

Data Mining

Rapport de Projet

Raphaël Diana, Jose Francisco Saray Villamizar & Alix Gonnnot

Janvier 2017

Sujet : Exploitation de données Flickr

1 Présentation des données utilisées

L'objectif de ce projet était de réaliser une fouille de données sur un data set de taille importante. Nous avons choisi de travailler avec l'un des data sets fournis, celui des données Flickr. Ce data set contient des méta-données de photos de la ville de Lyon stockées sur le site de partage de photographie Flickr. Les données fournies sont stockées dans un fichier csv qui contient les colonnes suivantes :

- **id user** : identifiant de l'utilisateur ayant uploadé la photo
- **longitude** : longitude à laquelle la photo a été prise
- **latitude** : latitude à laquelle la photo a été prise
- **hashtags** : liste des hashtags associés à la photo sous la forme : “#1, #2, etc.”
- **legend** : la description de la photo donnée par l'utilisateur
- **minute_taken** : la minute à laquelle la photo a été prise
- **hour_taken** : l'heure à laquelle la photo a été prise
- **day_taken** : le jour où la photo a été prise
- **month_taken** : le mois où la photo a été prise
- **year_taken** : l'année pendant laquelle la photo a été prise
- **hour_upload** : l'heure à laquelle la photo a été uploadée
- **day_upload** : le jour où la photo a été uploadée
- **month_upload** : le mois où la photo a été uploadée
- **year_upload** : l'année pendant laquelle la photo a été uploadée

Le fichier d'origine contient les méta-données de 83 111 photos.

2 Objectifs

Notre objectif dans ce projet était d'identifier les points d'intérêts de la ville de Lyon en recherchant les endroits de la ville où de gros volumes de photos ont été prises. Nous souhaitions extraire les points d'intérêts “physiques”, comme des endroits à visiter, et les points d'intérêts temporels comme des festivals, des événements, ...

Nous voulions également pouvoir visualiser les tags fréquemment associés aux points d'intérêts afin de trouver la fonction ou le nom de ces points d'intérêts.

Finalement, nous nous sommes aussi intéressés aux intérêts de chaque utilisateur, qui sont extraits à partir de mots clés utilisés pour décrire ses photos.

3 Réalisation du projet

Pour réaliser ce projet, nous avons utilisé deux outils, tout d'abord Knime, que nous avons choisi car nous l'avions utilisé en séances de travaux pratiques et Sci-Kit que nous avons choisi car il nous a permis de réaliser des opérations qui n'étaient pas disponibles dans Knime. Pour extraire les intérêts par utilisateur, nous avons utilisé un script R, car certaines opérations (partitionnement, tokenization) nécessaires pour extraire la base de transaction propre à chaque utilisateur nous ont semblé plus facilement réalisables à l'aide du langage R.

3.1 Clustering

Pour commencer, nous avons nettoyé les données d'origines car elles comportaient des incohérences (e.g. : year_taken aberrante ou encore year_taken supérieure à year_uploaded). Nous nous sommes concentrés sur les photos prises entre 2004 et 2014, qui comportaient une longitude et une latitude non nulles ainsi que des hashtags et une description non vides. Pour cela, nous avons utilisé le noeud Rule-based Row Filter de Knime. Nous avons également filtré les individus du data set par position géographique en se restreignant à la ville de Lyon et sa proche périphérie. Après filtrage notre data set ne contenait plus que 51454 lignes, soit une réduction d'environ 40% de la taille de la base de données.

Pour identifier les points d'intérêt de la ville, nous avons cherché à réaliser un clustering. Nous avons utilisé expérimenté différentes méthodes de façon à obtenir la meilleure répartition des points possible.

