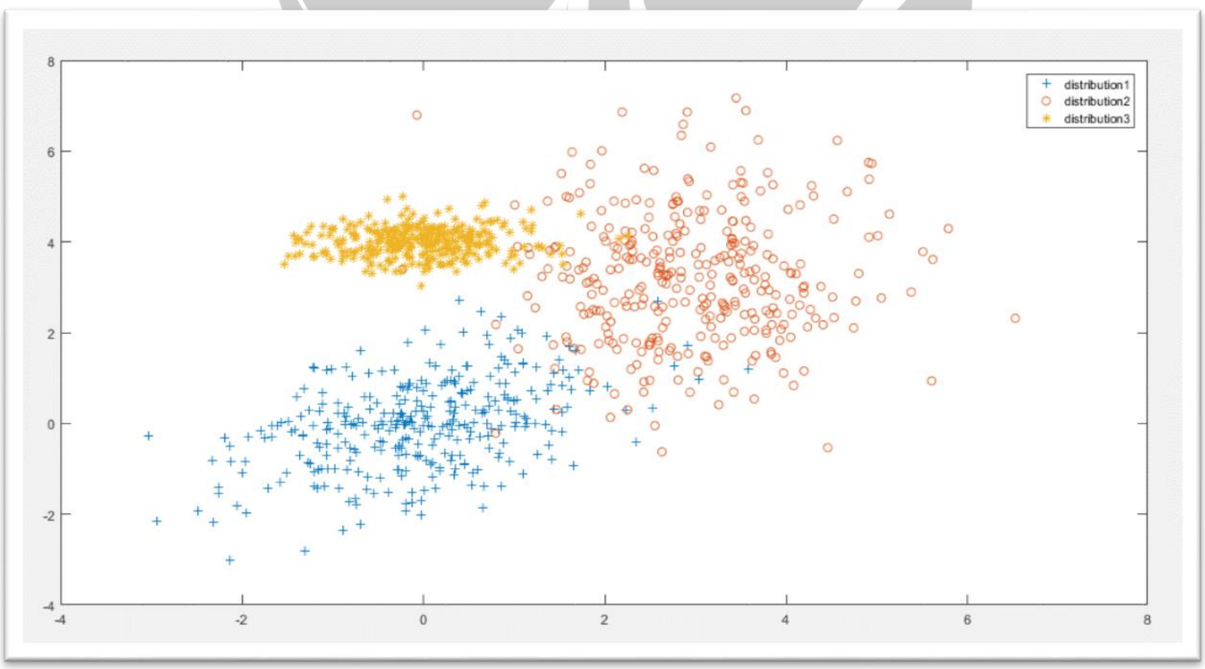
GAUSSIAN MIXTURE MODELS
MACHINE LEARNING

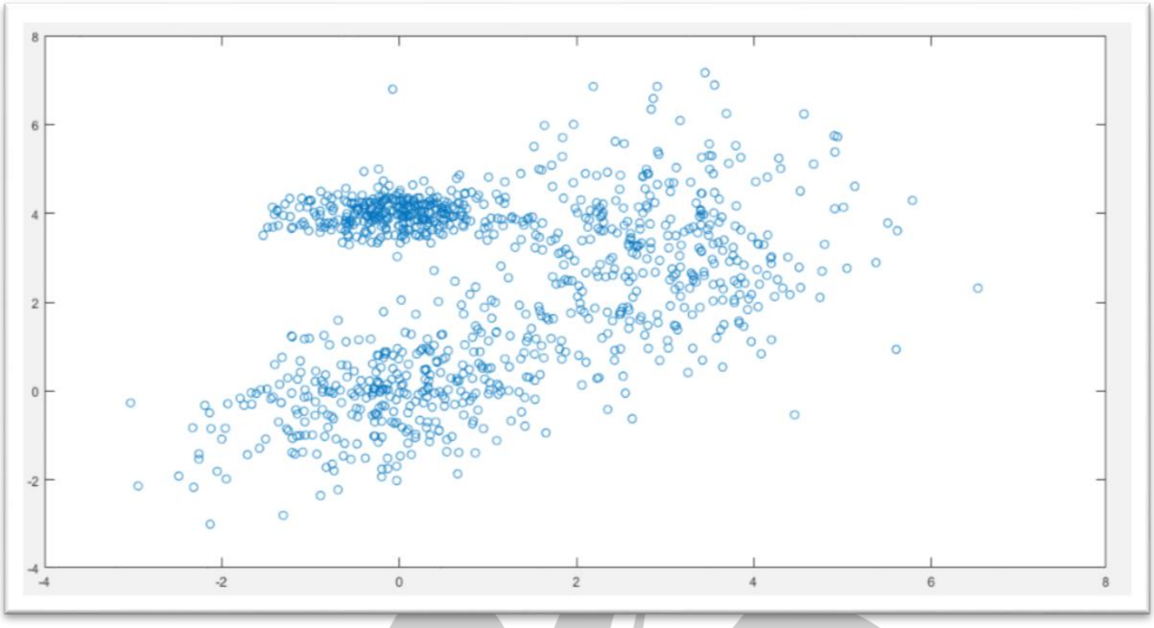
Mahnoor Anjum Dated: 4/30/2018

TASK 1:

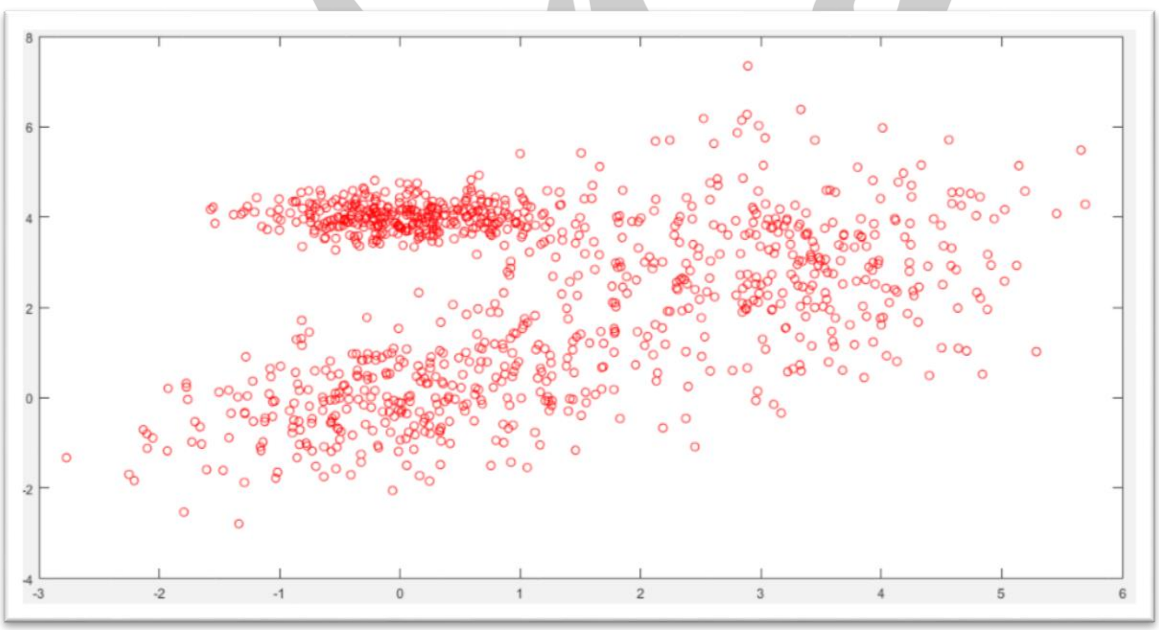
```
%The following is a detailed coding of the multivariate gaussians
%Presented by Mahnoor Anjum for Dr. Hassan Aqeel Khan
%the references will be detailed in the end
%============================================================
%I will be starting with multivariate gaussians through mvnrnd()
%Let us create the mean and sigma matrixes as specified in the assignment
u1 = [0, 0];
u2 = [3,3];
u3 = [0,4];
sigma1 = [1,0.4;0.4,1];
sigma2 = [1,0;0,2];
sigma3 = [0.4,0;0,0.1];
weights = [1/3, 1/3,1/3];
dist1 = mvnrnd(u1,sigma1,333);
dist2 = mvnrnd(u2,sigma2,333);
dist3 = mvnrnd(u3,sigma3,333);
%The following is the result of three separate gaussians
```




```
%Now We will be creating ONE distribution i.e a mixture of gaussians:
finaldist = [dist1;dist2;dist3];
plot(finaldist(:,1), finaldist(:,2), 'o');
```




```
%let's create the same distribution with a better function in the following
%steps:
combinedu = [u1;u2;u3];
combinedsigma=cat(3,sigma1,sigma2,sigma3);
weights

    0.3333    0.3333    0.3333

gmmodel = gmdistribution(combinedu, combinedsigma, weights);
X = random(gmmodel,1000);
plot(X(:,1),X(:,2), 'ro')
```



TASK 2:

Without the Kmeans algorithm, we get the following results:

```
obj = gmdistribution.fit(X,3);

obj =

Gaussian mixture distribution with 3 components in 2 dimensions

Component 1:

Mixing proportion: 0.344373

Mean:    0.0614    4.0093

Component 2:

Mixing proportion: 0.340421

Mean:    3.0507    2.9603

Component 3:

Mixing proportion: 0.315205

Mean:    0.0549    0.0450
```

1- LET'S ESTIMATE THE MEAN PRIORS AND COVARIANCES WITH KMEANS

Now we estimate the parameters with Kmeans.

```
[prior mu sigma]=EM_init_kmeans(X.', 3)

prior =

    0.3780    0.3000    0.3220

mu =

    0.1724    3.2713    0.0944

    3.9988    2.9572    0.0122

sigma(:,:,1) =

    0.5208    0.0274

    0.0274    0.2308

sigma(:,:,2) =

    0.7996    0.1268

    0.1268    1.5067

sigma(:,:,3) =

    1.0387    0.3249

    0.3249    0.7883
```

2- LET'S FIT THE MODEL WITH THE ESTIMATES

```
s = struct('mu',mu.','Sigma',sigma,'PComponents',prior);

options = statset('Display','final');

e =1e-5;

GMModel=gmdistribution.fit(X,3,'CovType','full','Options',options,'Start',s,'
Regularize',e);

 muModel = GMModel.mu; SigModel = GMModel.Sigma;
```

16 iterations, log-likelihood = -3343.06

Lets look at the model:

```
GMModel =

Gaussian mixture distribution with 3 components in 2 dimensions

Component 1:

Mixing proportion: 0.315339

Mean:     0.0555     0.0456

Component 2:

Mixing proportion: 0.340249

Mean:     3.0515     2.9608

Component 3:

Mixing proportion: 0.344412

Mean:     0.0615     4.0092
```

3- COMPARISON OF TRUE VALUES WITH THE MODEL:

The final values we have obtained are as follows:

Distribution 1:

Mean:
[0.0555   0.0456]

Covariance:

0.5208	0.0274
0.0274	0.2308

Mixing proportion: 0.315339

The original value of mixing is 1/3=33.33%, here we have 31.53% of distribution 1.
The mean originally was centered around 0,0. We obtain a slight shift towards the north east. The covariance matrix observes the most deflection, we varied

the original by [1 0.4; 0.4 1], this distribution is somewhat smaller/more compact than the original. i.e some of the datapoints of the original distribution have been assigned to other distributions during the mixing.

Distribution 2:

Mean: 3.0515 2.9608

Covariance:

0.7996	0.1268
0.1268	1.5067

Mixing proportion: 0.340249

The original value of mixing is 1/3=33.33%, here we have 34.02% of distribution 1.
The mean originally was centered around 3,3. We obtain a slight shift towards the south east.

Distribution 3:

Mean: 0.0615 4.0092

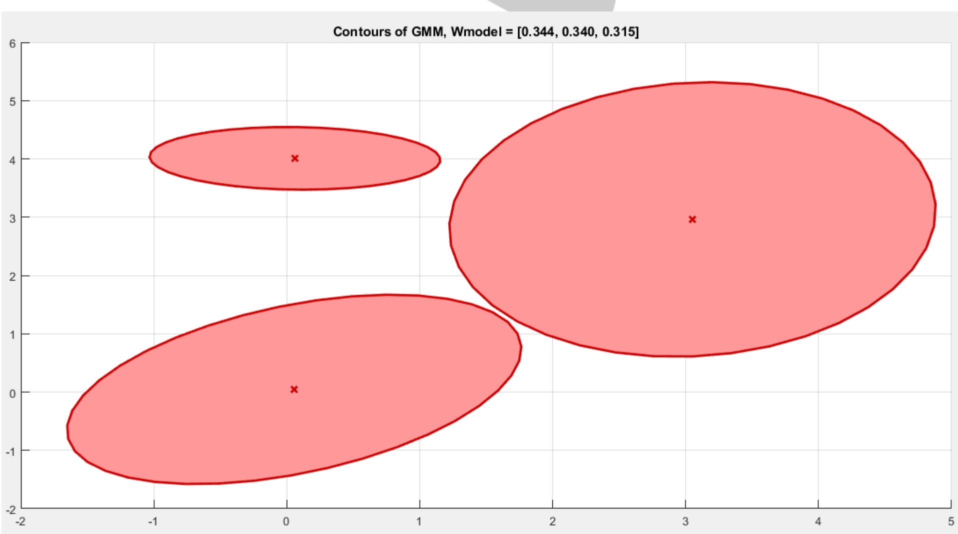
Covariance:

1.0387	0.3249
0.3249	0.7883

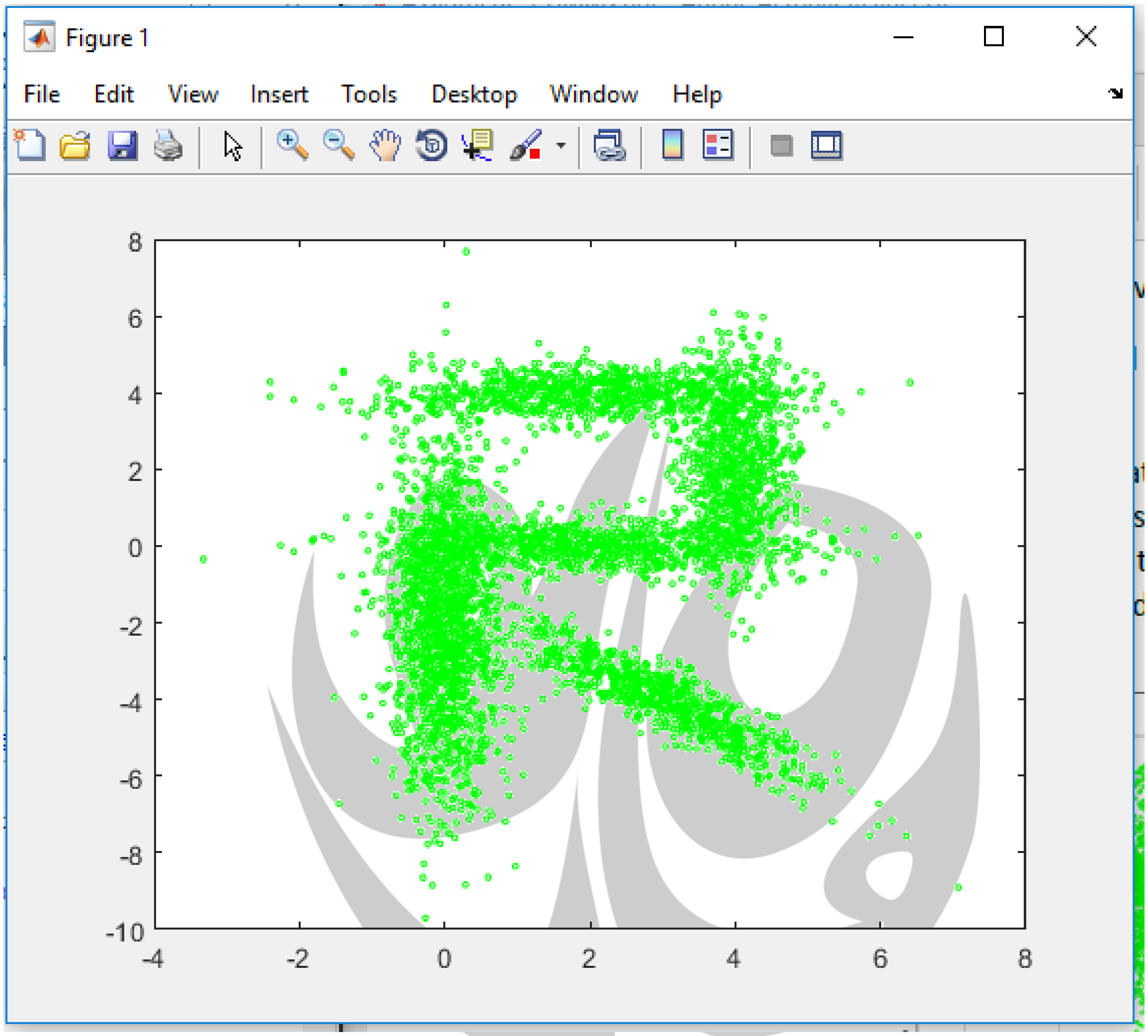
Mixing proportion: 0.344412

The original value of mixing is 1/3=33.33%, here we have 34.44% of distribution 1.
The mean originally was centered around 0,4. We obtain a slight shift towards the north east.

TASK 4:

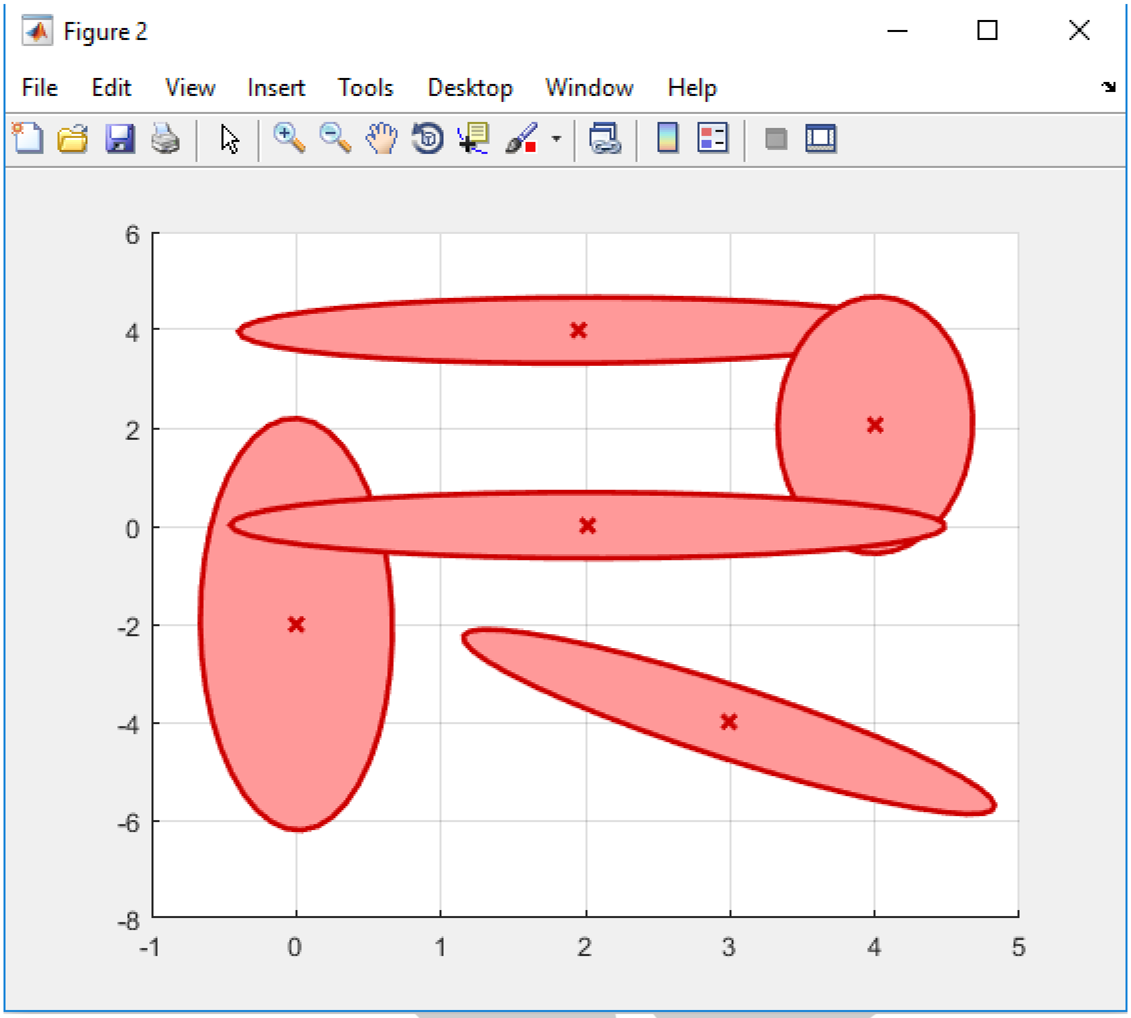


TASK 5:




```
>> [prior mu sigma]=EM_init_kmeans(X.', 5);
>>  s = struct('mu',mu.','Sigma',sigma,'PComponents',prior);
options = statset('Display','final');
e =1e-5;
GMModel
=gmdistribution.fit(X,5,'CovType','full','Options',options,'Start',s,'Regular
ize',e);
 muModel = GMModel.mu; SigModel = GMModel.Sigma;
22 iterations, log-likelihood = -18947.8
>> figure;
>> hold on; grid on;
```

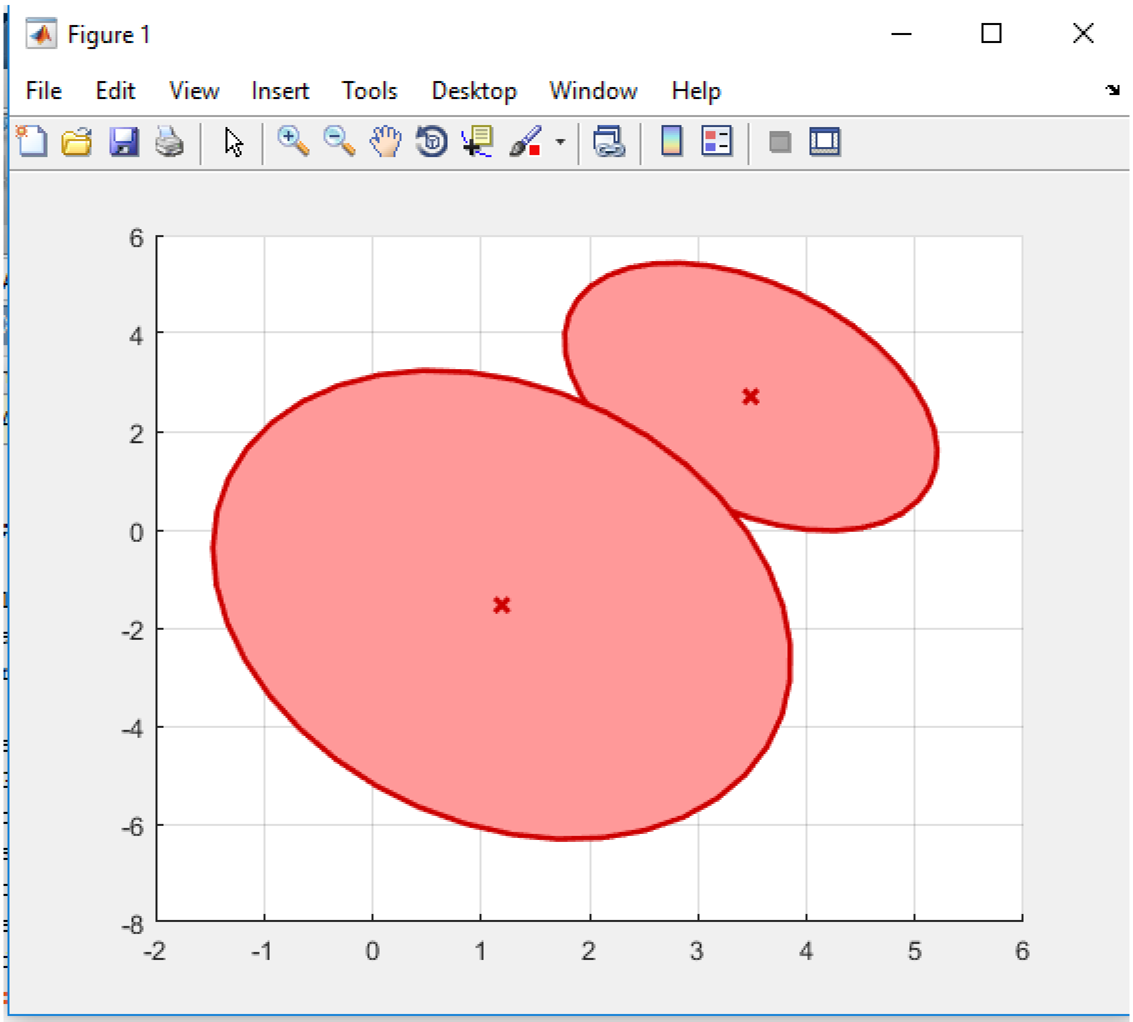
```
>> Wmodel = GMModel.ComponentProportion;
>> plotGMM(muModel', SigModel, [.8 0 0],1);
```



TASK 6:

```
>> [prior mu sigma]=EM_init_kmeans(X.', 2);
>> s = struct('mu',mu.','Sigma',sigma,'PComponents',prior);
>> options = statset('Display','final');
>> e =1e-5;
>> GMModel
=gmdistribution.fit(X,2,'CovType','full','Options',options,'Start',s,'Regular
ize',e);
Warning: Failed to converge in 100 iterations for gmdistribution with 2
components
> In gmcluster (line 198)
  In gmdistribution.fit (line 95)
```

```
100 iterations, log-likelihood = -22070.2
>> muModel = GMModel.mu; SigModel = GMModel.Sigma;
>> figure;
>> hold on; grid on;
>> Wmodel = GMModel.ComponentProportion;
>> plotGMM(muModel', SigModel, [.8 0 0],1);
```

