7/12/20

EXPERIMENT NO. 8

UKS17CS030

AIM: 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 two algorithms and comment on the quality of clustering.

Algorithm:

1. An initial guess is made for the model's parameters and a probability distribution is created. This is sometimes called the "E-step" for the "expected" distribution.
2. Newly observed data is fed into the model.
3. The probability from the E-step is drawn to include the new data. This is sometimes called the "M-step".
4. Step 2 through 4 are repeated until stability.

Program:

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
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)
plt.figure(figsize=(14,14))
colormap = np.array(['red', 'lime', 'black'])
```

```
plt.subplot(2,2,1)
plt.scatter(x.Petal_Length, x.Petal_Width, c=colormap[y.targets], s=40)
plt.title('Real Clusters')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.subplot(2,2,2)
plt.scatter(x.Petal_Length, x.Petal_Width, c=colormap[model.labels_], s=40)
plt.title('K-Means Clustering')
plt.xlabel('Petal length')
plt.ylabel('Petal width')
from sklearn import preprocessing
scaler = preprocessing.StandardScaler()
scaler.fit(x)
xsa = scaler.transform(x)
xs = pd.DataFrame(xsa, columns = x.columns)
from sklearn.mixture import GaussianMixture
gmm = GaussianMixture(n_components = 3)
gmm.fit(xs)
gmm_y = gmm.predict(xs)
plt.subplot(2,2,3)
plt.scatter(x.Petal_Length, x.Petal_Width, c= colormap[gmm_y], s=40)
plt.title('GMM clustering')
plt.xlabel('Petal length')
plt.ylabel('Petal width')
print('observation: The GMM using EM algorithm based clustering
  matched the true labels more closely than the kmeans')
```

output: