

Assignment 4

K-Means Clustering

41323_Gayatri

In [2]:

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

In [3]:

```
#create dataset using DataFrame
df=pd.DataFrame({'X':[0.1,0.15,0.08,0.16,0.2,0.25,0.24,0.3],
                 'y':[0.6,0.71,0.9,0.85,0.3,0.5,0.1,0.2]})
```

In [4]:

```
f1 = df['X'].values
f2 = df['y'].values
X = np.array(list(zip(f1, f2)))
print(X)
```

```
[[0.1  0.6 ]
 [0.15 0.71]
 [0.08 0.9 ]
 [0.16 0.85]
 [0.2  0.3 ]
 [0.25 0.5 ]
 [0.24 0.1 ]
 [0.3  0.2 ]]
```

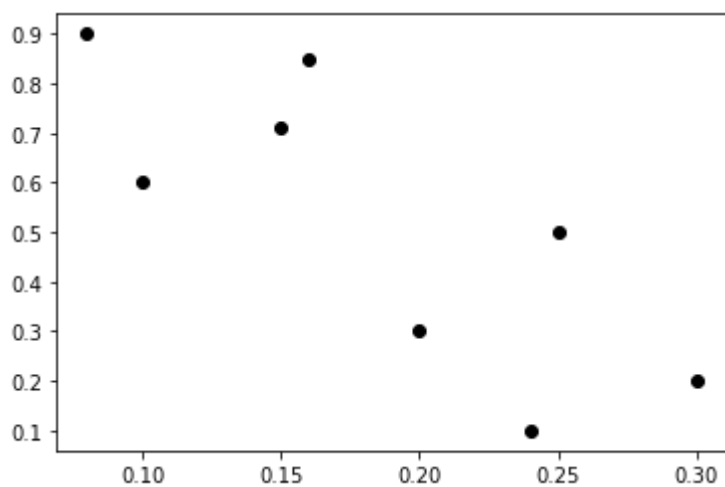
In [5]:

```
#centroid points
C_x=np.array([0.1,0.3])
C_y=np.array([0.6,0.2])
centroids=C_x,C_y
```

In [6]:

```
#plot the given points
```

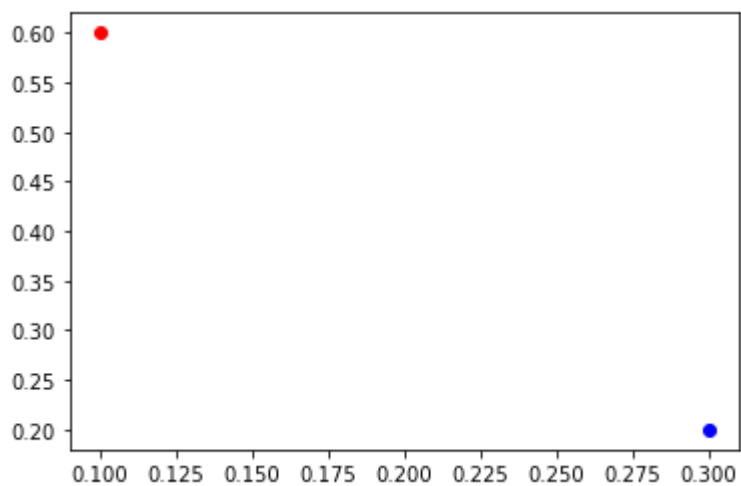
```
plt.scatter(f1, f2, color='k')  
plt.show()
```



In [7]:

```
#for i in centroids():
```

```
colmap = {1: 'r', 2: 'b'}  
plt.scatter(C_x[0],C_y[0], color=colmap[1])  
plt.scatter(C_x[1],C_y[1], color=colmap[2])  
plt.show()
```



