Assignment No: Title: Implement Agglomerative hierarchical clustering Algorithm.

Problem Definition:

ImplementAgglomerative hierarchical clustering. (Use python or beautiful soup forimplementation).

Outcome:

Students will be able to,

1. The horizontal lines represent the order in which clusters were merged, from bottom to top.
2. The vertical lines represent individual data points at the leaves of the tree .

Theory:

Introduction to Agglomerativehierarchicalclustering Algorithm:

Hierarchical clustering is a popular method for cluster analysis. It creates a tree-like hierarchy of clusters, which can be visualized as a dendrogram. Agglomerative hierarchical clustering is one of the two main approaches, where each data point starts in its own cluster and, at each step, the two closest clusters are merged into a single cluster until only one cluster remains.

Here's a practical write-up of the Agglomerative Hierarchical Clustering algorithm, along with Python code and a sample dataset.

Hierarchical clustering is a technique used to group similar data points into clusters or groups. Agglomerative hierarchical clustering, which we'll discuss in this write-up, is a bottom-up approach where each data point initially forms its own cluster, and these clusters are successively merged based on similarity until a single cluster is formed.

Algorithm:

1.Initialization:

Start with each data point as its own cluster. For N data points, you have N clusters.

2.Compute Pairwise Distances:

Calculate the pairwise distances (similarity) between all clusters. The distance can be based on different linkage methods, such as single, complete, or average linkage.

3.Find Closest Clusters:

Identify the two clusters that are closest to each other based on the computed distance. These clusters will be merged in the next step.

4.Merge Clusters:

Combine the two closest clusters into a single cluster. This reduces the total number of clusters by one.

5.Update Distance Matrix:

Recalculate the distances between the newly formed cluster and all other clusters, using the chosen linkage method. This step is crucial for updating the hierarchical structure.

6.Repeat:

Continue steps 3-5 until only one cluster remains, which contains all data points. This forms a hierarchical tree (dendrogram) showing the merging process.

7.Dendrogram Visualization:

Optionally, you can visualize the dendrogram, which illustrates the hierarchy of clusters and helps in determining the number of clusters at various levels.

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

cd C:\Users\Dev\Desktop\Kaggle\Credit_Card

X = pd.read_csv('CC_GENERAL.csv')

X = X.drop('CUST_ID', axis = 1)

X.fillna(method ='ffill', inplace = True)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_normalized = normalize(X_scaled)

X_normalized = pd.DataFrame(X_normalized)


pca = PCA(n_components = 2)
X_principal = pca.fit_transform(X_normalized)
X_principal = pd.DataFrame(X_principal)
X_principal.columns = ['P1', 'P2']

plt.figure(figsize =(8, 8))
plt.title('Visualising the data')
Dendrogram = shc.dendrogram((shc.linkage(X_principal, method ='ward')))

ac2 = AgglomerativeClustering(n_clusters = 2)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
            c = ac2.fit_predict(X_principal), cmap ='rainbow')
plt.show()

ac3 = AgglomerativeClustering(n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
            c = ac3.fit_predict(X_principal), cmap ='rainbow')
plt.show()

ac4 = AgglomerativeClustering(n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
```

```
                c = ac4.fit_predict(X_principal), cmap ='rainbow')
plt.show()
ac5 = AgglomerativeClustering(n_clusters = 5)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
                c = ac5.fit_predict(X_principal), cmap ='rainbow')
plt.show()

ac6 = AgglomerativeClustering(n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
                c = ac6.fit_predict(X_principal), cmap ='rainbow')
plt.show()

k = [2, 3, 4, 5, 6]

silhouette_scores = []
silhouette_scores.append(
        silhouette_score(X_principal, ac2.fit_predict(X_principal)))
silhouette_scores.append(
        silhouette_score(X_principal, ac3.fit_predict(X_principal)))
silhouette_scores.append(
        silhouette_score(X_principal, ac4.fit_predict(X_principal)))
silhouette_scores.append(
        silhouette_score(X_principal, ac5.fit_predict(X_principal)))
silhouette_scores.append(
        silhouette_score(X_principal, ac6.fit_predict(X_principal)))

plt.bar(k, silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('S(i)', fontsize = 20)
plt.show()
```

Conclusion:-

Agglomerative hierarchical clustering is a powerful method for identifying hierarchical structures within your data. It's widely used in various fields such as biology, social sciences, and marketing for pattern recognition and structure analysis.

Performance & Understanding	Innovation	Timely Completion	Total	Sign & Date
3	1	1	5	