Datamining uber-rides

AZOUI, Aymen
azoui.aymen@etu.univ-lyon1.fr
p1612255

Vincent, Toinon
vincent.toinon@etu.univ-lyon1.fr
p1408764

Etienne, Vareille
etienne.vareille@etu.univ-lyon1.fr
p1719493

January 2021

1 Introduction

De plus en plus de données sont disponibles sur internet. Le grand public ne s'étonne plus de savoir que de nombreuses informations personnelles sont disponibles en ligne, et ne s'en méfie même plus.

Dans le cadre de l'UE data mining, nous avons opté pour un projet sur l'élaboration d'une représentation des centres d'intérêt et des habitudes (emploi du temps) d'une personne à partir des données récupérées par l'application Uber qu'elle utilisait pour se déplacer. Ces données ont été mises à disposition par un utilisateur de Kaggle, à propos de ses propres voyages. Il nous a semblé que c'était l'occasion parfaite de nous intéresser à ce que des données, volontairement mise à disposition de tous, pouvait nous dire sur une personne.

Ce projet à donc deux buts. Le premier consiste à produire de la valeur ajoutée sur les données. Il nous a semblé que les trajets Uber pouvait donner des informations intéressantes pour des agences publicitaires ou des algorithmes de publicité ciblée. Plus particulièrement, le fait de pouvoir diffuser des publicités aux moments précis pendant lesquels elles sont susceptibles d'intéresser la personne est attrayant.

Le second but consiste en la sensibilisation du public aux données personnelles. Nous souhaitons monter qu'il est possible d'obtenir des informations caractéristiques de la personne étudiée, à partir de données qui sont parcellaires (un utilisateur de Uber prend en général peu ce mode de transport, ce qui rend intuitivement l'analyse pour une personne précise plus difficile).

Pour faire celà, nous avons procédés en trois étapes : la première a consisté en l'enrichissement des données. Grâce à l'API Google Places, nous avons récupéré des informations sur le voisinage des destinations de chaque trajet. Celà a permis d'avoir des informations supplémentaires sur le but possible de chaque trajet.

La deuxième a été l'utilisation de Knime, comme outil d'exploration préliminaire, mais aussi comme outil d'analyse à part entière. Par simplicité, toute l'analyse faite avec Knime a été résumée en une seule section.

La troisième a été de trouver un algorithme pour essayer d'exploiter plus particulièrement les tendances temporelles des lieux renvoyés par Google Places. Comme nous le verrons plus tard, gérer ces lieux nous a posé problème.

Tous les algorithmes utilisés sont chargés sur le github du groupe :
`https://github.com/azouiaymen/Data-minig`

2 Jeu de données

Les donnés que nous avons utilisées pour ce projet proviennent du challenge **Uber-rides** disponible sur le site de kaggle.

https://www.kaggle.com/stantyan/uber-rides

2.1 Données Uber

Les donnés contiennent tous types d'informations concernant le trajet .

On peut donc y trouver : les coordonnées du lieu de départ, de l'endroit d'arrivée, l'heure du départ ainsi que celle d'arrivée, le prix du trajet, le sexe et l'âge du conducteur, le type du vehicule, la météo , la distance parcourue ...etc. Toutes ces informations ne nous ont pas intéressées. Par exemple, la note de ressenti du passager, ou les informations relatives au véhicule ou à son conducteur. Ces dernières en particulier ne dépendent pas du passager, il est donc peu utile de les utiliser dans notre optique.

Les données sont dans un format duquel on ne peut pas retirer grand-chose , vu que les endroits de départ et d'arrivée ne contiennent pas de description du lieu, et que parfois l'adresse des lieux ne contient que le nom de la rue (pas de numéro de porte) ce qui nous renvoie que une quantité d'information relativement faible par rapport au buts de ces déplacements. Il a donc fallu enrichir le jeu de données de base. Nous avons pour ce faire utilisé l'API Google Places.

2.2 API Google places

Notre idée de base était de trouver les lieux d'intérêt de l'utilisateur de Uber à partir des lieux de départ et les lieux d'arrivée, mais vu que les donnés qu'on avait ne contenaient pas beaucoup d'informations, nous avons eu l'idée d'utiliser les coordonnés GPS (latitude et longitude) pour essayer de récupérer les lieux de dépôt ainsi que les différentes informations sur l'endroit (tel que le type d'endroit , les points d'intérêt les plus proches de ce lieu...etc). Pour cela nous avons pensé à utiliser l'API Google places, et l'API Google search.

Après avoir appris l'utilisation des API Google ainsi que le fonctionnement des requêtes sous python et l'utilisation des fichiers du type json ; nous avons d'abord envoyé les cordonnées avec des requêtes à l'API Google places qui les prennait en paramètre, puis l'API se chargait de renvoyer l'adresse exacte (nom de la rue, numéro du bâtiment...) sous format JSON.

Une fois l'adresse récupérée, elle était envoyée cette fois-ci à l'API Google Search qui renvoyait quant à elle tous les points d'intérêt autour de l'endroit (un magasin, une gare, une bouche de métro, un distributeur ATM, un hôpital... etc). Les données étaient aussi sous le format JSON.

Tout le code qui nous a permis de faire cela est détaillé dans le fichier jupyter notebook **Geo-Uber.ipynb**, après extraction des résultats du json on a réécrit ces derniéres dans le bon ordre dans le fichier **résultats-api.csv**.