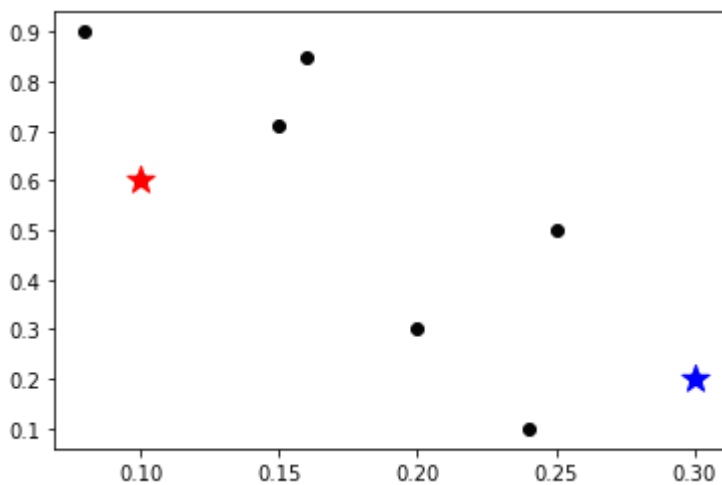
In [8]:

```
C = np.array(list((C_x, C_y)), dtype=np.float32)
print (C)
```

```
[[0.1 0.3]
 [0.6 0.2]]
```

In [9]:

```
#plot given elements with centroid elements
plt.scatter(f1, f2, c='#050505')
plt.scatter(C_x[0], C_y[0], marker='*', s=200, c='r')
plt.scatter(C_x[1], C_y[1], marker='*', s=200, c='b')
plt.show()
```



In [10]:

```
#import KMeans class and create object of it
from sklearn.cluster import KMeans
model=KMeans(n_clusters=2,random_state=0)
model.fit(X)
labels=model.labels_
print(labels)
```

```
[1 1 1 1 0 0 0 0]
```

In [11]:

```
#using labels find population around centroid
count=0
for i in range(len(labels)):
    if (labels[i]==1):
        count=count+1

print('No of population around cluster 2:',count-1)
```

```
No of population around cluster 2: 3
```

In [12]:

```
#Find new centroids  
new_centroids = model.cluster_centers_
```

In [13]:

```
print('Previous value of m1 and m2 is:')  
print('M1==',centroids[0])  
print('M1==',centroids[1])  
  
print('updated value of m1 and m2 is:')  
print('M1==',new_centroids[0])  
print('M1==',new_centroids[1])
```

Previous value of m1 and m2 is:

M1== [0.1 0.3]

M1== [0.6 0.2]

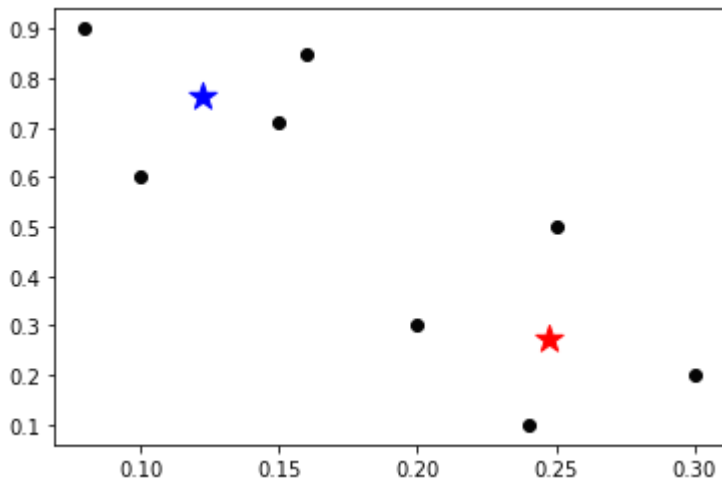
updated value of m1 and m2 is:

M1== [0.2475 0.275]

M1== [0.1225 0.765]

In [14]:

```
#plot given elements with centroid elements  
plt.scatter(f1, f2, c='#050505')  
plt.scatter(new_centroids[0][0], new_centroids[0][1], marker='*', s=200, c='r')  
plt.scatter(new_centroids[1][0], new_centroids[1][1], marker='*', s=200, c='b')  
plt.show()
```



In []: