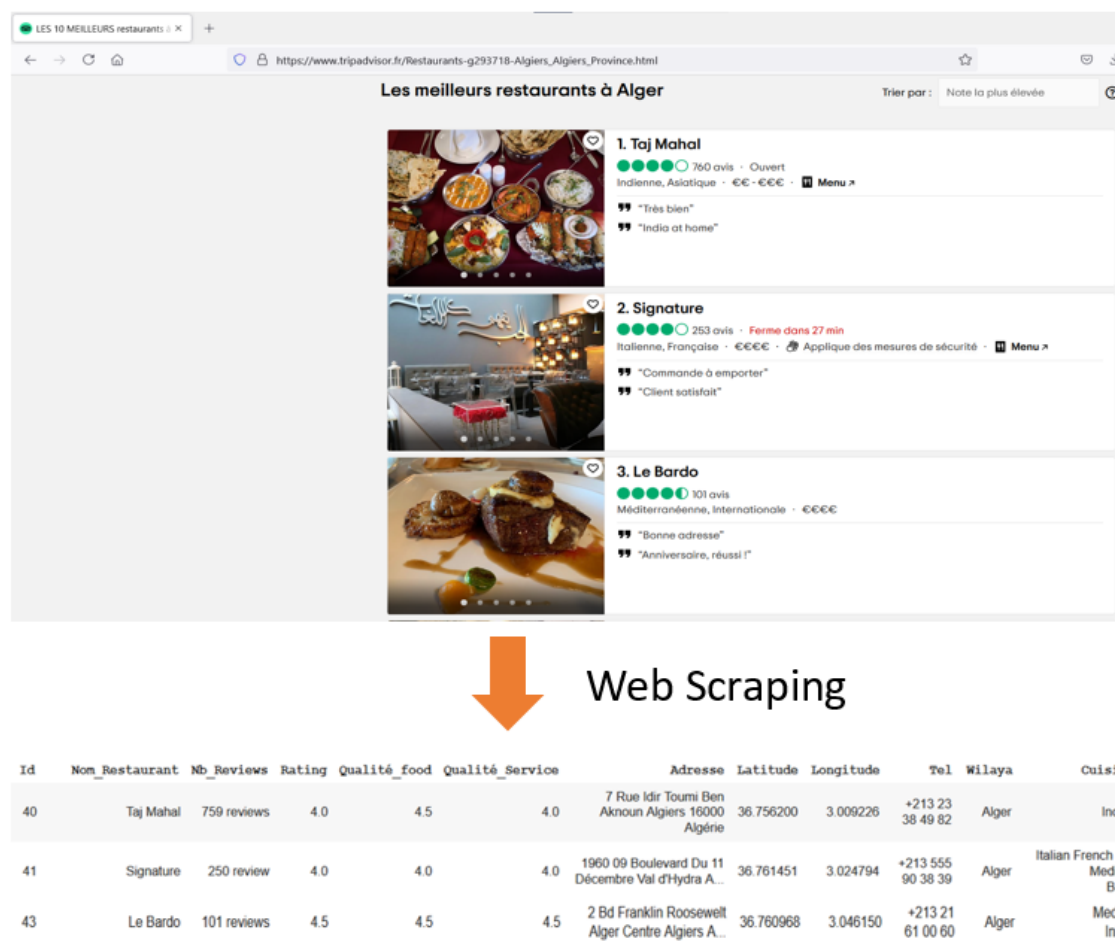


FIGURE 3.2 – Exemple du résultat obtenu en utilisant le web scraping sur TripAdvisor

3.3.1.1 Statistiques

Voici les statistiques des deux bases de données :

Base de données	Nb restaurant	Wilaya	Nb catégories
Restaurant	4001	Alger, Oran, Setif, Annaba, Skikda, Boumerdess, Tizi Ouazou, Blida, Ain Defla, Mostghanem, Constantine, Batna, Sidi Bel Abbès, Tlemcen, Relizane , El Tarf, Jijel, Media, El Oued, Buira, Bejaia.	60

TABLE 3.1 – Statistiques de la base de données restaurant

Base de données	Nb catégories
Utilisateur	60

TABLE 3.2 – Statistiques de la base de données utilisateur

3.4 Implémentation du système de recommandation

3.4.1 Représentation des résultats du système

Dans cette section nous présentons un résultat acquis par le système de recommandation sur les données des deux datasets (restaurant, utilisateur).