3.1.1 Méthodes hiérarchiques

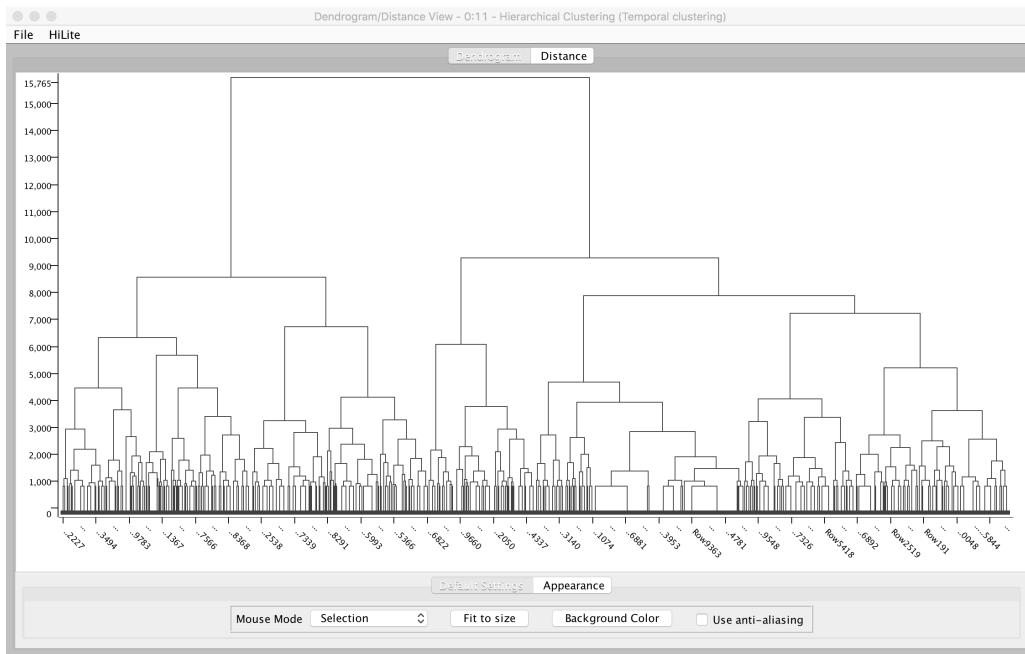


FIGURE 1 – Clustering Hiérachique

Nous avons essayé le clustering hiérarchique pour voir si des classes étaient identifiable dans nos données. La distance utilisée est la distance euclidienne, calculée sur les champs latitude, longitude, day_taken et month_taken normalisés entre 0 et 1, pour découvrir les clusters géographique récurrent chaque année. Le dendrogramme n'a pas permis de déterminer des zones d'intérêt particulièrement définit.

3.1.2 k-Means

En utilisant la méthode des k-Means sur notre data set avec les longitudes et latitudes normalisées, nous avons obtenu les 60 clusters suivant.

