Name: Atharva Phalke
Class: D15C
Roll No. : 39

ML & DL : Experiment - 6

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

Dataset Description:

The **Mall Customers Dataset** is a real-world retail analytics dataset that contains basic demographic and behavioral information of customers visiting a shopping mall. The primary objective of this dataset is to help businesses **understand customer segments based on income levels and spending behavior**, making it highly suitable for **customer segmentation and clustering tasks**.

The dataset captures essential customer attributes such as **age, gender, annual income, and spending score**, which together provide insights into purchasing capacity and consumption patterns of different customer groups. Unlike supervised learning datasets, this dataset does not include a predefined target variable; instead, it is designed for **unsupervised learning techniques** such as **Hierarchical Clustering**, where the goal is to discover natural groupings among customers based on similarity measures.

Due to its structured numerical feature space, simple retail context, and manageable size, the dataset is widely used in machine learning experiments and academic projects related to **market segmentation, customer profiling, and personalized marketing strategies**. With appropriate preprocessing such as **feature selection and scaling**, the dataset enables meaningful identification of customer clusters and interpretation of shopping behavior patterns.

- **File Type:** CSV (Comma Separated Values)
- **Dataset Size:** 200 rows × 5 columns
- **Target Variable:** Not applicable (unsupervised learning)

Dataset Source:

https://raw.githubusercontent.com/SteffiPeTaffy/machineLearningAZ/master/Machine%20Learning%20A-Z%20Template%20Folder/Part%204%20-%20Clustering/Section%2025%20-%20Hierarchical%20Clustering/Mall_Customers.csv

K - Means Clustering

Theory:

Hierarchical Clustering is an unsupervised learning technique used to group data points into clusters based on their similarity, without requiring a predefined number of clusters. Unlike partition-based methods such as K-Means, hierarchical clustering builds a multi-level hierarchy of clusters, which is typically visualized using a dendrogram. This tree-like structure represents how individual data points are progressively merged (or split) into larger clusters based on a chosen distance metric and linkage criterion.

$$X = \{x_1, x_2, x_3, \ldots, x_n\}, \quad x_i \in \mathbb{R}^d$$

The goal of K-Means is to partition the dataset into K disjoint clusters:

$$C = \{C_1, C_2, \ldots, C_K\}$$

The most commonly used form in practical applications is agglomerative hierarchical clustering (bottom-up approach). In this method, each data point initially forms its own cluster, and pairs of clusters are iteratively merged based on their similarity until all points belong to a single cluster. The similarity between clusters is computed using different linkage methods such as single linkage, complete linkage, average linkage, or Ward's method. Each linkage strategy defines how the distance between clusters is measured and influences the final cluster structure.

Given a dataset with *n* data points and *d* features, hierarchical clustering aims to organize the data into a nested sequence of clusters that captures the underlying structure of the data. By analyzing the dendrogram, an appropriate number of clusters can be selected by cutting the tree at a suitable height. This makes hierarchical clustering especially useful when the optimal number of clusters is not known in advance.

Each cluster Ck is represented by a centroid μk , defined as the mean of all points assigned to that cluster:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

Limitations:

1. **High Computational Complexity:**
   Hierarchical clustering has a computational complexity of O(n²), which makes it resource-intensive for large datasets. As the number of customers increases, the time and memory required to compute pairwise distances and construct the dendrogram grow rapidly. This limits its scalability for very large retail databases.

2. **Sensitivity to Noise and Outliers:**
   Outliers can significantly influence the cluster formation process in hierarchical clustering. Since early merges or splits cannot be undone, the presence of anomalous customer records may distort the dendrogram structure and lead to less meaningful clusters.

3. **Irreversibility of Cluster Merging:**
   In agglomerative hierarchical clustering, once two clusters are merged, the decision cannot be reversed. If an incorrect merge occurs at an early stage, it can propagate through subsequent levels of the hierarchy and negatively affect the final cluster structure.

4. **Dependence on Distance Metric and Linkage Method:**
   The quality of clustering results is highly dependent on the chosen **distance metric** (e.g., Euclidean distance) and **linkage criterion** (single, complete, average, or Ward's method). Different choices can lead to different dendrogram structures and cluster interpretations, requiring careful experimentation and validation.

Workflow:

1. **Data Collection:**
   The Mall Customers dataset (Mall_Customers.csv) is loaded into a Pandas DataFrame. The dataset contains customer attributes such as age, gender, annual income, and spending score, which are used for customer segmentation.

2. **Data Cleaning and Preparation:**
   The customer identifier column is removed as it does not contribute to clustering. Any missing values (if present) are handled appropriately to ensure data consistency. The dataset is checked for duplicates and basic anomalies.

3. **Feature Selection:**
   Relevant numerical features such as **Annual Income** and **Spending Score** are

selected for clustering, as these variables best represent purchasing power and consumption behavior. Age may also be included to capture demographic variation.

4. **Feature Scaling:**
   Selected features are standardized using **StandardScaler** to eliminate scale differences. This step is essential for distance-based clustering methods to ensure that no single feature dominates the distance computation.

5. **Distance Computation:**
   Pairwise distances between customers are computed using an appropriate distance metric (commonly Euclidean distance) to measure similarity between data points.

6. **Model Selection (Hierarchical Clustering):**
   Agglomerative Hierarchical Clustering is applied using suitable linkage methods such as **Ward's method** or **complete linkage** to iteratively merge similar customer groups.

7. **Dendrogram Analysis:**
   A dendrogram is constructed to visualize the hierarchical cluster structure. The optimal number of clusters is determined by identifying a suitable cut-off level where distinct cluster separations are observed.

8. **Final Cluster Formation:**
   Based on the dendrogram cut, the final number of clusters is selected, and each customer is assigned to a cluster accordingly.

9. **Cluster Profiling:**
   Cluster-wise summary statistics (such as mean income and average spending score) are computed to interpret the characteristics of each customer segment and derive meaningful business insights.

10. **Visualization:**
    The resulting clusters are visualized using 2D plots (for example, Annual Income vs. Spending Score) to illustrate the separation between customer groups and support qualitative analysis of the clustering results.

Performance Analysis:

The performance of the **Hierarchical Clustering** model on the **Mall Customers dataset** is evaluated using **internal validation techniques and visual analysis**, as no ground-truth labels are available in unsupervised learning. The evaluation focuses on cluster separation, structural clarity, stability, and interpretability using **dendrogram analysis, distance-based inspection, and cluster profiling**.

1. **Determination of Optimal Number of Clusters (Dendrogram Analysis):**
   The dendrogram provides a hierarchical view of how individual customers are merged into larger clusters. A clear increase in linkage distance at higher levels of the dendrogram indicates natural separation points in the data. By selecting a suitable cut level in the dendrogram, an optimal number of customer clusters is identified. This cut reveals distinct groupings that balance cluster compactness and interpretability, without over-fragmenting the dataset.

2. **Cluster Separation and Visualization:**
   Two-dimensional visualizations based on selected features such as **Annual Income vs. Spending Score** show clear separation between customer clusters. Most customers are grouped into compact and well-defined regions, with limited overlap between clusters. This indicates that hierarchical clustering successfully captures meaningful structure in customer purchasing behavior.

3. **Cluster Size Distribution:**
   The distribution of customers across clusters shows that each cluster contains a reasonable number of data points, with no extremely small or disproportionately large clusters. This balanced distribution suggests that the clustering solution is stable and not dominated by noise or outliers.

4. **Cluster Profiling and Interpretation:**
   Cluster-wise summary statistics (such as average income, mean spending score, and age distribution) highlight clear differences in purchasing behavior across clusters. For example, one cluster may represent **high-income, high-spending customers**, while another may correspond to **moderate-income, moderate-spending customers**. These distinctions validate the practical usefulness of the clustering results for customer segmentation and targeted marketing strategies.

Hyperparameter Tuning:

Hierarchical clustering does not require explicit specification of the number of clusters during model fitting; however, the quality of clustering depends heavily on the choice of **distance metric** and **linkage method**. Therefore, careful tuning and evaluation of these parameters are essential to obtain meaningful and stable customer segments.

The following aspects are considered during model tuning:

1. **Linkage Method Selection:**
   Different linkage strategies such as **single linkage, complete linkage, average linkage, and Ward's method** are evaluated. Ward's method generally produces more

compact and well-separated clusters by minimizing within-cluster variance, making it suitable for customer segmentation tasks.

2. **Distance Metric:**
   The **Euclidean distance** metric is commonly used for numerical features such as income and spending score. Alternative distance measures may be explored to analyze their impact on cluster formation and separation.

3. **Dendrogram Cut-Level (Number of Clusters):**
   The final number of clusters is determined by selecting an appropriate cut level on the dendrogram. This cut-level acts as an implicit hyperparameter, controlling the granularity of segmentation. The chosen cut balances interpretability with cluster compactness.

4. **Feature Selection and Scaling:**
   The choice of input features and proper **feature scaling** significantly influence clustering results. Standardization ensures that all features contribute equally to distance computation, leading to more meaningful and stable clusters.

Code & Output:

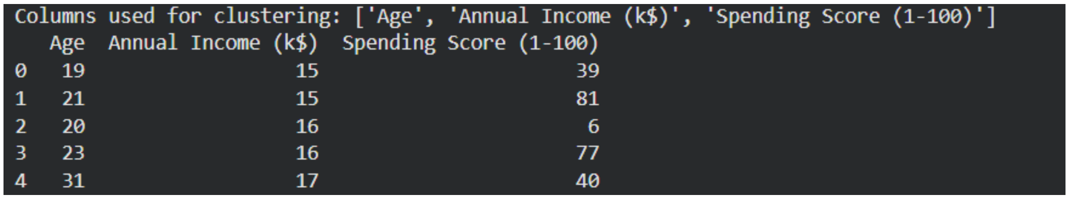
```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette_score

from sklearn.decomposition import PCA

from scipy.cluster.hierarchy import dendrogram, linkage



# 1. Load Dataset directly for Google Colab

url =
"https://raw.githubusercontent.com/SteffiPeTaffy/machineLearningAZ/ma
ster/Machine%20Learning%20A-Z%20Template%20Folder/Part%204%20-%20Clus
tering/Section%2025%20-%20Hierarchical%20Clustering/Mall_Customers.cs
v"
```

```
df = pd.read_csv(url)

# 2. Preprocessing
df.columns = df.columns.str.strip()

# FIX: The dataset uses 'Genre' instead of 'Gender'
# We drop non-numeric columns and ID for clustering
cols_to_drop = []
if 'CustomerID' in df.columns: cols_to_drop.append('CustomerID')
if 'Genre' in df.columns: cols_to_drop.append('Genre')
if 'Gender' in df.columns: cols_to_drop.append('Gender')

df_numeric = df.drop(columns=cols_to_drop)

# Handle missing values
df_numeric = df_numeric.fillna(df_numeric.mean())

print("Columns used for clustering:", df_numeric.columns.tolist())
print(df_numeric.head())

# 3. Feature Scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_numeric)

# 4. Finding Optimal K (Elbow Method & Silhouette Score)
wcss = []
```

