République Algérienne Démocratique et Populaire

Université des sciences et technologies Houari Boumediene

Rapport DATA MINING

Réalisé par : BOUACHAT Anfel

ATCHI imene

Définition :

Clustering ou analyse des clusters est une des méthodes de DATA MINING pour l'analyse des grands ensembles de données. Le clustering gère les taches d'apprentissage non supervisé.

Kmeans est l'un des algorithmes de clustering les plus simples, il sert à partitionner l'ensemble de données en k cluster, spécifié par l'utilisateur. Chaque observation appartient au cluster avec la moyenne plus proche.

Algorithme de Kmeans :

1. Sélectionner K clusters initiaux contenant des objets choisis arbitrairement.
2. Calculer les centres des clusters.
3. Calculer une liste des distances inter clusters et la trier dans l'ordre croissant
4. Générer une nouvelle partition en affectant chaque objet au cluster du centre le plus proche.
5. Calculer les nouveaux centres des clusters.
6. Répéter 3 et 5 jusqu'à ce que les objets se stabilisent dans leurs clusters.

Les avantages de l'algorithme Kmeans :

1. Relativement simple à implémenter
2. Facile à comprendre
3. Flexible, c'est-à-dire applicable sur tout type d'ensemble de données
4. Complexité algorithmique relativement faible

Les inconvénients de l'algorithme Kmeans :

1. Le choix de k (nombre de clusters) est important
2. S'applique sur des données numériques
3. La difficulté de trouver une bonne fonction de distance pour les données non numériques

Les maths derrière le clustering :

Kmeans utilise la distance euclidienne entre les centres et les points. La distance euclidienne est donnée par la loi dessous

$$d(p,q) = \sqrt{(q1-p1)^2 + (q2-p2)^2}$$

Pour assigner chaque point à un cluster, il faut calculer la distance minimale entre le point et les centres.

$$\min[dist(C_i,x)^2\ C_i \epsilon\ C]$$

Tel que C : c'est l'ensemble des centres

C_i : I ème centroid

X : l'ensemble de données

Et dist() : c'est la fonction qui calcule la distance euclidienne

Après le calcul des nouveaux clusters, il faut calculer les nouvelles valeurs des centres avec l'équation suivante :

$$CG = (\frac{1}{|S_1|}\sum_{x_1 \in S_1} x_1\ , \dots, \frac{1}{|S_n|}\sum_{x_n \in S_n}\ )$$

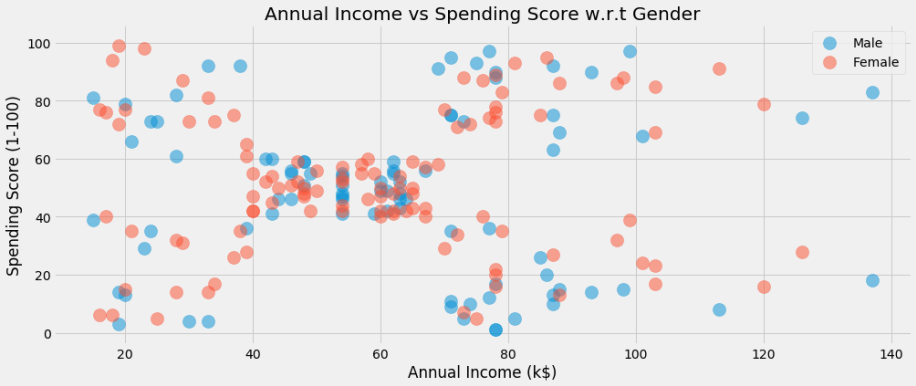
Tel que n : c'est le nombre de centres

S_i c'est l'ensemble des points assigné au i éme cluster

Jeu de données utilisé :

L'ensemble de données utilisé pour l'implémentation de cet algorithme est un fichier csv contenant les informations basiques sur les clients d'un mal [1], le fichier contient les informations (features) des 200 clients différents (samples).

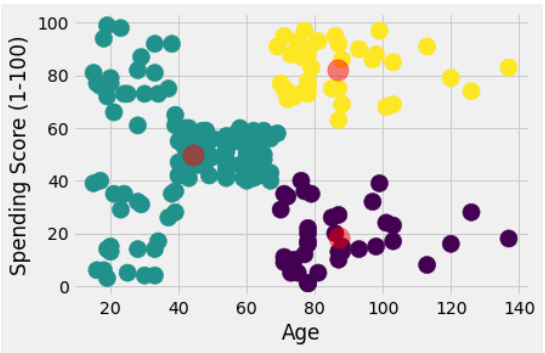
Le graphe dessous montre la distribution du score des dépenses en fonction des revenus annuels.

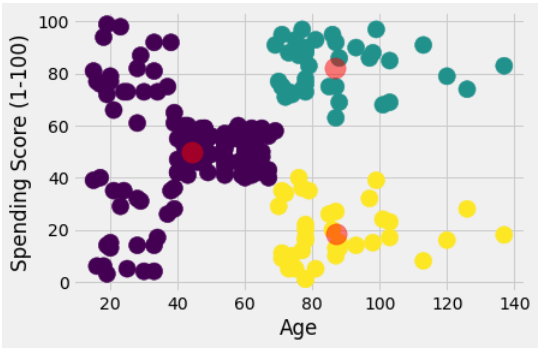


Le jeu de données contient une variable Age qui est de type non numérique, dans le cas où on veut appliquer l'algorithme sur cette variable, on doit la transformer en type numérique, c'est pourquoi on a utilisé la fonction LabelEncode qui est prédéfinit sur la librairie SKlearn pour associer le code 1 pour les mâles et 0 pour les femelles.

Nous avons utilisé 8 procédures pour le calcul de distance, création des clusters, calcul du centre plus proche, vérification de convergence, calcul de l'inertie et l'affichage.

Pour les tests, on a initialisé le k a 3 et le nombre d'itérations à 300, et puis on a appliqué l'algorithme vu en cours.





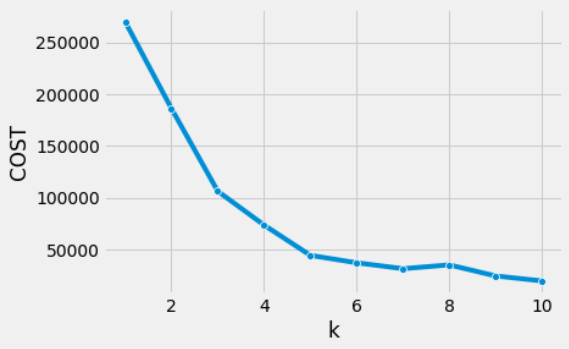
Les deux figures représentent les 3 cluster de l'ensemble de données, la figure à gauche c'est le résultat obtenu avec l'algorithme écrit avec python from

scratch, celui qui est à droite est le résultat obtenu avec l'algorithme définit déjà dans la librairie SKLearn.

L'une des étapes importantes de KMeans Clustering est de déterminer le nombre optimal des clusters que nous devons donner en entrée pour notre algorithme.

Cela peut être fait en visualisant l'ensemble de donnée sur un graphe, on peut déterminer le nombre de cluster juste en regardant le graphe. On peut aussi itérer à travers un certain nombre de valeurs n, on calcul la somme des distances entre les observations et les points du centre plus proche à chaque exemple, puis on trouvera la valeur n optimale qui corresponds à la valeur du n ou la somme des distances calculé est petite. Cette méthode est appelée Elbow méthode.

Nous avons développé cette méthode.



Les deux graphes représentent la somme des distances pour chaque n. le graphe à gauche et fait from scratch, et à droite est réalisé avec l'algorithme de SKLearn.

On remarque bien que c'est la valeur 5 qui est la plus intéressante pour l'ensemble de données qu'on travaille avec. Voici le graphe avec 5 clusters.