5. a) Below are the mu and covariance matrices for 4 classes.

mu1 =	mu2 =	mu3 =	mu4 =
7.9167	5.7083	7.5833	8.5000
10.4167	5.2917	7.4167	8.2917
8.3333	5.4167	6.6250	6.9583
5.5833	6.3333	6.0417	6.1250
9.5833	4.6250	5.5417	9.2917
11.1667	4.3750	5.6250	9.4167
3.1250	7.9167	3.8333	6.6667
3.6250	9.6250	4.1667	7.4167

covariance1 =

7.6597	0.3264	2.1528	2.4653	1.3819	-1.6528	-1.9062	-0.9896
0.3264	5.9097	1.3611	-1.6181	-0.9514	1.9722	-1.5104	-0.9271
2.1528	1.3611	11.6389	1.7222	-1.1111	1.5694	-1.0000	0.1250
2.4653	-1.6181	1.7222	5.7431	0.8681	-1.3889	0.0104	-0.8229
1.3819	-0.9514	-1.1111	0.8681	1.5764	-1.2222	-0.5313	-0.6146
-1.6528	1.9722	1.5694	-1.3889	-1.2222	1.7222	0.0208	0.3542
-1.9062	-1.5104	-1.0000	0.0104	-0.5313	0.0208	2.2760	1.3385
-0.9896	-0.9271	0.1250	-0.8229	-0.6146	0.3542	1.3385	1.6510

covariance2 =

2.7066	1.1684	0.2882	0.0556	1.2656	-0.0156	-2.3576	-0.4427
1.1684	1.7899	0.3368	1.3611	0.5260	0.2240	-0.8924	0.5260
0.2882	0.3368	1.8264	-0.5556	-0.2187	0.1771	0.5764	-0.6354
0.0556	1.3611	-0.5556	9.8056	-1.0833	-0.8333	1.5694	1.6667
1.2656	0.5260	-0.2187	-1.0833	1.2344	0.3906	-1.8646	0.1510
-0.0156	0.2240	0.1771	-0.8333	0.3906	0.9010	-0.7188	0.1406
-2.3576	-0.8924	0.5764	1.5694	-1.8646	-0.7188	8.6597	-1.9063
-0.4427	0.5260	-0.6354	1.6667	0.1510	0.1406	-1.9063	5.9844

covariance3 =

0.8264	0.3819	0.1354	0.3090	0.9757	0.5521	0.3889	0.2361
0.3819	0.8264	0.0729	0.1493	0.6493	0.9062	-0.1806	0.1806
0.1354	0.0729	0.5677	0.3490	0.4948	0.5260	-0.0208	-0.2708
0.3090	0.1493	0.3490	0.7899	0.6441	0.5156	0.0903	0.0764
0.9757	0.6493	0.4948	0.6441	2.0816	1.3698	-0.1597	-0.3819
0.5521	0.9062	0.5260	0.5156	1.3698	1.8177	-0.4792	-0.2708
0.3889	-0.1806	-0.0208	0.0903	-0.1597	-0.4792	1.3056	0.8194
0.2361	0.1806	-0.2708	0.0764	-0.3819	-0.2708	0.8194	1.3056

covariance4 =

3.5833	-0.2708	-1.8542	-0.6458	1.1042	-0.5833	-3.5000	2.0417
-0.2708	5.1233	0.9705	0.5469	0.1233	0.5451	1.9306	-3.4132
-1.8542	0.9705	4.4566	0.6302	-0.6962	1.4340	1.1944	-1.3993
-0.6458	0.5469	0.6302	4.2760	-0.0365	-0.3437	2.0000	-2.1771
1.1042	0.1233	-0.6962	-0.0365	0.8733	-0.0799	-0.7778	0.5035
-0.5833	0.5451	1.4340	-0.3437	-0.0799	0.9097	0.1806	-0.3819
-3.5000	1.9306	1.1944	2.0000	-0.7778	0.1806	8.7222	-3.1111
2.0417	-3.4132	-1.3993	-2.1771	0.5035	-0.3819	-3.1111	6.9097

The MATLAB code that I have used for calculation of the above measures is given below:

```
% load data
load 'imox.data';

% get train data
train1 = imox(1:24,1:8);
train2 = imox(49:72,1:8);
train3 = imox(97:120,1:8);
train4 = imox(145:168,1:8);

% define variables
mu1 = mean(train1)'
mu2 = mean(train2)'
mu3 = mean(train3)'
mu4 = mean(train4)'
```

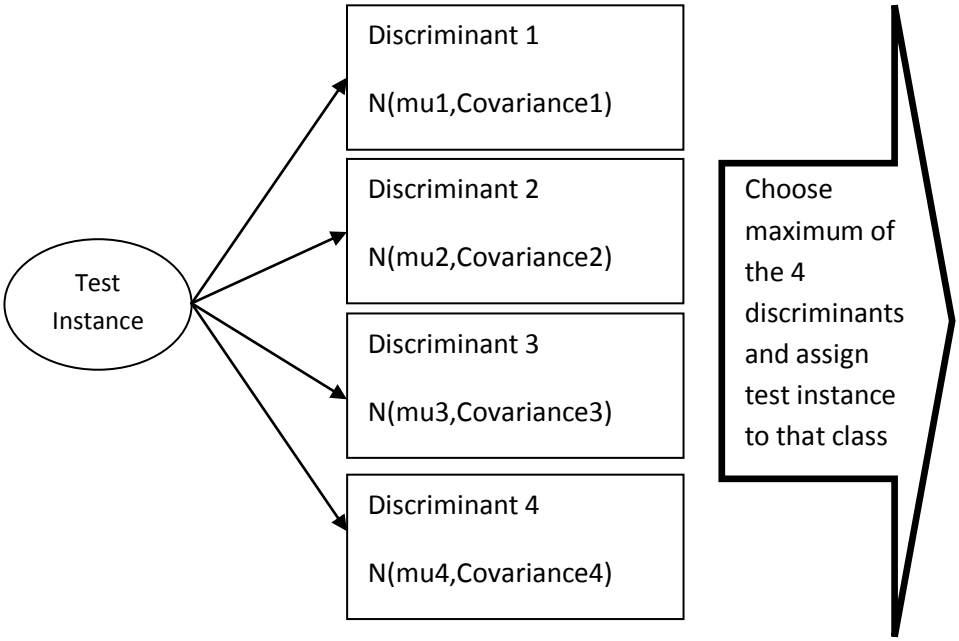
```
covariance1 = zeros(size(mu1,1));
for i = 1:size(train1,1)
    covariance1 = covariance1 + (train1(i,:)' - mu1) * (train1(i,:)' - mu1)';
end
covariance1 = covariance1 / size(train1,1)

covariance2 = zeros(size(mu2,1));
for i = 1:size(train2,1)
    covariance2 = covariance2 + (train2(i,:)' - mu2) * (train2(i,:)' - mu2)';
end
covariance2 = covariance2 / size(train2,1)

covariance3 = zeros(size(mu3,1));
for i = 1:size(train3,1)
    covariance3 = covariance3 + (train3(i,:)' - mu3) * (train3(i,:)' - mu3)';
end
covariance3 = covariance3 / size(train3,1)

covariance4 = zeros(size(mu4,1));
for i = 1:size(train4,1)
    covariance4 = covariance4 + (train4(i,:)' - mu4) * (train4(i,:)' - mu4)';
end
covariance4 = covariance4 / size(train4,1)
```

5.b) In that problem, we have four different classes and therefore we will have four different discriminant functions. Normally, the discriminant function in our case would be the posterior probability. In other words, for each data point in our test set, we execute each one of the four different discriminant functions and we assign this test instance to the class whose posterior has yielded the highest value. In our case, since the priors are given as equal, our discriminant functions can be the likelihood values, that is, we assign our test instance to the class for whom the likelihood value was maximum. Since likelihood values are normal distributions, we can define our classification model as in the figure given below:



5.c) The confusion matrix that we elicit from the classifier described above is given below:

	Class1 Predicted	Class2 Predicted	Class3 Predicted	Class4 Predicted
Class1 Actual	24	0	0	0
Class2 Actual	0	24	0	0
Class3 Actual	0	1	23	0
Class4 Actual	5	0	0	19

As we can see form the above confusion matrix, **24+24+23+19=90** instances were correctly classified and **6** of them were misclassified. Then our overall accuracy prediction accuracy will be (90*100)/96 = **93.75%**.