L'exemple suivant démontre les performances acquises par le système sur les données pour la recommandation des restaurants à un utilisateur X avec une distance égale à 20 km.

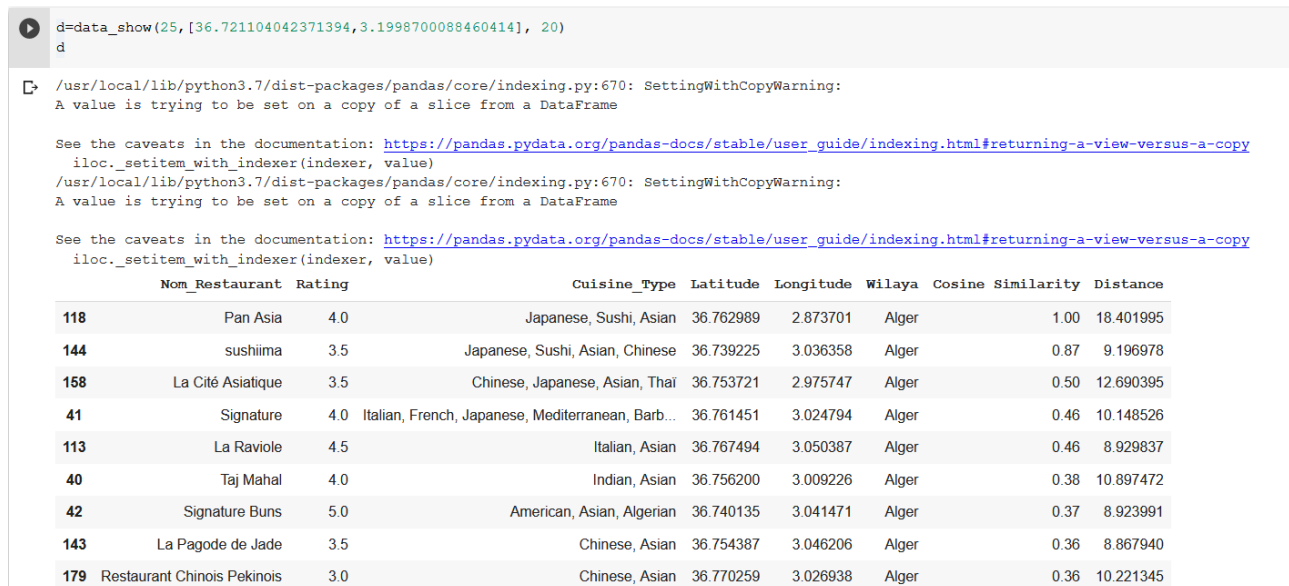
Id_User	Nom	Cuisines	Location (latitude, longitude)
25	Hocine	Asian, Japanese, Sushi	36.721104042371394, 3.1998700088460438

TABLE 3.3 – Les données de l'utilisateur X

Input :

`d=Recommander(Id_user,[latitude,longitude],distance)`

Output :



```
d=data_show(25, [36.721104042371394, 3.1998700088460414], 20)
d
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:670: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  iloc._setitem_with_indexer(indexer, value)
/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:670: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  iloc._setitem_with_indexer(indexer, value)
```

	Nom_Restaurant	Rating	Cuisine_Type	Latitude	Longitude	Wilaya	Cosine Similarity	Distance
118	Pan Asia	4.0	Japanese, Sushi, Asian	36.762989	2.873701	Alger	1.00	18.401995
144	sushima	3.5	Japanese, Sushi, Asian, Chinese	36.739225	3.036358	Alger	0.87	9.196978
158	La Cité Asiatique	3.5	Chinese, Japanese, Asian, Thai	36.753721	2.975747	Alger	0.50	12.690395
41	Signature	4.0	Italian, French, Japanese, Mediterranean, Barb...	36.761451	3.024794	Alger	0.46	10.148526
113	La Raviole	4.5	Italian, Asian	36.767494	3.050387	Alger	0.46	8.929837
40	Taj Mahal	4.0	Indian, Asian	36.756200	3.009226	Alger	0.38	10.897472
42	Signature Buns	5.0	American, Asian, Algerian	36.740135	3.041471	Alger	0.37	8.923991
143	La Pagode de Jade	3.5	Chinese, Asian	36.754387	3.046206	Alger	0.36	8.867940
179	Restaurant Chinois Pekinois	3.0	Chinese, Asian	36.770259	3.026938	Alger	0.36	10.221345

FIGURE 3.3 – Exemple d'un résultat du recommandation des restaurants pour un utilisateur

Les résultats obtenus montrent que le système donne des bons résultats.

