PROGRAM

```
import numpy as np.
import pandas as pd
from matplot import pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import Kmeans
X, y = make_blobs (n_samples = 300, centres=4,
                   cluster_std = 0.60, random_state=0)
plt.scatter (X[:,0],X[:,1])
plt.title ('UnrelatedData')
plt.xlabel('feature1')
plt.ylabel('feature2')
plt.show()
WCSS = []
for i in range (1,11):
    kmeans = KMeans (n_clusters = i, init = 'k-means++',
    Kmeans.fit(X)                    max_iter=300, n_init=10,
    WCSS.append(Kmeans.inertia_)     random_state=0)
plt.plot(range(1,11) WCSS)
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,
pred_y = Kmeans.fit_predict(X)          n_init = 10, random_state=0)

plt.scatter (X[:,0], X[:,1], c = pred_y, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter (centers[:,0], centers[:,1], c='red', s=200,
plt.title ('KMeans')                    alpha = 0.5, marker='x')
plt.xlabel ('Feature1')
plt.ylabel ('Feature2')
plt.show().
```

classmate
Date
Page

EXP NO: 8

AIM: Implement clustering Techniques using K-MEANS in python

PROCEDURE

1. Generate Data: Create synthetic data
2. Visualise Scatter: scatter plot data points

3. Determine optimal clusters (Elbow Method):
   - For each number of cluster perform Kmeans
   - Record WCSS for each number of clusters
   - Plot WCSS against number of clusters.

4. Apply KMeans clustering: Run KMeans using the identified optimal number of clusters

5. plot final clusters: Scatter plot the data points

OUTPUT

Thus the KMeans clustering is performed in python.