La difficulté principale de cette API, est qu'elle est à destination des professionnels. Elle s'adapte aussi aux utilisateurs qui ne cherchent pas à l'intégrer dans un modèle économique, mais le cadre reste compliqué. Le service est payant à partir d'un certain nombre de transactions. Les transactions sont de différents types, et en fonction du nombre de requêtes, le prix peut exploser. Les utilisateurs se voient accorder un crédit gratuit à la création de leur compte. Nous avons dû vérifier que notre usage nous permettrai de rester dans des volumes de requêtes sous la barre à partir de laquelle tout devient payant. Le moins sécurisant était la création du compte qui implique de donner une carte bleu/moyen de payement à l'avance. Nous avons passé pas mal de temps à éplucher la documentation pour être sûr que tout se passerait bien, ainsi que s'assurer que python était bien adapté pour

utiliser l'API.

Une fois celà fait, nous disposions d'un jeu de données enrichit, qui nous a permis d'aller plus en avant par rapport au jeu de données brut.

3 Exploration préliminaire des données

Une première exploration des données a été faite sous Knime. Nous avons pas à pas regardé la forme que prennait les données, et ce qu'un traitement et une approche standard pouvait apporter.

La première étape a été de restreindre le nombre de variables que nous avions. Peu d'informations nous intéressaient : les coordonnées GPS de départ et d'arrivée des trajets, la date et l'heure de départ/arrivée, et la liste des types de lieux (récupérés avec l'API Google) autour des points d'arrivée.

Un affichage des points sur une carte openstreetmap nous a montré que les trajets se concentraient sur trois grandes villes de Russie. En vue d'un clustering spatial, nous avons restreint les données à Saint Petersburg seule, ville qui concentrait la plupart des voyages. [3]

Nous avons ensuite tenté d'identifier les habitudes de l'utilisateur. Il fallait en premier filter les trajets : nous cherchions à écarter les trajets isolés. Pour celà, nous avons appliqué l'algorithme DBSCAN, en sélectionnant une taille de cluster minimale conséquente, puis filtré les clusters pour éliminer le bruit trouvé. La sélection de variables pour le clustering ne comportait que les coordonnées GPS du point d'arrivée, avec une distance euclidienne. [4]

Une fois celà fait, on pouvait déjà identifier des regroupements remarquables de points en les affichant sur une carte openstreetmap : certains autours de zones commerciales, certains autour d'un hopital, d'autres difficilement assimilables à des lieux évidents.

Un autre traitement que nous avons essayé, est de faire un clustering similaire au précédent, mais en incluant l'heure de la journée. En équilibrant et normalisant les données, il a été possible de faire un clustering rassemblant

les points séparés de quelques heures au plus (modulo la date bien sûr) et de coordonnées GPS proches. Les clusters trouvés ont été bien moins nombreux.

On peut faire un clustering uniquement sur l'heure, et ensuite utiliser un itemset finder sur chaque cluster, dans l'espoir de trouver des horaires sur lequels certains mots clés sont très présents. Mais le volume de mots clés n'a pas permis de trouver des résultats concluants.

Un résultat intéressant est l'émergence de deux zones de Saint Petersbourg qui rassemblaient des trajets à beaucoup d'heures différentes. On peut supposer que ces points étaient les lieux d'habitations ou de travail/étude de la personne étudiée. Une analyse avec Knime peut donc trouver des informations comme le lieu d'habitation probable [5], éventuellement le lieu de travail, et les habitudes de consommations comme le fait que la personne utilise de grands centres commerciaux. Des informations privées sensibles comme l'utilisation probable d'un hopital ou d'une maison de santé peuvent aussi être supposées.

4 Analyse des données

4.1 But de l'analyse

Comme définit dans l'introduction, nous nous intéressons aux données personnelles qu'on peut faire émerger des données Uber. Ces données sont recherchées dans deux buts : créer de la valeur sur les données, et renseigner sur la vulnérabilité des données personnelles.

Les données sont essentiellement spaciales. C'est pour celà que nous avons utilisé premièrement python pour essayer différents algorithmes de clustering, afin de sélectionner des algorithmes compatibles avec les données.

Nous avons aussi travaillé sur les mots clés fournis par l'API google. Ces mots clés sont difficiles à traiter (on expliquera plus tard pourquoi), il fallait donc trouver un moyen original de trouver des informations à partir d'eux.

4.2 Clustering spatial

4.2.1 Problématique

Ici on a une problématique simple: est-il possible de détecter des zones géographiques qui peuvent être qualifiées de points d'intérêt pour la personne ciblée et dans ce cas là pouvons-nous identifier de façon plus ou moins précise ses zones d'activité, voire son addresse? (ou bien une zone dans laquelle il es probable que la personne réside). A partir des données obtenues, on va donc tester différents algorithmes de clustering et essayer de trouver celui qui permettra d'avoir une représentation réaliste des zones d'activité / d'intérêt de l'utilisateur Uber. Pour cela, nous avons utilisé des méthodes comme celle du clustering hiérarchique, MeanSift, Affinity Propagation ou encore le Clustering Spectral dont les sorties sont décrites pour la plupart en Annexe C.

4.2.2 Clustering Hiérarchique

Les résultats du clustering hiérarchique sont disponibles en Annexe 2. Le problème dans ce résultat est que la ville est scindée en deux gros clusters et les autres clusters ne représentent très peu de points ce qui donne une importante dispartité dans le clustering.

4.2.3 Propagation d'Affinité

L'intérêt principal du clustering par propagation d'affinité est que cet algorithme marche très bien sur de petits / moyens jeux de données comme le nôtre et ne nécessite pas d'initialiser le nomre de clusters souhaités. Le fonctionnement est le suivant: a caque iteration l'algorithme va choisir un point qui est représentatif d'un maximum d'autres points afin d'en faire le centre d'un cluster jusqu'à ne plus obtenir de changement dans les centres. Le problème cependant est que les coordonnées géographiques sont toutes très proches sur la carte et donc on obtient des clusters qui se chevauchent et il est impossible d'en tirer des zones facilement identifiables ce qui rend l'utilisation de cet algorithme difficile.

