2024

RAYYAN MASOOD.
Enrollment no: 01-134212-151
Class: BSCS 6-C
Date: May 12th ,2023

["ARTIFICIAL INTELLIGENCE"]

["LAB ASSIGNMENT NO:8"]

Lab Journal 8:

1. Apply k-means using following sample dataset without using any module.

Feature 1	Feature 2
1.1	1.1
1.5	2.1
3.1	4.1
5.1	7.1
3.5	5.1
4.5	5.1
3.5	4.5

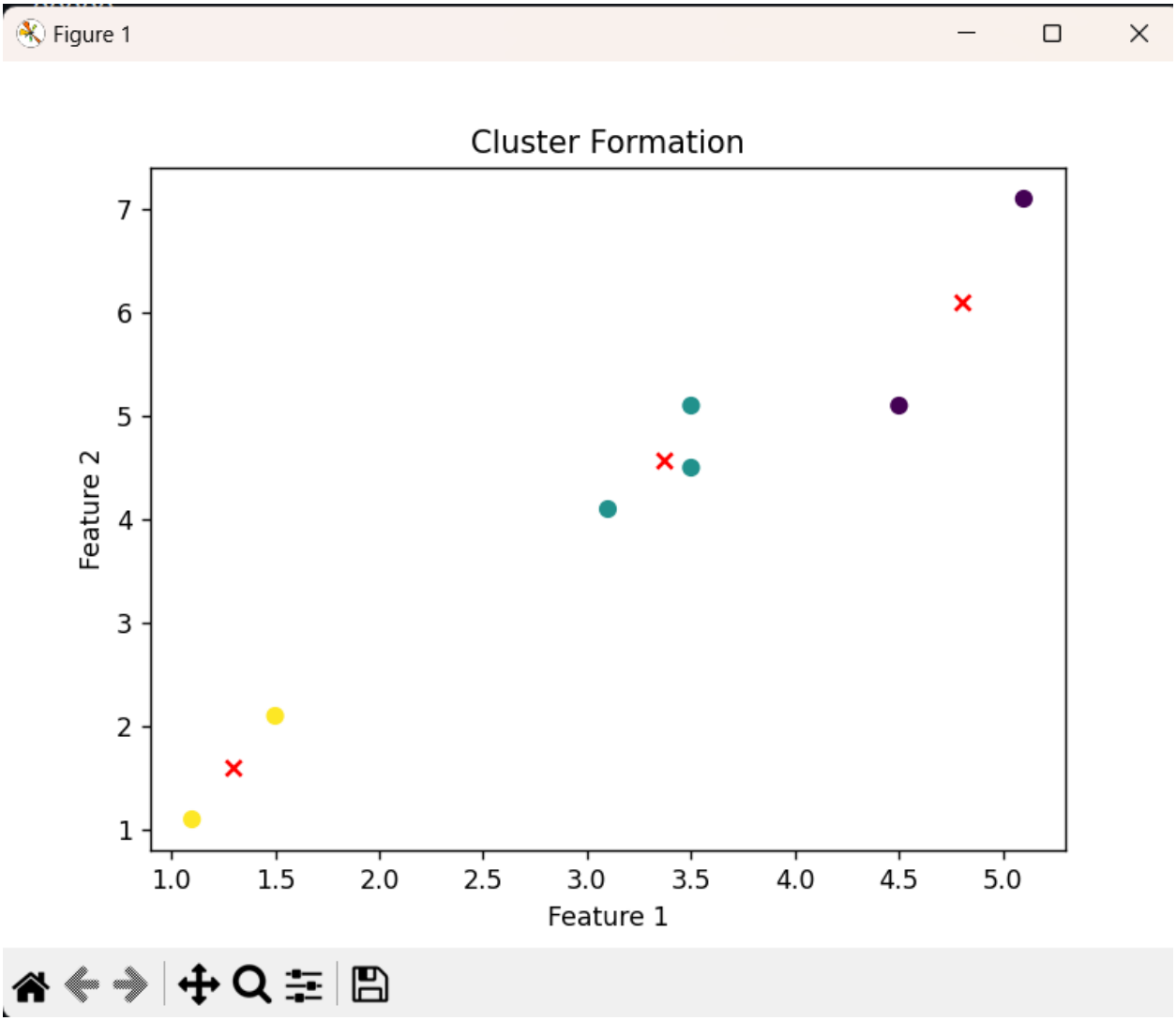
Code:

```
import pandas as pd
import numpy as np
from sklearn.cluster import k_means
import matplotlib.pyplot as plt

def k_means(X, k, max_iterations=100):

    centroids = X[np.random.choice(range(X.shape[0]), size=k,
replace=False)]

    for _ in range(max_iterations):

        labels = np.argmin(np.linalg.norm(X[:, np.newaxis] -
centroids, axis=-1), axis=-1)


        new_centroids = np.array([X[labels == i].mean(axis=0) for i in
range(k)])


        if np.all(centroids == new_centroids):
            break

        centroids = new_centroids

    return labels, centroids
```

```
data = pd.read_csv("feature.csv")


X = data.values


labels, centroids = k_means(X, k=3)
plt.scatter(X[:, 0], X[:, 1], c=labels)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='x')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Cluster Formation')
plt.show()

print("Labels:", labels)
print("Centroids:", centroids)
```

