

Projet fouille de données

Compte Rendu



Groupe 4:

HUA Yang

SHU Yuting

Introduction	3
Présentation du projet	3
Etude et préparation des données	3
Composition de la base de donnée	3
Nettoyage des données	4
Retrait des doublons	4
Retrait des vieilles photos	4
Retrait des photos en dehors de Grand Lyon	5
Retrait des valeurs manquantes et étrangers	7
Chaîne de traitement des données	7
OSM Map Viewer	8
Clustering : découverte des points d'intérêt	8
K-Means	8
Interprétation du K-means	8
Première découverte de l'algorithme	9
Variation du nombre de cluster	9
Stabilité du K-Means	11
Clustering hiéarchique	11
Interprétation du clustering hiearchique	11
Concentration sur le parc de la tête d'or	12
DBSCAN	14
Interprétation du DBSCAN	14
Variation du paramères de clustering DBscan	14
Visualisation des points d'intérêt	16
Interprétation des points d'intérêt	18
Description des points d'intérêt grâce à la fouille de motifs	18
Motif entre tags et tags	19
Motif entre les tags et clusters	21
Recherche d'événements : zone dense dans le temps et/ou dans l'espace	23
Clustering sur les données temporelles	23
Workflow	24
Résultats	24
Clustering sur les données spatiales et temporelles	25
Conclusion	27

Introduction

Présentation du projet

Le grand Lyon est une métropole urbaine centrée autour de la ville de Lyon. Dans ce TP là, nous allons utiliser plusieurs techniques dans le domaine du big data afin de trouver des points d'intérêts lyonnais et améliorer ses transports en communs.

Pour cela nous disposons d'une base de donnée flickr de photos prises dans la région lyonnaise. Afin de dégager des points d'intérêt, nous allons filtrer les données inutiles, enrichir les données manquantes, se regrouper les données similaires et finalement interpréter ces données.

Etude et préparation des données

Composition de la base de donnée

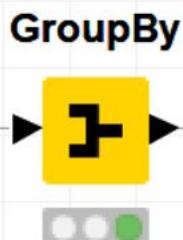
Attribut	Description	Utilité
id	identifiant	filtrer les doublons
user	utilisateur	
lat	latitude	coordonnée de lieu prise en photo
long	longitude	
tags	commentaire (avec valeur null)	fouille de motif
title	commentaire (avec valeur null)	
date_taken_minute	minute prise en photo	recherche d'événement en fonction du temps
date_taken_hour	heure prise en photo	
date_taken_day	date prise en photo	
date_taken_month	mois prise en photo	
date_taken_year	année prise en photo	
date_upload_minute	minute de mise en ligne	
date_upload_hour	heure de mise en ligne	

date_upload_day	date de mise en ligne	
date_upload_month	mois de mise en ligne	
date_upload_year	annee de mise en ligne	

Nettoyage des données

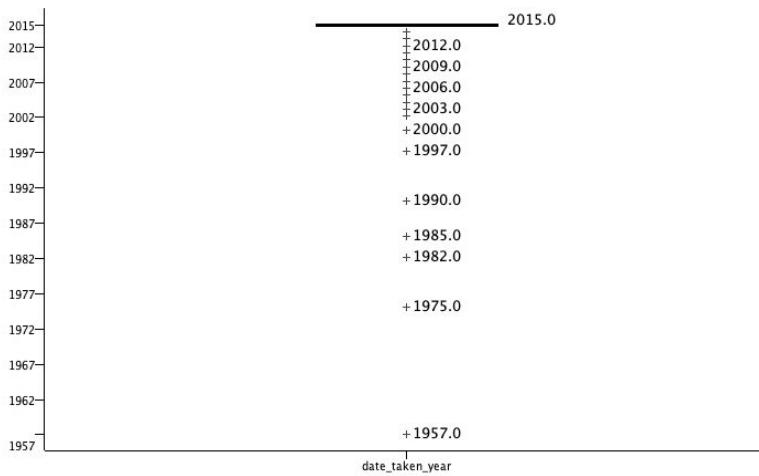
Retrait des doublons

Pour supprimer des doublons, en passant par noeuds GroupBy sur l'attribut ID, des données passe de 83851 lignes à 15188 lignes

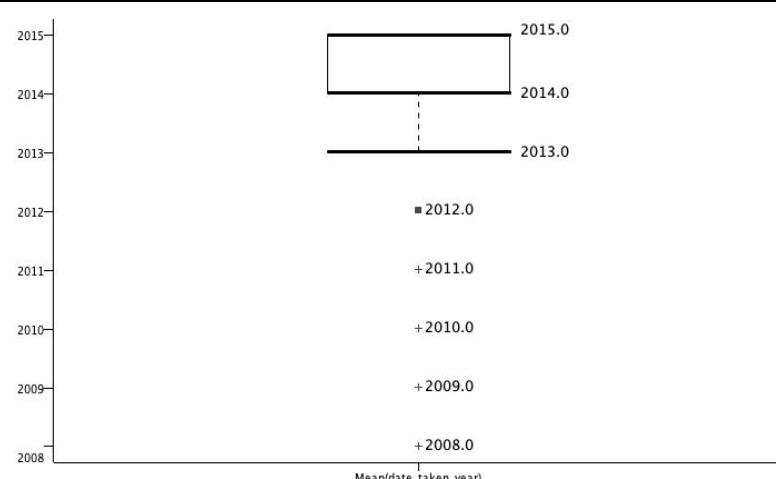


Paramètre :
 Group column : id
 Manual Aggregation : tous les colonnes restant

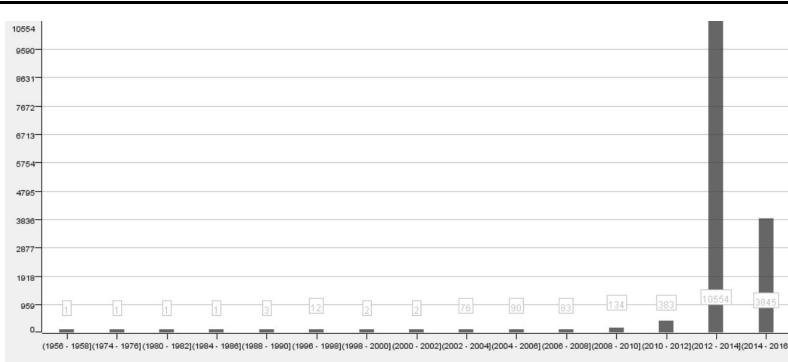
Retrait des vieilles photos



Nous remarquons que certaines photos ont été prises en 20eme siècle. Il faudra les retirer de notre étude, puisque nous voulons chercher des zones à forte densité de nos jours. En plus, le nombre de vieilles photos reste très peu donc ils ne sont pas représentatives.



A partir de Box Plot, nous observons que des photos prises avant 2012 (2012 inclus) sont gardées comme des valeurs aberrantes.



A l'aide de l'histogramme, nous observons aussi qu'il y a peu de photos prises avant 2012 (2012 inclus).

En gardant les traitements statistiques, nous choisissons retraitier des photos avant l'année 2012(2012 inclus).

Row Filter



after 2013

Paramètre :

Include rows by attribute value

Column to test: date_taken_year

Matching criteria:

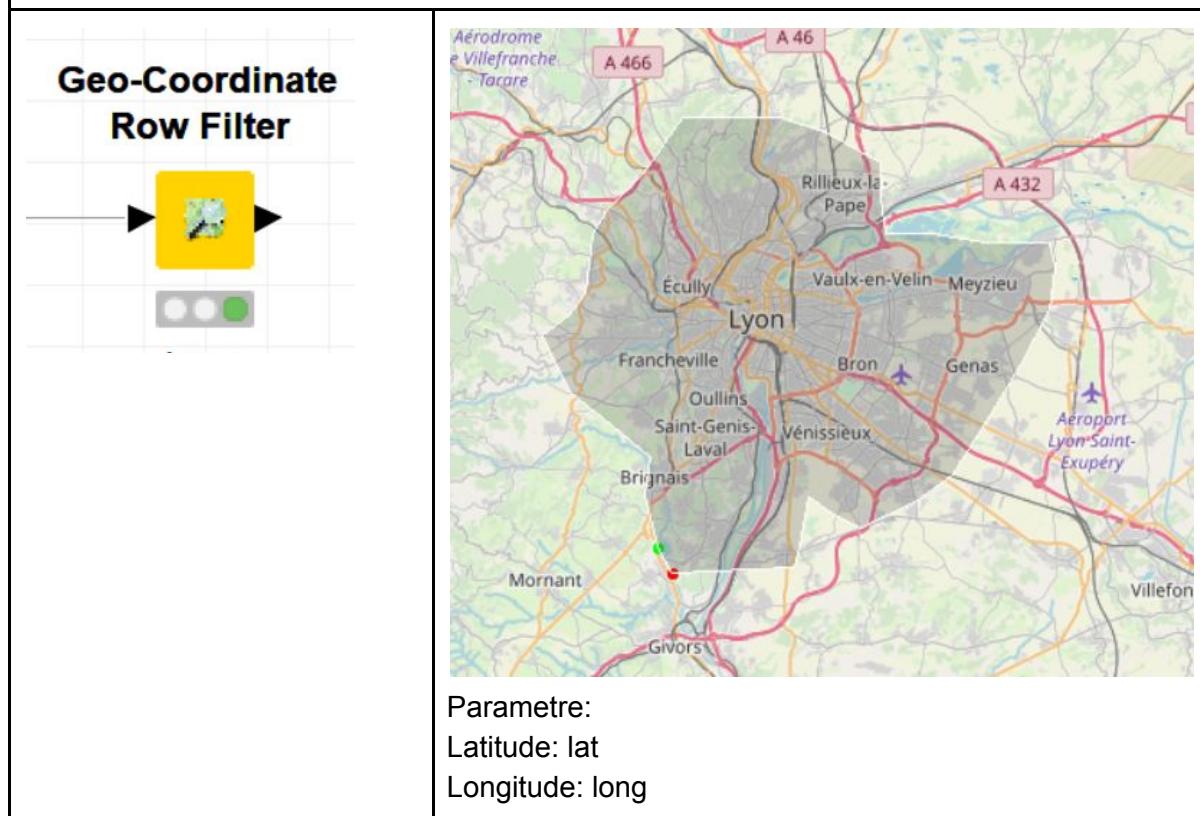
use range checking:

lower bound :2013

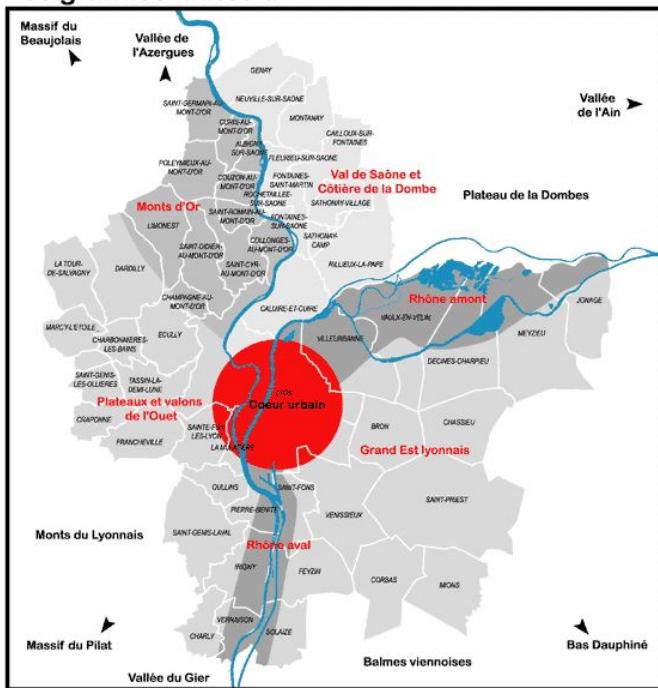
Retrait des photos en dehors de Grand Lyon

Nous ne analyserons que des données de Grand Lyon, alors filtrer des données en dehors de Grand Lyon est nécessaire. A l'aide de la carte de Grand Lyon, on dessine la

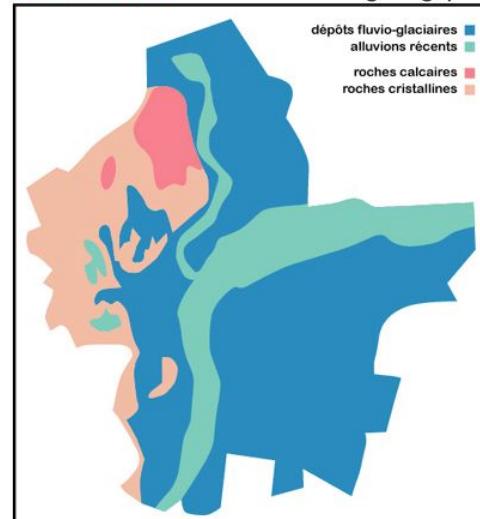
zone traitée comme le dessin ci-dessous en utilisant le noeud <<Geo-Coordinate Row Filter>>.



Les grandes unités du GRANDLYON

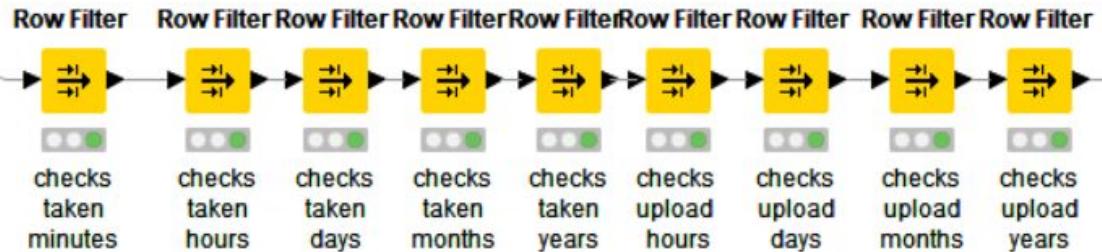


GRANDLYON carte géologique



Retrait des valeurs manquantes et étrangers

Nous filtrons toute les données manquants le temps, y compris le temps où on a pris les photos et les mise en ligne comme des informations incomplètes, ces données seront inutile pour nous.



Enlever les données qui sont prises après les mis en ligne, puisque ces données sont étranges.

Rule-based Row Filter

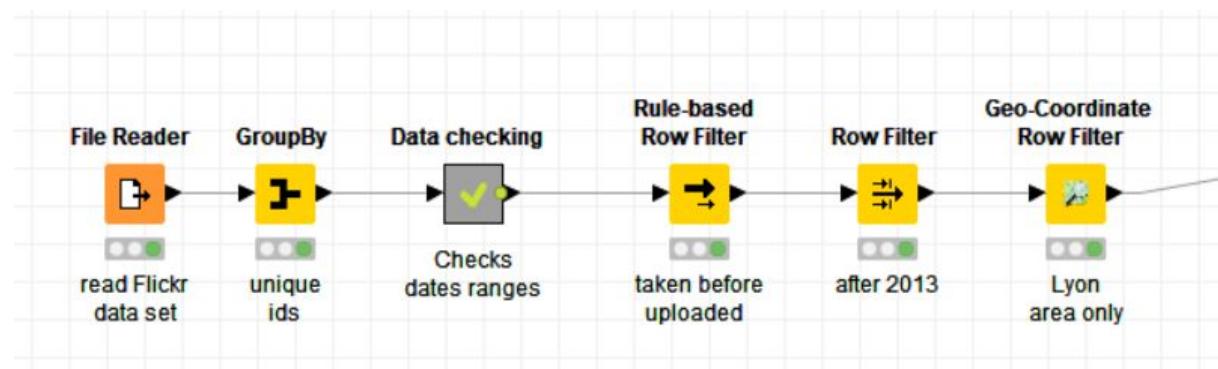


Expression:

`$date_upload_year$ >= $date_taken_year$ => TRUE`

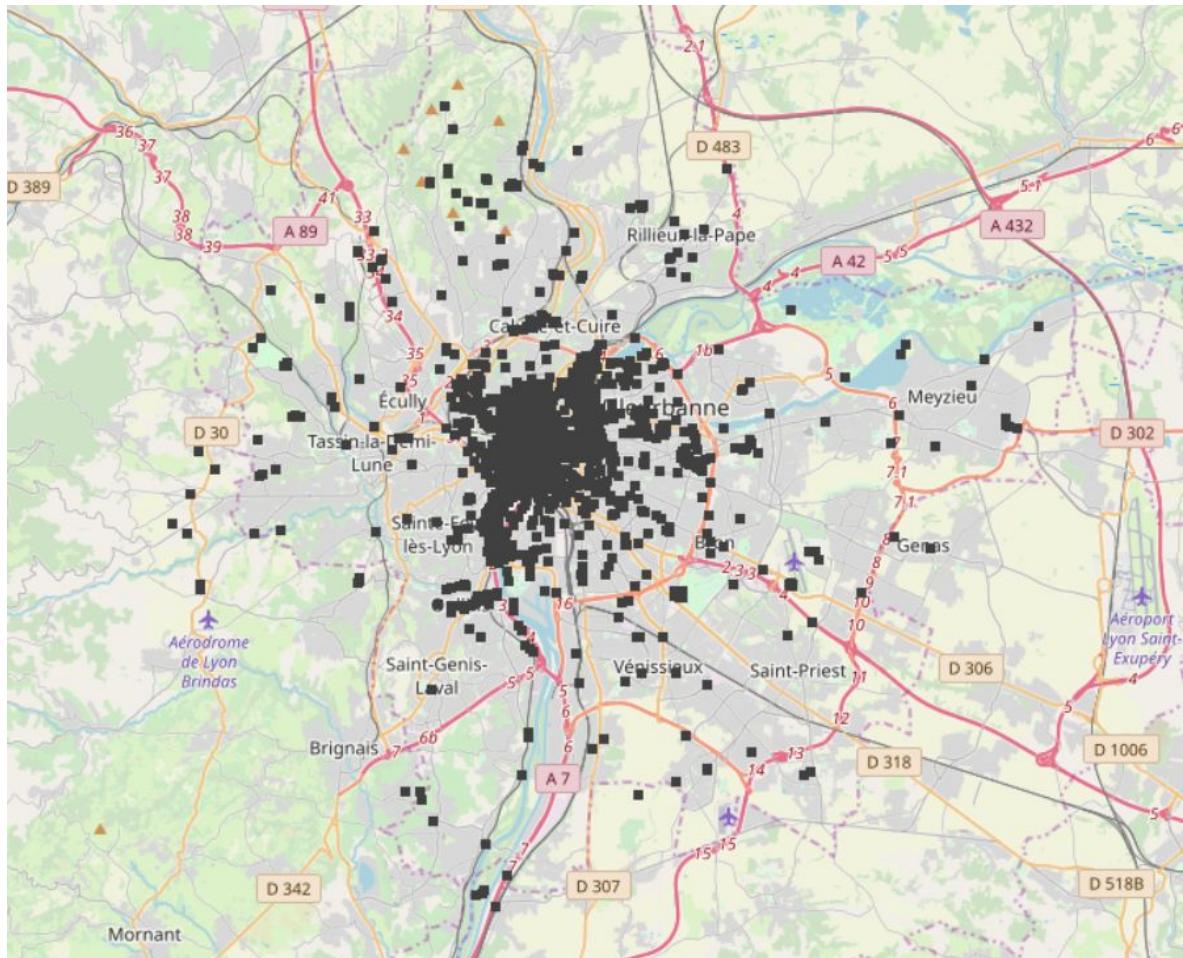
Include TRUE matches

Chaîne de traitement des données



OSM Map Viewer

Après avoir nettoyé des données, nous restons 11340 lignes qui est bien éliminé par rapport à 83851 lignes au début. Nous pouvons voir la répartition des données à l'aide de OSM Map Viewer. Nous observons que la densité de point est beaucoup plus forte à la centre ville par rapport à la banlieue.



