Customer Segmentation using K-Means Clustering

Introduction

Client segmentation is a data mining approach that splits a customer base into multiple groups based on characteristics such as gender, age, hobbies, and other buying habits. Customers are split into groups based on shared qualities, in other words

K-Means clustering is an unsupervised machine learning algorithm that divides the given data into the given number of clusters. Here, the "K" is the given number of predefined clusters, that need to be created.

It is a centroid based algorithm in which each cluster is associated with a centroid. The main idea is to reduce the distance between the data points and their respective cluster centroid.

K-Means is very easy and simple to implement. It is highly scalable, can be applied to both small and large datasets. There is, however, a problem with choosing the number of clusters or K. Also, with the increase in dimensions, stability decreases. But, overall K Means is a simple and robust algorithm that makes clustering very easy

Problem Statement

Let's Assume you own the mall and want to understand the customers like who can be easily converge as a target customer. So that the sense can be given to marketing team and plan the strategy to improve the business.

Advantages of Customer Segmentation

1. Determine appropriate product pricing.
2. Develop customized marketing campaigns.
3. Design an optimal distribution strategy.
4. Choose specific product features for deployment.
5. Prioritize new product development efforts.

Libraries used:

1. scikit-learn: Scikit-learn is a free Python machine learning software, sometimes known as sklearn. It is meant to interact with the Python numerical and scientific libraries NumPy and SciPy, and features support vector machines, random forests, gradient boosting, k-means, and DBSCAN, among other classification, regression, and clustering algorithms.

2. Seaborn: Seaborn is a matplotlib-based open-source Python library. It's used for exploratory data analysis and data visualization

3. Numpy: numpy is a package that contains multidimensional array objects and tools for manipulating them. NumPy is a Python library that allows us to perform mathematical and logical operations on arrays

4. Pandas: Python toolkit for data science, data analysis, and machine learning that is open-source. It is based on NumPy, a multi-dimensional arrays-supporting library.

5. Matplotlib: For 2D array charts, Matplotlib is a superb Python visualization library. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the entire SciPy stack.

Dataset Details

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Figure 1. Dataset used

The dataset(as shown in figure 1) contains basic details about the customers like Customer ID, age, gender, annual income and spending score. Spending Score is something you assign to the customer based on your defined parameters like customer behaviour and purchasing data.

Algorithm used: K-means Clustering

As the given data set is unlabelled and objective is to group the customers in to different clusters, we have used K-Means clustering algorithm. The flow chart is given in figure 2.

Algorithm:

Step 1: Select the number K to determine the number of clusters.

Step 2: At random, select K locations or centroids. (It's possible that it's not the same as the incoming dataset.)

Step 3: Form the present K clusters by assigning each data point to the centroid that is closest to it.

Step 4: Calculate the variance and move the centroid of each cluster.

Step 5: Reverse the previous three steps, reassigning each datapoint to the cluster's new closest centroid.

Step-6: Go to step-4 if there is a reassignment; otherwise, go to FINISH.

Step 7: The model is now complete.

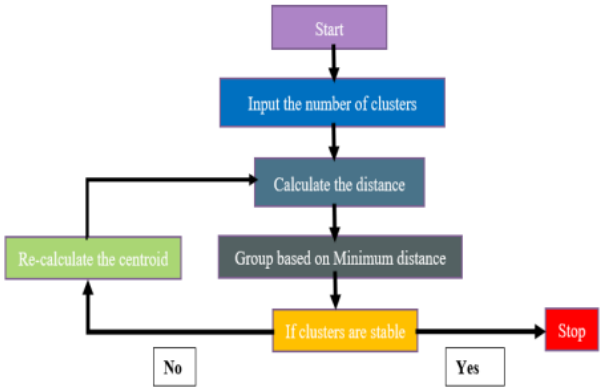

Figure 2. Flowchart of K-means algorithm

The Elbow Method

Calculate the Within Cluster Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS first starts to diminish. In the plot of WSS-versus k, this is visible as an elbow as shown in figure 3.

