Université des Sciences et de la Technologie
Houari Boumediene

Fouille de données

Rapport du TP N°2

Binôme:

HADDAD Sarah 191931066384

EL-FECIH Sarah 191931066236

Mai 2023

Introduction :

K-means est une technique puissante d'apprentissage non supervisé pour extraire des informations à partir de données non étiquetées. Son importance ne cesse de croître à mesure que la quantité de données disponibles augmente. Cependant, le choix du nombre optimal de clusters (K) est l'un des principaux défis de cette méthode. Il est essentiel de sélectionner judicieusement K en utilisant des méthodes telles que l'indice de Calinski-Harabasz ou la méthode du coude pour éviter une segmentation sous-optimale ou sur-optimale des données. Le choix de la méthode de clustering dépendra de la taille et de la complexité des données à traiter, ainsi que des objectifs de l'analyse.

Dans ce rapport, nous examinerons deux méthodes différentes qui utilisent le critère du rapport de variance (l'indice de Calinski-Harabasz) pour automatiser le nombre de clusters dans K-means. Nous décrirons d'abord chacune des méthodes en expliquant leurs avantages et leurs inconvénients.

1-l'indice de Calinski-Harabasz:

Qu'est-ce que l'indice de Calinski-Harabasz ?

L'indice de Calinski-Harabasz (également connu sous le nom de critère du ratio de la variance) est calculé comme le rapport entre la somme de la dispersion inter-cluster et la somme de la dispersion intra-cluster pour tous les clusters (où la dispersion est la somme des distances au carré).

Comment interpréter l'indice de Calinski-Harabasz ?

Un indice Calinski-Harabasz élevé signifie un meilleur regroupement car les observations dans chaque cluster sont plus proches les unes des autres (plus denses), tandis que les clusters eux-mêmes sont plus éloignés les uns des autres (bien séparés).

Algorithme de l'indice de Calinski-Harabasz :

Entrées :
Un ensemble de données X
Le nombre de clusters K

Sortie :
La valeur de l'indice de Calinski-Harabasz CH

L'Algorithme :

Calcule le centre de gravité du jeu de données X, noté C.
Initialise la variable CH à 0.
Pour chaque cluster k allant de 1 à K :
a. Calcule le centre de gravité du cluster k, noté Ck.
b. Calcule la distance entre Ck et C, notée $||Ck-C||$.
c. Calcule le nombre d'observations dans le cluster k, noté nk.
d. Calcule la somme des carrés des distances entre les observations du cluster k et Ck, notée Wk.
e. Calcule le produit nk * $||Ck-C||$^2 et ajoutez-le à la somme BGSS.
f. Ajouter Wk à la somme WGSS.
Calcule CH en utilisant la formule suivante :
CH = (BGSS / WGSS) * ((N - K) / (K - 1))
Retourne la valeur de CH.

2-l'indice de Calinski-Harabasz et Clustering hiérarchique Ascendant (CAH):

1.Algorithme de l'indice de Calinski-Harabasz et Clustering hiérarchique Ascendant pour trouver le meilleur nombre de clusters:

Cet algorithme utilise une approche ascendante pour le clustering hiérarchique et l'indice de Calinski-Harabasz pour évaluer la qualité des solutions de clustering pour différents nombres de clusters, pour trouver le meilleur nombre de clusters:

Entrée: Données à clusteriser
Sortie: Le nombre optimal de clusters

1: Placer chaque objet dans son propre cluster ;
2: Tant qu'il y a des objets à agglomérer
2.1: Calculer une liste des distances entre les clusters et la trier dans l'ordre croissant ;
2.2: Agglomérer les objets ayant la distance minimale;
2.3: Calculer le centre du nouveau cluster;
2.4: Calculer l'indice de Calinski-Harabasz pour la solution de clustering, en utilisant la formule CH=($BGSS$/$WGSS$)*((N-K)/(K-1)), où $BGSS$ est la somme des carrés entre les groupes et $WGSS$ est la somme des carrés intra-groupes pour le nombre de clusters K donné;
2.5: Stocker l'indice de Calinski-Harabasz pour le nombre de clusters K;
Fait;
3: Retourner le nombre de clusters K qui a la plus grande valeur de CH;

2.Implémentation de l'algorithme:

Nous avons choisi Python pour l'implémentation de l'algorithme car c'est un langage de programmation qui dispose d'une grande variété de bibliothèques et d'outils pour le traitement des données.

