**K-Means Clustering on Mall Customer Dataset**

**1. Introduction**

This project applies K-Means clustering to group customers of a retail store based on their annual income and spending score. The goal is to identify distinct customer segments that can be used to inform marketing strategies.

**2. Objective**

To perform customer segmentation using K-Means clustering on the Mall Customers dataset and determine the optimal number of clusters using the Elbow Method.

**3. Methodology**

* **Dataset Used**: Mall\_Customers.csv  from kaggle
* Link for dataset: https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python
* **Features Selected**: 'Annual Income (k$)' and 'Spending Score (1-100)'
* **Libraries Used**:
  + pandas for data handling
  + matplotlib for visualization
  + sklearn.preprocessing.StandardScaler for feature scaling
  + sklearn.cluster.KMeans for clustering

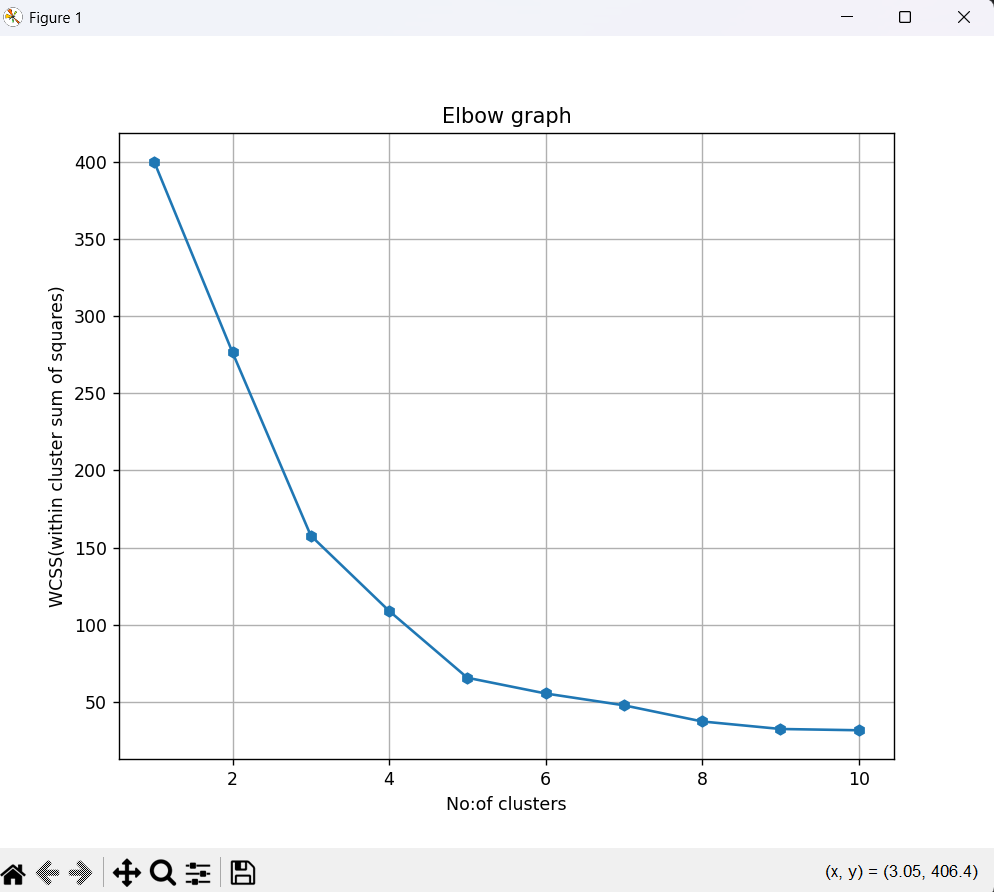
**Steps:**

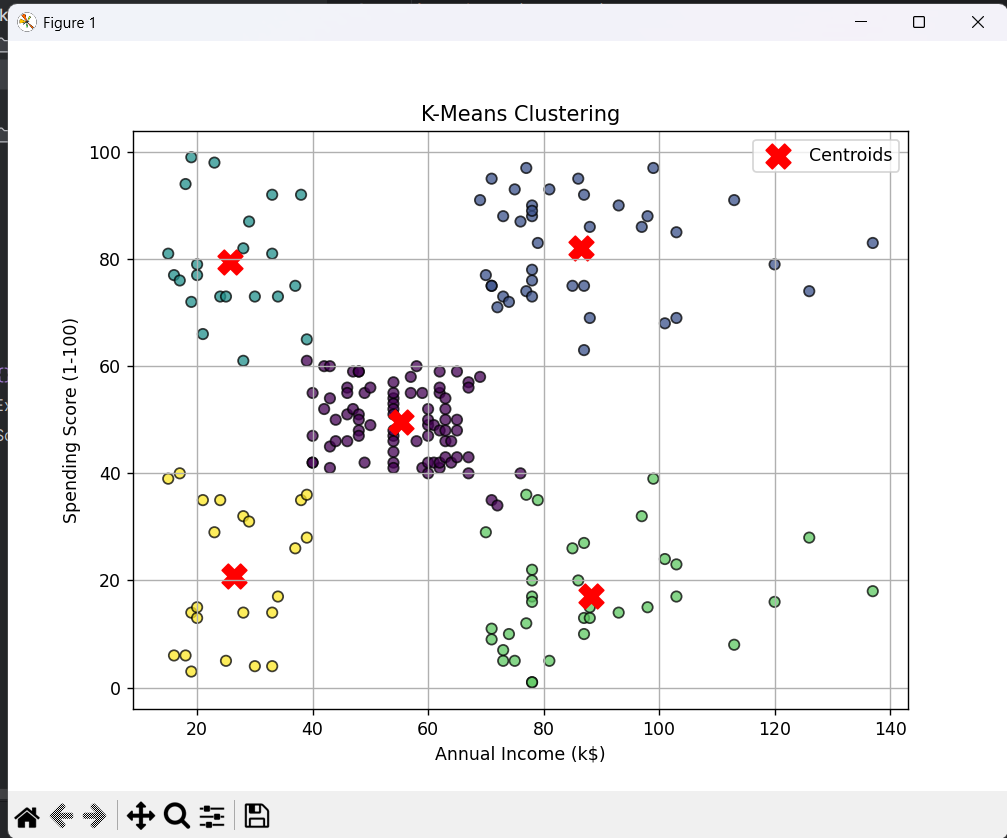
1. Load the dataset.
2. Extract relevant features for clustering.
3. Standardize the features using StandardScaler.
4. Use the Elbow Method to identify the optimal number of clusters.
5. Apply K-Means with the selected number of clusters.
6. Visualize the clustered data and centroids.

**4.code**

import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.cluster import KMeans  
from sklearn.preprocessing import StandardScaler  
  
# Load data  
points\_df = pd.read\_csv('Mall\_Customers.csv')  
  
# Select features  
X = points\_df[['Annual Income (k$)', 'Spending Score (1-100)']]  
  
# Scale the features  
scaler = StandardScaler()  
X\_scaled = scaler.fit\_transform(X)  
  
# Use the Elbow Method to find the optimal number of clusters  
wcss=[]  
for i in range (1,11):  
 kmeans=KMeans(n\_clusters=i)  
 kmeans.fit(X\_scaled)  
 wcss.append(kmeans.inertia\_)  
  
