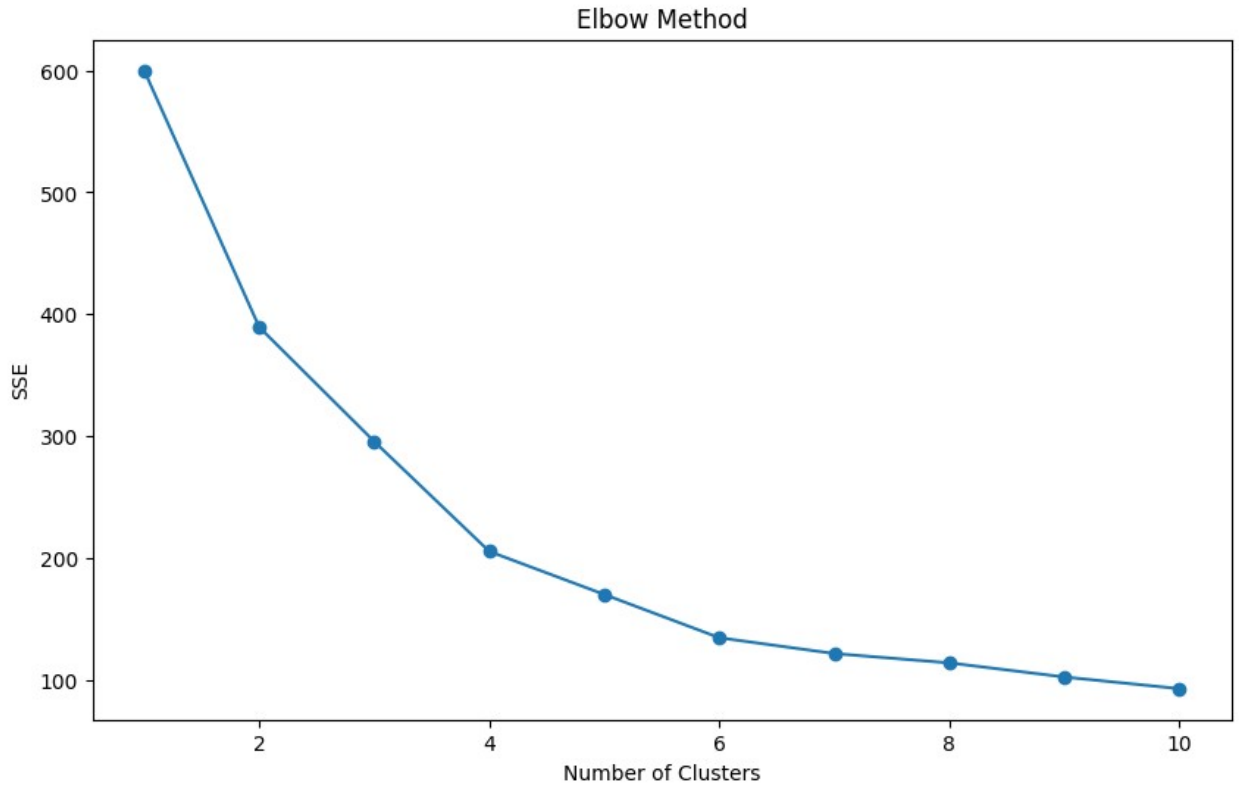
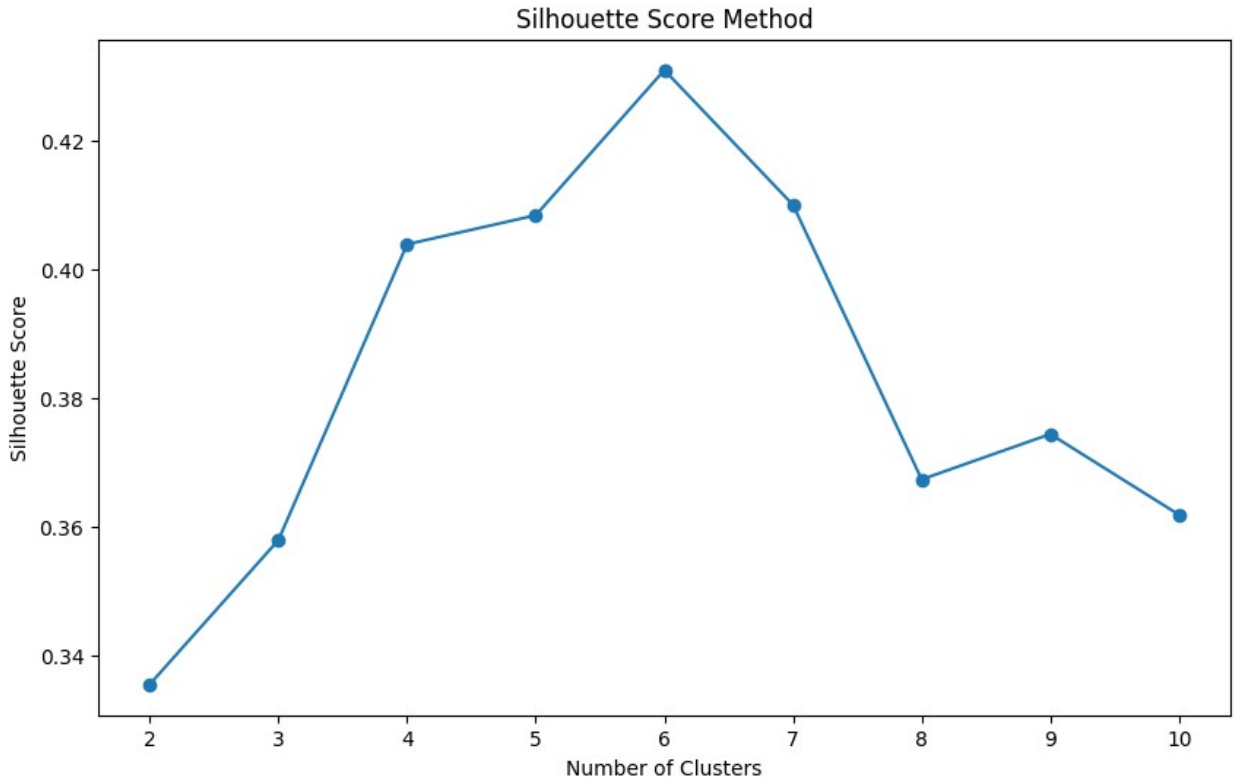
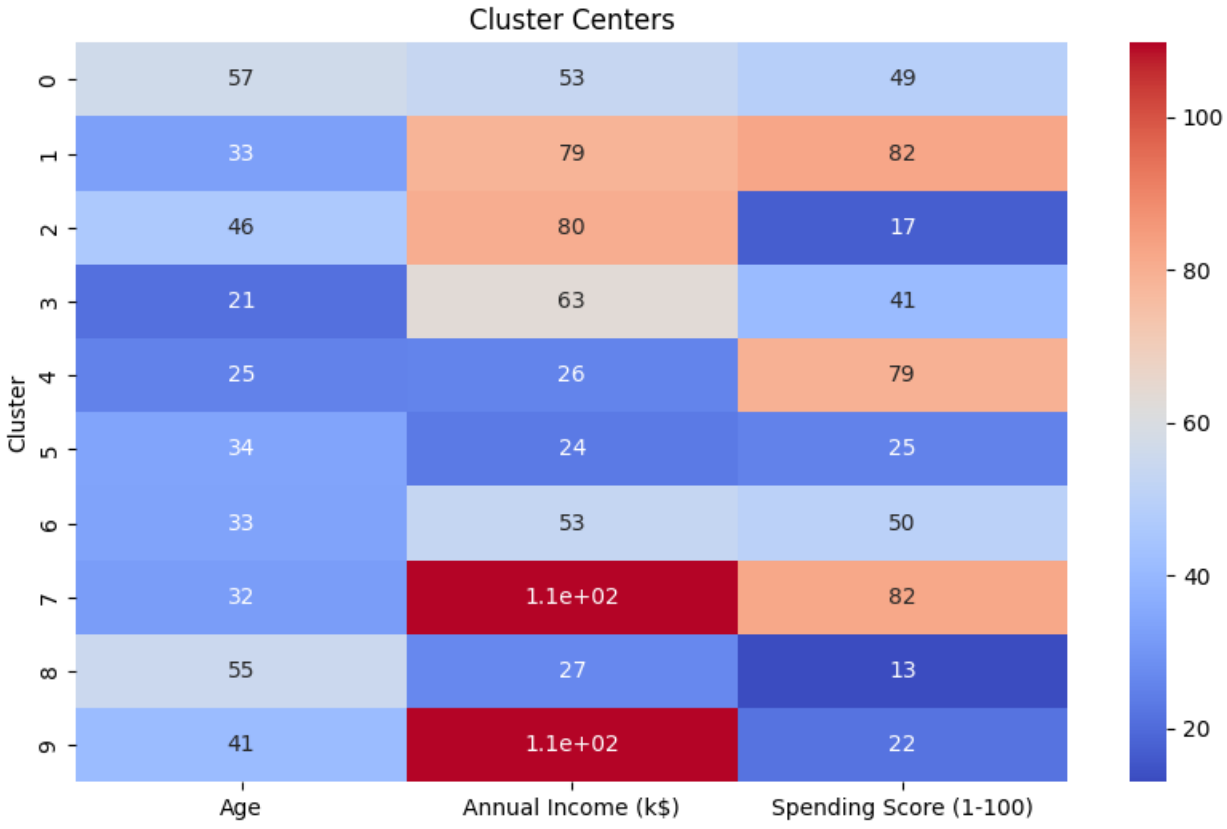
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn.preprocessing
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
optimal_k = 5