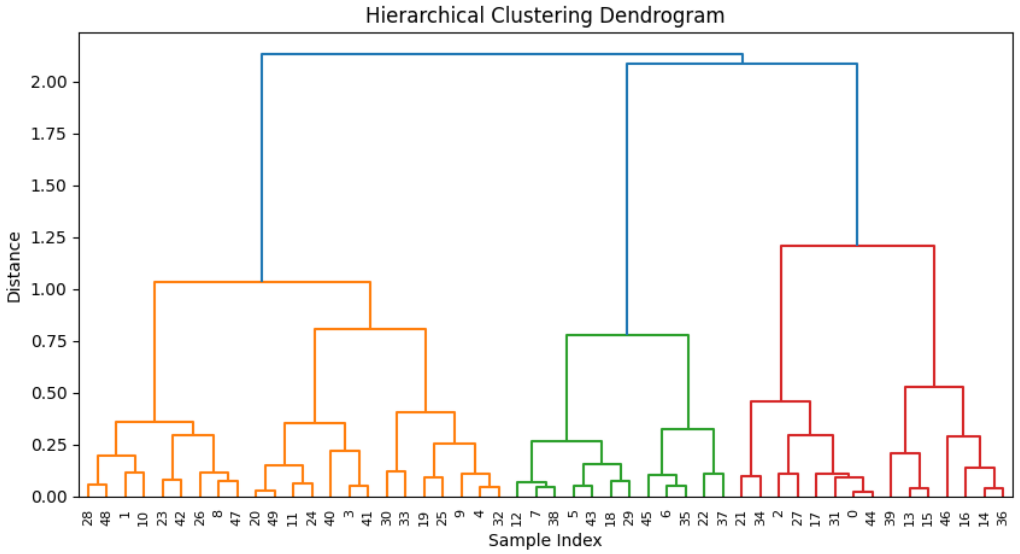
Nous avons utilisé les bibliothèques numpy, matplotlib et scikit-learn pour charger les données, prétraiter les données, effectuer le clustering et visualiser les résultats. Nous avons également écrit une fonction pour calculer le score de Calinski-Harabasz, qui mesure la qualité du clustering.

Nous avons appliqué l'algorithme de clustering agglomératif à un ensemble de données pour déterminer le nombre optimal de clusters.

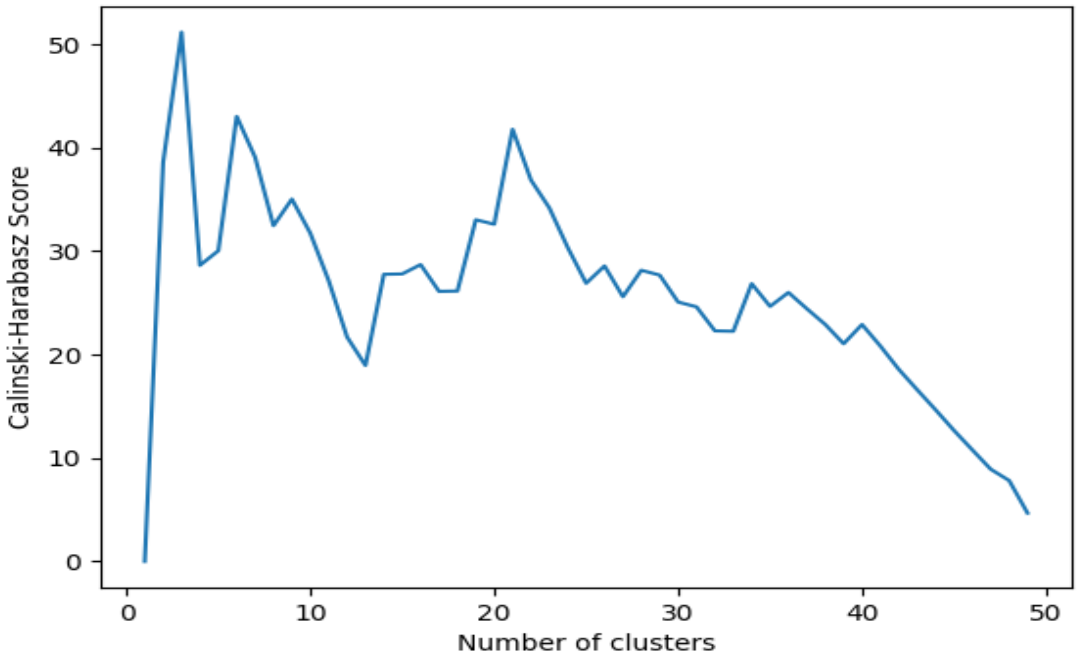
Nous avons ensuite appliqué l'algorithme K-means sur ces données en utilisant le nombre optimal de clusters déterminé précédemment.

La bibliothèque matplotlib a été utilisée pour visualiser le dendrogramme et la courbe du score de Calinski-Harabasz pour chaque nombre de clusters, ainsi que pour afficher le graphique des clusters obtenus à partir de KMeans.

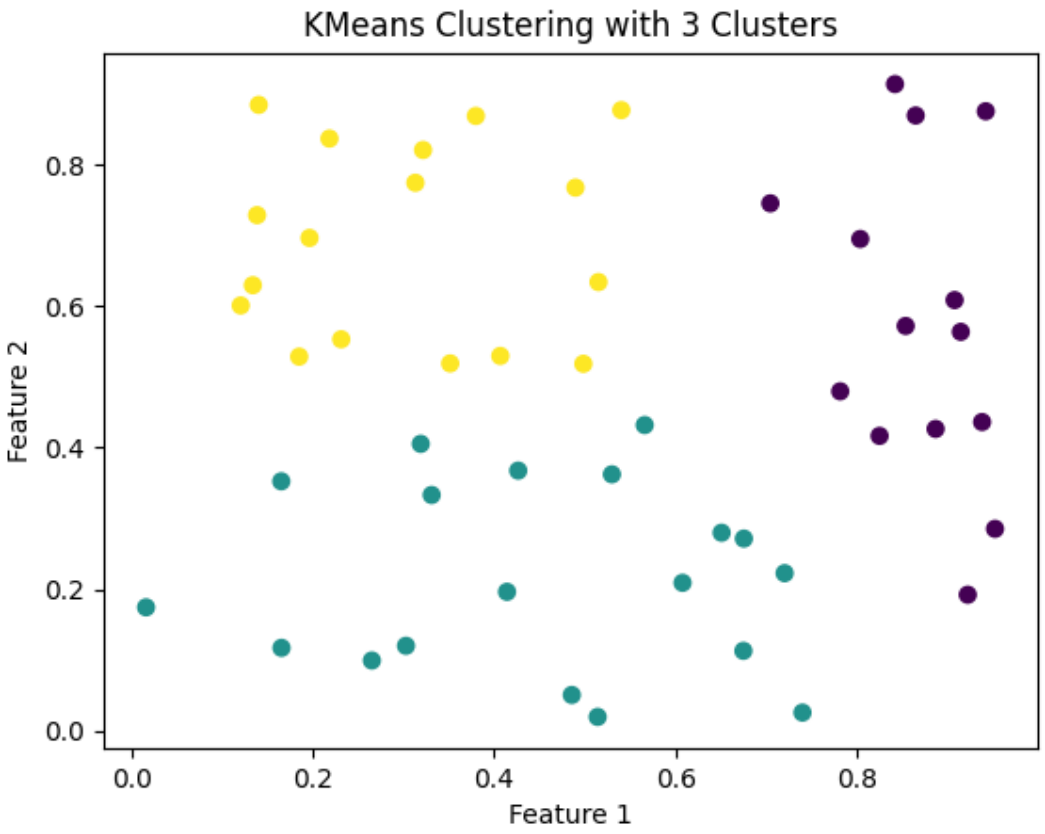
Le dendrogramme:



La courbe du score de Calinski-Harabasz pour chaque nombre de clusters:



Le graphique des clusters obtenus à partir de KMeans (avec le meilleur K trouvé):



3.Avantages:

1.L'indice de Calinski-Harabasz fournit une mesure quantitative de la qualité de la solution de clustering, en utilisant la variance entre les groupes et la variance intra-groupes.
2.Le CAH est une méthode exploratoire qui ne nécessite pas de spécification a priori du nombre de clusters.
3.Le CAH peut être utilisé avec différents types de distance et de lien pour s'adapter aux caractéristiques des données.

4.Inconvénients:

1.Le CAH est sensible aux valeurs aberrantes et aux données bruyantes.
3.Le CAH est sensible aux choix de la distance et du lien, qui peuvent influencer la forme et la structure des clusters.
2.Le CAH est calculairement intensif pour les grandes quantités de données.
L'indice de Calinski-Harabasz ne mesure que la compacité et la séparation des clusters, mais ne fournit pas d'informations sur la forme et la taille des clusters.

3-l'indice de Calinski-Harabasz et K-means:

1.Algorithme de l'indice de Calinski-Harabasz et K-means pour trouver le meilleur nombre de clusters:

Voici les étapes de l'algorithme de l'indice de Calinski-Harabasz et K-means pour trouver le meilleur nombre de clusters :

Entrée : Données à clusteriser, Kmax, Kmin
Sortie : Le nombre optimal de clusters

Pour chaque nombre de clusters K dans une plage donnée(de Kmin à Kmax) :
1. Calculer l'indice de Calinski-Harabasz pour la solution de clustering, en utilisant la formule CH=($BGSS$/$WGSS$)*((N-K)/(K-1)), où $BGSS$ est la somme des carrés entre les groupes et $WGSS$ est la somme des carrés intra-groupes pour le nombre de clusters K donné ;
2. Stocker l'indice de Calinski-Harabasz pour le nombre de clusters K ;
3.Retourner le nombre de clusters K qui a la plus grande valeur de CH.

2.Implémentation de l'algorithme:

Cette implémentation utilise Python pour appliquer l'algorithme de K-means et l'indice de Calinski-Harabasz afin de trouver le nombre optimal de clusters dans un ensemble de données.

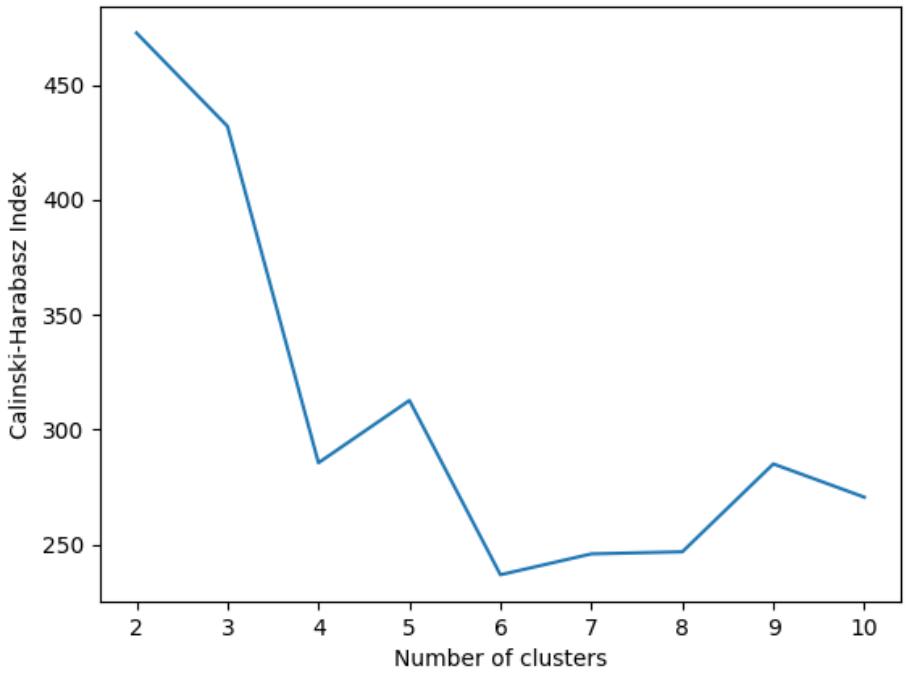
La bibliothèque scikit-learn est utilisée pour la mise en œuvre de l'algorithme K-means et pour la normalisation des données. La bibliothèque numpy est utilisée pour les opérations mathématiques et matplotlib pour la visualisation des données.

