EXPERIMENT 5 – CLUSTERING

AIM: Introduce the basic concepts and methods for data clustering, including an overview of basic cluster analysis and hierarchical methods. It also introduces methods for the evaluation of clustering.

SOFTWARE REQUIRED:

Spyder IDE 5.1.5

Anaconda3 2021.11 (Python 3.9.7 64-bit)

Anaconda Inc., 2021.11

DATA SET: Real Estate Data Set

PYTHON CODE:

```
# Importing Libraries:-
import matplotlib.pyplot as plt # Provides an implicit way of plotting
import pandas as pd  # Library for working with data sets
import scipy.cluster.hierarchy as shc # These functions cut hierarchical
clusterings into flat clusterings or find the roots of the forest formed by a
cut by providing the flat cluster ids of each observation.
from sklearn.cluster import AgglomerativeClustering # Recursively merges pair
of clusters of sample data; uses linkage distance.
from sklearn.cluster import KMeans # K-Means clustering

import warnings
warnings.filterwarnings('ignore') # Never print matching warnings

def print_csv_file(df, heading):
    print("\n")
    print('{:s}'.format('\u0332'.join(heading.center(100))))
    print(df)

# The K-Means Clustering Method:-
def k_means_clustering():

    # Finding the optimal number of clusters using the elbow method:-
    wcss_list= [] # Initialize the list for the values of WCSS

    for i in range(1, 11):
        kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42) # K-
Means clustering
```

```
        kmeans.fit(x) # Compute k-means clustering
        wcss_list.append(kmeans.inertia_) # Sum of squared distances of
samples to their closest cluster centre, weighted by the sample weights if
provided.

    plt.xlabel("Number of clusters(k)")
    plt.ylabel("wcss_list")
    plt.title("The Elbow Method Graph")
    plt.plot(range(1, 11), wcss_list)
    plt.grid(True)
    plt.show()

    print("\nFrom the above plot, we can see the elbow point is at 3. Hence,
the number of clusters here will be 3.")

    # Training the K-means algorithm on the training dataset:-
    kmeans = KMeans(n_clusters=3, init='k-means++', random_state= 42) # K-
Means clustering
    y_predict = kmeans.fit_predict(x) # Compute cluster centers and predict
cluster index for each sample

    # Visualizing the clusters:-
    plt.xlabel("Age of House in Year(s)")
    plt.ylabel("House Price per Local Unit Area")
    plt.title("Clusters of Customers")

    plt.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c =
'blue', label = 'Cluster 1') # For first cluster
    plt.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c =
'green', label = 'Cluster 2') # For second cluster
    plt.scatter(x[y_predict == 2, 0], x[y_predict == 2, 1], s = 100, c =
'red', label = 'Cluster 3') # For third cluster
    plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s = 300, c = 'yellow', label = 'Centroid')

    plt.legend()
    plt.show()

# Hierarchical Clustering - Dendrogram:-
def hierarchial_clustering_dendrogram():

    # Finding the optimal number of clusters using the Dendrogram:-
    plt.xlabel("Customers")
    plt.ylabel("Euclidean Distances") # It is a metric used to compute the
linkage.
    plt.title("Dendrogram Plot")
```

```
    shc.dendrogram(shc.linkage(x, method="ward")) # This method is the popular
linkage method that we have already used for creating the Dendrogram. It
reduces the variance in each cluster.
    plt.grid(True)
    plt.show()

    print("\nUsing this Dendrogram, we will now determine the optimal number
of clusters for our model. For this, we will find the maximum vertical
distance that does not cut any horizontal bar. Accordingly, the number of
clusters will be 3.")

    # Training the hierarchical clustering model:-
    hc = AgglomerativeClustering(n_clusters=5, affinity='euclidean',
linkage='ward') # Recursively merges pair of clusters of sample data; uses
linkage distance.
    y_predict = hc.fit_predict(x) # Compute cluster centers and predict
cluster index for each sample

    # Visualizing the clusters:-
    plt.xlabel("Age of House in Year(s)")
    plt.ylabel("House Price per Local Unit Area")
    plt.title("Clusters of Customers")

    plt.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c =
'blue', label = 'Cluster 1') # For first cluster
    plt.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c =
'green', label = 'Cluster 2') # For second cluster
    plt.scatter(x[y_predict == 2, 0], x[y_predict == 2, 1], s = 100, c =
'red', label = 'Cluster 3') # For third cluster

    plt.legend()
    plt.show()

# Driver Code: main() ; Execution starts here.

print("No - Serial Number")
print("X2 - Age of House in Year(s)")
print("X3 - Distance to Nearest MRT Station in Meter(s)")
print("X4 - Number of Convenience Stores Within Walking Distance")
print("X5 - Latitude Coordinates")
print("X5 - Longitude Coordinates")
print("Y - House Price per Local Unit Area")

# Importing the data set:-
heading = "Original Data Set"
df = pd.read_csv("Real Estate Data Set.csv")
print_csv_file(df, heading)
```

```python
x = df.iloc[:, [1, 6]].values # Extracting Independent Variables; X2 and Y

print("\n"); heading = "The K-Means Clustering Method"
print('{:s}'.format('\u0332'.join(heading.center(100))))
k_means_clustering()

print("\n"); heading = "Hierarchical Clustering - Dendrogram"
print('{:s}'.format('\u0332'.join(heading.center(100))))
hierarchial_clustering_dendrogram()
```

PLOTS:

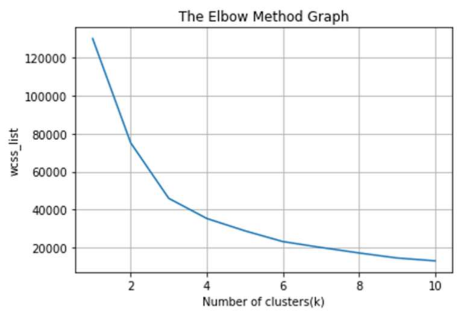

Figure 1. The Elbow Method Graph

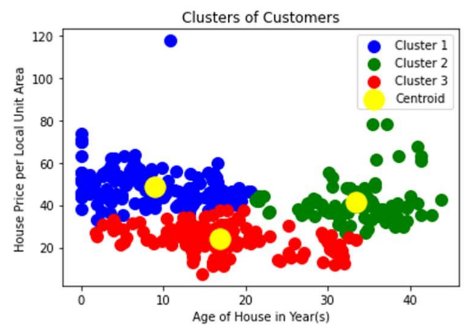

Figure 2. K-Means Clusters of Customers

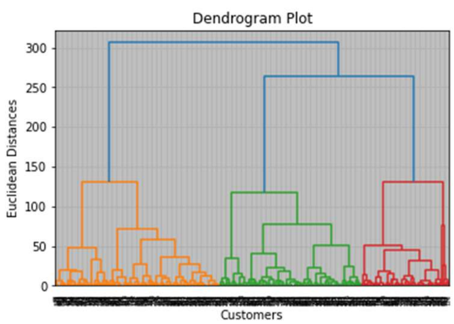

Figure 3. Dendrogram Plot

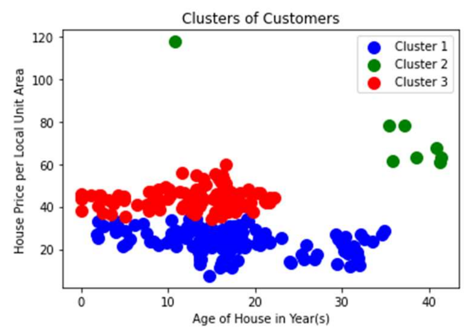

Figure 4. Hierarchical Clusters of Customers

RESULT:

Thus, presented the basic concepts and methods of cluster analysis. Learnt several basic clustering techniques and briefly discussed how to evaluate clustering methods. All the simulation results were verified successfully.

```
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

Restarting kernel...
```

In [1]: 'E:/Plan B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester IV/19CCE213 - Machine Learning and Artificial Intelligence/Lab/Experiment 5 - Clustering/Expt_5_Code.py' ='E:/Plan B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester IV/19CCE213 - Machine Learning and Artificial Intelligence/Lab/Experiment 5 - Clustering'

```
There are six regressor variables and one response variable (namely, y):
No - Serial Number
X2 - Age of House in Year(s)
X3 - Distance to Nearest MRT Station in Meter(s)
X4 - Number of Convenience Stores Within Walking Distance
X5 - Latitude Coordinates
X5 - Longitude Coordinates
Y - House Price per Local Unit Area


                                        Original Data Set
      No  X2 house age  ...  X6 longitude  Y house price of unit area
0      1          32.0  ...     121.54024                        37.9
1      2          19.5  ...     121.53951                        42.2
2      3          13.3  ...     121.54391                        47.3
3      4          13.3  ...     121.54391                        54.8
4      5           5.0  ...     121.54245                        43.1
..   ...           ...  ...           ...                         ...
409  410          13.7  ...     121.50381                        15.4
410  411           5.6  ...     121.54310                        50.0
411  412          18.8  ...     121.53986                        40.6
412  413           8.1  ...     121.54067                        52.5
413  414           6.5  ...     121.54310                        63.9

[414 rows x 7 columns]


                                  The K-Means Clustering Method

From the above plot, we can see the elbow point is at 3. Hence, the number of clusters
here will be 3.


                               Hierarchical Clustering - Dendrogram

Using this Dendrogram, we will now determine the optimal number of clusters for our model.
For this, we will find the maximum vertical distance that does not cut any horizontal bar.
Accordingly, the number of clusters will be 3.
```

In [2]: