

**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.**

```
from sklearn.cluster import KMeans
from sklearn import preprocessing
from sklearn.mixture import GaussianMixture
from sklearn.datasets import load_iris
import sklearn.metrics as sm
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dataset=load_iris()
# print(dataset)
X=pd.DataFrame(dataset.data)
X.columns=['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width']
y=pd.DataFrame(dataset.target)
y.columns=['Targets']
# print(X)
plt.figure(figsize=(14,7))
colormap=np.array(['red','lime','black'])

# REAL PLOT
plt.subplot(1,3,1)
plt.scatter(X.Petal_Length,X.Petal_Width,c=colormap[y.Targets],s=40)
plt.title('Real')
```

## # K-PLOT

```
plt.subplot(1,3,2)

model=KMeans(n_clusters=3)

model.fit(X)

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

plt.scatter(X.Petal_Length,X.Petal_Width,c=colormap[predY],s=40)

plt.title('KMeans')
```

## # GMM PLOT

```
scaler=preprocessing.StandardScaler()

scaler.fit(X)

xsa=scaler.transform(X)

xs=pd.DataFrame(xsa,columns=X.columns)

gmm=GaussianMixture(n_components=3)

gmm.fit(xs)

y_cluster_gmm=gmm.predict(xs)

plt.subplot(1,3,3)

plt.scatter(X.Petal_Length,X.Petal_Width,c=colormap[y_cluster_gmm],s=40)

plt.title('GMM Classification')
```

## OUTPUT