Une fonction calinski_harabasz_score a été implémentée pour calculer l'indice de Calinski-Harabasz. Une fonction find_optimal_k a été implémentée pour déterminer le nombre optimal de clusters en itérant sur un intervalle donné et en stockant les indices de Calinski-Harabasz pour chaque solution de clustering. Cette fonction retourne l'indice de Calinski-Harabasz et le nombre optimal de clusters (K, le nombre de clusters qui a la plus grande valeur de l'indice de Calinski-Harabasz).

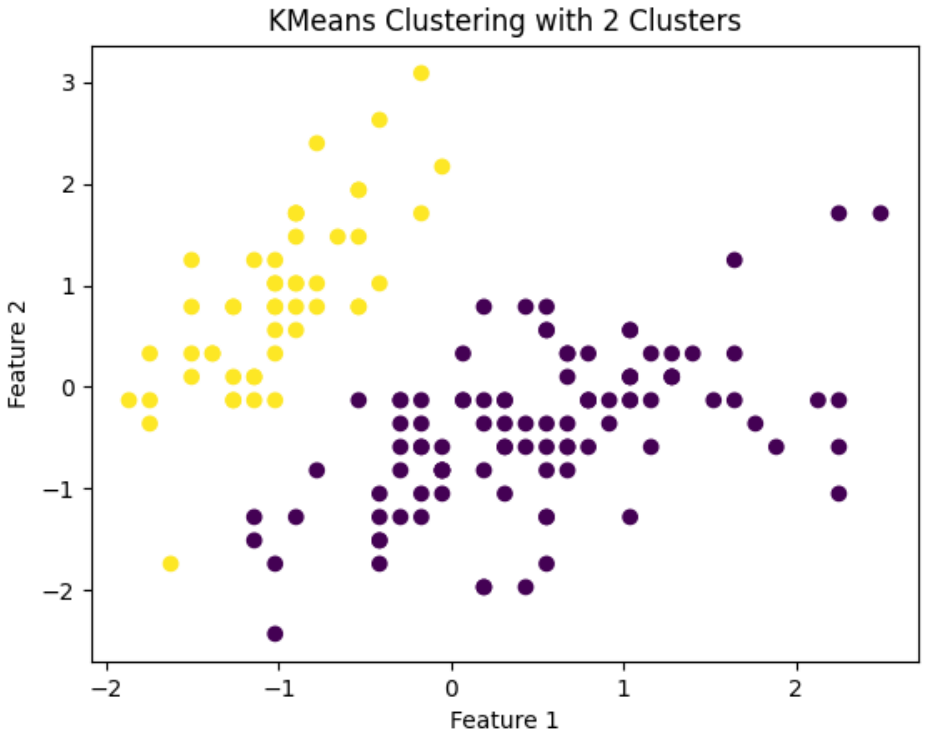
Le code réduit les dimensions de l'ensemble de données à l'aide de l'analyse en composantes principales (PCA) pour diminuer le temps d'exécution de l'itération de Kmin jusqu'à Kmax.

Après avoir déterminé le nombre optimal de clusters, l'algorithme K-means est appliqué avec ce nombre de clusters pour obtenir une solution de clustering. La solution est ensuite visualisée à l'aide de matplotlib.

