***Task 6***

***Write a program to implement the Gaussian mixture model. Compute the accuracy of the classifier, considering few test data sets.***

**Algorithm**

1. Initialize the mean μk, the covariance matrix μk, and the mixing coefficients μk by some random variables.
2. Compute the μk
3. Again, estimate all the parameters using the current μk values
4. Compute log-likelihood function.
5. Put some convergence criterion
6. If the log-likelihood value converges to some value (or if all the parameters converge to some values) then stop, else return to Step 2.

**Python Code**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from pandas import DataFrame

from sklearn import datasets

from sklearn.mixture import GaussianMixture

# load the iris dataset

iris = datasets.load\_iris()

# select first two columns

X = iris.data[:, :2]

# turn it into a dataframe

d = pd.DataFrame(X)

# plot the data

plt.scatter(d[0], d[1])

gmm = GaussianMixture(n\_components = 3)

# Fit the GMM model for the dataset

# which expresses the dataset as a

# mixture of 3 Gaussian Distribution

gmm.fit(d)

# Assign a label to each sample

labels = gmm.predict(d)

d['labels']= labels

d0 = d[d['labels']== 0]

d1 = d[d['labels']== 1]

d2 = d[d['labels']== 2]

plt.show()

# plot three clusters in same plot

plt.scatter(d0[0], d0[1], c ='r')

plt.scatter(d1[0], d1[1], c ='yellow')

plt.scatter(d2[0], d2[1], c ='g')

plt.show()

# print the converged log-likelihood value

print("The mean:",gmm.means\_)

print('\n')

print("The covariance matrix:",gmm.covariances\_)

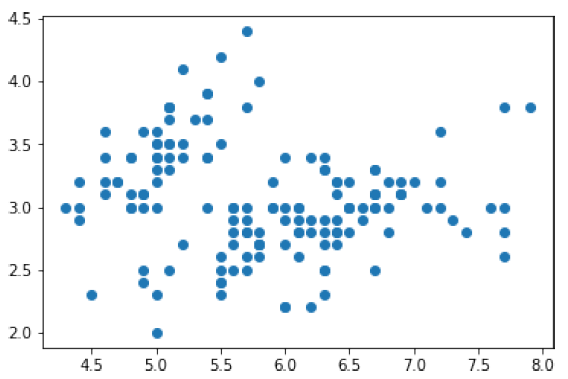
print("The converged log-likelihood value:",gmm.lower\_bound\_)

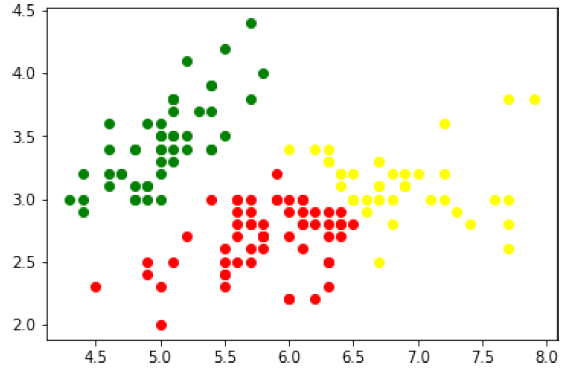
# print the number of iterations needed

# for the log-likelihood value to converge

print("The number of iterations needed for the log-likelihood value to converge:",gmm.n\_iter\_)

**Output:**





The mean:

[[5.9009976 2.74387546]

[6.68055626 3.02849627]

[5.01507898 3.4514463 ]]

The covariance matrix:

[[[0.27671149 0.08897036]

[0.08897036 0.09389206]]

[[0.36153508 0.05159664]

[0.05159664 0.08927917]]

[[0.11944714 0.08835648]

[0.08835648 0.11893388]]]

The converged log-likelihood value: -1.4987505566235166

The number of iterations needed for the log-likelihood value to converge: 8