Output:

```
Labels: [1 1 2 0 2 2 2]
Centroids: [[5.1  7.1 ]
 [1.3  1.6 ]
 [3.65 4.7 ]]
```



2. Apply **K-means and Agglomerative clustering** algorithms using modules on the data setgiven in .csv file, which is uploaded on miscellaneous section of LMS. (Use Annual Income (k$) and Spending Score (1-100) columns from the given dataset and then decide number of clusters as well)

Code:

```
import pandas as pd
import numpy as np
from sklearn.cluster import k_means
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram,linkage
import matplotlib.pyplot as plt
```

```python
data = pd.read_csv("Mall_Customers.csv")


X = data[['Annual Income (k$)', 'Spending Score (1-100)']]


kmeans = k_means(X, n_clusters=3)
kmeans_labels = kmeans[1]


agg_clustering = AgglomerativeClustering(n_clusters=3)
agg_labels = agg_clustering.fit_predict(X)


print("K-means labels:", kmeans_labels)
print("Agglomerative labels:", agg_labels)



linkage_matrix = linkage(X, method='ward')

plt.scatter(X['Annual Income (k$)'], X['Spending Score (1-100)'],
c=agg_labels, cmap='rainbow')

plt.show()

plt.figure(figsize=(10, 6))
dendrogram(linkage_matrix)
plt.title('Dendrogram')
plt.xlabel('Samples')
plt.ylabel('Distance')
plt.show()
```

Output:

```
K-means labels: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]
Agglomerative labels: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 0 2 1 2 1 2 0 2 1 2 1 2 1 2 1 2 0 2 1 2 0 2
 1 2 1 2 1 2 1 2 1 2 1 2 0 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2]
```