The steps can be summarized in the below steps:

1. Compute K-Means clustering for different values of K by varying K from 1 to 10 clusters.
2. For each K, calculate the total within-cluster sum of square.
3. Plot the curve of WCSS vs the number of clusters K.
4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

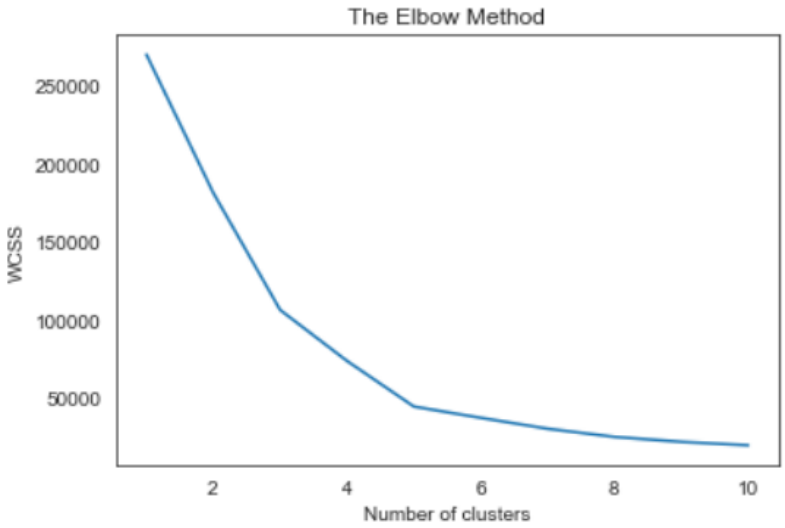

Figure 3. Elbow method graph

Implementation:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.cluster import KMeans

df = pd.read_csv("Mall_Customers.csv")
wcss = []
for k in range(1,11):
    kmeans = KMeans(n_clusters=k, init="k-means++")
    kmeans.fit(df.iloc[:,1:])
    wcss.append(kmeans.inertia_)
km = KMeans(n_clusters=5)
clusters = km.fit_predict(df.iloc[:,1:])
df["label"] = clusters
dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100,c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label =
'Centroids')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Results

we made a box plot of spending score and annual income to better visualize the distribution range. The range of spending score is clearly more than the annual income range as shown in figure 4.

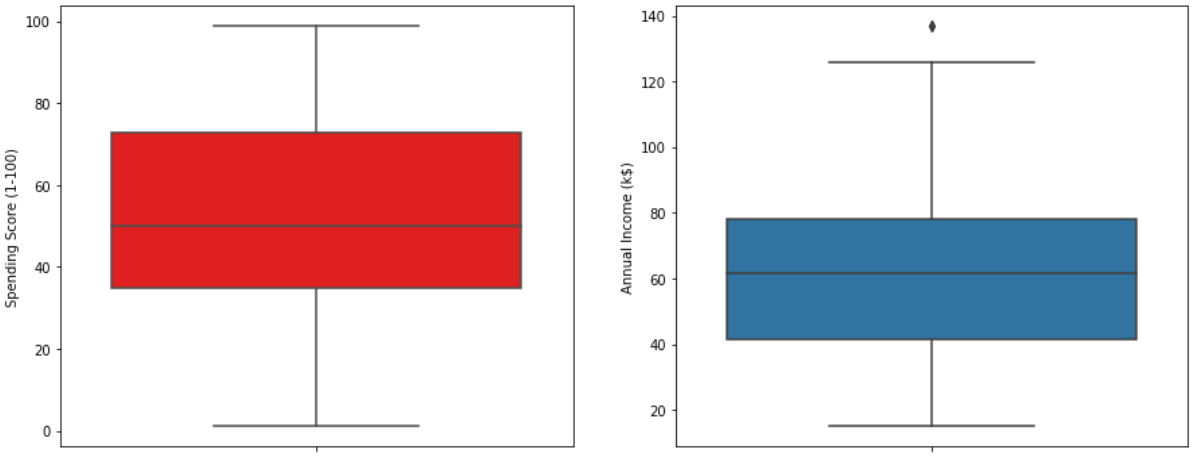

Figure 4. Box Plot

We made a bar plot to check the distribution of male and female population in the dataset. The female population clearly outweighs the male counterpart as shown in figure 5.

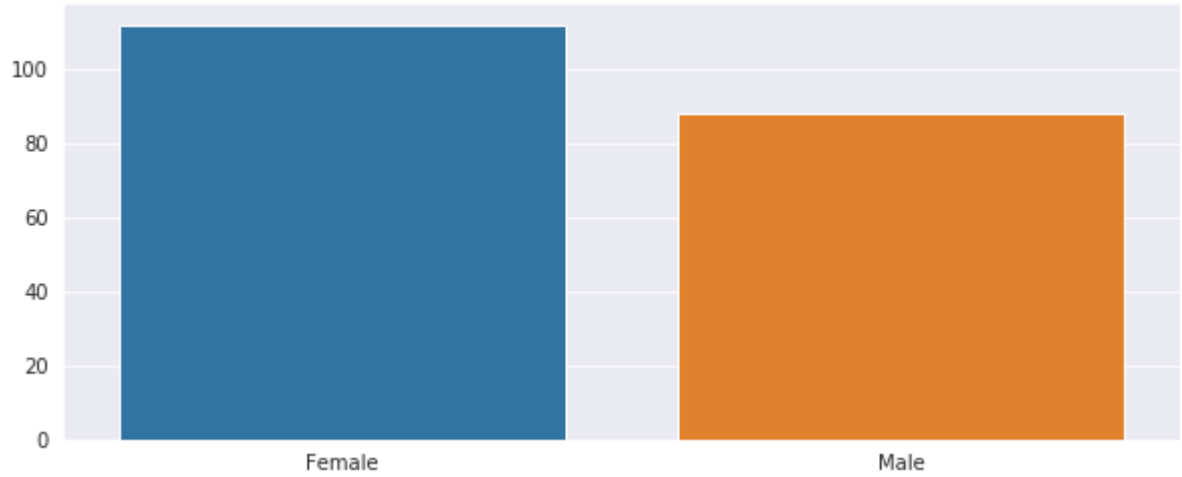

Figure 5. gender bar plot

Next We made a bar plot to check the distribution of number of customers in each age group. Clearly the 26–35 age group outweighs every other age group as shown in figure 6.

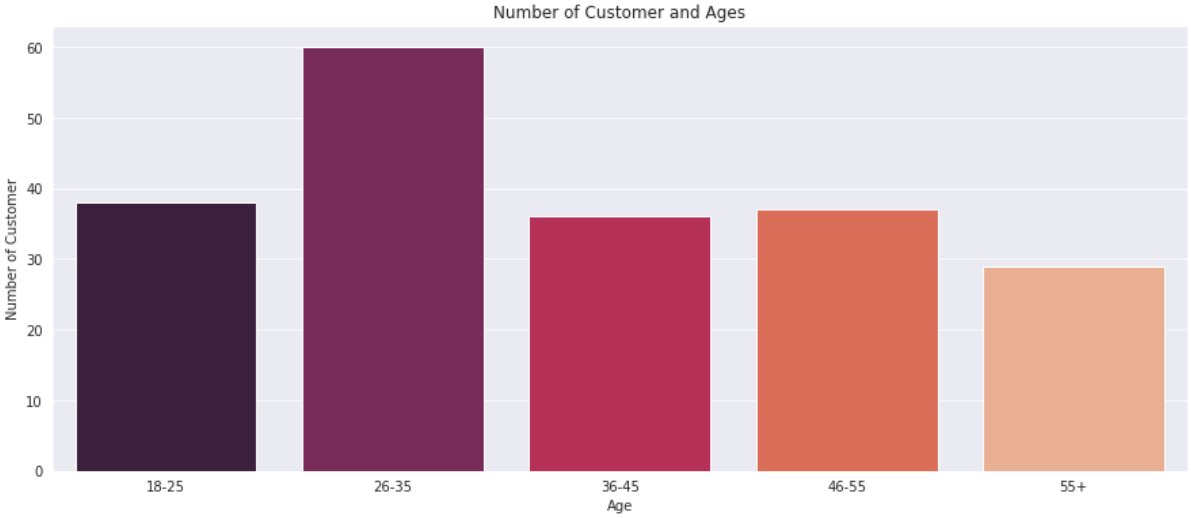

Figure 6. age group bar plot

We made a bar plot to visualize the spreading of spending scores of customers. The highest number of customers are in range of 41-60 as shown in figure 7.

