Customer Segmentation Analysis

Python Data Analysis Project

➢ Introduction :

Customer segmentation is a crucial analytical technique used in marketing to divide a company's customer base into distinct groups or "segments." Each segment is composed of customers with similar characteristics, behaviours, or needs, allowing businesses to tailor their marketing strategies, products, or services to better meet these specific preferences.

This analysis primarily leverages customer attributes such as age, annual income, spending behaviour, interests, and other demographic or behavioural data. By identifying patterns in this data, businesses can better understand their customer base and create personalized strategies to maximize customer satisfaction and profitability.

For this analysis, k-means clustering is commonly used due to its ability to efficiently group customers into clusters based on proximity in a multi-dimensional feature space.

➢ Purpose of Analysis :

The primary goal of customer segmentation is:

- To understand the behaviour and characteristics of different customer groups.
- To enhance marketing efforts by targeting the right audience with relevant messages.
- To identify high-value customers for premium products or services.
- To improve resource allocation by focusing on profitable segments.

This analysis will empower decision-makers to optimize business strategies, improve customer retention, and drive sales growth. The use of the dataset containing attributes like customer age, annual income, and spending score provides a strong foundation for building these actionable insights.

➢ Overview of Data Set :

The Mall_Customers.csv dataset is designed for customer segmentation analysis and contains 200 records with the following attributes:

1. CustomerID :

   A unique identifier for each customer in the dataset. (Numerical)

2. Gender :

   The gender of the customer (Male/Female). (Categorical)

3. Age :

   The age of the customer (in years). (Numerical)

4. Annual Income (k$):

The annual income of the customer, measured in thousands of dollars.(Numerical)

5. Spending Score (1-100):

A score assigned by the mall based on customer spending habits and behaviour. Higher scores indicate higher spending or loyalty.(Numerical)

➢ Importance of the Analysis :

Customer segmentation analysis is essential for businesses aiming to optimize their marketing strategies, resource allocation, and overall customer relationship management. By grouping customers based on similarities in their demographics, behaviours, and spending patterns, businesses can create targeted campaigns that resonate more effectively with specific customer segments. For instance, identifying high-spending, high-income customers allows organizations to focus on premium product offerings, while understanding low-income, high-spending groups can help in designing affordable yet appealing options.

This analysis also helps businesses uncover hidden patterns in customer behaviour, such as preferences, purchasing tendencies, and loyalty factors. It enhances decision-making by providing actionable insights, ensuring that marketing efforts are not only personalized but also cost-efficient. Moreover, segmentation facilitates the identification of underserved or high-potential customer groups, enabling businesses to tap into new revenue streams.

Ultimately, customer segmentation analysis strengthens customer retention and acquisition strategies, improves customer satisfaction, and drives revenue growth. It is a powerful tool for maintaining a competitive edge in an increasingly dynamic market by ensuring that businesses address the unique needs of their diverse customer base with precision and effectiveness.

❖ Import and Load Data set :

[2]:
```
import pandas as pd

# Load the dataset
file_path = 'C:/Users/vinay/OneDrive/Documents/Mall_Customers.csv'
data = pd.read_csv(file_path)

# Display the first few rows
print(data.head())

# Get basic information about the dataset
print(data.info())
```

```
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
None
```

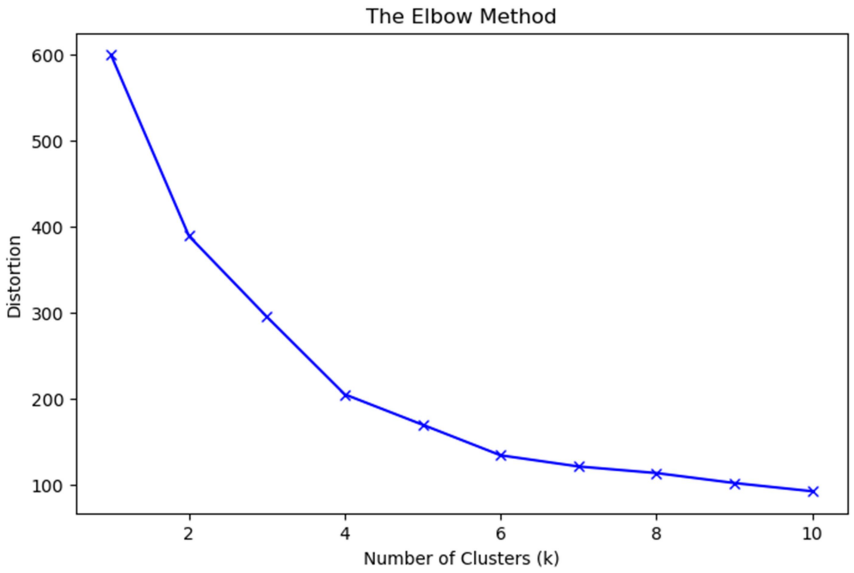
❖ Process the Data :

[3]:
```python
from sklearn.preprocessing import StandardScaler

# Check for missing values
print(data.isnull().sum())

# Select relevant features for segmentation
# Assuming columns include 'Age', 'Annual Income (k$)', and 'Spending Score (1-100)'
features = data[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']]

# Standardize the data
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Convert back to a DataFrame for better readability
scaled_features_df = pd.DataFrame(scaled_features, columns=features.columns)
print(scaled_features_df.head())
```

```
CustomerID              0
Gender                  0
Age                     0
Annual Income (k$)      0
Spending Score (1-100)  0
dtype: int64
        Age  Annual Income (k$)  Spending Score (1-100)
0 -1.424569           -1.738999               -0.434801
1 -1.281035           -1.738999                1.195704
2 -1.352802           -1.700830               -1.715913
3 -1.137502           -1.700830                1.040418
4 -0.563369           -1.662660               -0.395980
```

❖ Determine Optimal Number of Clusters :

[4]:
```python
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Calculate distortions for different k values
distortions = []
K = range(1, 11)  # Test k from 1 to 10
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_features)
    distortions.append(kmeans.inertia_)

# Plot the Elbow curve
plt.figure(figsize=(8, 5))
plt.plot(K, distortions, 'bx-')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Distortion')
plt.title('The Elbow Method')
plt.show()
```



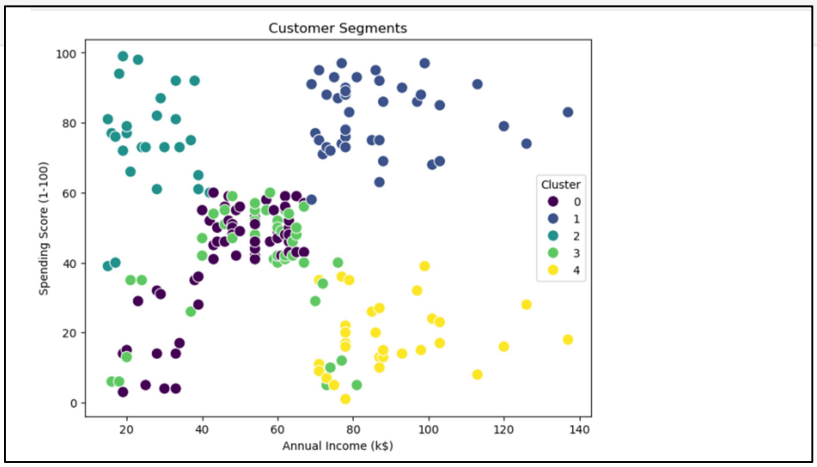
❖ Apply K-Means Clustering :

[5]:
```
# Apply K-Means with the optimal number of clusters
optimal_k = 5  # Replace with the chosen k from the Elbow Method
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
clusters = kmeans.fit_predict(scaled_features)

# Add the cluster labels to the original dataset
data['Cluster'] = clusters
print(data.head())
```

```
   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

   Cluster
0        2
1        2
2        3
3        2
4        2
```

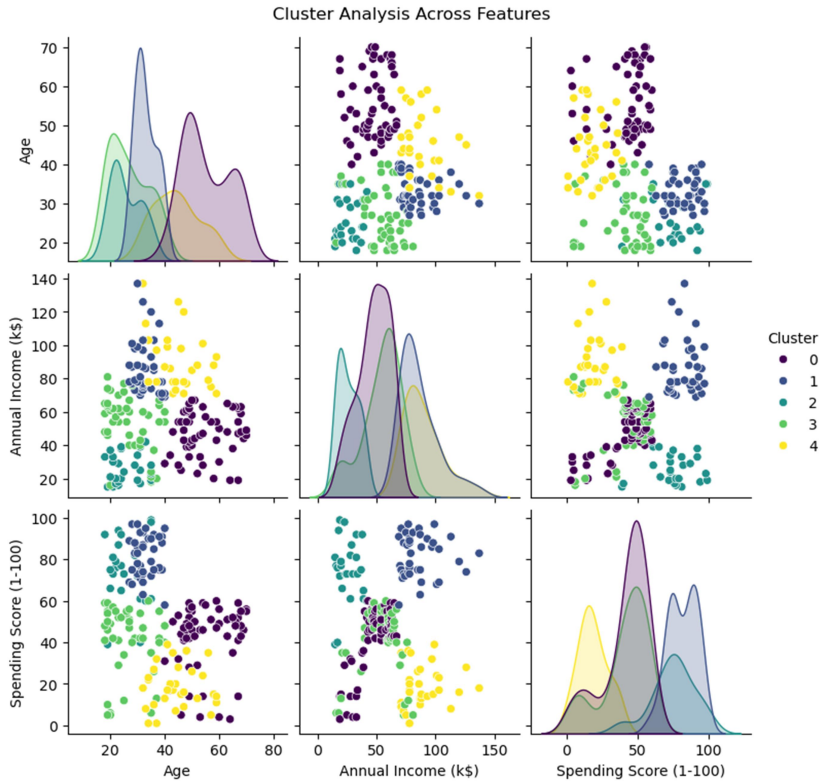
❖ Visualize the Clusters :