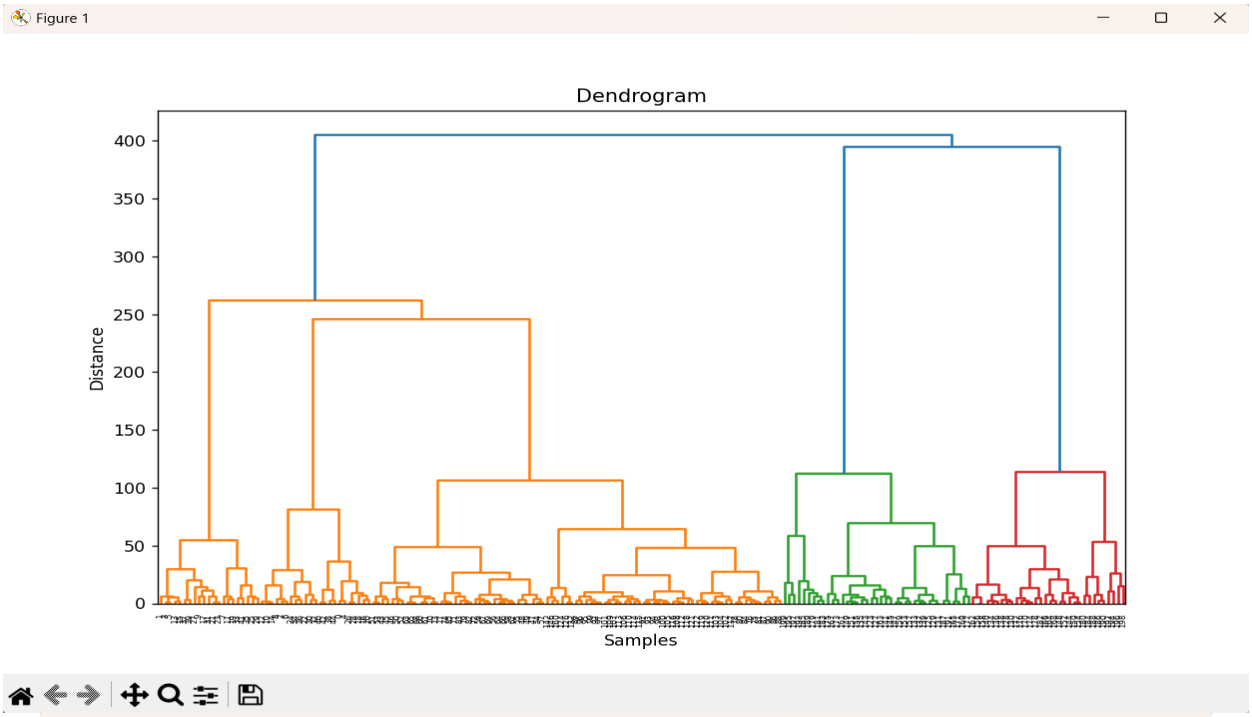


Figure 1
Dendrogram
Distance
Samples
400
350
300
250
200
150
100
50
0

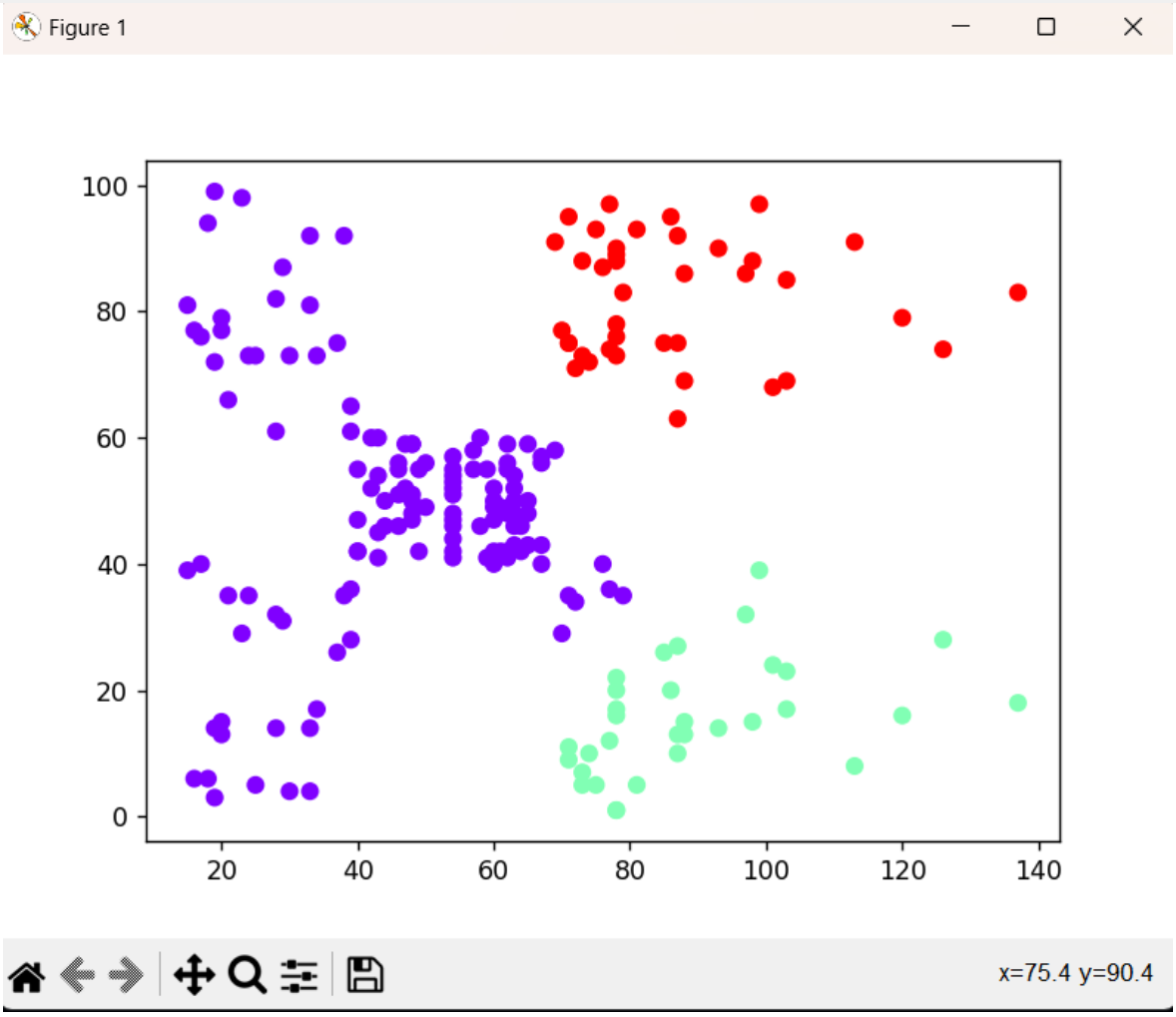


Figure 1
100
80
60
40
20
0
20
40
60
80
100
120
140
x=75.4 y=90.4