

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin d'étude

En vue de l'obtention du diplôme de Master Académique en Systèmes d'information avancés

Thème

Recommandation de transport dans les villes intelligentes

Réalisé par

MAMMAR KOUADRI Youcef

Devant le jury composé de

Dr GHANEM Souhila : **Présidente**
Dr BATAT Nadia : **Examinatrice**
Dr CHIBANI Samia epse SADOUKI : **Encadrante**
Dr EL BOUHISSI Houda Epse BRAHAMI : **Co-Encadrante**

Promotion 2022/2023

Dédicace

À mes parents, qui ont été ma source d'inspiration, mon soutien constant et ma motivation tout au long de ce voyage académique. Votre amour et vos encouragements ont été inestimables.

Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui m'ont soutenu tout au long de la réalisation de ce mémoire.

Je remercie chaleureusement ma directrice de mémoire, Dr. Samia CHIBANI, pour sa précieuse orientation, son expertise, et son soutien constant. Ses conseils éclairés ont été essentiels pour l'aboutissement de ce travail. Je tiens à remercier le reste du jury, Dr. Souhila GHANEM, Dr. Nadia BATAT, et Dr. Houda EL BOUHISSI Epse BRAHAMI, pour leur précieuse contribution à l'évaluation de ce mémoire.

Mes remerciements vont également à mes collègues et amis qui ont partagé leurs idées et m'ont apporté un soutien moral tout au long de cette aventure académique.

Enfin, je tiens à exprimer ma reconnaissance envers l'ensemble de l'équipe pédagogique du Département d'Informatique de l'Université A. Mira de Béjaïa pour leur dévouement à l'excellence académique. Je dédie ce mémoire à ma famille mon pere, ma mere, et mes frere ABDALLAH et MOHAMMED et ma soeur WISSAM, à mes amis, et à toutes les personnes qui ont joué un rôle dans la réalisation de ce travail ilyes, aymen, aymen, raouf, abderazak, agheslan, rabeH, amine et ma meilleur amie sara. Votre encouragement et votre soutien ont été précieux et ont contribué à mon succès.

Table des matières

Dédicace	3
Remerciements	5
Table des matières	6
Table des figures	7
Liste des tableaux	10
1 Introduction	11
1.1 Contexte	11
1.2 Problématique	11
1.3 Contributions	11
1.4 Organization du mémoire	12
2 Concepts de base	13
2.1 Les systèmes de recommandations	13
2.2 Les smart cities	18
2.3 Les systèmes de transport intelligent	19
2.4 Les modèles de machine learning	21
3 état de l'art	25
3.1 Traveau connexes	25
3.2 Tableau comparatif	27
3.3 Discussion et comparaison	31
3.4 Conclusion	31
4 Contributions	33
4.1 Approche proposée	33
4.2 Architecture du système	35
5 Expérimentation	39
5.1 Description du Dataset	39
5.2 Environnement de developement	42
5.3 Les Interfaces de l'application	45
5.4 Evaluation	51
Bibliographie	57

Table des figures

2.1	Les quatre composants du système de recommandation	14
2.2	Un système de recommandation basé sur le contenu (adapté de [15]).	15
2.3	Un système de recommandation collaboratif (adapté de [15]).	16
2.4	Le système de recommandation hybride (adapté de [15]).	17
2.5	Conception d'hybridation monolithique (traduction de [15]).	18
2.6	Conception d'hybridation parallèle (traduction de [15]).	18
2.7	Conception d'hybridation parallèle Conception d'hybridation tubulaire (traduction de [15]).	19
4.1	L'architecture de systeme de recommandation.	36
5.1	Dataset des trajets enregistrés	40
5.2	Dataset après le prétraitement	41
5.3	Interface du navigateur Anaconda	43
5.4	Interface de jupyter notebook	44
5.5	Interface de l'application	46
5.6	Carte de l'application	47
5.7	La simulation de taxis	48
5.8	Epingler la position du départ de l'utilisateur	49
5.9	Epingler la position de destination de l'utilisateur	50
5.10	la recommandation des taxis	51
5.11	Temps du trajet prédit	52

List of Algorithms

1 Conception du Système de Recommandation (SR) pour les Taxis 35

Liste des tableaux

5.1 Performance des modèles de régression 51

Chapitre 1

Introduction

1.1 Contexte

De nos jours, on observe une augmentation importante du nombre d'internautes qui est estimé aujourd'hui à 4,9 milliards d'internautes dans le monde. De ce fait, les agglomérations sont devenues de plus en plus connectées, ce qui génère un volume important de données utilisateurs.

Ces données représentent une mine d'or pour les entreprises qui les utilisent pour la publicité et la recommandation de produits et services très ciblés pour les utilisateurs afin de réaliser un bénéfice mutuel : faciliter la vie de l'utilisateur et augmenter les bénéfices des entreprises.

La recommandation est une technique de l'intelligence artificielle qui permet, à partir d'un profil utilisateur, de recommander un item qui lui correspond. Elle est utilisée dans de nombreux domaines, à l'instar de la vente en ligne, la musique, les films, les jeux, les voyages, et le transport. La recommandation fournit une expérience de consommation personnalisée pour les utilisateurs, ce qui peut les aider à trouver plus facilement ce qu'ils cherchent et à découvrir de nouveaux produits qui les intéressent.

Les systèmes de recommandation sont largement utilisés pour aider les utilisateurs à découvrir de nouveaux produits, contenus ou services qui correspondent à leurs préférences.

1.2 Problématique

L'un des services qui peut réaliser un bénéfice mutuel en appliquant la recommandation est le service de transport. Dans les agglomérations, il est difficile de se déplacer, ce qui rend les services VTC (Voiture de transport avec chauffeur) un service crucial. Il existe plusieurs entreprises qui proposent des services VTC, ce qui peut être déroutant pour l'utilisateur. D'où la nécessité d'avoir un outil qui recommande le meilleur fournisseur pour arriver rapidement à la destination et à moindre coût.

Avoir une recommandation de transport optimale est un enjeu important dans n'importe quelle ville connectée où le temps est important et où rater une correspondance d'une minute peut causer un retard considérable, notamment avec le trafic routier très dense. D'où l'idée d'un système de recommandation pour les VTC.

1.3 Contributions

Dans ce mémoire, nous adressons la problématique de recommandation d'un véhicule appartenant à un fournisseur VTC parmi plusieurs concurrents qui existent. Nous proposons un système de recommandation hybride qui utilise la régression linéaire associée à une mesure de similarité basée sur la distance haversine pour prédire le temps de trafic associé à un VTC donné. En se basant sur un enregistrement de plusieurs courses

effectuées par les fournisseurs pendant une année, on recommande le meilleur VTC pour un point de prise et un point d'arrivée, c'est-à-dire le VTC qui fait le moins de temps entre les deux points.

Ainsi, nous résumons les contributions dans les points suivants :

- Etablissement d'un état de l'art couvrant les différentes techniques de recommandation existantes dans la littérature. Nous avons mis un focus sur les travaux de recommandation de transport dans les villes connectées.
- Proposition d'une nouvelle approche recommandation de transport basée sur la régression linéaire associée à la mesure de similarité distance de haversine.
- Implémentation de l'approche proposée et son évaluation. Nous avons développé une application Web pour la recommandation de transport basée sur le modèle de recommandation proposé et les services de géolocalisation de Google. L'application recommande le meilleur VTC pour un point de prise.

1.4 Organization du mémoire

La présente étude se structure en six chapitres, chacun contribuant à la compréhension et à l'approfondissement du thème de la recommandation de transport dans les villes intelligentes :

- Chapitre 1 : Introduction générale Ce premier chapitre offre une introduction générale à l'étude. Il présente le contexte et la problématique de recherche, mettant en évidence l'importance des systèmes de recommandation (SR) dans les villes intelligentes. Nous exposons également les objectifs visés par cette recherche, ainsi que la méthodologie adoptée pour atteindre ces objectifs.
- Chapitre 2 : Généralités sur la recommandation, le transport intelligent, les villes connectées et les modèles de machine learning Dans ce chapitre, nous fournissons un aperçu complet sur les systèmes de recommandation, le concept du transport intelligent, les caractéristiques des villes connectées, et les principaux modèles de machine learning utilisés dans le domaine. Cette mise en perspective permettra d'acquérir une base solide pour la compréhension des développements ultérieurs.
- Chapitre 3 : État de l'art des travaux de recommandation de transport Le troisième chapitre constitue une revue exhaustive des travaux de recommandation de transport existants. Nous analysons et comparons ces différentes approches afin de mettre en évidence leurs avantages et inconvénients respectifs. Cette analyse critique permettra de mieux cerner les lacunes existantes et d'identifier les opportunités pour notre système de recommandation.
- Chapitre 4 : Architecture et conception du système de recommandation Dans ce chapitre, nous présentons en détail l'architecture et la conception de notre propre système de recommandation de transport destiné aux villes intelligentes. Nous expliquons les choix technologiques et méthodologiques qui ont été faits pour la mise en œuvre de ce système, et nous exposons les différentes étapes suivies pour son développement.
- Chapitre 5 : Environnements logiciels, outils et interfaces de l'application Le cinquième chapitre traite des aspects techniques de notre application, y compris les environnements logiciels et les outils utilisés pour la mise en place du système de recommandation. Nous détaillons également les interfaces utilisateur développées, offrant aux utilisateurs une expérience conviviale et pratique.
- Chapitre 6 : Conclusion et perspectives Le dernier chapitre de ce mémoire clôture notre étude en présentant les conclusions tirées de nos recherches. Nous dressons un bilan du travail accompli tout au long du mémoire, en mettant en évidence les contributions, les découvertes et les limites du système de recommandation développé. Enfin, nous proposons des perspectives pour de futurs travaux, soulignant les opportunités d'amélioration et d'extension de notre recherche dans le domaine de la recommandation de transport dans les villes intelligentes.

Chapitre 2

Concepts de base

Introduction

Dans ce chapitre, nous présentons des généralités sur les systèmes de recommandation d'une façon générale. Nous présentons une définition pour les systèmes de recommandation, l'intuition derrière leur fonctionnement, et une classification pour les différents systèmes de recommandations. Ensuite, nous présentons des généralités sur le transport dans les villes intelligentes et les différentes méthodes d'apprentissage machine.

2.1 Les systèmes de recommandations

Les systèmes de recommandation sont des algorithmes informatiques permettant de recommander des produits, des services ou du contenu adapté à un profil utilisateur.

Le profil de l'utilisateur se constitue de facteurs pertinents qui peuvent être explicitement déclarés par l'utilisateur (exemple ses préférences) ou des données implicites comme l'historique de recherche et navigation.

Les facteurs caractérisant un profil utilisateur ne se limitent pas à l'utilisateur mais peuvent être étendus à d'autres sources comme les préférences des amis, des règles métier pour orienter la recommandation, des données démographiques ainsi que les tendances du marché et les préférences des profils similaires.

2.1.1 Les composants d'un système de recommandation

La plupart des systèmes de recommandation sont composés de quatre éléments clés : Le modèle utilisateur, le modèle objet, l'algorithme de recommandation, et le système d'évaluation de la recommandation. La Figure 2.1 montre l'interaction entre les différents composants d'un système de recommandation.

- Le modèle utilisateur : Ce composant se concentre sur la compréhension des préférences et des habitudes des utilisateurs. Les informations collectées pour chaque utilisateur peuvent inclure les produits précédemment consultés, achetés ou notés, ainsi que d'autres données telles que les données démographiques et les intérêts.
- Le modèle d'objet de recommandation : Ce composant se concentre sur la compréhension des produits ou des services qui sont recommandés. Les informations collectées pour chaque produit peuvent inclure les catégories, les caractéristiques, les coûts et les avis des utilisateurs.
- L'algorithme de recommandation : Ce composant utilise les informations collectées à partir du modèle utilisateur et d'objet de recommandation pour générer des recommandations. Il existe plusieurs algorithmes couramment utilisés, tels que les algorithmes de filtrage collaboratif, les algorithmes de filtrage basé sur les contenus et les algorithmes hybrides.

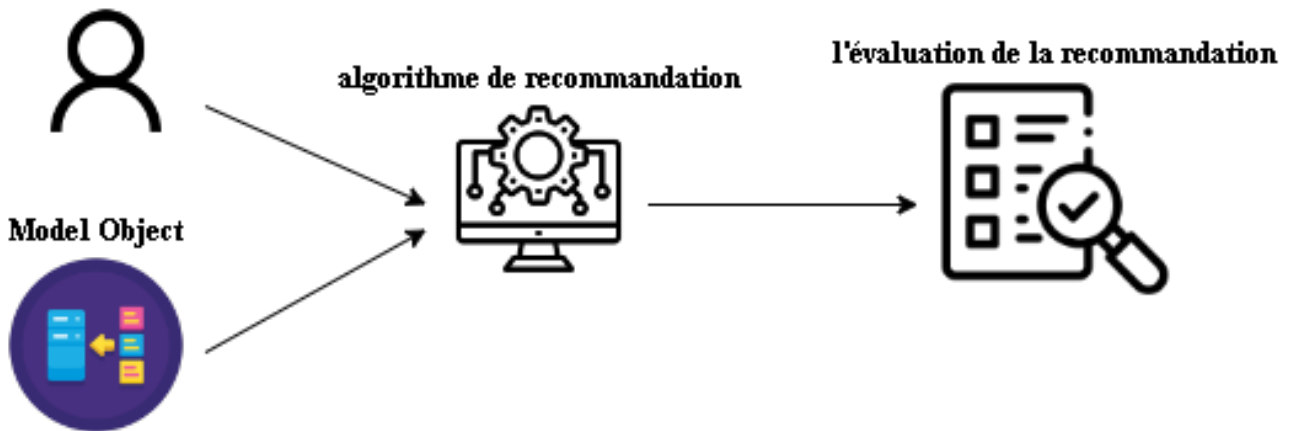
Model utilisateur

FIGURE 2.1 – Les quatre composants du système de recommandation

- L'évaluation des recommandations : Ce composant évalue la qualité des recommandations générées par l'algorithme de recommandation. Les mesures d'évaluation courantes incluent la précision, la rappel, la pertinence et la satisfaction des utilisateurs.

Les quatre composants du système de recommandation interagissent ensemble pour aider les utilisateurs à trouver des produits, des contenus ou des services qui correspondent à leurs préférences, ce qui améliore leur expérience utilisateur.

2.1.2 Les types des systèmes de recommandation

Il existe plusieurs types de systèmes de recommandation dans la littérature, selon la catégorie de l'algorithme de recommandation utilisé.

2.1.2.1 Filtrage basé sur le contenu

Le filtrage basé sur le contenu est une approche pour les systèmes de recommandation qui se concentre sur les caractéristiques des produits ou des services pour recommander des articles pertinents. Cette approche utilise les informations telles que les métadonnées des produits, les descriptions de produits, les images, les commentaires d'utilisateurs, etc. pour déterminer des similarités entre les produits et recommander des produits similaires à ceux que l'utilisateur a précédemment consultés ou achetés. Le profil de l'utilisateur est exprimé sous forme d'une liste d'intérêts basée sur les mêmes caractéristiques. La coïncidence entre les caractéristiques des éléments et le profil de l'utilisateur peut être mesurée de différentes manières :

- l'indice de Dice ou d'autres mesures de similarité [3].
- le TF-IDF (Term Frequency-Inverse Document Frequency) [25].
- les techniques basées sur la similarité des espaces vectoriels (les approches bayésiennes [23], les arbres de décision, etc.) couplées avec des techniques statistiques, lorsqu'il y a trop de mots-clés.

le filtrage basé sur le contenu est un type important d'approche pour les systèmes de recommandation qui peut aider les entreprises à recommander des produits pertinents pour les utilisateurs.

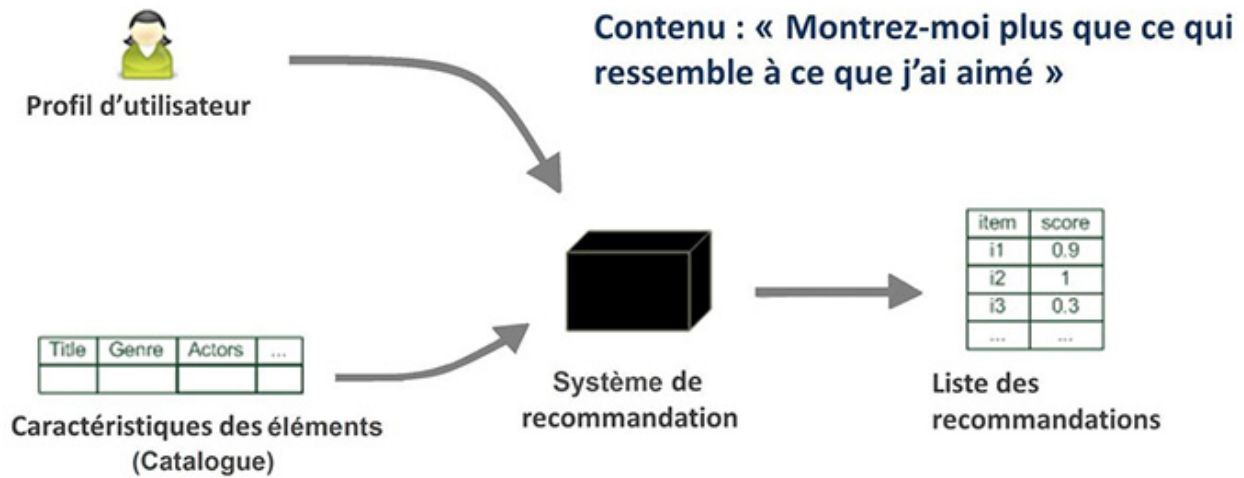


FIGURE 2.2 – Un système de recommandation basé sur le contenu (adapté de [15]).

2.1.2.2 Filtrage collaboratif

Les systèmes de recommandation basés sur le filtrage collaboratif utilisent les données historiques de l'utilisateur et les données d'autres utilisateurs pour recommander des produits ou des services pertinents. Ils peuvent être divisés en deux types : filtrage collaboratif basé sur les utilisateurs et filtrage collaboratif basé sur les articles.

Filtrage collaboratif le filtrage collaboratif est une technique importante pour la recommandation en ligne. Les systèmes de recommandation basés sur le filtrage collaboratif peuvent améliorer l'expérience de l'utilisateur en suggérant des éléments pertinents en fonction de leurs préférences et comportements antérieurs [26].

Le filtrage collaboratif basé sur les utilisateurs se concentre sur les préférences et les comportements des utilisateurs similaires pour recommander des éléments pertinents. Cette approche compare les préférences d'un utilisateur donné avec celles d'autres utilisateurs similaires pour trouver des correspondances.

Le filtrage collaboratif basé sur les articles se concentre sur les caractéristiques des articles eux-mêmes plutôt que sur les préférences des utilisateurs. Cette approche utilise des algorithmes pour trouver des articles similaires en se basant sur les caractéristiques telles que le contenu, la catégorie ou la date de publication.

2.1.2.3 Systèmes de recommandation hybrides

Les systèmes de recommandation hybrides combinent les algorithmes de filtrage collaboratif et de filtrage basé sur le contenu pour fournir des recommandations plus précises. Ils peuvent utiliser des informations sur les utilisateurs et les produits pour recommander des produits pertinents. La figure 2.4 illustre ce processus.

Le filtrage hybride utilise les avantages de chaque approche pour compenser les faiblesses des autres et produire des recommandations plus précises.

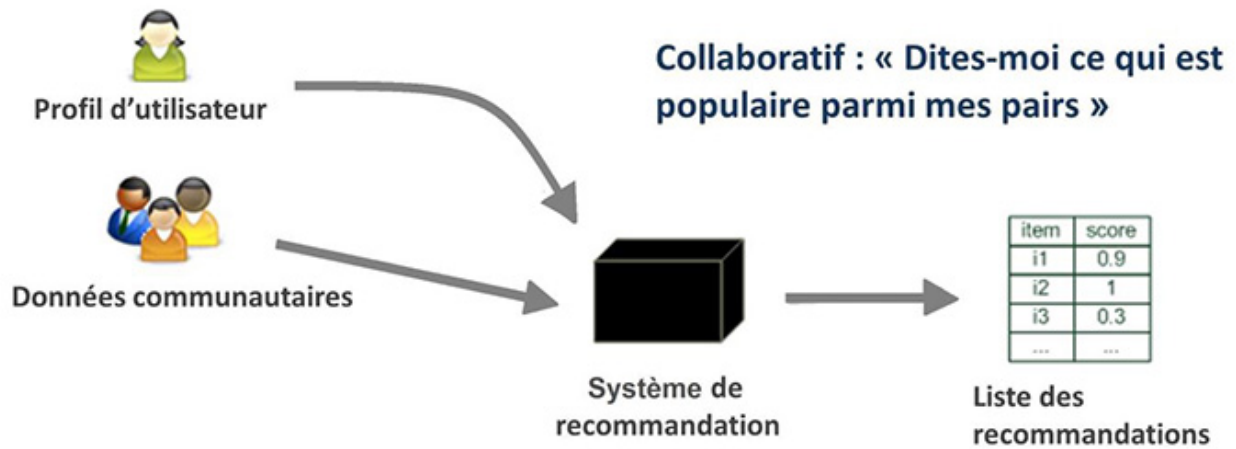


FIGURE 2.3 – Un système de recommandation collaboratif (adapté de [15]).

Il existe trois grandes catégories de combinaisons de systèmes de recommandation pour concevoir un système de recommandation hybride [6, 15] :

La combinaison monolithique (monolithic hybridization design) : Monolithique décrit une conception d'hybridation qui intègre les aspects de différentes stratégies de recommandation en un seul algorithme. Comme illustré sur la figure 5, différents systèmes de recommandation y contribuent puisque l'approche hybride utilise des données d'entrée additionnelles qui sont spécifiques à un autre algorithme de recommandation, ou bien les données d'entrée sont complétées par une technique et exploitées par une autre. Par exemple, un système de recommandation basé sur le contenu qui exploite également des données communautaires pour déterminer des similarités entre éléments relève de cette catégorie.

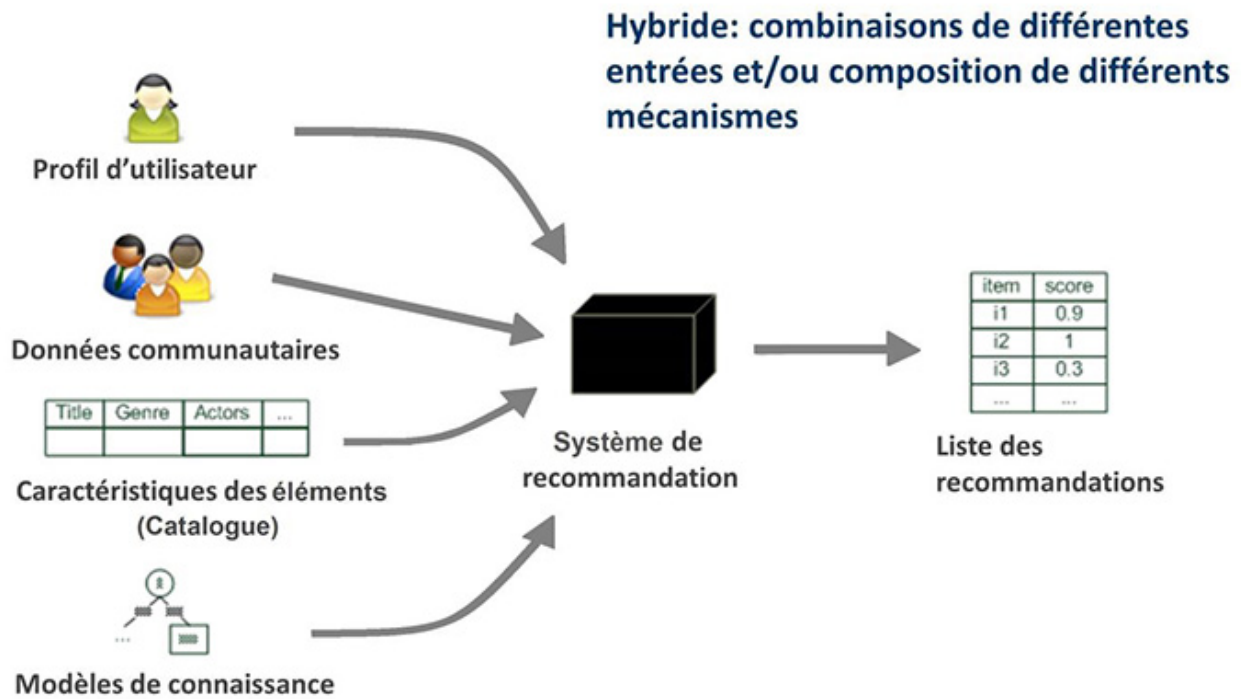


FIGURE 2.4 – Le système de recommandation hybride (adapté de [15]).

La combinaison parallèle (parallelized hybridization design) : ces approches hybrides nécessitent au moins deux mises en œuvre de recommandations séparées qui sont combinées en conséquence. Sur la base de leurs données d'entrée, les systèmes hybrides de recommandation parallèles fonctionnent indépendamment l'un de l'autre et produisent des listes de recommandations distinctes, comme illustré sur la figure 6. Dans une étape ultérieure d'hybridation, leurs sorties sont combinées en un ensemble final de recommandations.

La combinaison tubulaire (pipelined hybridization design) : Lorsque plusieurs systèmes de recommandation sont joints dans une architecture tubulaire, comme illustré par la figure 7, la sortie de l'un des systèmes de recommandation devient une partie des données d'entrée du système suivant.

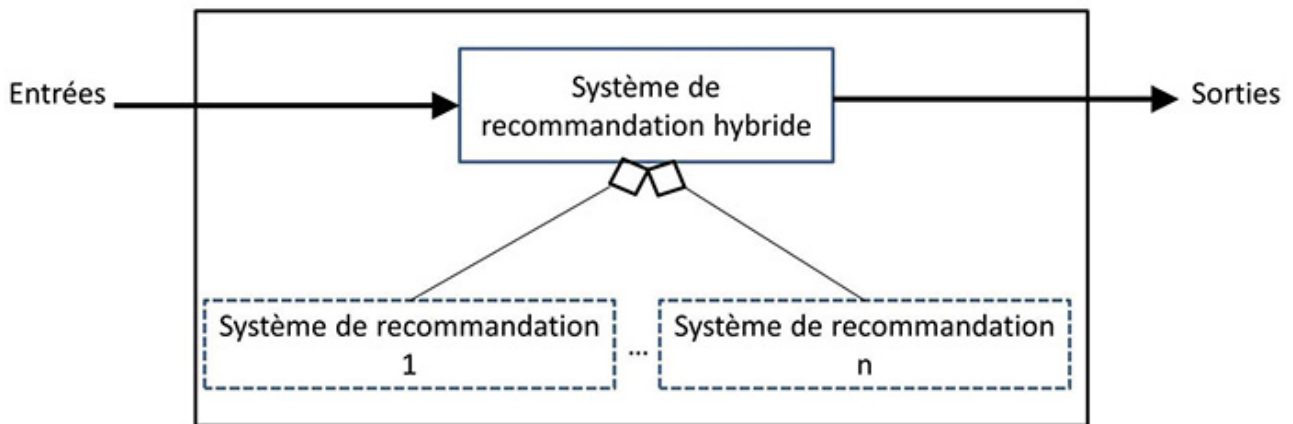


FIGURE 2.5 – Conception d’hybridation monolithique (traduction de [15]).

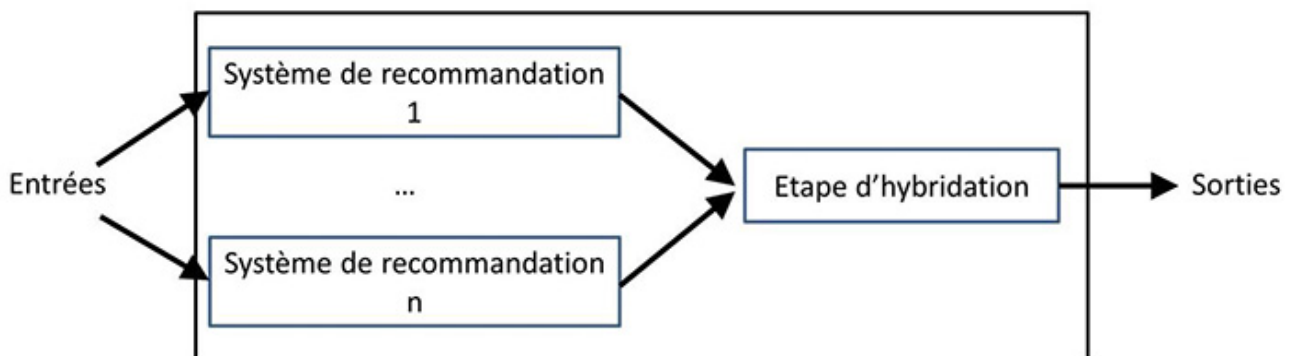


FIGURE 2.6 – Conception d’hybridation parallèle (traduction de [15]).

2.2 Les smart cities

2.2.1 Définition

Les smart cities sont des villes qui intègrent les technologies de l’information et de la communication (TIC) dans leur infrastructure et leur gouvernance pour améliorer la qualité de vie de leurs habitants. Ces villes utilisent des systèmes intelligents pour gérer efficacement les ressources, optimiser les services publics, favoriser la participation citoyenne et améliorer la durabilité environnementale [7] [2]. Les smart cities visent à créer un écosystème urbain connecté où la technologie et l’innovation sont au service de la mobilité, de la gestion des infrastructures, de l’énergie, de l’environnement et des interactions sociales, dans le but d’améliorer le bien-être et la compétitivité de la ville [14] [11]

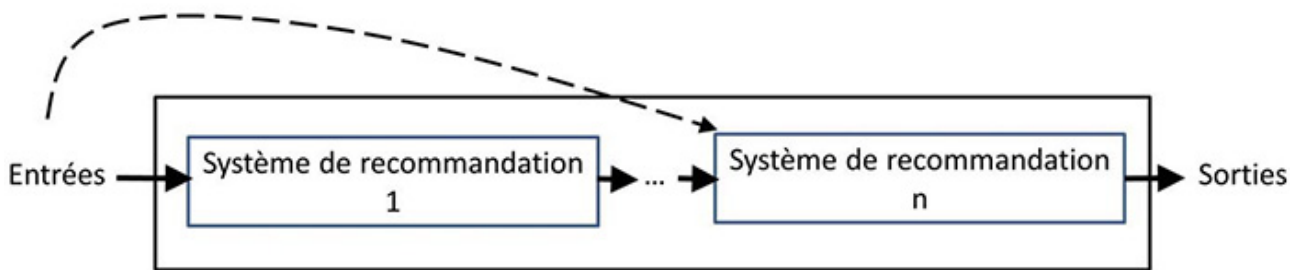


FIGURE 2.7 – Conception d'hybridation parallèle Conception d'hybridation tubulaire (traduction de [15]).

2.2.2 Les enjeux des Smart Cities

Les Smart Cities ont plusieurs enjeux et objectifs à satisfaire pour améliorer le bien-être des citoyens et assurer une durabilité pour l'environnement. Nous citons dans la suite un ensemble d'enjeux, critères et objectifs qu'une ville intelligente doit satisfaire et respecter : [des references!!!!!!!!!!!!!!!]

1. **Mobilité intelligente [20]** : Les Smart Cities utilisent des technologies avancées pour améliorer la gestion du transport urbain, y compris la gestion du trafic, les systèmes de transport en commun intelligents, le covoiturage, les véhicules électriques, etc.
2. **Gestion de l'énergie [29]** : Les Smart Cities intègrent des solutions pour optimiser la consommation d'énergie, favoriser les sources d'énergie renouvelable, mettre en place des réseaux électriques intelligents (smart grids), et encourager l'efficacité énergétique dans les bâtiments et les infrastructures.
3. **Environnement durable [29]** : Les Smart Cities s'engagent dans des initiatives de préservation de l'environnement en utilisant des technologies pour la gestion des déchets, la surveillance de la qualité de l'air et de l'eau, la conservation des ressources naturelles, et la promotion d'une planification urbaine respectueuse de l'environnement.
4. **Gouvernance et participation citoyenne [28]** : Les Smart Cities facilitent la participation citoyenne en utilisant des plateformes numériques pour recueillir les commentaires des résidents, encourager la transparence et la collaboration entre les gouvernements et les citoyens, et mettre en œuvre des services en ligne pour faciliter les interactions avec les administrations publiques.
5. **Sécurité publique [16]** : Les Smart Cities utilisent des technologies de surveillance, de détection et d'analyse pour améliorer la sécurité publique, y compris la vidéosurveillance intelligente, les systèmes d'alerte précoce, la gestion des urgences et la prévention de la criminalité.
6. **Qualité de vie [16, 28, 29]** : Les Smart Cities cherchent à améliorer la qualité de vie de leurs habitants en proposant des services innovants tels que les systèmes de santé connectés, les applications mobiles pour la gestion quotidienne, l'accès à l'éducation et à la culture, et la création d'espaces publics conviviaux.