data = pd.read_csv('/content/Mall_Customers.csv')

numeric_columns = data.select_dtypes(include=[np.number]).columns
non_numeric_columnms = data.select_dtypes(exclude=[np.number]).columns

data[numeric_columns] =
data[numeric_columns].fillna(data[numeric_columns].mean())

label_encoder = LabelEncoder()
for column in data.select_dtypes(exclude=[np.number]).columns:
    data[column] =
label_encoder.fit_transform(data[column].astype(str))

features = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
X = data[features]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    sse.append(kmeans.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), sse ,marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('SSE')
plt.title('Elbow Method')
plt.show()
```



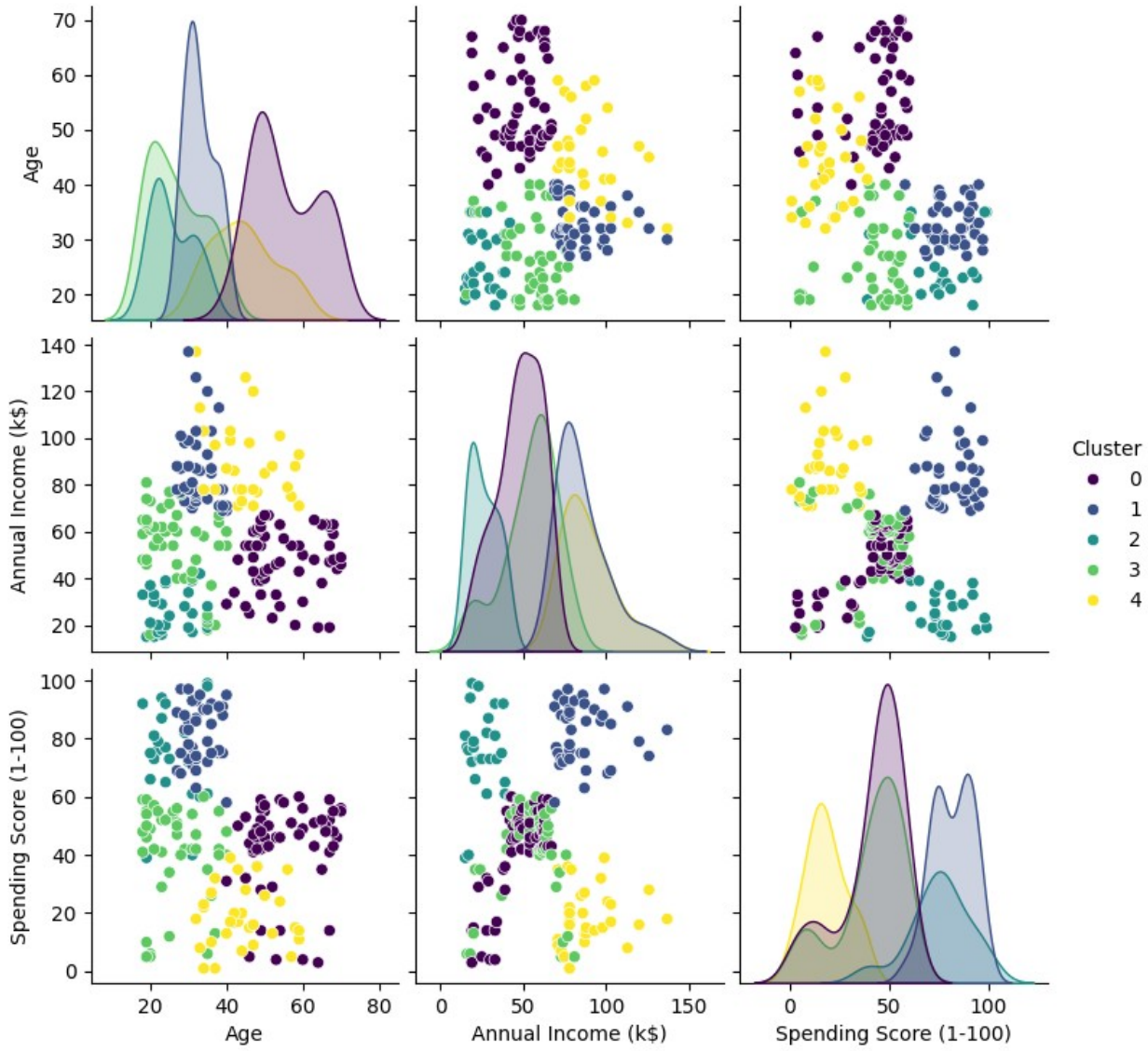