3.4.2 Evaluation du Filtrage basé contenu

Le système que nous avons construit ne contient pas l'historique des choix des utilisateurs, ce qui rend l'évaluation par les différentes méthodes RMSE (Erreur Quadratique Moyenne) et MAE (Erreur Moyenne Absolue) ou d'autres difficiles à réaliser. Nous avons contourné ce problème en calculant le degré de similarité avec la méthode « similarité cosinus » pour calculer la justesse entre les préférences utilisateur et ceux du profil restaurant. Nous avons obtenu de bons résultats.

Les métriques d'évaluation RMSE et MAE sont les deux métriques les plus utilisées et les plus pertinentes pour évaluer le système de recommandation car elles permettent le calcul de la différence entre deux variables continues.

3.5 Implémentation de l'application

3.5.1 Les interfaces de l'application

Les interfaces de notre application RestoFinder sont les suivantes :

Les interfaces du "client" :

- L'interface de connexion.
- L'interface d'inscription.
- L'interface des choix de préférences.
- L'interface de la liste des recommandations.
- L'interface de l'affichage des informations du restaurant.
- L'interface de gestion de compte du client.

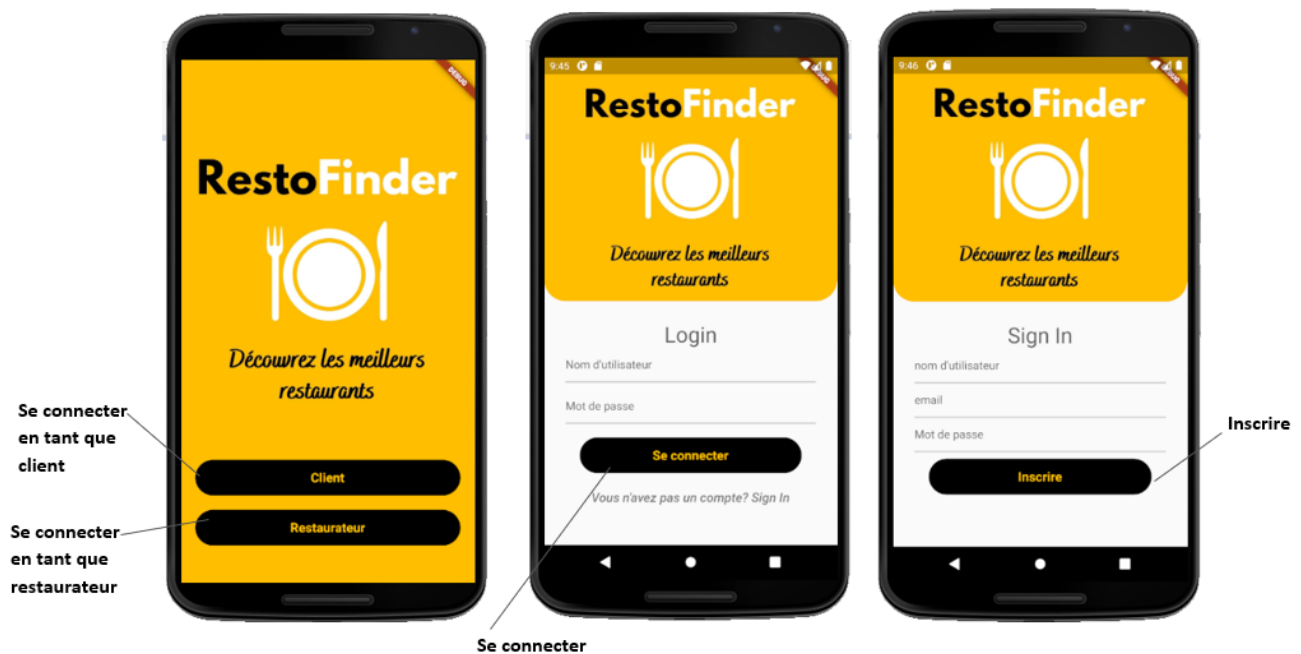