Ces domaines d'application montrent la diversité des initiatives mises en place dans les Smart Cities pour créer des environnements urbains plus durables, efficaces et inclusifs.

2.3 Les systèmes de transport intelligent

2.3.1 Définition

les systèmes de transport intelligent (STI) se réfèrent à l'utilisation de technologies avancées, telles que les communications sans fil, les capteurs, les systèmes de géolocalisation et les algorithmes d'intelligence artificielle,

pour améliorer l'efficacité, la sécurité et la durabilité du système de transport. Ces technologies permettent une gestion intégrée des infrastructures de transport, des véhicules et des utilisateurs, en fournissant des informations en temps réel, en optimisant la circulation et en offrant des services intelligents aux voyageurs [20, 24].

2.3.2 Intégration des systèmes de transport intelligent avec les villes intelligentes

L'intégration des systèmes de transport intelligent (STI) avec les villes intelligentes constitue une étape essentielle pour optimiser la mobilité urbaine et améliorer la qualité de vie des citoyens. Les (STI) sont conçus pour améliorer l'efficacité et la sécurité des systèmes de transport en utilisant des technologies avancées telles que les capteurs, les communications sans fil et les systèmes d'information géographique. En les combinant avec les principes des villes intelligentes, qui visent à exploiter les technologies de l'information et de la communication pour une gestion plus efficace des infrastructures urbaines, on peut créer des synergies puissantes [17]. L'intégration des STI avec les villes intelligentes permet d'optimiser la planification des itinéraires, la gestion du trafic, la réduction des émissions de CO₂ et la promotion de modes de transport durables tels que le covoiturage et les transports en commun [19].

Le domaine du transport intelligent joue un rôle crucial dans le développement des Smart Cities. Les technologies et les concepts innovants sont utilisés pour améliorer la mobilité urbaine et créer des systèmes de transport plus efficaces et durables. Voici quelques-uns des aspects clés du transport intelligent dans les Smart Cities :

- **Gestion du trafic** : Les Smart Cities utilisent des systèmes de gestion du trafic avancés qui intègrent des capteurs, des caméras de surveillance, des algorithmes d'optimisation et des systèmes de signalisation intelligents pour réduire les congestions et améliorer la fluidité du trafic.
- **Transport en commun intelligent** : Les Smart Cities investissent dans des systèmes de transport en commun intelligents qui utilisent des technologies telles que les cartes de paiement sans contact, les applications mobiles pour la planification d'itinéraires et les horaires en temps réel, les systèmes de suivi des véhicules, et les stations connectées pour offrir des services plus efficaces et adaptés aux besoins des usagers.
- **Mobilité partagée** : Les Smart Cities encouragent l'utilisation de modes de transport partagés tels que le covoiturage, les vélos en libre-service, les scooters électriques, et les services de partage de voitures. Ces solutions de mobilité partagée réduisent la congestion, les émissions de gaz à effet de serre et favorisent une utilisation plus efficace des infrastructures de transport.
- **Véhicules connectés et autonomes** : Les Smart Cities intègrent les véhicules connectés et autonomes dans leurs infrastructures de transport. Cela permet une communication entre les véhicules, les infrastructures routières et les systèmes de gestion du trafic, améliorant ainsi la sécurité, la fluidité du trafic et l'efficacité énergétique.
- **Analyse des données et prise de décision** : Les Smart Cities collectent et analysent de grandes quantités de données liées au transport, telles que les données de capteurs, les données de localisation et les données de trafic en temps réel. Ces analyses permettent de prendre des décisions éclairées pour l'optimisation des réseaux de transport, la planification des infrastructures et l'amélioration de l'expérience des utilisateurs.

Le transport intelligent dans les Smart Cities contribue à réduire les problèmes de congestion, à améliorer l'efficacité énergétique, à promouvoir des modes de transport durables et à offrir une meilleure qualité de vie aux citoyens.

2.4 Les modèles de machine learning

2.4.1 Concepts et principes fondamentaux

Le machine learning est un domaine de l'intelligence artificielle qui se concentre sur le développement d'algorithmes et de modèles capables d'apprendre à partir des données et de prendre des décisions ou d'effectuer des prédictions sans être explicitement programmés [4, 13]. Cette approche permet aux machines de généraliser à partir des exemples donnés et d'adapter leur comportement en fonction des données d'entrée.

Les principes fondamentaux de statistiques, d'optimisation et de modélisation probabiliste jouent un rôle essentiel dans le domaine du machine learning. Les statistiques permettent de comprendre la distribution des données et d'effectuer des inférences à partir de celles-ci. L'optimisation est utilisée pour ajuster les paramètres des modèles de manière à minimiser une fonction de coût ou à maximiser une mesure de performance. La modélisation probabiliste permet de capturer l'incertitude inhérente aux données et aux prédictions du modèle [12].

2.4.2 Types de problèmes résolus par le machine learning

Le machine learning est utilisé pour résoudre une grande variété de problèmes dans de nombreux domaines. Il peut être appliqué à des problèmes de classification, de régression, de clustering, de détection d'anomalies, de recommandation, de traitement du langage naturel, de vision par ordinateur, etc. [4, 13, 21].

Dans les tâches de classification, le machine learning permet de catégoriser des données en classes prédéfinies. Par exemple, il peut être utilisé pour classer des e-mails en spam ou en non-spam, pour détecter des fraudes financières ou pour prédire le diagnostic d'une maladie à partir de symptômes.

En ce qui concerne la régression, le machine learning permet de prédire des valeurs continues à partir de données d'entrée. Par exemple, il peut être utilisé pour prédire le temps d'un trajet des taxis en fonction de ses caractéristiques.

Le clustering est une technique de machine learning utilisée pour regrouper des données similaires en groupes ou en clusters. Cela peut être utile pour segmenter les clients en fonction de leurs comportements d'achat ou pour découvrir des motifs dans de grands ensembles de données non étiquetées.

La détection d'anomalies vise à identifier les observations rares ou inhabituelles dans un ensemble de données. Cela peut être appliqué, par exemple, pour détecter des fraudes financières, des pannes d'équipement ou des comportements anormaux dans un réseau informatique.

Les systèmes de recommandation utilisent le machine learning pour proposer des suggestions personnalisées à partir des préférences ou du comportement d'un utilisateur. Ces systèmes sont largement utilisés dans les plateformes de streaming de musique et de vidéo, les sites de commerce électronique et les réseaux sociaux.

Le traitement du langage naturel (NLP) et la vision par ordinateur sont également des domaines importants du machine learning. Ils permettent respectivement de comprendre et de générer du texte, ainsi que de reconnaître, classer et analyser des images et des vidéos.

En somme, le machine learning offre une gamme d'outils et de techniques permettant de résoudre des problèmes complexes dans de nombreux domaines, en exploitant le pouvoir de l'apprentissage à partir des données.

2.4.3 Les algorithmes de machine learning les plus couramment utilisés

Les algorithmes de machine learning sont des méthodes et des modèles utilisés pour extraire des informations à partir des données et réaliser des tâches d'apprentissage automatique. Voici une sélection des algorithmes les plus couramment utilisés dans le domaine :

Régression linéaire : L'algorithme de régression linéaire [4] est utilisé pour modéliser et prédire une variable continue en fonction d'autres variables indépendantes. Il s'agit d'un modèle simple mais puissant qui trouve de nombreuses applications dans divers domaines.

Réseaux de neurones : Les réseaux de neurones [?, 12] sont des modèles inspirés du fonctionnement du cerveau humain. Ils sont capables d'apprendre à partir de grandes quantités de données et sont particulièrement efficaces pour des tâches telles que la classification, la reconnaissance d'images et la prédiction.

Arbres de décision : Les arbres de décision [22] sont des modèles de classification et de régression qui prennent des décisions en fonction des caractéristiques des données. Ils sont faciles à interpréter et à visualiser, ce qui en fait des outils populaires pour l'analyse de données.

Forêts aléatoires : Les forêts aléatoires [13] sont des ensembles d'arbres de décision qui combinent les prédictions de plusieurs modèles pour obtenir des résultats plus précis. Ils sont robustes, faciles à utiliser et performants pour des problèmes de classification et de régression.

Machines à vecteurs de support (SVM) : Les SVM [4] sont des algorithmes d'apprentissage supervisé qui sont utilisés pour la classification et la régression. Ils sont basés sur la recherche de la meilleure séparation linéaire ou non linéaire entre les différentes classes.

K-plus proches voisins (KNN) : L'algorithme KNN [9] est une méthode simple et intuitive qui attribue une classe à un point de données en se basant sur les classes de ses voisins les plus proches. Il est largement utilisé pour la classification et la recommandation.

Naïve Bayes : Les classificateurs Naïve Bayes [22] sont basés sur le théorème de Bayes et supposent une indépendance conditionnelle entre les caractéristiques. Ils sont efficaces pour la classification de textes, la détection de spam et d'autres tâches de catégorisation.

Ces algorithmes représentent seulement une petite sélection parmi les nombreux algorithmes de machine learning existants.

2.4.4 Prétraitement des données pour le machine learning

Avant d'appliquer des algorithmes de machine learning, il est souvent nécessaire de prétraiter les données. Cela peut inclure des étapes telles que le nettoyage des données, la gestion des valeurs manquantes, la normalisation, la réduction de dimension et la sélection de caractéristiques [13].

2.4.5 Apprentissage supervisé versus apprentissage non supervisé : Différences et applications

L'apprentissage supervisé et l'apprentissage non supervisé sont deux approches fondamentales du machine learning, chacune ayant ses propres caractéristiques et applications spécifiques. Dans cette section, nous examinerons les différences entre ces deux types d'apprentissage et explorerons certaines de leurs applications respectives.

L'apprentissage supervisé consiste à entraîner un modèle à partir de données d'entraînement étiquetées [4] [?], où les exemples sont associés à des étiquettes ou des catégories connues. Le modèle apprend à prédire des étiquettes pour de nouvelles instances en se basant sur les exemples étiquetés qu'il a déjà vus. Les algorithmes d'apprentissage supervisé les plus couramment utilisés incluent les arbres de décision, les réseaux neuronaux, les machines à vecteurs de support (SVM) et les méthodes de régression.

En revanche, l'apprentissage non supervisé consiste à trouver des structures et des modèles intrinsèques dans les données non étiquetées [13] [27], sans utiliser d'étiquettes ou de catégories préexistantes. Les algorithmes d'apprentissage non supervisé visent à découvrir des regroupements (clustering), à réduire la dimensionnalité, à détecter des anomalies ou à extraire des caractéristiques représentatives. Des techniques telles que la classification non supervisée, la réduction de dimension, l'analyse en composantes principales (PCA) et l'algorithme des k-moyennes sont largement utilisées dans le cadre de l'apprentissage non supervisé.

Les applications de l'apprentissage supervisé et non supervisé sont diverses et dépendent du contexte spécifique. L'apprentissage supervisé est souvent utilisé dans des tâches de prédiction, de classification et de régression, telles que la détection de fraude, la recommandation de produits, la reconnaissance d'images et la prédiction de prix. D'autre part, l'apprentissage non supervisé est souvent appliqué pour découvrir des structures

cachées dans les données, comme la segmentation de clients, la détection de groupes similaires, l'exploration de données et la visualisation de données.

Il convient de noter que les frontières entre l'apprentissage supervisé et non supervisé peuvent parfois être floues, et des approches hybrides peuvent également être utilisées pour combiner les avantages des deux méthodes. Par exemple, les techniques de prétraitement des données non supervisées peuvent être utilisées avant l'apprentissage supervisé pour améliorer les performances du modèle.

2.4.6 Applications du machine learning dans le domaine du transport

Le machine learning offre des opportunités prometteuses pour améliorer les systèmes de transport et résoudre les défis complexes auxquels ils font face. Les chercheurs et les praticiens du domaine ont exploré diverses applications du machine learning dans le secteur du transport, ouvrant la voie à des avancées significatives. Dans cette section, nous examinerons plusieurs domaines spécifiques du transport où le machine learning a été utilisé avec succès.

Prédiction de la demande de transport : L'un des défis majeurs dans la planification du transport est de prédire avec précision la demande future. Le machine learning a été appliqué pour développer des modèles de prédiction de la demande basés sur des données historiques [4]. Ces modèles peuvent aider les opérateurs de transport à optimiser l'allocation des ressources et à améliorer l'efficacité des systèmes de transport [13].

Optimisation du trafic et de la gestion des flux : Le machine learning est utilisé pour analyser les données en temps réel provenant de diverses sources telles que les capteurs de trafic, les caméras de surveillance et les données de localisation des véhicules [12]. Ces analyses permettent de prédire les congestions, d'optimiser les feux de signalisation et de mettre en œuvre des stratégies de gestion du trafic intelligentes [22].

Systèmes de recommandation pour les trajets : Les techniques de machine learning sont utilisées pour développer des systèmes de recommandation de trajets personnalisés [?]. Ces systèmes prennent en compte les préférences individuelles des utilisateurs, les conditions de trafic actuelles et les informations contextuelles pour proposer des itinéraires optimaux [?].

Maintenance prédictive des infrastructures de transport : Le machine learning est utilisé pour analyser les données de surveillance et les données de maintenance des infrastructures de transport [27]. En utilisant ces données, des modèles prédictifs peuvent être construits pour détecter les problèmes potentiels, prévoir les défaillances et planifier la maintenance de manière proactive [?].

Prédiction des temps de trajet : Le machine learning est utilisé pour prédire les temps de trajet en fonction de différents facteurs tels que l'heure de la journée, les conditions météorologiques et les événements spéciaux [?]. Ces prédictions peuvent être utilisées pour informer les utilisateurs des transports en commun et les conducteurs afin d'optimiser leurs itinéraires [9].

Conclusion

Dans ce chapitre, nous avons fourni un aperçu global des concepts de base liés à notre problématique à savoir : les systèmes de recommandation, les villes intelligentes, le transport intelligent, et les principaux modèles de machine learning utilisés dans le domaine. Le chapitre suivant sera consacré à une revue des travaux de littérature sur les systèmes de recommandation des transport intelligents.

Chapitre 3

état de l'art

Introduction

Les smart cities sont des villes intelligentes qui utilisent les technologies de l'information et de la communication pour améliorer la qualité de vie de leurs citoyens. Dans ce contexte, les systèmes de recommandation de transport intelligent sont des outils précieux pour aider les usagers à trouver les meilleurs itinéraires et modes de transport adaptés à leurs préférences et au contexte actuel. Dans cet état de l'art, nous allons explorer les différentes techniques utilisées dans la recherche pour proposer des systèmes de recommandation de transport intelligent, les jeux de données disponibles, ainsi que les résultats obtenus.

3.1 Traveau connexes

Mohamed Yacine Gheraibia [10] propose un framework utilisant l'algorithme de machine learning pour organiser la transportation des biens dans les smart cities, en utilisant un système de recommandation basé sur des critères tels que les émissions de CO2, la consommation de carburant et la congestion routière.

Le framework proposé est divisé en trois phases : la collecte des informations, l'apprentissage et la prédiction/recommandation. Pour rendre le système de transport plus efficace et facile à utiliser pour les personnes, deux systèmes de recommandation ont été développés : un système adaptatif d'interface utilisateur personnalisée pour le transport (PATRASH) et un système de recommandation pour les applications de transport intelligent. En outre, le cadre utilise l'analyse en composantes principales (PCA) pour réduire la dimensionnalité des données, puis l'algorithme des k-means pour regrouper les données en clusters.

Le cadre proposé utilise des algorithmes d'apprentissage automatique pour mieux organiser le transport de marchandises dans les villes intelligentes. Le cadre recommande le transporteur optimal aux clients en fonction des trois critères.

Yang Liu, Cheng Lyu, Zhiyuan Liu et Jinde Cao [18] traitent le développement des services de planification de voyages rendus possibles par les smartphones et d'internet mobile. le modèle se concentre sur la navigation de transport multimodal et les avantages qu'elle offre par rapport à la navigation de transport unimodal.

L'article décrit la formulation du problème pour recommander des modes de transport en se basant sur des données de comportement et historiques des utilisateurs.

Les données sont collectées par un fournisseur de service de navigation, Baidu Maps, et comprennent des enregistrements de requêtes, d'affichage et de clics, tous reliés à des horodatages et à des identifiants de session.

L'article fournit une vue d'ensemble des services de planification de voyages rendus possibles par les smartphones et d'internet mobile. Il décrit également les fonctionnalités spécifiques créées pour la recommandation

de modes de transport, notamment une table pivot pour tenir compte des variations de longueur des modes de transport candidats renvoyés par Baidu Maps et pour s'assurer que tous les modes de transport potentiels sont couverts. Les auteurs proposent une méthode d'embedding de graphe et ils ont montré que cette méthode a amélioré les performances de recommandation de transport multimodal en utilisant des données de comportement historiques des utilisateurs.

Fanyou Wua, Cheng Lyub, Yang Liuc [30] proposent un cadre de recommandation de modes personnalisé pour les systèmes de transport multimodal. Ce cadre combine des systèmes de recommandation avec l'ingénierie des transports pour recommander des modes de déplacement aux utilisateurs.

Les principales contributions comprennent le développement d'un cadre conceptuel pour la recommandation proactive de mode de déplacement basé sur les préférences des utilisateurs et les informations sur l'état de la circulation en temps réel, la conception d'une méthode de balayage incrémental avec plusieurs fenêtres temporelles pour extraire des caractéristiques multi-échelles à partir des comportements des utilisateurs, et le développement d'une structure de comportement hiérarchique pour atténuer la charge de calcul causée par la grande taille des données. Le document propose également une classe de méthodes pour le système de recommandation personnalisé, où la fonction objective et la sortie peuvent être conçues selon les exigences réelles. La méthode proposée peut aider à améliorer l'utilisation des transports en commun, atténuer la congestion du trafic et réduire la pollution environnementale en recommandant des modes de déplacement personnalisés aux utilisateurs.

Ayat Abedalla et al. [1] présentent un modèle de recommandation de transport multimodal sensible au contexte en utilisant un ensemble de données public fournies par le défi de recommandation de transport multimodal sensible au contexte.

Les auteurs ont utilisé un ensemble de données contenant des données historiques sur le comportement des utilisateurs et des attributs d'utilisateurs collectées à partir de Baidu Map. Les données ont été réparties à 80 pour la formation et à 20 pour la validation. Le modèle MTRecS-DLT combine deux modèles : l'apprentissage profond et le modèle d'arbre. Le CNN est utilisé pour apprendre des motifs spatiaux dans les données comportementales des utilisateurs, tandis que le modèle d'arbre est utilisé pour capturer les interactions de caractéristiques et les non-linéarités dans les données d'attributs d'utilisateurs. Les modèles ont été évalués sur la base de leur score F1-pondéré sur un ensemble de validation et un ensemble de test.

Le modèle proposé a été évalué et comparé à plusieurs autres modèles. Le meilleur modèle sur l'ensemble de test était l'ensemble de CNN et XGBoost avec un score F1-pondéré de 0,68898702. Le deuxième meilleur modèle était l'ensemble de CNN, CNN-LSTM et XGBoost. Les auteurs concluent que le processus d'analyse et d'extraction de caractéristiques est crucial pour améliorer les résultats, et que l'utilisation de méthodes d'ensemble peut améliorer les performances. L'algorithme XGBoost s'est révélé être un algorithme puissant qui a surpassé les modèles d'apprentissage profond dans ce problème.

A A Borodinov, V V Myasnikov [5] présentent les aspects théoriques et algorithmiques pour la construction d'un système de recommandation personnalisé pour les utilisateurs de transports publics en commun. Les auteurs se concentrent sur l'identification et la formalisation du concept de "préférences de l'utilisateur".

Leur proposition est organisée en trois parties principales. La première partie fournit des informations sur les données et les définitions utilisées dans l'analyse des services de transport public dans une application mobile appelée "Pribyvalka-63". La deuxième partie propose une solution au problème de détermination des arrêts préférés des utilisateurs à certaines coordonnées spatio-temporelles. La troisième partie souligne les défis des systèmes de recommandation pour le routage multimodal et suggère quelques solutions possibles.

Les auteurs proposent une approche basée sur le calcul des estimations et l'estimation non paramétrique de la densité de probabilité de Parzen pour résoudre le problème de détermination des arrêts préférés des utilisateurs. L'algorithme pour résoudre le problème implique le calcul des valeurs pour tous les arrêts d'un ensemble et leur classement par ordre décroissant pour former une permutation qui fournit la solution. Pour résoudre le problème

de "démarrage à froid", les auteurs suggèrent de compléter l'ensemble précédent avec des valeurs initiales pour les week-ends et les jours ouvrables précédant la date de lancement du système.

L'article conclut avec les résultats d'études expérimentales sur des données réelles obtenues à l'aide de l'application mobile "Pribyvalka-63". Les auteurs proposent une des façons possibles de décrire et de résoudre le problème de détermination des préférences individuelles des utilisateurs de transports publics en commun et de créer un système de recommandation personnalisé. Le texte met également en évidence les défis des systèmes de recommandation pour le routage multimodal et suggère quelques solutions possibles.

3.2 Tableau comparatif

Nous avons établi un tableau comparatif des différentes méthodes utilisées dans chacun des travaux lus. Ce tableau résume les résultats obtenus des approches proposées. Ce tableau comparatif est constitué de 8 colonnes comportant les caractéristiques suivantes :

- 1ere colonne "catégorie" : représente le type de recommandation pour laquelle appartiennent ces approches.
- 2eme colonne "Approche" : représente l'approche elle même (l'auteur ou les auteurs avec l'année d'édition).
- 3eme colonne "Titre" : c'est le titre de l'article en question.
- 4ème colonne "Source de données" : indique les données en entrée.
- 5eme colonne "Sorties" : indique les/le résultat de l'approche.
- 6eme colonne "Used technique" : explicite toutes les techniques utilisées pour aboutir au en résultat (les types de filtrages, les outils de calculs de similarités...).
- 7eme colonne "Thechnique utilisées" : indique le nom de l'outil logiciel utilisé dans le cas où l'approche a été implémentée.
- 8eme colonne "Paramètres de recommandation" : indique les paramètre de recommandation qui ont été utilisées

Catégorie de l'approche	Approche	Titre	Source de données (données en entrées)	Sorties (le résultat de l'approche)	Techniques utilisées	Outils utilisés	Paramètres de recommandation
Approche unsupervisée	Mohamed Yacine Gheraibia	Intelligent Mobile-Based Recommender System Framework for Smart Freight Transport	Données sur les émissions de CO2, la consommation de carburant et la congestion routière	Transporteur optimal recommandé	Clustering-based approach, Analyse en Composantes Principales (PCA), méthode des k-means	PATRASH (système de recommandation basé sur l'historique de l'utilisateur des transports en commun)	CO2, consommation de carburant, congestion routière, évaluation globale de la voiture et la position
Approche unsupervisée	[Yang Liu, Cheng Lyu, Zhiyuan Liu, Jinde Cao b]	"Exploring a large-scale multi-modal transportation recommendation system"	Données de comportement utilisateur collectées par Baidu Maps, comprenant des enregistrements de requêtes, d'affichage et de clics reliés à des horodatages et des identifiants de session	Prédiction du mode de transport choisi par un utilisateur donné un enregistrement de requête Q, un enregistrement d'affichage P et des attributs d'utilisateur U	Graphe biparti pour l'extraction d'embeddings, Node2vec pour la linéarisation du graphe	Python, Tensor-Flow	Placement du mode de voyage le plus préféré par l'utilisateur en haut de la liste de recommandations

Approche supervisée	[Fanyou Wua, Cheng Lyub, Yang Liuc]	"A personalized recommendation system for multi-modal transportation systems"	Données d'interaction utilisateur historiques, données de requête utilisateur, données de candidats, données de sélection et données d'attributs d'utilisateur	Choix de mode de transport recommandé	Balayage incrémental avec plusieurs fenêtres temporelles, structure de comportement hiérarchique, algorithme amélioré de boosting d'arbre LightGBM	LightGBM (algorithme d'apprentissage), outils de traitement de données	Paramètres de trafic en temps réel, préférences des utilisateurs
Approche supervisée	[Ayat Abedalla, Ali Fadel, Ibraheem Tuffaha, Hani Al-Omari, Mohammad Omari, Malak Abdullah, Mahmoud Al-Ayyoub]	MTRecS-DLT : Multi-Modal Transport Recommender System using Deep Learning and Tree Models	Données historiques sur le comportement des utilisateurs et des attributs d'utilisateurs collectées à partir de Baidu Map	Recommandation du mode de transport le plus approprié	Réseau de neurones à convolution (CNN), modèle d'arbre, algorithme XGBoost	Python, TensorFlow, XGBoost	Les modèles ont été évalués sur la base de leur score F1-pondéré sur un ensemble de validation et un ensemble de test. Le meilleur modèle sur l'ensemble de test était l'ensemble de CNN et XGBoost avec un score F1-pondéré de 0,68898702.

Approche un supervisée	[A A Borodin, V V Myasnikov]	Aspects théoriques et algorithmiques de la construction d'un système de recommandation personnalisé pour les utilisateurs de transports publics en commun	Application mobile Pribyvalka-63 faisant partie du service tosamara.ru	Détermination des préférences de l'utilisateur dans un contexte spatial-temporel spécifique, tels que la définition des arrêts préférés et des "correspondances de transport" préférées	Formulations informelles et formelles des problèmes de détermination des préférences de l'utilisateur, approche basée sur le calcul des estimations et l'estimation non paramétrique de la densité de probabilité de Parzen	Application mobile "Pribyvalka-63"	Coordonnées spatio-temporelles, identifiants uniques, paramètres de demande, types d'itinéraires, véhicules, qualité du service
Approche supervisée	Notre approche	Transportation recommendation in smart cities	Données sur des trajets est déjà fait NYC taxi trip	ID de taxi qui a le plus court temps, cours moins chère, tarif est unique	Méthode hybride qui combine entre la régression linéaire et la distance haversine	Python, Flask, HTML, CSS, JavaScript	Cordonnées de l'utilisateur, la destination et les taxi

3.3 Discussion et comparaison

La discussion et la comparaison des travaux sur les systèmes de recommandation de transport montrent des objectifs similaires, qui visent à améliorer l'expérience des utilisateurs dans les transports en commun/multimodaux. Cependant, chaque article met l'accent sur des aspects spécifiques de la recommandation.

L'article de Mohamed Yacine Gheraibia met l'accent sur la recommandation de modes de transport alternatifs en se basant sur des critères environnementaux et d'efficacité. Cela peut être utile pour encourager l'utilisation de modes de transport plus durables et réduire l'impact environnemental.

Fanyou Wu et al. mettent l'accent sur la prise en compte des préférences des utilisateurs et de l'état du trafic en temps réel. Cela permet de proposer des recommandations qui tiennent compte des contraintes personnelles des utilisateurs ainsi que des conditions actuelles du trafic.

Ayat Abedalla et al. proposent une recommandation sensible au contexte en utilisant des attributs utilisateurs. Cela signifie que les recommandations prennent en compte des informations spécifiques sur les utilisateurs, telles que leur historique de trajets, leurs préférences et leurs besoins, afin de fournir des recommandations plus pertinentes et personnalisées.

AA Borodinov et VV Myasnikov se concentrent sur la personnalisation en fonction des préférences spatio-temporelles des utilisateurs. Ils cherchent à recommander des itinéraires en tenant compte non seulement des préférences des utilisateurs, mais aussi des conditions spatio-temporelles, telles que l'heure de la journée et la localisation géographique.

Les recherches sur les systèmes de recommandation de transport ont démontré l'utilisation de différentes techniques d'apprentissage automatique, telles que le clustering, les arbres de décision et les réseaux de neurones. Cependant, certains articles ont proposé des approches spécifiques pour améliorer la recommandation en utilisant des techniques plus avancées. Par exemple, Yang Liu et al. ont utilisé l'embedding de graphe pour représenter les relations entre les différents modes de transport et les utilisateurs, ce qui leur a permis de capturer des informations complexes sur les préférences et les besoins des utilisateurs. D'autre part, AA Borodinov et VV Myasnikov ont utilisé l'estimation de densité de probabilité de Parzen pour mieux comprendre les préférences des utilisateurs et recommander des itinéraires adaptés en se basant sur ces estimations.

Les jeux de données utilisés dans ces recherches varient en fonction des sources disponibles. Certains articles ont utilisé des données sur le fret pour comprendre les besoins logistiques, d'autres ont utilisé les données clients de Baidu Maps pour étudier les préférences des utilisateurs, et certains ont utilisé les données de l'application mobile Pribyvalka-63 pour obtenir des informations sur les habitudes de déplacement des utilisateurs. Cette diversité de jeux de données permet d'explorer et de tester les systèmes de recommandation dans différents contextes et avec différentes sources d'informations.

Les résultats des différentes recherches montrent que les systèmes de recommandation de transport proposés peuvent réellement améliorer l'expérience des utilisateurs dans les transports en commun. Les recommandations fournies permettent aux utilisateurs de choisir des itinéraires qui correspondent à leurs préférences et au contexte actuel, ce qui peut réduire le temps de trajet, minimiser les transferts, et fournir une expérience de voyage plus agréable.

En conclusion, bien que les recherches sur les systèmes de recommandation de transport partagent des objectifs communs, elles proposent des approches variées en fonction des cas d'usage et des données disponibles. Les techniques d'apprentissage automatique et d'intelligence artificielle semblent prometteuses pour améliorer l'efficacité et la personnalisation des systèmes de recommandation de transport. Ces recherches ouvrent de nouvelles perspectives pour l'amélioration des transports dans les smart cities, en permettant de mieux répondre aux besoins des utilisateurs et d'optimiser l'utilisation des infrastructures de transport.

3.4 Conclusion

En conclusion, les systèmes de recommandation de transport intelligent sont un domaine de recherche en pleine croissance. Les résultats des différents travaux montrent que les techniques d'apprentissage automatique

et d'intelligence artificielle sont prometteuses pour améliorer l'efficacité et la personnalisation de ces systèmes. Les différentes approches proposées dans la recherche permettent d'explorer et de tester les systèmes de recommandation dans différents contextes et avec différentes sources d'informations. Ces recherches ouvrent de nouvelles perspectives pour l'amélioration des transports dans les smart cities, en offrant aux usagers des solutions personnalisées pour leurs besoins de mobilité.

Chapitre 4

Contributions

Introduction

Les villes intelligentes ont de plus en plus besoin de mettre en place des systèmes de transport intelligents pour améliorer la mobilité des citoyens et réduire les problèmes de congestion. Dans ce contexte, les systèmes de recommandation de transport jouent un rôle crucial en proposant des trajets personnalisés et efficaces aux utilisateurs. Cependant, la recommandation de trajets dans les villes intelligentes nécessite de prendre en compte à la fois la durée des trajets et la distance géographique entre les points de départ et d'arrivée.

Dans ce chapitre, nous présentons une approche complète et détaillée pour le système de recommandation de transport dans les villes intelligentes. Notre approche repose sur l'utilisation de la régression linéaire pour prédire la durée des trajets et la distance Haversine pour calculer la distance géographique entre les points de départ et d'arrivée. Nous combinons ces deux mesures pour recommander le fournisseur de transport le plus rapide et le taxi le plus proche à l'utilisateur, offrant ainsi une solution intégrée et pratique pour les déplacements.

Nous commençons par détailler les différentes étapes de notre approche, qui comprennent le prétraitement des données, Ensuite, nous entraînons plusieurs modèles de régression, tels que le GradientBoostingRegressor, SVR, LinearRegression et RandomForestRegressor, sur notre jeu de données historiques de trajets. Nous évaluons les performances de ces modèles en utilisant la mesure de la racine de l'erreur quadratique moyenne (RMSE) et sélectionnons le modèle de régression linéaire comme modèle principal pour notre système de recommandation en raison de sa meilleure performance.

Enfin, nous concluons en soulignant les avantages de notre approche hybride combinant la régression linéaire et la distance Haversine. Cette approche permet de recommander efficacement les trajets les plus rapides et les taxis les plus proches, en prenant en compte à la fois la durée des trajets et la distance géographique. Notre système de recommandation de transport contribue ainsi à faciliter les déplacements dans les villes intelligentes.

4.1 Approche proposée

Dans le contexte d'une ville intelligente où les utilisateurs souhaitent se déplacer d'une position à une destination donnée, notre système de recommandation de transport vise à optimiser l'expérience de mobilité en proposant les fournisseurs de services taxi les plus adaptés à chaque situation. Pour ce faire, nous exploitons un ensemble de données de trajets déjà effectués, qui comprend des informations sur les distances parcourues et les fournisseurs de services associés.

L'idée fondamentale de notre approche est de fournir aux utilisateurs une recommandation de taxi basée sur deux critères principaux : le temps de trajet prévu et la proximité géographique entre l'utilisateur et le taxi recommandé.

L'algorithme de recommandation se déroule en plusieurs étapes, tel que décrit ci-dessous :

1. Capturer des coordonnées de l'utilisateur et de sa destination : Nous récupérons les coordonnées géographiques de l'utilisateur (latitude et longitude) ainsi que celles de sa destination, ces données représentent la requête utilisateur.
2. Calcul de la distance entre les coordonnées de l'utilisateur et de sa destination : En utilisant la formule de Haversine, nous calculons la distance géographique entre les coordonnées de l'utilisateur et de sa destination.
3. Normalisation de la distance calculée : La distance calculée est ensuite normalisée entre 0 et 1 à l'aide de la technique de normalisation 'Min-Max'. Cela nous permet de représenter la distance de manière relative et d'obtenir une valeur normalisée pour une comparaison plus précise.
4. Prédiction de la durée de trajet : Nous entraînons un modèle de régression linéaire en utilisant un ensemble de caractéristiques pertinentes, telles que la distance calculée précédemment et l'identifiant du fournisseur de services de taxis. Ce modèle nous permet de prédire la durée de trajet prévue en fonction de ces caractéristiques.
5. Sélection du fournisseur de services de taxis le plus rapide : Nous identifions le fournisseur de services de taxis ayant la durée de trajet la plus courte prédite. Ceci est réalisé en comparant les durées prédites pour chaque fournisseur et en sélectionnant celui qui offre le temps de trajet le plus court.
6. Calcul de la distance entre les taxis du fournisseur recommandé et l'utilisateur : Pour déterminer le taxi le plus proche de l'utilisateur parmi ceux proposés par le fournisseur recommandé, nous calculons la distance entre chaque taxi et l'utilisateur en utilisant la formule de Haversine.
7. Recommandation du taxi le plus rapide et le plus proche : En utilisant les distances calculées précédemment, nous identifions le taxi qui se trouve le plus près de l'utilisateur et qui est proposé par le fournisseur recommandé. Nous recommandons ensuite ce taxi à l'utilisateur, en lui fournissant également la durée de trajet prédite.

Pour atteindre cet objectif, nous avons développé une approche soutenue par un outil logiciel qui permet de comprendre les différentes étapes de notre approche. L'algorithme 1 décrit la séquence des différentes étapes du processus de recommandation.

Dans ce qui suit, nous allons présenter les étapes de notre approche en détails :

4.1.1 Collecte des coordonnées

Nous collectons les coordonnées géographiques de l'utilisateur (latitude et longitude) ainsi que celles de sa destination, ces données représentent la requête utilisateur.

4.1.2 Calcul de la distance

En utilisant la formule de Haversine [8], nous calculons la distance géographique entre les coordonnées de l'utilisateur et de sa destination. La formule de haversine permet de déterminer la plus courte distance entre deux points sur une sphère, à partir de leurs longitudes et latitudes.

4.1.3 Normalisation de la distance

La distance calculée est normalisée entre 0 et 1 à l'aide de la technique de normalisation 'Min-Max'.

4.1.4 Prédiction de la durée de trajet

Nous entraînons un modèle de régression linéaire en utilisant un ensemble de caractéristiques pertinentes, telles que la distance calculée précédemment et l'identifiant du fournisseur de services de taxis. Ce modèle nous permet de prédire la durée de trajet prévue en fonction de ces caractéristiques.

Algorithm 1 Conception du Système de Recommandation (SR) pour les Taxis

```

1: Entrée :
2: Coordonnées de du depart de l'utilisateur  $C_{dep}$  (latitude et longitude),
3: Coordonnées de destination  $C_{arr}$  (latitude et longitude),
4:
5: Sortie :
6: Recommandation du taxi le plus proche du fournisseur le plus rapide,
7: Durée de trajet prédite,
8:
9: Début
10: Capturer les coordonnées de l'utilisateur (latitude et longitude) et de destination désirée,
11: Calculer la distance entre les coordonnées de l'utilisateur et sa destination en utilisant la distance haversin,
12: Normaliser la distance calculée entre 0 et 1 en utilisant La normalisation ' $Min - Max$ ',
13: Appliquer un algorithme de machine learning (modele de regression linéaire) pour prédire un fournisseur de
    service offrant une durée minimale,
14: Récupérer l'ID du fournisseur correspondant et la durée prédite associée,
15: Pour tous taxi  $i$  du fournisseur ID,
16:     Calculer la distance entre taxi  $i$  et  $C_{dep}$ ,
17:     Comparer les distances,
18:     Récupérer la plus petite distance,
19: Fin Pour,
20: Afficher le taxi le plus proche et le plus rapide et la durée prédite,
21: Fin.

```

4.1.5 Sélection du fournisseur de services de taxis le plus rapide

Nous identifions le fournisseur de services de taxis ayant la durée de trajet la plus courte prédite.

4.1.6 Calcul de la distance entre les taxis et l'utilisateur

Nous calculons la distance entre chaque taxi du fournisseur recommandé et l'utilisateur en utilisant la formule de Haversine.

4.1.7 Recommandation du taxi le plus rapide et le plus proche

En utilisant les distances calculées précédemment, nous identifions le taxi qui se trouve le plus près de l'utilisateur parmi ceux proposés par le fournisseur recommandé. Nous recommandons ensuite ce taxi à l'utilisateur, en lui fournissant également la durée de trajet prédite.

Avec cette approche, nous cherchons à améliorer l'efficacité et la satisfaction des utilisateurs dans leurs déplacements en ville en leur proposant les taxis les plus adaptés à leur situation, à la fois en termes de temps de trajet prévu et de proximité géographique.

4.2 Architecture du système

Ce processus de recommandation de taxi est basé sur des données de trajet et d'emplacement, et utilise la regression linéaire combiné avec la distance haversine pour recommander le taxi le plus rapide et le plus proche à l'utilisateur.

Plus précisément, la regression linéaire commence par s'entraîner en utilisant les données des fournisseurs disponibles, qui peuvent inclure des informations sur les temps de trajet et les emplacements des taxis. Ensuite,

à partir des informations de trajet fournies par l'utilisateur, le moteur de recherche récupère la position de destination de l'utilisateur et recommande le fournisseur de taxi le plus rapide.

Une fois que le fournisseur de taxi le plus rapide a été identifié, l'algorithme de recommandation de taxis entre en jeu. Cet algorithme récupère l'identifiant du fournisseur de taxi le plus rapide, ainsi que les emplacements des taxis et de l'utilisateur. À partir de ces informations, l'algorithme recommande le taxi le plus rapide et le plus proche à l'utilisateur.

En résumé, ce processus de recommandation de taxi utilise des données de trajet et d'emplacement pour recommander le taxi le plus rapide et le plus proche à l'utilisateur, en combinant les avantages d'un moteur de recherche avec un algorithme de recommandation de taxis.

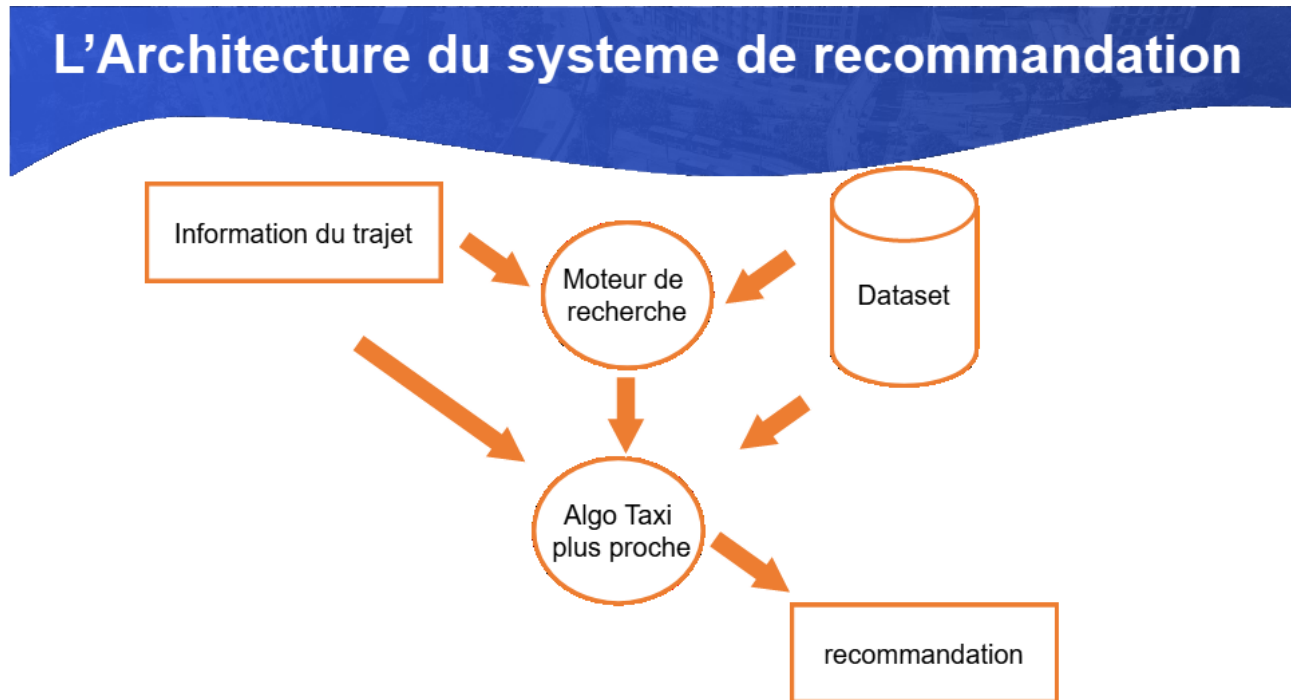


FIGURE 4.1 – L'architecture de système de recommandation.

Conclusion

Dans ce chapitre, nous avons présenté notre approche de recommandation de transport basée sur l'optimisation de l'expérience de mobilité dans le contexte d'une ville intelligente. Notre système de recommandation de taxi vise à proposer les fournisseurs de services les plus adaptés à chaque situation en se basant sur deux critères principaux : le temps de trajet prévu et la proximité géographique entre l'utilisateur et le taxi recommandé.

Nous avons décrit en détail les différentes étapes de notre approche, depuis la collecte des coordonnées de l'utilisateur et de sa destination jusqu'à la recommandation du taxi le plus rapide et le plus proche. Pour ce faire, nous avons utilisé des techniques telles que le calcul de la distance géographique avec la formule de Haversine, la normalisation des distances et l'utilisation d'un modèle de régression linéaire pour prédire la durée de trajet.

Notre approche repose sur l'exploitation d'un ensemble de données de trajets déjà effectués, ce qui permet d'améliorer la précision des recommandations en se basant sur des informations réelles. De plus, nous avons

développé un outil logiciel qui met en œuvre les différentes étapes du processus de recommandation, facilitant ainsi sa mise en pratique dans un environnement réel.

En proposant des recommandations de taxis adaptées à chaque utilisateur, notre approche vise à améliorer l'efficacité et la satisfaction des utilisateurs dans leurs déplacements en ville. En combinant le temps de trajet prévu et la proximité géographique, nous cherchons à optimiser la qualité de l'expérience de mobilité des utilisateurs en leur offrant des options de transport plus efficaces.

Chapitre 5

Expérimentation

Introduction

Le cinquième chapitre constitue une étude de cas de notre contribution sur un dataset pour la prédiction de la durée des trajets en taxi. Nous abordons la description du dataset utilisé dans notre travail. Le dataset représente les données d'enregistrement des trajets de taxi jaune de New York City (NYC) de l'année 2016. En plus de la description du dataset, nous présentons également l'environnement de développement utilisé pour la réalisation de ce projet. Cela inclut des informations sur le matériel utilisé, ainsi que des détails sur les logiciels utilisés, ainsi que les langages utilisés pour la création de l'application. Enfin, nous exposerons et expliquerons les différentes interfaces de notre application,

5.1 Description du Dataset

Le dataset représente les données d'enregistrement des trajets de taxi jaune de New York City de 2016, disponible dans Big Query sur la plateforme Google Cloud. Les données ont été initialement publiées par la NYC Taxi et Limousine Commission (TLC). la taille du DataSet est l'ordre de 1 500 000 (1 458 644) enregistrements de trajets.

Le dataset est décrit par les attributs suivant (figure 5.1) :

1. id : un identifiant unique pour chaque trajet
2. vendor_id : un identifiant indiquant le fournisseur associé à l'enregistrement du trajet
3. pickup_datetime : date et heure du départ du taxi
4. dropoff_datetime : date et heure d'arrivée du taxi
5. passenger_count : le nombre de passagers dans le taxi (valeur saisie par le chauffeur)
6. pickup_longitude : la longitude où le compteur a été enclenché(Départ)
7. pickup_latitude : la latitude où le compteur a été enclenché(Départ)
8. dropoff_longitude : la longitude où le compteur a été désactivé (Arrivée)
9. dropoff_latitude : la latitude où le compteur a été désactivé (Arrivée)
10. store_and_fwd_flag : cette étiquette indique si l'enregistrement du trajet a été stocké en mémoire dans le véhicule avant d'être envoyé au fournisseur
11. trip_duration durée du trajet en secondes

Nous avons effectué un prétraitement sur le dataset en supprimant les redondances et les colonnes non pertinentes(Figure 5.2), telles que :

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	40.767937	-73.964630	40.765602	N	455
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	40.738564	-73.999481	40.731152	N	663
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	40.763939	-74.005333	40.710067	N	2124
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040	40.719971	-74.012268	40.706718	N	429
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	40.793209	-73.972923	40.782520	N	435
...
1458639	id2376096	2	2016-04-08 13:31:04	2016-04-08 13:44:02	4	-73.982201	40.745522	-73.994911	40.740170	N	778
1458640	id1049543	1	2016-01-10 07:35:15	2016-01-10 07:46:10	1	-74.000946	40.747379	-73.970184	40.796547	N	655
1458641	id2304944	2	2016-04-22 06:57:41	2016-04-22 07:10:25	1	-73.959129	40.768799	-74.004433	40.707371	N	764
1458642	id2714485	1	2016-01-05 15:56:26	2016-01-05 16:02:39	1	-73.982079	40.749062	-73.974632	40.757107	N	373
1458643	id1209952	1	2016-04-05 14:44:25	2016-04-05 14:47:43	1	-73.979538	40.781750	-73.972809	40.790585	N	198

1458644 rows × 11 columns

FIGURE 5.1 – Dataset des trajets enregistrés

- pickup_longitude
- pickup_latitude
- dropoff_longitude
- dropoff_latitude
- pickup_datetime
- dropoff_datetime
- store_and_fwd_flag
- passenger_count

Un nouveau attribut est rajouté au dataset cet attribut represente la distance entre la poission du depart et l'arrivée du trajet, cet attribut a été calculé en fonction des quatres attributs supprimés pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude.

	vendor_id	trip_duration	distance
0	2	455	0.001208
1	1	663	0.001455
2	2	2124	0.005146
3	2	429	0.001197
4	2	435	0.000958
...
1458639	2	778	0.000987
1458640	1	655	0.004875
1458641	2	764	0.006306
1458642	1	373	0.000880
1458643	1	198	0.000914

1458644 rows × 3 columns

FIGURE 5.2 – Dataset après le prétraitement

5.2 Environnement de developement

5.2.1 Environnement matériel

Pc DELL

Processeur : Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz

Mémoire installée(RAM) :8,00 Go.

Type de Système Système d'exploitation : 64 bits, processeur x64. Windows 10 Professionnel N.

5.2.2 Environnement logiciel

Anaconda Anaconda est une plate-forme open-source et un logiciel gratuit largement utilisé dans le domaine de la science des données et du développement d'applications basées sur l'apprentissage automatique. Il offre un environnement complet pour coder en Python ou en R, ainsi que des outils et des bibliothèques préinstallées.

Avec plus de 250 packages et plus de 7500 packages supplémentaires disponibles, Anaconda permet une installation facile des bibliothèques nécessaires à différents projets. L'utilisation du système de gestion de paquets conda facilite la gestion et la mise à jour de ces packages.

Anaconda propose également des environnements de développement intégrés (IDE) conviviaux, qui sont des plates-formes ou des applications permettant de développer et de gérer le code de manière plus efficace. Cette plate-forme simplifie l'installation et la configuration de Python, des IDE et des bibliothèques essentielles en une seule installation, évitant ainsi la nécessité d'installer séparément chaque composant.

L'interface graphique utilisateur (GUI) d'Anaconda offre une expérience conviviale et intuitive. Elle permet de naviguer facilement à travers les différents environnements et de gérer les bibliothèques et les fonctionnalités disponibles. L'interface du navigateur Anaconda, représentée dans la figure 5.1, offre une vue d'ensemble pratique des fonctionnalités et des outils disponibles.

Jupyter notebook Jupyter Notebook (figure 5.2), anciennement connu sous le nom d'IPython, est une application web interactive open-source largement utilisée dans le domaine de la science des données et de l'apprentissage automatique. Il offre un environnement convivial pour l'exploration, l'analyse et la modélisation de données.

L'application Jupyter Notebook permet aux utilisateurs de créer des documents appelés "notebooks" qui peuvent contenir du code exécutable, des équations mathématiques, des visualisations et du texte explicatif. Ces notebooks sont organisés en cellules, où chaque cellule peut contenir du code (Python, Julia, R, etc.) ou du texte au format Markdown.

Les notebooks Jupyter offrent une expérience interactive, permettant aux utilisateurs d'exécuter du code directement dans les cellules et de voir les résultats immédiatement. Cela facilite l'exploration des ensembles de données, l'expérimentation avec différents algorithmes d'apprentissage automatique et la création de visualisations interactives.

De plus, Jupyter Notebook permet aux utilisateurs de documenter leur travail en intégrant du texte explicatif, des images et des équations mathématiques au sein du notebook. Cela facilite la communication des résultats, des méthodologies et des conclusions aux collègues et aux parties prenantes.

Les notebooks Jupyter peuvent être exportés dans différents formats tels que HTML, PDF et LaTeX, ce qui permet de partager facilement les résultats et les analyses réalisées avec d'autres personnes.

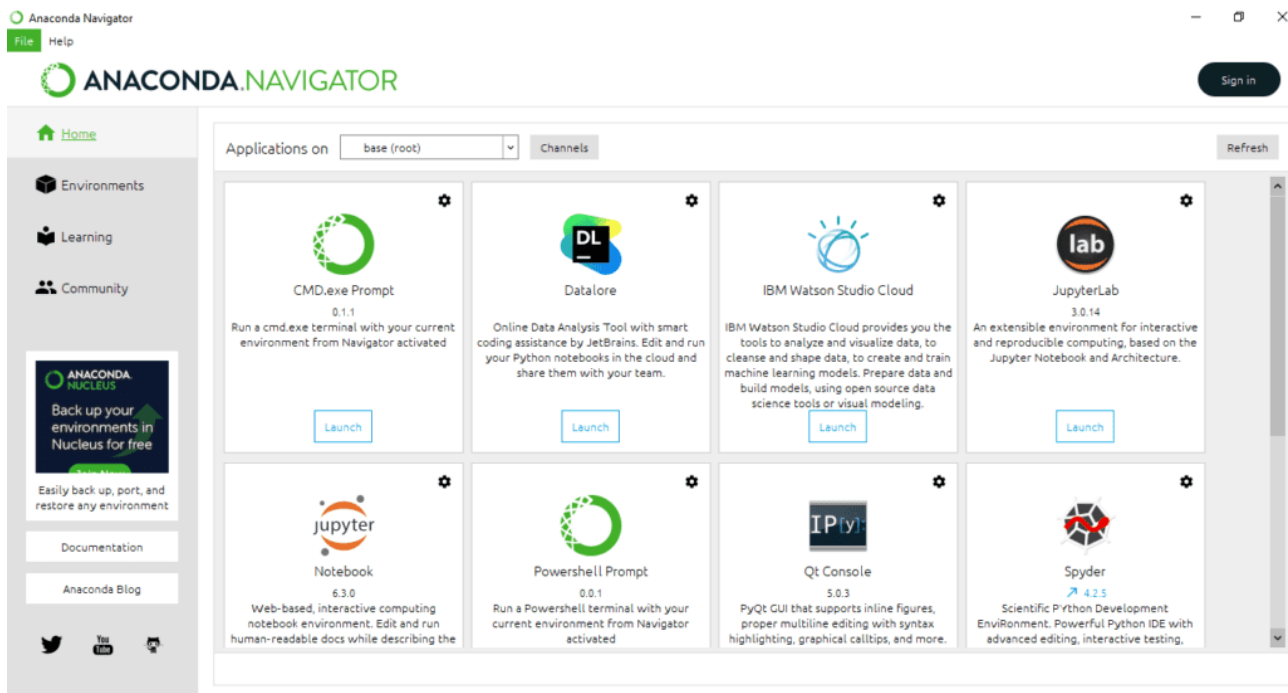


FIGURE 5.3 – Interface du navigateur Anaconda

Langage de programmation Python Python est un langage de programmation orienté objet de haut niveau, open source et largement utilisé dans le domaine du développement logiciel. Il est réputé pour sa simplicité, sa lisibilité et sa syntaxe claire, ce qui en fait un langage très populaire parmi les programmeurs.

Créé par Guido van Rossum en 1991, Python tire son nom de l'émission de télévision "Monty Python's Flying Circus". Il est largement utilisé dans divers domaines, tels que le développement d'applications, la création de services web, la génération de code, la métaprogrammation, le scripting et l'automatisation.

Python est un langage interprété, ce qui signifie qu'il ne nécessite pas de compilation préalable pour exécuter le code. Cela permet une approche de développement plus rapide et plus itérative, où les modifications de code peuvent être immédiatement testées et mises en œuvre.

Python est disponible en deux versions principales, Python 2.x et Python 3.x. Bien qu'il y ait une certaine concurrence entre les deux versions, Python 3.x est considéré comme la version recommandée et la plus récente. Cependant, en raison de la nécessité de migration des bibliothèques existantes, les deux versions coexistent actuellement.

Python possède une vaste bibliothèque standard qui offre de nombreuses fonctionnalités prêtes à l'emploi, ce qui facilite le développement de divers types d'applications. De plus, Python dispose d'une communauté active de développeurs qui contribuent à l'écosystème Python en créant et en partageant des bibliothèques tierces, ce qui étend les capacités du langage.

Bibliothèques de Python Les bibliothèques de Python sont des collections de modules contenant du code pré-écrit qui permettent aux développeurs d'implémenter des fonctionnalités spécifiques en réduisant le temps de codage. Dans le contexte du traitement des données et de l'apprentissage automatique, Python offre un large

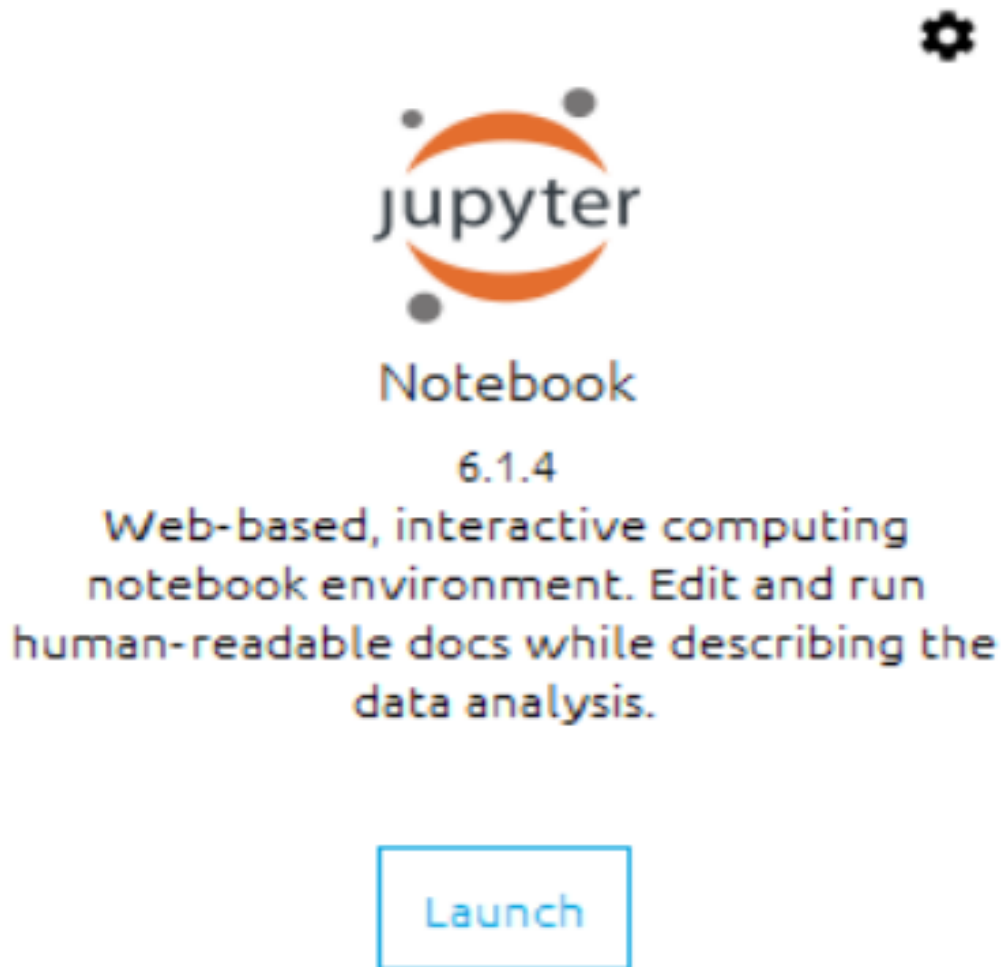


FIGURE 5.4 – Interface de jupyter notebook

éventail de bibliothèques qui fournissent des outils pour des tâches telles que les mathématiques, l'exploration des données et la visualisation.

Parmi ces bibliothèques, nous utilisons les suivantes dans notre code :

Pandas Pandas est une bibliothèque populaire utilisée pour les structures et les analyses de données de haut niveau. Elle simplifie la lecture de fichiers de différents formats tels que CSV, Excel, JSON, etc., et permet de manipuler et de transformer les données de manière efficace. Pandas utilise une structure de données appelée "DataFrame" pour stocker et organiser les données, ce qui facilite les opérations de fusion, de filtrage et de regroupement.

Scikit-Learn Scikit-Learn est une bibliothèque essentielle pour l'apprentissage automatique en Python. Elle offre une large gamme d'outils et de modèles pour la modélisation statistique et le machine learning.

Scikit-Learn fournit une interface cohérente pour effectuer des tâches telles que la classification, la régression, le regroupement et la réduction de la dimensionnalité. Elle permet également d'évaluer et de comparer les performances des modèles.

Flask Flask est une bibliothèque légère et flexible utilisée pour créer des applications web en Python. Elle facilite la création de routes, la gestion des requêtes et des réponses HTTP, ainsi que la génération de pages HTML dynamiques. Flask est souvent utilisé pour créer des API web pour les applications basées sur le modèle client-serveur.

En utilisant ces bibliothèques, notre code est capable de charger et de manipuler des données à partir de fichiers, de diviser les données en ensembles d'entraînement et de test, d'entraîner des modèles de régression linéaire, de calculer des distances géographiques, de normaliser les données, et de prédire la durée des trajets en fonction des coordonnées géographiques fournies.

Visual studio code Visual Studio Code (VS Code) est un environnement de développement intégré (IDE) populaire utilisé pour la programmation dans de nombreux langages, y compris Python. Il offre une interface conviviale et de nombreuses fonctionnalités avancées pour améliorer l'efficacité et la productivité des développeurs.

En utilisant Visual Studio Code ainsi que les langages HTML, CSS et JavaScript, nous sommes en mesure de créer une interface utilisateur interactive et conviviale pour notre application web, où les utilisateurs peuvent saisir les coordonnées géographiques, obtenir des prédictions de durée de trajet et afficher les résultats de manière conviviale.

HTML (Hypertext Markup Language) est le langage de balisage standard utilisé pour la création de pages web. Dans le contexte de notre développement, nous utilisons HTML pour créer des modèles de pages web qui permettent d'afficher les résultats de nos prédictions et d'interagir avec les utilisateurs.

CSS (Cascading Style Sheets) est un langage de feuille de style utilisé pour décrire la présentation et le style des documents HTML. Nous utilisons CSS pour appliquer des styles visuels à nos modèles HTML, tels que la mise en forme du texte, la couleur, la disposition des éléments et les effets visuels.

JavaScript JavaScript est un langage de programmation de haut niveau principalement utilisé pour ajouter des fonctionnalités interactives aux pages web. Dans notre développement, nous utilisons JavaScript pour la manipulation dynamique des éléments de la page, l'interaction avec les utilisateurs et l'envoi de requêtes asynchrones aux serveurs.

Leaflet Leaflet est une bibliothèque JavaScript open-source largement utilisée pour la création de cartes interactives et réactives sur le web. Elle fournit une solution légère et flexible pour intégrer des cartes interactives dans des applications web et mobiles.

Avec Leaflet, vous pouvez afficher des cartes avec des tuiles (fond de carte) provenant de différents fournisseurs, tels que OpenStreetMap, Mapbox, ou encore des sources personnalisées. La bibliothèque prend en charge le zoom, le défilement et le déplacement fluide de la carte, ainsi que l'ajout de marqueurs, de polygones, de lignes et d'autres formes géométriques pour représenter des points d'intérêt ou des données géospatiales.

5.3 Les Interfaces de l'application

L'interface principale de l'application est représentée dans la Figure 5.3. Elle comprend une carte géographique où les utilisateurs peuvent interagir et effectuer différentes actions. Les éléments clés de l'interface sont les suivants :

- Une carte géographique interactive.
- Une simulation de taxis avec 10 taxis pour le fournisseur 1 et 10 taxis pour le fournisseur 2.

Trois boutons :

- "Départ" : permet à l'utilisateur de sélectionner l'emplacement de départ en cliquant sur le bouton puis en cliquant sur la carte.
- "Arrivée" : permet à l'utilisateur de sélectionner l'emplacement d'arrivée en cliquant sur le bouton puis en cliquant sur la carte.
- "Recommander" : permet à l'utilisateur de prédire la durée du trajet et de recommander le taxi le plus proche du fournisseur le plus rapide, en saisissant l'emplacement de départ et d'arrivée, puis en cliquant sur le bouton "Recommander".

Une zone de texte :

- Temps du trajet Prédit : une prédiction du temps du trajet recommandé en secondes.

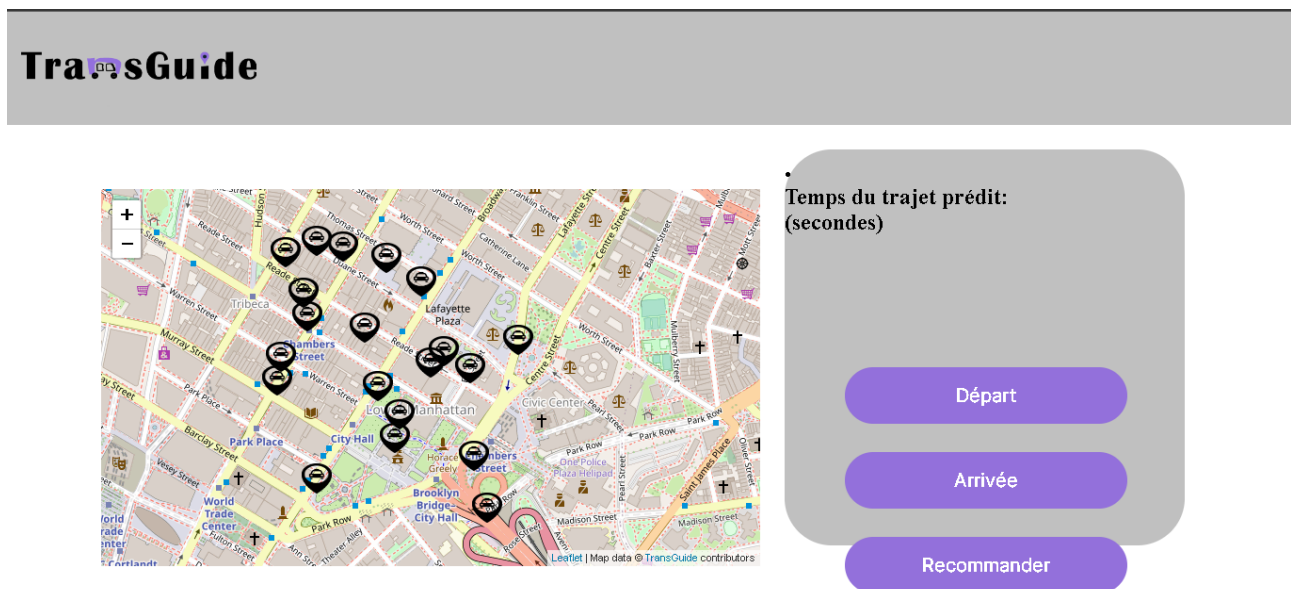


FIGURE 5.5 – Interface de l'application

5.3.1 Carte géographique interactive

La carte géographique interactive utilisée dans notre application est basée sur OpenStreetMap. OpenStreetMap est un projet collaboratif visant à créer une carte libre et gratuite du monde entier, à la manière d'un wiki cartographique. Il permet à quiconque de visualiser, d'éditer et d'utiliser les données géographiques.

OpenStreetMap est construit à partir de données géographiques collectées par des contributeurs du monde entier. Les données comprennent des informations sur les routes, les bâtiments, les points d'intérêt, les limites administratives et bien d'autres éléments géographiques. Ces données sont collectées à l'aide de GPS, de photographies aériennes, de cartes papier et d'autres sources.

La carte OpenStreetMap est affichée dans notre application web en utilisant une bibliothèque JavaScript appelée Leaflet. Leaflet est une bibliothèque légère et réactive qui facilite l'intégration de cartes interactives dans les applications web. Elle prend en charge la superposition de marqueurs, de polygones, de lignes et de tuiles cartographiques.

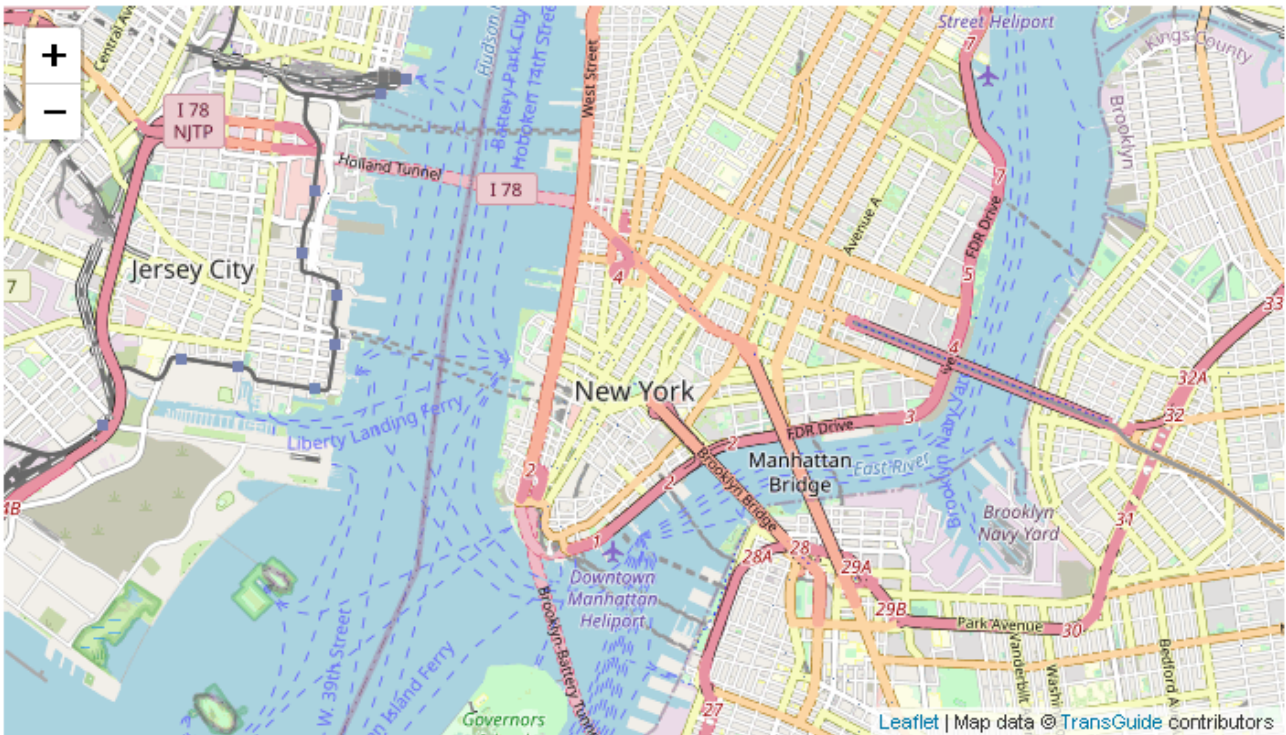


FIGURE 5.6 – Carte de l'application

5.3.2 Simulation des taxis

Nous ajoutons des marqueurs de taxis à la carte, représentés par l'icône d'un taxi. Les emplacements des taxis sont définis aléatoirement dans la cité de New York.

Nous utilisons une boucle pour créer un marqueur de taxi pour chaque emplacement et l'ajoute à la carte. Il lie également une fenêtre contextuelle (popup) au marqueur, affichant des informations telles que le fournisseur et les coordonnées géographiques.

Le bouton Départ Ce bouton permet à l'utilisateur de saisir sa position. Lorsque l'utilisateur clique sur ce bouton et clique sur la carte, ça va épingler la position de l'utilisateur sur la carte



FIGURE 5.7 – La simulation de taxis

Le bouton Arrivée Ce bouton permet à l'utilisateur de saisir sa destination. Lorsque l'utilisateur clique sur ce bouton et clique sur la carte, ça va épingler la position de destination sur la carte

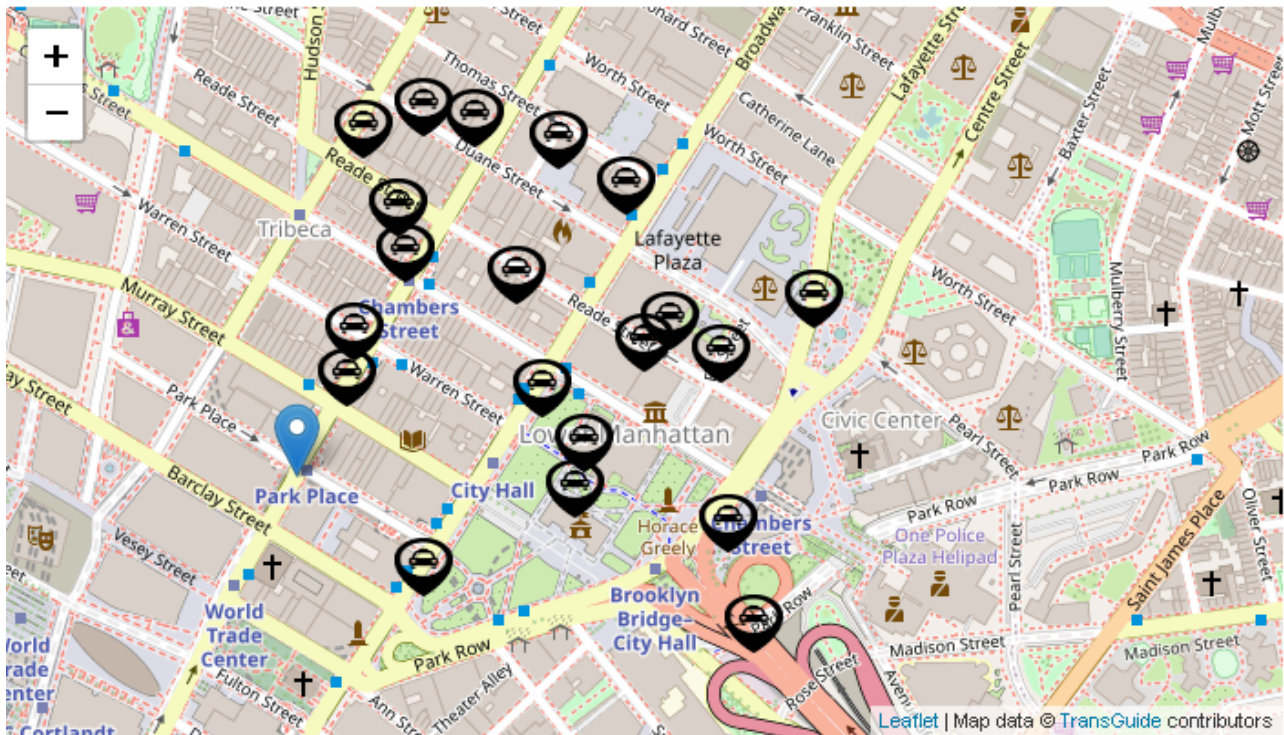


FIGURE 5.8 – Epingler la position du départ de l'utilisateur

Le bouton Recommander Ce bouton permet à recommander à l'utilisateur trois taxi les plus rapide et les plus proche a l'utilisateur, le premier taxi recommandé est coloré en rouge et les deux autre taxi sont colorés en jaune

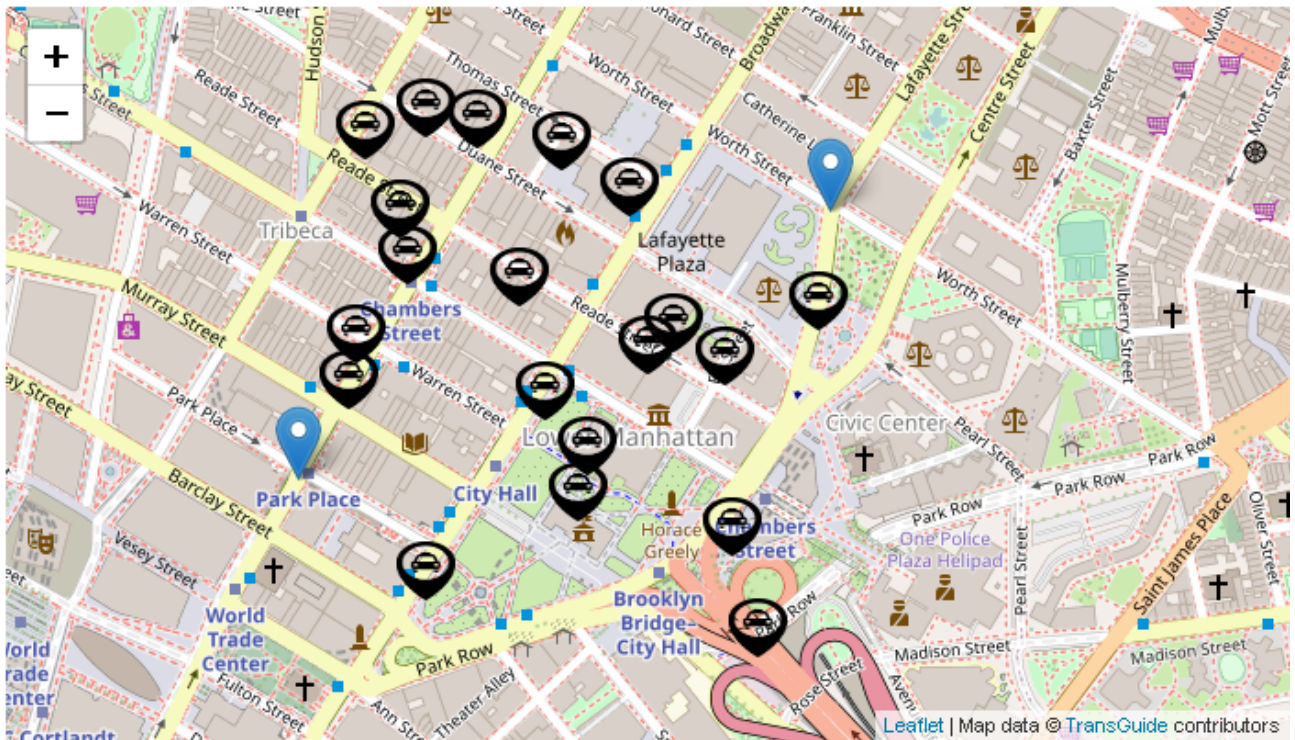


FIGURE 5.9 – Epingleur la position de destination de l'utilisateur

Temps du trajet Prédit Quand l'utilisateur fixe son trajet et clique sur explore ça va afficher la prédiction du temps de son trajet (Figure 5.11)

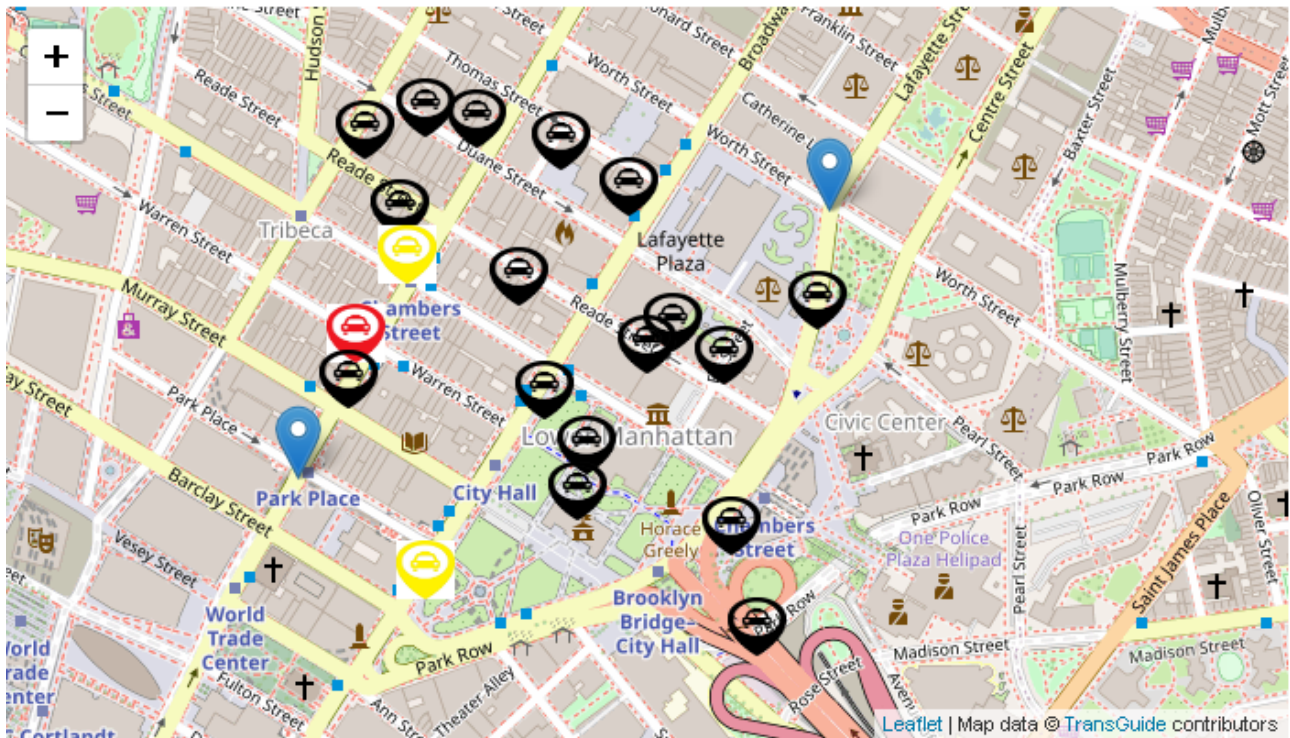


FIGURE 5.10 – la recommandation des taxis

5.4 Evaluation

Nous avons évalué notre système de recommandation de transport basé sur le modèle de régression linéaire en le comparant avec plusieurs modèles d'apprentissage supervisés en se basant sur la mesure de performance de la racine de l'erreur quadratique moyenne (RMSE). Les modèles entraînés et leurs scores RMSE correspondants sont récapitulés dans le tableau suivant :

Modèle utilisé	RMSE
GradientBoostingRegressor	3254.096
SVR	3174.076
LinearRegression	3166.221
RandomForestRegressor	4024.467

TABLE 5.1 – Performance des modèles de régression

Nous avons entraîné plusieurs modèles populaires de régression, tels que le GradientBoostingRegressor, SVR, LinearRegression et RandomForestRegressor. Chaque modèle a été entraîné sur notre jeu de données historiques de trajets pour prédire la durée des trajets en fonction de la distance et de l'ID du fournisseur de transport. Les scores RMSE obtenus fournissent une indication de la précision de chaque modèle dans la prédiction de la durée des trajets.