```
silhouette_scores = []
for k in range(2, 11):
    kmeans = KMeans(n_clusters=optimal_k, random_state=42)
    kmeans.fit(X_scaled)
    silhouette_scores.append(silhouette_score(X_scaled,
kmeans.labels_))

plt.figure(figsize=(10, 6))
plt.plot(range(2, 11), silhouette_scores, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score Method')
plt.show()
```




```
clusters_centers = scaler.inverse_transform(kmeans.cluster_centers_)
clusters_centers_df = pd.DataFrame(clusters_centers, columns=features)
clusters_centers_df['Cluster'] = range(kmeans.n_clusters)
data['Cluster'] = kmeans.labels_

plt.figure(figsize=(10, 6))
sns.heatmap(clusters_centers_df.set_index('Cluster'), annot=True,
cmap='coolwarm')
plt.title('Cluster Centers')
plt.show()
```



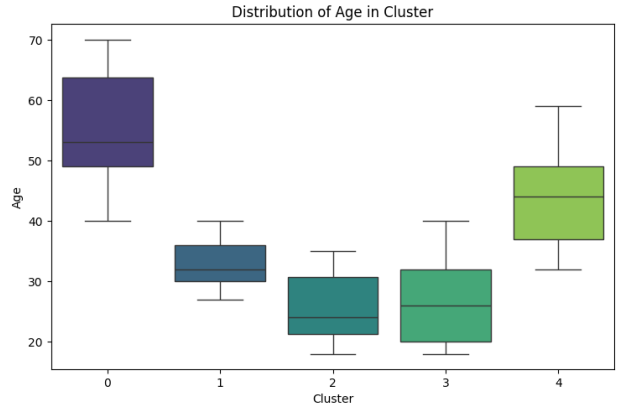

```
sns.pairplot(data, hue='Cluster', palette='viridis',vars=features)
plt.show()
```

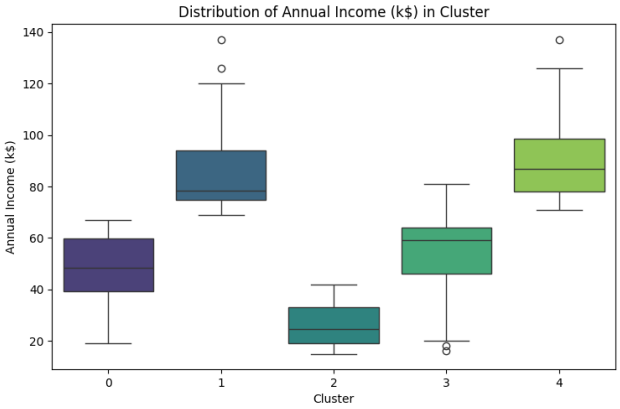


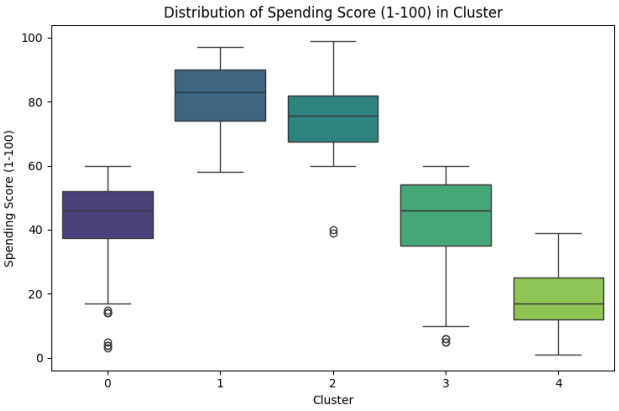

```
plt.figure(figsize=(15, 10))
for i, feature in enumerate(features, 1):
    plt.subplot(2, 2, i)
    sns.boxplot(x='Cluster', y=feature, data=data, palette='viridis')
    plt.title(f'Distribution of {feature} in Cluster')
plt.tight_layout()
plt.show()
```

```
<ipython-input-49-3b36a88aea1a>:4: 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=False` for the same effect.

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

```
<ipython-input-49-3b36a88aea1a>:4: 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=False` for the same effect.

  sns.boxplot(x='Cluster', y=feature, data=data, palette='viridis')
<ipython-input-49-3b36a88aea1a>:4: 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=False` for the same effect.

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








```
cluster_summary = data.groupby('Cluster')
[features].mean().reset_index()
print(cluster_summary)

data.to_csv('Clustered_customers.csv', index=False)
```

	Cluster	Age	Annual Income (k$)	Spending Score (1-100)
0	0	55.275862	47.620690	41.706897
1	1	32.875000	86.100000	81.525000

2	2	25.769231	26.115385	74.846154
3	3	26.733333	54.311111	40.911111
4	4	44.387097	89.774194	18.483871