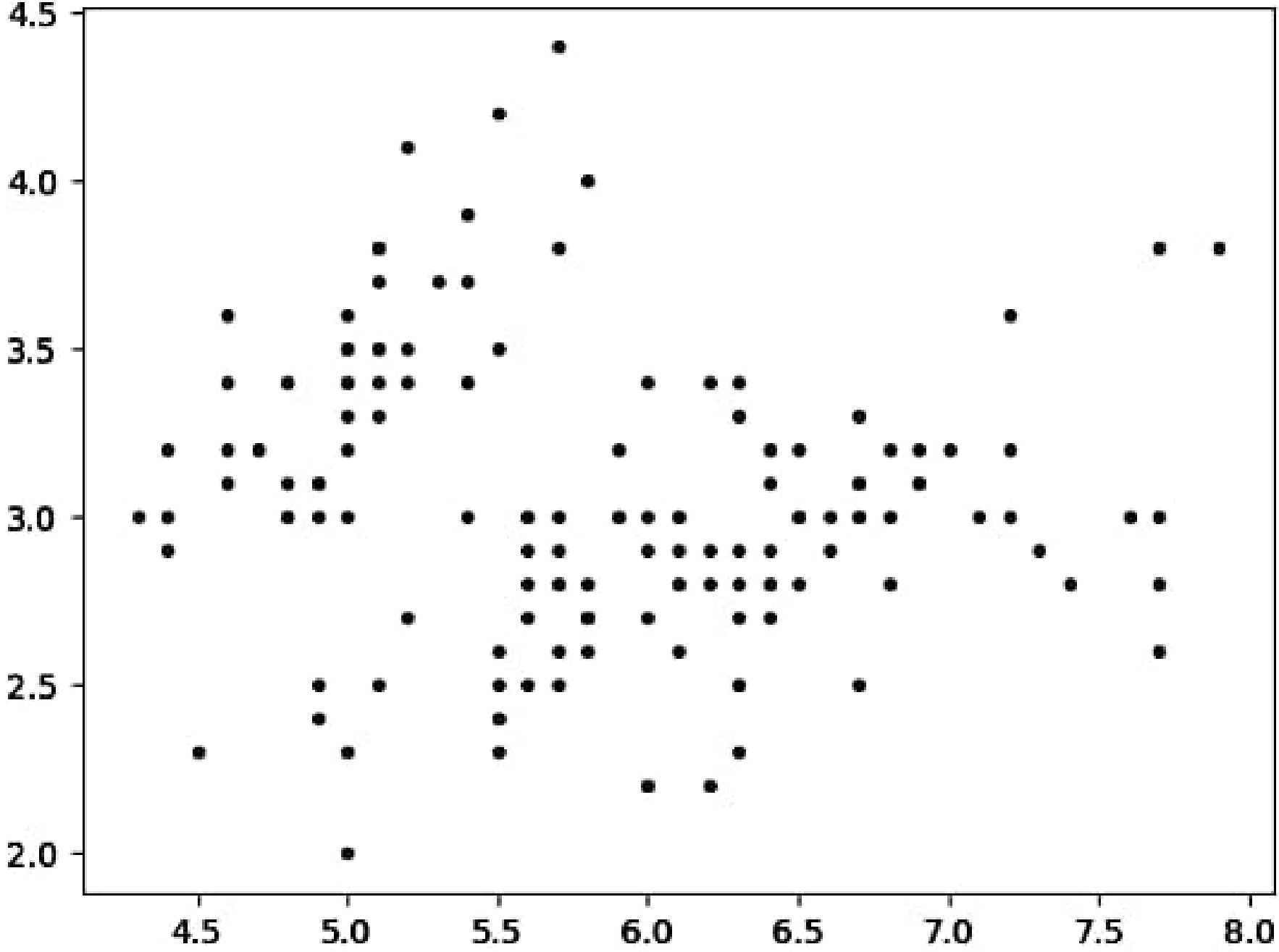
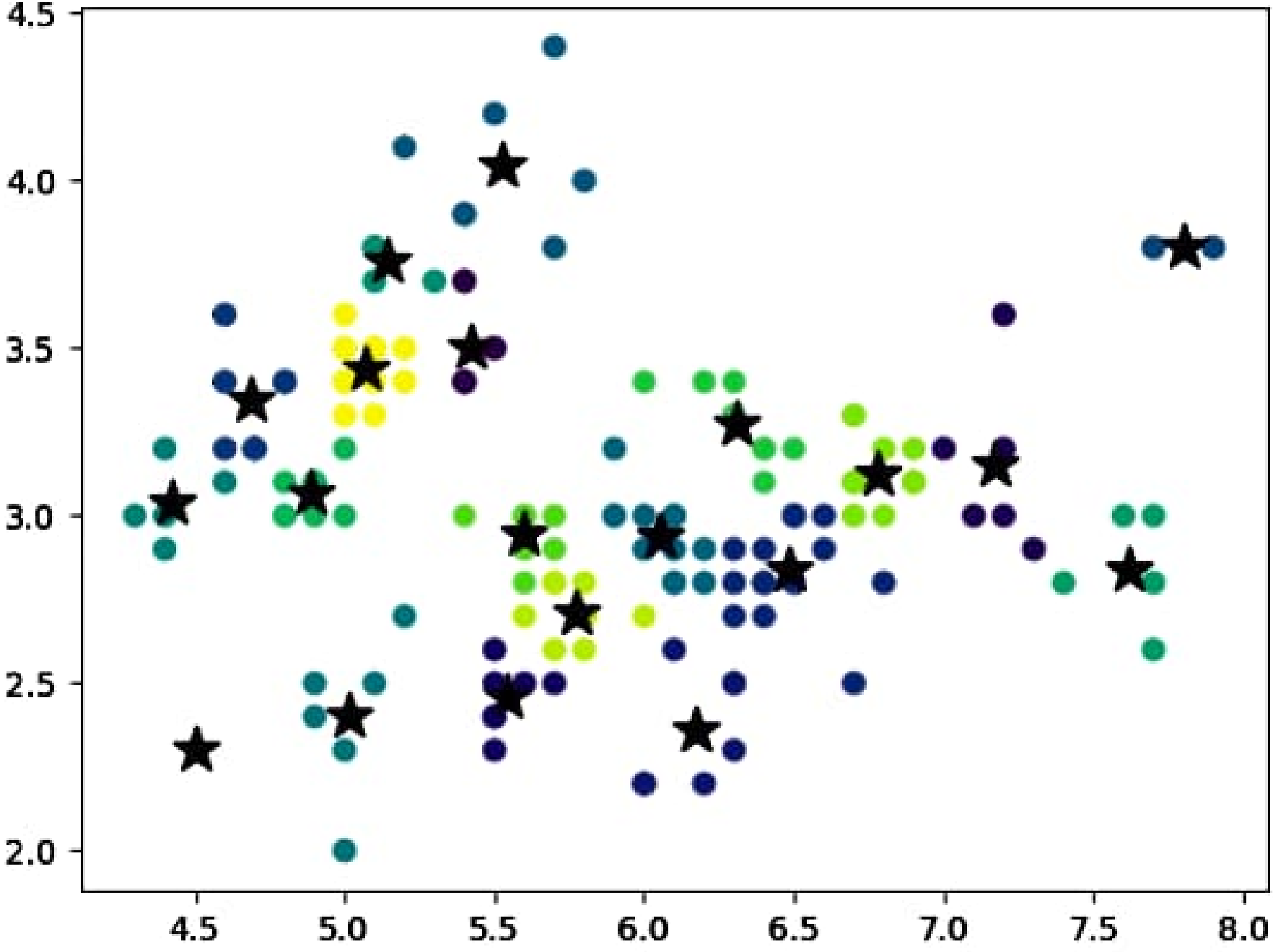
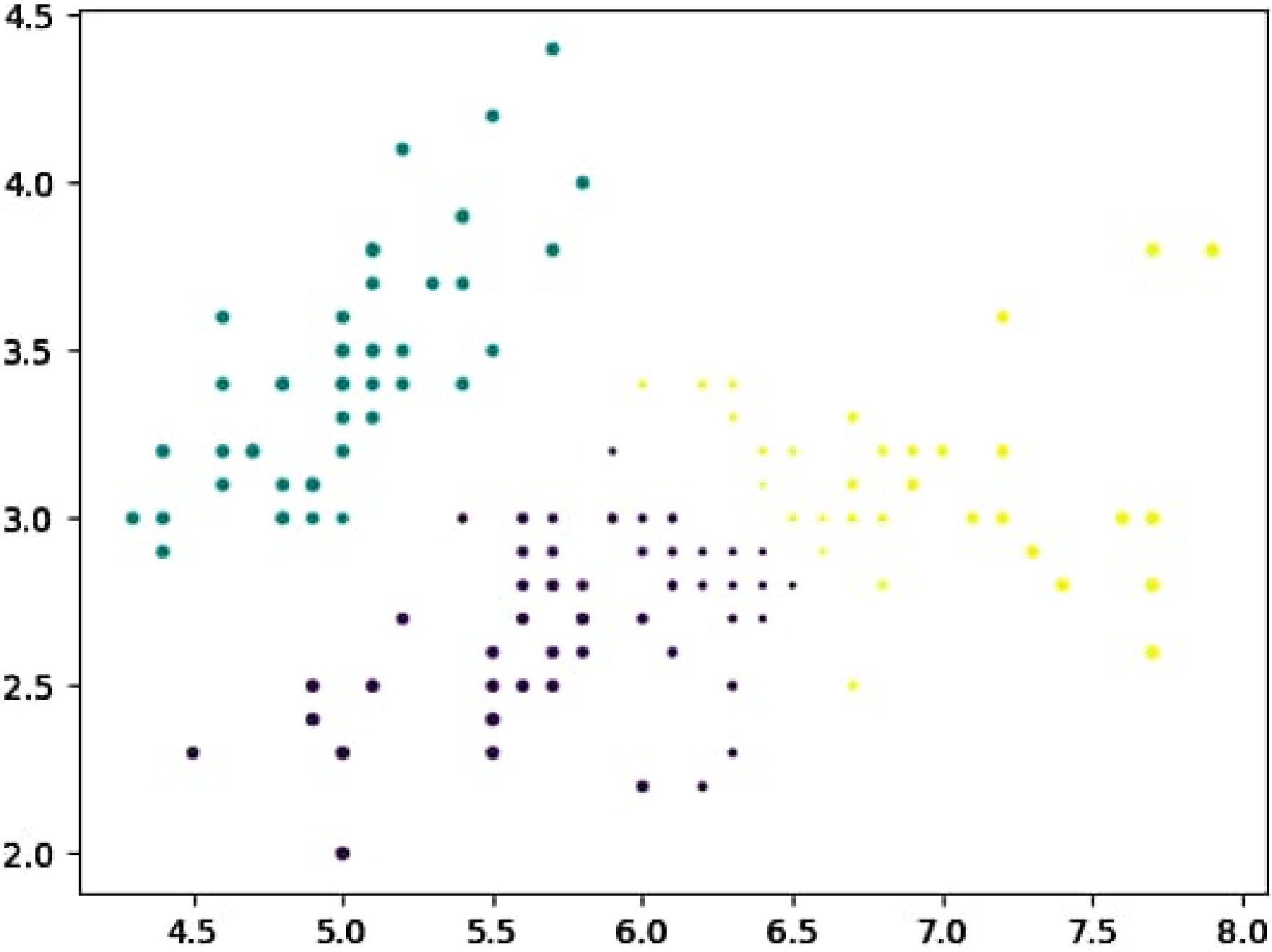
```python
"""AI&ML Lab-7:
Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for
clustering using k-Means algorithm. Compare the results of these two algorithms and comment
on the quality of clustering. You can add Java/Python ML library classes/API in the program.
"""

import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.mixture import GaussianMixture
from sklearn.cluster import KMeans

data=pd.read_csv('em.csv')
print("input data and shape")
print(data.shape)
data.head()
f1=data['v1'].values
f2=data['v2'].values