4.2.4 MeanShift

L'algorithme MeanShift est un algorithme basé sur une métode de centroide, a chaque iteration de l'algorithme, les centres sont mis à jour de façon a ce

que le nouveau centre soit équivalent à la moyenne de son voisinage selon un rayon définit en paramètre. Ici on rencontre un problème similaire que celui du clustering hiérarchique. Effectivement, comme les point sont très proches, un centre de cluster va avoir un voisinage étendu et au final on aura une séparation en deux gros clusters avec d'autres petits clusters de quelques éléments chacun. Dans les deux cas on peut tout de même déduire deux zones importantes dans l'analyse du comportement du client Uber.

4.2.5 Clustering Spectral

C'est avec cet algorithme que l'on obtient le meilleur clustering: les données sont transformées en un graphe d'adjacence de type KNearestNeighbors ce qui permet de différencier les données par rapport à leur voisinage proche. L'algoritme effectue ensuite plusieurs fois un clustering KMeans sur des coupes du graphes qui a été construit et garde le meilleur clustering.

Au final, on obtient un groupe de 5 gros clusters sur la ville de St-Petersburg qui nous permettent de mettre en lumière des lieux et plus généralement des quartier dignes d'intérêt.

Plus tard, on isole une période de temps ou le client Uber est le plus susceptible de prendre le Uber pour rentrer ou partir de chez lui. On obtient de manière moins évidente que sur KNIME un résultat montrant la carte de la ville épurée avec quelques clusters dont un contenant un groupe très dense de points que l'on peut donc supposer être le domicile ou proche de celui ci (Annexe C).

4.3 Selection de mots clés remarquables

Cette partie concerne le fichier `python/analyseEntropy.ipynb` du github.

4.3.1 Problème posé

Dans le jeu de donnée obtenu en combinant l'API google Maps et les données Uber, des points de l'espace sont associés à une liste de lieux proches (magasins, parcs, ...). On définit donc une base de données qui comporte des transactions, qui chacune caractérise un point. Cette caractérisation peut par exemple permettre de faire de l'itemset mining, pour regrouper des points similaires ou trouver des agencements de lieux fréquents dans ces points.

Cette modélisation est limitée. En effet, l'utilisateur de Uber ne se rend pas à un point précis pour profiter de tous les lieux proches. Il y a de grandes chances qu'il se rende à un lieu en particulier. On ne peut donc pas totalement considérer les transactions comme exactes : les lieux trouvés représentent des possibilités, pas des certitudes. On sort du formalisme des transactions.

On voudrait identifier, parmi les lieux trouvés, lesquels sont les plus pertinents. On pourrait regarder point à point sur une carte quel lieu est le plus remarquable/plausible. Par exemple, lorsqu'un trajet Uber qui amène sur le parking d'une grande surface ou d'une gare, on peut estimer que l'utilisateur a pour but d'entrer dans ces structures, et pas de profiter d'un lieu proche. On voudrait un moyen automatique qui permette d'indentifier des mots clés intéressants. Mais il est difficile de trouver quel lieu est visité sans créer des critères de priorité à la main, compte tenu des données que nous avons.

On peut tout de même avoir des informations intéressantes sans avoir à identifier à la main chaque lieu probable. On a voulu ici tirer des informations de la répartition temporelle des visites des lieux. Le but est de trouver des lieux (i.e. mots clés) fréquentés à certains moments de la semaine, pour identifier des habitudes.

4.3.2 Méthode - L'entropie

On cherche à trouver des lieux fréquentés plus particulièrement à certains moments de la semaine. Pour celà, on sépare la semaine en 4×7 plages temporelles : matin, après-midi, soirée, nuit pour chaque jour.

On considère la notion suivante. Un mot-clé de lieu est intéréssant si il est fréquenté particulièrement sur certaines plages horaires. Plus il est fréquenté homogonènement, moins il est intéressant.

Pour formaliser cette notion, on peut utiliser l'entropie. On considère la répartition des occurences du mot clé sur les plages temporelles définies. Elle définit une distribution. En mesurant l'entropie de cette distribution, on trouve le caractère "aléatoire" du mot clé : plus l'entropie est faible, plus le nombre de plages horaires que ce mot clé concerne est faible. Et inversement.

On cherche donc des mots clés d'entropie faible.

4.3.3 Méthode - Ajuster par rapport au nombre d'occurences du mot-clé

La mesure de la fréquence d'occurence pour chaque plage horaire est fortement influencée par le nombre total d'occurences d'un mot clé. Dans la figure 1, on voit que les mots-clés ayant un nombre faible d'occurences dans le jeu de données ont une entropie plus faible en général, qui augmente rapidement vers un palier lorsqu'il y a assez d'occurences.

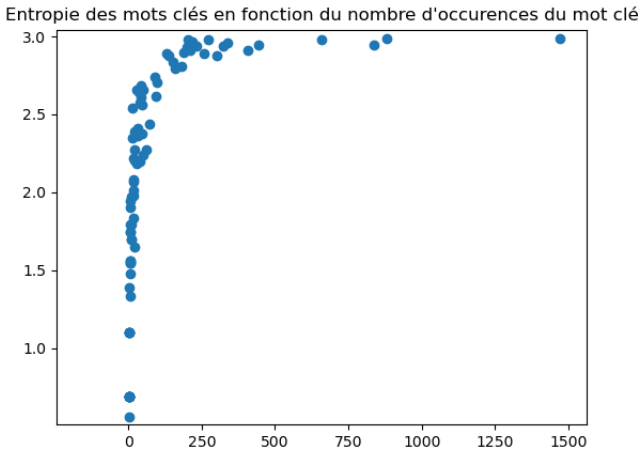

Figure 1: On observe des disparités entre l'entropie des mots clés à faible occurence et à forte occurence

Devant ce phénomène, on se rend compte que les mots clés d'entropie faible sont les mots clés à faible occurence dans le jeu de données. C'est un problème de la méthode, la variabilité est plus forte avec un échantillon moindre.

Il faut donc rectifier la mesure d'entropie, en prenant en compte le fait que le nombre d'occurence rend inégale la mesure.

On identifie sur le graphique une tendance générale : les données se répartissent autour d'une courbe. Si on arrive à estimer cette courbe, on pourra estimer pour chaque mot clé la différence entre l'entropie mesurée, et le point sur la courbe associé au nombre d'occurences du mot clé. La courbe sera un standard. En dessous de la courbe, un point sera d'entropie comparativement faible, et au dessus, comparativement forte.

Pour cela, nous avons utilisée une régression linéaire. La courbe n'étant pas une droite, il a fallut transformer les données des nombres d'occurences pour obtenir une droite affine. En faisant la transformation $X = \log\log X$, la régression linéaire a donné un R^2 de 0.96, ce qui valide la méthode (figure 2)

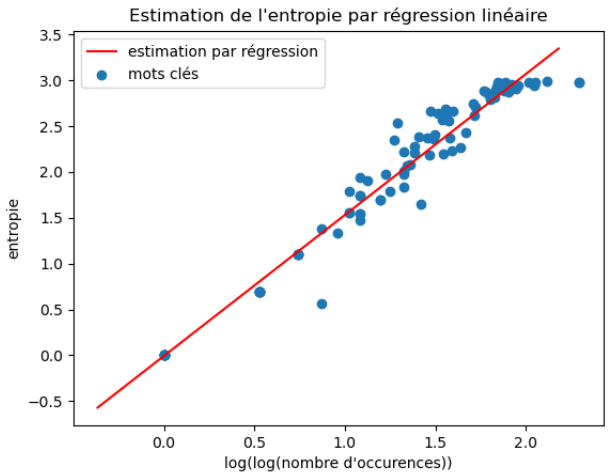

Figure 2: On observe qu'à part sur les bords, la droite convient à peu près à la courbe. On peut donc s'en servir pour estimer si l'entropie d'un mot clé est comparativement grande ou faible.

4.3.4 Résultats

Deux mots clés se démarquent du reste. Ces mots clés correspondent au deux points qui sont le plus écartés de la droite de régression, bien visibles sur la figure 2. Les mots clés sont `shoe_store` et `furniture_store`. En regardant le compte d'occurences par plage horaire, on se rend compte que les occurences de chacun se concentrent fortement sur une seule plage.

Une analyse plus fine (en affichant les coordonnées GPS des points associés à ces mots clés pour la plage horaire principale) montre ces deux mots clés sont concentrés sur des hypercentres commerciaux. On peut en déduire une habitude de consommation : c'est toujours le même moment de la semaine que ces hypercentres sont visités. On a donc déduit une habitude de l'utilisateur.

5 Conclusion

5.1 Conclusion

Nous vivons dans une société qui sait de plus en plus de chose sur chacune de nos actions. Des capteurs et indicateurs sont activés de toute part. Même si ces données ne sont que des signaux parcelaires, sans signification immédiatement interprétable, il est possible pour une personne avertie et connaissant les usages des outils de data mining, de trouver des informations interprétables, utiles et parfois intrusives sur une personne.

Le grand public pense souvent que c'est uniquement grâce à la corrélation de somme immense de métadonnées, récoltées sur des applications et sites variés, que des informations importantes sur les individus peuvent être récoltées. Ici, nous montrons que ce n'est pas le cas.

La simple données des coordonnées de départ et d'arrivée des trajets Uber, nous permet d'accéder à un certain nombre d'informations privées. L'information dont nous disposions était pourtant peu conséquente : la personne étudiée n'utilisait pas Uber pour tous ses déplacements, il est donc impossible de la pister totalement avec les données présentes. Tout au plus nous avions des voyages isolés. Mais malgré celà, il est possible d'identifier des lieux d'habitation probable, des motifs récurrents comme la visite de centres commerciaux, ou encore certains passages près d'hopitaux ou de complexes médicaux.

L'analyse a été permise par les outils de Data Mining, et par leur utilisation et l'interprétation des résultats par une personne réelle. Tout ce que pouvait nous apporter les données n'a sans doute pas été exploité, mais nous estimons que le principal est devenu apparent.

5.2 Discussion

Notre projet n'a pas été facile à mener. La base de données que nous avions choisie était de taille restreinte, avec moins de 700 entrées pour une durée de 5 ans. Le caractère du phénomène représenté, à savoir les courses Uber, fait que les voyages sont naturellement rare. On peut aussi supposer que ces voyages comportent une grande part d'atypique : des voyages non répétés

qui ne permettent pas d'identifier intuitivement des habitudes. On peut le vérifier par l'expérience, en remarquant que les bruits enlevés par les algorithmes de clustering (voir partie Knime) représentent la majorité des trajets.

Bien que nous ayons enrichit les données avec l'API Google Places, l'information obtenue était difficilement exploitable car incertaine. Parmis toutes les adresses renvoyées par les requêtes pour chaque trajet, une seule est vraisemblablement juste. Voir même aucune. Le problème était difficile. Nous avons passé en revue tous les algorithmes suggérés, ceux vu en cours, et d'autres encore, mais aucun ne nous a permis de recouper d'une manière ou d'une autre les résultats incertains des requêtes en certitudes.

