

ML LAB ASSIGNMENT

SUPRATIM NAG -- CSE-AIML/22/057 -- GROUP-B

Q-6:Write a python code to implement K-Means clustering algorithm.

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn import metrics
from collections import Counter
from matplotlib.colors import ListedColormap
```

```
In [3]: meddata = pd.read_csv(r"C:\Users\SUPRATIM NAG\OneDrive\Documents\ML\Personal_Datasets\Dataset.csv")
meddata.head(1)
```

```
Out[3]:
```

	Patient ID	Age	Blood Pressure	Cholesterol Levels	Heart Rate	BMI	Diagnosis	Treatment Plan	Recovery Status	Medication Type	Follow-up Requirement
0	101	65	130	250	72	28.0	Hypertension with high cholesterol.	Medication: Lisinopril (blood pressure), Stati...	Active Recovery	Lisinopril, Statins.	Quarterly.

```
In [20]: data=meddata[["Age", "Blood Pressure", "Cholesterol Levels", "Heart Rate"]]
data.head(1)
```

```
Out[20]:
```

	Age	Blood Pressure	Cholesterol Levels	Heart Rate
0	65	130	250	72

```
In [6]: S = MinMaxScaler()
data = S.fit_transform(data)
data
```

```

Out[6]: array([[0.8      , 0.53846154, 0.72222222, 0.24      ],
 [0.34     , 0.23076923, 0.16666667, 0.32      ],
 [0.66     , 0.69230769, 0.44444444, 0.4       ],
 [0.92     , 1.         , 0.55555556, 0.56      ],
 [0.2      , 0.38461538, 0.33333333, 0.28      ],
 [0.7      , 0.61538462, 0.61111111, 0.36      ],
 [0.4      , 0.15384615, 0.         , 0.3       ],
 [0.4      , 0.38461538, 0.44444444, 0.2       ],
 [0.2      , 0.69230769, 0.33333333, 0.5       ],
 [0.5      , 0.61538462, 0.16666667, 0.4       ],
 [0.3      , 0.46153846, 0.22222222, 0.3       ],
 [0.6      , 0.53846154, 0.27777778, 0.4       ],
 [0.7      , 0.61538462, 0.33333333, 0.3       ],
 [0.4      , 0.30769231, 0.55555556, 0.2       ],
 [0.5      , 0.53846154, 0.36111111, 0.4       ],
 [0.2      , 0.46153846, 0.5         , 0.5       ],
 [0.8      , 0.69230769, 0.44444444, 0.3       ],
 [0.9      , 0.53846154, 0.27777778, 0.4       ],
 [0.2      , 0.38461538, 0.44444444, 0.5       ],
 [0.4      , 0.53846154, 0.72222222, 0.6       ],
 [0.5      , 0.69230769, 0.55555556, 0.7       ],
 [0.7      , 0.84615385, 0.88888889, 0.8       ],
 [0.8      , 1.         , 1.         , 1.        ],
 [0.54     , 0.53846154, 0.44444444, 0.3       ],
 [0.2      , 0.23076923, 0.16666667, 0.4       ],
 [0.86     , 0.69230769, 0.55555556, 0.5       ],
 [0.4      , 0.38461538, 0.27777778, 0.2       ],
 [0.06     , 0.07692308, 0.05555556, 0.3       ],
 [1.       , 0.84615385, 0.66666667, 0.6       ],
 [0.5      , 0.46153846, 0.33333333, 0.4       ],
 [0.1      , 0.15384615, 0.11111111, 0.3       ],
 [0.3      , 0.30769231, 0.22222222, 0.2       ],
 [0.7      , 0.76923077, 0.5         , 0.6       ],
 [0.6      , 0.53846154, 0.38888889, 0.3       ],
 [0.26     , 0.07692308, 0.         , 0.4       ],
 [0.8      , 0.92307692, 0.72222222, 0.6       ],
 [0.4      , 0.38461538, 0.27777778, 0.2       ],
 [0.       , 0.         , 0.05555556, 0.3       ],
 [0.5      , 0.61538462, 0.44444444, 0.5       ],
 [0.9      , 0.84615385, 0.66666667, 0.6       ],
 [0.3      , 0.23076923, 0.16666667, 0.3       ],
 [0.4      , 0.38461538, 0.55555556, 0.4       ],
 [0.7      , 0.69230769, 0.72222222, 0.3       ],
 [0.5      , 0.53846154, 0.33333333, 0.2       ],
