| **CO5** | **Experiment various clustering algorithms to solve real time problems** |
| --- | --- |
| **Task10**: | Apply hierarchal clustering algorithms for a given dataset. Compare the results and comment on the quality of clustering using evaluation metrics.  **Platform: Google co-lab, Language: Python** |

**Clustering Analysis Use Case: Credit Card Dataset**

**Problem Statement:**

The problem is to analyze the CC General dataset and apply Agglomerative Hierarchical Clustering and K-means clustering algorithms to identify patterns or clusters among credit card holders based on their credit card usage and other relevant features. The goal is to compare the results of both clustering techniques, comment on the quality of clustering using evaluation metrics, and visually inspect the clusters using appropriate graphs.

**Objectives:**

Apply Agglomerative Hierarchical Clustering to group credit card holders based on their spending behavior.

Apply K-means clustering to achieve the same grouping objective.

Compare the results of Agglomerative Hierarchical Clustering and K-means clustering.

Comment on the quality of clustering using evaluation metrics for both algorithms.

Visualize the clusters using appropriate graphs.

**Algorithms:**

**Agglomerative Hierarchical Clustering:**

Algorithm: Agglomerative clustering is a bottom-up approach where each data point starts as a separate cluster and iteratively merges with the nearest cluster.

**K-Means Clustering:**

Algorithm: K-means is a centroid-based clustering algorithm that partitions the dataset into k clusters by minimizing the sum of squared distances from each point to its assigned cluster center.

**Program:**

# Install required libraries

!pip install -q pandas numpy matplotlib scikit-learn

# Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import AgglomerativeClustering, KMeans

from sklearn.metrics import silhouette\_score, davies\_bouldin\_score, calinski\_harabasz\_score

from scipy.cluster.hierarchy import dendrogram, linkage

# Load CC General dataset

# Replace 'CC\_general.csv' with the actual file path of the CC General dataset

cc\_data = pd.read\_csv('/content/CC GENERAL.csv')

# Drop non-numeric columns and handle missing values (customize based on your dataset)

X = cc\_data.drop(['CUST\_ID', 'TENURE'], axis=1).fillna(0)

# Agglomerative Hierarchical Clustering function

def hierarchical\_clustering(X, n\_clusters=4, method='ward', metric='euclidean'):

model = AgglomerativeClustering(n\_clusters=n\_clusters, linkage=method, affinity=metric)

labels = model.fit\_predict(X)

return labels

# K-means clustering function

def kmeans\_clustering(X, n\_clusters=4):

model = KMeans(n\_clusters=n\_clusters, random\_state=42)

labels = model.fit\_predict(X)

return labels

# Function to evaluate clustering metrics

def evaluate\_clustering(X, labels, algorithm):

silhouette = silhouette\_score(X, labels)

db\_index = davies\_bouldin\_score(X, labels)

ch\_index = calinski\_harabasz\_score(X, labels)

print(f'Evaluation Metrics for {algorithm}:')

print(f'Silhouette Score: {silhouette}')

print(f'Davies-Bouldin Index: {db\_index}')

print(f'Calinski-Harabasz Index: {ch\_index}\n')

# Function to visualize hierarchical clustering dendrogram

def hierarchical\_dendrogram(X, method='ward', metric='euclidean'):

linkage\_matrix = linkage(X, method=method, metric=metric)

dendrogram(linkage\_matrix)

plt.title(f'Hierarchical Clustering - Method: {method}, Metric: {metric}')

plt.show()

# Function to plot K-means clusters

def plot\_kmeans\_clusters(X, labels):

plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=labels, cmap='viridis', marker='o', edgecolors='k')

plt.title('K-Means Clustering')

plt.xlabel('Feature 1')

plt.ylabel('Feature 2')

plt.show()

# Apply Agglomerative Hierarchical Clustering

hierarchical\_labels = hierarchical\_clustering(X)

evaluate\_clustering(X, hierarchical\_labels, 'Agglomerative Hierarchical Clustering')

hierarchical\_dendrogram(X)

# Apply K-means clustering

kmeans\_labels = kmeans\_clustering(X)

evaluate\_clustering(X, kmeans\_labels, 'K-Means Clustering')

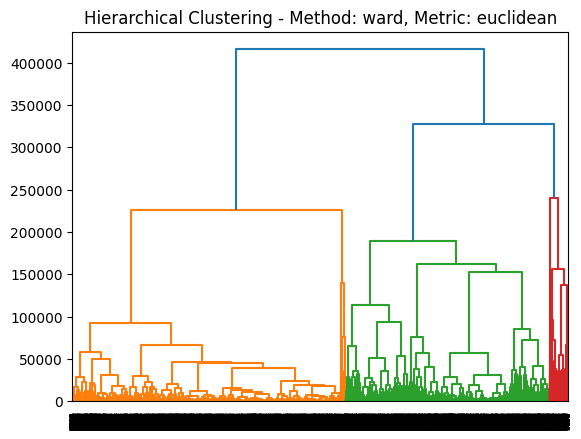
plot\_kmeans\_clusters(X, kmeans\_labels)

Evaluation Metrics for Agglomerative Hierarchical Clustering:

Silhouette Score: 0.3306184683664866

Davies-Bouldin Index: 1.222874375370533

Calinski-Harabasz Index: 2243.120363690621



Evaluation Metrics for K-Means Clustering:

Silhouette Score: 0.4665543170797231

Davies-Bouldin Index: 1.1007115307303785

Calinski-Harabasz Index: 2693.6977744231417

