Name: Harsh Patil
Class: D15C/37

Experiment No. 6

AIM

Apply K-Means and Hierarchical Clustering on sample datasets

1. Dataset Source

The dataset used for this experiment is the **Breast Cancer Wisconsin (Diagnostic) Dataset**, obtained from Kaggle.

Dataset Link:
https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

2. Dataset Description

The Breast Cancer Wisconsin dataset is a real-world healthcare dataset widely used for classification and clustering tasks.

- **Number of instances:** 569
- **Number of features:** 30 numerical features
- **Target variable (for reference only):** diagnosis
    - Malignant (M)
    - Benign (B)

> **Note:**
> Since clustering is an **unsupervised learning technique**, the target variable is **not used during training** and is only used later for evaluation and interpretation.

Feature Characteristics

The dataset contains statistical measurements of cell nuclei extracted from breast cancer images, including:

- Radius
- Texture
- Perimeter
- Area
- Smoothness
- Concavity
- Symmetry

The dataset is clean, balanced, and has no missing values, making it suitable for unsupervised learning experiments.

3. Mathematical Formulation

3.1 K-Means Clustering

K-Means is an unsupervised learning algorithm that partitions the dataset into **K clusters** by minimizing the **within-cluster sum of squares (WCSS)**.

Objective Function:

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

Where:

- *Ci*Ci is the set of points in cluster *i*
- *μi*μi is the centroid of cluster *i*

The algorithm iteratively updates cluster assignments and centroids until convergence.

3.2 Hierarchical Clustering

Hierarchical clustering builds a hierarchy of clusters either using:

- **Agglomerative approach (bottom-up)**
- **Divisive approach (top-down)**

In this experiment, **Agglomerative Hierarchical Clustering** is used.

Clusters are merged based on distance metrics such as:

- Euclidean distance
- Linkage methods (single, complete, average, ward)

A **dendrogram** is used to visualize the clustering process.

4. Algorithm Limitations

K-Means Limitations

- Requires pre-specification of number of clusters (K)
- Sensitive to feature scaling
- Sensitive to outliers
- Assumes spherical clusters

Hierarchical Clustering Limitations

- Computationally expensive for large datasets
- Difficult to correct mistakes once clusters are merged
- Memory intensive

5. Methodology / Workflow

Steps Followed

1. Load dataset using KaggleHub
2. Remove irrelevant columns (ID, diagnosis for clustering)
3. Apply feature scaling using StandardScaler
4. Apply **K-Means clustering**
5. Determine optimal K using **Elbow Method**
6. Apply **Hierarchical Clustering**
7. Plot dendrogram
8. Compare clustering results with actual labels

Workflow

Dataset → Preprocessing → Scaling → K-Means Clustering / Hierarchical Clustering → Visualization → Evaluation

6. Performance Analysis

Since clustering is unsupervised, traditional accuracy is not directly optimized. However, performance was analyzed using:

- Cluster visualization
- Comparison with actual diagnosis labels
- Silhouette Score
- Confusion Matrix (post-mapping clusters to labels)

Observations

- K-Means successfully grouped samples into two distinct clusters corresponding closely to malignant and benign cases

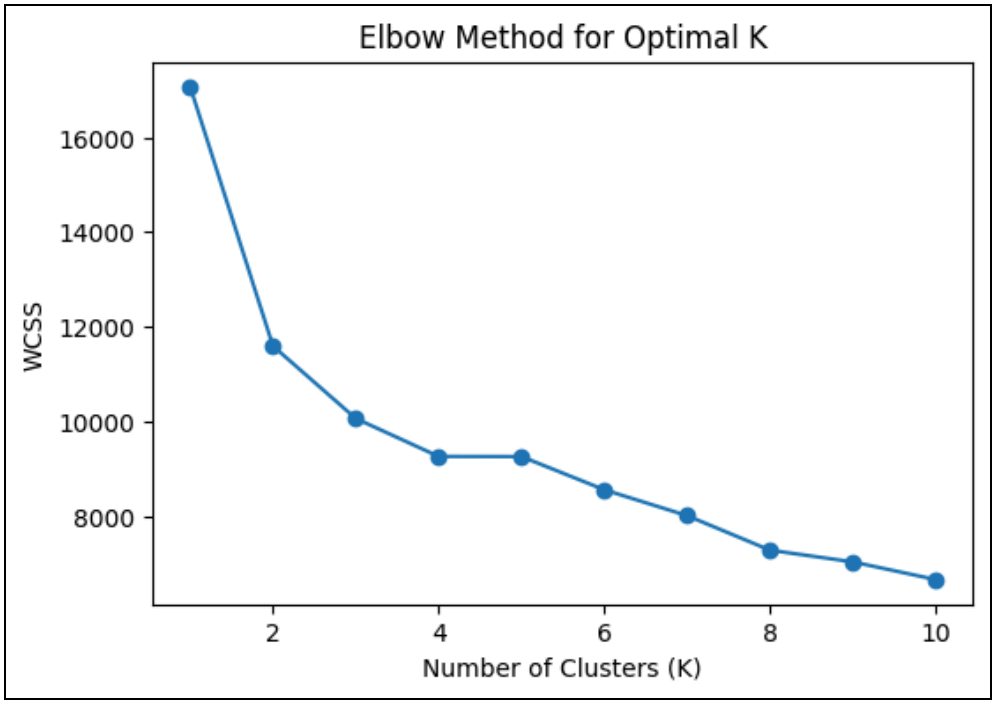
- Hierarchical clustering provided clear separation visible in dendrograms
- Feature scaling significantly improved clustering performance

