Name : Bhoomi Mangesh Naik
Class : D15C
Roll No. : 34
Practical No. : 6

Aim : Apply K-Means and Hierarchical Clustering on sample datasets.

Theory :

1. Dataset Source

The dataset used for this experiment is obtained from Kaggle:

Mall Customers Dataset
https://www.kaggle.com/datasets/shwetabh123/mall-customers

2. Dataset Description

The Mall Customers dataset contains customer information collected from a shopping mall for market segmentation.

Dataset Characteristics

- **Total Records:** 200
- **Type:** Structured numerical dataset
- **Learning Type:** Unsupervised learning

Features Used

Feature	Description
Annual Income (k$)	Customer's annual income
Spending Score (1–100)	Customer spending behavior

There is **no target variable**, making the dataset suitable for clustering.

3. Mathematical Formulation

3.1 K-Means Clustering

K-Means partitions data into *k*k clusters by minimizing within-cluster variance.

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

Where:

- *Ci*Ci = cluster *i*i
- *μi*μi = centroid of cluster *i*i

3.2 Hierarchical Clustering

Hierarchical clustering builds a hierarchy of clusters.

Ward's Method:

$$\Delta E = \sum (x - \bar{x})^2$$

Clusters are merged based on minimum variance increase.

4. Algorithm Limitations

K-Means Limitations

- Requires predefined number of clusters
- Sensitive to outliers
- Assumes spherical clusters

Hierarchical Clustering Limitations

- Computationally expensive for large datasets
- Cannot undo previous merges
- Sensitive to noise

5. Methodology / Workflow

1. Dataset collection from Kaggle
2. Data upload in Google Colab
3. Data cleaning and duplicate removal
4. Feature selection
5. K-Means clustering
6. Hierarchical clustering
7. Visualization of clusters
8. Result interpretation

Workflow Diagram:

Dataset → Cleaning → Feature Selection → K-Means → Visualization → Hierarchical → Dendrogram → Visualization

6. Performance Analysis

Since clustering is unsupervised:

- Performance is evaluated using **visual inspection**
- Compact and well-separated clusters indicate good performance

K-Means produced clearly separated customer segments, while Hierarchical Clustering provided insight into cluster formation using dendrograms.

7. Hyperparameter Tuning

K-Means

- n_clusters tuned using elbow method
- Optimal value improves cluster compactness

Hierarchical Clustering

- n_clusters
- Linkage method (ward)

Impact:
Proper tuning leads to meaningful and interpretable clusters.

Code and Output :

```
from google.colab import files
import pandas as pd
```

```
# Upload dataset
uploaded = files.upload()

# Load dataset
df = pd.read_csv("Mall_Customers.csv")

print("Initial Shape:", df.shape)
print("\nMissing Values:\n", df.isnull().sum())

# Data Cleaning
df = df.drop_duplicates()

# Select relevant features for clustering
X = df[['Annual Income (k$)', 'Spending Score (1-100)']]

print("\nFinal Shape for clustering:", X.shape)
X.head()
```

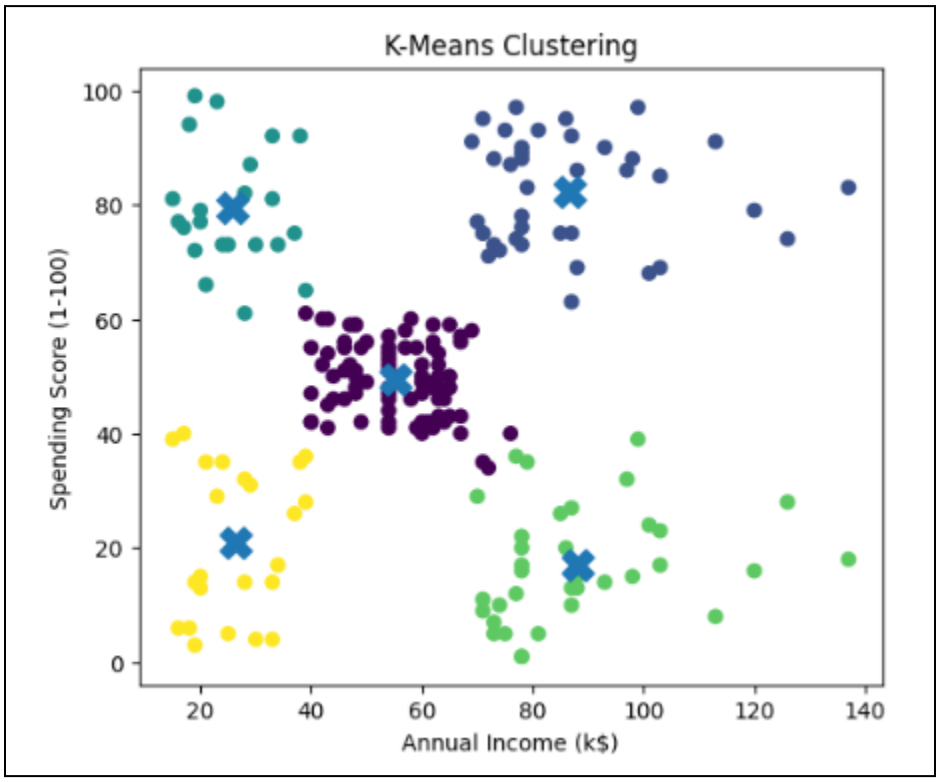
Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Mall_Customers.csv to Mall_Customers.csv
Initial Shape: (200, 5)

Missing Values:
 CustomerID              0
Genre                    0
Age                      0
Annual Income (k$)       0
Spending Score (1-100)   0
dtype: int64

Final Shape for clustering: (200, 2)
```

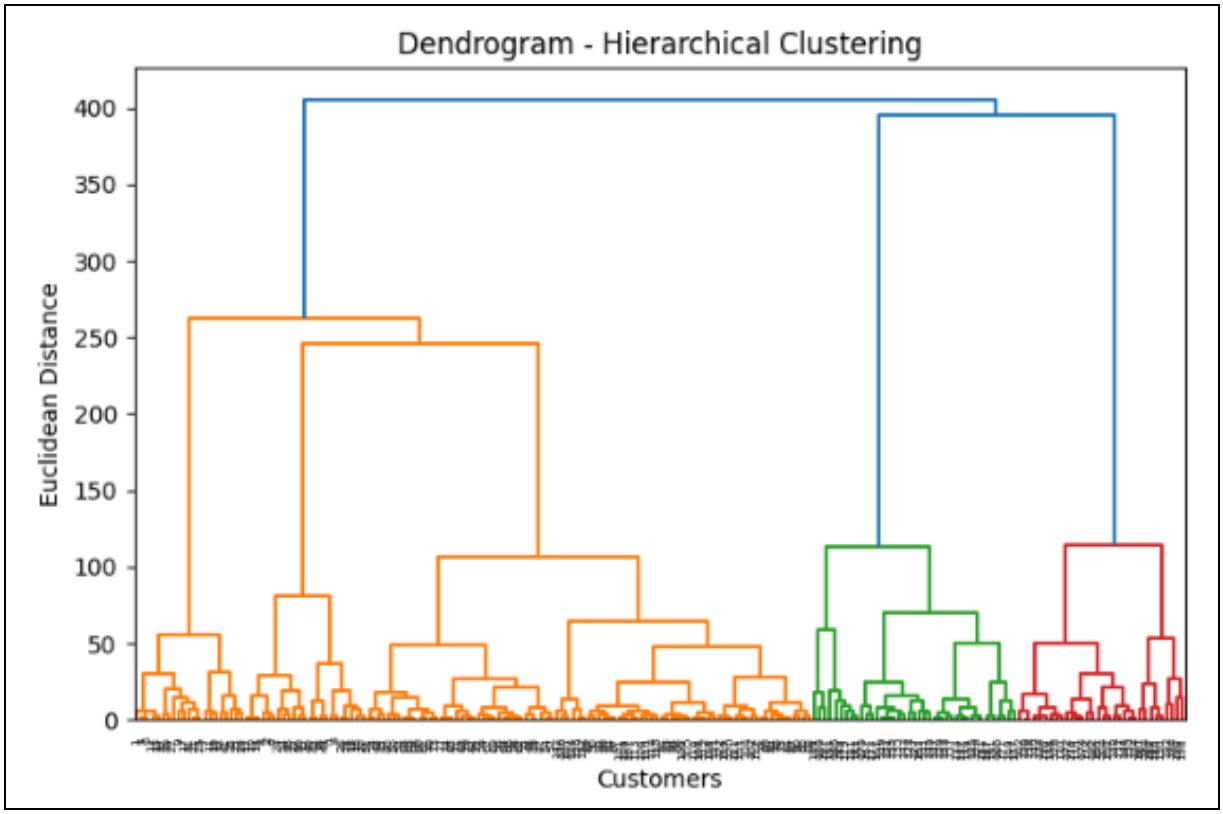
	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

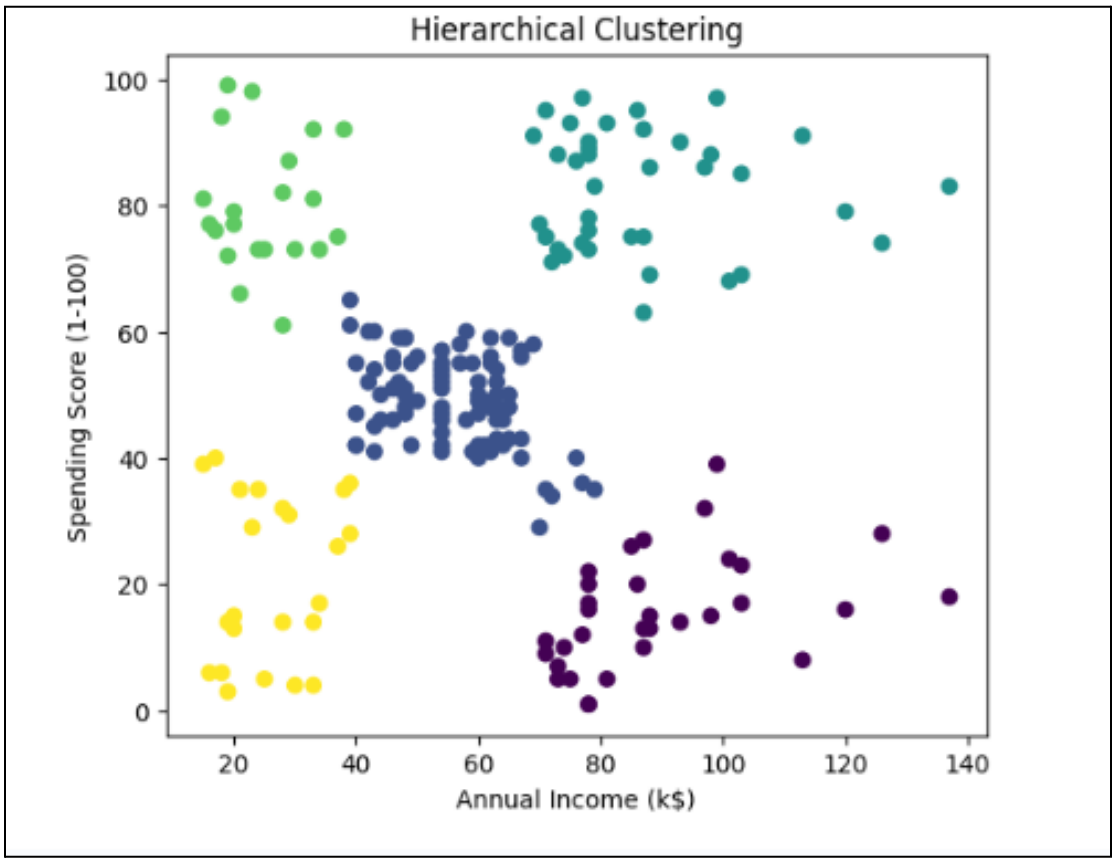
```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# K-Means model
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(X)
```

```
# Cluster labels
labels_kmeans = kmeans.labels_

# Plot clusters
plt.figure(figsize=(6,5))
plt.scatter(
    X.iloc[:, 0],
    X.iloc[:, 1],
    c=labels_kmeans
)
plt.scatter(
    kmeans.cluster_centers_[:, 0],
    kmeans.cluster_centers_[:, 1],
    marker='X',
    s=200
)
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.title("K-Means Clustering")
plt.show()
```



```
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering

# Dendrogram
linked = linkage(X, method='ward')

plt.figure(figsize=(8,5))
dendrogram(linked)
plt.title("Dendrogram - Hierarchical Clustering")
plt.xlabel("Customers")
plt.ylabel("Euclidean Distance")
plt.show()

# Agglomerative Clustering
hc = AgglomerativeClustering(n_clusters=5)
labels_hc = hc.fit_predict(X)

# Plot clusters
plt.figure(figsize=(6,5))
plt.scatter(
    X.iloc[:, 0],
    X.iloc[:, 1],
    c=labels_hc
)
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.title("Hierarchical Clustering")
plt.show()
```





Google Colab Link for Code and Output : [Link for Code and Output]

Conclusion :

In this experiment, K-Means and Hierarchical Clustering were successfully applied to a real-world customer dataset. K-Means efficiently grouped customers based on spending behavior, while Hierarchical Clustering provided a hierarchical structure of clusters. The experiment demonstrates the effectiveness of clustering techniques for market segmentation tasks.