La courbe du score de Calinski-Harabasz pour chaque nombre de clusters:



Le graphique des clusters obtenus à partir de KMeans (avec le meilleur K trouvé:



3.Avantages:

1.Le nombre de clusters donnés est toujours efficace.
2.Facile à mettre en œuvre
3.La méthode Calinski-Harabasz fournit un score quantitatif pour évaluer la qualité de la solution de clustering, ce qui facilite l'interprétation des résultats
4.L'algorithme K-means est adaptable à différents types de données et de domaines d'application.
5.L'ACP réduit la dimensionnalité des données en conservant les principales composantes, ce qui peut améliorer les performances de l'algorithme K-means et faciliter l'interprétation des résultats.

4.Inconvénients:

1.Pour identifier Kmin et Kmax, l'algorithme utilise la méthode de l'indice de Calinski-Harabasz pour déterminer le nombre optimal de clusters. Il ne nécessite donc pas de choisir manuellement les valeurs de Kmin et Kmax.
3.Le temps d'exécution peut être un inconvénient, en particulier pour des ensembles de données très volumineux.
2.La complexité de l'ACP est en effet de O(D^3) ou de O(D^2), ce qui peut être un inconvénient pour des ensembles de données très volumineux.
4.L'indice de Calinski-Harabasz peut ne pas toujours être efficace pour déterminer le nombre optimal de clusters

3-Comparer les deux méthodes:

Les deux méthodes ont en commun l'utilisation de l'indice de Calinski-Harabasz pour évaluer la qualité des solutions de clustering pour différents nombres de clusters et pour trouver le nombre optimal de clusters.

La différence entre les deux méthodes est la méthode de clustering utilisée pour créer les clusters à chaque étape. La méthode 2 utilise l'algorithme de K-means, qui est une méthode de clustering partitionnelle, tandis que la méthode 1 utilise le clustering hiérarchique ascendant.

Dans la méthode 1, les clusters sont créés en agglomérant les objets ayant la distance minimale à chaque étape, jusqu'à ce qu'un seul cluster contenant tous les objets soit formé. L'indice de Calinski-Harabasz est calculé pour chaque nombre de clusters lors de la création des clusters, et le nombre optimal de clusters est déterminé en sélectionnant celui qui a la plus grande valeur de CH.
La complicité de la méthode 1 est O(N) (N le nombre de données).

La méthode 1 est plus rapide et plus efficace pour les grands ensembles de données et peut être plus robuste aux valeurs initiales des centres de cluster, car elle ne dépend pas de valeurs initiales aléatoires pour créer les clusters.

La méthode 2 est plus lente car elle fait des itérations de Kmin jusqu'à Kmax pour trouver le meilleur k et peut être affectée par la sensibilité aux valeurs initiales des centres de cluster et elle nécessite de donner manuellement les valeurs de kmin et kmax.

Mais dans les tests, les valeurs de k trouvées par la méthode 2 étaient plus intéressantes. La méthode 1 est plus adaptée aux grands ensembles de données.

Conclusion Générale :

En conclusion, nous avons vu que le choix du nombre optimal de clusters est crucial pour obtenir des résultats significatifs avec la méthode de clustering K-means. Les méthodes basées sur l'indice de Calinski-Harabasz se sont révélées très efficaces pour déterminer le nombre optimal de clusters en utilisant le critère de variance inter-classe et de variance intra-classe. Nous avons également examiné deux méthodes différentes, l'une basée sur le clustering hiérarchique ascendant et l'autre sur l'algorithme K-means. Chaque méthode a ses avantages et ses inconvénients, et le choix dépendra de la taille et de la complexité des données ainsi que des objectifs de l'analyse.

En fin de compte, la méthode K-means reste l'une des méthodes les plus populaires pour le clustering non supervisé en raison de sa simplicité, de sa rapidité et de sa facilité d'utilisation. Cependant, il est important de choisir judicieusement le nombre optimal de clusters. En utilisant ces méthodes, nous pouvons tirer le meilleur parti des données non étiquetées pour obtenir des informations utiles et des insights précieux pour une variété de domaines d'application.

Références:

https://www.geeksforgeeks.org/calinski-harabasz-index-cluster-validity-indices-set-3/

https://scikit-learn.org/stable/modules/clustering.html#calinski-harabasz-index

https://pyshark.com/calinski-harabasz-index-for-k-means-clustering-evaluation-using-python/