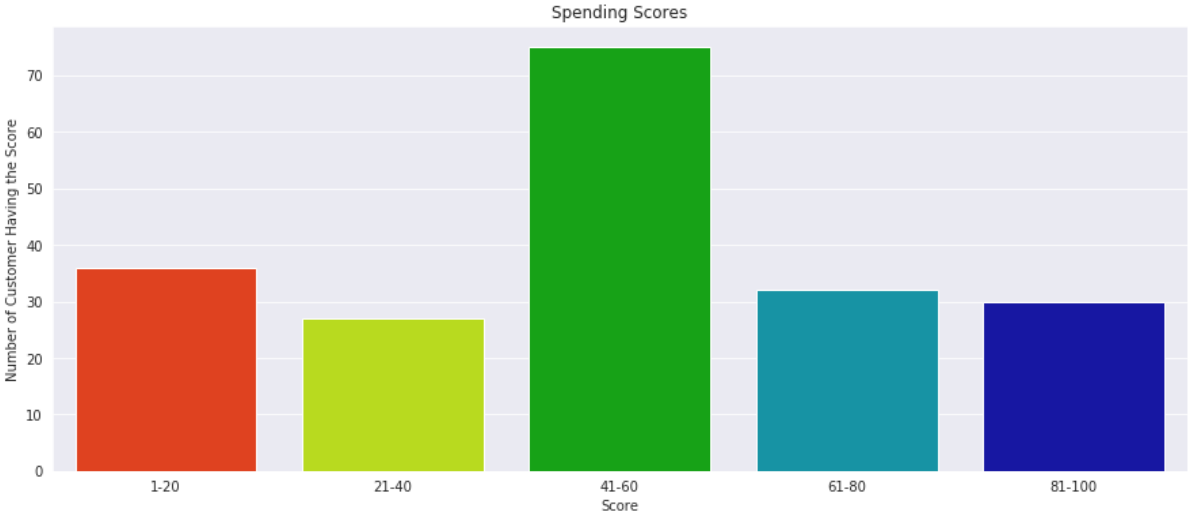

Figure 7. Spending score bar plot

We made a bar plot to visualize the number of customers according to their annual income. The majority of the customers have annual income in the range 60000 and 90000 as shown in figure 8.

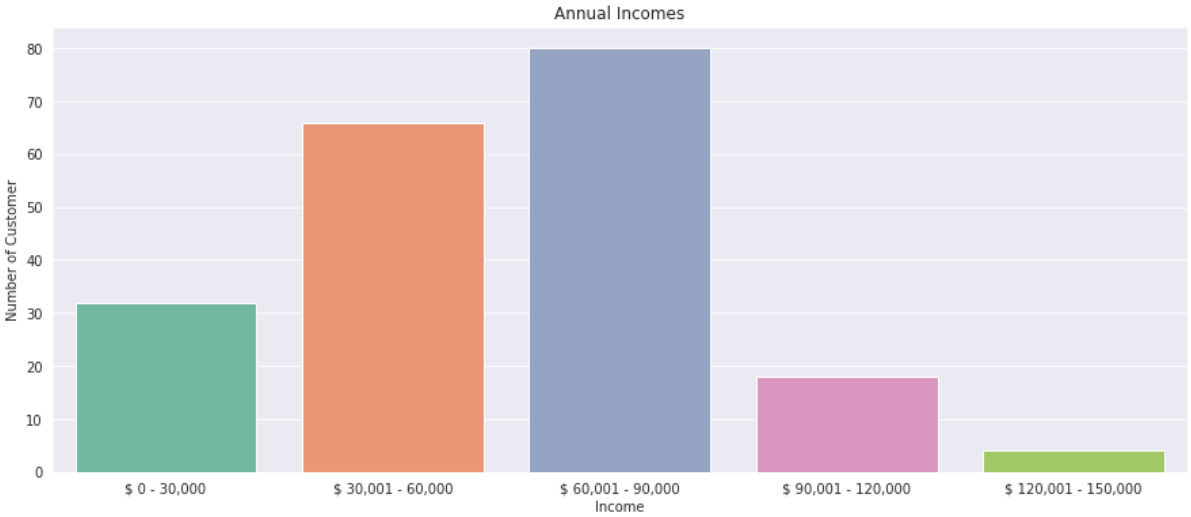

Figure 8. Annual income bar plot

We made a 3D scatter plot to visualize the cluster so that we can make sure that clusters are not overlapping as shown in figure 9.

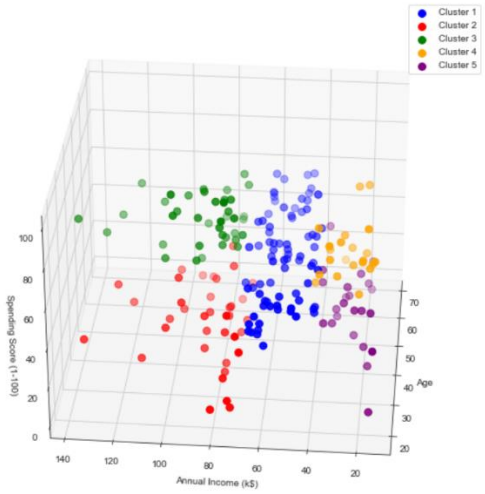

Figure 9. 3D scatter plot

We made a scatter plot with the centroid based on spending score and annual income as shown in figure 10.

Figure 10. Scatter plot with centroid

Conclusions

This study demonstrates that client segmentation in shopping malls is achievable despite the fact that this form of machine learning application is highly useful in the market, a manager can concentrate all of his or her attention on each cluster that has been discovered and meet all of their requirements. Mall managers must be able to understand what customers require and, more importantly, how to meet those needs. analyze their purchasing habits, and establish frequent encounters with customers that make them feel comfortable in order to satisfy their demands.

References

1. https://matplotlib.org/3.5.2/tutorials/introductory/usage.html
2. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
3. https://www.geeksforgeeks.org/introduction-to-seaborn-python/