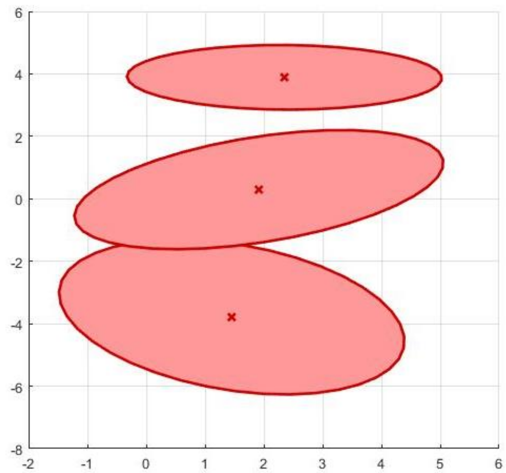


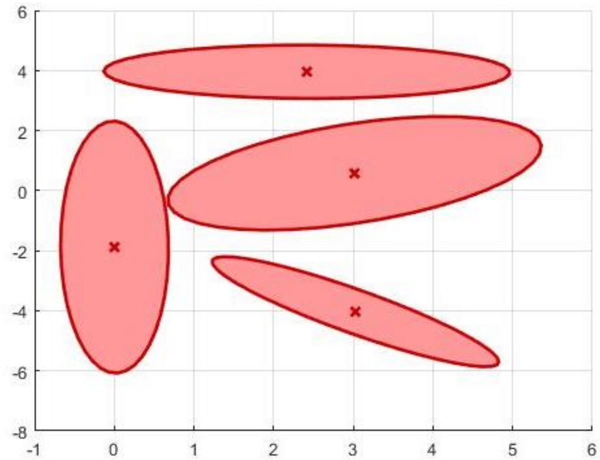
TASK7:

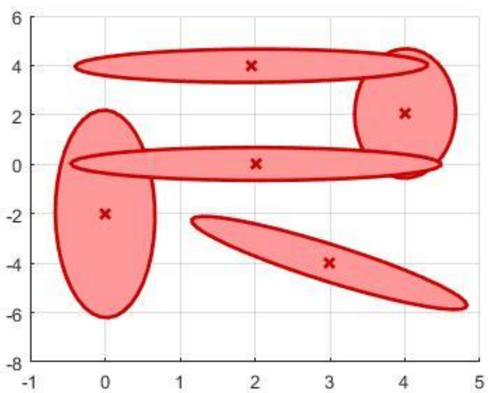
```
>> for k=3:9
[prior mu sigma]=EM_init_kmeans(X.', k);
s = struct('mu',mu.','Sigma',sigma,'PComponents',prior);
options = statset('Display','final');
e =1e-5;
```

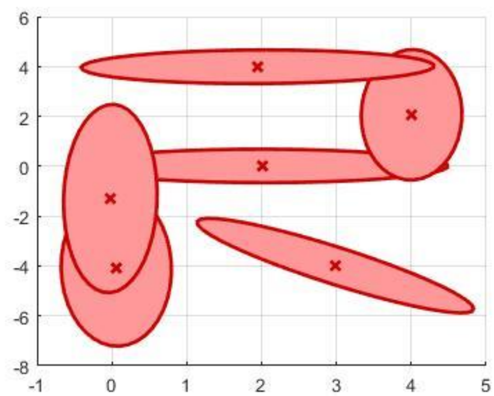
```
GMModel
=gmdistribution.fit(X,k,'CovType','full','Options',options,'Start',s,'Regular
ize',e);

muModel = GMModel.mu; SigModel = GMModel.Sigma;

figure;

hold on; grid on;

Wmodel = GMModel.ComponentProportion;

plotGMM(muModel', SigModel, [.8 0 0],1);

i = k-2;

Gm2BIC(i)=GMModel.BIC;

End
61 iterations, log-likelihood = -21492.8

77 iterations, log-likelihood = -19714.6

22 iterations, log-likelihood = -18947.8

45 iterations, log-likelihood = -18944

84 iterations, log-likelihood = -18936.4

Warning: Failed to converge in 100 iterations for gmdistribution with 8
components

> In gmcluster (line 198)

  In gmdistribution.fit (line 95)

100 iterations, log-likelihood = -18941.5

97 iterations, log-likelihood = -18938.6

Gm2BIC =

   1.0e+04 *

    4.3130    3.9625    3.8143    3.8186    3.8222    3.8283    3.8329
```

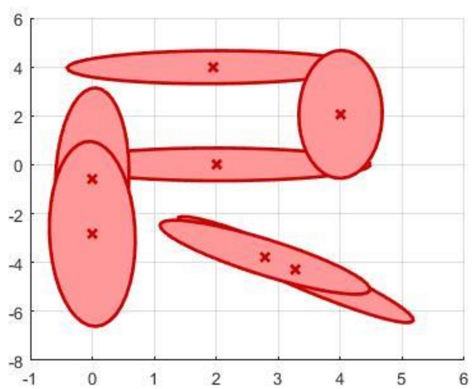
The following plots show the GMMs of 3,4,5,6,7 and 8 gaussians.

