In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```python
df=pd.read_csv("Mall_Customers.csv")
```

In [3]:
```python
df.head()
```

Out[3]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [4]:
```python
df.shape
```

Out[4]:
```
(200, 5)
```

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [6]:
```python
df.describe()
```

Out[6]:

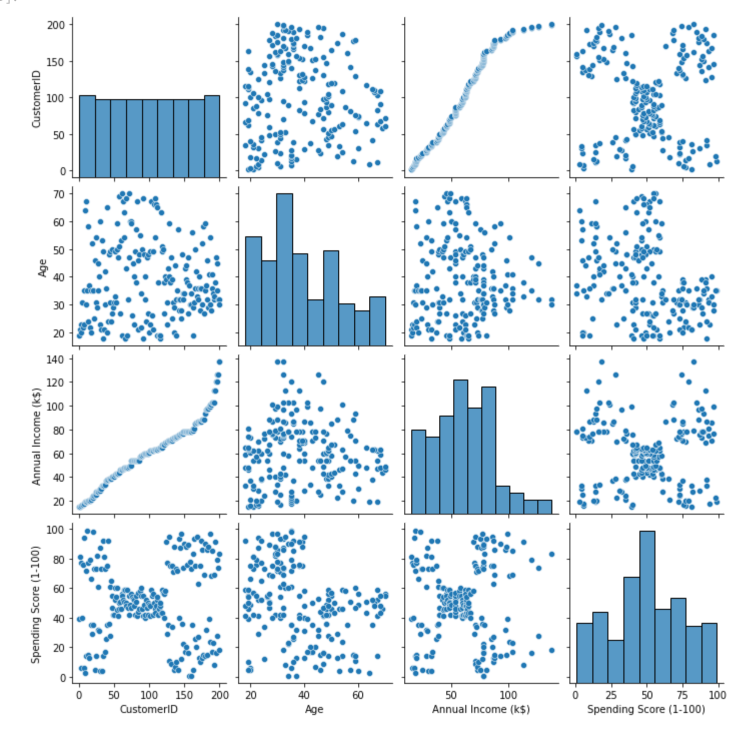
	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

In [10]:
```python
df.isna().sum()
```

Out[10]:
```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

In [9]:
```python
sns.pairplot(df)
```

Out[9]:
```
<seaborn.axisgrid.PairGrid at 0x1ed83015cd0>
```



In [12]:
```python
x=df.iloc[:,[3,4]]
x
```

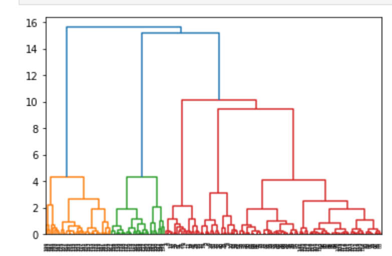
Out[12]:

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

In [13]:
```python
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x=sc.fit_transform(x)
```

In [15]:
```python
from scipy.cluster import hierarchy as hi
lk=hi.linkage(x,method="ward")
ddg=hi.dendrogram(lk)
```



In [16]:
```python
from sklearn.cluster import AgglomerativeClustering
hc=AgglomerativeClustering(n_clusters=5)
ylabel=hc.fit_predict(x)
```

In [17]:
```python
df["hcy"]=ylabel
```

In [18]:
```python
df["hcy"].value_counts()
```

Out[18]:
```
2    85
1    39
0    32
4    23
3    21
Name: hcy, dtype: int64
```

In [19]:
```python
df[df.hcy==0].describe()
```

Out[19]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)	hcy
count	32.000000	32.000000	32.000000	32.000000	32.0
mean	166.250000	41.000000	89.406250	15.593750	0.0
std	21.005376	11.036596	16.612975	8.936548	0.0
min	129.000000	19.000000	71.000000	1.000000	0.0
25%	150.500000	34.000000	78.000000	9.750000	0.0
50%	168.000000	41.500000	86.500000	15.000000	0.0
75%	183.500000	47.000000	98.250000	20.500000	0.0
max	199.000000	59.000000	137.000000	39.000000	0.0

In [20]:
```python
df[df.hcy==3].describe()
```

Out[20]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)	hcy
count	21.000000	21.000000	21.000000	21.000000	21.0
mean	22.000000	25.333333	25.095238	80.047619	3.0
std	12.409674	5.378971	7.133756	10.249274	0.0
min	2.000000	18.000000	15.000000	61.000000	3.0
25%	12.000000	21.000000	19.000000	73.000000	3.0
50%	22.000000	23.000000	24.000000	77.000000	3.0
75%	32.000000	30.000000	30.000000	87.000000	3.0
max	42.000000	35.000000	38.000000	99.000000	3.0

In [23]:
```python
df.groupby("hcy")[["Annual Income (k$)","Spending Score (1-100)"]].mean()
```

Out[23]:

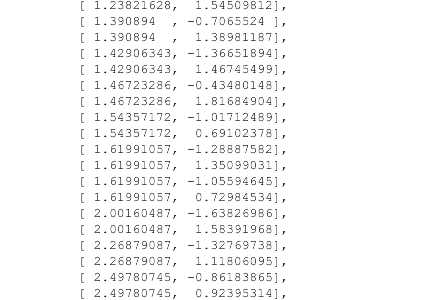
hcy	Annual Income (k$)	Spending Score (1-100)
0	89.406250	15.593750
1	86.538462	82.128205
2	55.811765	49.129412
3	25.095238	80.047619
4	26.304348	20.913043

In [24]:
```python
x
```

Out[24]:
```
array([[-1.73899919, -0.43480148],
       [-1.73899919,  1.19570407],
       [-1.70082976, -1.71591298],
       [-1.70082976,  1.04041783],
       [-1.66266033, -0.39597992],
       [-1.66266033,  1.00159627],
       [-1.62449091, -1.71591298],
       [-1.62449091,  1.70038436],
       [-1.58632148, -1.83237767],
       [-1.58632148,  0.84631002],
       [-1.58632148, -1.4053405 ],
       [-1.58632148,  1.89449216],
       [-1.54815205, -1.36651894],
       [-1.54815205,  1.04041783],
       [-1.54815205, -1.44416206],
       [-1.54815205,  1.11806095],
       [-1.50998262, -0.59008772],
       [-1.50998262,  0.61338066],
       [-1.43364376, -0.82301709],
       [-1.43364376,  1.8556706 ],
       [-1.39547433, -0.59008772],
       [-1.39547433,  0.88513158],
       [-1.3573049 , -1.75473454],
       [-1.3573049 ,  0.88513158],
       [-1.24279661, -1.4053405 ],
       [-1.24279661,  1.23452563],
       [-1.24279661, -0.7065524 ],
       [-1.24279661,  0.41927286],
       [-1.20462718, -0.74537397],
       [-1.20462718,  1.42863343],
       [-1.16645776, -1.7935561 ],
       [-1.16645776,  0.88513158],
       [-1.05194947, -1.7935561 ],
       [-1.05194947,  1.62274124],
       [-1.05194947, -1.4053405 ],
       [-1.05194947,  1.19570407],
       [-1.01378004, -1.28887582],
       [-1.01378004,  0.88513158],
       [-0.89927175, -0.93948177],
       [-0.89927175,  0.96277471],
       [-0.86110232, -0.59008772],
       [-0.86110232,  1.62274124],
       [-0.82293289, -0.55126616],
       [-0.82293289,  0.41927286],
       [-0.82293289, -0.86183865],
       [-0.82293289,  0.5745591 ],
       [-0.78476346,  0.18634349],
       [-0.78476346, -0.12422899],
       [-0.78476346, -0.3183368 ],
       [-0.78476346, -0.3183368 ],
       [-0.70842461,  0.06987881],
       [-0.70842461,  0.38045129],
       [-0.67025518,  0.14752193],
       [-0.67025518,  0.38045129],
       [-0.67025518, -0.20187212],
       [-0.67025518, -0.35715836],
       [-0.63208575, -0.00776431],
       [-0.63208575, -0.16305055],
       [-0.55574689,  0.03105725],
       [-0.55574689, -0.16305055],
       [-0.55574689,  0.22516505],
       [-0.55574689,  0.18634349],
       [-0.51757746,  0.06987881],
       [-0.51757746,  0.34162973],
       [-0.47940803,  0.03105725],
       [-0.47940803,  0.34162973],
       [-0.47940803, -0.00776431],
       [-0.47940803, -0.08540743],
       [-0.47940803,  0.34162973],
       [-0.47940803, -0.12422899],
       [-0.4412386 ,  0.18634349],
       [-0.4412386 , -0.3183368 ],
       [-0.40306917, -0.04658587],
       [-0.40306917,  0.22516505],
       [-0.25039146, -0.12422899],
       [-0.25039146,  0.14752193],
       [-0.25039146,  0.10870037],
       [-0.25039146, -0.08540743],
       [-0.25039146,  0.06987881],
       [-0.25039146, -0.3183368 ],
       [-0.25039146,  0.03105725],
       [-0.25039146,  0.18634349],
       [-0.25039146, -0.35715836],
       [-0.25039146, -0.24069368],
       [-0.25039146,  0.26398661],
       [-0.25039146, -0.16305055],
       [-0.13588317,  0.30280817],
       [-0.13588317,  0.18634349],
       [-0.09771374,  0.38045129],
       [-0.09771374, -0.16305055],
       [-0.05954431,  0.18634349],
       [-0.05954431, -0.35715836],
       [-0.02137488, -0.04658587],
       [-0.02137488, -0.39597992],
       [-0.02137488, -0.3183368 ],
       [-0.02137488,  0.06987881],
       [-0.02137488, -0.12422899],
       [-0.02137488, -0.00776431],
       [ 0.01679455, -0.3183368 ],
       [ 0.01679455, -0.04658587],
       [ 0.05496398, -0.35715836],
       [ 0.05496398, -0.08540743],
       [ 0.05496398,  0.34162973],
       [ 0.05496398,  0.18634349],
       [ 0.05496398,  0.22516505],
       [ 0.05496398, -0.3183368 ],
       [ 0.09313341, -0.00776431],
       [ 0.09313341, -0.16305055],
       [ 0.09313341, -0.27951524],
       [ 0.09313341, -0.08540743],
       [ 0.09313341,  0.06987881],
       [ 0.09313341,  0.14752193],
       [ 0.13130284, -0.3183368 ],
       [ 0.13130284, -0.16305055],
       [ 0.16947227, -0.08540743],
       [ 0.16947227, -0.00776431],
       [ 0.16947227, -0.27951524],
       [ 0.16947227,  0.34162973],
       [ 0.24581112, -0.27951524],
       [ 0.24581112,  0.26398661],
       [ 0.24581112,  0.22516505],
       [ 0.24581112, -0.39597992],
       [ 0.32214998,  0.30280817],
       [ 0.32214998,  1.58391968],
       [ 0.36031941, -0.82301709],
       [ 0.36031941,  1.04041783],
       [ 0.39848884, -0.59008772],
       [ 0.39848884,  1.73920592],
       [ 0.39848884, -1.52180518],
       [ 0.39848884,  0.96277471],
       [ 0.39848884, -1.5994483 ],
       [ 0.39848884,  0.96277471],
       [ 0.43665827, -0.62890928],
       [ 0.43665827,  0.80748846],
       [ 0.4748277 , -1.75473454],
       [ 0.4748277 ,  1.46745499],
       [ 0.4748277 , -1.67709142],
       [ 0.4748277 ,  0.88513158],
       [ 0.51299713, -1.56062674],
       [ 0.51299713,  0.84631002],
       [ 0.55116656, -1.75473454],
       [ 0.55116656,  1.6615628 ],
       [ 0.58933599, -0.39597992],
       [ 0.58933599,  1.42863343],
       [ 0.62750542, -1.48298362],
       [ 0.62750542,  1.81684904],
       [ 0.62750542, -0.55126616],
       [ 0.62750542,  0.92395314],
       [ 0.66567484, -1.09476801],
       [ 0.66567484,  1.54509812],
       [ 0.66567484, -1.28887582],
       [ 0.66567484,  1.46745499],
       [ 0.66567484, -1.17241113],
       [ 0.66567484,  1.00159627],
       [ 0.66567484, -1.32769738],
       [ 0.66567484,  1.50627656],
       [ 0.66567484, -1.91002079],
       [ 0.66567484,  1.07923939],
       [ 0.66567484, -1.91002079],
       [ 0.66567484,  0.88513158],
       [ 0.70384427, -0.59008772],
       [ 0.70384427,  1.27334719],
       [ 0.78018313, -1.75473454],
       [ 0.78018313,  1.6615628 ],
       [ 0.93286085, -0.93948177],
       [ 0.93286085,  0.96277471],
       [ 0.97103028, -1.17241113],
       [ 0.97103028,  1.73920592],
       [ 1.00919971, -0.90066021],
       [ 1.00919971,  0.49691598],
       [ 1.00919971, -1.44416206],
       [ 1.00919971,  0.96277471],
       [ 1.00919971, -1.56062674],
       [ 1.00919971,  1.62274124],
       [ 1.04736914, -1.44416206],
       [ 1.04736914,  1.38981187],
       [ 1.04736914, -1.36651894],
       [ 1.04736914,  0.72984534],
       [ 1.23821628, -1.4053405 ],
       [ 1.23821628,  1.54509812],
       [ 1.390894  , -0.7065524 ],
       [ 1.390894  ,  1.38981187],
       [ 1.42906343, -1.36651894],
       [ 1.42906343,  1.46745499],
       [ 1.46723286, -0.43480148],
       [ 1.46723286,  1.81684904],
       [ 1.54357172, -1.01712489],
       [ 1.54357172,  0.69102378],
       [ 1.61991057, -1.28887582],
       [ 1.61991057,  1.35099031],
       [ 1.61991057, -1.05594645],
       [ 1.61991057,  0.72984534],
       [ 2.00160487, -1.63826986],
       [ 2.00160487,  1.58391968],
       [ 2.26879087, -1.32769738],
       [ 2.26879087,  1.11806095],
       [ 2.49780745, -0.86183865],
       [ 2.49780745,  0.92395314],
       [ 2.91767117, -1.25005425],
       [ 2.91767117,  1.27334719]])
```

In [34]:
```python
plt.scatter(x[ylabel==0,0],x[ylabel==0,1],s=100,c="red",label="Cluster1")
plt.scatter(x[ylabel==1,0],x[ylabel==1,1],s=100,c="green",label="Cluster2")
plt.scatter(x[ylabel==2,0],x[ylabel==2,1],s=100,c="blue",label="Cluster3")
plt.scatter(x[ylabel==3,0],x[ylabel==3,1],s=100,c="yellow",label="Cluster4")
plt.scatter(x[ylabel==4,0],x[ylabel==4,1],s=100,c="cyan",label="Cluster5")

plt.xlabel("Annual Income")
plt.xlabel("Spending Score")
plt.xlabel("Clustering of Customer")
plt.legend()
plt.show()
```



In [28]:
```python
y=df.iloc[:,-1]
y
```

Out[28]:
```
0      4
1      3
2      4
3      3
4      4
      ..
195    1
196    0
197    1
198    0
199    1
Name: hcy, Length: 200, dtype: int64
```

In [29]:
```python
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [30]:
```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
```

In [31]:
```python
def indmodel(model):
    model.fit(xtrain,ytrain)
    ypred=model.predict(xtest)

    train=model.score(xtrain,ytrain)
    test=model.score(xtest,ytest)

    print(f' Training Accuracy : {train} \n Testing Accuracy : {test} \n')
    print(classification_report(ytest,ypred))

    return model
```

In [32]:
```python
cladt=indmodel(DecisionTreeClassifier(random_state=2))
```

```
 Training Accuracy : 1.0
 Testing Accuracy : 0.95

              precision    recall  f1-score   support

           0       0.90      0.90      0.90        10
           1       0.92      1.00      0.96        11
           2       1.00      0.95      0.98        22
           3       1.00      0.88      0.93         8
           4       0.90      1.00      0.95         9

    accuracy                           0.95        60
   macro avg       0.94      0.95      0.94        60
weighted avg       0.95      0.95      0.95        60
```

In [ ]: