Personal Project

Mall Customer Segmentation

Vuong Huy Hoang

Table of contents

Business Understanding

As the owner of a supermarket mall, you have customer data from membership cards, including Customer ID, age, gender, annual income, and spending score. Analyzing this data helps identify target customers by segmenting them based on demographics and purchasing behavior. This insight allows you to tailor marketing strategies to the most profitable customer groups, design focused advertising campaigns, and create personalized promotions. By doing so, your marketing team can enhance customer engagement, boost sales, and foster loyalty, driving overall growth and success for the mall.

About the dataset

This dataset is composed by the following 5 features:

CustomerID: Unique ID assigned to the customer
Gender: Gender of the customer
Age: Age of the customer
Annual Income (**k$**): Annual Income of the customer
Spending Score (**1-100**): Score assigned by the mall based on customer behavior and spending nature.
In this particular dataset we have **200 samples** to study.
All information and data related to this problem can be found here: Mall Customer Segmentation Data

Data Exploration

```
df = pd.read_csv('Mall_Customers.csv')
df.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

```
df.isnull().sum()
```

```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Checking for null values, object data types and other things we might consider in order to know well about the dataset and keep our data clean.

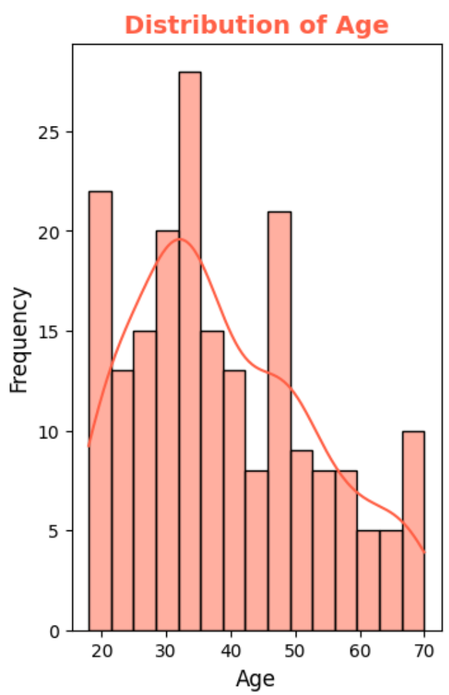
```
df.info()
```

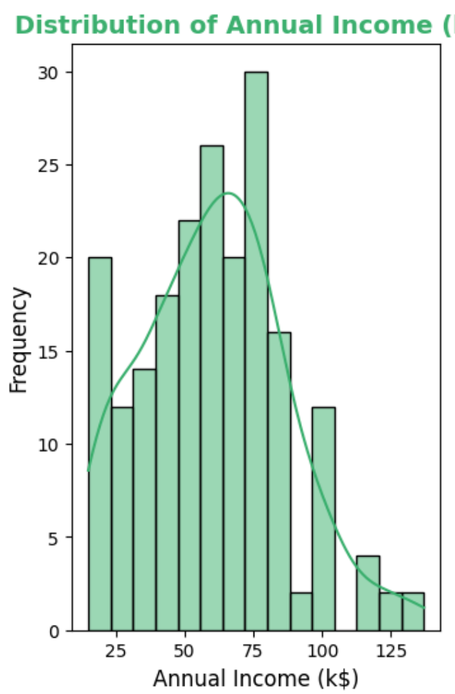
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
```

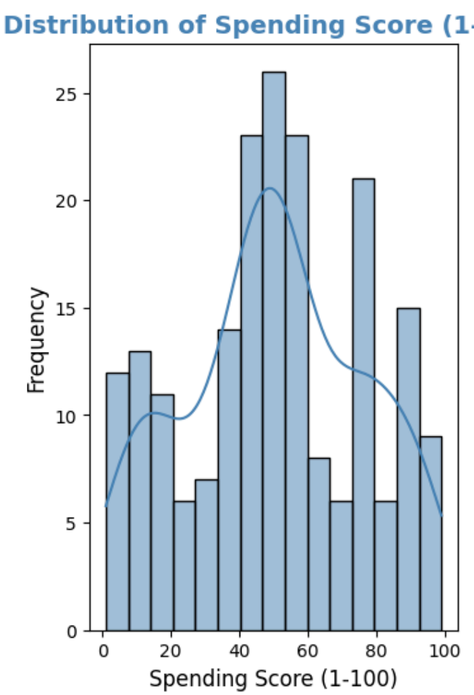
```
df.describe()
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Data Exploration

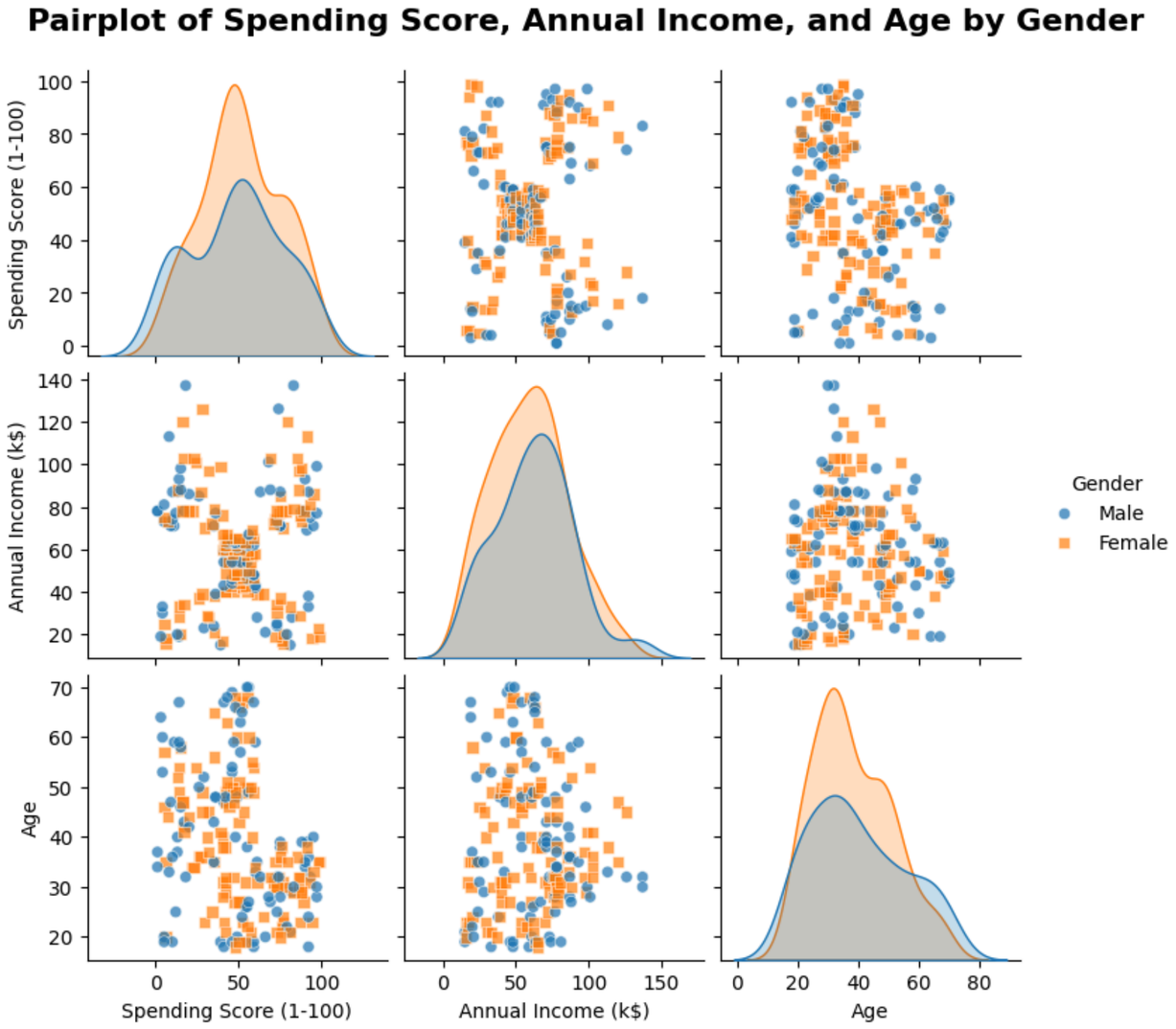






- Distribution of 3 features resembles a Gaussian distribution, where the vast majority of the values lay in the middle with some exceptions in the extremes.
- There are more females than males in the dataset

Data Exploration



- Spending Score: The distribution of spending scores (ranging from 1 to 100) for females appears to be slightly skewed towards higher scores compared to males.
- Annual Income: Both genders have a similar distribution for annual income, with a slight difference where females have a marginally broader range in the higher income bracket.
- Age: The age distribution is quite similar for both genders, although there appears to be a slight difference in the density distribution, especially around the mid-age range (30-50 years).

=> ***The distributions are quite similar across genders with only minor differences.***

Customer Segmentation using Kmeans-clustering

Defining functions

Function to find the optimal number of clusters using elbow method

```
[25] def elbowOptimizer(data):
         wcss = []
         for i in range(1, 11):
             kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
             kmeans.fit(data)
             wcss.append(kmeans.inertia_)
         plt.plot(range(1, 11), wcss, marker='o')
         plt.title('The Elbow Method')
         plt.xlabel('Number of clusters')
         plt.ylabel('WCSS')
         plt.show()
```

Function to find the optimal number of clusters using Silhoette Score

```
[19] def shsr(data):

         range_n_clusters = list(range(2, 10))
         silhouette_avg_scores = []
         for n_clusters in range_n_clusters:
             clusterer = KMeans(n_clusters=n_clusters,init='k-means++', random_state= None)
             cluster_labels = clusterer.fit_predict(data)
             silhouette_avg = silhouette_score(data,cluster_labels)
             silhouette_avg_scores.append(silhouette_avg)
             print(f"For n_clusters = {n_clusters}, the average silhouette score is: {silhouette_avg}")
         plt.figure(figsize=(10, 6))
         plt.plot(range_n_clusters, silhouette_avg_scores, marker='o')
         plt.xlabel('Number of Clusters')
         plt.ylabel('Average Silhouette Score')
         plt.title('Silhouette Scores for Various Numbers of Clusters')
         plt.grid(True)
         plt.show()
```

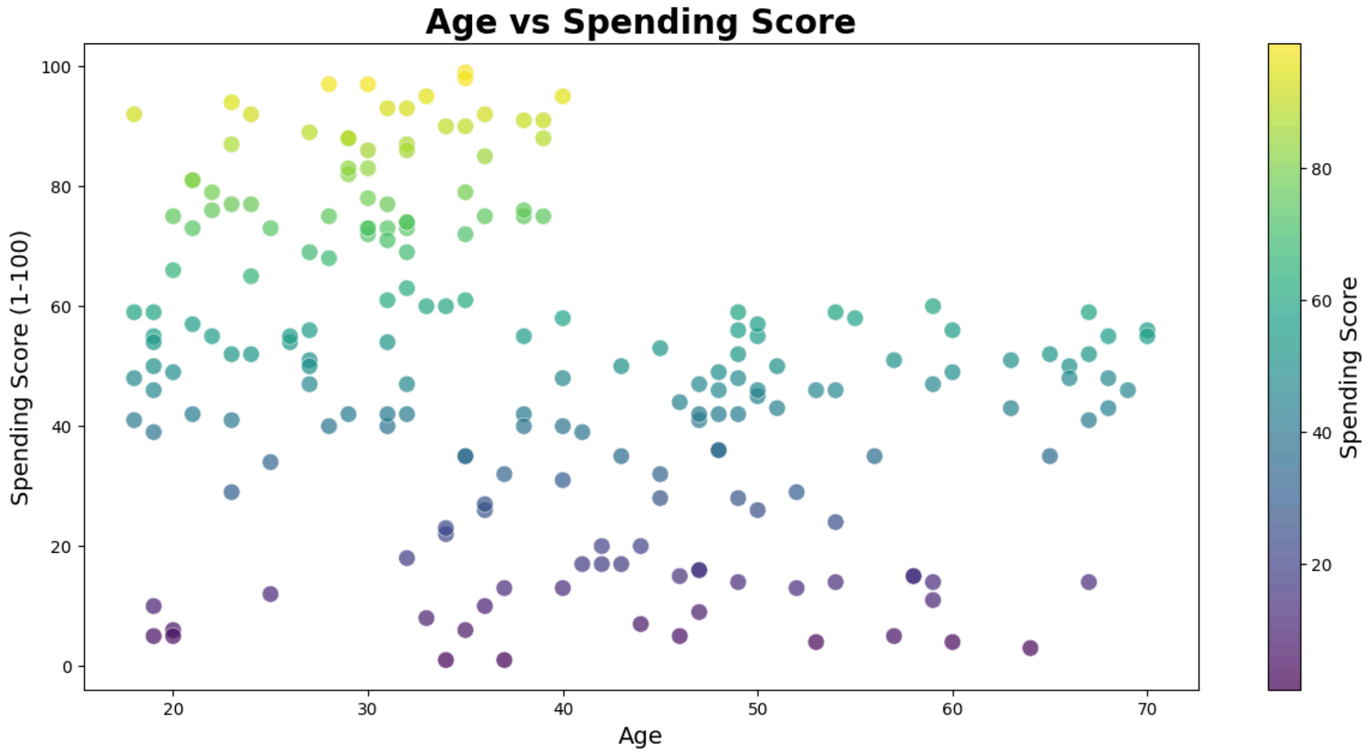
Function for Training K-Means Model on Given Data

```
[28] def kmeansTrainer(numberOfClusters, data):

         kmeans = KMeans(n_clusters=numberOfClusters, init='k-means++', random_state=None)
         labels = kmeans.fit_predict(data)
         return (kmeans,labels)
```

Function for visualising 2-d Clusters

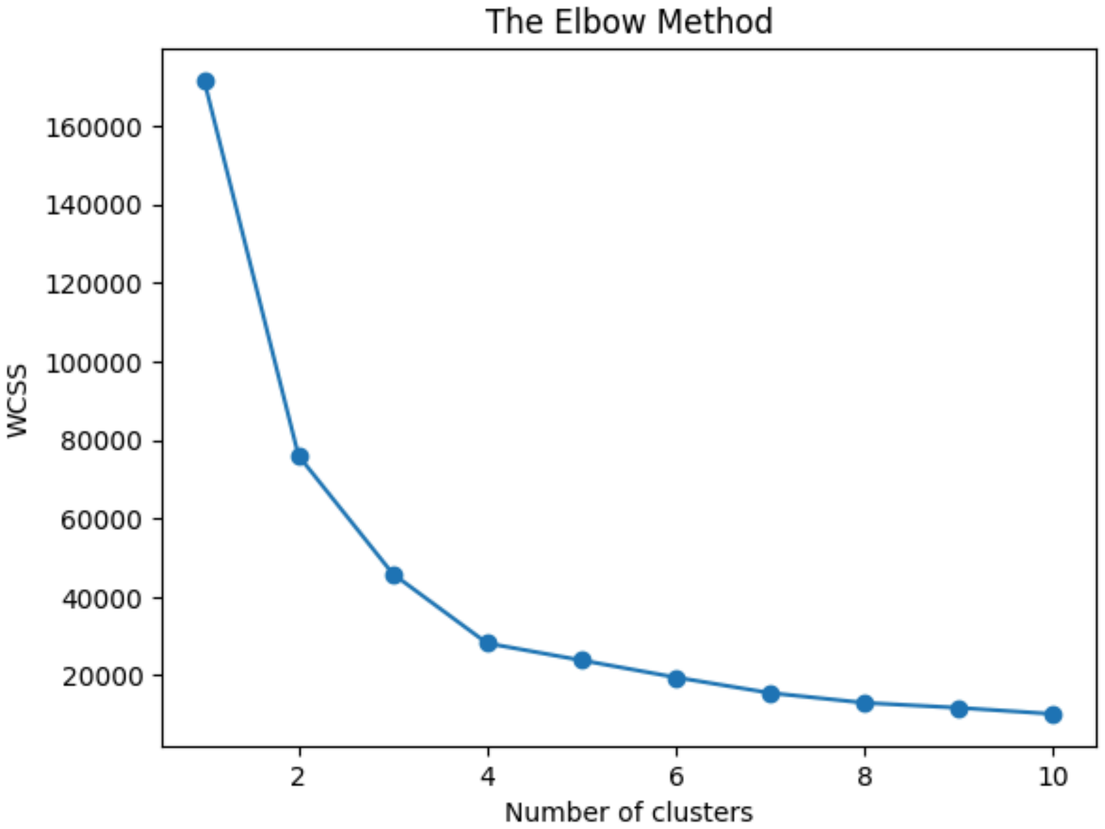
```
[103] def clusterVisualiser(data, model, noOfClusters, labels, xlabel, ylabel):
         colors = ['#FF6347', '#3CB371', '#4682B4', '#FFD700', '#EE82EE', '#8A2BE2', '#FF4500', '#00CED1']

         plt.figure(figsize=(10, 6))

         for i in range(0, noOfClusters):
             plt.scatter(data[labels == i, 0], data[labels == i, 1],
                         s=200, c=colors[i % len(colors)], label='Cluster ' + str(i + 1), alpha=0.6, edgecol

         plt.scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:, 1],
                     s=300, c='yellow', label='Centroids', edgecolor='black', linewidth=1, marker='X')

         plt.title('Clusters of Customers using KMeans Clustering', fontsize=16, fontweight='bold')
         plt.xlabel(xlabel, fontsize=14)
         plt.ylabel(ylabel, fontsize=14)

         plt.legend()
         plt.show()
```

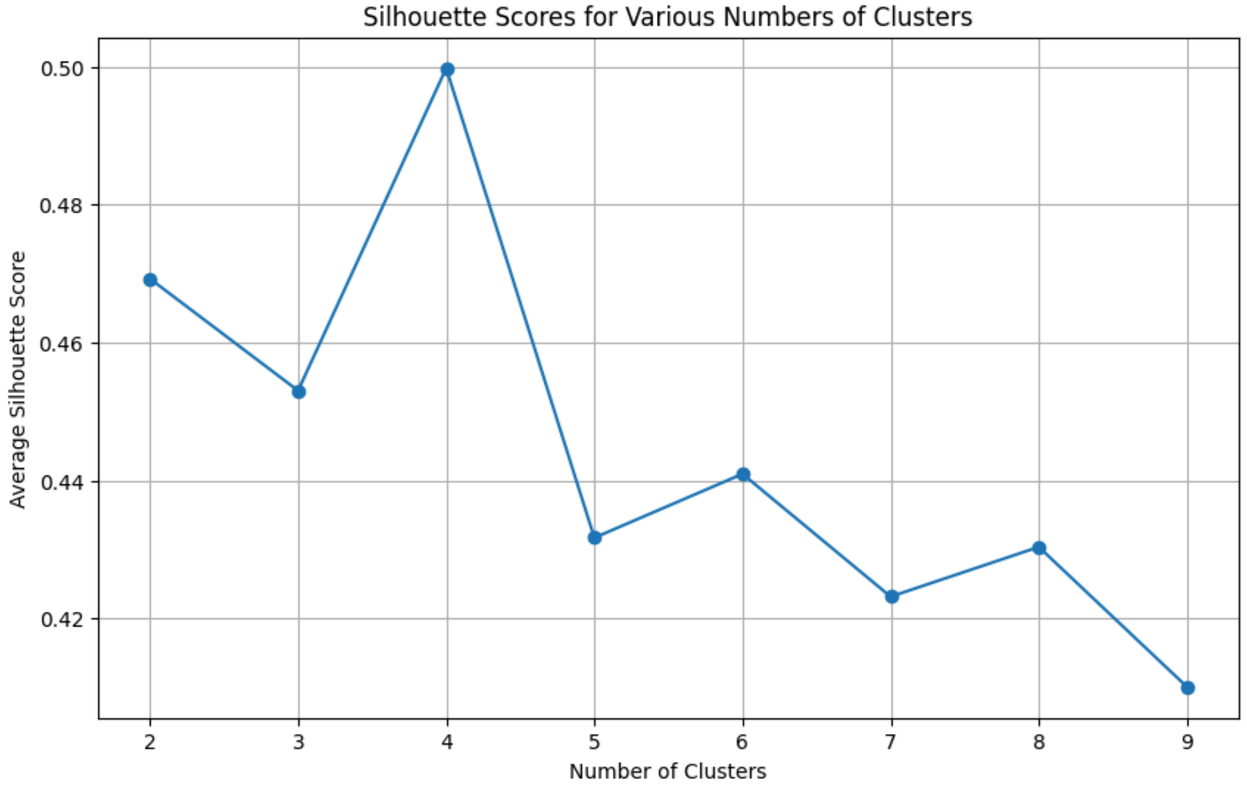
Clustering based on Age and Spending Score



- Younger individuals (ages 20-40) tend to have a higher variability in spending scores, ranging from low to very high.
- Middle-aged individuals (ages 40-60) tend to have more moderate spending scores, with fewer individuals in the extreme high or low spending score ranges.

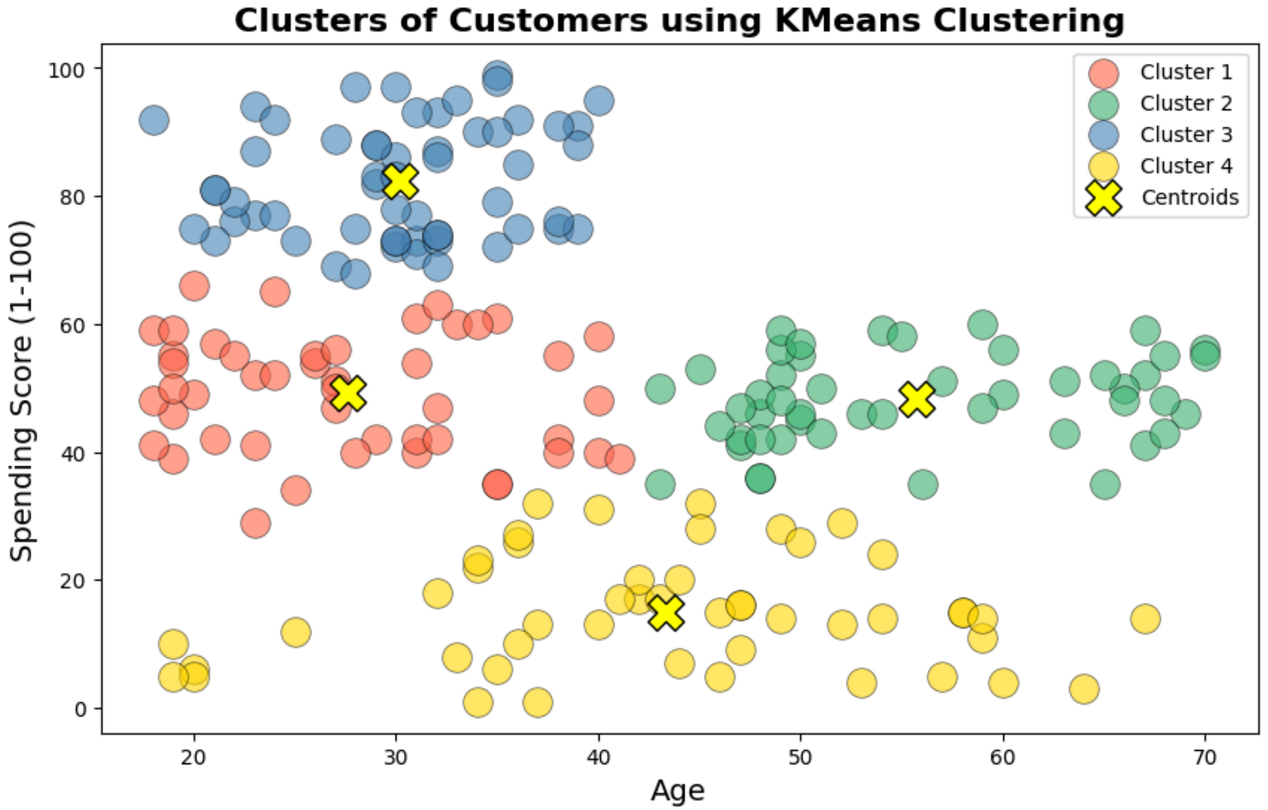
Clustering based on Age and Spending Score





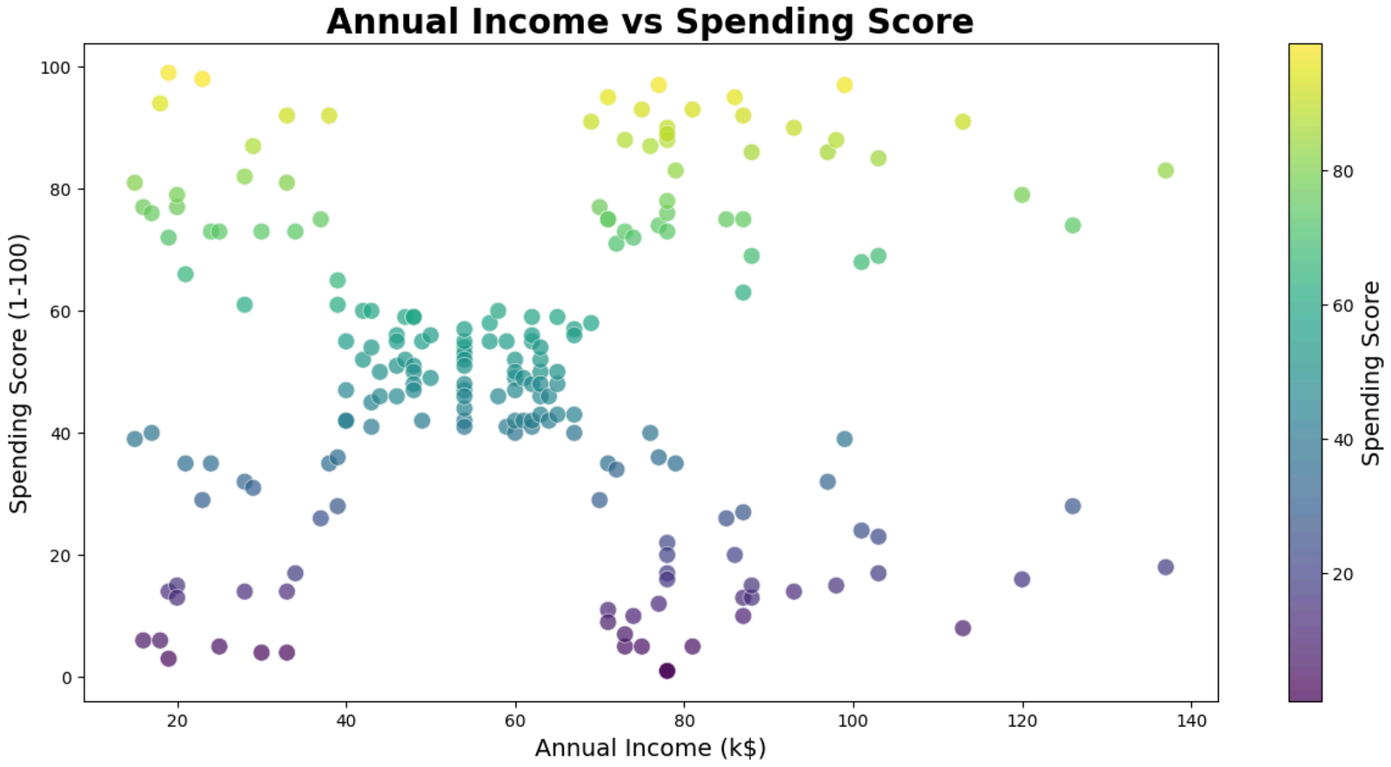
- Deciding K value by 2 methods => selecting 4 as the number of clusters to divide our data in.