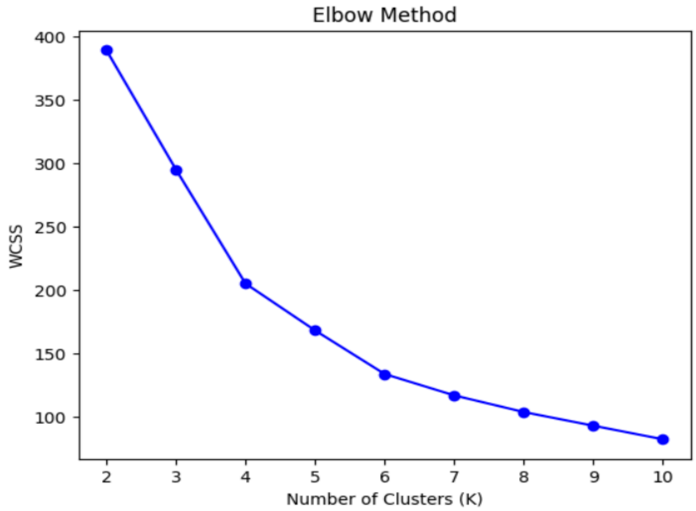
```
silhouette_scores = []
K_range = range(2, 11)

for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(X_scaled)
    wcss.append(kmeans.inertia_)
    silhouette_scores.append(silhouette_score(X_scaled, labels))

# Plotting Elbow and Silhouette side-by-side

plt.figure(figsize=(14, 5))

plt.subplot(1, 2, 1)
plt.plot(range(2, 11), wcss, marker='o', color='blue')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')

plt.subplot(1, 2, 2)
plt.plot(K_range, silhouette_scores, marker='o', color='purple')
plt.title('Silhouette Score vs K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Silhouette Score')
plt.show()
```

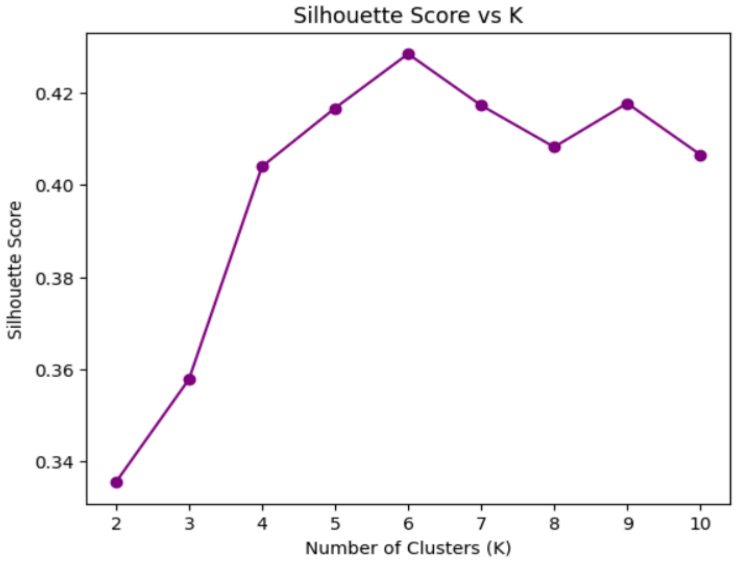
```
optimal_k = K_range[np.argmax(silhouette_scores)]
print(f"Optimal number of clusters based on Silhouette Score:
{optimal_k}")


# 5. Final K-Means Clustering
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
cluster_labels = kmeans.fit_predict(X_scaled)


# 6. PCA for 2D Visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)


plt.figure(figsize=(8, 6))
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=cluster_labels,
cmap='viridis', s=60, edgecolor='k')
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.title(f"Customer Segments (K-Means K={optimal_k})")
plt.colorbar(scatter, label="Cluster ID")
plt.show()


# 7. Hierarchical Clustering (Dendrogram)


linked = linkage(X_scaled, method='ward')


plt.figure(figsize=(12, 7))
dendrogram(linked, truncate_mode='level', p=5)
```

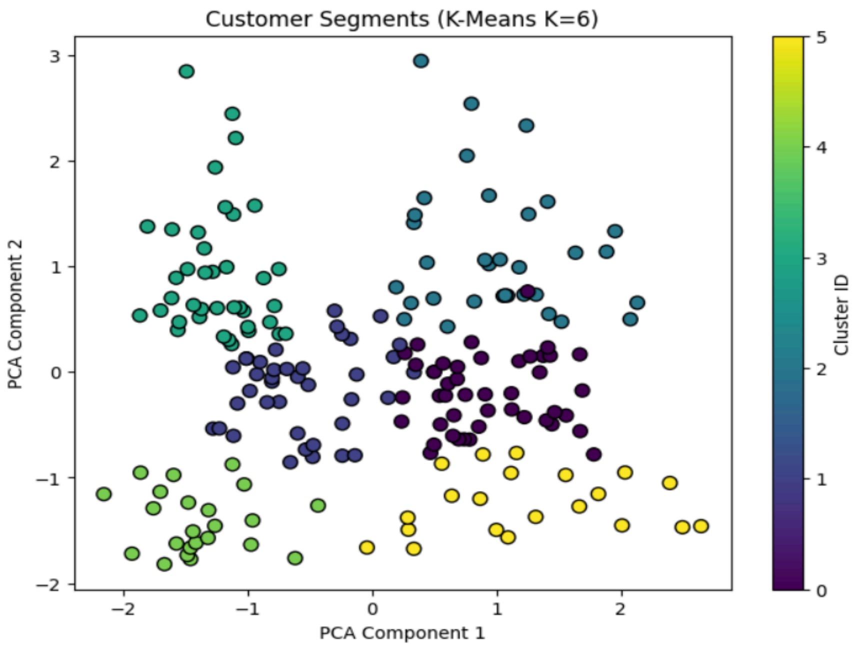
```
plt.title("Hierarchical Clustering Dendrogram (Ward Linkage)")

plt.xlabel("Customer Clusters/Indices")

plt.ylabel("Euclidean Distance")

plt.axhline(y=7, color='r', linestyle='--') # Visual threshold

plt.show()
```

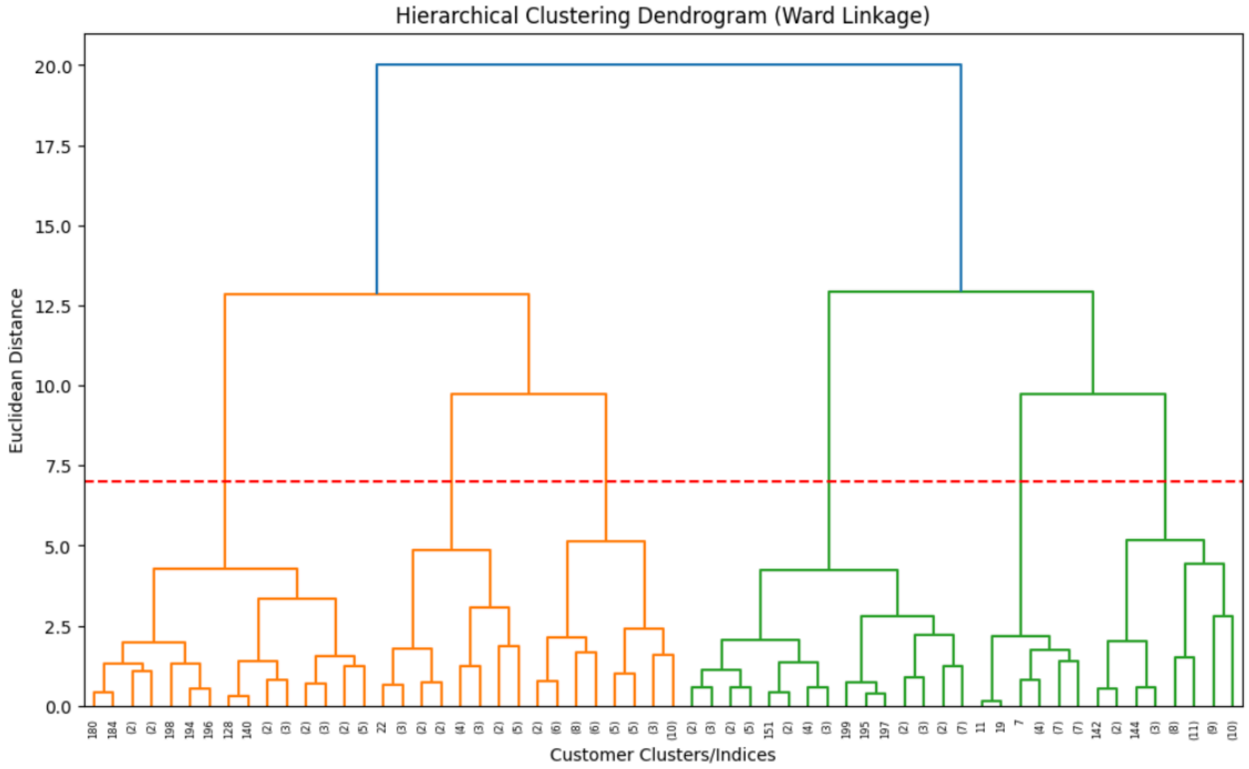


```
Columns used for clustering: ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
   Age  Annual Income (k$)  Spending Score (1-100)
0   19                  15                      39
1   21                  15                      81
2   20                  16                       6
3   23                  16                      77
4   31                  17                      40
```









Conclusion: K-Means clustering was effectively used to segment customers based on income, spending, and purchasing behavior. Hyperparameter tuning and validation identified two well-separated clusters, representing low-value and high-value customers. The results confirm the suitability of K-Means for practical customer segmentation and marketing analysis.

