Université Claude Bernard Lyon 1

Rapport de stage

Intégration de données géographiques pour la recommandation de quartiers

Du 2 mai au 27 juillet 2018

LIRIS, Laboratoire d'Informatique en Image et Systèmes d'information
8 Boulevard Niels Bohr
69622 Villeurbanne

Étudiante	Nelly Barret (Licence 3 Informatique, UCBL1)
Tuteurs entreprise	Fabien Duchateau (Maître de conférences, LIRIS)
	Franck Favetta (Maître de conférences, LIRIS)
Tuteur universitaire	Fabien De Marchi (Maître de conférences, LIRIS)
Rapporteur	Matthieu Heitz (Doctorant, LIRIS)

Remerciements

Tout d'abord, je souhaite remercier Fabien Duchateau et Franck Favetta, mes maîtres de stage, sans qui ce stage n'aurait pas été possible. Ce sont eux qui m'ont suivi, avec patience et pédagogie, tout au long de ces trois mois.

J'adresse ensuite mes remerciements à la start-up HiL (Home in Love) avec qui le LIRIS collabore sur le projet Home in Love. Je remercie aussi les membres du projet, Maryvonne Miquel, Loïc Bonneval et Aurélien Gentil, avec qui j'ai pu échanger pendant les réunions.

Je souhaite aussi remercier le LabEx IMU (Laboratoire d'Excellence, Intelligence des Mondes Urbains), qui a financé ce stage à titre exceptionnel puisque c'est un stage de Licence.

Je souhaite aussi remercier tout le personnel du LIRIS pour leur accueil.

Enfin, je souhaite remercier l'Université Claude Bernard Lyon 1 pour mes trois années de Licence pendant lesquelles j'ai beaucoup appris et, en plus, découvert la recherche.

Table des matières

1 Introduction

Étant en troisième année de Licence informatique à l'Université Claude Bernard Lyon 1, je dois effectuer un stage validant mon premier cycle d'études supérieures. Ce stage peut se faire en entreprise ou en laboratoire pour une durée de 12 semaines (de mai à juillet). Il a pour but de nous faire acquérir une expérience professionnelle enrichissante et de nous faire prendre conscience des contraintes du monde professionnel. Envisageant un master recherche à Lyon 1, faire un stage au LIRIS est pour moi une opportunité rêvée. Ma principale attente vis-à-vis de ce stage est de découvrir le monde de la recherche afin de consolider mon projet professionnel proche et futur. Je souhaite aussi découvrir les contraintes du monde professionnel à travers ce stage, comme les projets pluridisciplinaires et l'intégration sur un projet en cours. C'est ainsi que j'ai pris contact avec mes professeurs pour leur faire part de mon souhait. Ce sont Fabien Duchateau et Franck Favetta, chercheurs au LIRIS, qui m'ont alors proposé un stage orienté recherche dans leur équipe et plus particulièrement sur un de leurs projets.

Le LIRIS, Laboratoire d'InfoRmatique en Imagerie et Systèmes d'information, est une unité mixte de recherche (UMR 5205). Il dépend du CNRS, de l'INSA Lyon, de l'Université Claude Bernard Lyon 1, de l'Université Lumière Lyon 2 et de l'École Centrale de Lyon. Il s'intéresse à l'informatique et plus généralement aux sciences et technologies de l'information. Le LIRIS est composé de 14 équipes réparties en 6 pôles de recherche. Le pôle Science des données est composé de 3 équipes : BD (Bases de Données), D2ML (Data Mining et Machine Learning) et GOAL (Graphes, algOrithmes et AppLications). Les principaux objectifs de l'équipe Bases de données sont la conception de nouveaux modèles face à la génération massive de données hétérogènes. Elle s'intéresse aussi à développer des outils pour maîtriser ces données (accès, diffusion, usage...). Les projets de recherche du LIRIS sont financés par différentes sources dont le LabEx IMU (Laboratoire d'Excellence, Intelligence des Mondes Urbains) qui finance des projets pluridisciplinaires. Le LabEx IMU fédère un ensemble de travaux et de projets de recherche qui traitent de la ville et des phénomènes urbains dans leurs relations aux processus de mondialisation. Il finance le projet pluridisciplinaire Home in Love dans lequel s'inscrit mon stage.

Le LabEx IMU finance le projet HiL qui a pour intitulé : « Système de recommandation avec visualisation spatiale et non spatiale pour la recherche immobilière ». Ce projet met en collaboration des chercheurs en informatique du LIRIS, des chercheurs en sociologie du Centre Max Weber et la start-up Home in Love. Il a pour but de proposer une plateforme qui aide à la recherche de logements en améliorant la recommandation actuelle et en intégrant des critères sociologiques lors de la recommandation.

Mon stage, financé par IMU, porte principalement sur la recommandation de quartiers. Il se compose de deux parties : l'analyse des quartiers et la recommandation de quartiers. Pour recommander des quartiers pertinents, il faut avoir des informations sur ces quartiers. Ces informations proviennent de différentes sources, c'est pourquoi une étape d'intégration de données est nécessaire. Mon stage étant à connotation recherche, il y a eu une étape de recherche bibliographique. À partir de la lecture de plusieurs articles sur l'intégration de données et la recommandation, un état de l'art a pu être établi. Plusieurs contributions ont été identifiées : l'implémentation d'un module d'intégration et l'implémentation de différentes stratégies de recommandation. Ces stratégies ont été

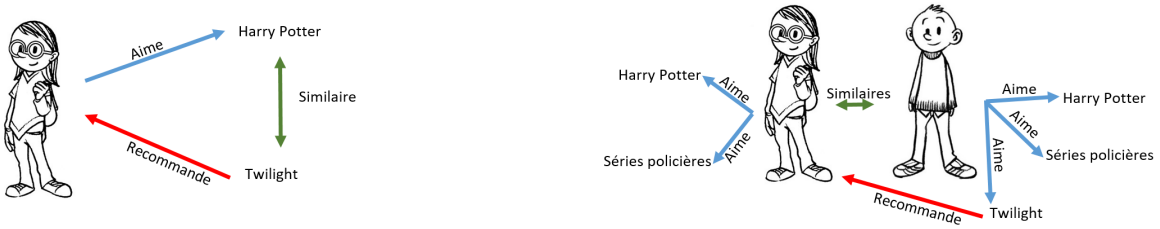
testées pour évaluer leur pertinence sur des données réelles fournies par la start-up. Dans ce rapport, nous résumons ce travail de recherche et développement.

2 État de l'art

L'état de l'art porte sur deux domaines : l'intégration de données et la recommandation. L'intégration de données est un processus de regroupement de données provenant de différentes sources en une source d'information commune et pertinente. Cela permet de créer une vue logique homogène même si les données physiques sont hétérogènes. Plusieurs étapes se distinguent dans l'intégration de données : l'identification et la compréhension des sources, la détection des correspondances et des mappings et éventuellement la fusion des données. C'est un domaine qui est fortement étudié depuis des décennies et de nombreux travaux ont été proposés dans le but d'automatiser cette tâche [2]. L'automatisation n'est pas nécessaire dans le cadre du projet puisque ce n'est pas une tâche récurrente. L'intégration des données a donc été manuelle. L'intégration de différentes sources de données apporte *de facto* des conflits d'identité et de noms d'attributs. Les conflits d'identité apparaissent lorsque le même concept est identifié par des attributs différents ou lorsque la même clé dans deux sources différentes n'a pas la même signification. Deux types de conflits de noms d'attributs peuvent survenir : l'homonymie (le même nom est utilisé pour désigner des concepts différents) et la synonymie (le même concept est décrit par plusieurs noms différents). Dans la littérature, des solutions sont proposées pour les résoudre. Un exemple de solution pour les conflits d'identité est la définition d'une fonction de conversion de clé et, pour les conflits de noms d'attributs, le renommage.

La recommandation est un domaine très étudié depuis les années 1990. Les systèmes de recommandation sont des systèmes capables de proposer des recommandations personnalisées ou de guider un utilisateur vers des ressources intéressantes. Face au volume de données toujours plus grand, il est nécessaire de filtrer et de hiérarchiser les informations. Par exemple, en magasin, un utilisateur aura le choix parmi des centaines de DVD tandis que sur Internet ce même utilisateur aura le choix parmi des milliers de DVD. Les outils de recommandation n'ont cessé de s'améliorer et d'enrichir l'expérience utilisateur. Plusieurs stratégies peuvent être adoptées face à un système de recommandation. Ils se divisent principalement en trois catégories [3] :

- Basiques (utilisant la similarité ou la popularité, e.g. dans le cadre de la recommandation de films, la saga Harry Potter et la trilogie Le Seigneur Des Anneaux sont similaires).
- Basés sur le contenu (*content-based filtering*) [4]. Ce type de système de recommandation va proposer des recommandations similaires aux goûts d'un utilisateur. La Figure (1a) illustre un exemple simple de filtrage sur le contenu. L'utilisatrice aime la saga Harry Potter et le système sait que la saga Twilight est une saga similaire à Harry Potter. Le système va donc lui recommander Twilight.
- Basés sur le collaboratif (*collaborative filtering*) [5]. Ce type de système de recommandation va détecter des utilisateurs qui partagent les mêmes goûts pour leur proposer des recommandations similaires. La Figure (1b) illustre un exemple simple de filtrage collaboratif. Une utilisatrice aime la saga Harry Potter ainsi que les séries policières. Un utilisateur aime la saga Harry Potter, les séries policières et la saga Twilight. Ces deux utilisateurs ayant des préférences similaires, le système va recommander la saga Twilight à l'utilisatrice.