Clustering : découverte des points d'intérêt

K-Means

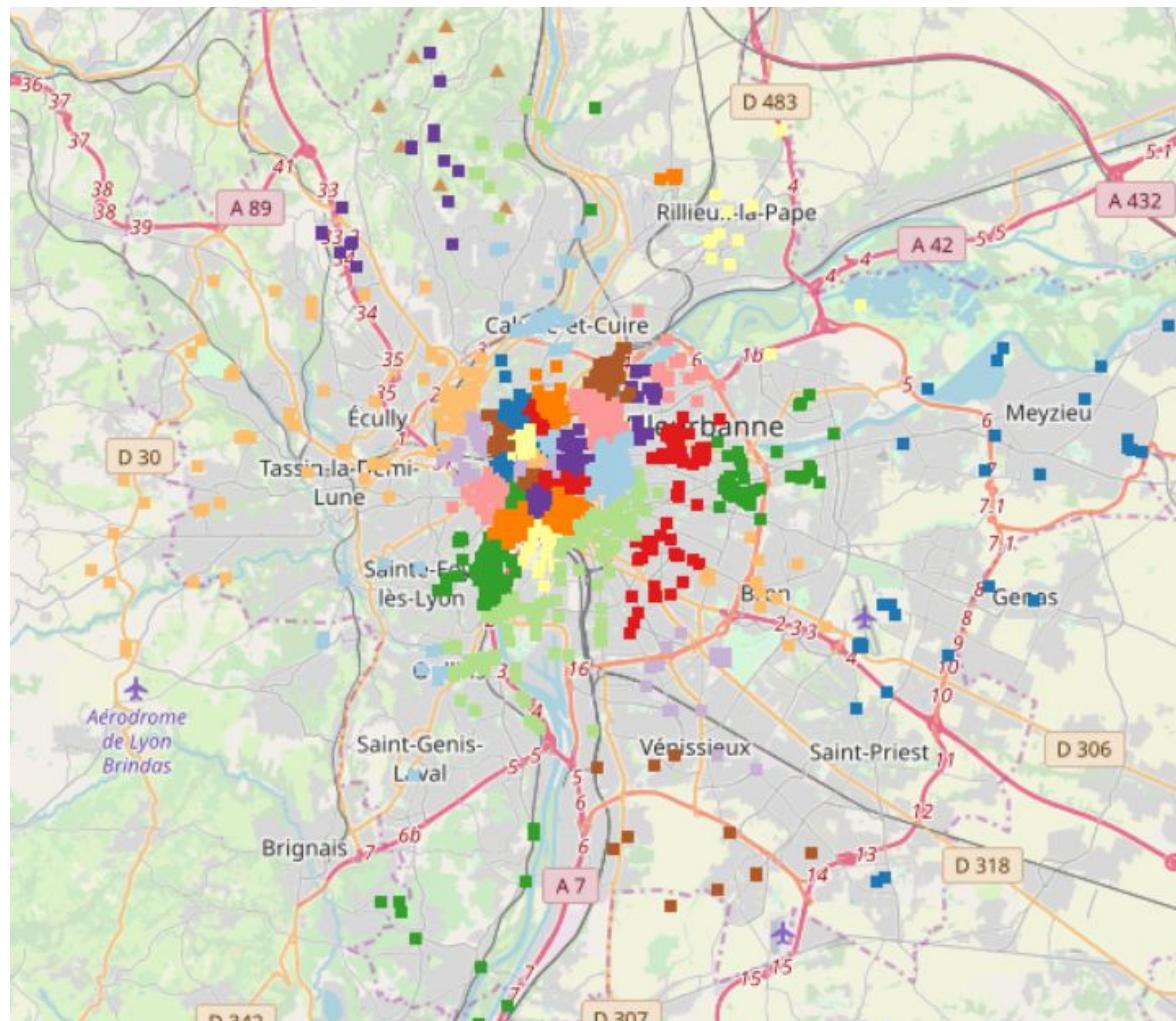
Interprétation du K-means

L'algorithme K-means est basé sur le prototype. Des points similaires sont définis et inclus dans un cluster s'ils sont plus similaires avec le prototype d'un cluster et pas celui de l'autre. Dans le domaine de travail sur les plans, intuitivement, le prototype d'un cluster s'agit du

centroïde d'une zone. A cause de celui-ci, les clusters calculés par K-means sont globulaires. De plus, l'algorithme K-means cause des problèmes quant aux clusters de tailles différentes ou de densité différente et quant aux données contenant des valeurs aberrantes.

Première découverte de l'algorithme

En regardant le résultat, nous trouvons des clusters très étendus en périphérie et des clusters très concentrés dans le centre. Nous devrons concentrer plutôt sur la centre ville pour découverte des points d'intérêt à la suite des études.