FIGURE 3.4 – Interface de connexion et d'inscription de RestoFinder

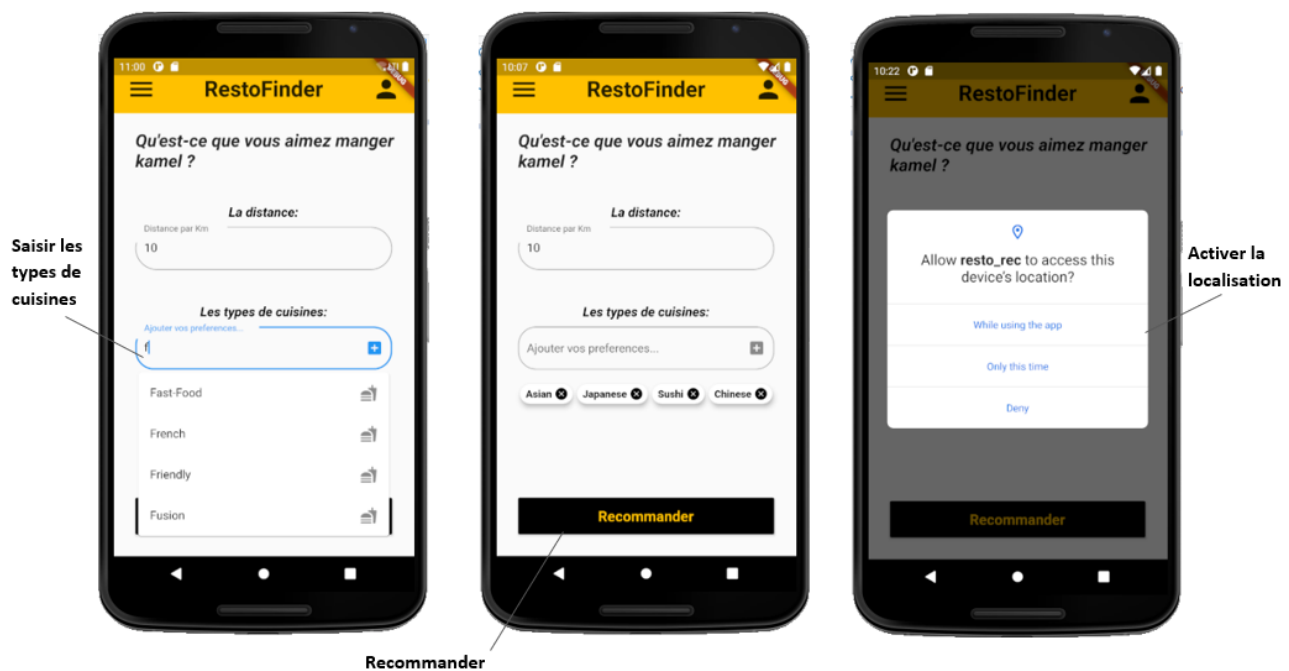


FIGURE 3.5 – L'interface des choix de préférences

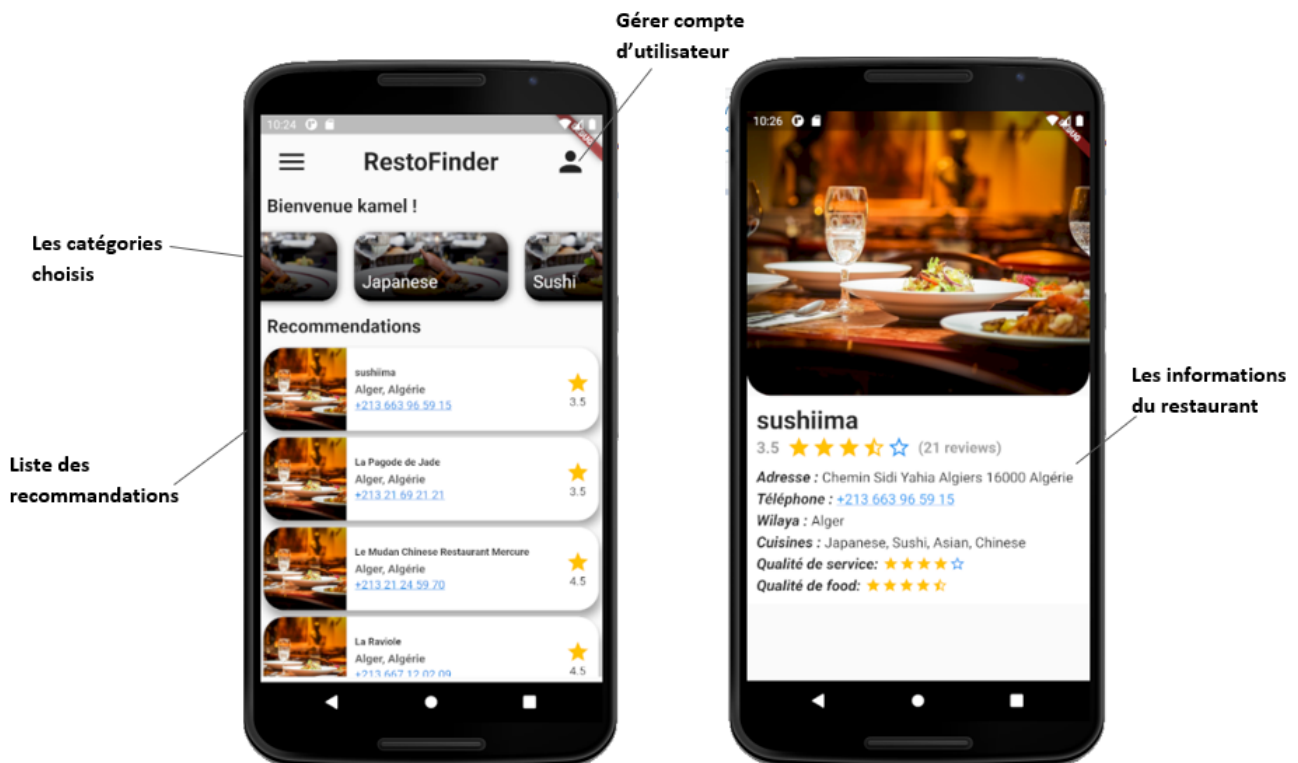


FIGURE 3.6 – L'interface de la liste de recommandations et l'affichage des informations du restaurant

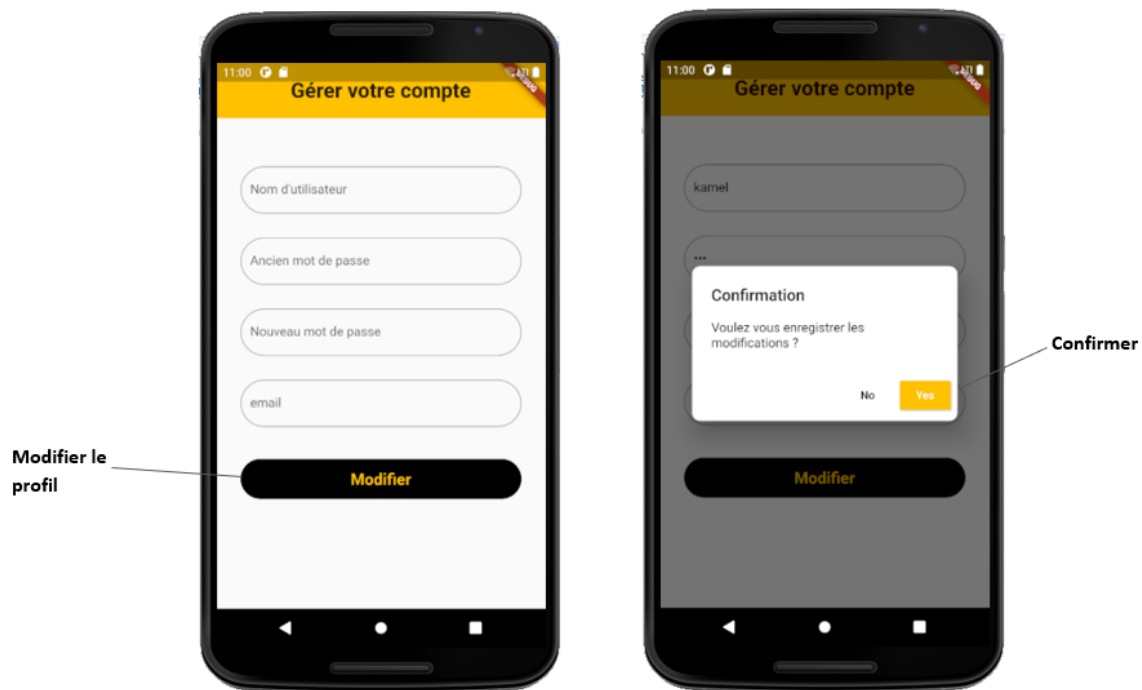


FIGURE 3.7 – L'interface de gestion de compte du client

Les interfaces du "Restaurateur" :

- L'interface de connexion.
- L'interface d'inscription.
- L'interface de la liste des restaurants du restaurateur.
- L'interface de l'ajout d'un restaurant.