 [0.2      , 0.23076923, 0.16666667, 0.1       ],
 [0.06     , 0.07692308, 0.         , 0.        ],
 [0.3      , 0.61538462, 0.5         , 0.5       ],
 [0.6      , 0.46153846, 0.44444444, 0.2       ],
 [0.9      , 0.84615385, 0.61111111, 0.3       ],
 [0.8      , 0.69230769, 0.66666667, 0.4       ],
 [0.5      , 0.53846154, 0.55555556, 0.5       ],
 [0.1      , 0.07692308, 0.11111111, 0.1       ],
 [0.4      , 0.46153846, 0.38888889, 0.2       ],
 [0.9      , 1.         , 0.72222222, 0.6       ],
 [0.66     , 0.38461538, 0.5         , 0.4       ],
 [0.18     , 0.23076923, 0.22222222, 0.3       ],
 [0.74     , 0.53846154, 0.61111111, 0.2       ],
 [0.4      , 0.30769231, 0.27777778, 0.5       ],
 [0.9      , 0.76923077, 0.69444444, 0.4       ],
 [0.06     , 0.38461538, 0.38888889, 0.3       ],
 [0.54     , 0.15384615, 0.36111111, 0.4       ],
 [0.28     , 0.23076923, 0.47222222, 0.2       ],
 [0.8      , 0.61538462, 0.72222222, 0.5       ],
 [0.34     , 0.38461538, 0.55555556, 0.3       ],
 [0.6      , 0.23076923, 0.44444444, 0.3       ],
 [1.       , 0.84615385, 0.77777778, 0.6       ],
 [0.46     , 0.07692308, 0.38888889, 0.2       ],
 [0.7      , 0.46153846, 0.61111111, 0.6       ],
 [0.14     , 0.30769231, 0.5         , 0.4       ],
 [0.4      , 0.38461538, 0.44444444, 0.4       ],
 [0.1      , 0.23076923, 0.16666667, 0.2       ],
 [0.7      , 0.53846154, 0.55555556, 0.5       ],
 [0.5      , 0.69230769, 0.5         , 0.6       ],
 [0.3      , 0.30769231, 0.27777778, 0.3       ],
 [0.2      , 0.46153846, 0.38888889, 0.4       ],
 [0.6      , 0.61538462, 0.61111111, 0.5       ],
 [0.8      , 0.76923077, 0.72222222, 0.6       ],
 [0.4      , 0.23076923, 0.5         , 0.3       ],
 [0.1      , 0.38461538, 0.33333333, 0.4       ],
 [0.34     , 0.53846154, 0.47222222, 0.44      ],
 [0.66     , 0.46153846, 0.52777778, 0.36      ],
 [0.86     , 0.69230769, 0.61111111, 0.6       ],
 [0.24     , 0.30769231, 0.33333333, 0.3       ],
 [0.56     , 0.61538462, 0.58333333, 0.5       ],
 [0.46     , 0.38461538, 0.41666667, 0.3       ],
 [0.76     , 0.76923077, 0.69444444, 0.6       ],
 [0.14     , 0.15384615, 0.16666667, 0.2       ],
 [0.2      , 0.38461538, 0.38888889, 0.5       ],
 [0.34     , 0.61538462, 0.61111111, 0.6       ],
 [0.66     , 0.76923077, 0.5         , 0.3       ],
 [0.06     , 0.23076923, 0.22222222, 0.2       ],
 [0.74     , 0.46153846, 0.72222222, 0.4       ],
 [0.6      , 0.30769231, 0.44444444, 0.3       ],
 [0.1      , 0.23076923, 0.27777778, 0.4       ],
 [0.5      , 0.53846154, 0.55555556, 0.4       ],
 [0.4      , 0.69230769, 0.77777778, 0.5       ],
 [0.9      , 0.46153846, 0.44444444, 0.2       ],
 [0.6      , 0.30769231, 0.33333333, 0.4       ],
 [0.1      , 0.38461538, 0.38888889, 0.3       ],
 [0.4      , 0.38461538, 0.66666667, 0.4       ]])

```

```
In [7]: data = pd.DataFrame(data, columns=['A','B','C','D'])
data
```

```
Out[7]:
```

	A	B	C	D
0	0.80	0.538462	0.722222	0.24
1	0.34	0.230769	0.166667	0.32
2	0.66	0.692308	0.444444	0.40
3	0.92	1.000000	0.555556	0.56
4	0.20	0.384615	0.333333	0.28
...
95	0.40	0.692308	0.777778	0.50
96	0.90	0.461538	0.444444	0.20
97	0.60	0.307692	0.333333	0.40
98	0.10	0.384615	0.388889	0.30
99	0.40	0.384615	0.666667	0.40

100 rows × 4 columns