D'après les résultats présentés dans le tableau 5.1, nous pouvons observer que le modèle LinearRegression a obtenu le score RMSE le plus bas, avec une valeur de 3166.221. Cela indique que ce modèle a la meilleure

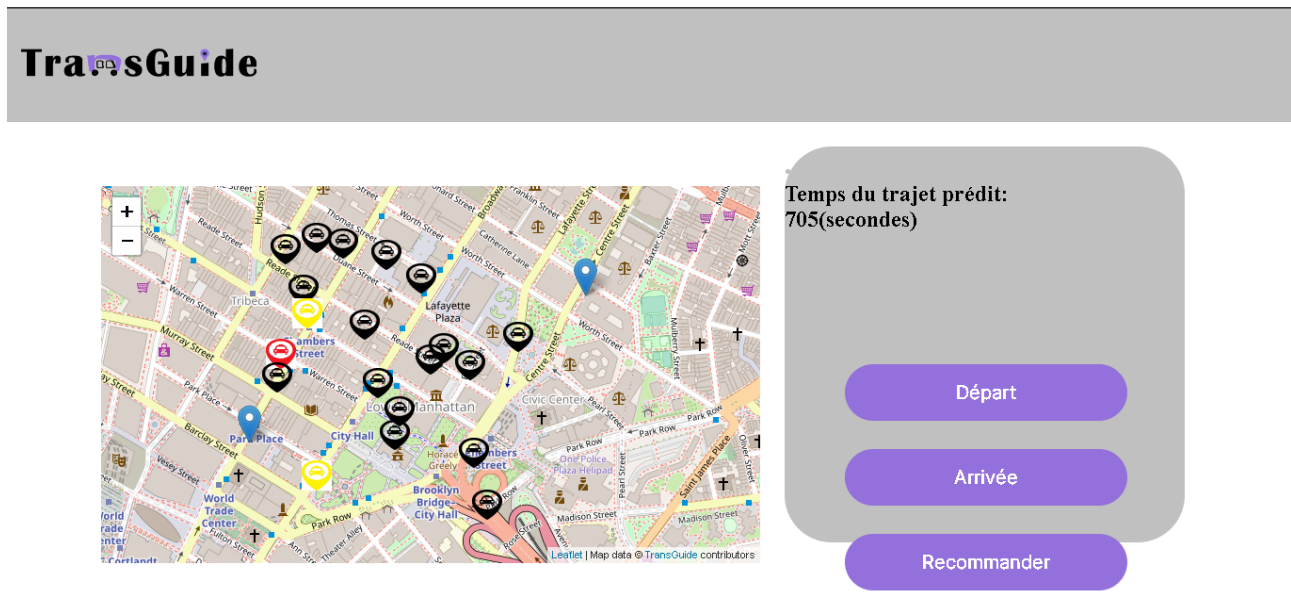


FIGURE 5.11 – Temps du trajet prédit

performance parmi les modèles entraînés pour prédire la durée des trajets. Les autres modèles ont également montré des performances compétitives, mais légèrement supérieures au modèle de régression linéaire.

Il convient de noter que le choix du modèle peut dépendre des exigences spécifiques de l'application et des contraintes du système de recommandation de transport. Dans notre cas, nous avons sélectionné le modèle LinearRegression comme modèle principal pour notre système de recommandation en raison de sa performance globalement meilleure. Cependant, d'autres facteurs tels que la scalabilité, la complexité et le temps de calcul peuvent également être pris en compte lors du choix du modèle approprié pour une application réelle.

L'utilisation de l'apprentissage supervisé et de différents modèles de régression nous a permis de prédire avec précision la durée des trajets dans notre système de recommandation de transport. Le modèle LinearRegression s'est avéré être le plus performant parmi les modèles entraînés, ce qui constitue une base solide pour recommander les trajets les plus rapides et les plus pratiques aux utilisateurs dans les villes intelligentes.

Conclusion

Dans notre approche, nous avons développé une méthode hybride qui combine la régression linéaire et la distance Haversine pour recommander le fournisseur de transport le plus rapide et le taxi le plus proche à l'utilisateur. La régression linéaire est utilisée pour prédire la durée des trajets, tandis que la distance Haversine est utilisée pour calculer la distance géographique entre les points de départ et d'arrivée.

En utilisant la régression linéaire, nous sommes en mesure de prédire avec précision la durée des trajets en fonction de caractéristiques telles que la distance et l'ID du fournisseur de transport. Cela nous permet de comparer les temps de trajet prédits pour les différents fournisseurs de transport et de sélectionner celui qui offre le trajet le plus rapide.

D'autre part, en utilisant la distance Haversine, nous pouvons calculer la distance géographique entre l'utilisateur et les fournisseurs de transport disponibles. Cela nous permet de déterminer le taxi le plus proche de l'utilisateur en prenant en compte la distance réelle sur la carte.

En combinant ces deux approches, nous sommes en mesure de recommander le fournisseur de transport le plus rapide et le taxi le plus proche à l'utilisateur, offrant ainsi une solution complète et précise pour les déplacements dans les villes intelligentes.

En conclusion, dans ce chapitre nous avons évoqué les principales étapes de notre implémentation. En premier lieu, nous avons défini les jeux de données en général, et présenté les datasets que nous avons utilisés pour former notre système, illustrés des captures d'écran de quelques affichages obtenus. Ensuite, nous avons donné l'environnement et les outils utilisés pour construire un système solide permettant de recommander des taxi aux utilisateurs et prédire la durée de trajet. Enfin, nous avons expliqué l'utilité et l'objectif de l'apprentissage supervisé l'évaluation des performances de notre système.

Ce chapitre a permis aux lecteurs de comprendre les composants essentiels de l'environnement de développement utilisé dans le cadre de la compétition et de l'application développée. La connaissance de ces composants est cruciale pour mettre en place un environnement de travail efficace et pour faciliter la compréhension des étapes de développement et des résultats obtenus.

Conclusion Générale

Dans ce mémoire, nous avons abordé une problématique majeure dans le domaine des transports urbains intelligents, à savoir la recommandation des services de taxi. En complément des modes de transport traditionnels, les VTC jouent un rôle crucial dans les déplacements au sein des grandes agglomérations, en particulier pour accéder aux zones mal desservies ou pour se déplacer tard le soir. Cela a conduit à l'émergence de plusieurs entreprises dans le domaine des VTC, créant ainsi la nécessité d'un système de recommandation de transport parmi plusieurs fournisseurs. Pour résoudre ce problème, nous avons proposé dans ce mémoire un système permettant de choisir un taxi (véhicule VTC) parmi différents prestataires. La recommandation est basée sur l'historique des courses effectuées par les taxis, ainsi que sur le calcul de la distance de Haversine afin de recommander le taxi le plus proche.

Pour adresser ses problèmes, nous présentons dans le deuxième chapitre, les principes généraux des systèmes de recommandation, leurs types et leur fonctionnement, suivi d'une introduction aux transports dans les villes intelligentes, et nous avons terminé le chapitre par une présentation des modèles d'apprentissage automatique.

Dans le troisième chapitre, nous avons passé en revue les travaux de recherche antérieurs proposant des systèmes de recommandation de transport. Nous avons identifié des travaux récents et pertinents, et nous les avons comparés en fonction de plusieurs critères tels que le type d'approche (supervisée ou non supervisée), les données utilisées pour la recommandation, les modèles utilisés et les paramètres de recommandation. Ensuite, nous avons analysé ces travaux pour identifier leurs problèmes et proposer notre approche.

Le quatrième chapitre présentait notre approche. Nous avons décrit l'ensemble de données utilisé, les méthodes que nous avons suivies pour l'analyse et l'exploration des données. Nous avons présenté l'algorithme de recommandation que nous avons proposé et détaillé les étapes d'entraînement et de test. Nous avons évalué les performances du système à l'aide de la métrique RMSE (racine de l'erreur quadratique moyenne).

Dans le cinquième chapitre, nous avons présenté les détails de l'implémentation de notre application qui utilise le système de recommandation. En effet, nous avons conçu une application mobile qui utilise le système de recommandation pour afficher les taxis à proximité de l'utilisateur et recommander un taxi en fonction de sa position et de sa destination. Nous avons également présenté les détails d'implémentation.

Pour conclure, nous avons proposé une nouvelle approche pour la recommandation des services de taxi dans les villes intelligentes. Cependant, il reste encore plusieurs perspectives à explorer pour améliorer les systèmes de recommandation. Nous citons :

L'apprentissage incremental : Actuellement, notre système de recommandation se base sur l'historique des courses. Une perspective intéressante consisterait à intégrer des données en temps réel, et des courses les plus récentes.

Utilisation de d'autres paramètres contextuels : En prenant en compte les caractéristiques de trafic comme le temps d'encombrement et les conditions météorologiques qui peuvent impacter le temps de déplacement.

Bibliographie

- [1] Ayat Abedalla, Ali Fadel, Ibraheem Tuffaha, Hani Al-Omari, Mohammad Omari, Malak Abdullah, and Mahmoud Al-Ayyoub. Mtrecs-dlt : multi-modal transport recommender system using deep learning and tree models. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 274–278. IEEE, 2019.
- [2] Leonidas G Anthopoulos. Understanding the smart city domain : A literature review. *Transforming city governments for successful smart cities*, pages 9–21, 2015.
- [3] Ricardo A Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [4] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [5] AA Borodinov and VV Myasnikov. Analysis of the preferences of public transport passengers in the task of building a personalized recommender system. In *CEUR Workshop Proc*, volume 2391, pages 198–205, 2019.
- [6] Robin Burke. Hybrid recommender systems : Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4) :331–370, November 2002.
- [7] Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. Smart cities in europe. *Journal of urban technology*, 18(2) :65–82, 2011.
- [8] Nitin R Chopde and Mangesh Nichat. Landmark based shortest path detection by using a* and haversine formula. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2) :298–302, 2013.
- [9] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification. john wiley & sons. *Inc.*, New York, 5, 2000.
- [10] Mohamed Yacine Gheraibia and Charles Gouin-Vallerand. Intelligent mobile-based recommender system framework for smart freight transport. In *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, pages 219–222, 2019.
- [11] Rudolf Giffinger, Christian Fertner, Hans Kramar, Robert Kalasek, Natasa Pichler-Milanovic, and Evert J Meijers. Smart cities. ranking of european medium-sized cities. final report. 2007.
- [12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [13] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning : data mining, inference, and prediction*, volume 2. Springer, 2009.
- [14] Robert G Hollands. Will the real smart city please stand up ? : Intelligent, progressive or entrepreneurial ? In *The Routledge companion to smart cities*, pages 179–199. Routledge, 2020.

- [15] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems : An Introduction*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.
- [16] Julian Laufs, Hervé Borrión, and Ben Bradford. Security and the smart city : A systematic review. *Sustainable cities and society*, 55 :102023, 2020.
- [17] Tony S Lee. *Intelligent Transportation Systems : New Principles and Architectures*. Taylor & Francis, 2000.
- [18] Yang Liu, Cheng Lyu, Zhiyuan Liu, and Jinde Cao. Exploring a large-scale multi-modal transportation recommendation system. *Transportation Research Part C : Emerging Technologies*, 126 :103070, 2021.
- [19] Zhihan Lv, Liang Qiao, Ken Cai, and Qingjun Wang. Big data analysis technology for electric vehicle networks in smart cities. *IEEE Transactions on Intelligent Transportation Systems*, 22(3) :1807–1816, 2020.
- [20] Hamid Menouar, Ismail Guvenc, Kemal Akkaya, A Selcuk Uluagac, Abdullah Kadri, and Adem Tuncer. Uav-enabled intelligent transportation systems for the smart city : Applications and challenges. *IEEE Communications Magazine*, 55(3) :22–28, 2017.
- [21] Tom M Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11) :30–36, 1999.
- [22] Kevin P Murphy. *Machine learning : a probabilistic perspective*. MIT press, 2012.
- [23] Michael J Pazzani and Daniel Billsus. The adaptive web. In *Content-based Recommendation Systems*, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.
- [24] Asier Perallos, Unai Hernandez-Jayo, Enrique Onieva, and Ignacio Julio García Zuazola. *Intelligent transport systems : technologies and applications*. John Wiley & Sons, 2015.
- [25] G Salton, A Wong, and CS Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11) :613–620, November 1975.
- [26] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. The adaptive web. In *Collaborative Filtering Recommender Systems*, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.
- [27] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning : From theory to algorithms*. Cambridge university press, 2014.
- [28] Si Ying Tan and Araz Taeihagh. Smart city governance in developing countries : A systematic literature review. *sustainability*, 12(3) :899, 2020.
- [29] Angeliki Maria Toli and Niamh Murtagh. The concept of sustainability in smart city definitions. *Frontiers in Built Environment*, 6 :77, 2020.
- [30] Fanyou Wu, Cheng Lyu, and Yang Liu. A personalized recommendation system for multi-modal transportation systems. *Multimodal transportation*, 1(2) :100016, 2022.

Résumé

Le mémoire présente une étude sur les systèmes de recommandation, qui sont largement utilisés pour fournir des suggestions adaptées aux préférences des utilisateurs dans divers domaines. Ces systèmes estiment l'intérêt d'un utilisateur pour une ressource spécifique en se basant sur ses informations personnelles et les choix des utilisateurs similaires.

Dans cette recherche, une approche hybride a été développée en combinant la régression linéaire et la distance Haversine pour recommander le fournisseur de transport le plus rapide et le taxi le plus proche à l'utilisateur. La régression linéaire est utilisée pour prédire la durée des trajets, tandis que la distance Haversine permet de calculer la distance géographique entre l'utilisateur et les taxis.

Cette approche hybride offre une solution efficace pour recommander des options de transport personnalisées aux utilisateurs, en prenant en compte à la fois la durée du trajet et la proximité géographique. Elle permet ainsi d'améliorer l'expérience utilisateur en proposant des choix pertinents et adaptés à leurs besoins.

Ce mémoire contribue à l'avancement des systèmes de recommandation en proposant une méthode hybride spécifique au domaine du transport, qui peut être étendue à d'autres domaines nécessitant des recommandations basées sur des critères géographiques et temporels. Les résultats obtenus démontrent l'efficacité de cette approche et ouvrent des perspectives intéressantes pour l'amélioration continue des systèmes de recommandation.

Mots clés : systèmes de recommandation, approche hybride, régression linéaire, distance Haversine, transport, expérience utilisateur, critères géographiques, critères temporels.

Abstract

The dissertation presents a study on recommendation systems, which are widely used to provide personalized suggestions to users in various domains. These systems estimate a user's interest in a specific resource based on their personal information and the choices of similar users.

In this research, a hybrid approach was developed by combining linear regression and the Haversine distance to recommend the fastest transportation provider and the nearest taxi to the user. Linear regression is used to predict travel durations, while the Haversine distance is used to calculate the geographical distance between the user and the taxis.

This hybrid approach offers an efficient solution for recommending personalized transportation options to users, considering both travel duration and geographical proximity. It improves the user experience by suggesting relevant and tailored choices to their needs.

This dissertation contributes to the advancement of recommendation systems by proposing a domain-specific hybrid method for transportation, which can be extended to other domains requiring recommendations based on geographical and temporal criteria. The results obtained demonstrate the effectiveness of this approach and open interesting perspectives for the continuous improvement of recommendation systems.

Keywords : recommendation systems, hybrid approach, linear regression, Haversine distance, transportation, user experience, geographical criteria, temporal criteria.