

# IMPLEMENTATION OF CLUSTERING TECHNIQUES

## K-MEANS

### AIM:-

To implement a K-means clustering technique using python language

### EXPLANATION

- Import KMeans from sklearn cluster
- Assign x and y
- call the function KMeans().
- perform scatter operation and display output

### SOURCE CODE:-

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
X, y = make_blobs (n_samples=300, centers=4, cluster_std=0.0, random_state=0)
```

```
plt.figure(figsize=(6,6))
plt.scatter(X[:,0], X[:,1])
plt.title('data points')
plt.xlabel('feature1')
plt.ylabel('feature2')
plt.show()
```

```
WCSS = []
```

```
for i in range(1, 11):
```

```
    KMeans = KMeans (n_clusters=i, init='K-means++',
                      max_iter=10, random_state=0)
```

```
    KMeans.fit(X)
```

```
    WCSS.append(KMeans.inertia_)
```

```
plt.figure(figsize=(8,6))
```

```
plt.plot(range(1, 11), WCSS, marker='o')
```

```
plt.title('Elbow Method')
```

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show

kmeans = KMeans(n\_clusters=4, init='k-means++',  
max\_iter=300, n\_init=10, random=0)

pred\_y = kmeans.fit\_predict(X)

plt.figure(figsize=(8,6))

plt.scatter(X[:, 0], X[:, 1], c=pred\_y, cmap='viridis',  
marker='o')

plt.title('KMeans clustering')

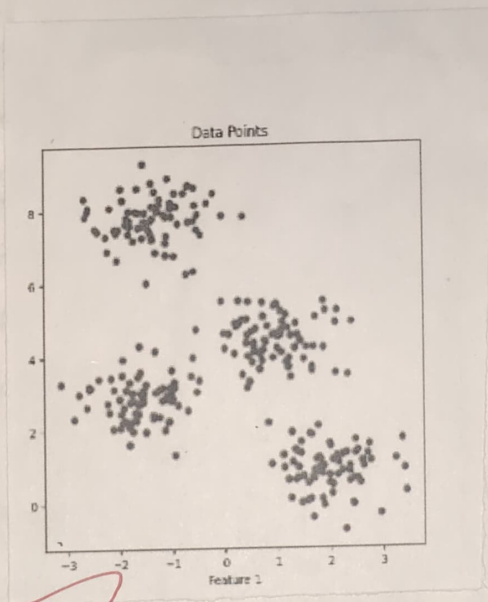
plt.xlabel('Feature 1')

plt.ylabel('Feature 2')

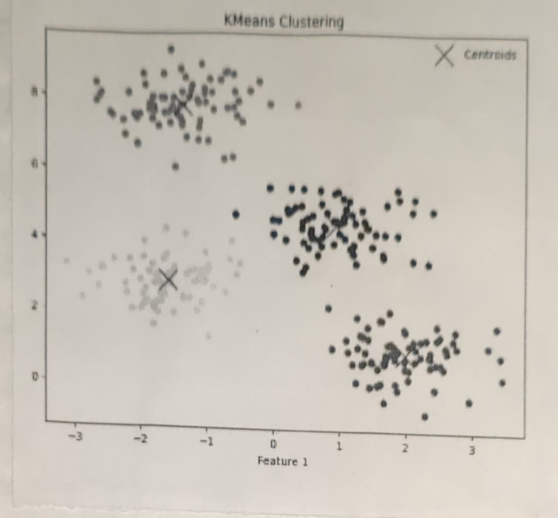
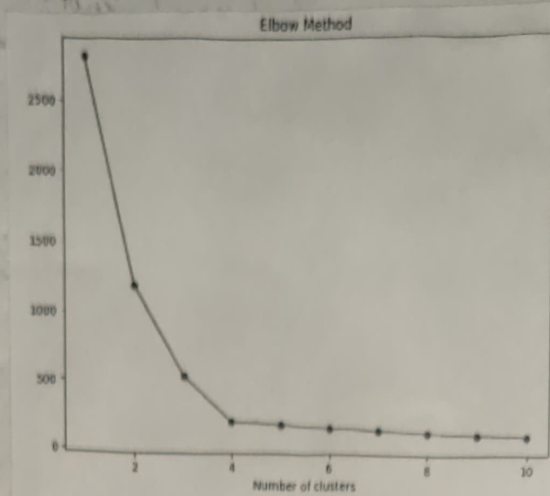
plt.legend()

plt.show()

OUTPUT:-



Result: This program is successful.  
This program is successful.  
This program is successful.



Result:

Thus implemented K-means clustering technique using python language is implemented and executed successfully.