In this homework, I implemented an expectation-maximization (EM) clustering algorithm.

Note: I took Gaussian Ellipse code from someone's github repository and I referenced it on the code.

Part 1

I read data from hw07_data_set which contains 300 data points generated randomly from five bivariate Gaussian densities.

Part 2

To initialize your EM algorithm, I took the centroids given in the file named hw07_initial_centroids.csv as the initial values for the mean vectors.

```
#initialize centroids,memberships
centroids = np.genfromtxt("hw07_initial_centroids.csv", delimiter = ",")
memberships = update_memberships(centroids, X)
print(centroids)
```

```
[[-3.0439416   0.32509753]
 [-0.30377397  3.73480678]
 [ 1.17587258 -2.50984601]
 [ 0.91037084  0.87608158]
 [-1.98588337  2.98466811]]
```

By assigning the data points to the nearest center, I estimated the initial covariance matrices and prior probabilities before applying EM algorithm.

Part 3

After the initialization step, I implemented EM algorithm.

E Step:

$$h_{ik} = E\left[z_{ik} | X, \Phi^{(t)}\right] = \frac{p(x_i | C_k, \Phi^{(t)}) P(C_k)}{\sum_{c=1}^{K} p(x_i | C_c, \Phi^{(t)}) P(C_c)}$$

M Step:

$$\hat{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^{N} h_{ik} x_i}{\sum_{i=1}^{N} h_{ik}}$$

$$\hat{\sigma}_k^{(t+1)} = \frac{\sum_{i=1}^{N} h_{ik} (x_i - \hat{\mu}_k^{(t+1)})(x_i - \hat{\mu}_k^{(t+1)})^T}{\sum_{i=1}^{N} h_{ik}}$$

$$\hat{P}(C_K) = \frac{\sum_{i=1}^{N} h_{ik}}{N}$$

For M Step I implemented three functions to update means, covariances and priors.

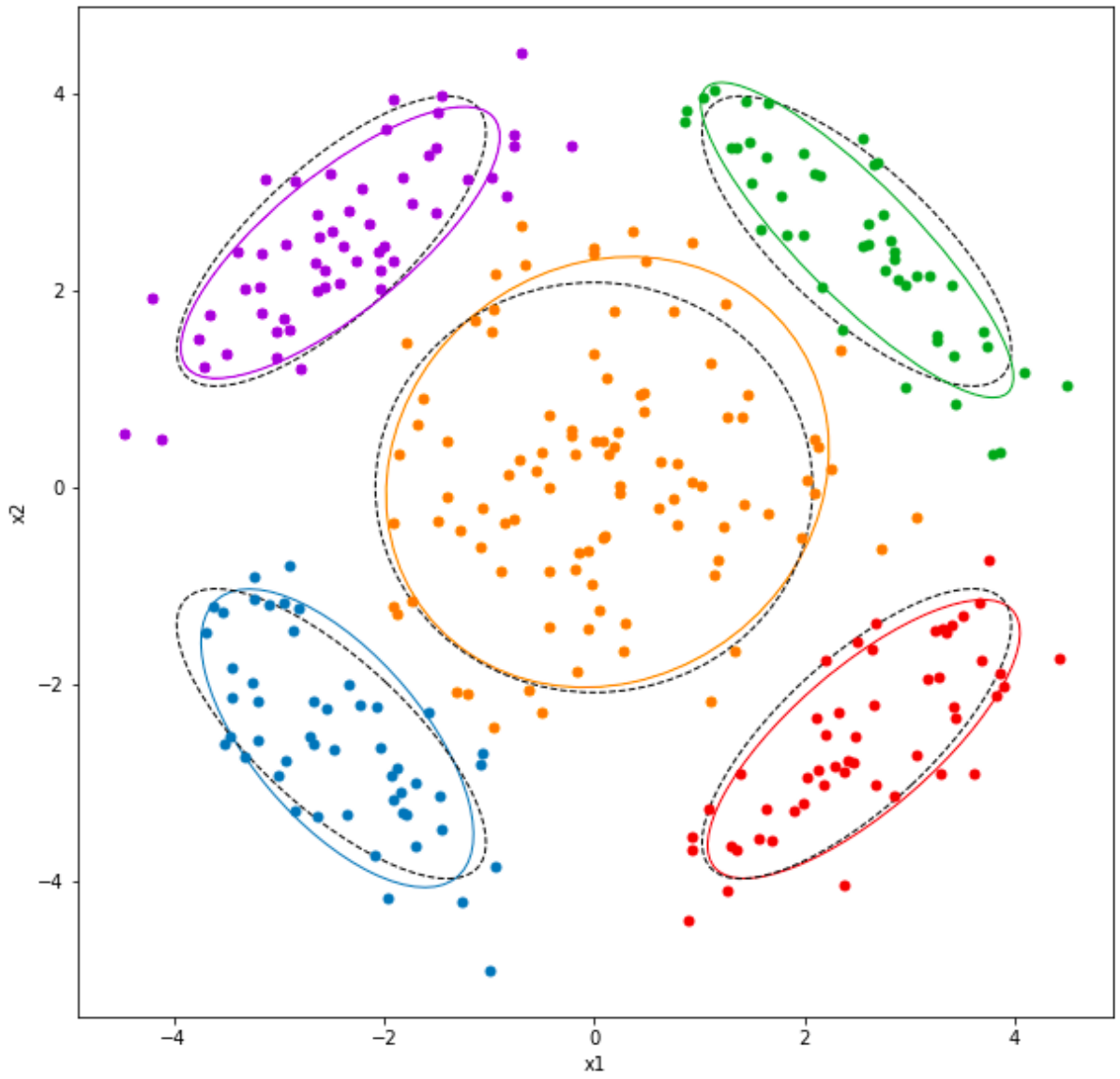
After defining functions, I run EM algorithm for 100 iterations. Then centroids which is mean vectors are found as follows:

```
print('centroids')
print(centroids)
```

```
centroids
[[-2.44390052 -2.5453942 ]
 [ 2.50354332  2.51134859]
 [ 2.56726404 -2.55477253]
 [ 0.12794677  0.15595776]
 [-2.41465305  2.4855615 ]]
```

Part 4

I drew the clustering result obtained by your EM algorithm by coloring each cluster with a different color. I also drew the original Gaussian densities used to generate data points with dashed lines and the Gaussian densities EM algorithm with solid lines. I used gaussian ellipse code which is taken from github that is referenced above function.


4
2
0
−2
−4
x2
−4
−2
0
2
4
x1