

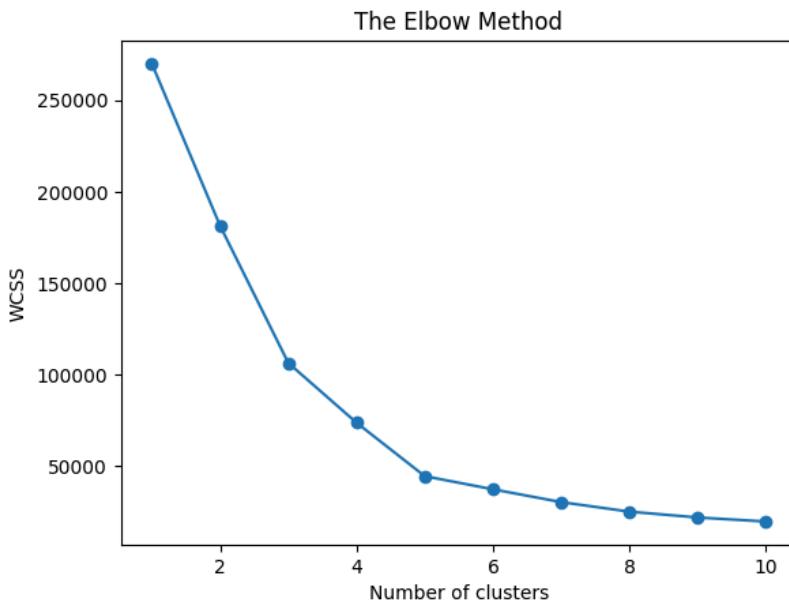
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings("ignore", message="KMeans is known to have a memory leak")
```

```
# Load dataset (make sure the path is correct in your environment)
dataset = pd.read_csv('Mall_Customers.csv')
print(dataset.head())
```

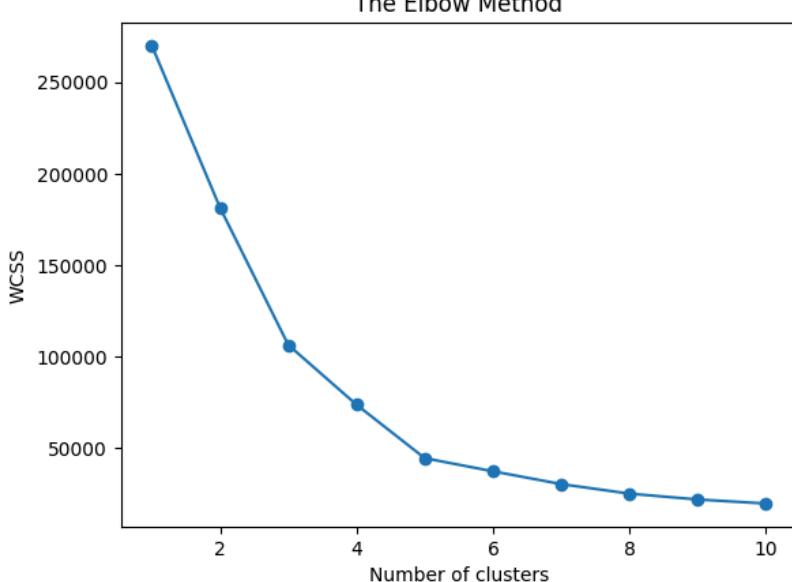
|   | CustomerID | Genre  | Age | Annual Income (k\$) | Spending Score (1-100) |
|---|------------|--------|-----|---------------------|------------------------|
| 0 | 1          | Male   | 19  | 15                  | 39                     |
| 1 | 2          | Male   | 21  | 15                  | 81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

```
# Display first few rows # Select features for clustering (Annual Income and Spending Score)
X = dataset.iloc[:, [3, 4]].values
```

```
# Elbow Method to find the optimal number of clusters
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i,
                     init='k-means++',
                     max_iter=300,
                     n_init=10,
                     random_state=0
    )
    kmeans.fit(X)
    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() # Fit KMeans with 5 clusters
```



Start coding or generate with AI.



```

kmeans = KMeans( n_clusters=5,
                 init='k-means++',
                 max_iter=300,
                 n_init=10,
                 random_state=0
                )
y_kmeans = kmeans.fit_predict(X)
print("Cluster labels:", y_kmeans)
print("Type of labels:", type(y_kmeans)) # Visualizing the clusters
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')
# Plot centroids
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='yellow', label='Centroids', edgecolor='black')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()

```

Type of labels: <class 'numpy.ndarray'>