```
In [8]: model = KMeans(n_clusters=2)
model.fit(data)
```

```
Out[8]:
```

KMeans

KMeans(n_clusters=2)

```
In [9]: y = model.fit_predict(data)
y
```

```
Out[9]: array([0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
        0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1,
        1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0,
        1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0,
        0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1])
```

```
In [10]: data['cluster'] = y
data
```

```
Out[10]:
```

	A	B	C	D	cluster
0	0.80	0.538462	0.722222	0.24	0
1	0.34	0.230769	0.166667	0.32	1
2	0.66	0.692308	0.444444	0.40	0
3	0.92	1.000000	0.555556	0.56	0
4	0.20	0.384615	0.333333	0.28	1
...
95	0.40	0.692308	0.777778	0.50	0
96	0.90	0.461538	0.444444	0.20	0
97	0.60	0.307692	0.333333	0.40	1
98	0.10	0.384615	0.388889	0.30	1
99	0.40	0.384615	0.666667	0.40	1

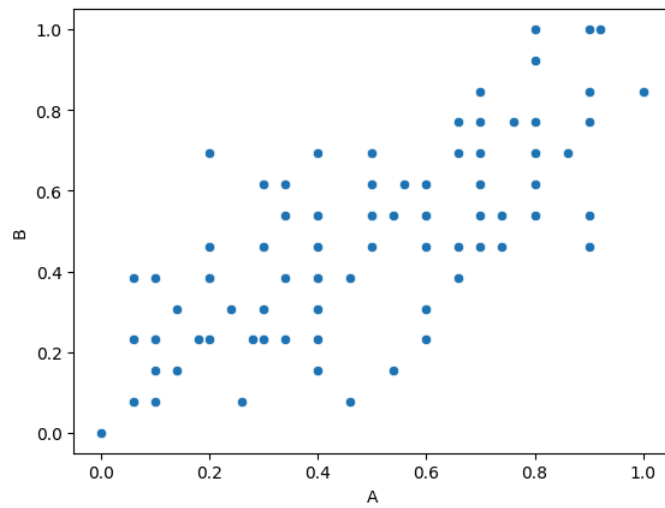
100 rows × 5 columns

```
In [11]: cent=model.cluster_centers_
cent
```

```
Out[11]: array([[0.71022222, 0.67008547, 0.5962963 , 0.47822222],
        [0.29927273, 0.32167832, 0.31868687, 0.31345455]])
```

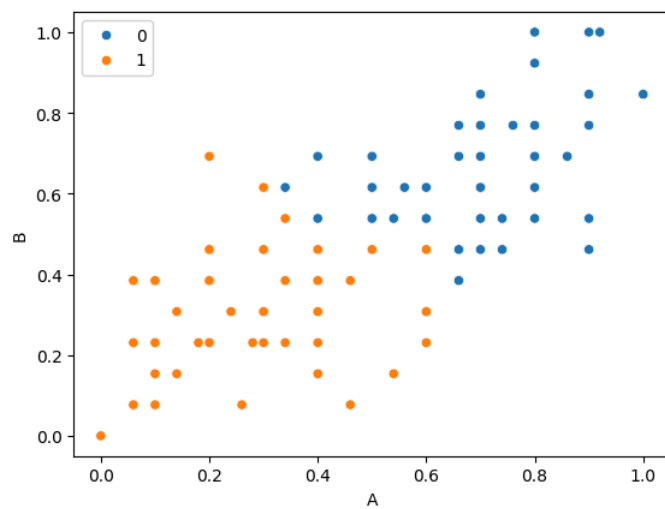
```
In [12]: Counter(model.labels_)
Counter({1 : 50, 0 : 100})
sns.scatterplot(data=data, x='A', y='B')
```

```
Out[12]: <Axes: xlabel='A', ylabel='B'>
```



```
In [13]: sns.scatterplot(data=data, x='A', y='B', hue=model.labels_)
```

```
Out[13]: <Axes: xlabel='A', ylabel='B'>
```



```
In [16]: K_rng=range(1,11)
Sse=[]
for K in K_rng:
    km = KMeans(n_clusters=K)
    km.fit(data[['A', 'B']])
    Sse.append(km.inertia_)

Sse
```

```
Out[16]: [12.102806414201183,
4.608999057910252,
2.968851249329662,
2.40675571146802,
2.02628787114846,
1.5418946136970129,
1.312926303579462,
1.1306159632764903,
0.9454375923878888,
0.967232773081623]
```