(a) Illustration du filtrage sur le contenu

(b) Illustration du filtrage collaboratif

FIGURE 1 – Illustration du filtrage sur le contenu et du filtrage collaboratif

Quatre algorithmes de recommandation ont été étudiés parmi les plus populaires : SVD, NMF, KNN et decision tree. Chacun de ces algorithmes appartient à un type de système de recommandation.

L'algorithme SVD (*Singular-Value Decomposition*) [1], algorithme de type collaborative filtering, a été popularisé grâce à Netflix, entreprise américaine qui propose un service de location de DVD en ligne. Chaque client peut noter (entre 1 et 5) les films qu'il a vu. Grâce à ces retours utilisateurs, Netflix propose à ses clients des films qu'ils seraient susceptibles d'aimer. Ces propositions étaient générées par leur système de recommandation CinéMatch. Netflix juge alors qu'un système de recommandation plus évolué leur permettrait de fidéliser leurs clients et d'augmenter leurs bénéfices. Netflix décide de lancer, en 2009, un concours, « Le Netflix Prize ». Le but de ce concours était de construire un algorithme de recommandation qui pourrait surpasser les tests de 10%, i.e. de réduire de 10% la RMSE (Root-Mean-Square Error) de CinéMatch. La RMSE est une mesure qui permet de calculer la différence entre les valeurs prédites par un modèle et les valeurs réellement observées. L'équipe de recherche gagnante a construit un système de recommandation à partir d'une centaine de modèles, dont la SVD. Son principe est de décomposer une matrice en trois sous-matrices afin de faire apparaître des « profils ». Dans le cas de Netflix par exemple, en décomposant leur matrice de notes, des profils de films et d'utilisateurs vont apparaître. Comme 99% des notes ne sont pas connues, la matrice n'est pas dense et la difficulté sera de prédire les notes manquantes. Ainsi le système pourra recommander des films susceptibles d'intéresser les utilisateurs grâce aux notes qu'il aura prédites.

L'intuition de l'algorithme NMF (*Non negative Matrix Factorization*) [6], algorithme de type collaborative filtering, est de décomposer une matrice non-négative en deux sous-matrices non-négatives, toujours dans le but de faire apparaître des profils. Comme pour la SVD, le système pourra prédire les notes manquantes et ainsi recommander des films pertinents à un utilisateur. La différence entre les algorithmes SVD et NMF est que la SVD propose une décomposition unique tandis que la NMF en propose plusieurs. En effet, la SVD permet d'obtenir des profils de films et d'utilisateurs ainsi qu'une matrice « de pondération ». C'est cette matrice qui rend la SVD unique. La SVD est donc plus robuste puisqu'elle propose toujours les mêmes recommandations pour une matrice donnée mais la NMF favorise la sérendipité puisqu'elle propose plusieurs décompositions.

L'algorithme KNN (*K-Nearest Neighbors*), algorithme de type content-based et collaborative filtering, permet de classifier un objet en fonction de son voisinage. Les données d'entrée sont un ensemble d'objets et un ensemble de classes. Chaque objet appartient à une classe et peut être représenté en tant que point dans un graphe. L'objectif est de classifier les nouveaux objets, qui prendront comme identité la classe majoritaire de leurs k voisins les plus proches. L'intuition des arbres de décision est de diviser les critères en décisions afin de recommander un produit qui s'approche au plus près des critères souhaités. Les arbres de décision peuvent aussi servir à justifier une recommandation.

Dans notre contexte, il n'existe qu'un seul article traitant de la recommandation immobilière et celui-ci a été écrit en Corée du Sud. Cet article [7] est peu détaillé au niveau de la partie technique de la recommandation basée sur le CBR (Case-Based Reasoning) et algorithme ou librairie n'ont pas été mentionnés. Il est plutôt focalisé sur l'étude de l'ergonomie de l'interface afin de faciliter la recherche de logement. Il faut donc adapter un des algorithmes existants à notre contexte immobilier. Pour baser notre approche sur le collaboratif, dans le cadre de recommandations au niveau de la France, il faudrait avoir plusieurs avis sur les quartiers et les communes pour un même groupe d'utilisateurs. Or ce type de données n'est pas disponible. Ainsi, la comparaison des quatre algorithmes détaillés ci-dessus a permis de baser notre approche sur le contenu et non sur le collaboratif.

3 Vue d'ensemble des réalisations

Cette section présente une vue d'ensemble de l'approche utilisée où nous cherchons à recommander des quartiers à partir de leurs caractéristiques. Les principaux verrous scientifiques de ce projet sont la performance et la qualité. D'une part, il est nécessaire de choisir un algorithme de recommandation efficient tout en limitant la recherche d'IRIS candidats à un sous-ensemble idéalement sélectionné (e.g. selon une distance). D'autre part, la recommandation doit offrir une qualité acceptable, ce qui nécessite de détecter et de regrouper en amont les indicateurs les plus pertinents. Les données sur les quartiers sont spatialement découpées en IRIS. Un IRIS, anagramme de Ilots Regroupés pour l'Information Statistique, est un découpage du territoire en mailles de taille homogène. L'IRIS est l'unité de référence pour la diffusion de données infra-communales. Il existe trois types d'IRIS : habitat, activité et divers. Par exemple, les IRIS d'habitat ont une population entre 1 800 et 5 000 habitants. La Figure 2 illustre un schéma de système de recommandation adapté pour la recommandation d'IRIS.

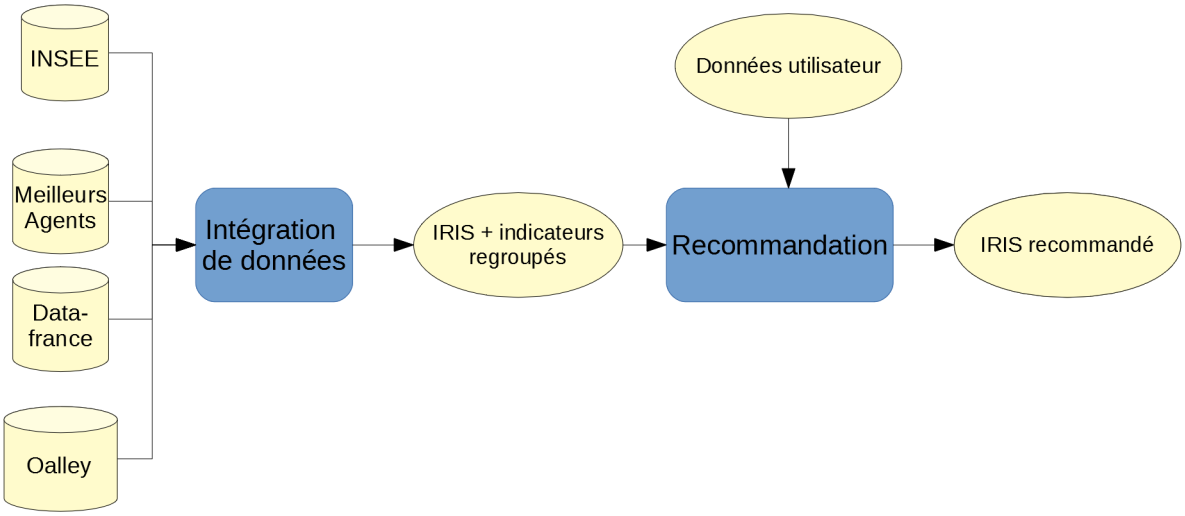


FIGURE 2 – Schéma d'un système de recommandation adapté pour la recommandation d'IRIS

Les sources de données intégrées dans le système de recommandation sont l'INSEE (Institut National de la Statistique et des Études Économiques) et le site des Meilleurs Agents. Le site meilleursagents.com fournit les prix moyens au mètre carré (pour les maisons et les appartements) par commune. L'INSEE fournit des indicateurs bruts pour chaque IRIS. Un indicateur brut est une valeur représentant le nombre de services, par exemple le nombre de boulangeries ou le nombre de terrains de foot. Comme les sources sont multiples et hétérogènes, un module d'intégration de données est nécessaire, comme le montre le premier rectangle de la Figure 2. L'INSEE fournit beaucoup d'indicateurs bruts, environ 800. Avoir autant d'indicateurs ne permet pas de caractériser les IRIS à un bon niveau de détail. Par exemple, connaître le nombre de restaurants ou le nombre de cinémas pour un IRIS n'est pas forcément pertinent, en revanche savoir si un IRIS est animé semble beaucoup plus pertinent dans le cadre d'un système de recommandation. Afin d'avoir des critères pertinents, un regroupement des indicateurs bruts a été effectué avec l'aide des sociologues.

Après intégration de nos sources et regroupement de nos indicateurs, la recommandation peut être commencée comme l'indique le deuxième rectangle de la Figure 2. C'est le cœur du système et il va trouver les éléments les plus pertinents à recommander. Ce module va proposer un ensemble d'IRIS similaires à l'IRIS sélectionné dans le prototype. Dans un premier temps, il a été décidé d'utiliser un système de recommandation qui se base sur le contenu. En effet, nous avons peu de profils utilisateur (qui sont des clients suivis par Home in Love). Il y a un travail de saisie et de transcription de données à effectuer sur ces profils et ce travail est en cours par les sociologues.

Par la suite, les modules d'intégration de données et de recommandation seront détaillés. Nous terminerons par le prototype et les résultats préliminaires obtenus avant de conclure.

4 Module d'intégration de données

L'étape préliminaire à l'intégration de données a été la recherche de sources d'informations. Celles qui ont été retenues sont l'INSEE et le site des Meilleurs Agents. Cette partie de l'étape d'intégration de données a permis de récupérer les indicateurs bruts caractérisant les IRIS et les prix immobiliers par commune. Ainsi apparaît le problème de l'hétérogénéité puisque les formats des données diffèrent d'une source à l'autre. Par exemple, l'INSEE fournit des fichiers Excel contenant des indicateurs bruts pour chaque IRIS tandis que le site des Meilleurs Agents fournit une page Web contenant les prix immobiliers par commune. Afin de regrouper toutes ces informations sous un même modèle, il est nécessaire de parcourir et de transformer ces données. Chaque source nécessite donc un parser spécifique. Pour les prix immobiliers, le logiciel d'extraction de données Octoparse a été utilisé. Ce logiciel permet de récupérer des données issues de sites Web en automatique. Après la sélection des champs à récupérer, Octoparse lance les requêtes sur la liste d'URL donnée en paramètre. L'intégration des fichiers Excel a été faite en Python via la librairie XLRD. Cette librairie propose des méthodes telles que la lecture et le formatage de données stockées dans des fichiers Excel.

L'intégration des données engendre des conflits d'identité et de noms d'attributs ainsi qu'un problème de granularité. Les IRIS sont identifiés par un code composé du code INSEE de la commune (5 chiffres) et du code de l'IRIS (4 chiffres). Ce code à 9 chiffres permet de les identifier avec unicité. Des conflits d'identité surviennent lorsque certaines communes ne sont pas découpées en IRIS. Dans ce cas, la commune qui n'est composée que d'un seul IRIS n'a pas de code spécifique. Il est alors composé de celui de la commune et des quatre lettres ZZZZ. Ces lettres commencent à être utilisées en 2016, alors qu'en 2014, quatre zéros 0000 étaient utilisés. Des conflits de noms d'attributs apparaissent lorsque des codes représentant les mêmes données portent des noms différents, e.g. le nom d'un IRIS peut être représenté par « LIBIRIS » ou « LIB_IRIS ». Le problème de granularité provient du fait que les données peuvent être au niveau d'un IRIS, de la commune ou du département. Ces problèmes ont été résolus en codant les mappings manuellement.

Une fois les données intégrées, pour l'étape suivante, le regroupement des indicateurs a été discuté avec les sociologues. Le site des Meilleurs Agents fournit les prix moyens au mètre carré (pour les maisons et les appartements) par commune. Dans un deuxième temps, ces tarifs seront également intégrés. Afin d'avoir un regroupement uniforme pour tous les IRIS, l'arborescence des indicateurs regroupés a été stockée dans un fichier JSON. Grâce à cette arborescence, chaque IRIS a comme caractéristiques ses indicateurs bruts (e.g. le nombre de boulangeries, le nombre de terrains de football) ainsi que les indicateurs regroupés (e.g. l'animation ou les loisirs). Une fonction permet de calculer la valeur de chaque indicateur regroupé et ce pour chaque IRIS. Des critères principaux de niveau 1 sont apparus ainsi que des sous-critères de niveaux 2, 3 et 4. Les 800 indicateurs fournis par l'INSEE ont été regroupés en 12 indicateurs principaux et 38 indicateurs feuilles. La Figure 3 montre le regroupement des indicateurs pour le critère d'animation. Ce critère se divise en plusieurs niveaux intermédiaires. Ce sont les feuilles de cette arborescence qui servent à la recommandation, puisque ce sont elles qui caractérisent les IRIS objectivement.

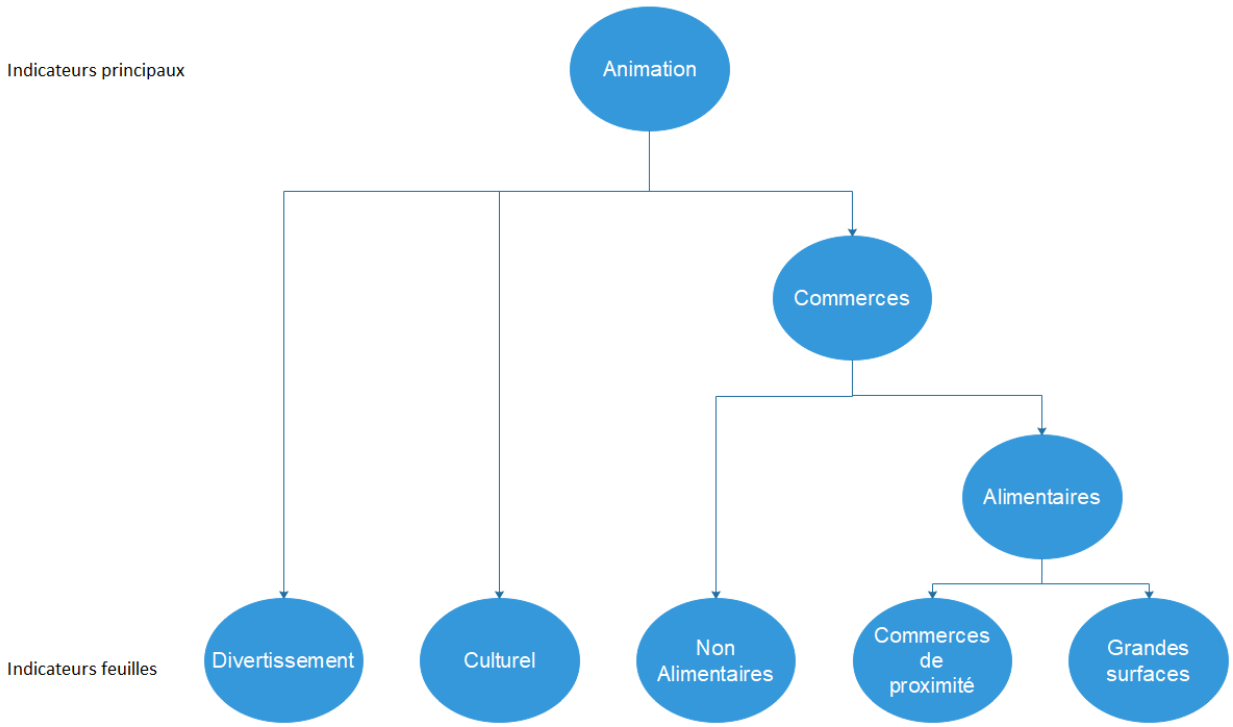


FIGURE 3 – Exemple d'indicateurs regroupés pour le critère animation

Un IRIS étant par définition, une petite maille de territoire, il n'est pas forcément pertinent de l'analyser seul au vu de sa petite taille. En effet, un IRIS peut ne pas avoir de restaurants et de commerces étant donné sa petite taille alors que ses voisins auront plusieurs dizaines de restaurants et de commerces. La finesse de découpage apporte donc un avantage et un inconvénient. L'avantage est que cela permet d'avoir beaucoup de données, ce qui permet de faire des recommandations précises. L'inconvénient est que le découpage n'est pas toujours représentatif pour les indicateurs regroupés. C'est pour cela que prendre en compte le voisinage plus ou moins proche d'un IRIS est une piste pour améliorer la recommandation et pallier à la finesse du découpage. La France compte environ 50000 IRIS, il faut donc calculer le voisinage pour ces 50000 IRIS. Ce n'est pas un calcul qui peut se faire à la volée au vu du temps de calcul que cela nécessite. La solution retenue pour résoudre ce problème a été de créer un fichier JSON qui contient pour chaque IRIS son voisinage direct, i.e. la liste des IRIS adjacents à l'IRIS donné. Ce fichier sera pré-chargé, ce qui permettra d'améliorer la fluidité du prototype.

Plusieurs perspectives sont possibles pour ce module d'intégration. La première est la construction d'un voisinage. Deux types de voisinages sont possibles : les voisinages en niveaux et les voisinages par rapport à un rayon. Ces voisinages permettraient de ne pas considérer seulement un IRIS mais un ensemble d'IRIS (e.g. une boulangerie qui se situe dans un IRIS voisin à 200 mètres peut être considérée comme appartenant aussi à l'IRIS considéré). Pour le voisinage en niveaux, plusieurs niveaux sont possibles : le voisinage direct (niveau 1), proche (niveau 2) et moins proche (niveau 3). La deuxième perspective est l'utilisation d'un système d'indexation. Quand un IRIS est sélectionné dans le prototype, il faut retrouver l'IRIS en question, (i.e. être capable de retrouver dans quelle géométrie se trouve la coordonnée souhaitée par l'utilisateur). Pour le faire il faut utiliser du geoparsing. La solution actuelle est l'indexation par département. Les données

de chaque département sont stockées dans des fichiers GEOJSON (propres à chaque département). Visualiser dans un navigateur un fichier GEOJSON qui contiendrait tous les départements français est impossible en temps raisonnable. En effet, la taille du fichier ne permettrait pas une navigation fluide. Le découpage des données en plusieurs fichiers est une alternative qui permet de réduire leurs tailles et de ne pas surcharger le navigateur. L'utilisation d'un SIG (Système d'Information Géographique) permettrait d'améliorer les performances en profitant de l'implémentation native de fonctions spatiales. Ce genre de système permet de recueillir, stocker et gérer des données spatiales et géographiques.

Ce module d'intégration a permis de regrouper les différentes sources de données en un ensemble cohérent. En effet, les 800 indicateurs bruts de l'INSEE ont été intégrés et regroupés en une trentaine. Ce regroupement a permis de rendre plus objectifs les indicateurs bruts proposés par l'INSEE. À terme, le voisinage qui sera pré-calculé, permettra de relativiser l'influence des indicateurs. Nous pouvons donc maintenant continuer avec le module de recommandation.

5 Module de recommandation

Un IRIS, avec sa trentaine d'indicateurs regroupés, peut être représenté comme un vecteur (voir Figure 11 en Section 8). Plusieurs scénarios sont possibles pour la recommandation de quartiers à partir d'un lieu de départ. En effet, le quartier de départ peut être composé d'un ou plusieurs IRIS. Par exemple, il peut être composé de l'IRIS où la personne vit ou d'une liste d'IRIS composée, soit de son lieu de vie et des lieux où elle a vécu, soit des quartiers qu'elle aime, soit des quartiers idéaux. Le quartier d'arrivée est la plupart du temps défini en fonction du futur lieu de travail, dans le cadre d'une mutation professionnelle. En fonction des scénarios, différents algorithmes existants ont été adaptés pour la recommandation. La bibliothèque scikit-learn a été utilisée pour implémenter ce module. Cette bibliothèque implémente des algorithmes d'apprentissage, qui peuvent être utilisés pour du data mining par exemple. Elle permet, entre autres, de faire de la classification, de la régression et du clustering.

Premièrement, nous proposons l'implémentation d'un algorithme basique qu'est la mesure cosine. La mesure cosine, ou similarité cosine, permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux. Comme la valeur du cosinus est comprise dans l'intervalle [-1,1], la valeur -1 indiquera des vecteurs totalement opposés, 0 des vecteurs indépendants et 1 des vecteurs similaires. Les valeurs intermédiaires permettent d'évaluer le degré de similarité. Cette mesure, appliquée entre un IRIS de départ et une liste d'IRIS candidats à la recommandation, permet d'évaluer le degré de similarité entre l'IRIS de départ et chaque IRIS d'arrivée. L'IRIS de départ est l'ancien domicile de l'utilisateur, la liste d'IRIS candidats est sa zone d'arrivée (e.g. les IRIS situés à une certaine distance de l'IRIS du futur lieu de travail). Pour chaque IRIS candidat, il sera comparé à l'IRIS de départ et ainsi il sera possible de sélectionner les IRIS les plus pertinents en fonction du résultat de la similarité cosine. La Figure 4 illustre un exemple de calcul de similarité cosine. Le vecteur de similarité regroupe les similarités obtenues pour chaque IRIS candidat. L'IRIS 2 ressemble le plus à l'IRIS de départ en terme de critères puisque sa similarité est proche de 1, c'est donc lui qui sera recommandé.

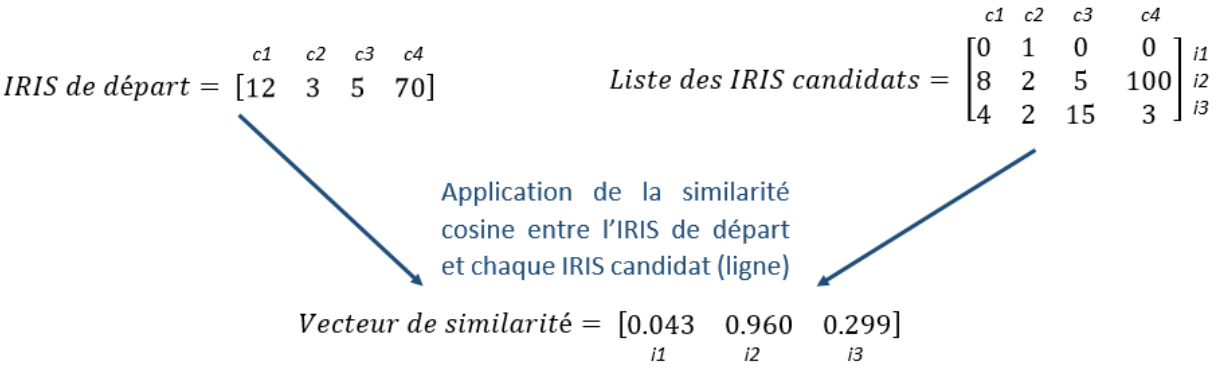


FIGURE 4 – Application de la similarité cosine entre un IRIS de départ et une liste d'IRIS candidats

Deuxièmement, nous avons proposé un algorithme de type content-based qui exploite l'écart-type (standard deviation). L'écart-type sert à mesurer la dispersion (ou l'étale-

ment) d'un ensemble de valeurs autour de leur moyenne. Plus l'écart-type est faible, plus les données sont homogènes. Pour chaque indicateur de l'ensemble des IRIS de départ, on fait l'hypothèse que l'inverse de son écart type représente son poids (i.e. son importance). Cet algorithme calcule, à partir d'une liste d'IRIS de départ (e.g. tous les IRIS où un utilisateur a habité), un vecteur profil qui contiendra les poids de chaque critère. Ensuite la similarité cosine est appliquée entre ce vecteur profil et les IRIS candidats. Les IRIS ayant obtenu les meilleurs résultats seront alors recommandés. Un autre algorithme que la similarité cosine peut être appliqué. La Figure 5 illustre un exemple de calcul de vecteur profil et de calcul de similarité. Ici, c'est l'IRIS 3 qui correspond le plus au vecteur profil (similarité proche de 1) construit via la liste des IRIS de départ.

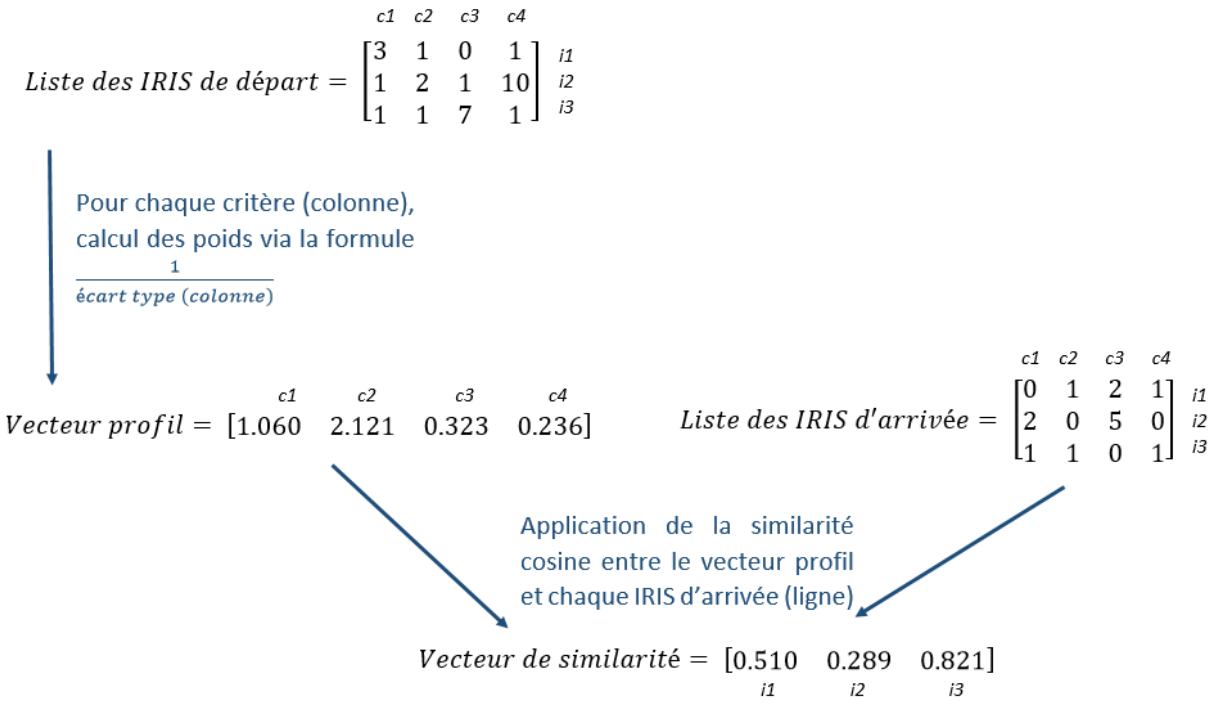


FIGURE 5 – Application de l'écart-type pour calculer la similarité entre des IRIS

Troisièmement, la technique du clustering a été implémentée. Le clustering est une méthode de regroupement de données qui ont des similarités en un ensemble de « nuages », que l'on appelle clusters. Cette méthode permet de détecter des grandes catégories au sein de données. L'implémentation du clustering permet, dans le cadre de la recommandation, deux utilisations. La première est le regroupement des quartiers par types. En effet, le clustering va regrouper les quartiers qui se ressemblent et ainsi faire apparaître des « profils » de quartiers. La deuxième est la recommandation de quartiers. Après avoir appliqué le clustering sur les IRIS candidats à la recommandation, le cluster dans lequel serait rangé l'IRIS de départ est le cluster que l'on souhaite recommander. Le prototype comprend l'implémentation de 8 algorithmes qui utilisent le clustering : **affinity propagation, agglomerative clustering, Birch, DBSCAN, feature agglomeration, KMeans, Mean Shift, Mini Batch KMeans et spectral clustering**. Ils permettent, entre autres, de montrer les zones les plus animées, les centres villes et de regrouper les IRIS entre eux.

Enfin, 5 algorithmes de type SVM (Support Vector Machine) ont été codés. Les

SVM sont des modèles d'apprentissage supervisé auxquels on associe des algorithmes d'apprentissage qui analysent les données pour faire de la classification ou de la régression. Le choix entre la classification et la régression peut être déterminé par le type de sortie de l'algorithme. La classification est utilisée lorsque la sortie est une valeur discrète (e.g. une catégorie) tandis que la régression est utilisée dans le cas d'une sortie continue (e.g. une valeur). Selon les données de départ, chaque élément appartiendra à une classe. L'algorithme d'apprentissage va construire un modèle et sera capable de prédire la classe d'un nouvel élément. Il a donc besoin de données d'apprentissage (« *training data* »). Dans notre cas, ce sont les IRIS de départ qui servent de données d'entraînement. Pour one-class-svm, une seule classe est nécessaire donc les IRIS de départ représentent la classe (classe 1). Pour les autres algorithmes de type SVM, deux classes sont nécessaires. Dans ce cas, les IRIS de départ sont de classe 1, et les voisins directs de chaque IRIS de départ (qui ne font pas partie des IRIS de départ) sont de classe 0. Les coefficients appris lors de l'entraînement (« *training* ») servent de vecteur représentatif des IRIS de départ. Enfin, pour recommander, soit la similarité cosine soit le clustering sont appliqués en utilisant ce vecteur représentatif. En effet, ce vecteur peut être utilisé comme IRIS de départ avec un autre algorithme (e.g. mesure cosine ou clustering). Lorsque les données ne peuvent pas être classifiées, on ne peut pas utiliser l'apprentissage supervisé. Il existe alors une autre approche, celle de l'apprentissage non-supervisé. Cette approche tente de trouver un regroupement entre les données. Dans le prototype, les données peuvent être classifiées donc c'est l'apprentissage supervisé qui a été retenu.

6 Prototype et résultats préliminaires

Dans cette section, l'implémentation du prototype et l'évaluation sur données réelles seront présentées.

6.1 Prototype

Le prototype permet de faire la recommandation d'un IRIS ou d'appliquer les algorithmes sur les IRIS au niveau de la France. Dans la suite, nous montrerons les principales fonctionnalités du prototype en prenant comme étude le département du Rhône. Plusieurs cas de figures sont possibles et les données en entrée peuvent varier (soit un IRIS, soit une liste d'IRIS). Le premier cas est la recommandation simple pour un IRIS. Dans ce cas, c'est la similarité cosine qui est appliquée. Le second est le clustering simple qui permet de regrouper et de visualiser les IRIS similaires (i.e. qui appartiennent au même cluster). Dans ce cas, le clustering est appliqué sur tous les IRIS. Le dernier est la recommandation via le clustering. Dans ce cas, le clustering s'applique sur les IRIS du Rhône afin de les regrouper en clusters et de déterminer dans quel cluster mettre l'IRIS à recommander. Les IRIS qui composent le cluster le plus pertinent seront les IRIS recommandés. Les captures d'écran ci-dessous illustrent ces cas de figure. Par défaut les IRIS sont bleus, l'IRIS sélectionné est violet et les IRIS recommandés sont colorés. Le prototype est sous la forme d'un site web. La structure a été codée en HTML et la mise en page en CSS. Le JavaScript a été utilisé pour les interactions avec la carte et les menus. Le lien entre les choix faits par l'utilisateur et l'exécution des fonctions est réalisé avec des requêtes AJAX. Comme mentionné précédemment, la recommandation est faite en Python.

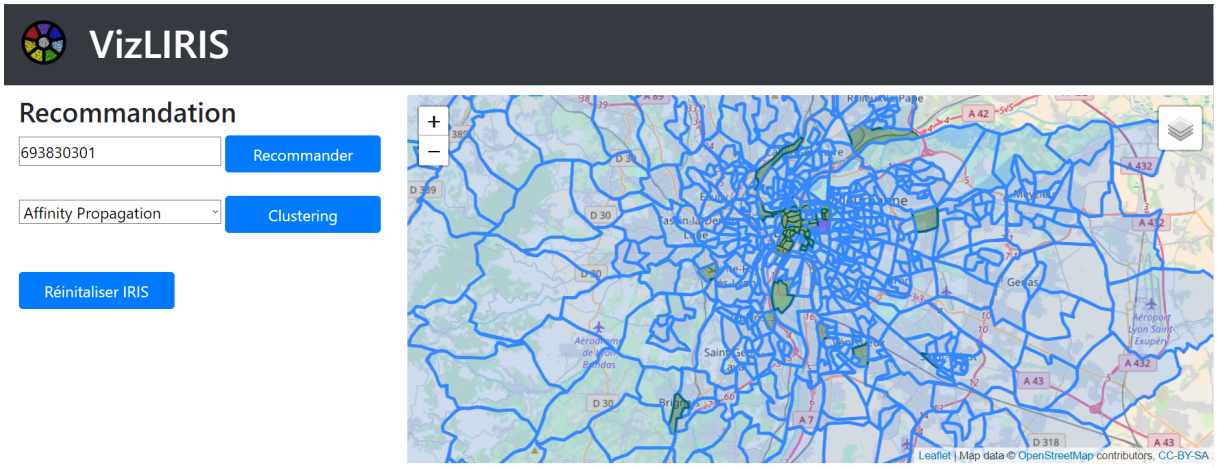


FIGURE 6 – Recommandations (vert) pour l'IRIS de la Part-Dieu

La Figure 6 montre les recommandations proposées pour l'IRIS de la Part-Dieu grâce à l'algorithme de la similarité cosine. Les quartiers recommandés sont des IRIS animés puisque celui de la Part-Dieu est animé du fait de son nombre de restaurants (indicateur majeur pour le critère d'animation).