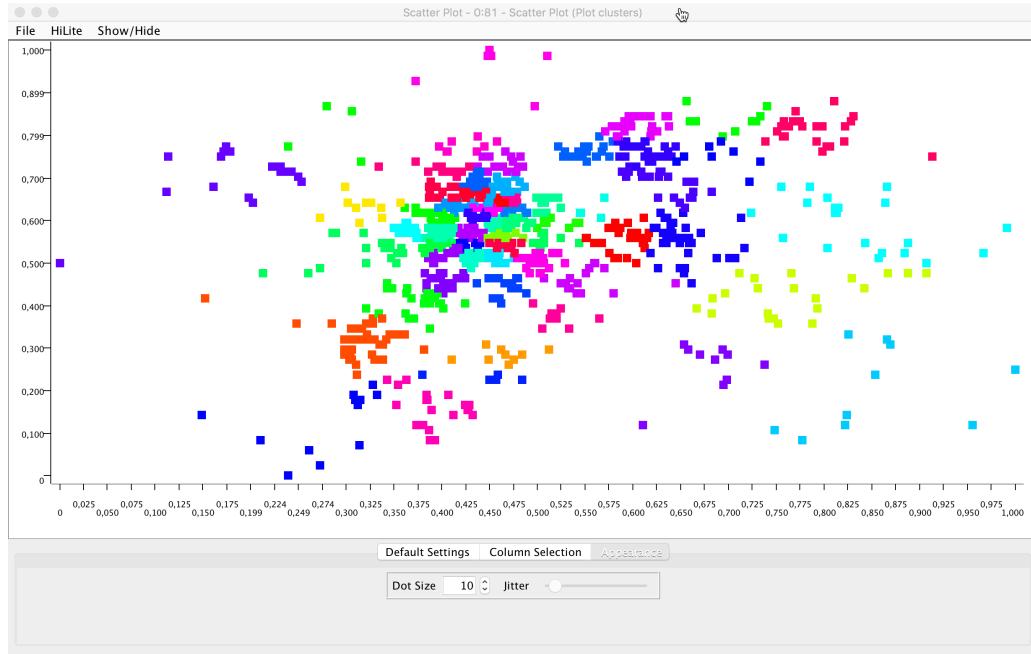


FIGURE 2 – k-Means

Avec cette méthode on peut déterminer que le Vieux Lyon est la zone où le plus de photos sont prises. D'autres zones sont détectables comme le parc de la Tête d'Or ou le quartier de Confluence. En revanche le fait de devoir spécifier à l'avance le nombre de clusters signifie que l'on doit juger la granularité a priori, or nous ne savons pas combien de POI contient notre data set.

3.1.3 DBSCAN

Pour pallier aux limitations de k-means nous avons utiliser l'algorithme de segmentation DBSCAN qui travaille sur la densité des données et ne nécessite pas de connaître à l'avance le nombre de cluster. Le noeud DBSCAN a besoin d'une matrice de distance pour calculer les densités. Nous avons utiliser la formule de Haversine pour calculer les distances à partir des latitudes/longitudes. Avec un $\epsilon = 200$ et un nombre de points minimum par zone dense de 10, nous obtenons les résultats suivants.

Nous obtenons des clusters qui se rapproche plus des POI. En revanche le fait que le epsilon soit fixe ne permet pas de faire ressortir toute la variété de densité des POI. De plus pour toutes les méthodes de clustering présentées jusqu'ici, nous avons travailler sur une partition limitée du data set (2000 lignes) car nos machines ne permettait pas d'exécuter les algorithmes en temps raisonnable sur plus de données.

3.1.4 Mean-shift

Nous nous sommes tournés vers des méthodes de clustering non paramétrées pour explorer le data set. Nous avons choisis l'algorithme Mean-shift. Pour pouvoir utiliser cet algorithme de clustering dans notre projet, nous nous sommes inspirés du travail réalisé par Rémi Domingues ([github](#)) sur le même dataset. Il a en effet écrit un script python qui permettait d'exécuter l'algorithme Mean Shift de la bibliothèque python Scikit-learn sur ses données et de filtrer