Pour améliorer les résultats, il y a plusieurs piste qui pourraient être suivies. La première serait d'obtenir des données sur un grand nombre d'utilisateurs. Celà nous permettrait plus facilement de faire émerger des points d'intérêts et d'identifier les zones significatives. Le problème est qu'il faudrait des données dans les mêmes zones, et on trouve peu de données sur la Russie. Paradoxalement, la personne étudiée utilisait beaucoup Uber (bien que pas suffisement pour établir un vrai emploi du temps), ce qui fait de ce dataset un des plus compréhensif qu'on ait trouvé pour une personne seule.

Une autre piste serait de trouver des données qui permette de donner des priorités aux lieux automatiquement. Si par exemple nous connaissions la fréquentation, la surface ou le chiffre d'affaire des différents lieux, nous pourrions établir des probabilités pour chaque lieu d'être celui visité par l'utilisateur d'Uber. Mais l'API ne nous a pas permis d'obtenir ces informations. Ces données sont sans doutes accessibles en lignes publiquement, mais il est fort probable qu'elles soient en langue russe. Nous n'avons pas réussi à trouver ces données.

6 Appendix

A Entropie

TET_weekday	TET_moment	trip_uid
0	Aprèm	2
	Soir	1
1	Aprèm	1
	Matin	1
	Soir	1
2	Aprèm	1
	Matin	1
	Soir	1
3	Aprèm	16
	Matin	2
	Soir	2
4	Aprèm	1
	Matin	1
	Soir	4
5	Aprèm	1
6	Matin	4

TET_weekday	TET_moment	trip_uid
0	Aprèm	2
	Matin	2
	Soir	2
1	Aprèm	3
	Matin	2
	Soir	1
2	Aprèm	5
	Matin	1
4	Aprèm	3
	Matin	2
	Soir	1
5	Aprèm	3
6	Aprèm	17
	Soir	6

Figure 1 : Nombre d'occurences par plage horaires du mot clé remarqué lors de l'analyse par entropy, shoe_store/furniture_store. On observe un horaire particulier.

B Knime

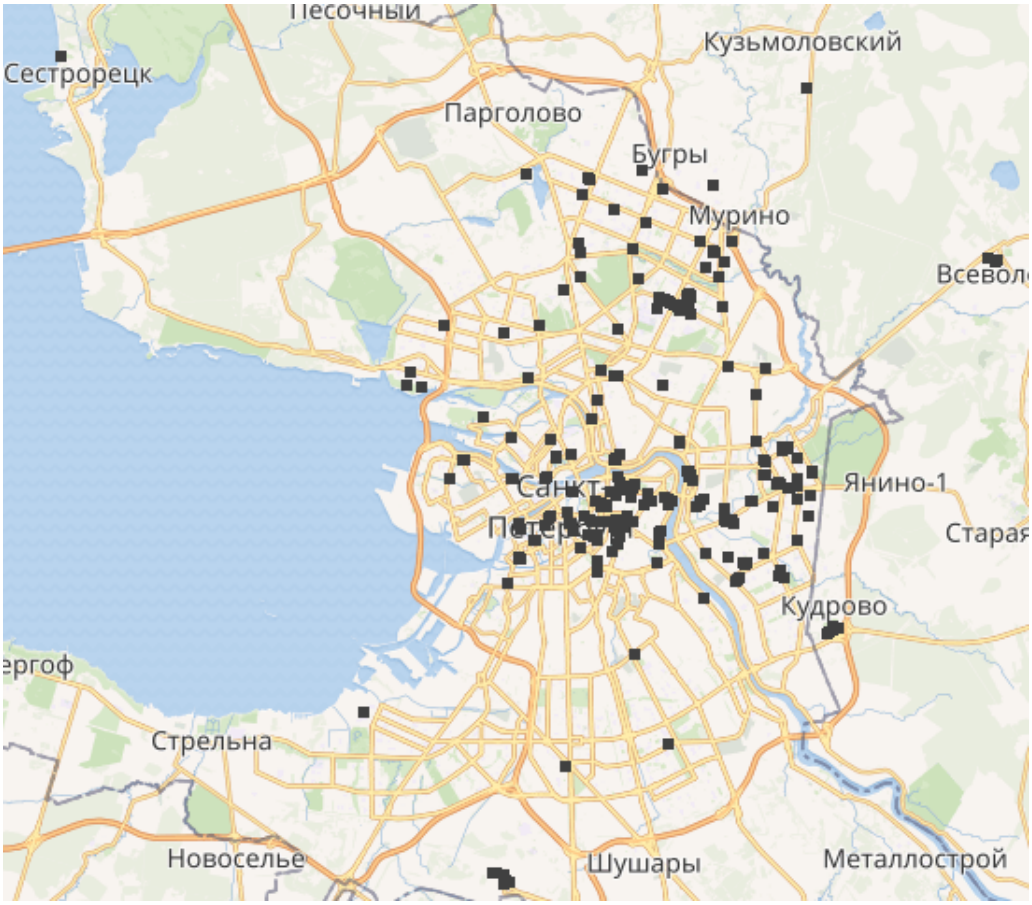

Figure 3: Plots bruts, sans filtrage

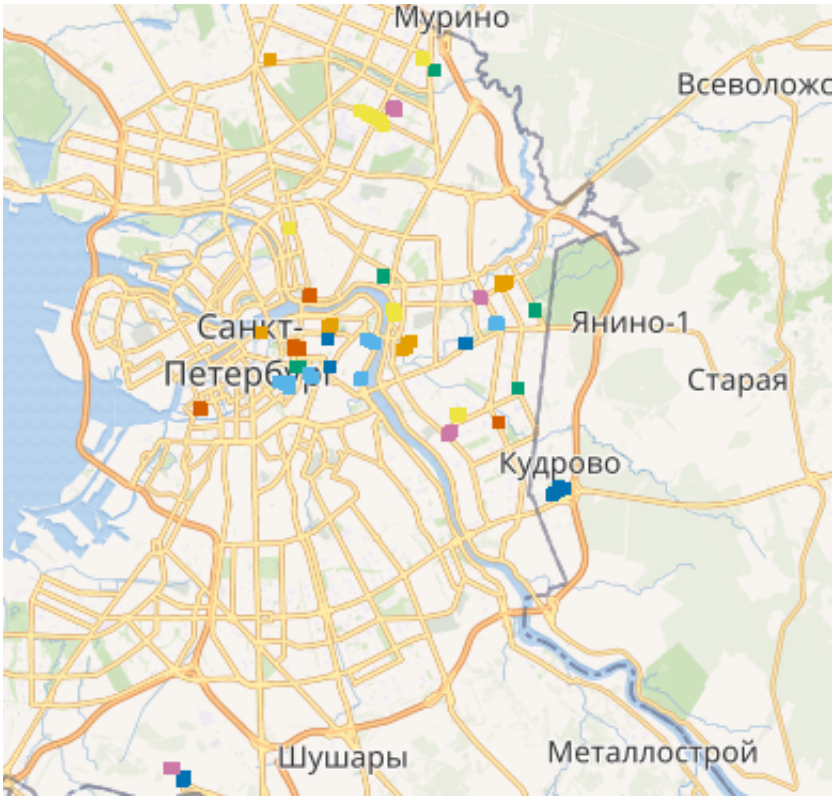

Figure 4: Plots après clustering selon les coordonnées spatiales, pour DBSCAN et un epsilon correspondant à une vingtaine de mètres.

Figure 5: Après clustering selon l'heure et le lieu, on observe cette zone de forte affluence à toute heures (beaucoup de couleurs). Cette zone, en banlieu résidencielle, est sans doute très proche du lieu d'habitation de la personne étudiée.

C Sorties des différents algorithmes de clustering et isolation d'une plage horaire tardive

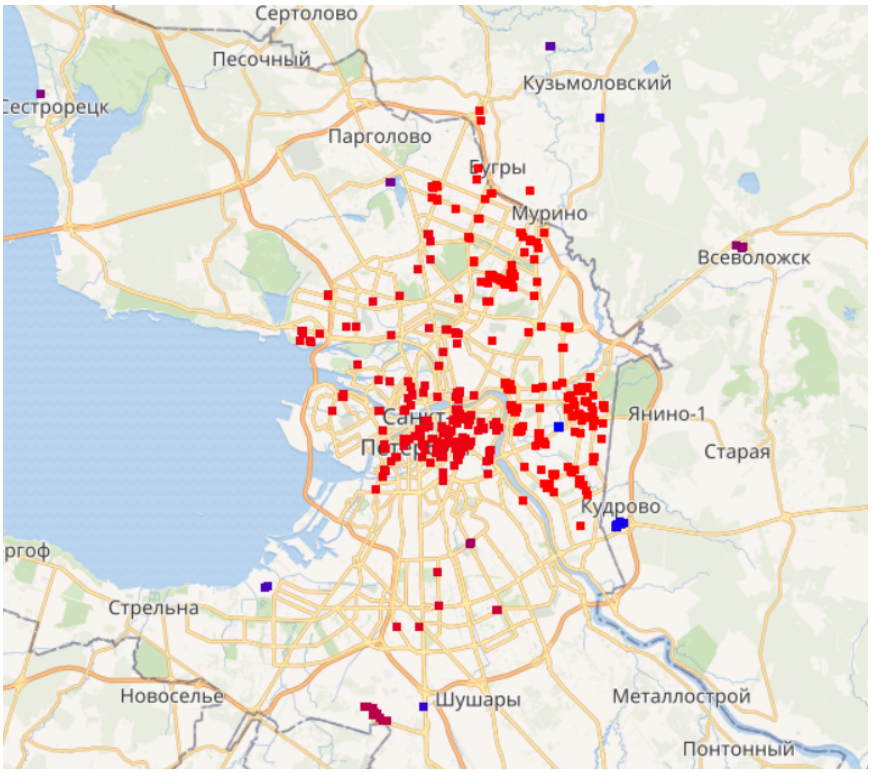

Figure 6: résultat obtenu par l'algorithme de clustering hiérarchique (15 clusters avec condition de fusion de deux clusters égale à ward pour équilibrer la taille des clusters.

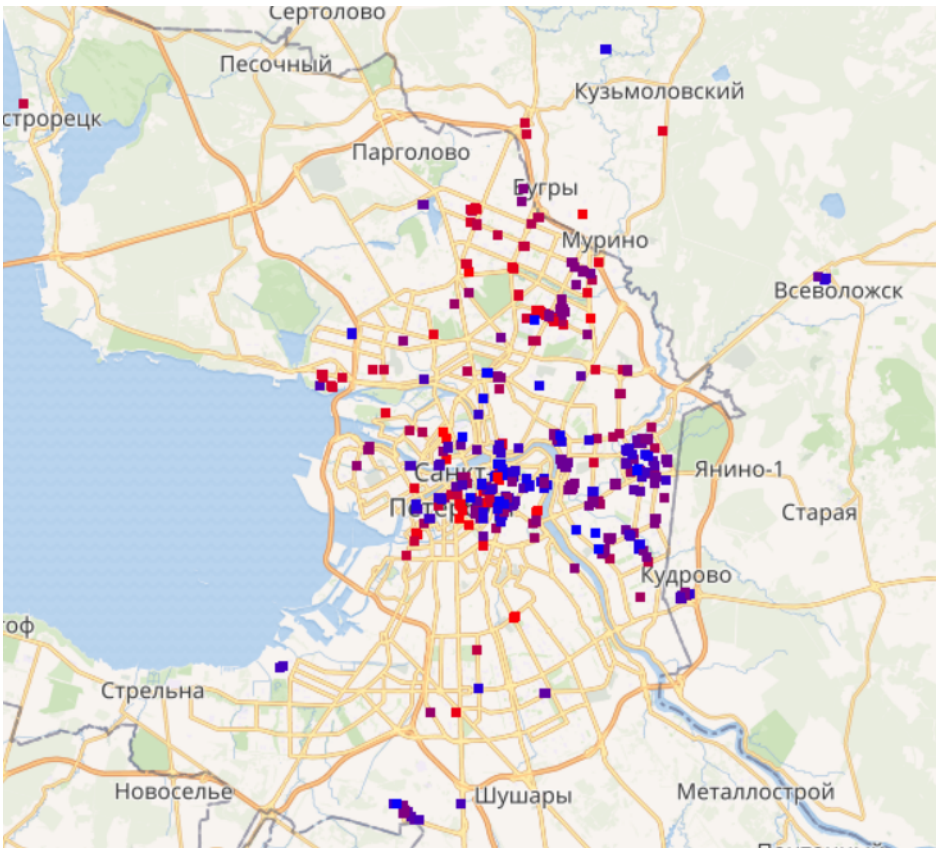

Figure 7: résultat obtenu par l'algorithme de propagation d'Affinité, les clusters se superposent à cause de la proximité des valeurs pour chaque point géographique.

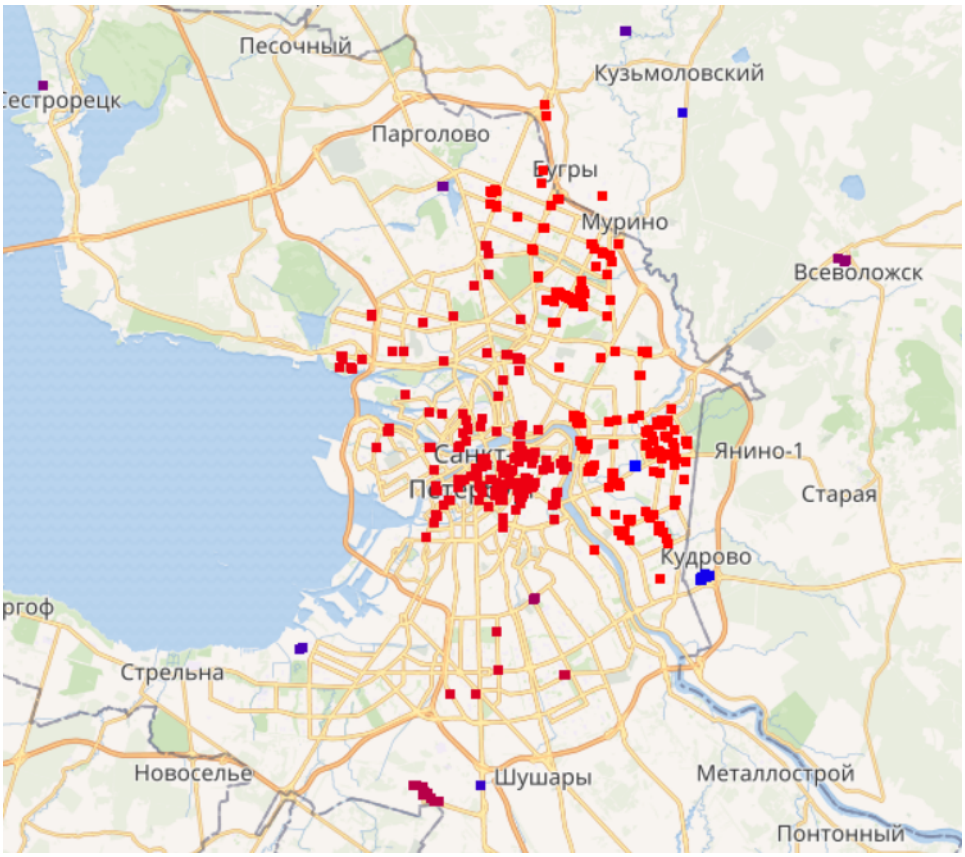

Figure 8: Clustering avec l'algorithme MeanShift et un périmètre de voisinage (bandwidth) fixé à 2. Ici il y a une disparité dans la taille des clusters similaire à celle trouvée avec le clustering hiérarchique.

