Clustering Analysis

- *Sanyog Mishra*

Algorithm used: K-means Clustering

Brief Description

K-means clustering is an unsupervised learning algorithm used for data clustering, which groups unlabeled data points into groups or clusters.

It is one of the most popular clustering methods used in machine learning. Unlike supervised learning, the training data that this algorithm uses is unlabeled, meaning that data points do not have a defined classification structure.

While various types of clustering algorithms exist, including exclusive, overlapping, hierarchical and probabilistic, the k-means clustering algorithm is an example of an exclusive or "hard" clustering method. This form of grouping stipulates that a data point can exist in just one cluster. This type of cluster analysis is commonly used in data science for market segmentation, document clustering, image segmentation and image compression. The k-means algorithm is a widely used method in cluster analysis because it is efficient, effective and simple.

K-means is an iterative, centroid-based clustering algorithm that partitions a dataset into similar groups based on the distance between their centroids. The centroid, or cluster center, is either the mean or median of all the points within the cluster depending on the characteristics of the data.

Source: https://www.ibm.com/think/topics/k-means-clustering

Approach 1

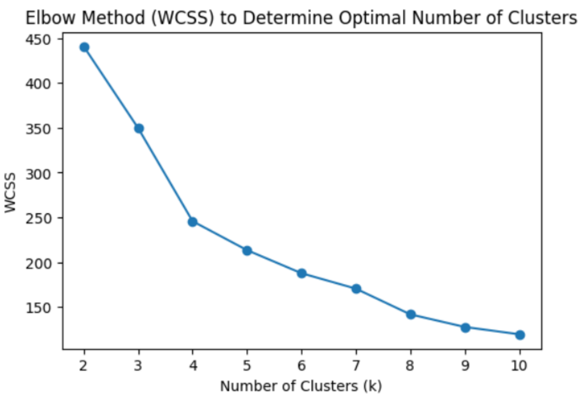
With appropriate feature engineering and selection methodologies, the following features have been selected:

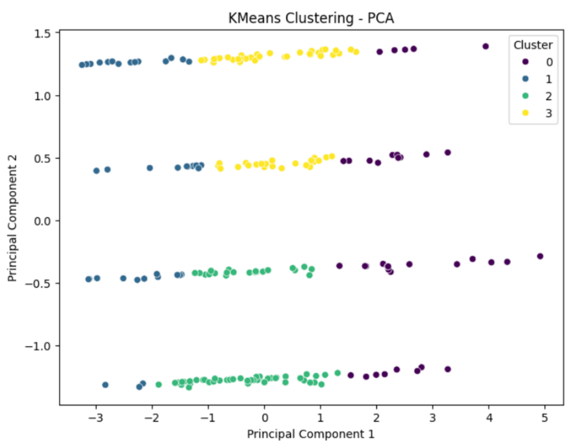
'TotalQuantity': numerical feature. Refers to the total item quantity bought by a customer.

'TotalSpend': numerical feature. Refers to the total amount the customer spent.

'NumTransactions': numerical feature. Refers to the total number of transactions made by a user.

'MostPurchasedCategory': originally a categorical feature but converted to numerical type. Refers to the product category from which the user buys most products from.





For k = 4:

DB Index: 0.9219513061110165 (lower the better)
Silhouette Score: 0.34741013893297057 (higher the better)