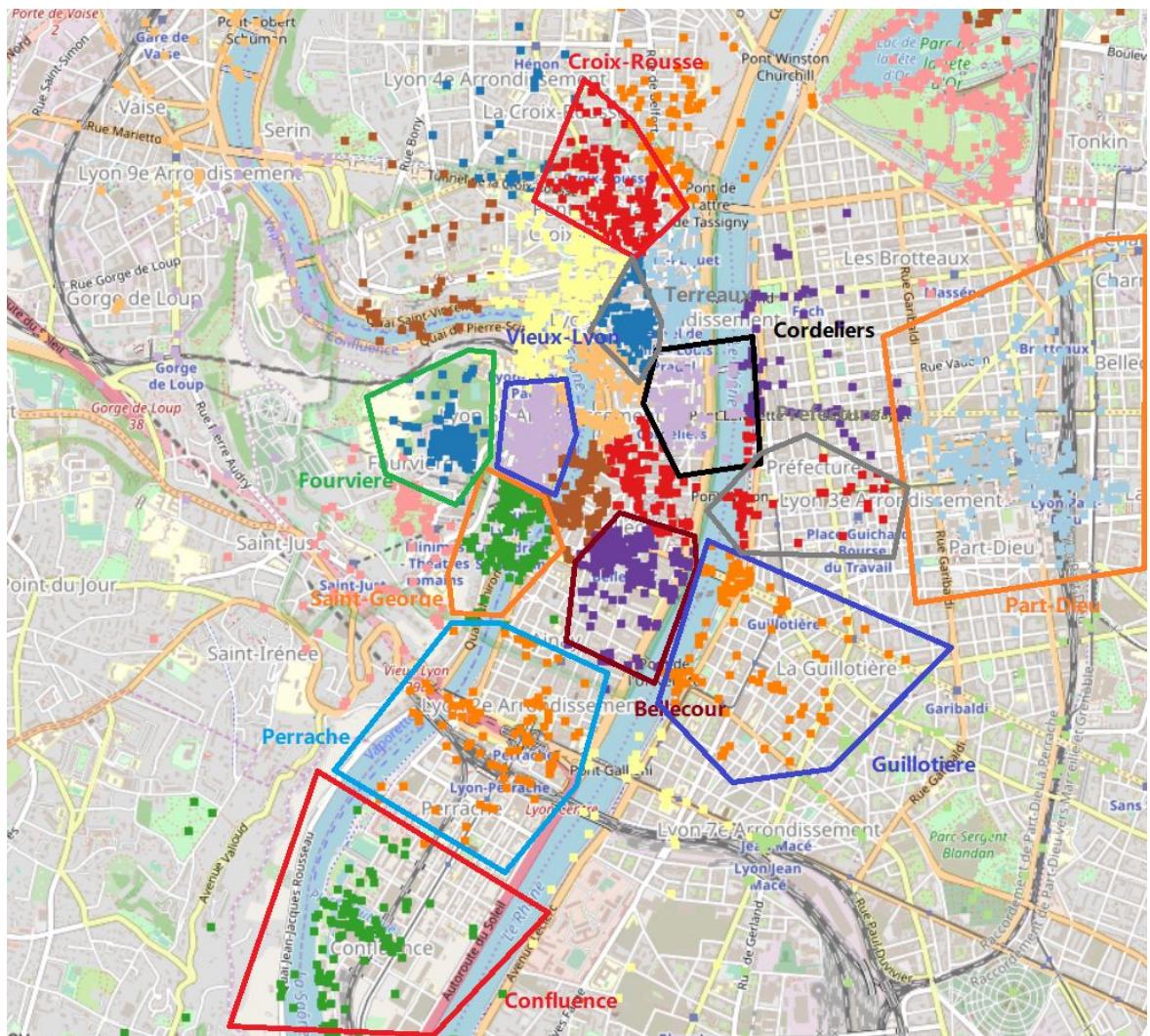


Variation du nombre de cluster

50 clusters

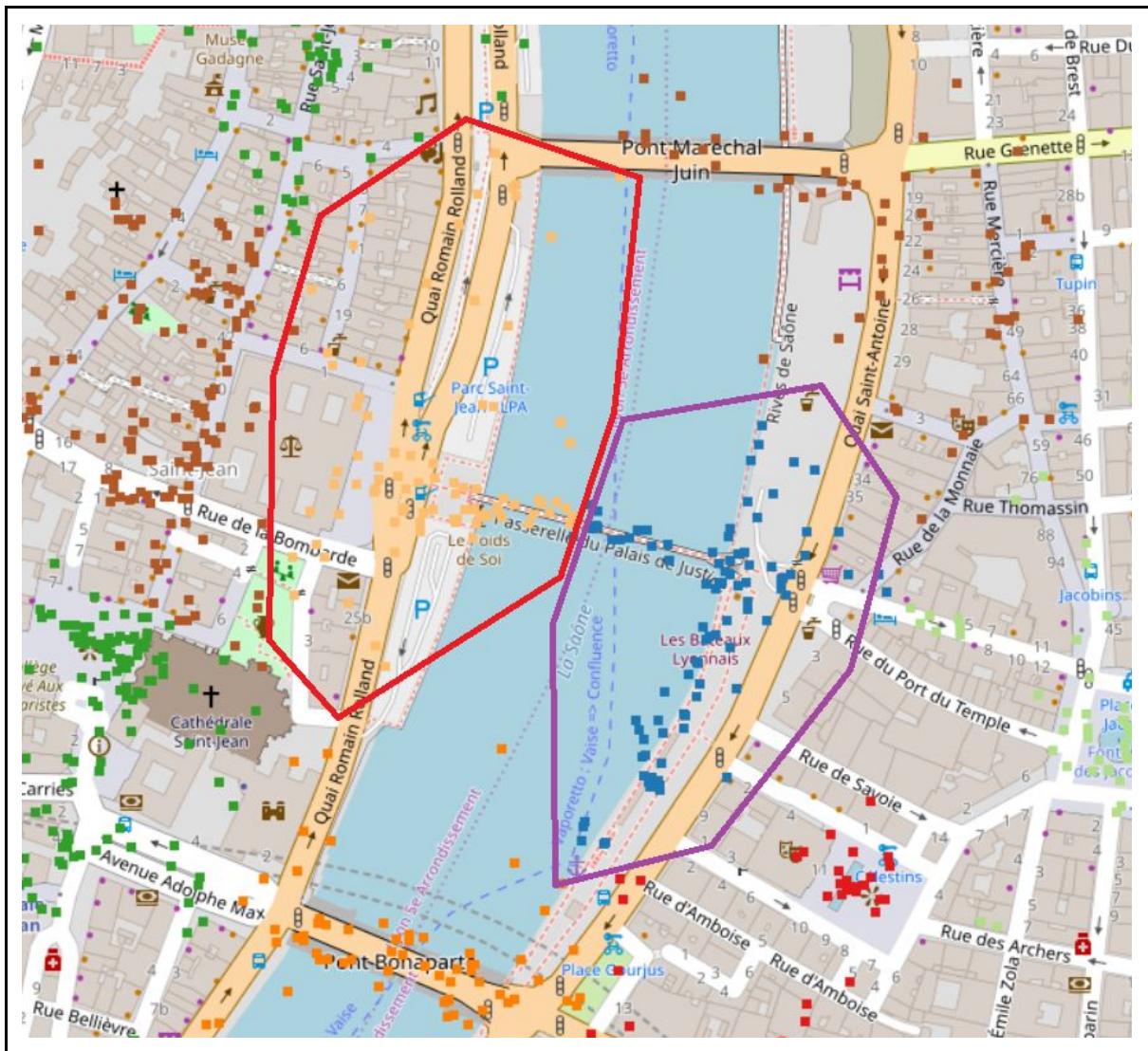
Les 50 clusters nous donnent les grands quartiers mais nous souhaitons obtenir un niveau de précision plus important pour détecter des points d'intérêts plus spécifiques comme des monuments.

Alors, il faut augmenter le nombre de clusters.



100 clusters

L'algorithme K-means groupe des données aux clusters globulaires. Certes, beaucoup de photos sont pris autour d'attractions touristiques, dans ce cas là ce sera logique si les clusters soient globulaires mais c'est pas le cas pour un événement spécial comme la fête de la lumière ou une célébration de la fête. Il pose un problème des frontières à la zone de forte densité. Ici, nous observons qu'il groupe deux clusters globulaires en coupant des données dans un même pont. Il ne correspond pas le résultat attendu. L'algorithme K-means est imparfait dans le cas de notre étude. Alors nous devons changer l'algorithme à la suite des études.



Stabilité du K-Means

Ici, nous ne pouvons pas utiliser l'entropie pour étudier la stabilité car nous n'avons pas de colonne référence comme IrisDataset.

Par contre, il n'y a pas de besoin pour traiter la stabilité du K-means car ce n'est pas le bon algorithme qui corresponde notre étude.

Clustering hiéarchique

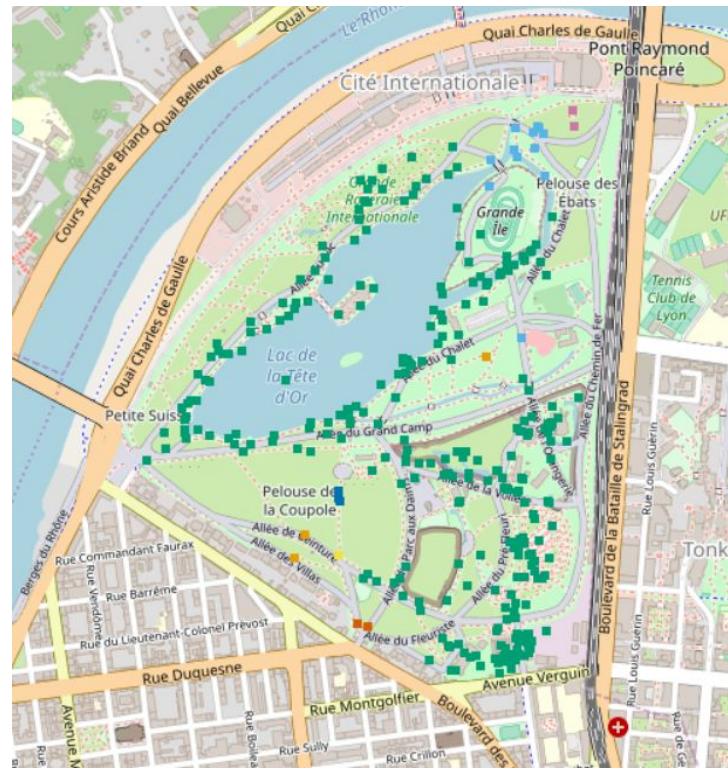
Interprétation du clustering hiéarchique

Au début de l'algorithme de clustering hiérarchique, chaque point est en tant que cluster singleton, puis en fusionnant de manière répétée les deux clusters les plus proches jusqu'à ce qu'il ne reste plus qu'un seul cluster. Et le point clé est comment définir la similarité entre deux clusters, et donc nous parlerons de la mise en pratique de la technique MIN (single link), MAX (complete link), et Group Average à la suite.

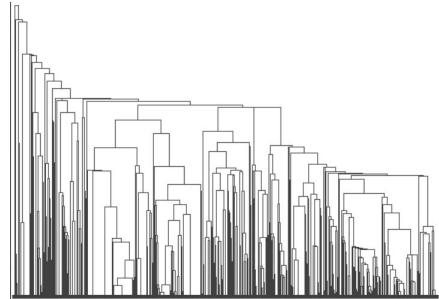
Cet algorithme produit des clusters de mieux qualité par rapport au K-means, par contre la complexité du clustering hiérarchique limite son usage sur notre jeu de données. Alors il faut que nous l'utilisons sur une zone géographique restreinte. Nous prenons l'exemple du parc de la tête d'or.

Concentration sur le parc de la tête d'or

Ici, nous avons 540 photos, et le temps de calcul est en seconde. Nous choisissons de regarder les résultats sur 10 clusters et la distance euclidienne comme la fonction de distance.

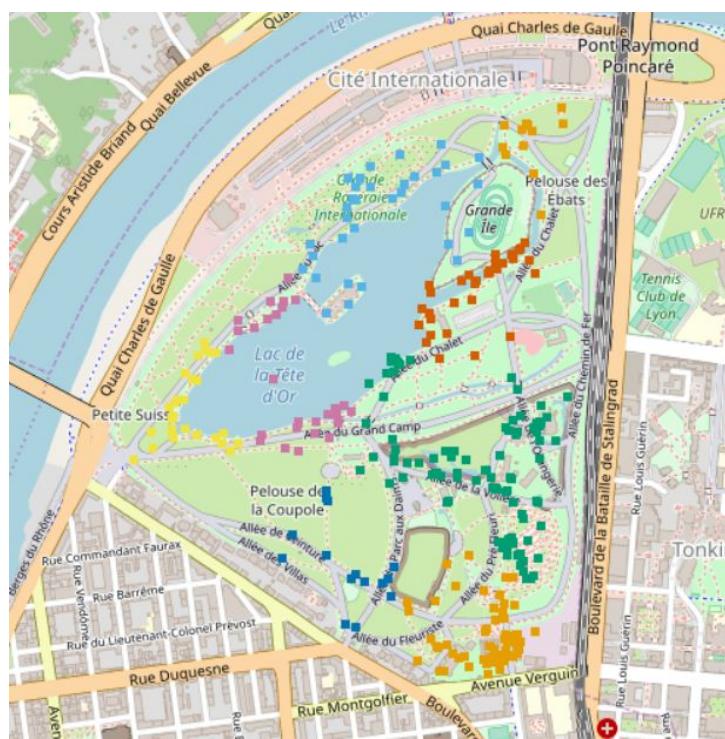


En mode SINGLE

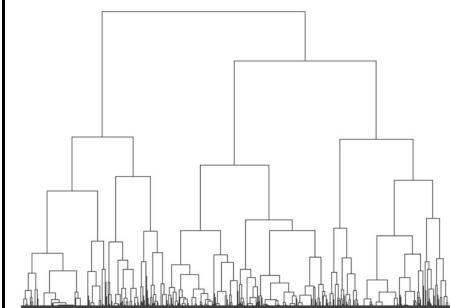


Il peut traiter des clusters non-globulaires, mais il est sensible aux bruits et aux valeurs aberrantes.

Il est totalement mis en échec : on obtient un seul gros cluster et des clusters contenant seulement certains points.