Clustering based on Age and Spending Score



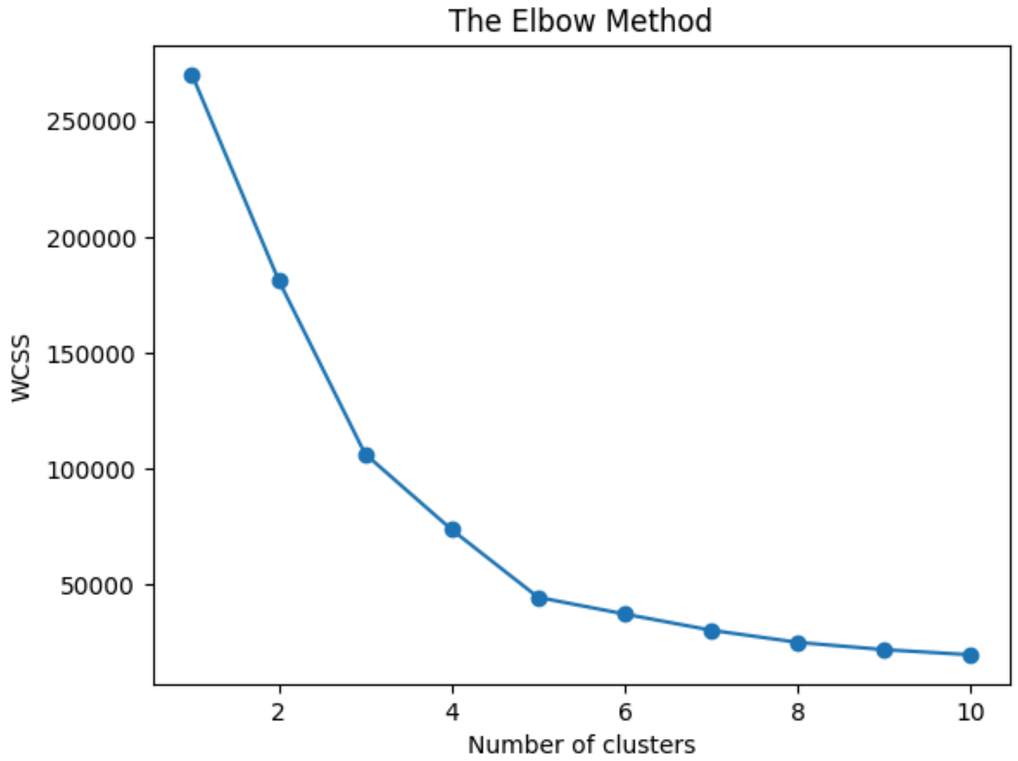
- Cluster 1 (Red): Younger customers (around 20-30 years old) with low to moderate spending scores (0-60).
- Cluster 2 (Green): Middle-aged customers (40-60 years old) with moderate spending scores (40-60).
- Cluster 3 (Blue): Younger customers (around 20-40 years old) with high spending scores (60-100).
- Cluster 4 (Yellow): Older customers (30-70 years old) with low spending scores (0-40).

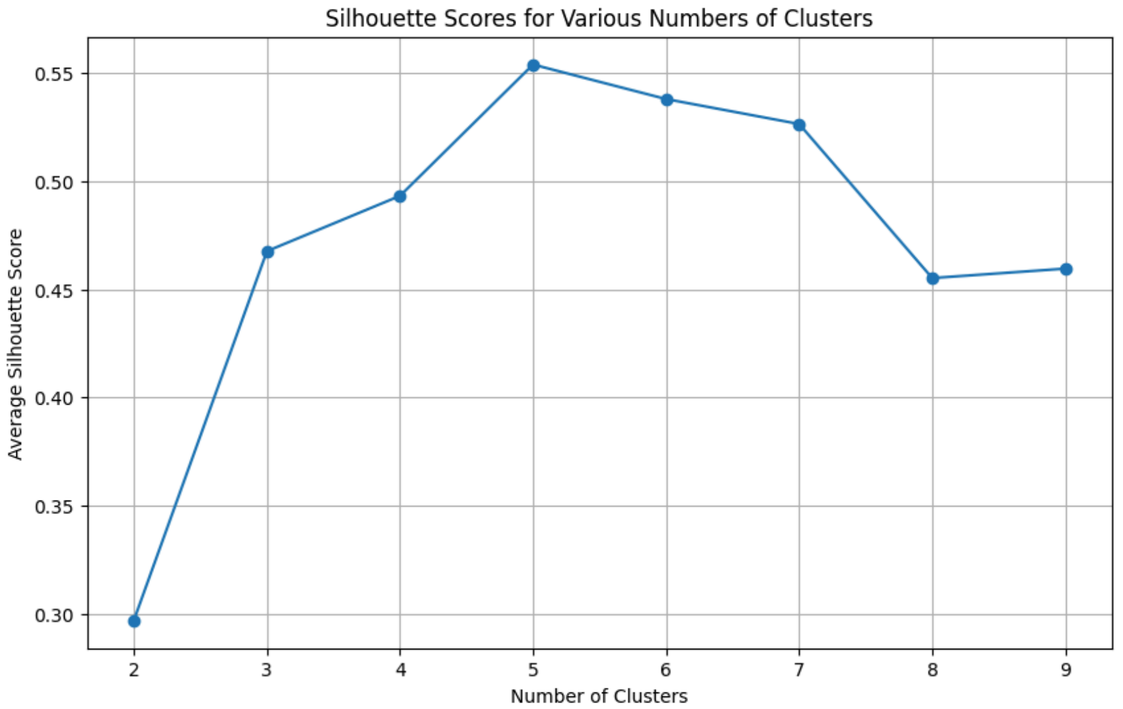
Clustering based on Annual Income and Spending Score



- High spending scores (above 60) appear at both lower and higher income levels, indicating that high spenders are not confined to a specific income range.
- Low spending scores (below 40) are also spread across the income spectrum, from low to high income levels.
- The middle-income group (annual income around $40k to $80k) shows a wide range of spending scores, indicating diverse spending behaviors within this income bracket. This group has individuals with both high and low spending scores.

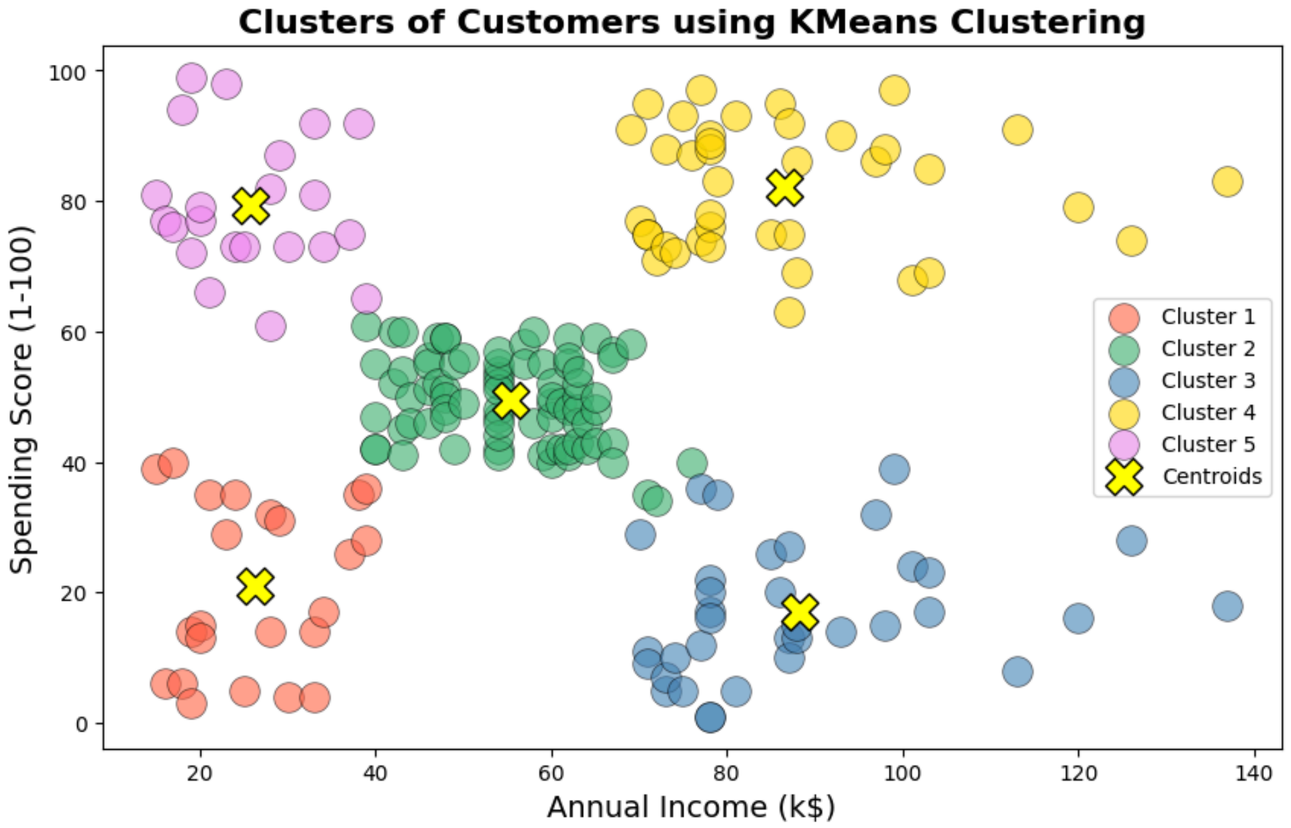
Clustering based on Annual Income and Spending Score





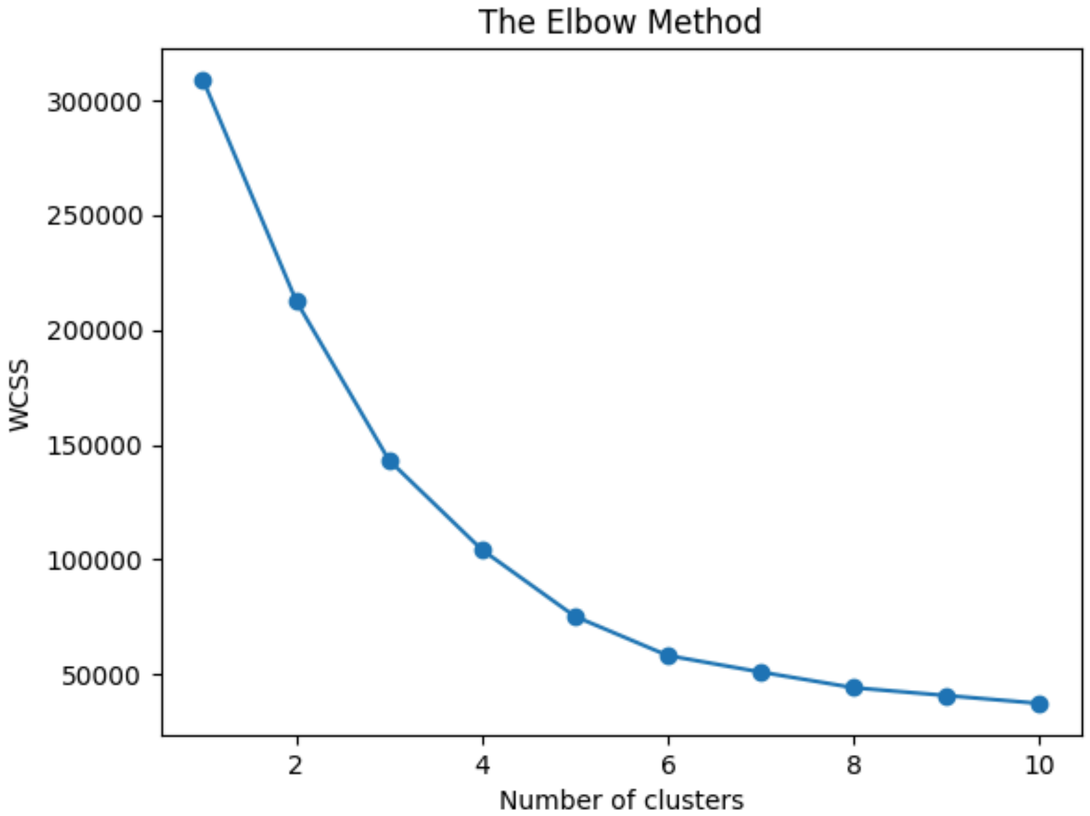
- Selecting k = 5 as the number of clusters to divide our data in.

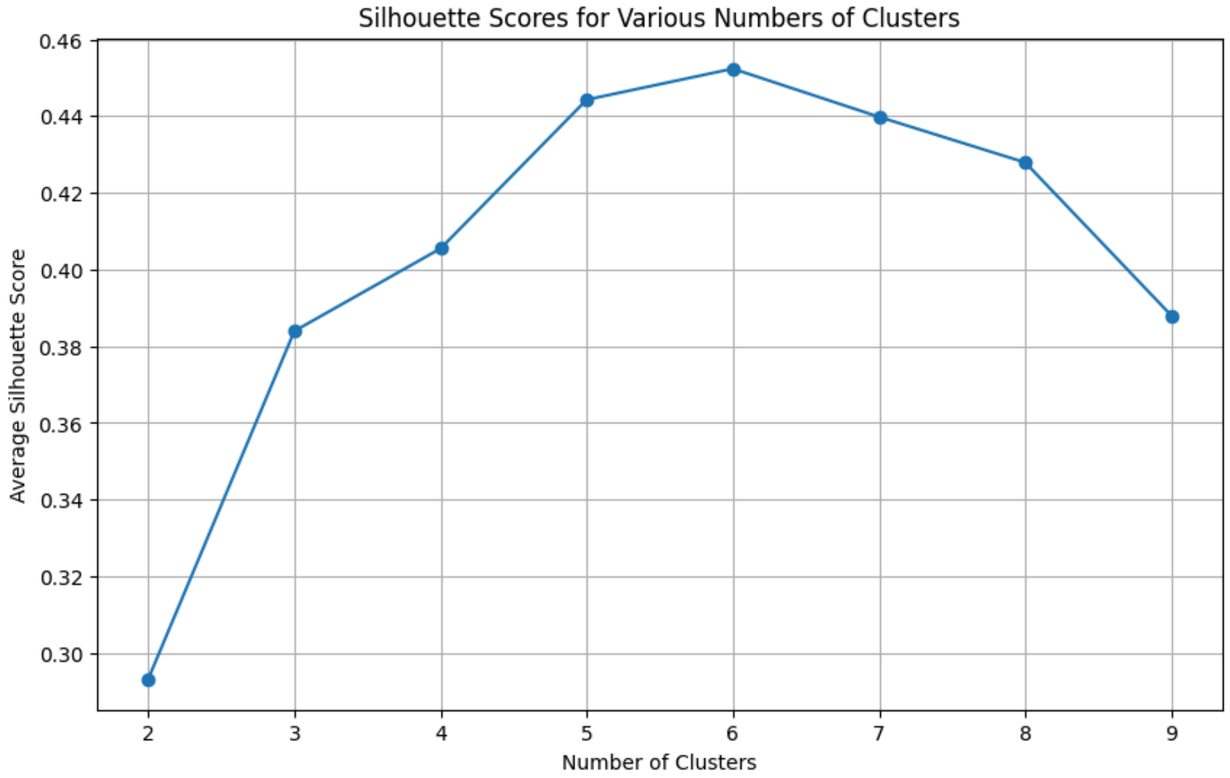
Clustering based on Annual Income and Spending Score



- Cluster 1 (Red): Low-income customers (annual income around $20k) with low spending scores (0-40).
- Cluster 2 (Green): Middle-income customers (annual income around $40k-$60k) with moderate spending scores (40-60).
- Cluster 3 (Blue): Higher-income customers (annual income around $70k-$120k) with low spending scores (0-40).
- Cluster 4 (Yellow): Higher-income customers (annual income around $70k-$120k) with high spending scores (60-100).
- Cluster 5 (Purple): Low to middle-income customers (annual income around $20k-$40k) with high spending scores (60-100).

Clustering based on Age , Annual Income and Spending Score



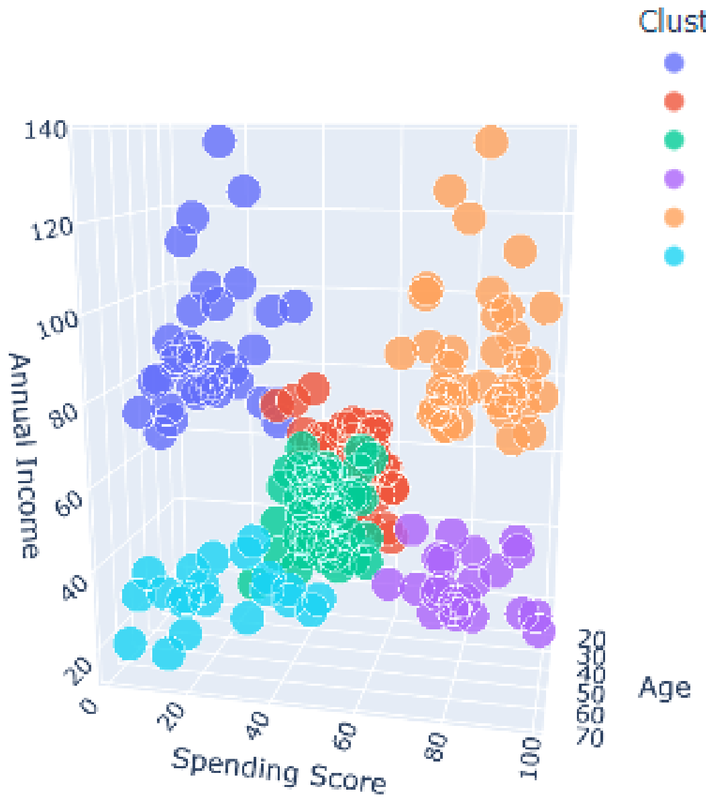


- Selecting k = 6 as the number of clusters to divide our data in.

Clustering based on Age , Annual Income and Spending Score

```
# Calculate customer count for each cluster
cluster_counts = df['cluster'].value_counts().sort_index()

# Create a separate trace for each cluster
traces = []
for cluster_id, name in cluster_names.items():
    cluster_data = df[df['cluster'] == cluster_id]
    trace = go.Scatter3d(
        x=cluster_data['Age'],
        y=cluster_data['Spending Score (1-100)'],
        z=cluster_data['Annual Income (k$)'],
        mode='markers',
        name=f"{name} ({cluster_counts[cluster_id]} customers)",
        marker=dict(
            size=10,
            line=dict(
                color='rgba(255, 255, 255, 0.5)',
                width=0.5
            ),
            opacity=0.8
        )
    )
    traces.append(trace)

layout = go.Layout(
    title='Clusters for Age, Income and Spending Scores',
    scene=dict(
        xaxis=dict(title='Age'),
        yaxis=dict(title='Spending Score'),
        zaxis=dict(title='Annual Income')
    ),
    margin=dict(l=0, r=0, b=0, t=40),
    legend=dict(
        x=1.05,
        y=1,
        title=dict(text='Cluster')
    )
)

fig = go.Figure(data=traces, layout=layout)
py.offline.iplot(fig)
```

```
kmeans, labels = kmeansTrainer(6, X3)
df['cluster'] = pd.DataFrame(labels)
df.head()
```



- 3D scatter plot shows the results of clustering on customer data based on three features: annual income, spending score, and age.