Hierarchical (Agglomerative) Clustering

Theory:

Hierarchical clustering is an **unsupervised machine learning technique** that organizes data points into a hierarchy of clusters by progressively combining similar groups or, less commonly, by recursively splitting larger groups. Unlike partition-based approaches such as K-Means, hierarchical clustering does **not require the number of clusters to be fixed in advance**. Instead, it constructs a hierarchical structure known as a **dendrogram**, which visually represents how individual data points and clusters are merged at different levels of similarity.

This method is particularly useful for **exploratory data analysis**, as it reveals relationships at multiple levels of granularity and allows analysts to choose an appropriate number of clusters by selecting a cut level on the dendrogram. The clustering process relies on a **distance metric** (commonly Euclidean distance for numerical data) to measure similarity between customers and a **linkage criterion** (such as single, complete, average, or Ward’s linkage) to determine how distances between clusters are computed.

Given a dataset with *n* customer records and *d* features, hierarchical clustering produces a nested sequence of cluster partitions that captures the underlying structure of customer similarities. In the context of the **Mall Customers dataset**, this approach helps uncover meaningful **customer segments** based on attributes such as annual income, spending score, and age, providing insights into purchasing behavior and market segmentation.

Limitations:

1. **High Computational and Memory Requirements:**
   Hierarchical clustering requires the computation and storage of a pairwise distance matrix, which leads to high time and space complexity. As the number of customers increases, memory usage and computational cost grow rapidly, making this method less suitable for very large datasets.

2. **Irreversibility of Merging Decisions:**
   In agglomerative clustering, once two clusters are merged, the decision cannot be undone. Incorrect merges occurring at early stages due to noise or data irregularities may propagate through the hierarchy and affect the final clustering structure.

3. **Sensitivity to Noise and Outliers:**
   Outliers can significantly influence distance calculations and linkage decisions. A few

extreme customer records may distort the dendrogram and result in less meaningful clusters or the formation of singleton clusters.

4. **Dependence on Distance Metric and Linkage Strategy:**
   The final cluster structure depends strongly on the selected distance measure and linkage method. Different combinations can yield different dendrograms and cluster interpretations, requiring careful experimentation and validation to obtain stable and meaningful customer segments.

Workflow:

1. **Data Collection:**
   The Mall Customers dataset (Mall_Customers.csv) is loaded into a Pandas DataFrame. The dataset includes customer attributes such as age, gender, annual income, and spending score, which are useful for retail customer segmentation.

2. **Data Cleaning and Preparation:**
   The customer identifier column is removed as it does not contribute to clustering. The dataset is checked for missing values and inconsistencies, and basic preprocessing is performed to ensure data quality.

3. **Feature Selection:**
   Relevant numerical features such as **Annual Income**, **Spending Score**, and optionally **Age** are selected, as these attributes best represent purchasing behavior and customer profiles. Categorical features (such as gender) may be encoded or excluded depending on the chosen distance metric.

4. **Feature Scaling:**
   All selected numerical features are standardized using **StandardScaler** to ensure that each variable contributes equally to distance calculations used in hierarchical clustering.

5. **Model Selection (Hierarchical Clustering):**
   Agglomerative hierarchical clustering is applied using an appropriate linkage method such as **Ward's linkage**, which tends to produce compact and well-separated clusters for numerical data.

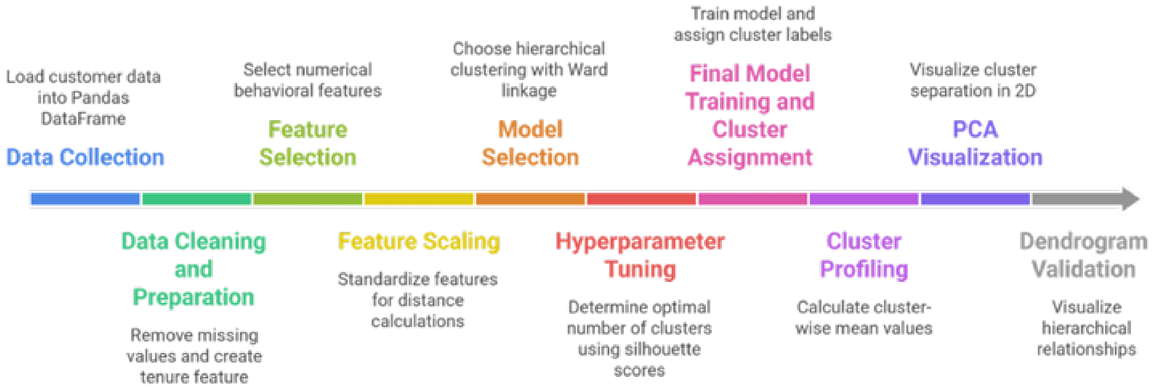
6. **Dendrogram Construction and Analysis:**
   A dendrogram is generated to visualize the hierarchical relationships among customers. The optimal number of clusters is determined by selecting a suitable cut level where large increases in linkage distance are observed.

7. **Final Cluster Formation:**
   Based on the dendrogram cut, the final number of customer clusters is selected, and each customer is assigned a cluster label accordingly.

8. **Cluster Profiling:**
   Cluster-wise summary statistics (e.g., mean income, average spending score, and age distribution) are computed to interpret behavioral differences across customer segments.

9. **Visualization:**
   Two-dimensional visualizations (such as Annual Income vs. Spending Score scatter plots) are used to illustrate cluster separation and support qualitative interpretation of customer groups.

10. **Interpretation and Business Insights:**
    The identified clusters are analyzed to derive actionable insights for **customer segmentation, targeted marketing strategies, and personalized service design** in retail environments.


Load customer data into Pandas DataFrame
Data Collection
Data Cleaning and Preparation
Remove missing values and create tenure feature
Select numerical behavioral features
Feature Selection
Feature Scaling
Standardize features for distance calculations
Choose hierarchical clustering with Ward linkage
Model Selection
Hyperparameter Tuning
Determine optimal number of clusters using silhouette scores
Train model and assign cluster labels
Final Model Training and Cluster Assignment
Cluster Profiling
Calculate cluster-wise mean values
Visualize cluster separation in 2D
PCA Visualization
Dendrogram Validation
Visualize hierarchical relationships

Performance Analysis

The performance of the **Hierarchical (Agglomerative) Clustering** model on the **Customer Personality Analysis dataset** is evaluated using **internal validation techniques and structural visualization**, since unsupervised learning does not involve ground-truth labels. The evaluation focuses on **cluster separation, stability, and interpretability** using silhouette-based tuning, dendrogram analysis, PCA visualization, and cluster profiling.

1. Optimal Number of Clusters Selection

Silhouette score analysis is used to determine the optimal number of clusters. Among the tested values of K (ranging from 2 to 10), **K = 2** yields the highest silhouette score, indicating the best

balance between intra-cluster cohesion and inter-cluster separation.
The **dendrogram generated using Ward linkage** supports this choice by showing a large vertical distance between the final merge steps. This significant gap indicates that merging the two major clusters would cause a sharp increase in within-cluster variance. Hence, cutting the dendrogram at this level produces a **natural and stable two-cluster structure**.

2. Cluster Separation and Visualization (PCA Analysis)

To visualize cluster separation, **Principal Component Analysis (PCA)** is applied to project the high-dimensional feature space into two dimensions. The PCA scatter plot shows **clear visual separation between the two hierarchical clusters**, primarily along the first principal component.
The minimal overlap between clusters suggests strong **inter-cluster dissimilarity**, confirming that the hierarchical algorithm has discovered meaningful structure in the data rather than arbitrary groupings.

3. Dendrogram Structural Validation

The dendrogram provides a hierarchical view of customer relationships and confirms that the final clusters are **internally cohesive**. The large linkage distance observed at the final merge stage reflects strong separation between the two clusters, validating that the selected clustering solution is **structurally meaningful and not artificially forced**.

Hyperparameter Tuning

Hierarchical clustering is sensitive to **hyperparameter selection**, particularly the number of clusters and the linkage strategy, which directly influence how data points are grouped and merged. Inappropriate parameter choices can result in unstable or poorly separated clusters.
Since hierarchical clustering is an **unsupervised learning method**, internal validation metrics are used instead of classification-based performance measures. In this implementation, multiple configurations are evaluated to ensure meaningful segmentation and strong cluster structure.

1. Number of Clusters (K)

The number of clusters is tuned by training Agglomerative Hierarchical Clustering models for values of **K ranging from 2 to 10**. For each configuration, cluster assignments are evaluated using the **Silhouette Score**, which measures how well each data point fits within its own cluster compared to neighboring clusters.
The value of **K that yields the highest silhouette score is selected as optimal**, indicating strong intra-cluster similarity and good inter-cluster separation.

2. Linkage Method

The model uses **Ward linkage**, which merges clusters by minimizing the increase in **within-cluster variance** at each step. Ward's method is well-suited for **numerical customer data** and typically produces compact and well-separated clusters. This linkage strategy enhances **cluster stability and interpretability**, making the resulting customer segments more meaningful for analysis.

3. Distance Metric

Ward linkage implicitly relies on **Euclidean distance** to measure similarity between data points. Since clustering decisions are based on geometric proximity, all features are **standardized prior to clustering** to ensure that variables with larger numerical ranges do not dominate the distance calculations.