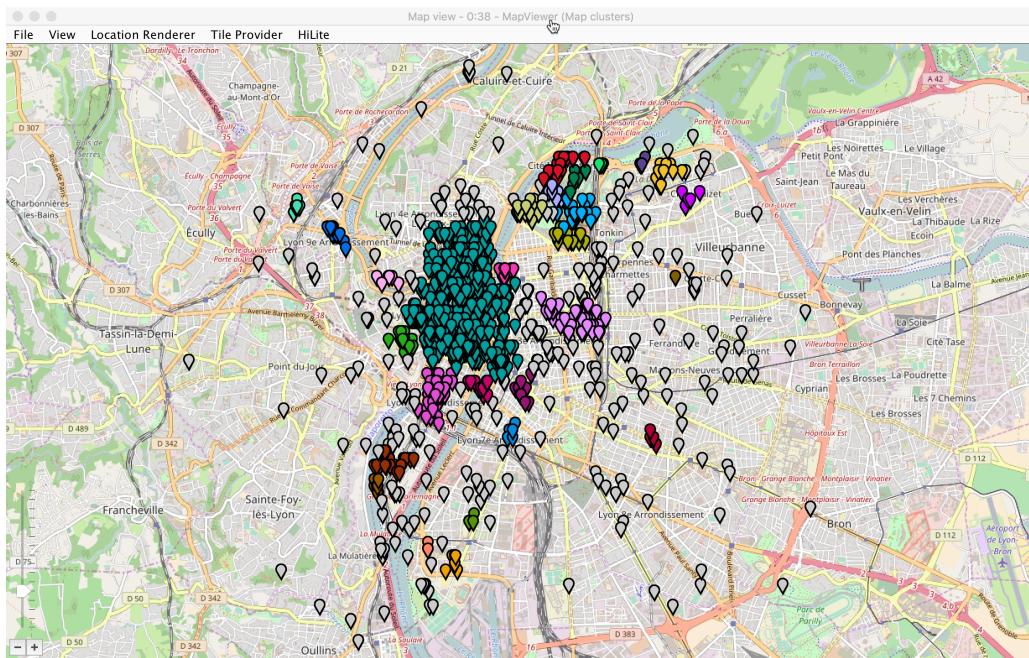


FIGURE 3 – DBSCAN

les résultats selon ses besoins. Nous avons adapté ce script pour pouvoir l'utiliser directement dans notre workflow Knime. En ne gardant que les clusters contenant plus de 10 élément nous obtenons les résultats suivants.

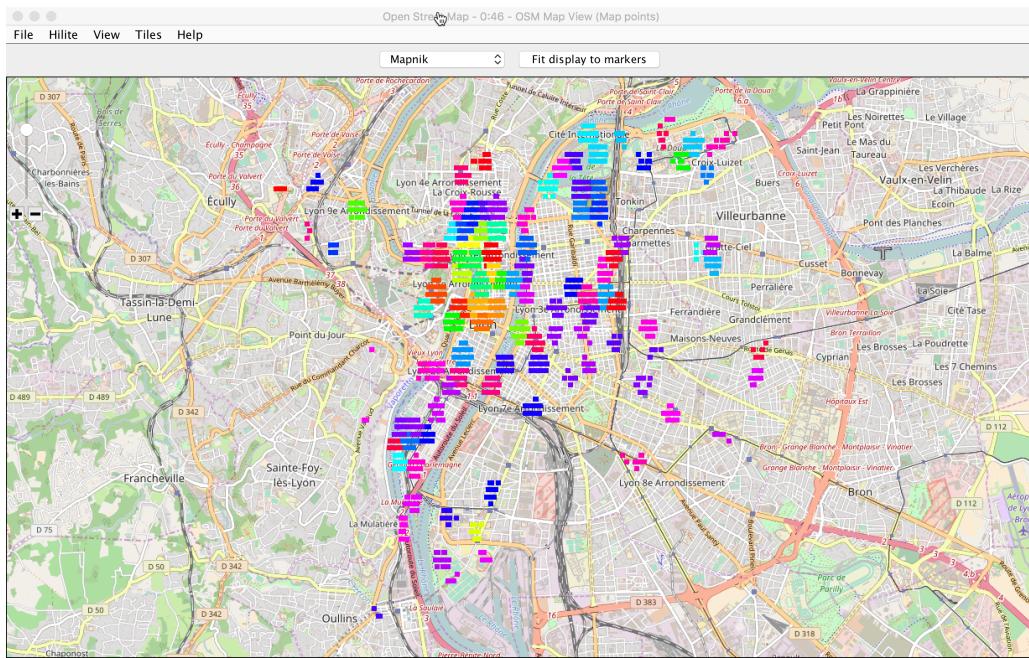


FIGURE 4 – Mean-shift

Nous avons également essayé d'utiliser Mean-shift pour récupérer des clusters temporels (récurrent d'année en année) mais cela n'a pas été concluant. Un simple histogramme montrant le nombre de photos prises par jour de l'année nous a donné plus d'information. On y remarque nettement la période de la fêtes des lumières en fin d'année.