Clustering based on Age , Annual Income and Spending Score

Cluster 0:

- Age: Middle-aged to older adults (mean age: 41.7)
- Spending Score: Moderate
- Annual Income: High
- Customer Count: 35

Cluster 1:

- Age: Young adults (mean age: 27)
- Spending Score: High
- Annual Income: Moderate
- Customer Count: 38

Cluster 2:

- Age: Older adults (mean age: 56.2)
- Spending Score: Low
- Annual Income: Moderate
- Customer Count: 45

Cluster 3:

- Age: Young adults (mean age: 25.3)
- Spending Score: Low
- Annual Income: Low
- Customer Count: 22

Cluster 4:

- Age: Adults (mean age: 32.7)
- Spending Score: High
- Annual Income: High
- Customer Count: 39

Cluster 5:

- Age: Varied ages, including middle-aged to older adults (mean age: 44.1)
- Spending Score: Moderate
- Annual Income: Low
- Customer Count: 21

Identifying target customers

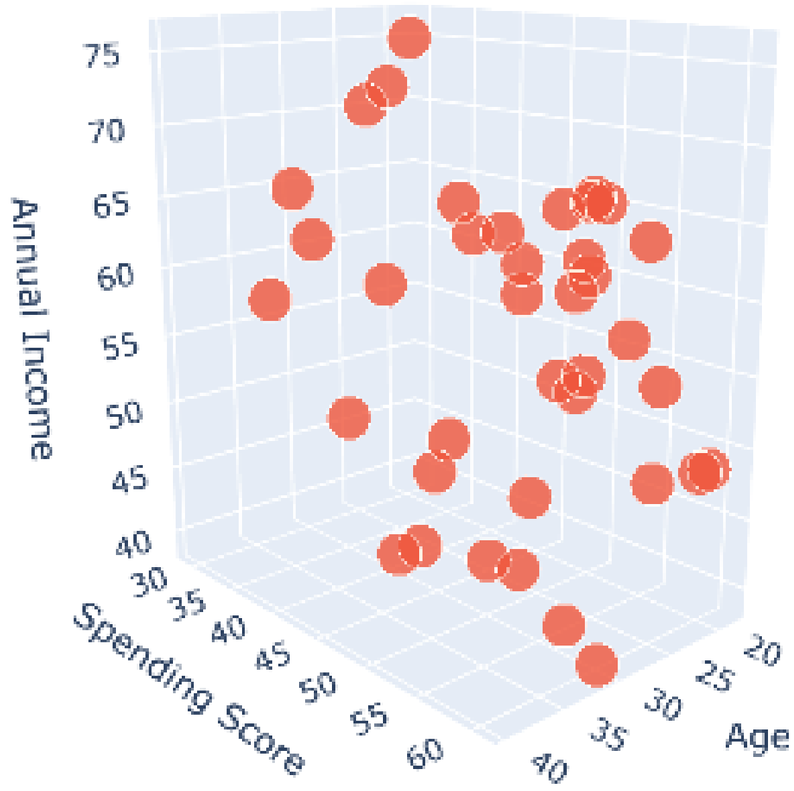
Based on the clusters' characteristics, the two target customer groups that seem most profitable and suitable for focused marketing strategies are Cluster 1 and Cluster 4.

Cluster 1:

Characteristics:

- Age: Young adults (mean age: 27)
- Spending Score: High
- Annual Income: Moderate
- Customer Count: 38

Reason: This cluster represents young adults with high spending scores and moderate income, indicating a group that actively spends money and is likely to respond well to targeted marketing.



Identifying target customers

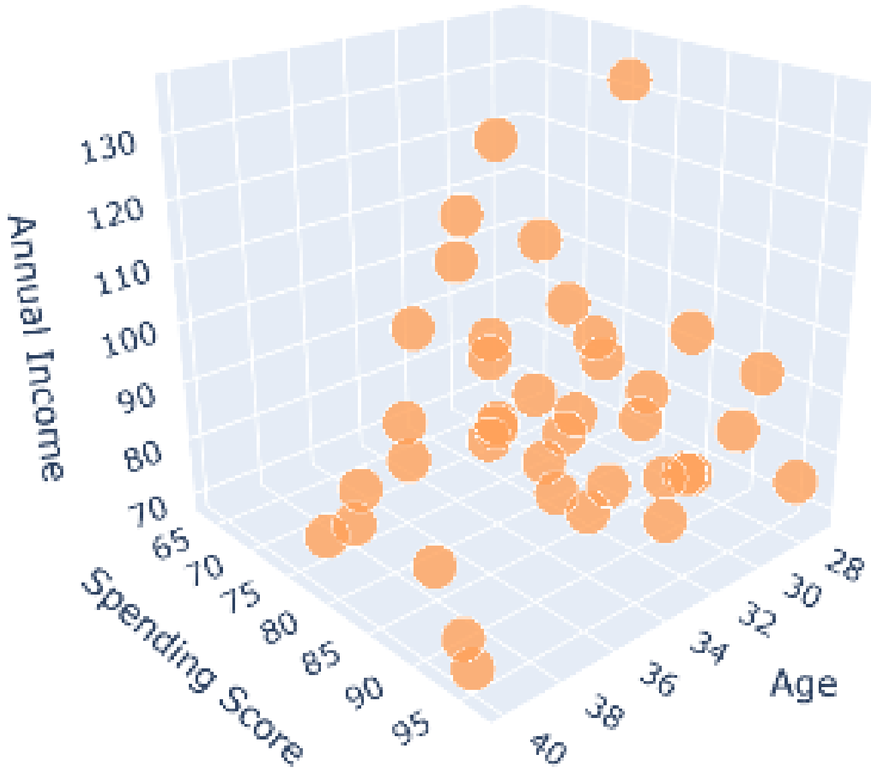
Based on the clusters' characteristics, the two target customer groups that seem most profitable and suitable for focused marketing strategies are Cluster 1 and Cluster 4.

Cluster 4:

Characteristics:

- Age: Adults (mean age: 32.7)
- Spending Score: High
- Annual Income: High
- Customer Count: 39

Reason: This cluster consists of adults with high spending scores and high incomes, making them an ideal target for high-value products and premium services.



Solutions

Marketing strategies for Cluster 1

1. **Social Media Campaigns:**
   - Utilize platforms like Instagram, TikTok and Facebook to run targeted ads and influencer partnerships. Highlight trendy products, seasonal sales, and limited-time offers that appeal to young adults.
2. **Loyalty Programs:**
   - Develop a loyalty program that rewards frequent purchases with points that can be redeemed for discounts, exclusive products, or experiences. Emphasize benefits like birthday rewards and early access to sales.
3. **Event Marketing:**
   - Host events such as fashion shows, product launches, and meet-and-greets with influencers. These events can attract young adults who are looking for engaging and social shopping experiences.
4. **Personalized Promotions:**
   - Use data analytics to send personalized offers and recommendations via email and SMS. Focus on items that align with their spending habits and preferences.

Marketing strategies for Cluster 4

1. **Premium Membership Programs**:
   - Introduce a premium membership tier with benefits such as free delivery, exclusive discounts, personalized shopping assistance, and VIP access to events.
2. **Luxury Product Promotion**:
   - Highlight high-end products and premium brands in marketing materials. Use platforms like LinkedIn and targeted email campaigns to reach this audience with messages about quality and exclusivity.
3. **Exclusive Events**:
   - Organize exclusive events such as private sales, gourmet food tastings, and high-end product demos. Provide a luxurious and personalized shopping experience that caters to their high spending capacity.
4. **Content Marketing**:
   - Create content that resonates with their lifestyle, such as blog posts and video content on financial planning, luxury travel, and high-end technology. Share this content through newsletters and social media channels.


Thank You