X=np.array(list(zip(f1,f2)))

print("X",X)
print('graph for whole dataset')
plt.scatter(f1,f2,c='black',s=7)
plt.show()

kmeans=KMeans(20,random_state=0)
labels=kmeans.fit(X).predict(X)
print("labels  ",labels)
centroids=kmeans.cluster_centers_
print("centroids  ",centroids)
plt.scatter(X[:,0],X[:,1],c=labels,s=40,cmap='viridis')
print('graph using kmeans algorithm')
plt.scatter(centroids[:,0],centroids[:,1],marker='*',s=200,c='#050505')
```

```
plt.show()

gmm=GaussianMixture(n_components=3).fit(X)
labels=gmm.predict(X)
probs=gmm.predict_proba(X)
size=10*probs.max(1)**3
print('graph using EM algorithm')
plt.scatter(X[:,0],X[:,1],c=labels,s=size,cmap='viridis')
plt.show()

"""
Output:

input data and shape
(150, 2)
X [[5.1 3.5]
 [4.9 3. ]
 [4.7 3.2]
 [4.6 3.1]
 [5.  3.6]
 [5.4 3.9]
 [4.6 3.4]
 [5.  3.4]
 .
 .
 .
 [6.7 3. ]
 [6.3 2.5]
 [6.5 3. ]
 [6.2 3.4]
 [5.9 3. ]]
graph for whole dataset

```

```
67 labels   [19 13  5 10 19  7 ....
68 .
69 .
70 16 16  3  4 14  8]
71 centroids   [[5.425      3.5        ]
72  [7.16666667 3.15      ]
73  [5.54285714 2.45714286]
74  [6.17142857 2.35714286]
75 .
76 .
77 .
78  [5.77272727 2.70909091]
79  [4.5        2.3       ]
80  [5.06666667 3.44166667]]
81 graph using kmeans algorithm
82 graph using EM algorithm
83 """
```


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0