Please use python3 to execute the code.

In problem 1 we are give a processed data of handwritten digits 5 and 6. The data set has 64 features and I calculate the gaussian probability density using this data set .

For classification of test data here I use bayesian decision theory in which I calculate the discriminant function for both the class and applied it to the test data set. If g1 > g2, the the given input will be classified as class1 else in class2.

The covariance matrix from the training data is find out to be as :
Cov_5:

```
[[ 38.87878788  21.48484848  23.59343434 ...,  24.71212121  24.8510101
   25.71969697]
 [ 21.48484848  38.50568182  23.01262626 ...,  27.85248316  23.5625
   25.52504209]
 [ 23.59343434  23.01262626  44.67401286 ...,  23.97237527  19.34871442
   26.90159167]
 ...,
 [ 24.71212121  27.85248316  23.97237527 ...,  45.62679191  26.25258264
   24.49300454]
 [ 24.8510101   23.5625      19.34871442 ...,  26.25258264  39.28575528
   24.95299587]
 [ 25.71969697  25.52504209  26.90159167 ...,  24.49300454  24.95299587
   41.08393914]]
```

And cov_6:

```
[[ 42.92450452  23.42862752  25.62609792 ...,  20.85917705  24.01364003
   28.49383099]
 [ 23.42862752  40.29649837  25.98059396 ...,  25.11684268  26.76751331
   27.75090417]
 [ 25.62609792  25.98059396  42.79041891 ...,  22.37846942  20.9158865
   29.15812098]
 ...,
 [ 20.85917705  25.11684268  22.37846942 ...,  40.86152617  26.69629584
   22.87389175]
 [ 24.01364003  26.76751331  20.9158865  ...,  26.69629584  45.98686975
   26.5309484 ]
 [ 28.49383099  27.75090417  29.15812098 ...,  22.87389175  26.5309484
   42.04373075]]
```

For the performance analysis , I use the confusion matrix which is defined as M = [[true 5, false 6], [false 5 , true 6]] and classification percentage.

For various cases I analyse the classification problem as follows:

Case 1: using true cov_5 and cov_6:

Confusion matrix = [[106, 27], [49, 151]]
% classification of 5 and 6 resp.: 68.38709677419355   84.8314606741573

Case 2: using true cov_5 and cov_6 but neglecting the determinant term:
Confusion matrix : [[128 44],[27 134]]
% classification of 5 and 6 resp.: 82.58064516129032   75.28089887640449

Case 3 : using cov_5 = cov_6:
Confusion matrix : [[110 11],[45 167]]
% classification of 5 and 6 resp.: 70.96774193548387   93.82022471910112

Case 4 : using cov_6 = cov_5:
Confusion matrix : [[139 50],[16 128]]
% classification of 5 and 6 resp.: 89.6774193548387   71.91011235955057

Case 5 : using cov_5 = cov_6 = cov where cov = p_5*cov_5+p_6*cov_6:
Confusion matrix : [[134 27], [21 151]]
% classification of 5 and 6 resp.: 86.45161290322581   84.8314606741573

I have tested the code for the 5 cases on test data and above findings are obtained. And it is found that case 5 is the best possible combination for classification.