[6]:
```
import seaborn as sns

# Scatter plot of clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x=data['Annual Income (k$)'],
                y=data['Spending Score (1-100)'],
                hue=data['Cluster'],
                palette='viridis',
                s=100)
plt.title('Customer Segments')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend(title='Cluster')
plt.show()
```



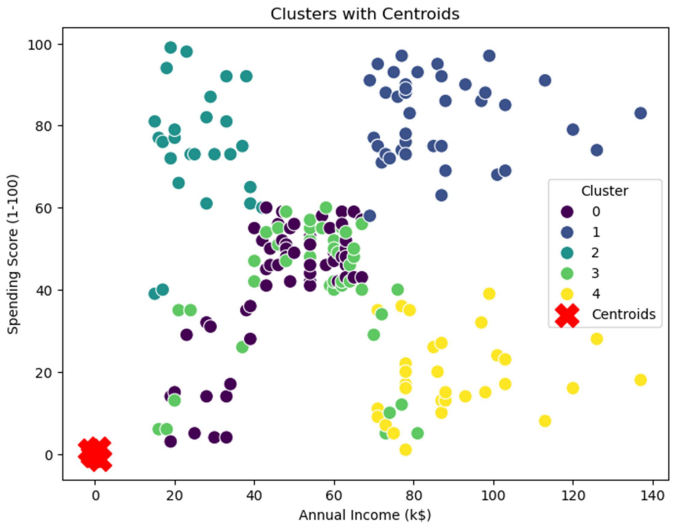
❖ Pairwise Scatterplots for Cluster Analysis :

[7]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Pairplot to visualize clusters
sns.pairplot(data, vars=['Age', 'Annual Income (k$)', 'Spending Score (1-100)'], hue='Cluster', palette='viridis', diag_kind='kde')
plt.suptitle('Cluster Analysis Across Features', y=1.02)
plt.show()
```



❖ Centroid Plot :

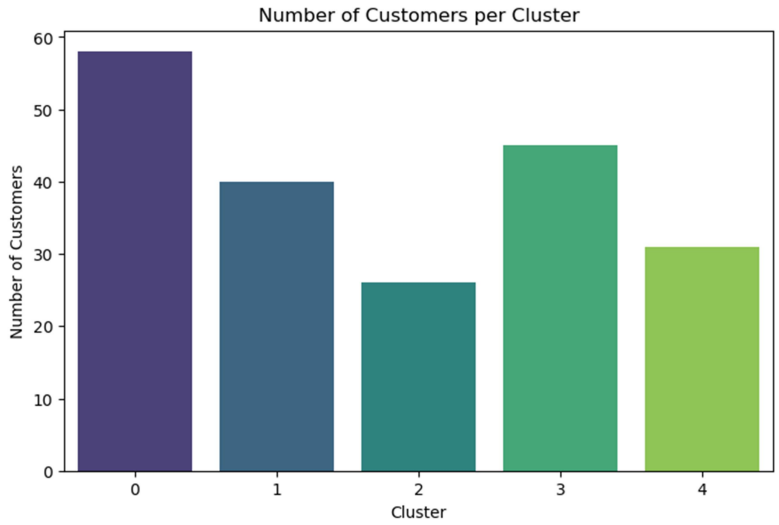
[8]:
```
# Add cluster centroids to the plot
centroids = kmeans.cluster_centers_

# Scatter plot with centroids
plt.figure(figsize=(8, 6))
sns.scatterplot(x=data['Annual Income (k$)'], y=data['Spending Score (1-100)'], hue=data['Cluster'], palette='viridis', s=100)
plt.scatter(centroids[:, 1], centroids[:, 2], s=300, c='red', marker='X', label='Centroids')
plt.title('Clusters with Centroids')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend(title='Cluster')
plt.show()
```



❖ Cluster Sizes :

[9]:
```
# Count the number of customers in each cluster
cluster_counts = data['Cluster'].value_counts()

# Bar plot
plt.figure(figsize=(8, 5))
sns.barplot(x=cluster_counts.index, y=cluster_counts.values, palette='viridis')
plt.title('Number of Customers per Cluster')
plt.xlabel('Cluster')
plt.ylabel('Number of Customers')
plt.show()
```

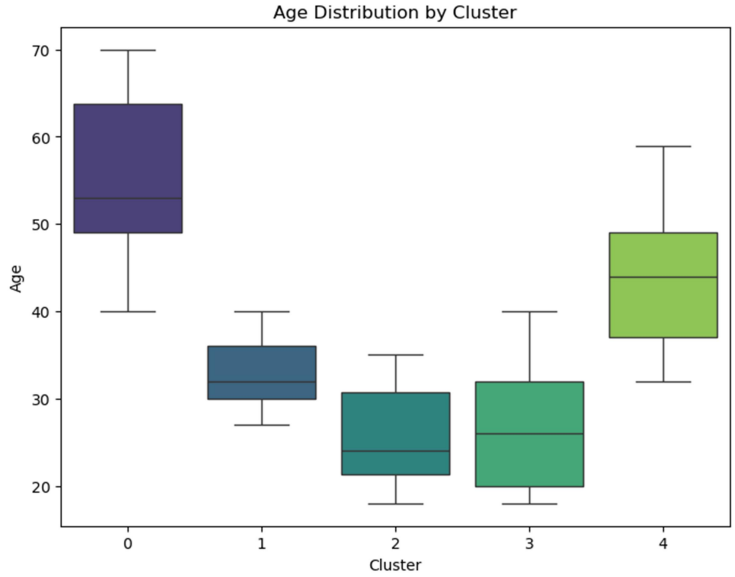
```
C:\Users\vinay\AppData\Local\Temp\ipykernel_12976\918833382.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend
same effect.

  sns.barplot(x=cluster_counts.index, y=cluster_counts.values, palette='viridis')
```



❖ Age Distribution by Cluster :

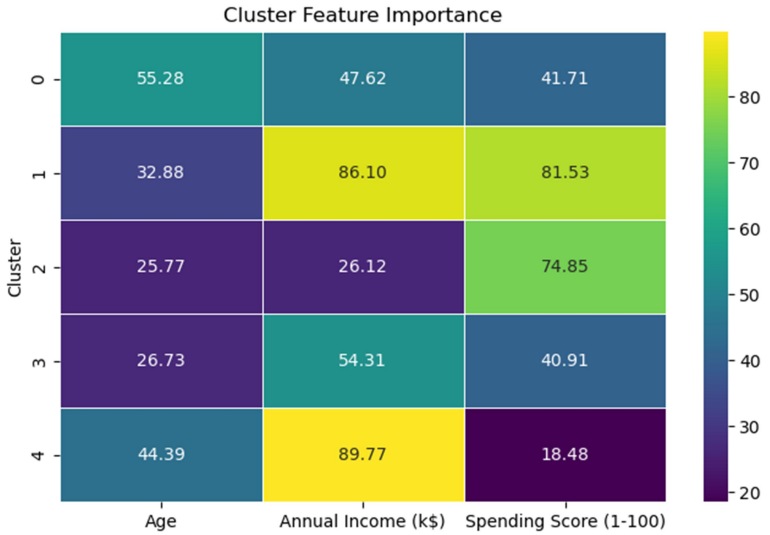
[10]:
```
# Boxplot for Age across clusters
plt.figure(figsize=(8, 6))
sns.boxplot(x='Cluster', y='Age', data=data, palette='viridis')
plt.title('Age Distribution by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Age')
plt.show()
```

```
C:\Users\vinay\AppData\Local\Temp\ipykernel_12976\16375532.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and s
same effect.

  sns.boxplot(x='Cluster', y='Age', data=data, palette='viridis')
```



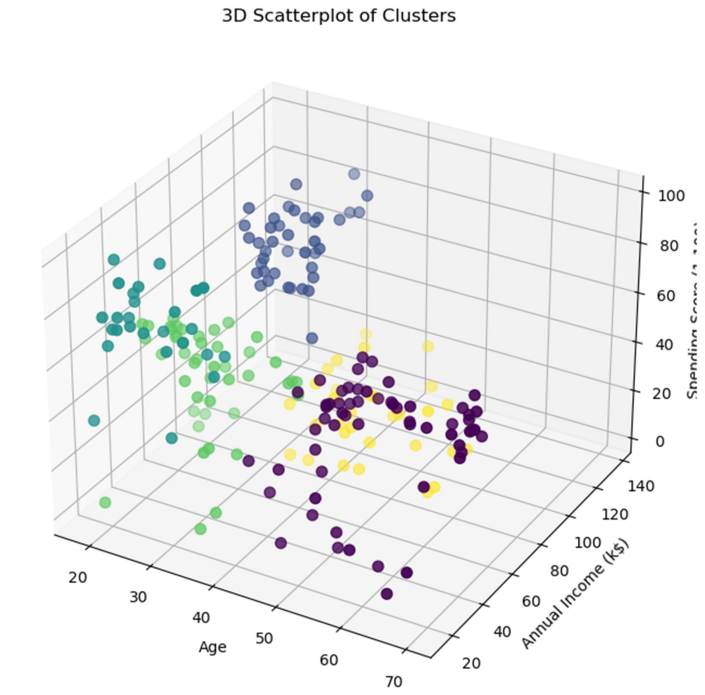
❖ Heatmap for Feature Importance :

[11]:
```
# Calculate mean values of features for each cluster
cluster_means = data.groupby('Cluster')[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']].mean()

# Heatmap
plt.figure(figsize=(8, 5))
sns.heatmap(cluster_means, annot=True, cmap='viridis', fmt='.2f', linewidths=0.5)
plt.title('Cluster Feature Importance')
plt.show()
```



❖ 3D Scatterplot for Better Visualization :

[13]:
```
from mpl_toolkits.mplot3d import Axes3D

# 3D Scatterplot
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(data['Age'], data['Annual Income (k$)'], data['Spending Score (1-100)'],
           c=data['Cluster'], cmap='viridis', s=50)

ax.set_title('3D Scatterplot of Clusters')
ax.set_xlabel('Age')
ax.set_ylabel('Annual Income (k$)')
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```



❖ Insights :

The analysis reveals distinct customer segments based on their demographics and spending behaviour. For example, one cluster might represent younger customers with moderate incomes and high spending scores, indicating they prioritize spending on lifestyle or leisure activities. This group can be targeted with premium product offerings and loyalty programs to maintain their interest and spending habits.

Another segment could include customers with high annual incomes but low spending scores, suggesting a cautious spending pattern despite their financial capacity. These individuals may be motivated with value-driven promotions, exclusive discounts, or tailored marketing campaigns to convert them into higher-spending customers. Similarly, a segment of low-income, high-spending customers might highlight impulsive buyers who could benefit from affordable but attractive offers.

Overall, the segmentation helps businesses identify key traits for effective marketing strategies, such as targeting high-value customers, retaining loyal spenders, and attracting cautious buyers. These insights can lead to personalized campaigns that improve customer satisfaction and boost profitability, while also guiding resource allocation to focus on profitable customer groups.

Summary

The customer segmentation analysis of the Mall_Customers.csv dataset provides valuable insights into different customer groups based on their demographic characteristics and spending behaviours. By utilizing k-means clustering, the analysis identifies distinct segments, such as younger customers with lower incomes but high spending scores, suggesting a strong inclination to purchase despite limited financial resources. This group may benefit from targeted, affordable products or loyalty programs that encourage sustained engagement. On the other hand, high-income, low-spending individuals highlight a different opportunity—tailored marketing efforts aimed at this group could include premium offers, exclusive deals, or incentives to increase their spending behaviour. The analysis also uncovers low-income, high-spending customers, emphasizing a more impulsive or emotion-driven buying tendency. Businesses can leverage this segment by offering trendy or impulse-purchase products at a value-driven price. Moreover, clustering allows the identification of clusters that might need more personalized strategies to improve customer loyalty, either by emphasizing cost-effective solutions or high-value incentives. By understanding these nuanced customer

behaviours, businesses can create specific marketing strategies that cater to each segment's preferences, ultimately enhancing customer satisfaction, driving sales growth, and improving customer retention. These findings demonstrate the power of segmentation in maximizing marketing efforts by ensuring that campaigns are aligned with the needs and behaviours of distinct customer groups. This targeted approach improves resource allocation and ensures that businesses can effectively meet their customers where they are, with the right message at the right time.

Vinay Upadhyay | LinkedIn