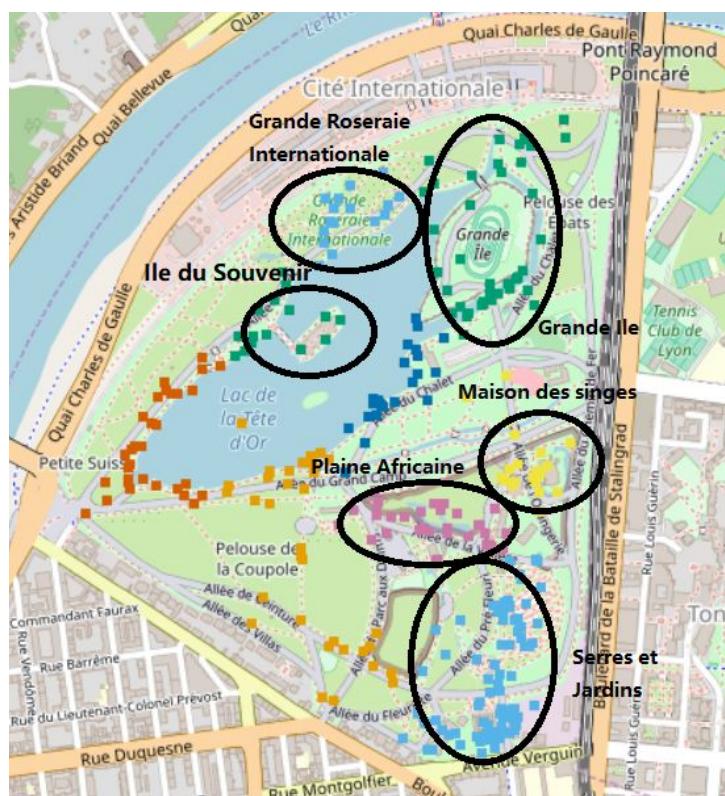


En mode COMPLETE

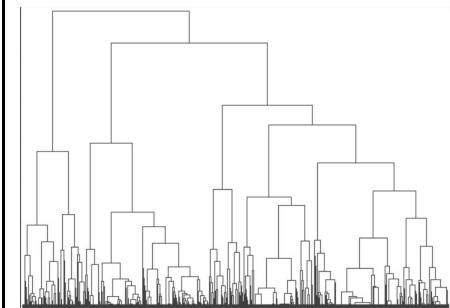


Il est moins sensible aux bruits et aux valeurs aberrantes, mais il a tendance de couper des grands clusters et grouper des clusters globulaires.

Nous observons une séparation en clusters plutôt pertinent.



En mode AVERAGE



Il est la compromis entre la mode SINGLE et COMPLETE. Il est donc moins sensible aux bruits et aux valeurs aberrantes, mais il a tendance de grouper des clusters globulaires.

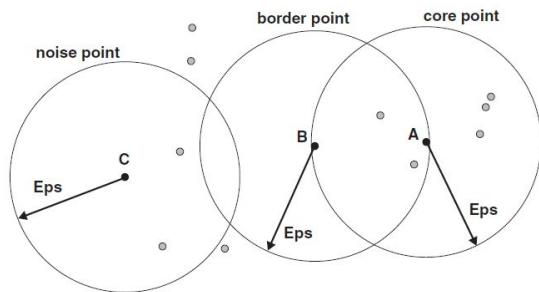
Nous obtenons un résultat un peu similaire au COMPLETE qui est assez satisfaisant pour le clustering, et distinguons bien les différentes zones du parc.

DBSCAN

Interprétation du DBSCAN

L'algorithme DBscan est un clustering algorithme basé sur la densité. Au contraire du K-means, le nombre de clusters sera calculé automatique par l'algorithme. Les points dans la zone de la densité au niveau très bas seront éliminés comme des bruits.

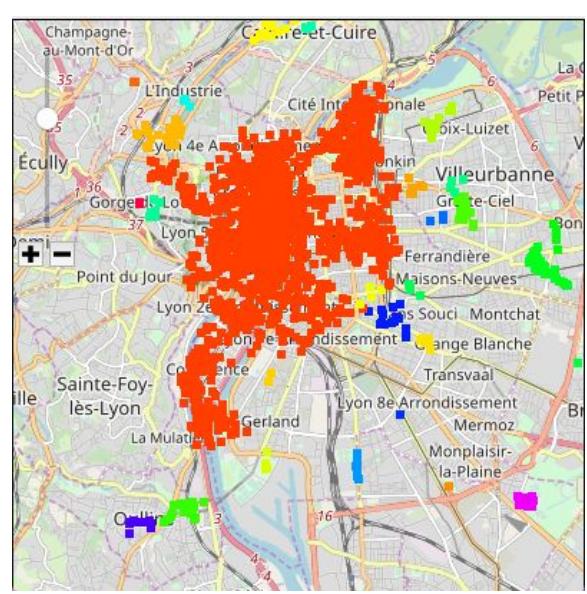
Il n'y a pas autant de définitions différents sur la terme "densité". Or dans notre cas d'étude, la densité sera définie sur l'approche basé sur la centre. Dans ce cas là, la densité sera estimé par un point particulier dans les données en comptant tous les points qui sont assez proches. Il introduit deux paramètres importants: **eps**, le rayon de la zone où on cherche les points proches, et **minPoints**, le nombre minimum des points dans une zone de rayon eps.



Cet algorithme peut bien éliminer des bruit et traiter des clusters avec des formes ou des tailles différentes. Par contre, il n'est pas forte en groupant des zones de densité variante et des données multidimensionnelles.

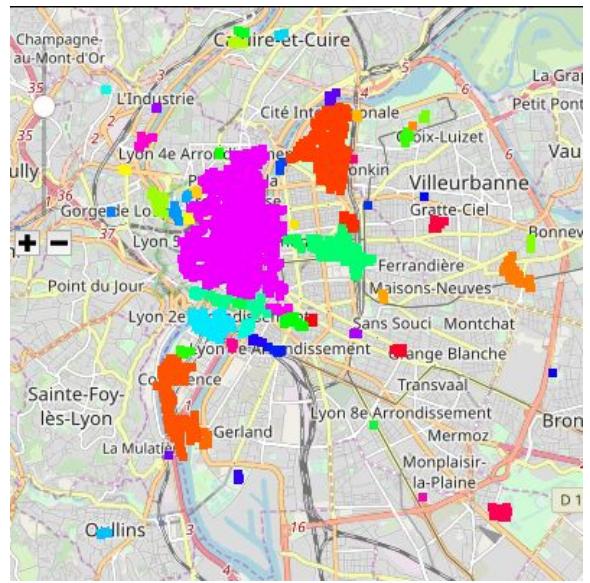
Variation du paramètres de clustering DBscan

Nous allons évoluer les deux paramètres eps et minPoints pour chercher des valeurs plus pertinent à notre étude.



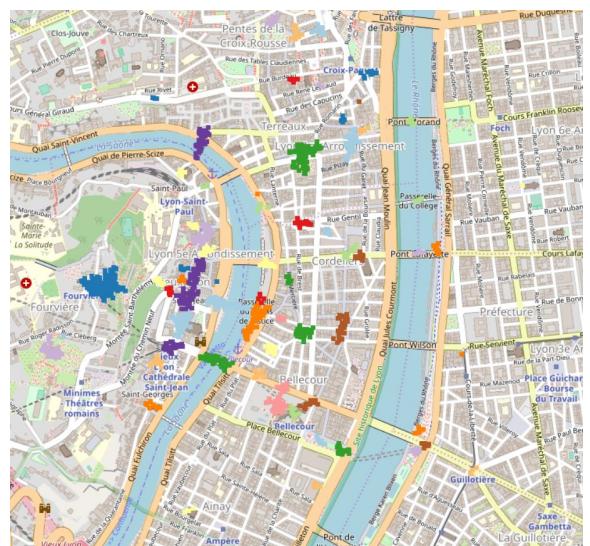
eps = 0.01
minPoints = 10

Nous voyons très bien que les points en centre villes sont plus dense que les points en banlieu. Nous sommes obligé d'analyser avec un eps plus petite pour analyser des zones en centre ville



eps = 0.005
minPoints = 10

La zone en centre ville est mieux découpé mais celle de 1er et 4ème arrondissements restent très flous. c'est-à-dire que la eps n'est pas encore assez petite pour l'analyser.



eps = 0.0008
minPoints = 10

Dans ce cas, de nombreux clusters en banlieu ont disparu et la répartition des clusters est assez discrète. Nous pouvons alors identifier des points d'intérêt à l'aide de ce résultat.

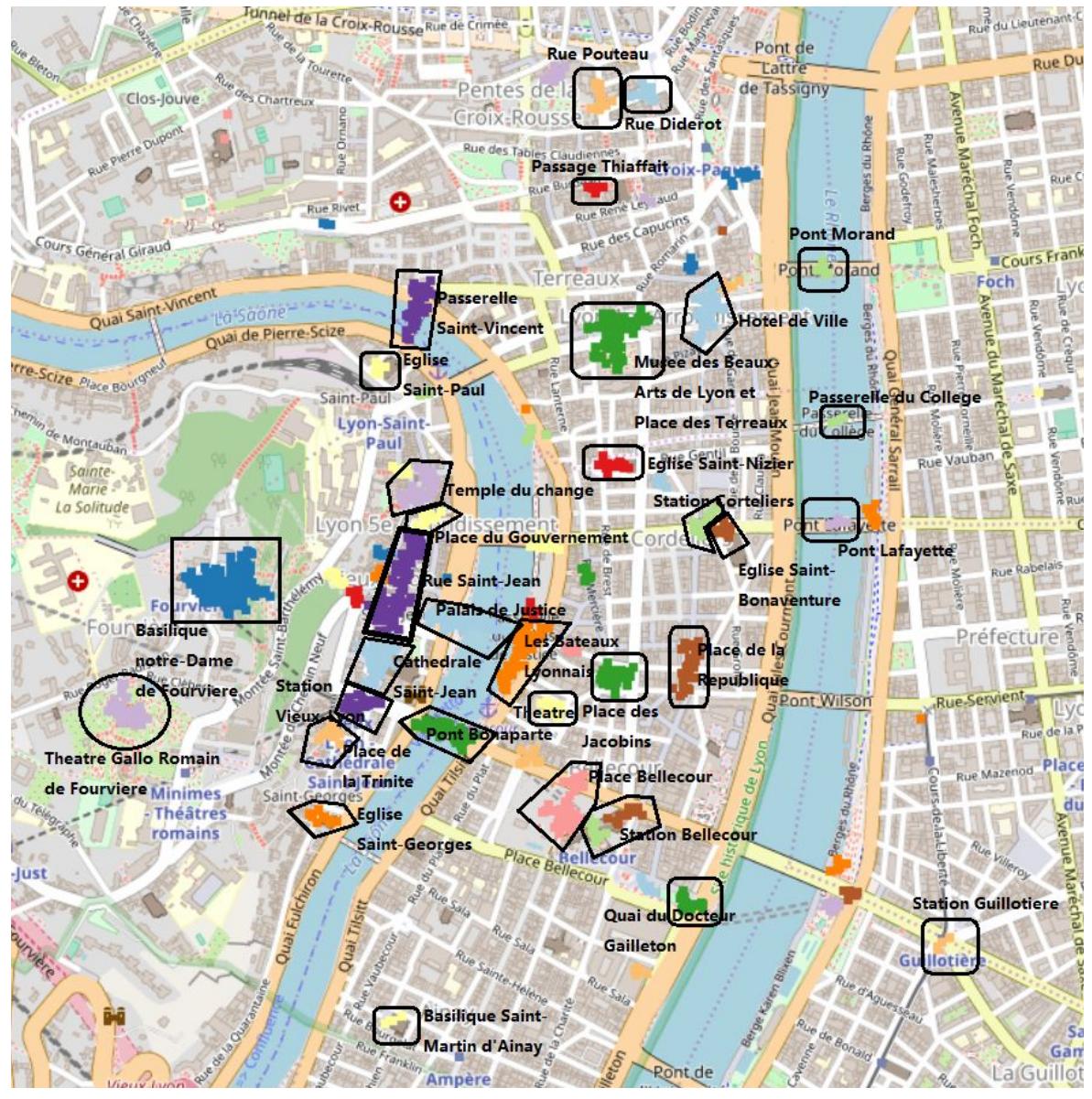


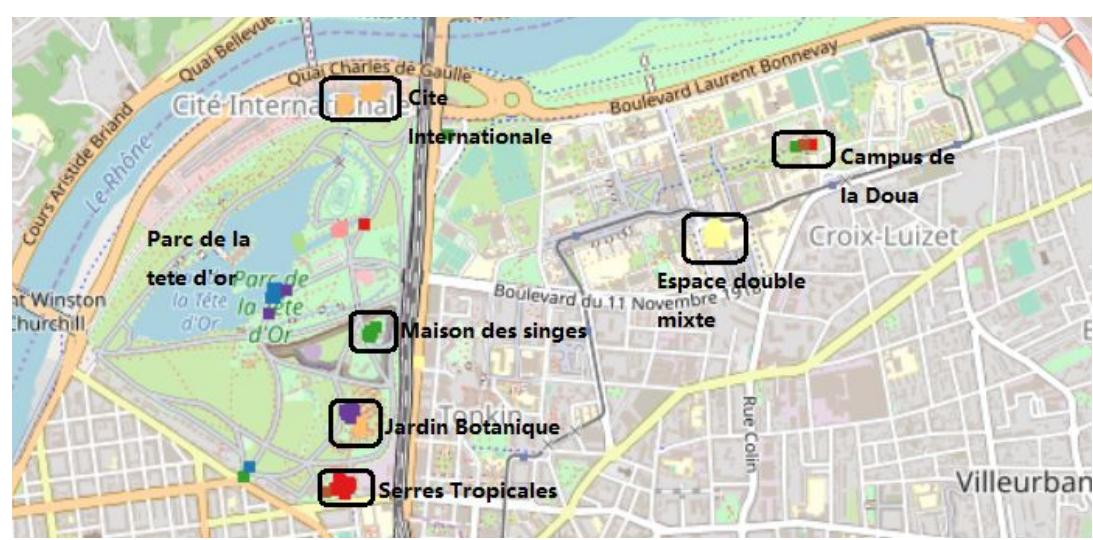
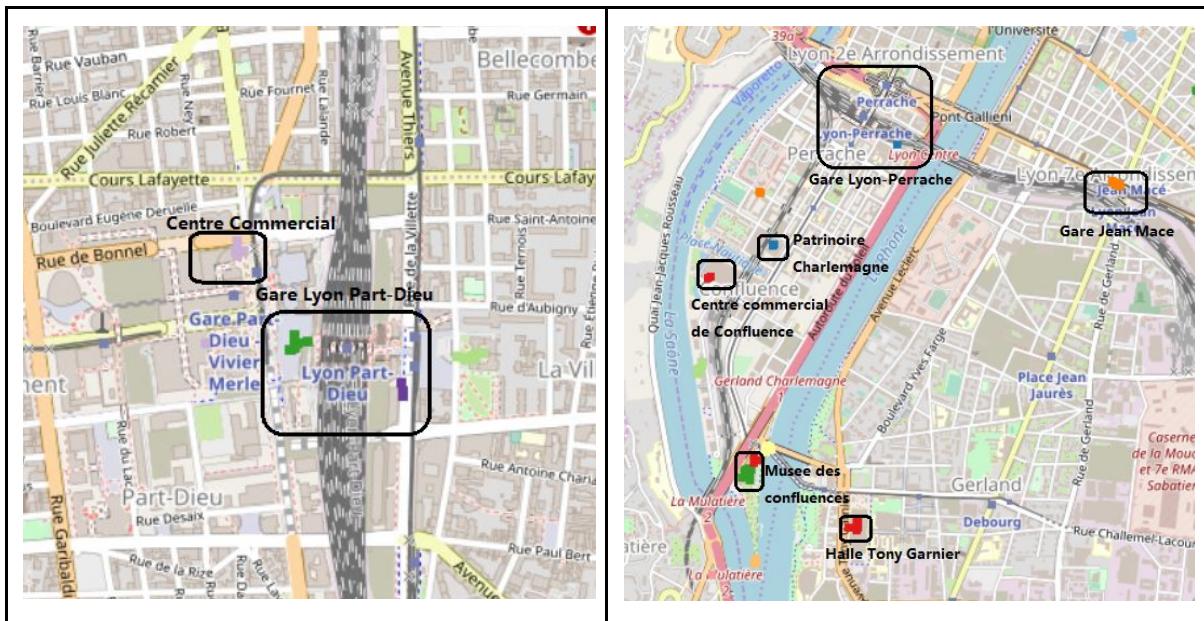
eps = 0.0005
minPoints = 10

Pour aller plus loin, nous avons diminué la valeur de eps à 0.0005. Dans ce cas là, les clusters deviennent plus discrets et nous voyons mieux les attractions principales dans la ville. Cependant, nous observons que la place bellecour a été divisé en deux clusters, ce fait résulte de eps très petite.

Visualisation des points d'intérêt

D'après des analyses précédentes, nous choisissons étudier des points d'intérêt en utilisant l'algorithme DBScan avec $\text{eps} = 0.0008$ et $\text{minPoints} = 10$

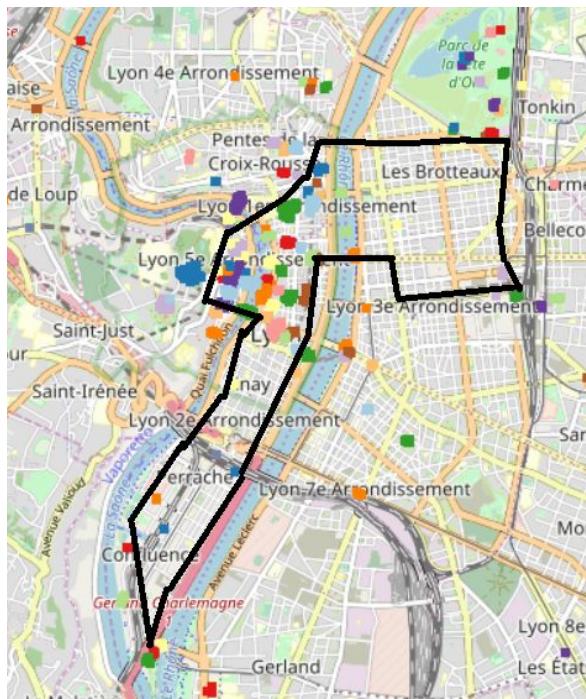




Interprétation des points d'intérêt

Nous avons bien chercher des point d'intérêt à partir des étapes précédentes. Nous pouvons bien observer que les zones plus denses sont Fourvière, Vieux-Lyon, Croix-Rousse, Terreaux, Cordelier, Bellecour, Confluence, Part-Dieu et Parc de la tête d'or. Pour améliorer ses transports en communs et la vie des touristes visitant Lyon, nous proposons d'ajouter certains trajets de bus, de tramway ou de métro.

1. un trajet direct de parc de la tête d'or a Croix-Rousse
2. un bus de plaine des sports du parc de Parilly a l'aeroport de Lyon-Bron
3. des trajets directs entre les 3 gares de Lyon
4. un bus de l'aeroport de Lyon-Bron au Gare Lyon part-dieu
5. une station de bus autour de chaque basilique ou eglise
6. une station de metro a l'entree du parc de la tete d'or
7. un bus touristique comme Paris. Ci-dessous, un trajet proposé qui traverse au maximum des points d'intérêt.



Description des points d'intérêt grâce à la fouille de motifs

Dans la fouille de motifs, nous avons deux paramètres importants, le premier c'est la fréquence, une motif avec peu de fréquence peut-être s'est produit par hasard. Le deuxième c'est la confiance. Dans une motif X->Y, Y est plus probablement présent en condition de X si la confiance est plus grande.