En revanche le clustering géographique nous permet de nous rapprocher des POI réels sans

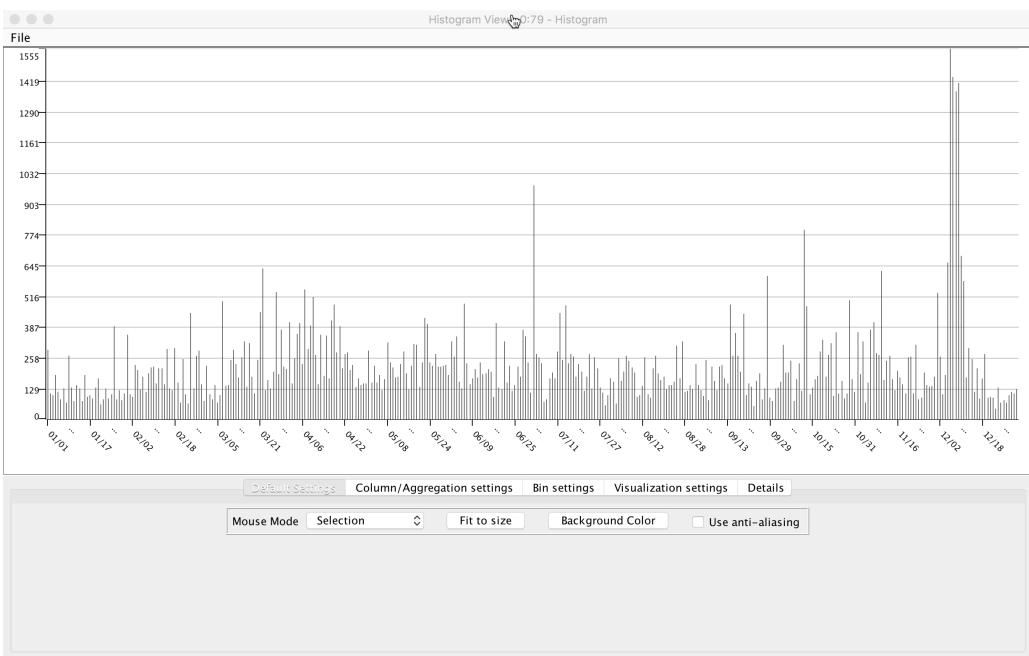


FIGURE 5 – Nombre de photos prises selon le jour de l'année

beaucoup de connaissances à priori sur le data set. Cela nous a permis de sélectionner des clusters pour les labeliser grâce aux hashtags.

3.2 Traitement des tags associés à un cluster

De façon à pouvoir extraire des informations à propos d'un cluster, nous avons appliqué deux traitements aux tags associés à ce cluster. Ces tags correspondent aux mots présents dans la colonne hashtags de toutes les photos appartenant au cluster.

Le premier traitement est la création d'un nuage de tags, c'est à dire une image comportant des mots, la taille des mots sur l'image représentant leur fréquence d'apparition dans l'ensemble considéré. Pour réaliser cette image, nous avons assemblé les noeuds Knime *Cell Splitter*, *Unpivot* et *Value Counting* pour compter le nombre d'apparition de chaque tag et fourni cette nouvelle table, filtrée et formatée au noeud *Tag Cloud*. Nous avons fait le choix de filtrer la table permettant de construire le nuage de tag de façon à le rendre plus lisible. Nous avons supprimé les tags qui n'apportaient aucune informations tels que "France" et "Lyon" ainsi que les tags qui n'apparaissaient pas un nombre suffisant de fois pour être considéré comme intéressant.