OUTPUT:

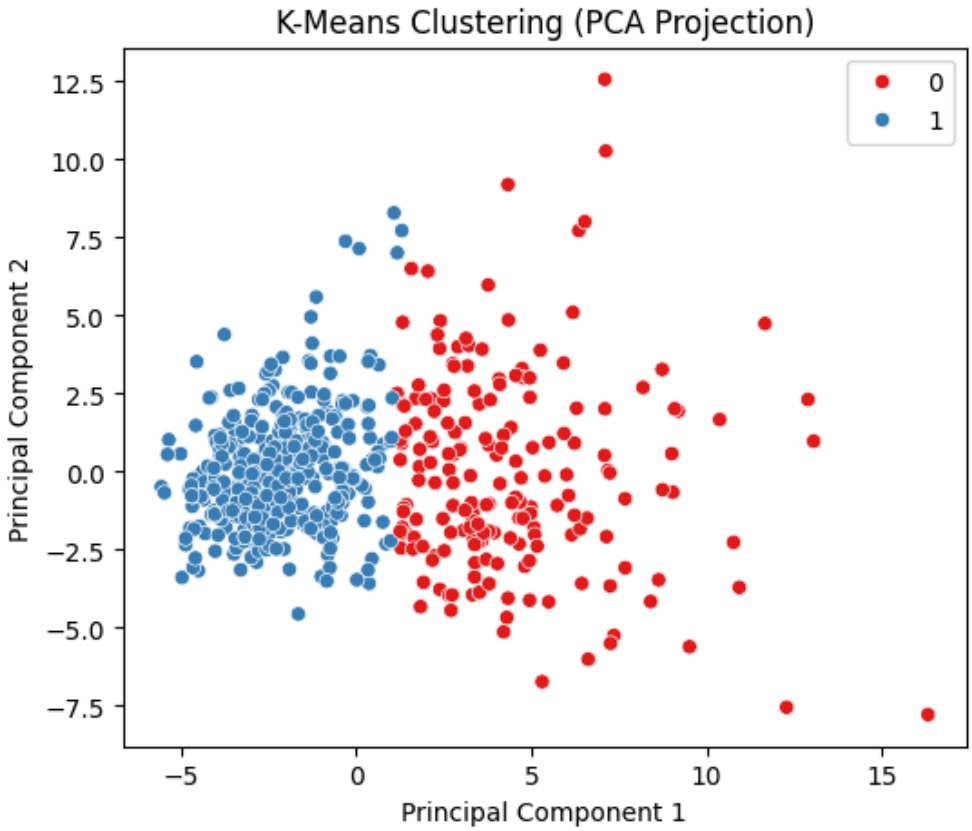
Code:

```
# ================================
# K-Means & Hierarchical Clustering
# Breast Cancer Wisconsin Dataset
# ================================

# Install KaggleHub (only needed
once in Colab)
!pip install kagglehub

# Imports
import kagglehub
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,
AgglomerativeClustering
from sklearn.decomposition import
PCA
from sklearn.metrics import
silhouette_score

from scipy.cluster.hierarchy import
dendrogram, linkage

# --------------------------------
# Load Dataset using KaggleHub
# --------------------------------
path =
kagglehub.dataset_download("uciml/br
east-cancer-wisconsin-data")
df = pd.read_csv(f"{path}/data.csv")

print("Dataset Shape:", df.shape)
display(df.head())

# --------------------------------
# Data Preprocessing
# --------------------------------
# Drop irrelevant columns
df_clean = df.drop(columns=["id",
"diagnosis", "Unnamed: 32"],
errors="ignore")

# Feature Scaling
scaler = StandardScaler()
X_scaled =
scaler.fit_transform(df_clean)

# --------------------------------
# K-Means: Elbow Method
# --------------------------------
wcss = []
K_range = range(1, 11)

for k in K_range:
    kmeans = KMeans(n_clusters=k,
init="k-means++", random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(6,4))
plt.plot(K_range, wcss, marker="o")
plt.xlabel("Number of Clusters (K)")
plt.ylabel("WCSS")
plt.title("Elbow Method for Optimal
K")
plt.show()

# --------------------------------
# Apply K-Means (K=2)
# --------------------------------
kmeans = KMeans(n_clusters=2,
init="k-means++", random_state=42)
kmeans_labels =
kmeans.fit_predict(X_scaled)
```