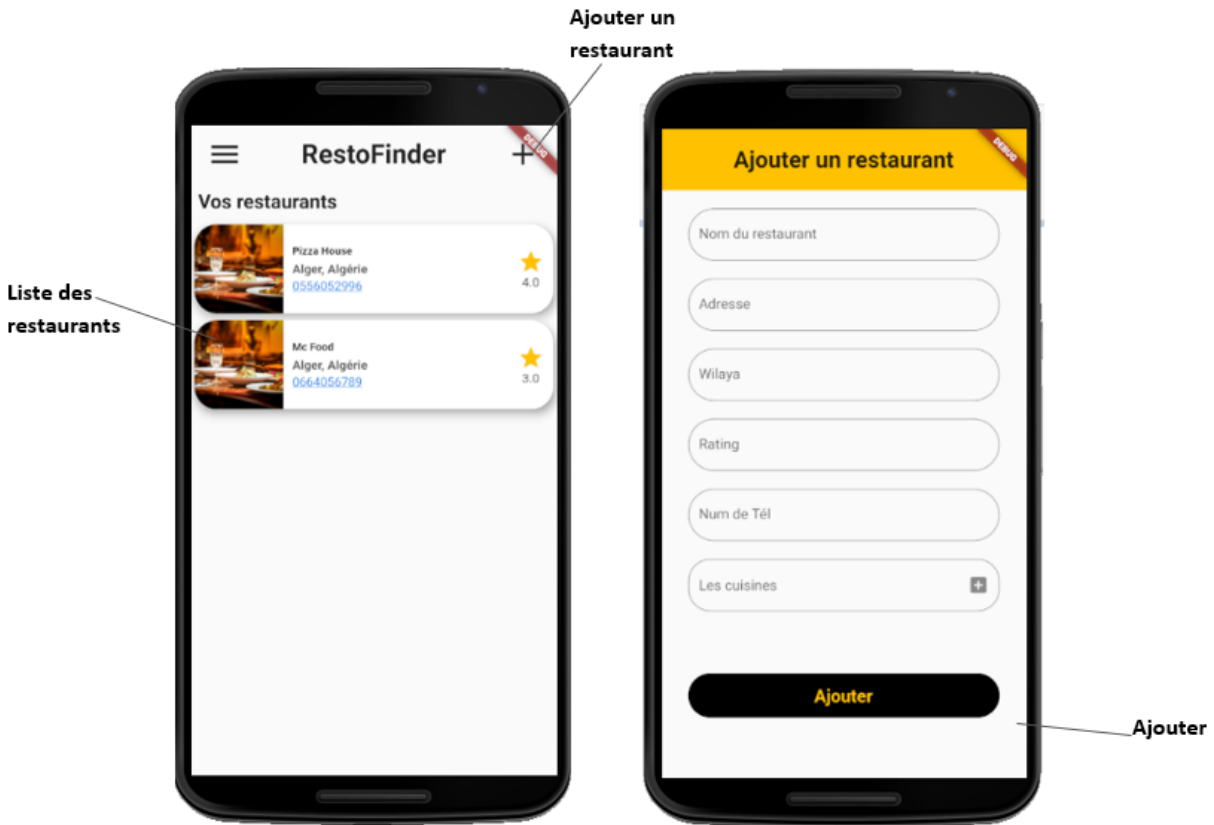


FIGURE 3.8 – L’interface de la liste des restaurants ajoutés et l’ajout.

3.6 Conclusion

Dans ce chapitre, nous avons abordé les différents outils et les langages de programmation utilisés dans la réalisations de ce projet. Puis, nous avons présenté les détails de la collecte de nos données et le résultat de notre système de recommandation, ainsi que l’évaluation de ce système. Enfin, nous avons présenté les différentes interfaces de notre application RestoFinder.