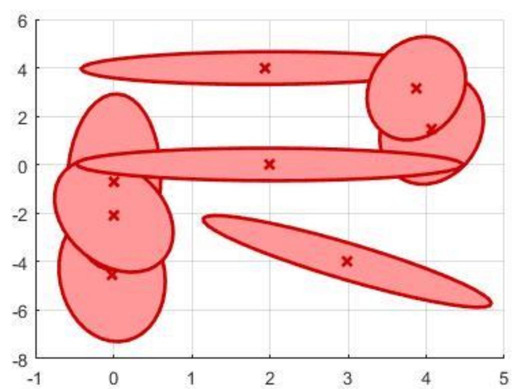




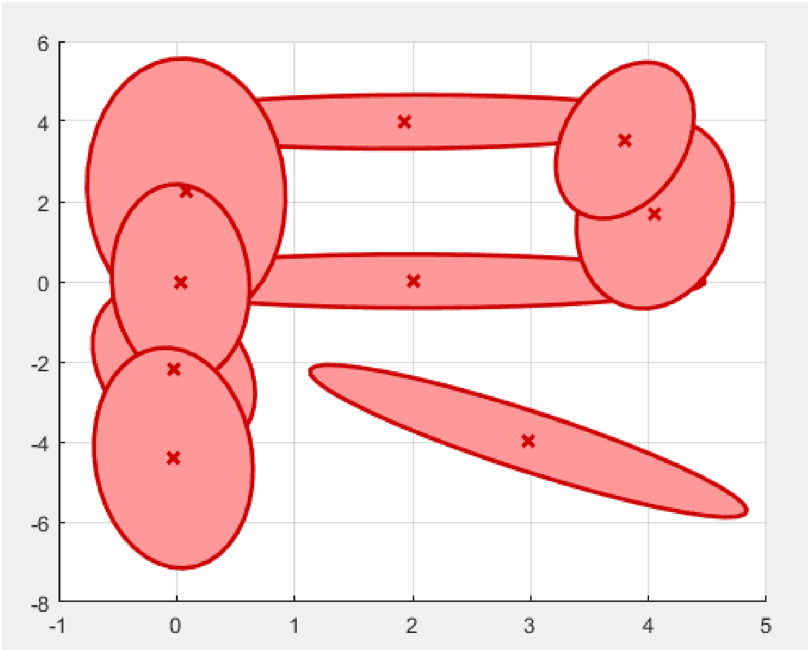






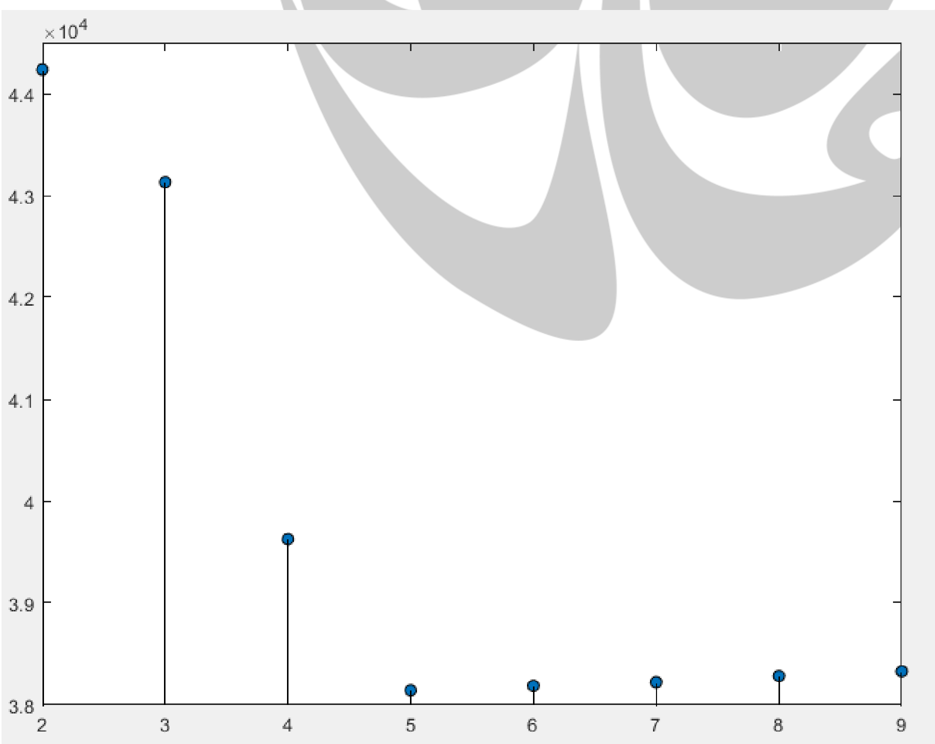


The following plots show the GMMs of 9 gaussians.



TASK 8:

BIC PLOT



As we can see from the following plot, we get the least value of BIC for K=5.
So the best model will consist of, as assumed in task 2, 5 Gaussians.

REFERENCES

https://www.mathworks.com/help/stats/gmdistribution.fit.html