Program 8

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 matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
import sklearn.metrics as sm
import pandas as pd
import numpy as np.

iris=datasets.load_iris()
X= pd.DataFrame(iris.data)
X.columns= ['Sepal_Length', 'Sepal_width, 'Petal_Length',
                                    'Petal_width']

y= pd.DataFrame(iris.target)
y.columns=['Targets']

model= KMeans(n_clusters=3)
model. fit(X)
model.labels_

plt.figure(figsize =(14,7))
```

```
colormap = np. array (['red'], 'lime', 'black'])

plt. subplot (1,2,1)
plt. scatter ( x. Petal_Length,  x. Petal_Width,  c= colormap
                                        [y. Targets], s = 40 )
plt. title ('Real  classification')

plt. subplot (1,2,2)
plt. scatter (x. Petal_Length,  x. Petal_Width,  c=colormap
                                        [model. labels_],  s=40)
plt. title ('K  Mean  classification')
plt. figure (figsize =(14,7))

predY = np. choose (model. labels_, [0,1,2]) . astype (np. int64)
print (predY)

plt. subplot(1,2,1)
plt. scatter (x. Petal_Length,  x. Petal_Width,  c= colormap [y.
                                             Targets],  s=40)
plt. title ('Real  classification')
plt. subplot (1,2,2)
plt. scatter (x. Petal_Length , x. Petal_Width  , c=colormap
                                             [predY],  s=40)
plt. title ('K mean  classification')

print ('The  accuracy  score  of  k-mean:', sm. accuracy_
                                   score (y, model. labels_))
print (' The  confusion  matrix of  k-mean:',  sm. confusion_matrix
                                       (y, model. labels_))
```

Teacher's Signature ____

```
from sklearn import preprocessing
scaler= preprocessing. Standardscaler()
scalor. fit(X)
xsa = scalar. transform(X)
xs= pd. DataFrame (xsa, columns= X. columns)

from sklearn. mixture import GaussianMixture.
gmm= GaussianMixture (n_components =3)
gmm. fit(xs)

y_cluster_gmm= gmm. predict (xs)

plt. support (2, 2, 3)
plt. scatter (X. Petal_Length, X. Petal_Width, c=colormap
                               [y_cluster_gm], s=40)
plt. title ('GMM Classification')

print( 'The accuracy sure of EM :', sm. accuracy_sure(y,
                                        y_cluster_gmm))
print( 'The confusion matrix of EM :', sm. confusion_matrix
                                   (y, y_cluster_gmm))
```

Output

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 0 2 2 2 2 0 2 2 2 0 2
0 2]
The accuracy sure of k-mean: 0.24
The confusion matrix of k-mean: [[0 50 0]
[45 0 2]
[14 0 36]]
The accuracy sure of EM: 0.0
The confusion matrix of EM: [ [0 50 0]
[5 0 45]
[50 0 0]]
```