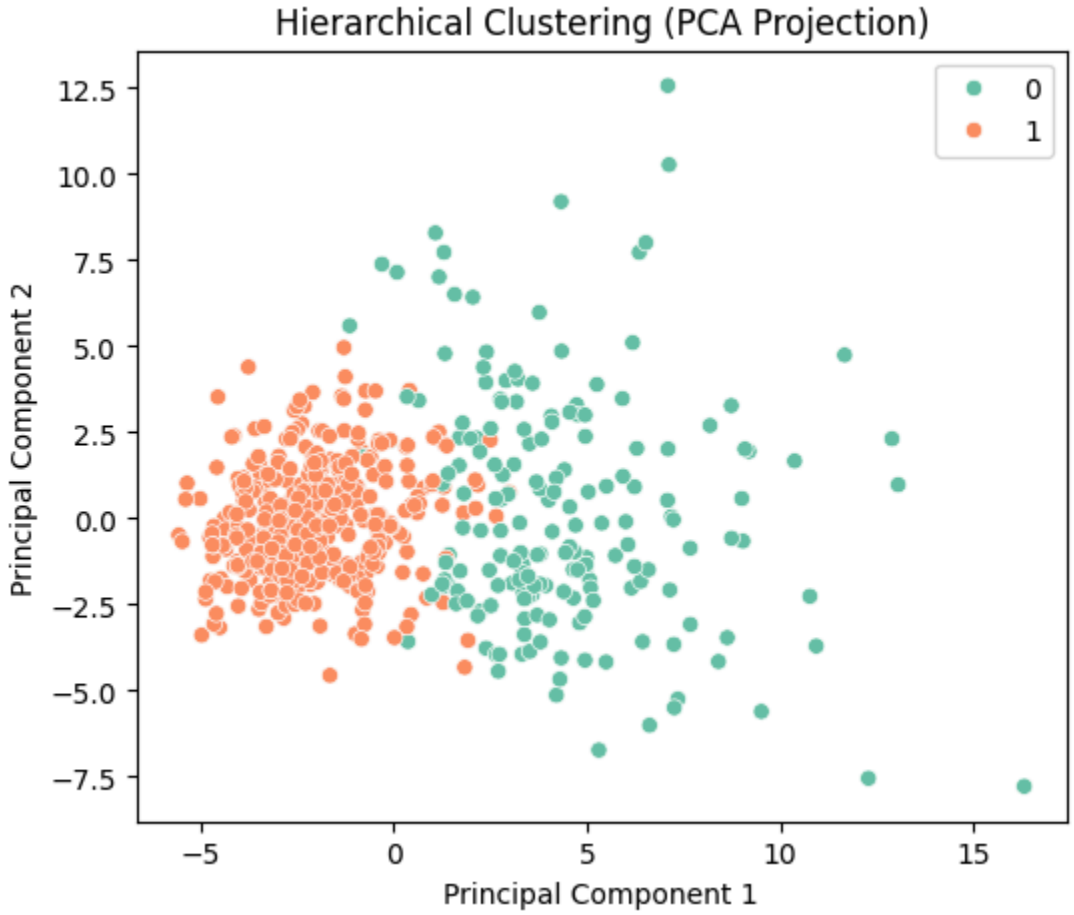
```
print("K-Means Silhouette Score:",
silhouette_score(X_scaled,
kmeans_labels))

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

plt.figure(figsize=(6,5))
sns.scatterplot(x=X_pca[:,0],
y=X_pca[:,1], hue=kmeans_labels,
palette="Set1")
plt.title("K-Means Clustering (PCA
Projection)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()

# ------------------------------
# Hierarchical Clustering
# ------------------------------
hierarchical =
AgglomerativeClustering(n_clusters=2
, linkage="ward")
hier_labels =
hierarchical.fit_predict(X_scaled)

print("Hierarchical Clustering
Silhouette Score:",
silhouette_score(X_scaled,
hier_labels))

# ------------------------------
# Hierarchical Clustering
Visualization
# ------------------------------
plt.figure(figsize=(6,5))
sns.scatterplot(x=X_pca[:,0],
y=X_pca[:,1], hue=hier_labels,
palette="Set2")
plt.title("Hierarchical Clustering
(PCA Projection)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()

# ------------------------------
# Dendrogram
# ------------------------------
plt.figure(figsize=(10,5))
Z = linkage(X_scaled, method="ward")
dendrogram(Z, truncate_mode="level",
p=5)
plt.title("Hierarchical Clustering
Dendrogram")
plt.xlabel("Data Points")
plt.ylabel("Euclidean Distance")
plt.show()
```