MATLAB code that I have written to calculate the above confusion matrix is given below:

```
% load data
load 'imox.data';

% get train data
train1 = imox(1:24,1:8);
train2 = imox(49:72,1:8);
train3 = imox(97:120,1:8);
train4 = imox(145:168,1:8);

% define variables
mu1 = mean(train1)'
mu2 = mean(train2)'
mu3 = mean(train3)'
mu4 = mean(train4)'

covariance1 = zeros(size(mu1,1));
for i = 1:size(train1,1)
    covariance1 = covariance1 + (train1(i,:)' - mu1) * (train1(i,:)' - mu1)';
end
covariance1 = covariance1 / size(train1,1)

covariance2 = zeros(size(mu2,1));
for i = 1:size(train2,1)
    covariance2 = covariance2 + (train2(i,:)' - mu2) * (train2(i,:)' - mu2)';
end
covariance2 = covariance2 / size(train2,1)

covariance3 = zeros(size(mu3,1));
for i = 1:size(train3,1)
    covariance3 = covariance3 + (train3(i,:)' - mu3) * (train3(i,:)' - mu3)';
end
covariance3 = covariance3 / size(train3,1)
```

```
covariance4 = zeros(size(mu4,1));
for i = 1:size(train4,1)
    covariance4 = covariance4 + (train4(i,:)' - mu4) * (train4(i,:)' - mu4)';
end
covariance4 = covariance4 / size(train4,1)

% get test instances
test1 = imox(25:48,1:8);
test2 = imox(73:96,1:8);
test3 = imox(121:144,1:8);
test4 = imox(169:192,1:8);

% define confusion matrix
confusionMatrix = zeros(4);

% run discriminants for test instances

% firstly for test instances of class 1
for i =1:size(test1,1)
    discValues = zeros(4,1); % keeps discriminant results
    x = test1(i,:)';
    discValues(1) = (1/(2 * pi * det(covariance1))) * exp((-1/2) * ((x-mu1)'
* inv(covariance1) * (x-mu1)));
    discValues(2) = (1/(2 * pi * det(covariance2))) * exp((-1/2) * ((x-mu2)'
* inv(covariance2) * (x-mu2)));
    discValues(3) = (1/(2 * pi * det(covariance3))) * exp((-1/2) * ((x-mu3)'
* inv(covariance3) * (x-mu3)));
    discValues(4) = (1/(2 * pi * det(covariance4))) * exp((-1/2) * ((x-mu4)'
* inv(covariance4) * (x-mu4)));
    [Y,I] = max(discValues); % get max value
    confusionMatrix(1,I) = confusionMatrix(1,I) + 1; % increment confusion
matrix
end

% then for test instances of class 2
for i =1:size(test2,1)
    discValues = zeros(4,1); % keeps discriminant results
    x = test2(i,:)';
    discValues(1) = (1/(2 * pi * det(covariance1))) * exp((-1/2) * ((x-mu1)'
* inv(covariance1) * (x-mu1)));
    discValues(2) = (1/(2 * pi * det(covariance2))) * exp((-1/2) * ((x-mu2)'
* inv(covariance2) * (x-mu2)));
    discValues(3) = (1/(2 * pi * det(covariance3))) * exp((-1/2) * ((x-mu3)'
* inv(covariance3) * (x-mu3)));
    discValues(4) = (1/(2 * pi * det(covariance4))) * exp((-1/2) * ((x-mu4)'
* inv(covariance4) * (x-mu4)));
    [Y,I] = max(discValues); % get max value
    confusionMatrix(2,I) = confusionMatrix(2,I) + 1; % increment confusion
matrix
end

% then for test instances of class 3
for i =1:size(test3,1)
    discValues = zeros(4,1); % keeps discriminant results
```

```
    x = test3(i,:)';
    discValues(1) = (1/(2 * pi * det(covariance1))) * exp((-1/2) * ((x-mu1)'
* inv(covariance1) * (x-mu1)));
    discValues(2) = (1/(2 * pi * det(covariance2))) * exp((-1/2) * ((x-mu2)'
* inv(covariance2) * (x-mu2)));
    discValues(3) = (1/(2 * pi * det(covariance3))) * exp((-1/2) * ((x-mu3)'
* inv(covariance3) * (x-mu3)));
    discValues(4) = (1/(2 * pi * det(covariance4))) * exp((-1/2) * ((x-mu4)'
* inv(covariance4) * (x-mu4)));
    [Y,I] = max(discValues); % get max value
    confusionMatrix(3,I) = confusionMatrix(3,I) + 1; % increment confusion
matrix
end

% lastly for test instances of class 4
for i =1:size(test4,1)
    discValues = zeros(4,1); % keeps discriminant results
    x = test4(i,:)';
    discValues(1) = (1/(2 * pi * det(covariance1))) * exp((-1/2) * ((x-mu1)'
* inv(covariance1) * (x-mu1)));
    discValues(2) = (1/(2 * pi * det(covariance2))) * exp((-1/2) * ((x-mu2)'
* inv(covariance2) * (x-mu2)));
    discValues(3) = (1/(2 * pi * det(covariance3))) * exp((-1/2) * ((x-mu3)'
* inv(covariance3) * (x-mu3)));
    discValues(4) = (1/(2 * pi * det(covariance4))) * exp((-1/2) * ((x-mu4)'
* inv(covariance4) * (x-mu4)));
    [Y,I] = max(discValues); % get max value
    confusionMatrix(4,I) = confusionMatrix(4,I) + 1; % increment confusion
matrix
end

confusionMatrix
```