Experiment-5

Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same dataset for clustering using k-Means algorithm. Compare the results of these twoalgorithms and comment on the quality of clustering. You can add Java/Python MLlibrary classes/API in the program.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
from sklearn.cluster import KMeans
```

In [3]:

```
X=np.random.rand(200,2)
```

In [4]:

```
model=KMeans(n_clusters=8)
```

In [5]:

```
model.fit(X)
```

```
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[5]:

KMeans

KMeans()

In [6]:

```
labels=model.labels_
```

In [7]:

```
labels
```

Out[7]:

```
array([3, 2, 7, 2, 1, 6, 7, 7, 5, 3, 2, 5, 6, 7, 6, 7, 1, 7, 4, 2, 7, 2,
       6, 6, 1, 5, 3, 3, 4, 3, 5, 6, 2, 4, 6, 5, 4, 2, 4, 1, 7, 4, 1, 2,
       6, 4, 4, 2, 5, 2, 5, 1, 6, 3, 1, 0, 1, 2, 0, 1, 4, 2, 6, 4, 1, 5,
       2, 6, 1, 0, 7, 5, 6, 4, 2, 4, 2, 2, 5, 1, 7, 3, 5, 7, 1, 1, 5, 2,
       0, 0, 7, 1, 1, 3, 0, 5, 6, 2, 4, 6, 1, 3, 0, 6, 2, 7, 3, 5, 3, 4,
       3, 1, 4, 7, 7, 5, 7, 5, 6, 7, 1, 2, 0, 7, 7, 6, 3, 5, 7, 6, 0, 6,
       5, 3, 2, 4, 5, 7, 4, 1, 2, 7, 1, 3, 1, 1, 0, 5, 0, 2, 2, 6, 0, 5,
       0, 7, 3, 7, 2, 6, 4, 3, 0, 3, 2, 2, 4, 4, 5, 6, 6, 6, 5, 5, 3, 6,
       6, 5, 2, 4, 4, 3, 0, 3, 2, 0, 2, 2, 6, 5, 3, 2, 2, 4, 7, 0, 0, 0,
       3, 6])
```

In [8]:

```
centroids=model.cluster_centers_
```

In [9]:

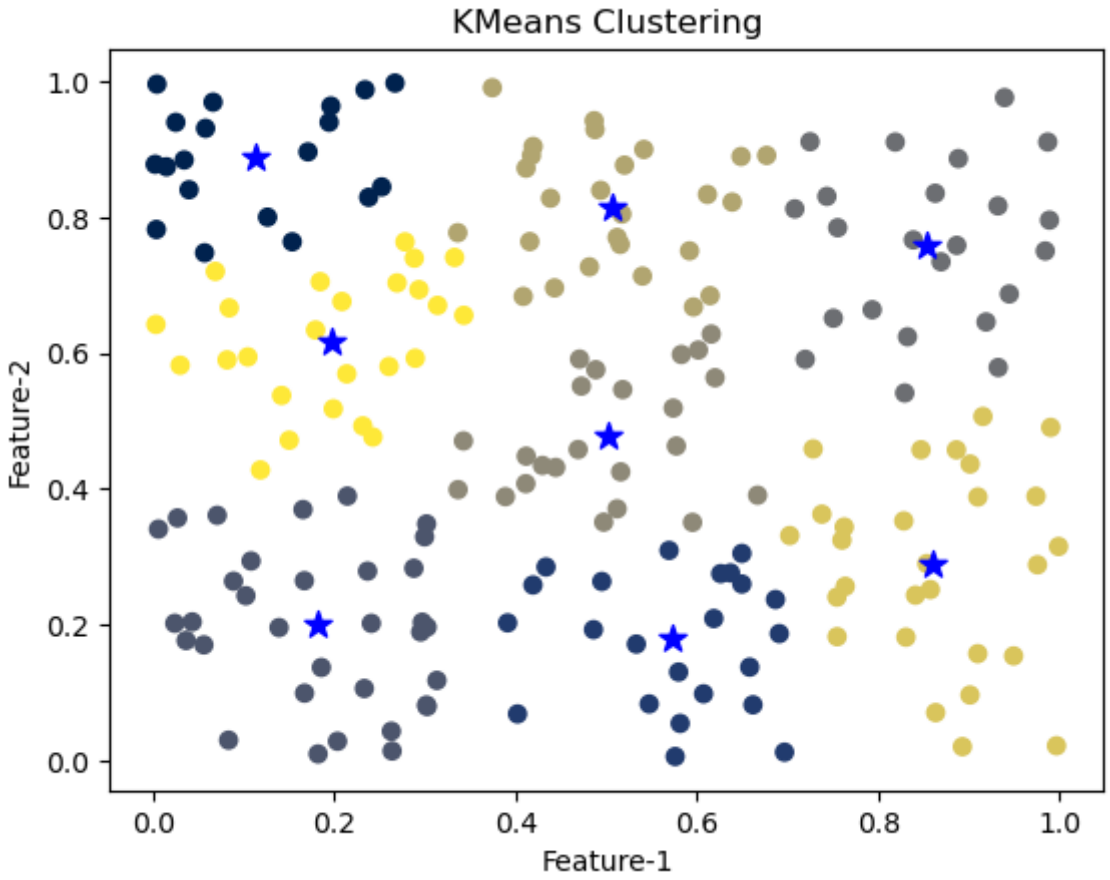
```
centroids
```

Out[9]:

```
array([[0.11225639, 0.8883121 ],
       [0.57374758, 0.17869007],
       [0.18186481, 0.20050173],
       [0.85440254, 0.75980558],
       [0.50174992, 0.47716062],
       [0.50525746, 0.81630327],
       [0.86038071, 0.28847714],
       [0.19617526, 0.61815977]])
```

In [10]:

```
plt.scatter(X[:,0],X[:,1],c=labels,cmap='cividis')
plt.scatter(centroids[:,0],centroids[:,1],c='b',s=100,marker='*')
plt.xlabel('Feature-1')
plt.ylabel('Feature-2')
plt.title('KMeans Clustering')
plt.show()
```



In [11]:

```
wcss=[]
for k in range(1,15):
    model=KMeans(n_clusters=k)
    model.fit(X)
    wcss.append(model.inertia_)
```

```
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
```

```
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\khana\miniconda3\envs\ML_Experiments\lib\site-packages\sklearn\cl
uster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. Yo
u can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [12]:

```
plt.plot(range(1,15),wcss)
plt.xlabel('Number of clusters (K)')
plt.ylabel("With in Cluster Sum of Squares(wcss)")
plt.title("Elbow method for optimal K")
plt.show()
```