SmartInternz (Evening Batch)

Assignment-5

Name: Prakhar Agarwal

Reg No.: 21BIT0034

Email ID: prakhar.agarwal2021@vitstudent.ac.in

Market Basket Magic: Extracting Insights for Retail Success Customer segmentation is a crucial aspect of retail and marketing strategy. Mall Customer Segmentation is a common data analysis project that involves categorizing mall customers into distinct groups or segments based on various characteristics and behaviors. This segmentation is valuable for tailoring marketing efforts, optimizing store layouts, and enhancing customer experiences.

Dataset link: Here Task:

1. **Understand the data**
2. **Data Preprocessing**
3. **Machine Learning approach with clustering algorithm**

```
import numpy as np
import pandas as pd
```

```
d=pd.read_csv("C:\\Users\\wwwad\\Downloads\\archive (1)\\Mall_Customers.csv")
df=pd.DataFrame(d)
print(d)
```

```
     CustomerID  Gender  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
..          ...     ...  ...                 ...                     ...
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83

[200 rows x 5 columns]
```

```
df.isnull().sum()
```

```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

```
df.dropna(inplace=True)
df
```

	CustomerID	Gender	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
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

```
X = df[['Age', 'Annual Income (k$)', 'Gender']]
y = df['Spending Score (1-100)']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
X_train
```

	Age	Annual Income (k$)	Gender
134	20	73	Male
66	43	48	Female
26	45	28	Female
113	19	64	Male
168	36	87	Female
...	...	...	...
67	68	48	Female
192	33	113	Male
117	49	65	Female
47	27	40	Female
172	36	87	Male

160 rows × 3 columns

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```
le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])
le
```

```
▾ LabelEncoder
LabelEncoder()
```

```
print(df.head())
```

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

```
scaler = StandardScaler()
df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']] = scaler.fit_transform(
    df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']])
print(df.head())
```

```
   CustomerID  Gender       Age  Annual Income (k$)  Spending Score (1-100)
0           1       1 -1.424569           -1.738999               -0.434801
1           2       1 -1.281035           -1.738999                1.195704
2           3       0 -1.352802           -1.700830               -1.715913
3           4       0 -1.137502           -1.700830                1.040418
4           5       0 -0.563369           -1.662660               -0.395980
```

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_train.drop(columns=['Gender']))
train_clusters = kmeans.predict(X_train.drop(columns=['Gender']))
test_clusters = kmeans.predict(X_test.drop(columns=['Gender']))
train_clusters
```

```
C:\Users\wwwad\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\wwwad\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OM
P_NUM_THREADS=1.
  warnings.warn(
```

```
array([1, 0, 2, 1, 1, 0, 0, 2, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       1, 2, 1, 0, 0, 1, 2, 0, 1, 0, 1, 0, 1, 0, 0, 2, 2, 1, 1, 0, 1, 1,
       0, 0, 2, 0, 1, 1, 2, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 2, 1, 0,
       1, 0, 1, 2, 2, 2, 2, 0, 2, 1, 2, 1, 1, 2, 2, 2, 0, 2, 1, 0, 2, 0,
       1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2, 0, 0, 2, 0, 2, 2, 1, 1, 0, 1,
       1, 1, 2, 2, 1, 0, 1, 0, 0, 2, 1, 1, 2, 1, 1, 1, 0, 1, 1, 1, 2, 1,
       0, 0, 1, 1, 1, 0, 0, 2, 1, 0, 1, 1, 1, 2, 2, 0, 0, 0, 2, 2, 2, 2,
       1, 0, 1, 0, 2, 1])
```

```
test_clusters
```

```
array([2, 1, 0, 0, 1, 1, 2, 1, 0, 1, 2, 1, 1, 1, 0, 2, 2, 1, 2, 0, 1, 1,
       2, 1, 0, 1, 1, 1, 1, 0, 2, 0, 1, 1, 2, 1, 1, 2, 0, 0])
```

```
silhouette_avg = silhouette_score(X_train.drop(columns=['Gender']), train_clusters)
print(f'Silhouette Score: {silhouette_avg}')
```

```
Silhouette Score: 0.41425604594750765
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.scatter(X_train['Age'], X_train['Annual Income (k$)'], c=train_clusters, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', label='Cluster Centers')
plt.xlabel('Age')
plt.ylabel('Annual Income (k$)')
plt.title('K-Means Clustering')
plt.legend()
plt.show()
```