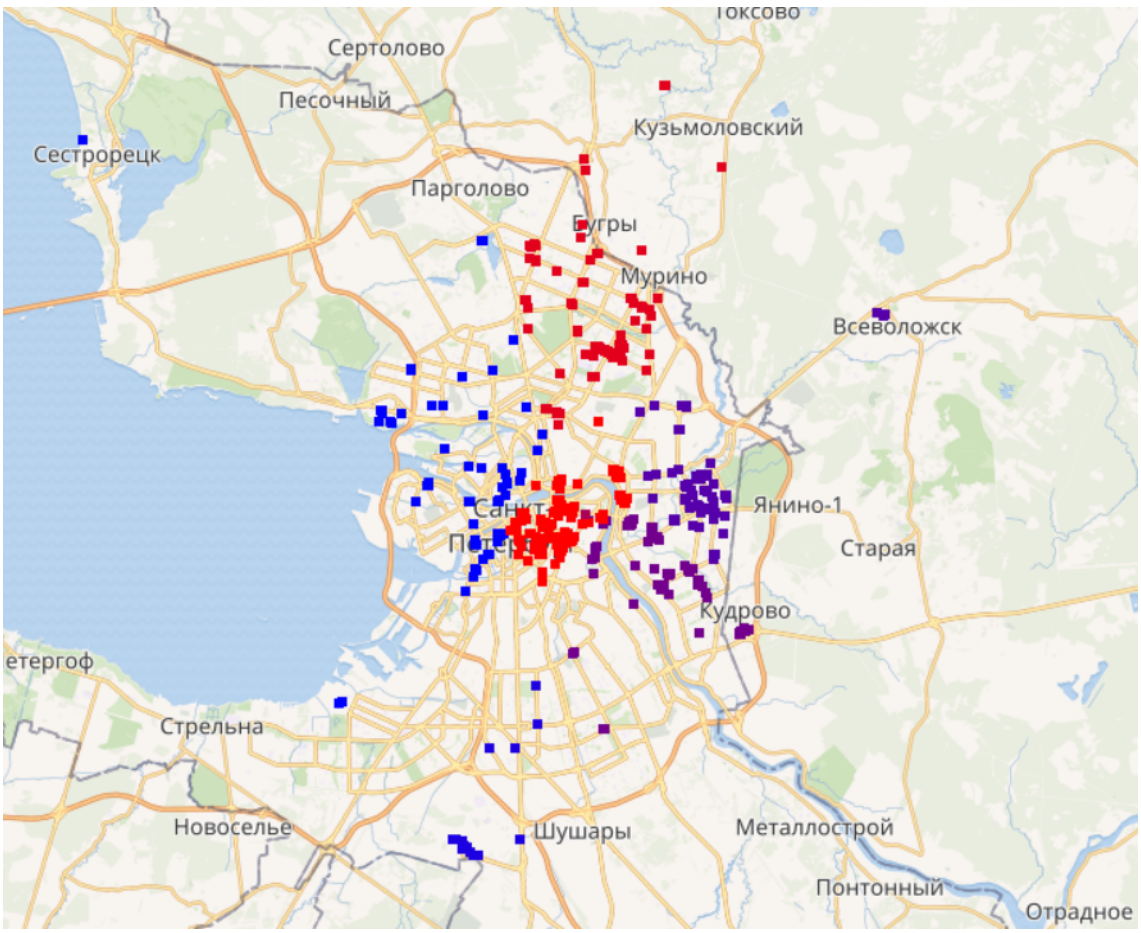

Figure 9: Spectral Clustering: on voit clairement les zones où le client Uber va fréquemment et ces zones sont dignes d'intérêt pour des traitements ultérieurs.

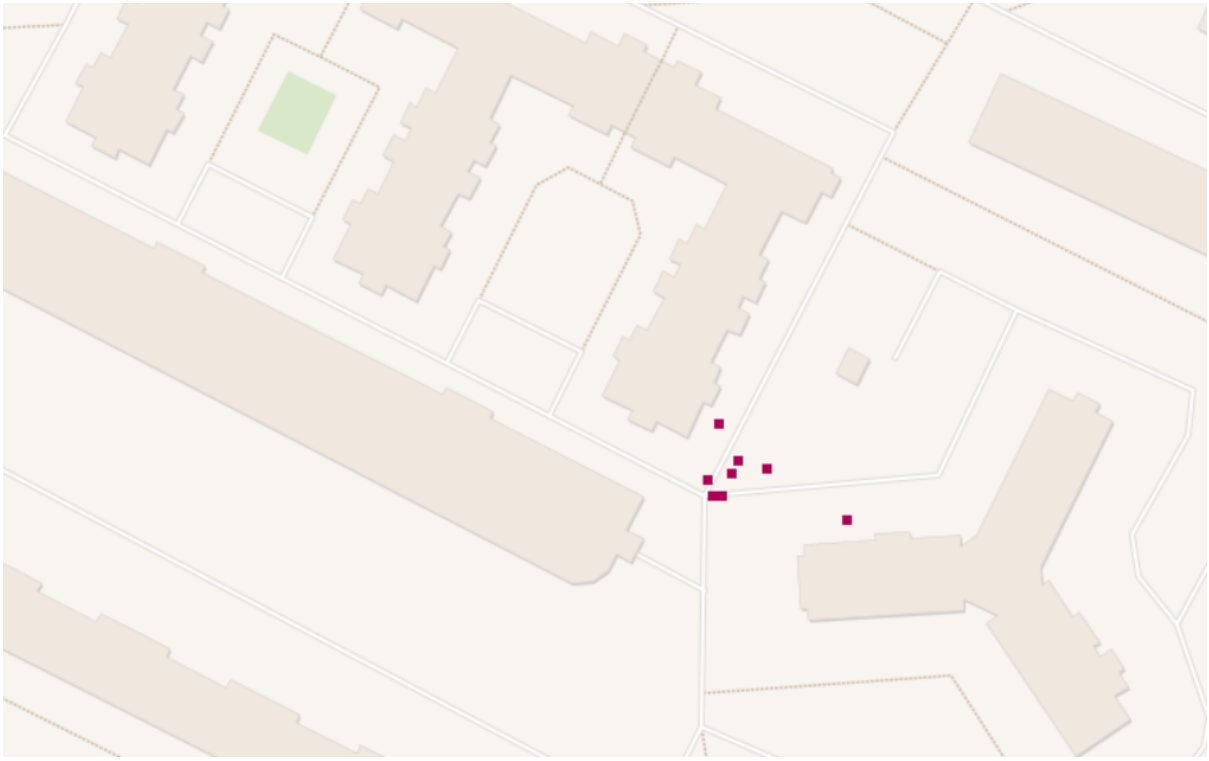

Figure 10: Ici on a isolé les plages horaires entre 20 h et 6h du matin, on remarque le même groupe compact de points que sur KNIME. Cela porte aux mêmes conclusions qu'en Annexe B.