FIGURE 6 – Nuage de tags du le cluster correspondant à la Basilique de Fourvière

Comme on peut le voir sur la capture d'écran ci-dessus, la création de ce nuage de tags permet d'obtenir des informations intéressantes sur le cluster considéré comme sa désignation ou ce qu'il s'y passe. Dans notre exemple, on peut voir que le terme "fourrière" est fréquemment

associé au cluster, il doit s'agir du nom du lieu, on voit également les termes “church” et “église”, il doit donc s'agir d'un lieu associé à la religion.

Le second traitement que nous avons appliqué à ces tags est l'extraction de règles d'association à l'aide de l'algorithme APriori. Pour cela, nous avons utilisé de nouveau le noeud *Cell Splitter* pour faire de la chaîne de caractère stockée dans la colonne hashtags une liste de mots que nous avons fourni au noeud *Association Rule Learner (Borgelt)* qui est adapté au traitement de chaînes de caractères.

Row ID	Antecedent	Consequent	RuleConfidence%	ItemSets	Relativ...	Absolut...	Relativ...	RuleLift	RuleLi...	Ab
Row175	[basilique]	fourrière	84.4	297	11.132	352	13.2	3.23	322.97	697
Row169	[church]	basilica	58.1	186	6.971	320	12	5.919	591.9	262
Row180	[church]	fourrière	46.6	149	5.585	320	12	1.782	178.23	697
Row174	[church]	basilique	42.8	137	5.135	320	12	3.245	324.5	352
Row176	[fourrière]	basilique	42.6	297	11.132	697	26.1	3.23	322.97	352
Row173	[basilique]	church	38.9	137	5.135	352	13.2	3.245	324.5	320
Row172	[basilique,fourrière]	church	38.7	115	4.31	297	11.1	3.228	322.83	320
Row139	[rhône]	grandlyon	37.9	105	3.936	277	10.4	9.632	963.18	105
Row154	[europe]	69005	37.1	165	6.184	445	16.7	5.785	578.51	171
Row158	[rhône]	europe	36.5	101	3.786	277	10.4	2.186	218.61	445
Row168	[basilique]	basilica	35.2	124	4.648	352	13.2	3.587	358.73	262
Row167	[basilique,fourrière]	basilica	34.7	103	3.861	297	11.1	3.531	353.15	262
Row143	[church]	cathedral	34.1	109	4.085	320	12	6.732	673.18	135
Row152	[church]	fourrière	30.6	98	3.673	320	12	3.731	373.09	219
Row183	[europe]	fourrière	29.7	132	4.947	445	16.7	1.135	113.54	697
Row156	[rhône]	light	28.9	80	2.998	277	10.4	3.759	375.87	205
Row153	[rhône]	69005	28.5	79	2.961	277	10.4	4.45	449.48	171
Row76	[church]	église	28.4	91	3.411	320	12	6.835	683.52	111
Row128	[rhône]	8décembre	28.2	78	2.924	277	10.4	6.367	636.68	118
Row135	[rhône]	fête	28.2	78	2.924	277	10.4	7.365	736.55	102
Row147	[rhône]	fêtedeslumières	28.2	78	2.924	277	10.4	3.683	368.27	204
Row160	[rhône]	fourrière	28.2	78	2.924	277	10.4	1.078	107.79	697
Row141	[basilique,fourrière]	cathedral	27.9	83	3.111	297	11.1	5.523	552.3	135
Row83	[rhône]	8thdecember	27.8	77	2.886	277	10.4	9.632	963.18	77
Row85	[rhône]	8december	27.8	77	2.886	277	10.4	9.632	963.18	77
Row87	[rhône]	dec8	27.8	77	2.886	277	10.4	9.632	963.18	77
Row89	[rhône]	8dec	27.8	77	2.886	277	10.4	9.632	963.18	77
Row91	[rhône]	december8	27.8	77	2.886	277	10.4	9.632	963.18	77
Row93	[rhône]	december8th	27.8	77	2.886	277	10.4	9.508	950.83	78
Row98	[rhône]	5èmearrondissement	27.8	77	2.886	277	10.4	9.044	904.45	82
Row100	[rhône]	lyon5	27.8	77	2.886	277	10.4	9.044	904.45	82
Row102	[rhône]	5é	27.8	77	2.886	277	10.4	9.044	904.45	82
Row104	[rhône]	cinquièmearrondissement	27.8	77	2.886	277	10.4	9.044	904.45	82
Row106	[rhône]	5earrondissement	27.8	77	2.886	277	10.4	9.044	904.45	82
Row108	[rhône]	cinquième	27.8	77	2.886	277	10.4	9.044	904.45	82
Row110	[rhône]	5ème	27.8	77	2.886	277	10.4	9.044	904.45	82
Row112	[rhône]	fêtedeslumières2009	27.8	77	2.886	277	10.4	9.044	904.45	82
Row114	[rhône]	festival	27.8	77	2.886	277	10.4	8.725	872.52	85