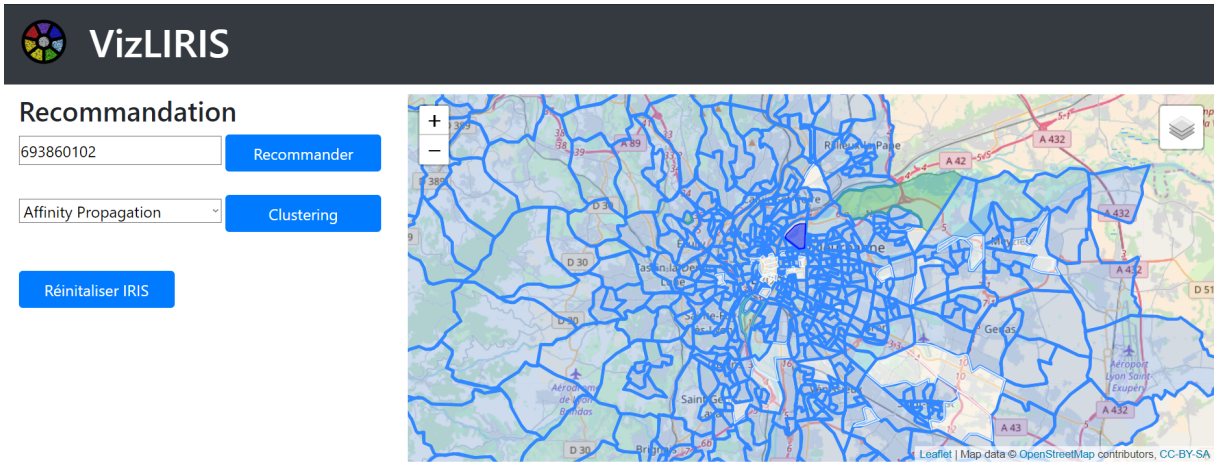


FIGURE 7 – Recommandations (blanc) pour l'IRIS du parc de la Tête d'or

La Figure 7 montre les recommandations pour le parc de la Tête d'or, toujours avec l'algorithme de la similarité cosine. La similarité cosine renvoie plutôt des IRIS d'activité puisque le parc de la Tête d'or recense peu de services.

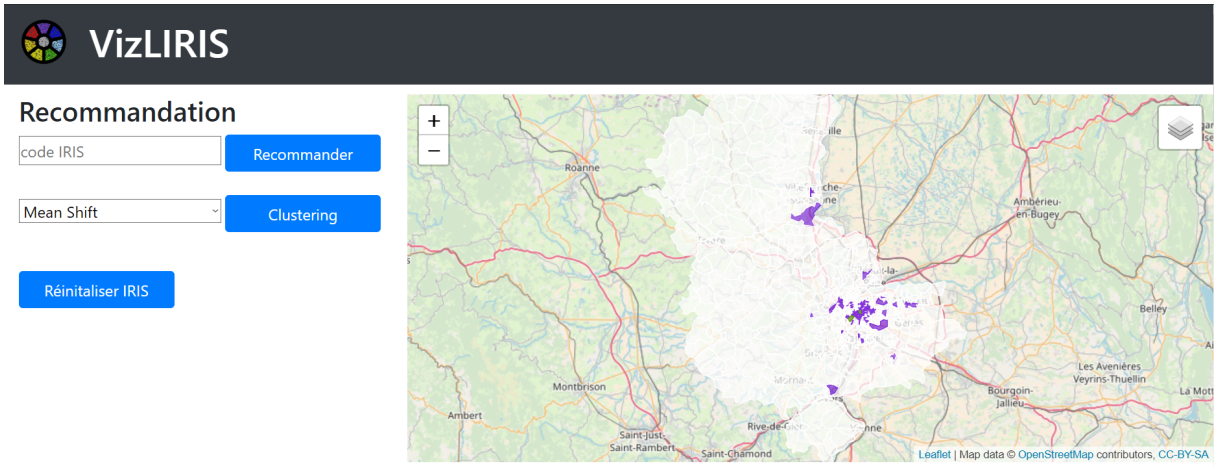


FIGURE 8 – Clustering appliqué via l'algorithme Mean Shift

La Figure 8 montre l'application du clustering sur les IRIS du Rhône avec l'algorithme Mean Shift. Cet algorithme permet de faire apparaître les zones les plus denses, e.g. les quartiers animés de Lyon et les centres villes des autres villes comme Tarare, Villefranche ou Givors.

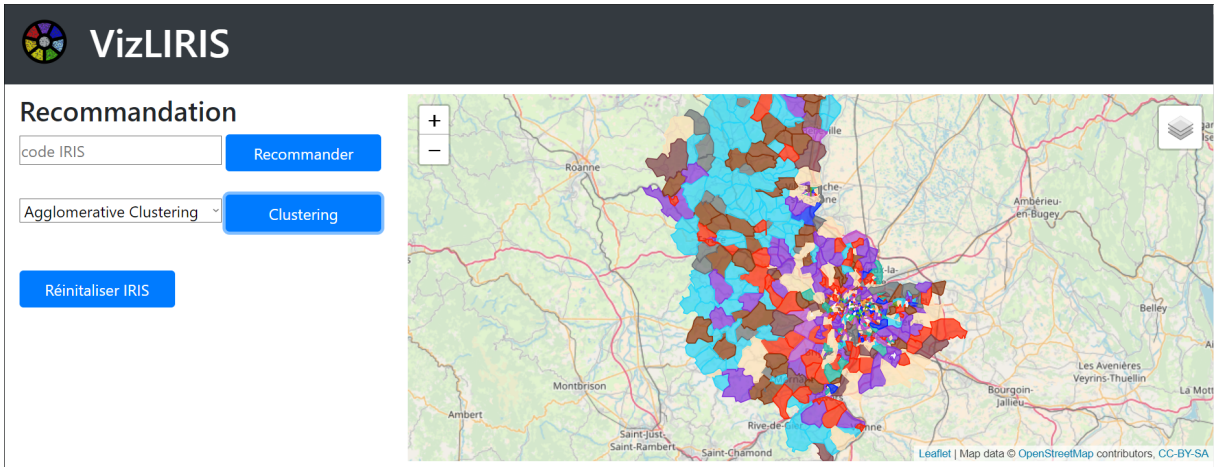


FIGURE 9 – Clustering appliqué via l'algorithme Agglomerative Clustering

La Figure 9 montre l'application du clustering sur les IRIS du Rhône avec l'algorithme Agglomerative Clustering. Cet algorithme permet de démarquer les centres (très colorés) des zones moins denses (majoritairement bleues).

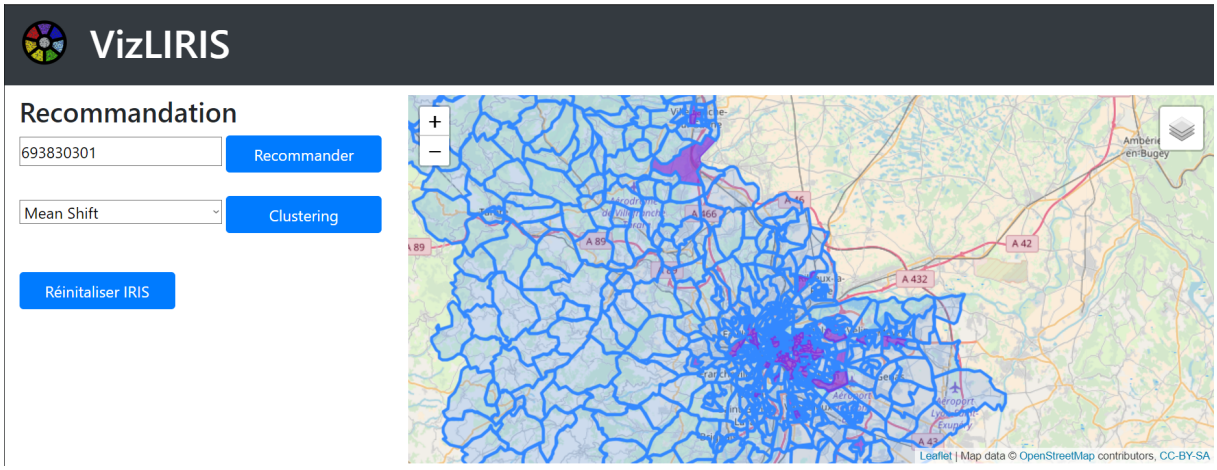


FIGURE 10 – Recommandations (violet) pour l'IRIS de la Part-Dieu via le clustering

La Figure 10 montre la recommandation via le clustering. Les clusters sont d'abord calculés grâce à l'algorithme Mean Shift. Ensuite, le module de recommandation cherche dans quel cluster l'IRIS de la Part-Dieu serait placé. Le cluster déterminé est alors le cluster le plus pertinent donc ses IRIS sont recommandés. Ce sont les zones animées des centres du Rhône qui sont recommandées, ce qui est pertinent par rapport à leurs indicateurs.

6.2 Protocole

Lorsqu'un système de recommandation est développé, il est nécessaire d'évaluer sa capacité à répondre aux objectifs définis. Cela nécessite à la fois un jeu de données et

des métriques d'évaluation. Les métriques, mesures quantifiables de la performance, permettent d'évaluer les algorithmes implémentés. Par exemple, lorsqu'un utilisateur lance une requête (e.g. une recommandation de quartiers), il s'attend à avoir un certain nombre de réponses qui correspondent à sa question. Ses attentes peuvent se caractériser par une mesure de précision/rappel. Ce couple de mesures permet de mesurer les performances d'un algorithme. La précision représente la probabilité qu'un élément recommandé soit pertinent et se définit par :

$$précision = \frac{nombre\ d'IRIS\ pertinents\ trouvés}{nombre\ d'IRIS\ pertinents\ stockés}$$

Le rappel représente la probabilité qu'un élément pertinent soit recommandé et se définit par :

$$rappel = \frac{nombre\ d'IRIS\ pertinents\ proposés\ à\ l'utilisateur}{nombre\ d'IRIS\ total\ proposés\ pour\ une\ recommandation}$$

Un système parfait aurait une précision à 1 (aucune erreur dans les éléments recommandés) et un rappel à 1 (tous les éléments pertinents sont recommandés). En plus des mesures de précision et de rappel, une troisième mesure d'évaluation peut être utilisée, c'est la moyenne harmonique. Elle combine la précision et le rappel et se définit par :

$$f\ measure = 2 \times \frac{précision \times rappel}{précision + rappel}$$

Des mesures telles que les vrais et faux positifs ainsi que les vrais et faux négatifs permettent aussi d'évaluer un algorithme. Les vrais positifs sont les IRIS pertinents qui sont détectés comme pertinents alors que les faux positifs sont les IRIS non pertinents détectés pertinents. Les vrais négatifs sont les IRIS non pertinents détectés en tant que tel alors que les faux négatifs sont les IRIS pertinents détectés comme non pertinents. Ces mesures permettent de calculer la sensibilité et la spécificité d'un algorithme. La sensibilité mesure la capacité d'un algorithme à donner un résultat positif quand l'hypothèse est vérifiée tandis que la spécificité mesure la capacité à donner un résultat négatif quand l'hypothèse n'est pas vérifiée. La sensibilité permet de détecter tous les IRIS qui ne sont pas à recommander car ils ne sont pas pertinents, sachant que cet ensemble ne contient pas forcément que des IRIS non pertinents. À l'inverse la spécificité permet de ne détecter que les IRIS non pertinents.

Après avoir défini les métriques d'évaluation, il est important de définir le jeu de données. Il n'existe pas de benchmark sur la France pour les IRIS. Nous avons donc utilisé les données réelles que nous fournit Home in Love. Sur la centaine de clients fournis, seuls 67 profils étaient valides (i.e. un IRIS de départ, un IRIS d'arrivée et un IRIS de travail). Le Tableau 1 illustre les statistiques des 67 profils valides et des IRIS candidats à la recommandation. Pour environ 50% des profils (30/67), la distance entre l'IRIS de travail et l'IRIS du nouveau domicile est inférieure à 5 kilomètres (première ligne de la Figure 1). Cela ne restreint pas forcément le nombre d'IRIS candidats à la recommandation, car il y a des profils avec une distance inférieure à 5 kilomètres qui contiennent plusieurs centaines d'IRIS candidats (e.g. dans les grandes villes). La somme des deux dernières lignes du Tableau 1 montre qu'environ les 3/4 des profils ont une zone

de recherche contenant plus de 50 IRIS candidats à la recommandation. Ce nombre peut monter jusqu'à plusieurs milliers d'IRIS candidats pour quelques cas.

		Nombre de profils
Distance (en kilomètres)	$0 <$ distance < 5	30
	$5 <$ distance < 10	16
	$10 <$ distance < 20	9
	distance > 20	12
Nombre d'IRIS candidats	candidats < 10	3
	$10 <$ candidats < 50	13
	$50 <$ candidats < 500	38
	candidats > 500	13

TABLE 1 – Statistiques sur les profils et les IRIS candidats

6.3 Résultats

Les résultats préliminaires de notre prototype permettent de montrer la pertinence de la recommandation sur un top 10. Les résultats ci-dessous sont effectués sur 67 profils et les 50000 IRIS qui composent la France. Chaque profil inclut l'IRIS de départ, l'IRIS d'arrivée et l'IRIS du futur lieu de travail de l'utilisateur. L'IRIS de départ est le lieu de vie actuel et l'IRIS d'arrivée est le nouveau domicile, dans le cadre d'une mutation professionnelle par exemple. À partir de l'IRIS de départ, nous souhaitons vérifier que les algorithmes utilisés obtiennent (parmi les 10 recommandations) l'IRIS d'arrivée (où la personne a finalement emménagé). C'est le rappel qui indique si l'IRIS recherché fait partie des IRIS recommandés. Les IRIS candidats sont détectés à partir de l'IRIS de travail. En effet, la personne cherchera à habiter à proximité de son nouveau lieu de travail.

Les algorithmes de type SVM qui nécessitent au moins deux classes (e.g. linear SVM et nu SVM) n'ont pas été inclus car les expérimentations ont été réalisées avec un seul IRIS de départ. C'est pourquoi seulement l'algorithme one-class SVM a été utilisé pour ces expérimentations. Le Tableau 2 montre les résultats obtenus lors des expérimentations.

Stratégie	Faux positifs	Vrais positifs	Faux négatifs	Précision	Rappel	F-measure
Cosine similarity	**645**	13	54	**0.02**	0.19	**0.04**
Standard deviation	**645**	13	54	**0.02**	0.19	**0.04**
Spectral clustering	29751	**60**	**7**	0.0	**0.9**	0.0
Agglomerative clustering	29751	**60**	**7**	0.0	**0.9**	0.0
DBSCAN	29751	**60**	**7**	0.0	**0.9**	0.0
Mini Batch k-means	5573	18	49	0.0	0.27	0.01
K-means	5739	18	49	0.0	0.27	0.01
Meanshift	18586	36	31	0.0	0.54	0.0
Affinity propagation	3406	8	59	0.0	0.12	0.0
Birch	6168	20	47	0.0	0.3	0.01
One-class SVM	6316	21	46	0.0	0.31	0.01

TABLE 2 – Résultats des évaluations pour 67 profils et 50000 IRIS

Les résultats obtenus avec la similarité cosine sont encourageants. En effet, le rappel est de 0.19, soit 20% donc environ une recommandation sur 5 est trouvée. Mon approche avec l'écart-type (standard deviation) propose des résultats cohérents avec la mesure cosine, qui est une mesure très utilisée. Elle obtient un rappel de 0.19 comme la similarité cosine. C'est un résultat préliminaire encourageant.

Le clustering obtient de bons résultats car son rappel est proche de 1. Certains algorithmes de clustering nécessitent de spécifier le nombre de clusters tandis que d'autres sélectionnent ce nombre automatiquement. Lorsqu'il faut fournir ce nombre, le nombre de clusters est défini par le nombre d'IRIS candidats divisé par 10, pour avoir en moyenne 10 IRIS par cluster. Cependant, en pratique, les algorithmes peuvent construire des clusters regroupant de nombreux IRIS (e.g. le spectral clustering trouve environ 30000 IRIS pour les 67 profils, soit des clusters d'une taille de 445 IRIS en moyenne). Même si une majorité d'IRIS d'arrivée sont trouvés, ceux-ci sont donc perdus au milieu d'un grand nombre de recommandations, ce qui rend l'algorithme peu pratique à utiliser.

6.4 Discussion

Pour terminer, nous analysons plus finement ces premiers résultats afin d'extraire des perspectives pour la fin de mon stage. L'hypothèse de départ choisie est très forte.

En effet, elle considère que les utilisateurs cherchent un quartier semblable à celui où ils habitaient. Or, cette hypothèse n'est pas toujours vraie. En effet, les sociologues ont analysé les pentes de carrière qui montrent que les utilisateurs cherchent, dans la plupart des cas, des quartiers différents de leur quartier actuel car leur situation (notamment financière) a évolué positivement lors de la mutation. Par exemple, les utilisateurs qui sont en ascension positive voient souvent leur salaire augmenté et donc peuvent chercher un quartier/logement au plus près de leurs attentes. Les alternants faussent également cette hypothèse puisqu'ils partent de chez leurs parents, donc ils cherchent souvent un quartier différent de celui de leurs parents (e.g. des quartiers à résidences étudiantes). Même constat pour les utilisateurs qui ont des liens forts avec leur ancien domicile ou leur famille car ils sont prêts à faire les allers-retours le week-end.

Les différents algorithmes utilisés ont chacun beaucoup de paramètres. Il faudrait pousser encore plus loin les expérimentations pour essayer de personnaliser au mieux les recommandations. En effet, les paramètres de ces algorithmes influencent les résultats obtenus. Par exemple, pour les algorithmes K-Means et Spectral Clustering, le nombre de clusters à former peut être modifié. Pour l'algorithme DBSCAN, c'est la taille du voisinage qui peut être changée. Ainsi, la paramétrage de ces algorithmes peut faire varier la précision des résultats obtenus et donc affiner les recommandations. La fin de mon stage devrait permettre d'approfondir cette analyse des paramètres.