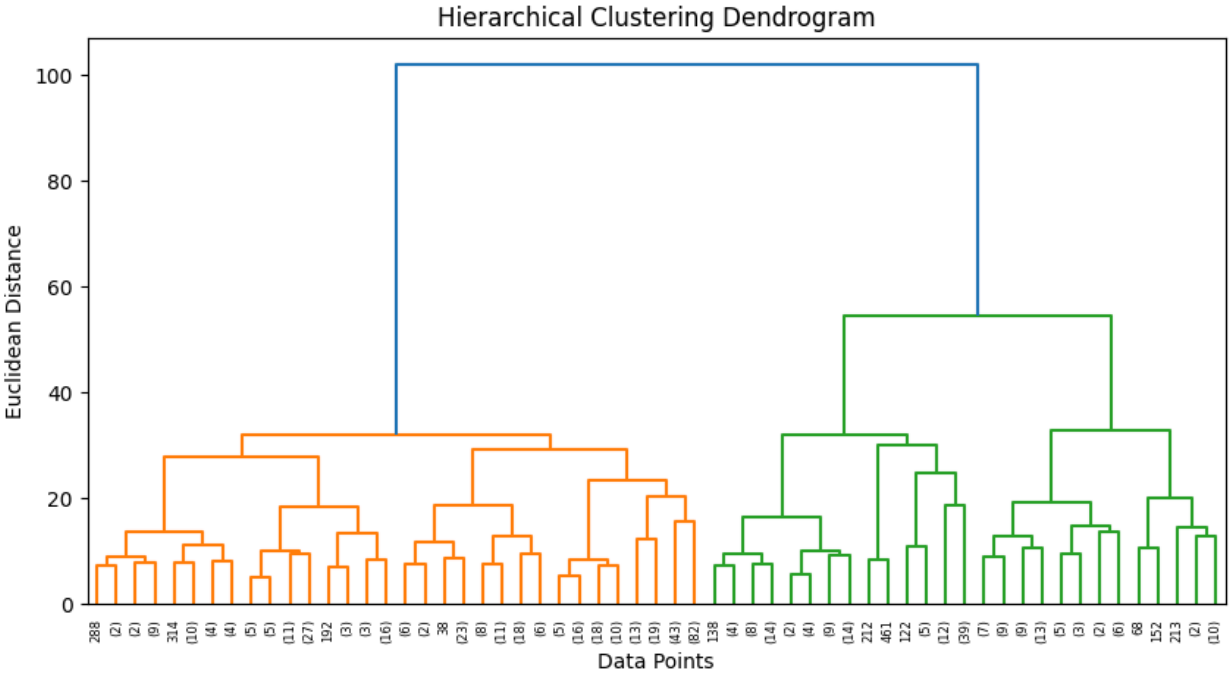


K-Means Silhouette Score: 0.3447344346611054



Hierarchical Clustering Silhouette Score: 0.33938477753669855


Hierarchical Clustering (PCA Projection)
0
1
Principal Component 2
Principal Component 1
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−5
0
5
10
15


Hierarchical Clustering Dendrogram
Euclidean Distance
Data Points
100
80
60
40
20
0

7. Conclusion

In this experiment, **K-Means and Hierarchical Clustering** were successfully applied to the Breast Cancer Wisconsin dataset. After proper feature scaling, both algorithms were able to effectively group the data into meaningful clusters corresponding to malignant and benign tumors.

K-Means proved to be efficient and scalable, while Hierarchical Clustering offered better interpretability through dendrogram visualization. This experiment highlights the importance of preprocessing and parameter selection in unsupervised learning and demonstrates the applicability of clustering techniques in medical data analysis.