FIGURE 7 – Règles d'associations pour le cluster correspondant à la Basilique de Fourvière

Le but de cette extraction de règles d'association est assez similaire à celui de la création du nuage de tags, en effet, cela permet d'obtenir plus d'informations sur le cluster. Ci-dessus, on peut voir les règles d'associations extraites du cluster associé géographiquement à l'emplacement de la basilique de Fourvière classées par valeur de confiance. Si on prend par exemple la septième règle de la liste (“Basilique,fourvière” -> “church”), on peut en déduire qu'une Basilique est construite à cette endroit et qu'il s'agit d'un bâtiment religieux.

3.3 Extraction des intérêts d'un utilisateur

Pour extraire les intérêts par utilisateur, nous avons partitionné le data set initial (flickr-original.csv), en N data sets (un par utilisateur) contenant l'ensemble de photos qui lui appartiennent.

Sur chaque sous-ensemble de photos, la colonne “hashtags” contient des mots séparés par virgule, on s'intéresse à trouver des ensembles de mots qui viennent toujours ensemble. Pour cela, on applique la fouille de itemsets à l'aide de la bibliothèque “arules” de R, et on transforme les résultats obtenus en un nuage d'ensembles de mots clé. Le traitement est appliqué à chaque

utilisateur et on obtient autant de nuages qu'on a d'utilisateurs dans les données.

Les nuages d'ensembles de mots clés associés à chaque utilisateur sont disponibles via la page web située [ici](#). On peut y voir par exemple le nuage associé à l'utilisateur *36097941@N05* montré ci-dessous.



FIGURE 8 – Nuage de mots clés de l'utilisateur *36097941@N05*

On peut voir sur cette image que les ensembles de mots clés comportent souvent les mots “église”, “basilique”, “fourvière”, “religion”, etc. On peut donc en déduire que cet utilisateur a visité la Basilique de Fourvière et il semble donc intéressé par les édifices religieux. On peut également déduire de ce nuage qu'il a probablement pris ses photos à l'aide d'un appareil photo Canon 450d car le mot “450d” revient souvent dans les ensembles de tags.

4 Conclusion

Au cours de ce projet, nous avons donc réalisés un workflow Knime qui permet d'identifier des points d'intérêts physiques(monuments, lieux de loisir) de la ville de Lyon ainsi que d'extraire des informations à leur sujet tels que leur désignation ou leur fonction. Un script R nous a permis d'extraire des informations à propos des utilisateurs.

Dans le but d'affiner cette analyse, il aurait été intéressant de parvenir à mieux filtrer les nuages de tags de façon à le rendre encore plus lisible en supprimant les répétitions liées aux différents langues utilisées dans les tags. Il nous aurait aussi été bénéfique de disposer de machines plus puissantes pour faire tourner les algorithmes des clusterisation avec des paramètres plus fins (notamment DBSCAN).

On peut aussi imaginer que le travail réalisé sur les tags fréquemment associés à un cluster pourrait servir de base à un outil permettant de taguer (labelliser) automatiquement les photos sans hashtags.