Pour obtenir des itemsets intéressants, nous avons descendu le support minimum relatif de itemset à 1%. Nous pouvons aussi choisir le type de itemset calculé. Nous observons bien que itemsets maximales sont inclus dans les itemsets fermés, et les itemsets

fermés sont inclus dans les itemsets fréquents. Donc ici la meilleure solution c'est de calculer les itemsets maximales vu qu'ils sont tous des itemsets fréquents et ils représentent tous les itemsets pour générer des motifs et en plus il n'y aura pas de redondance de la motif générée non plus.

les itemsets fermés:

Row ID	ItemSet	ItemsetSize	ItemSetSupport	RelativeItemSetSupport%
Row0	[street]	1	78	1.117
Row1	[instagramapp]	1	70	1.002
Row2	[squareformat,square]	2	71	1.016
Row3	[lumières]	1	72	1.031
Row4	[race]	1	73	1.045
Row5	[convention]	1	75	1.074
Row6	[basilique]	1	77	1.102
Row7	[byinstagram,uploaded]	2	78	1.117
Row8	[byinstagram,uploaded,square]	3	73	1.045
Row9	[sunset]	1	120	1.718
Row10	[sunset,nuages]	2	119	1.703
Row11	[nuages]	1	121	1.732

les itemsets fréquents:

Row ID	ItemSet	ItemsetSize	ItemSetSupport	RelativeItemSetSupport%
Row0	[street]	1	78	1.117
Row1	[instagramapp]	1	70	1.002
Row2	[squareformat,square]	2	71	1.016
Row3	[squareformat]	1	71	1.016
Row4	[lumières]	1	72	1.031
Row5	[race]	1	73	1.045
Row6	[convention]	1	75	1.074
Row7	[basilique]	1	77	1.102
Row8	[byinstagram,uploaded]	2	78	1.117
Row9	[byinstagram]	1	78	1.117
Row10	[byinstagram,square,uploaded]	3	73	1.045
Row11	[byinstagram,square]	2	73	1.045
Row12	[sunset]	1	120	1.718
Row13	[sunset,nuages]	2	119	1.703
Row14	[nuages]	1	121	1.732

les itemsets maximaux:

Row ID	ItemSet	ItemsetSize	ItemSetSupport	RelativeItemSetSupport%
Row0	[street]	1	78	1.117
Row1	[instagramapp]	1	70	1.002
Row2	[squareformat,square]	2	71	1.016
Row3	[lumières]	1	72	1.031
Row4	[race]	1	73	1.045
Row5	[convention]	1	75	1.074
Row6	[basilique]	1	77	1.102
Row7	[byinstagram,uploaded,square]	3	73	1.045
Row8	[sunset,nuages]	2	119	1.703
Row9	[taz]	1	84	1.202
Row10	[river]	1	85	1.217

Motif entre tags et tags

Dans un soucis de clarification des résultats, on doit supprimer les tags qui ne nous semblent pas pertinent pour l'étude (instagram, uploaded, iphone, etc)

Avec une confiance minimum de 40% et un support minimum de 3% nous obtenons:

conséquence	antécédent
villeurbanne	cross,insa
symbol	contemporaryart
secret	contemporaryart
nuage	sunset

Si on continue à descendre le support minimum des itemsets, on aura plus de motifs intéressants

conséquence	antécédent
contemporaryart	musée
confluence	musée
artific	feu

Nous avons bien observé que les tags insa, villeurbanne ont une forte association avec cross. Dans le tableau d'origine, on a trouvé qu'ils sont tous pris en année 2013. Donc nous avons cherché ces mot clés sur Google, et on a trouvé le page de CIA qui a organisé une match de cross en 2013.

cross	[insa,villeurbann]	100
cross	[insa]	100
cross	[villeurbann]	75

Le XXIIIe Cross de l'INSA est parti depuis 301 jours 16 heures 01 min 30 sec

Aux dernières nouvelles...

- Les partenaires de la 23ème édition
- Résultats de la 23ème édition
- XXXII-Cross de l'INSA
- Résultats du Cross 2014
- Le parcours 2016

Ce qui fait le buzz...

- Calendrier
- Résultats de la 23ème édition
- Tee-shirt Club 2012
- Initiation des membres de l'AS Féminisme
- XXXII-Cross de l'INSA



TEASER DU CROSS 2013

Parution: 04/04/2013 | Mis à jour: 03/04/2013 | 446 tags | 192



Nous avons observé aussi la cooccurrence de "feu" et "aritifice"

1	feu	<---	[artific]
0.884	artific	<---	[feu]

D'après le tableau d'origine, les photos avec le tags de "feu" "artifice" sont tous pris en 14 Juillet en 2014. Il s'agit de la Fête nationale française.

S tags	I date_taken_day	I date_taken_month	I date_t...
yon,feu,artifice,Sème	14	7	2014
yon,feu,artifice,Sème	14	7	2014
yon,feu,artifice,Sème	14	7	2014
yon,feu,artifice,Sème	14	7	2014
yon,feu,artifice,Sème	14	7	2014
yon,feu,artifice,Sème	14	7	2014
yon,feu,artifice,Sème	14	7	2014
yon,feu,artifice,Sème	14	7	2014

On obtient alors ce résultat, si l'on reporte les tags sur les clusters sur notre carte



Motif entre les tags et clusters

En utilisant le node "one to many", One to Many

nous arriverons de prendre tous les clusters comme colonnes. Ce fait nous permet de trouver des motifs associants les tags et les clusters.

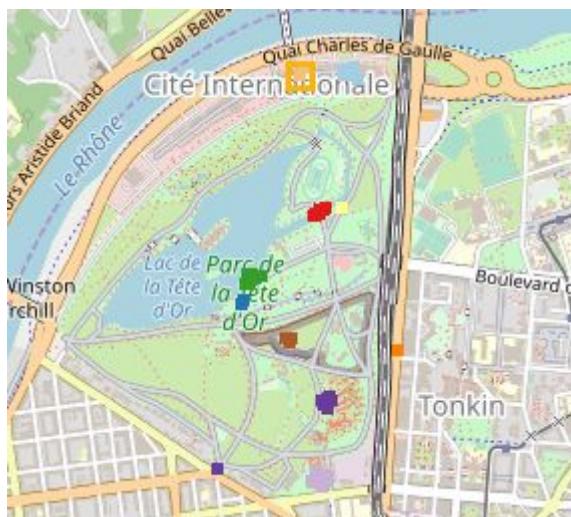
En posant une confiance minimum de 60% et support minimum de 3%, on a des motifs suivant:

conséquence	antécédent
cluster 85	nuage,sunset
cluster 6	fouvière

cluster 1	contemporaryart, secret, symbol,musée
cluster 157,cluster 158	insa,villeurbanne,cross
cluster 2	charlemagne

la motif qui a les antécédents “symbol”, “contemporary” “art”, “secret”, et la conséquence le cluster 1. Si nous observons sur le plan, on voit bien que le premier cluster(le cluster orange entouré) s’agit bien le Musée d’art contemporain. Nous observons aussi que sa confiance est très haut à 100%, donc cette motif est fiable.

1	[symbol,contemporaryart]	2	3.846	100
1	[symbol]	3	5.769	100
1	[contemporaryart,secret]	2	3.846	100
1	[contemporaryart]	4	7.692	100
1	[secret]	4	7.692	100

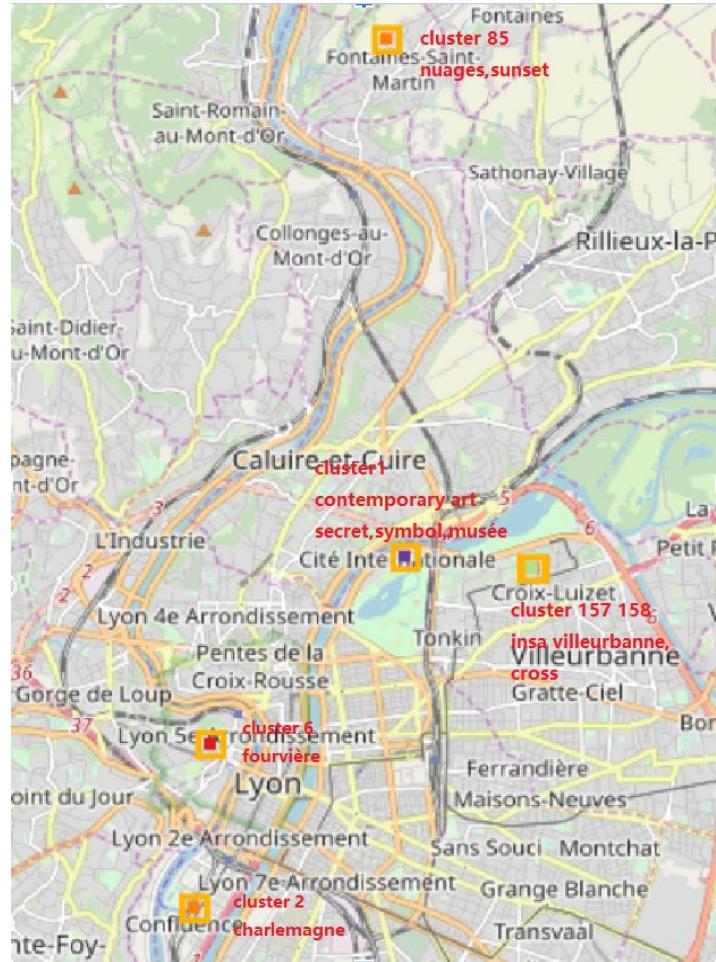


Nous observons que le cluster 63 s’agit des photos qui sont pris par un même user et qui a taggé ces photos “sunset” “nuages”.

63	[nuag]	1	1.923	50
63	[sunset]	1	1.923	50
6	[fourvi]	1	1.923	50

S	user	D	lat	D	long	S	tags	S	title	S	Winner Cluster
52638247@N00		45.848	4.852			sunset,nuages	DSC02506	63			
52638247@N00		45.848	4.852			sunset,nuages	DSC02503	63			
52638247@N00		45.848	4.852			sunset,nuages	DSC02505	63			
52638247@N00		45.848	4.852			sunset,nuages	DSC02510	63			
52638247@N00		45.848	4.852			sunset,nuages	DSC02512	63			
52638247@N00		45.848	4.852			sunset,nuages	DSC02518	63			

On obtient alors ce résultat, si l'on reporte les tags sur les clusters sur notre carte.

