

```

# K-Means Clustering on Mall Customers Dataset

from google.colab import files

uploaded = files.upload()

# -----

# Step 1: Import Libraries
# -----

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# -----

# Step 2: Load Dataset
# -----

dataset = pd.read_csv('/content/Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values # Annual Income vs Spending Score

print("Dataset Loaded Successfully!\n")
print(dataset.head())

# -----

# Step 3: Finding Optimal Clusters using Elbow Method
# -----

from sklearn.cluster import KMeans

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_)

```

```

# Plot Elbow Graph

plt.plot(range(1, 11), wcss)

plt.title('Elbow Method')

plt.xlabel('Number of Clusters')

plt.ylabel('WCSS')

plt.show()

# -----

# Step 4: Applying K-Means with 5 Clusters

# -----

kmeans = KMeans(n_clusters=5, init='k-means++', max_iter=300,
                 n_init=10, random_state=0)

y_kmeans = kmeans.fit_predict(X)

print("\nCluster Labels:\n", y_kmeans)

print("\nType of y_kmeans:", type(y_kmeans))

# -----

# Step 5: Plotting the Clusters

# -----

plt.figure(figsize=(8,6))

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')

```

```
plt.title('Clusters of Mall Customers')  
plt.xlabel('Annual Income (k$)')  
plt.ylabel('Spending Score (1-100)')  
plt.legend()  
plt.show()
```

...

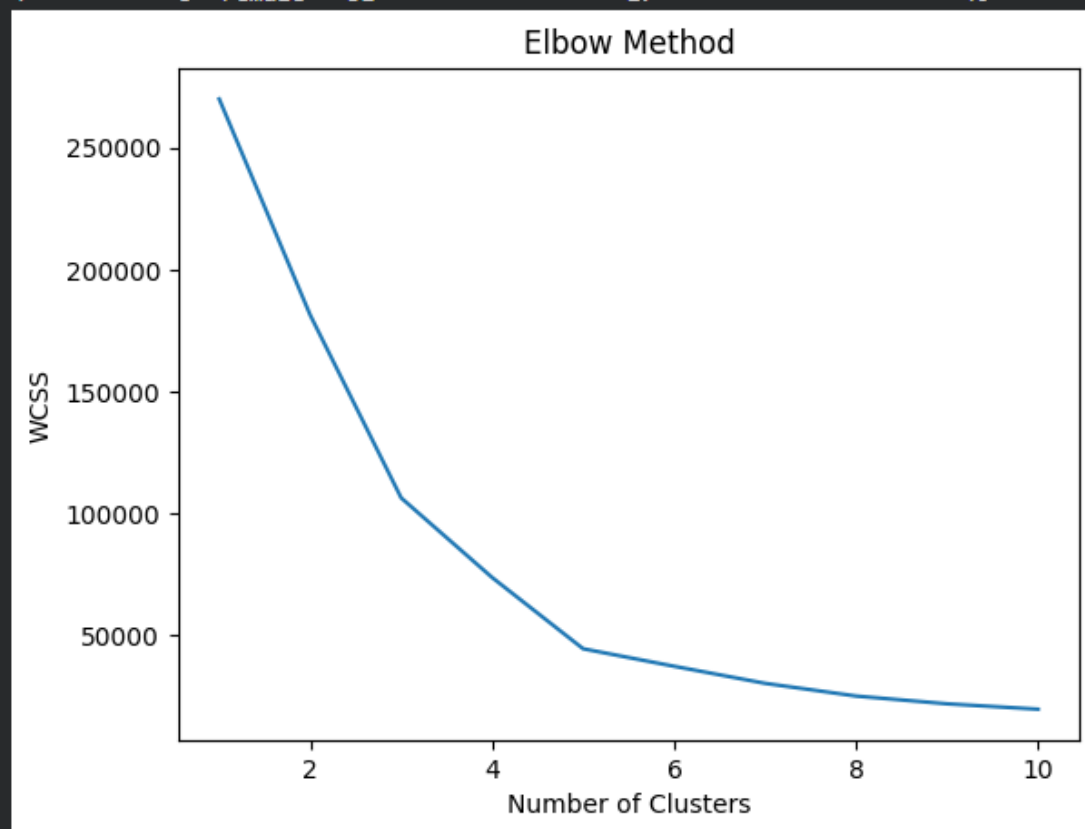
Choose Files Mall_Customers.csv

Mall_Customers.csv(text/csv) - 4286 bytes, last modified: 11/5/2025 - 100% done

Saving Mall_Customers.csv to Mall_Customers.csv

Dataset Loaded Successfully!

	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



```
*** Cluster Labels:
[3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3
 4 3 4 3 4 3 0 3 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 0 1 2 1 2 1 0 1 2 1 2 1 2 1 0
 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
 1 2 1 2 1 2 1 2 1 2 1 2 1]
```

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