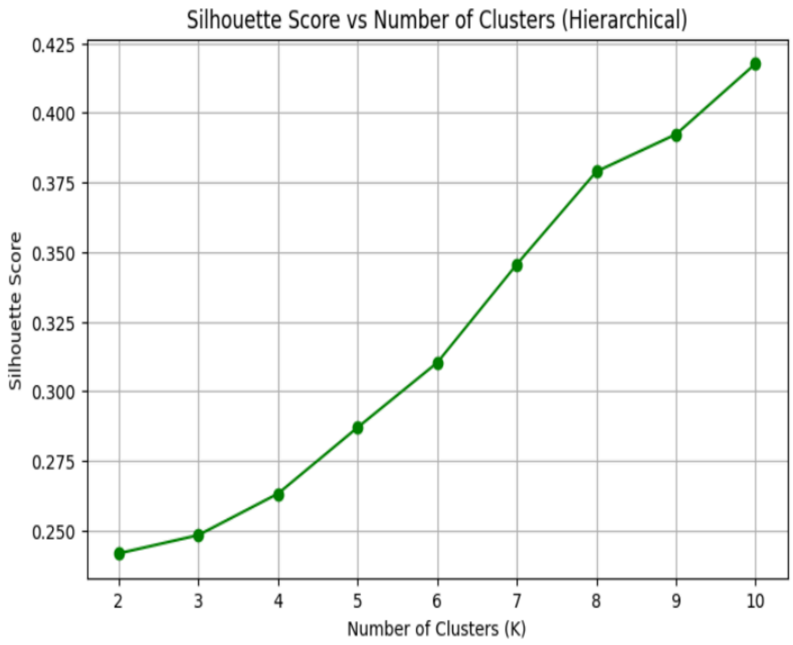
4. Model Validation

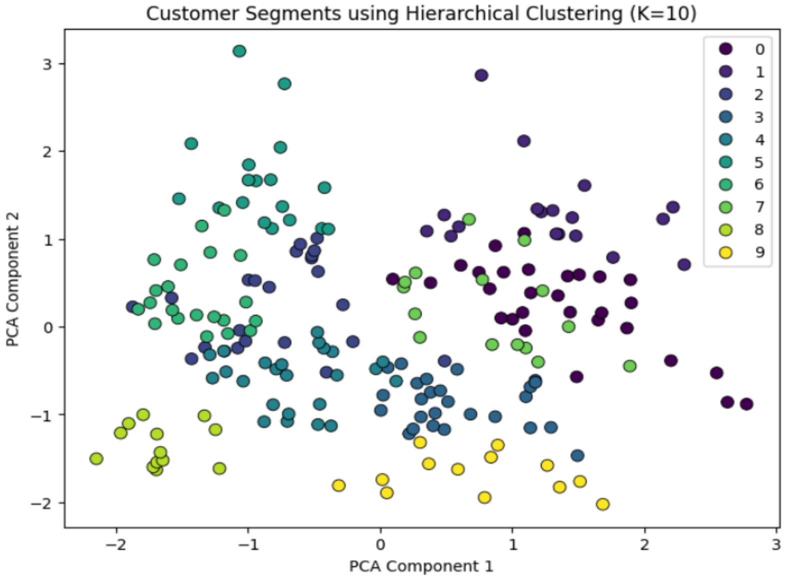
In addition to silhouette-based optimization, the **dendrogram structure** is used as a qualitative validation tool. A large vertical gap at the final merge stage indicates that the selected number of clusters corresponds to a **natural partition in the data** rather than an arbitrary cut, thereby supporting the robustness of the clustering solution.

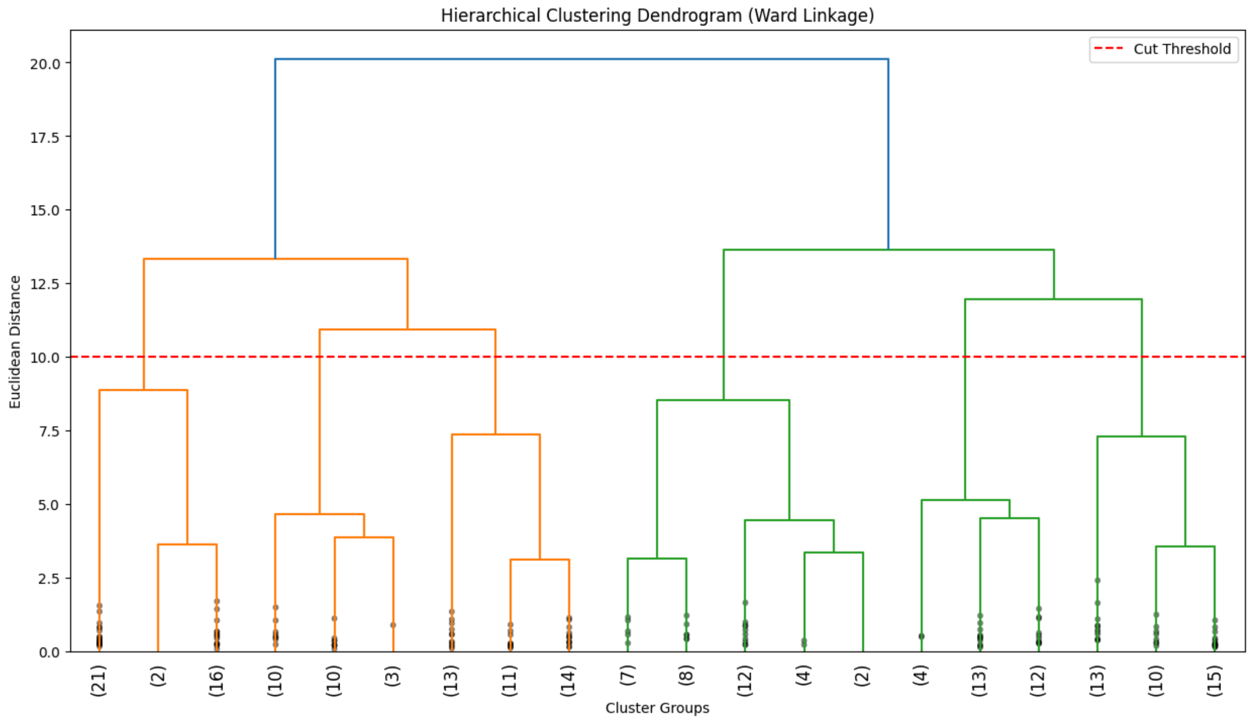
Code & Output:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
from scipy.cluster.hierarchy import dendrogram, linkage

# 1. Load Dataset directly for Google Colab
url =
"https://raw.githubusercontent.com/SteffiPeTaffy/machineLearningAZ/master/Machi
ne%20Learning%20A-Z%20Template%20Folder/Part%204%20-%20Clustering/Section%2025%
20-%20Hierarchical%20Clustering/Mall_Customers.csv"
df = pd.read_csv(url)

# 2. Preprocessing
# Drop ID and encode Gender (0 for Female, 1 for Male)
df['Gender'] = df['Genre'].map({'Male': 1, 'Female': 0})
df = df.drop(columns=['CustomerID', 'Genre'], errors='ignore')

# Handle missing values (though this dataset is usually clean)
df.fillna(df.mean(), inplace=True)

# Dataset has 200 rows, so we use all of them instead of sampling 600
df_sample = df

scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(df_sample)

# 3. Finding Best K for Hierarchical Clustering
cluster_range = range(2, 11)
silhouette_scores = []

for k in cluster_range:
    model = AgglomerativeClustering(n_clusters=k, linkage='ward')
    labels = model.fit_predict(X_scaled)
    score = silhouette_score(X_scaled, labels)
    silhouette_scores.append(score)

# Plot Silhouette vs K

plt.figure(figsize=(8,5))
plt.plot(cluster_range, silhouette_scores, marker='o', color='green')
plt.xlabel("Number of Clusters (K)")
plt.ylabel("Silhouette Score")
plt.title("Silhouette Score vs Number of Clusters (Hierarchical)")
plt.grid(True)
plt.show()

best_k = cluster_range[np.argmax(silhouette_scores)]
print(f"Best Number of Clusters: {best_k}")
print(f"Best Silhouette Score: {round(max(silhouette_scores), 4)}")

# 4. Final Hierarchical Clustering
hc = AgglomerativeClustering(n_clusters=best_k, linkage='ward')
final_labels = hc.fit_predict(X_scaled)

# 5. PCA for 2D Visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.figure(figsize=(8,6))
sns.scatterplot(
    x=X_pca[:,0],
    y=X_pca[:,1],
    hue=final_labels,
    palette='viridis',
    s=60,
    edgecolor='black',
    legend='full'
)
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.title(f"Customer Segments using Hierarchical Clustering (K={best_k})")
plt.show()

# 6. Dendrogram (Full Dataset)
linked = linkage(X_scaled, method='ward')


plt.figure(figsize=(15,8))
dendrogram(
    linked,
    truncate_mode='lastp', # Shows only the last p merged clusters
    p=20,                   # Better readability for 200 samples
    leaf_rotation=90,
    leaf_font_size=12,
    show_contracted=True
)
plt.axhline(y=10, color='red', linestyle='--', label='Cut Threshold')
plt.xlabel("Cluster Groups")
plt.ylabel("Euclidean Distance")
plt.title("Hierarchical Clustering Dendrogram (Ward Linkage)")
plt.legend()
plt.show()
```


Silhouette Score vs Number of Clusters (Hierarchical)


Customer Segments using Hierarchical Clustering (K=10)



Conclusion

Hierarchical (Agglomerative) Clustering successfully identified **two distinct and well-separated customer segments** based on key attributes such as income, spending behavior, purchasing activity, and engagement patterns. The combination of **silhouette score–based tuning and dendrogram analysis** confirmed that the chosen number of clusters provides a **stable and meaningful partition** of the customer dataset.

The clear separation observed in PCA visualizations and the strong structural gaps in the dendrogram indicate that the model captured **natural groupings present in the data** rather than producing arbitrary clusters. These findings demonstrate that hierarchical clustering is an effective and interpretable technique for **practical customer segmentation**, making it valuable for applications such as targeted marketing, customer profiling, and strategic decision-making in real-world business scenarios.