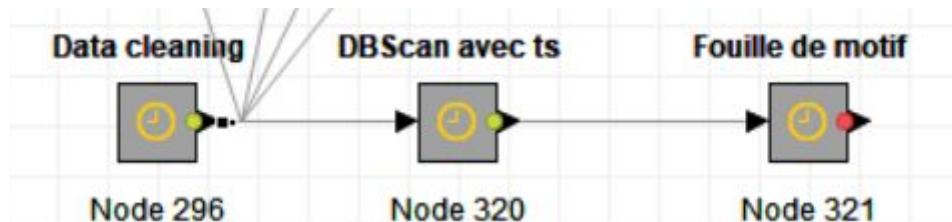


Recherche d'événements : zone dense dans le temps et/ou dans l'espace

Clustering sur les données temporelles

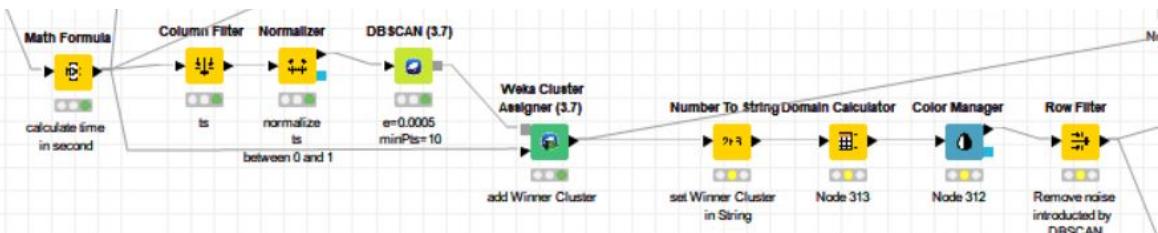
Nous pouvons identifier des événements en fonction du temps. Selon les discussions des différents algorithmes de clustering précédents, nous choisissons d'expérimenter en utilisant DBSCAN qui est basé sur la densité. Afin de pouvoir interpréter ces clusters, nous cherchons ensuite les tags associés qui sont en peu de support mais en forte de confiance.

Workflow



DBScan avec ts

Pour DBSCAN, nous prenons eps égale à 0.0005 et minpoints égale à 10.



Et nous calculons la fouille de motif de la même façon que la dernière fois, la partie fouille de motif. Ici, nous cherchons des tags avec un support minimum de 1% et une confidence minimale de 10%.

Résultats

Cluster	Date	Tags	Commentaire/événement
1	06/12/2013-08/12/2013	fetedeslumier,fetedeslumi, lumi, nuit, spectacl	fete des lumieres
124	04/03/2013	insa, villeurbann.cross	Même utilisateur
12	02/10/2015-06/10/2015	octogôn, gam, convent	OctoGônes 2015
14	19/12/2014	miniatur,starwarsidentit,cinem, vieuxlyon	Même utilisateur
35	30/10/2015-01/11/2015	saint-martin,basil,confluenc	
3	20/10/2015-22/10/2015	retour,delorean,futur	La DeLorean de "Retour vers le Futur"
44	30/07/2014-21/08/2014	detail,oullin,steel,secret	vacance d'ete

58	01/03/2014-06/03/2014	rouss,traboul,rhon,earlysunday morning,dimanch,jacquesmey nierdemalvial	
69	13/07/2014-20/07/2014	romain,artific,feu,nuit	feu d'artifice en fete nationale
8	23/10/2015-26/10/2015	ancient,automn,natur,fil,fourvi	Parc de la tête d'or collection automne
91	16/10/2014-20/10/2014	ldoll,extra,rac,cours	Ldoll 2014, “Lyon Extra Race”
95	12/09/2014-14/09/2014	sit,twentiethcentury, vieuxlyon,urban,rhônealp	
9	27/06/2015-28/06/2015	georg,bonaventur,sanctuair,ba sil,summicron,leitz,fourvi,frankr eich	Même utilisateur

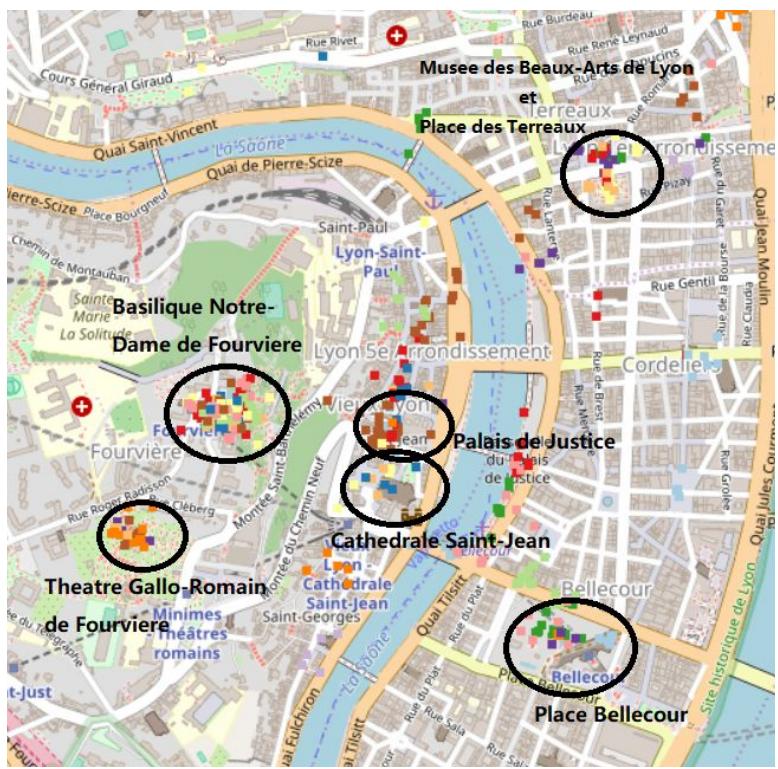
Selon le tableau ci-dessus, nous avons bien identifié certains événement au cours des années 2013-2015. Nous observons également des périodes où des touristes étaient en vacances à Lyon, et ils sont la vacance d'été, la fête nationale, la fête des lumières surtout au week-end et au jour férié. Par contre, nous ne savons pas des zones de ces événements, nous étudierons dans la partie suivante.

En observant des périodes de tourisme, nous pouvons distribuer plus de bus touristique à la fête, au week-end ou au jour férié, et moins de bus aux jours de la semaine.

Clustering sur les données spatiales et temporelles

Nous chercherons alors à caractériser divers types d'évènements : Un point d'intérêt peut être ponctuel ou récurrent. Il faut d'abord lors du prétraitement transformer les dates en timestamp comme le dernier. On peut ensuite appliquer l'algorithme de clustering sur les données spatiales et temporelles. De même, nous choisissons d'expérimenter en utilisant DBSCAN qui est basé sur la densité. Il regroupe des données en coordonnée similaire et au temps similaire.

DBSCAN
avec e = 0.003, minPts = 10



Des zones avec plusieurs clusters(avec couleurs variés) sont des points d'intérêt récurrents car dans un même zone, le temps de prise de photos est différent. Des touristes sont plus intéressés par ces zones.

Nous pouvons identifier des événements ponctuels par des zones avec un seul cluster.



20/7/2013-25/7/2013 (photos)
20/7/2013-28/7/2013 (événement)

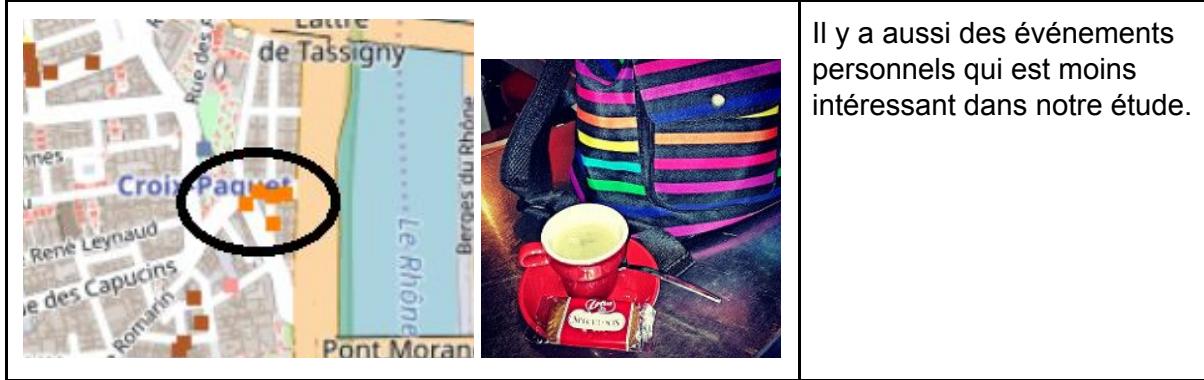
<<2013 IPC Athletics World Championships>> à Plaine des Sports du Parc de Pariilly

Alors, la proposition d'un bus de plaine des sports du parc de Pariilly à l'aéroport de Lyon-Bron peut être temporelle.



18/10/2014 (photos)
Chaque 2 ans au Octobre avec une durée de 2 jours (événement)

<<Idoll festival Lyon>> à La Doua - Double Mixte



Il y a aussi des événements personnels qui est moins intéressant dans notre étude.

Conclusion

L'analyse des données fournies nous permet de retirer plusieurs faits intéressants et nous rappelle certains événements qui ont lieu à Lyon.

Pendant ce TP, nous avons eu deux difficultés principales. Le premier c'est d'appliquer stratégies différents selon la situation différents et finalement selon le besoin de client. Par exemple, si nous cherchons des clusters sur la région de Grand Lyon en analysant la densité, la région banlieue a une grande désavantage par rapport à la région de centre ville puisque les banlieus ont plus grande surface et moins de habitants. C'est pour cela que si nous descendons eps et augmentons minPoints pour bien distinguer les événements en centre villes, les cluster en banlieue seront disparus. Nous devons appliquer des stratégies différents ainsi que les données de toute la région seront pas négligées. En d'autres mots, dans notre étude, parfois un seul algorithme ne suffit pas pour bien analyser les photos, nous devons appliquer plusieurs algorithmes dans certain ordre pour l'améliorer.

La deuxième c'est de traiter la similarité entre les photos. Effectivement, nous avons bien trouvé des événements à Lyon, mais on voit aussi des exceptions dans notre étude. Par exemple, le cluster "nuages, sunset" s'agit d'un user qui a pris beaucoup de photos. Ces photos là ont exactement le même tags et aussi ils sont pris en même temps dans le même endroit. Ces photos sont interprétées par l'algorithme avec un résultat de grande similarité or il y a plein de photos avec des tags avec la différence très petite qui ne sont pas traité comme similaires. Alors nous devons penser à modifier la définition de similarité dans notre cas d'étude.