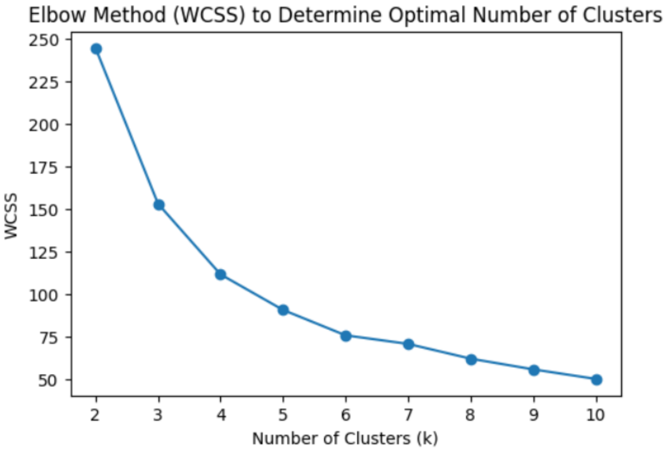
Approach 2

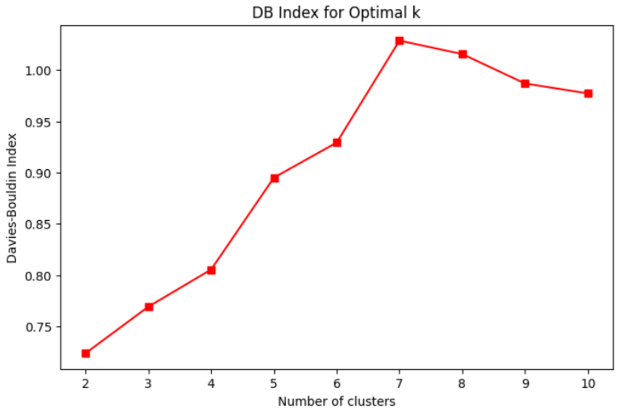
Features considered:

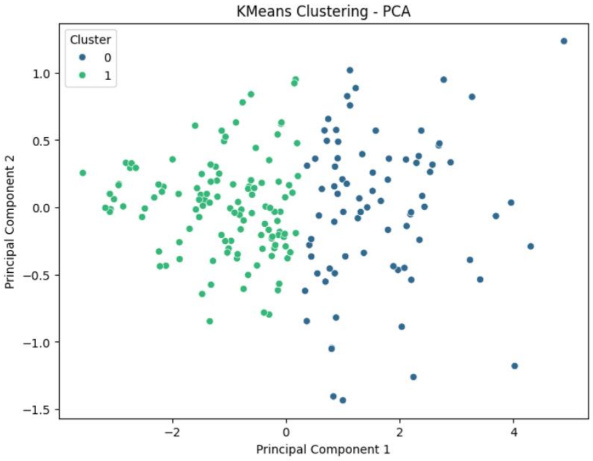
'TotalQuantity': numerical feature. Refers to the total item quantity bought by a customer.

'TotalSpend': numerical feature. Refers to the total amount the customer spent.

'NumTransactions': numerical feature. Refers to the total number of transactions made by a user.

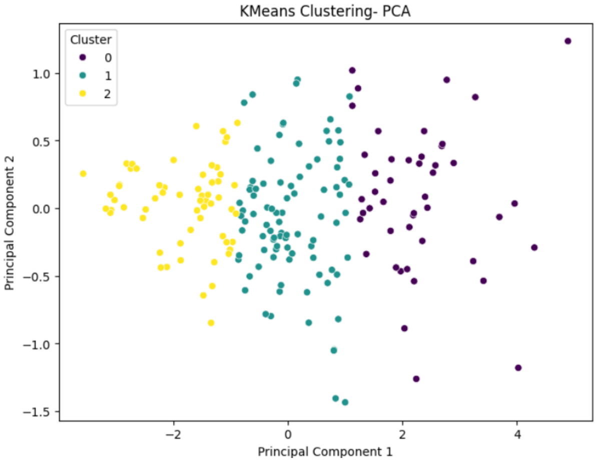






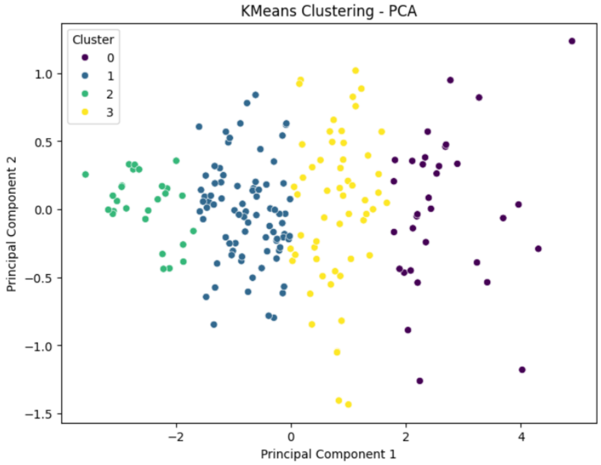
DB Index (k=2): 0.7234787545050064

Silhouette Score (k=2): 0.49373487735321214



DB Index (k=3): 0.7689520825401875

Silhouette Score (k=3): 0.4092392504726072



DB Index (k=4): 0.8052437830269734

Silhouette Score (k=4): 0.39004223332536625

In approach 2, it is observed that for **k = 2**, we get lowest DB Index value, but to incorporate multiple more clusters we can consider values k = 3 or 4 for almost similar DB index scores.

***