CONCLUSION GÉNÉRALE

Nous nous sommes intéressés dans le cadre de ce projet de fin d'études à la recommandation des restaurants en Algérie basée sur le contenu de manière automatique et ce selon les préférences de chaque utilisateur et la distance de sa localisation dans l'application RestoFinder dont nous avons développé.

À travers les expérimentations réalisées dans ce projet, nous pouvons affirmer que le système de recommandation est bien une solution à l'automatisation de la recherche des données auxquelles nous avons besoins. L'application a été développée avec succès, les performances du système ont été jugées bonnes. L'application RestoFinder est opérationnelle dès aujourd'hui.

Cependant, les principales difficultés rencontrées pendant la réalisation de ce travail fut le manque de base de données (dataset) qui n'est pas adapté aux autres approches de recommandation car il ne contient pas les informations personnalisés de chaque utilisateur, ce qui nous a poussé à personnaliser notre jeu de données où nous avons inclus des tags (qui représentent des types de cuisines préférés pour chaque utilisateur). Cette initiative a donné des bons résultats.

Comme perspectives futures à ce travail, nous commencerons par quelques suggestions reçues pendant le processus de validation pour améliorer notre système sur le plan de fonctionnalités, nous pouvons effectuer les améliorations suivantes :

- Continuer à construire et enrichir notre dataset et l'utilisation d'une méthode hybride qui compte trois approches qui se base sur le contenu, le filtrage collaboratif et le knowledge.
- Apporter des enrichissements à notre système de recommandation en considérant d'autres paramètres comme les prix, les menus proposés, l'avis des utilisateurs à proposé chaque restaurant, pour une meilleure recommandation aux avis des utilisateurs.
- Effectuer une évaluation approfondie de notre système de recommandation en considérant des données plus volumineuses, le contexte big data s'impose dans toute application.
- Ajouter des données dans le dataset des autres pays comme : Maroc, Tunisie, Égypte et d'autres pays qui sont proches de territoire algérien.
- Approfondir les caractéristiques de l'application en la dotant de subfonctions comme le choix de variété de cuisines.

BIBLIOGRAPHIE

- [1] R. BANIK. *Hands-On Recommendation Systems with Python*. Packt Publishing, 2018. ISBN : 9781788993753. URL : <https://books.google.dz/books?id=5hpNugEACAAJ>.
- [2] R. BANIK. *Hands-On Recommendation Systems with Python : Start building powerful and personalized, recommendation engines with Python*. Packt Publishing, 2018, p. 10-11, 46-47. ISBN : 9781788992534. URL : <https://books.google.dz/books?id=BglnDwAAQBAJ>.
- [3] *Dart*. [page consultée le 30/06/2021]. URL : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/le-langage-de-programmation-dart/>.
- [4] *Eclipse*. [page consultée le 01/07/2021]. URL : [https://fr.wikipedia.org/wiki/Eclipse_\(projet\)](https://fr.wikipedia.org/wiki/Eclipse_(projet)).
- [5] *Flask Documentation*. [en ligne; page consultée le 30/06/2021]. URL : <https://flask.palletsprojects.com/en/1.1.x/foreword/>.
- [6] *flutter-cest-quoi*. [en ligne; page consultée le 30/06/2021]. URL : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/flutter-cest-quoi/>.
- [7] *Formule de haversine*. URL : https://fr.wikipedia.org/wiki/Formule_de_haversine.
- [8] Judith HURWITZ et Daniel KIRSCH. *Machine Learning For Dummies®*, IBM Limited Edition. For dummies. John Wiley Sons, Inc, 2018, p. 14-17. ISBN : 9781119245513. URL : <https://www.ibm.com/downloads/cas/GB8ZMQZ3>.
- [9] D. JANNACH et al. *Recommender Systems : An Introduction*. Cambridge University Press, 2010. ISBN : 9781139492591. URL : https://books.google.dz/books?id=eygTJBd%5C_U2cC.
- [10] *Java*. [J.-M. DOUDOUX, «Présentation Java», page consultée le 01/07/2021]. URL : http://www.jmdoudoux.fr/accueil_java.html.
- [11] *JSON*. [«Manipuler des données JSON», [en ligne], page consultée le 02/07/2021]. URL : <https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON>.
- [12] S.N. MOHANTY et al. *Recommender System with Machine Learning and Artificial Intelligence : Practical Tools and Applications in Medical, Agricultural and Other Industries*. Wiley, 2020. ISBN : 9781119711599. URL : <https://books.google.dz/books?id=36TqDwAAQBAJ>.
- [13] A.C. MUELLER et S. GUIDO. *Machine learning avec Python*. edi8, 2018, p. 25-26. ISBN : 9782412037010. URL : <https://www.overleaf.com/project/60f32d10e312893115c9e786>.

