Task P1-Batch June

**Task:-3**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Load the Dataset

data = pd.read\_csv('/content/Mall\_Customers.csv')

# Explore the Data

print(data.head())

print(data.info())

# Feature Selection

selected\_features = data[['Annual Income (k$)', 'Spending Score (1-100)']]

# Standardize the Features

scaler = StandardScaler()

scaled\_features = scaler.fit\_transform(selected\_features)

# Determine the Optimal Number of Clusters

# You can use elbow method or silhouette analysis to determine the optimal number of clusters

# Apply K-means Clustering

k = 5  # Assuming 5 clusters

kmeans = KMeans(n\_clusters=k, random\_state=42)

kmeans.fit(scaled\_features)

# Analyze and Visualize the Results

data['Cluster'] = kmeans.labels\_

plt.scatter(data['Annual Income (k$)'], data['Spending Score (1-100)'],

            c=data['Cluster'], cmap='viridis')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.title('Customer Segmentation')

plt.show()

**OUTPUT:-**

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

<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 Genre 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

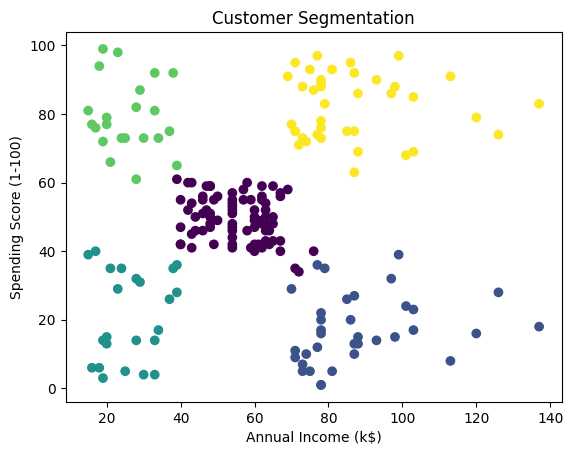
dtypes: int64(4), object(1)

memory usage: 7.9+ KB

None

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/\_kmeans.py:870: FutureWarning: The default value of `n\_init` will change from 10 to 'auto' in 1.4. Set the value of `n\_init` explicitly to suppress the warning

warnings.warn(

****