Notre approche se base pour l'instant sur les données de l'INSEE et des prix immobiliers mais des aspects sociologiques seront aussi à prendre en compte. Le post-doctorant est en train de convertir les profils donnés par la start-up HiL et d'en extraire des aspects sociologiques (e.g. la situation familiale, l'âge, la catégorie socio-professionnelle), dans le respect de la CNIL (Commission Nationale de l'Informatique et des Libertés). Ces aspects seront à intégrer dans un second temps afin d'améliorer les recommandations. Il n'existe pas de critères sociologiques comme source de données, c'est pourquoi le travail d'analyse des sociologues permettra de prendre en compte ces critères. Le regroupement des indicateurs est une première ébauche que les sociologues vont affiner une fois qu'ils auront mieux identifié les critères importants. Ces perspectives se placent dans l'ajout de « données utilisateur », comme l'indique le rond de la Figure 2.

7 Conclusion et perspectives

Je suis ravie d'avoir pu faire mon stage au LIRIS et ce mois et demi est une très belle expérience pour moi. La première partie de mon stage s'est penchée sur la découverte de la recommandation. J'ai ainsi pu lire un article sur la recommandation immobilière [7] et en faire un résumé. J'ai aussi fait un état de l'art sur les algorithmes de recommandation (voir Section 2). Il m'a permis de lire des articles scientifiques en anglais, de synthétiser les données lues et de les organiser en une présentation claire et concise. Enfin, je me suis renseignée sur les méthodes d'apprentissage des algorithmes. En parallèle de mon travail au LIRIS, j'ai assisté aux réunions organisées par Home in Love. Elles ont pour objectif de faire le point entre les avancées informatiques, les analyses sociologiques et les souhaits de la start-up. Cette collaboration entre différents domaines est pour moi une découverte d'une grande richesse. Ce stage m'a aussi été très profitable sur le plan technique puisque la deuxième partie de celui-ci se concentre sur le prototype. J'ai pu découvrir le langage Python ainsi que les requêtes AJAX. J'ai aussi pu mieux comprendre comment imbriquer les langages entre eux pour séparer la vue et le traitement. Grâce au module d'intégration de données, j'ai découvert la gestion de données hétérogènes.

La suite de mon stage me permettra de continuer à travailler sur le prototype ainsi que d'approfondir les expérimentations actuelles. Il me permettra éventuellement d'intégrer des critères sociologiques aux recommandations. Le prochain objectif est de soumettre un article. Il faudra d'abord résoudre les différents problèmes d'ergonomie et de performances du prototype ainsi que préparer les interfaces pour dérouler les scénarios que nous proposerons dans l'article. Ensuite, je participerai à la rédaction de l'article de démonstration qui sera soumis à EGC (conférence Extraction et de Gestion des Connaissances). La soumission de cet article est prévue pour octobre 2019.

Ce stage est une belle expérience professionnelle en plus d'avoir pu faire mes premiers pas en recherche. Sur le plan humain, j'apprécie beaucoup la pédagogie et l'implication de mes maîtres de stage. Ils prennent le temps de m'expliquer clairement mes missions et de m'aider lorsque je n'arrive pas à résoudre certains problèmes techniques. Ce stage m'a permis d'apprendre à mieux m'organiser et à être plus autonome dans la recherche de solutions. J'ai énormément apprécié ce mois et demi au sein du LIRIS et cette expérience me conforte dans mon envie de faire un master recherche.

8 Annexes

Les annexes ci-dessous illustrent l'instanciation du vecteur des indicateurs regroupés ainsi qu'un aperçu du prototype.

$$\begin{bmatrix} animationCommerceNonalimentaire \\ animationCommerceAlimentaireGrandesurface \\ animationCommerceAlimentaireProximite \\ animationCulturel \\ animationDivertissement \\ csp \\ educationCreche \\ educationPrimairePrive \\ educationPrimairePublic \\ educationSecondaireCycle1Prive \\ educationSecondaireCycle1Public \\ educationSecondaireCycle2GeneralPrive \\ educationSecondaireCycle2GeneralPublic \\ educationSecondaireCycle2ProfessionnelPrive \\ educationSecondaireCycle2ProfessionnelPublic \\ educationSuperieurPrive \\ educationSuperieurPublic \\ espacevert \\ logementAnnee \\ logementResidence \\ logementResident \\ logementType \\ loisir \\ securite \\ serviceActionsociale \\ serviceDiversPrive \\ serviceDiversPublic \\ serviceEmploi \\ serviceJustice \\ serviceSante \\ transportBusmetrotram \\ transportLonguedistance \\ transportVelo \end{bmatrix} = \begin{bmatrix} 209.0 \\ 4.0 \\ 7.0 \\ 0.0 \\ 102.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 2.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \\ 0.0 \\ 0.0 \\ 39.0 \\ 126.0 \\ 5.0 \\ 1.0 \\ 1.0 \\ 0.0 \\ 1.0 \\ 0.0 \end{bmatrix}$$

FIGURE 11 – Exemple d'instance des indicateurs regroupés pour l'IRIS de la Part-Dieu

VizLIRIS

Recommandation

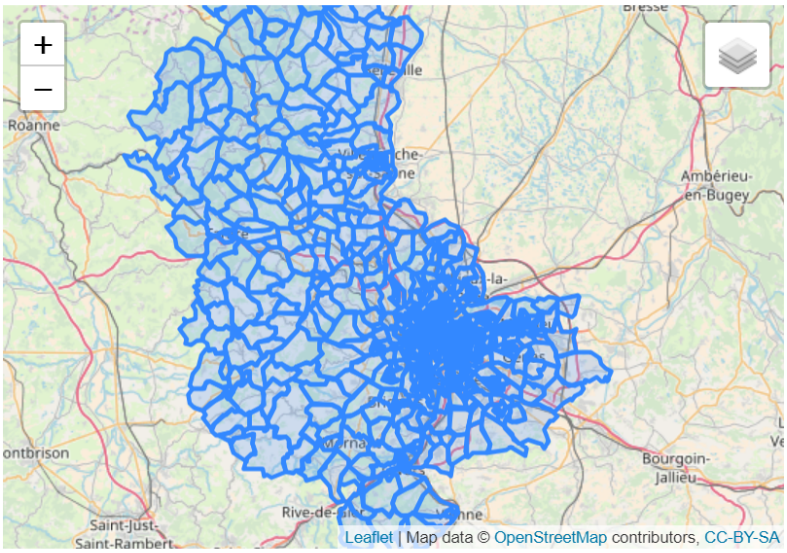
692660201

Recommander

Affinity Propagation

Clustering

Réinitaliser IRIS



Informations sur l'IRIS sélectionné

Code attribut	Nom attribut	Valeur
C101	École maternelle	0
C104	École élémentaire	0
C201	Collège	0
C301	Lycée d'enseignement général et/ou technologique	0
C302	Lycée d'enseignement professionnel	0
C303	Lycée technique ou/et professionnel agricole	0
C501	Unité de Formation et de Recherche	2
C502	Institut universitaire	1
C503	Ecole d'ingénieurs	3
C504	Enseignement général supérieur privé	0
C505	Ecole d'enseignement supérieur agricole	0
C509	Autre enseignement supérieur	4
C701	Résidence universitaire	0
C702	Restaurant universitaire	1
NB_A101	Police	0
NB_A104	Gendarmerie	0
NB_A105	Cour d'appel	0
NB_A106	Tribunal de grande instance	0

FIGURE 12 – Aperçu du prototype avec les indicateurs pour l'IRIS de la Part-Dieu

Références

[1] A. K. Cline and I. S. Dhillon, *Computation of the Singular Value Decomposition*, CRC Press, jan 2006.

[2] A. Halevy, A. Rajaraman, and J. Ordille, *Data integration : the teenage years*, in VLDB '06 : Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, 2006, pp. 9–16.

[3] F. Isinkaye, Y. Folajimi, and B. Ojokoh, *Recommendation systems : Principles, methods and evaluation*, Egyptian Informatics Journal, 16 (2015), pp. 261 – 273.

[4] M. J. Pazzani and D. Billsus, *Content-based recommendation systems*, in The Adaptive Web, 2007.

[5] X. Su and T. M. Khoshgoftaar, *A survey of collaborative filtering techniques*, 2009.

[6] Y. xiong Wang and Y. jin Zhang, *Nonnegative matrix factorization : A comprehensive review*, IEEE TRANS. KNOWLEDGE AND DATA ENG, (2013), pp. 1336–1353.

[7] X. Yuan, J.-H. Lee, S.-J. Kim, and Y.-H. Kim, *Toward a user-oriented recommendation system for real estate websites*, Information Systems, 38 (2013), pp. 231 – 243.