#plot for wcss(within cluster sum of squares) values from elbow method  
plt.figure(figsize=(8,11))  
plt.grid(True)  
plt.plot(range(1,11),wcss,marker='h')  
plt.title("Elbow graph")  
plt.ylabel("WCSS(within cluster sum of squares)")  
plt.xlabel("No:of clusters")  
plt.show()  
  
  
# Apply K-Means clustering with 5 cluster from elbow graph  
kmeans = KMeans(n\_clusters=5, random\_state=42)  
kmeans.fit(X\_scaled)  
labels = kmeans.labels\_  
centroids = kmeans.cluster\_centers\_  
centroids\_original = scaler.inverse\_transform(centroids)  
  
  
# Add cluster labels to the original dataframe (optional)  
points\_df['Cluster'] = labels  
  
# Plot the results  
plt.figure(figsize=(8, 6))  
plt.scatter(X['Annual Income (k$)'], X['Spending Score (1-100)'], c=labels, cmap='viridis', marker='o', edgecolor='k', alpha=0.75)  
plt.scatter(centroids\_original[:, 0], centroids\_original[:, 1], c='red', marker='X', s=200, label='Centroids')  
plt.title('K-Means Clustering')  
plt.xlabel('Annual Income (k$)')  
plt.ylabel('Spending Score (1-100)')  
plt.legend()  
plt.grid(True)  
plt.show()

**Output:**



****

**5. Results and Observations**

* The Elbow Method plot showed a clear elbow at k=3, indicating that three clusters is optimal.
* K-Means clustering successfully grouped customers into five distinct segments:
* High income, high spenders
* High income, low spenders
* Medium income, medium spenders
* Low income, low spenders
* Outliers or unique shoppers
* The clusters can be visually interpreted in the scatter plot with well-separated groups.

**6. Conclusion**

This project demonstrates how K-Means clustering can be used for customer segmentation using simple features like income and spending. The Elbow Method helped determine the ideal number of clusters, making the analysis more meaningful. The results provide insights that can be used for targeted marketing strategies.