Rapport TIW 9 TP1
Détection de points d'intérêt : DBSCAN vs K-means

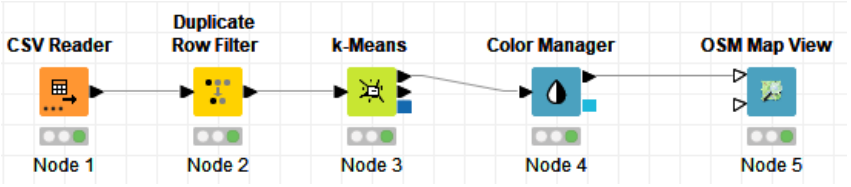
Informations du binôme:

Jeremy Thomas (11702137) et Julien Giraud (11704709)

KMEANS

Workflow

La latitude et la longitude provenant du CSV Reader sont filtrées par le nœud Duplicate Row Filter afin de supprimer les doublons. Le nœud K-means transmet ensuite chaque cluster au nœud Color Manager qui colorie chacun d'entre eux avant de les transmettre au nœud OSM Map View.

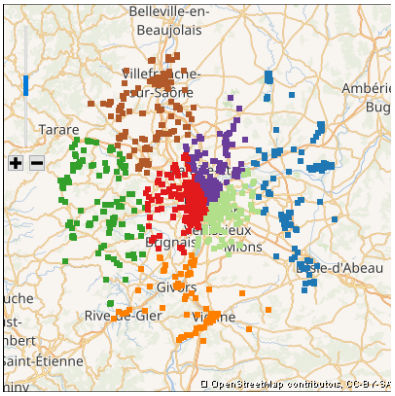


Paramétrage

Le paramétrage du nœud K-means nécessite l'entrée manuelle du nombre de clusters à générer. L'initialisation du centre du cluster est aléatoire. Nous avons essayé plusieurs seeds, le résultat est toujours le même visuellement.

Résultats

Pour la génération de 7 clusters par exemple, nous obtenons le résultat suivant :



DBSCAN

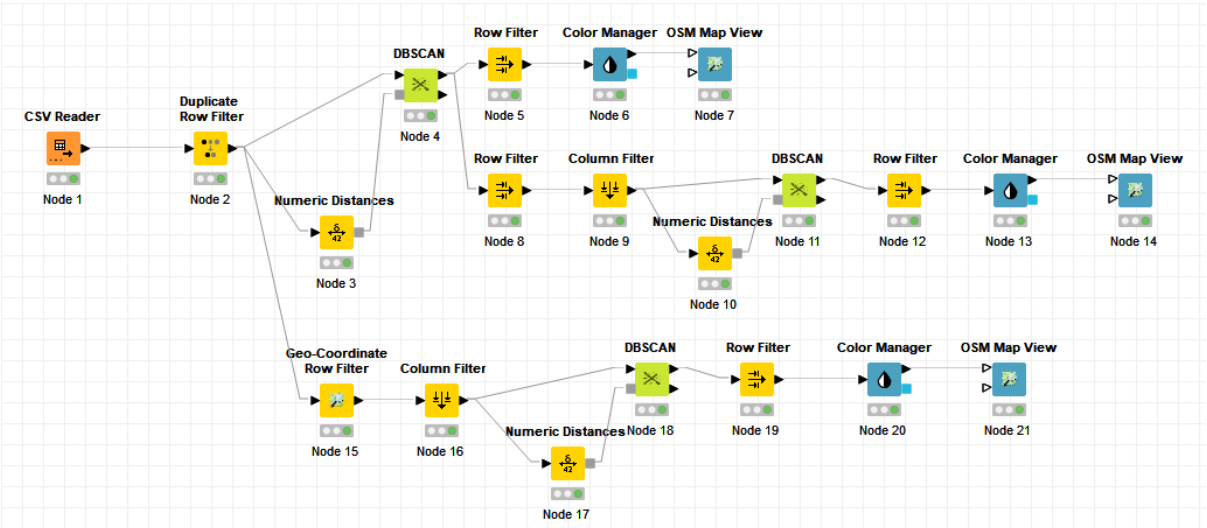
Workflow

Le workflow qui utilise l'algorithme DBSCAN est plus complet. Le nœud DBSCAN nécessite l'utilisation du nœud Numeric Distances qui force l'utilisation de la distance Euclidienne pour calculer la distance entre deux points. Nous pensons que cette distance est plus pertinente que celle de Manhattan car le jeu de données représente ici l'emplacement de photos prisent par des humains. Pour calculer la distance entre deux points du jeu Snake par exemple, la distance de Manhattan serait plus adaptée. Nous avons appliqué le DBSCAN à trois reprises sur différents ensembles de points afin de mettre en évidence des clusters plus pertinents :