- [14] J.P. MUELLER et L. MASSARON. *Machine Learning For Dummies*. For dummies. Wiley, 2016. ISBN : 9781119245513. URL : <https://books.google.dz/books?id=JLEyDAAAQBAJ>.
- [15] Amine NAAK. “Papyres : un système de gestion et de recommandation d’articles de recherche”. Mém. de mast. École Polytechnique de Montreal, 2009.
- [16] E. NEGRE. *Systèmes de recommandation : Introduction*. Systèmes d’information avancés. ISTE Editions, 2015. ISBN : 9781784050863. URL : <https://books.google.dz/books?id=KopmDwAAQBAJ>.
- [17] *Neural Network*. [Neural Network By : IBM Cloud Education ,page consultée le 17/04/2021]. URL : <https://www.ibm.com/cloud/learn/neural-networks>.
- [18] *NumPy*. NumPy. URL : <https://en.wikipedia.org/wiki/NumPy>.
- [19] *Pandas*. About pandas. URL : <https://pandas.pydata.org/about>.
- [20] *Python*. Python Documentation. URL : <https://docs.python.org/3/license.html>.
- [21] Shahzad QAISER et Ramsha ALI. “Text Mining : Use of TF-IDF to Examine the Relevance of Words to Documents”. In : *International Journal of Computer Applications* 181 (juil. 2018), p. 25-28. DOI : 10.5120/ijca2018917395. URL : https://www.researchgate.net/publication/326425709_Text_Mining_Use_of_TF-IDF_to_Examine_the_Relevance_of_Words_to_Documents.
- [22] Burke R. *Hybrid Recommender Systems : Survey and Experiments*. User Modeling et User-Adapted Interaction, 2002, p. 331-370.
- [23] E RICH. “User modeling via stereotypes”. In : H7245 (1979), p. 329-354.
- [24] *Scikit-learn*. Scikit-learn. URL : <https://fr.wikipedia.org/wiki/Scikit-learn>.
- [25] *Selenium*. «Commencer avec Selenium WebDriver (JAVA)», [en ligne], page consultée le 01/07/2021. URL : <https://www.all4test.fr/blog-du-testeur/commencer-avec-selenium-webdriver/>.
- [26] *SQL*. [en ligne,page consultée le 30/06/2021]. URL : <https://www.geeksforgeeks.org/structured-query-language/>.
- [27] *SQLite*. [What is SQLite, consultée le 30/06/2021]. URL : <https://www.sqlite.org/index.html>.
- [28] *TF-IDF*. [En ligne, consultée le 30/06/2021]. URL : <https://www.ionos.fr/digitalguide/web-marketing/analyse-web/analyse-tf-idf/>.
- [29] B. UNHELKAR. *Software Engineering with Uml*. Taylor & Francis Group, 2020, p. 19-27. ISBN : 9780367657383. URL : <https://books.google.dz/books?id=FU06zQEACAAJ>.
- [30] Jun ZENG et al. “A restaurant recommender system based on user preference and location in mobile environment”. In : *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. IEEE. 2016, p. 55-60.

Résumé

L'objet de ce projet de fin d'études de licence est la réalisation d'un système de recommandation de restaurants pour notre pays l'Algérie.

Notre approche est basé sur le contenu et ce selon les préférences des utilisateurs où nous avons pris en considération plusieurs critères comme : les types de cuisines, la qualité de service, ainsi que la proximité par rapport aux clients pour générer dynamiquement les résultats de la recommandation. Nous avons appliqué les algorithmes TF/IDF et Similarité cosinus pour générer les recommandations aux utilisateurs. Nous avons utilisé des datasets élaboré par nous-mêmes. Le système de recommandation que nous avons construit est capable de fournir aux utilisateurs des choix variés et intéressants à travers l'application mobile RestoFinder que nous avons développé pour intégrer notre système de recommandation de restaurants.