- ➔ Le chemin des nœuds 1 à 7 permet de visualiser les clusters à l'échelle de la métropole de Lyon. On y trouve un grand cluster plutôt concentré sur la ville de Lyon.

- ➔ Le chemin des nœuds 1 à 14 permet de visualiser les clusters à l'intérieur du grand cluster de Lyon. On y trouve un autre grand cluster concentré sur la Presqu'île. Nous avons utilisé les points du plus grand cluster du premier dbscan pour cette visualisation.

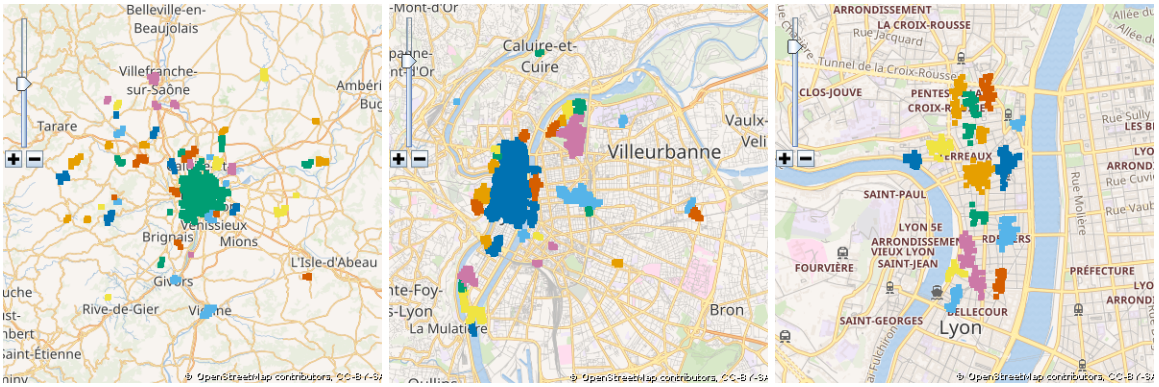
- ➔ Le chemin des nœuds 1 à 21 permet de visualiser les clusters de la Presqu'île. Cette fois nous avons utilisé les mêmes données que le premier dbscan, avec un Geo-Coordinate Row Filter paramétré pour ne laisser passer que les points présents sur la Presqu'île.



Paramétrages

- ➔ La visualisation de l'agglomération a pour paramètres ϵ = 0,01 et minimum 5 points.
- ➔ La visualisation de Lyon a pour paramètres ϵ = 0,001 et minimum 10 points.
- ➔ La visualisation de la Presqu'île a pour paramètres ϵ = 0,0005 et minimum 10 points.

Résultats



K-MEANS vs DBSCAN

K-means ne détecte pas le bruit, étant donné un nombre de clusters il a tendance à regrouper des éléments des extrémités vers le centre sans vraiment se préoccuper de la concentration des points. Les clusters obtenus n'ont pas vraiment de sens, même lorsque nous mettons à k le nombre de clusters obtenus par DBSCAN.

DBSCAN, une fois configuré pour une échelle et le bruit éliminé, permet de voir apparaître des clusters à des endroits qui ont du sens comme les centre-villes. Ce phénomène est explicable par ses deux paramètres :

- ➔ Epsilon, la distance maximale entre deux points à rapprocher.
- ➔ Le nombre minimum de points qu'il faut rassembler pour obtenir un cluster.

On peut cependant se demander s'il n'y avait pas des clusters dans le bruit, qu'un meilleur réglage aurait détecté.

Nous pouvons conclure que DBSCAN est bien plus adapté que K-means pour détecter des points d'intérêt sur une carte.