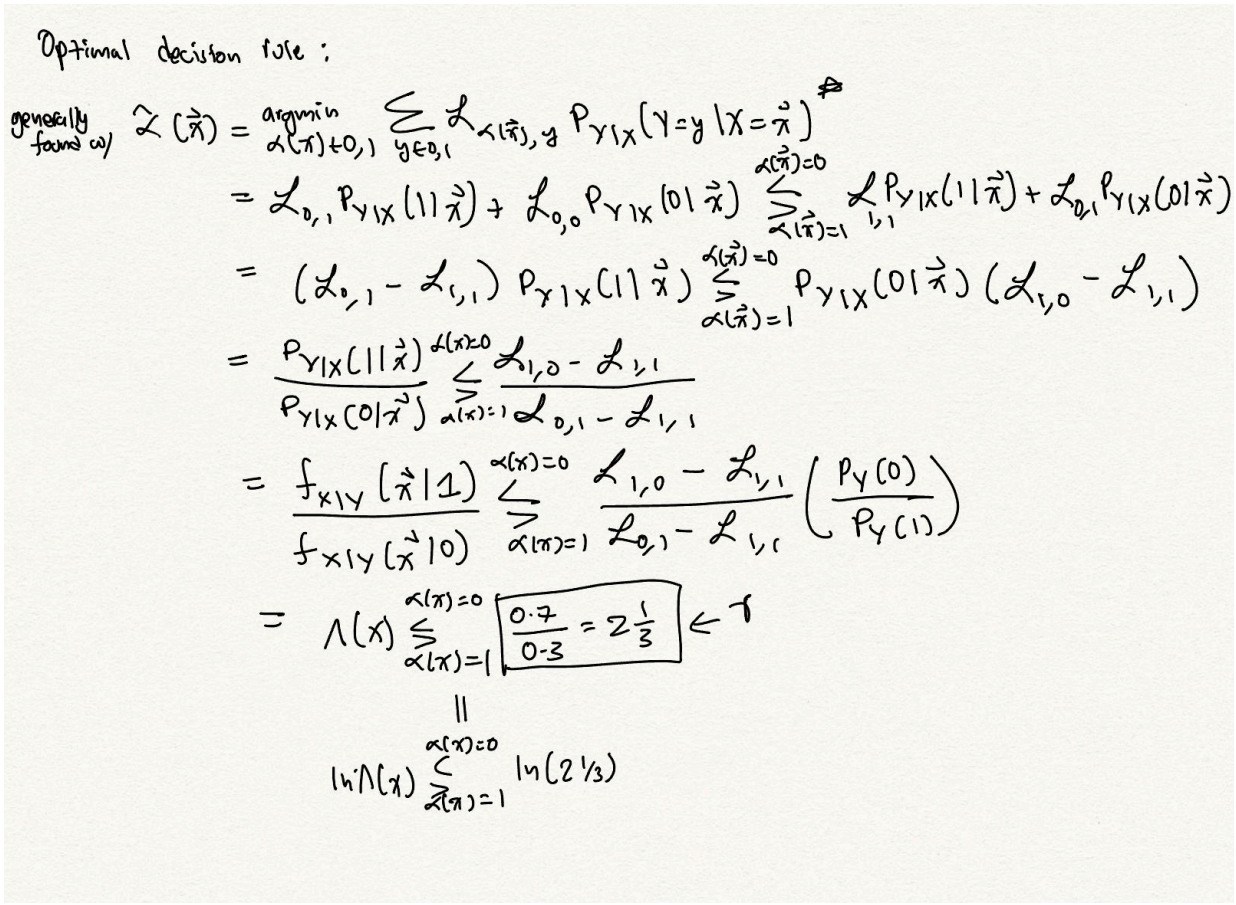
ML PS 1
Mohandeep Kapur

Problem 1.1

Part A:

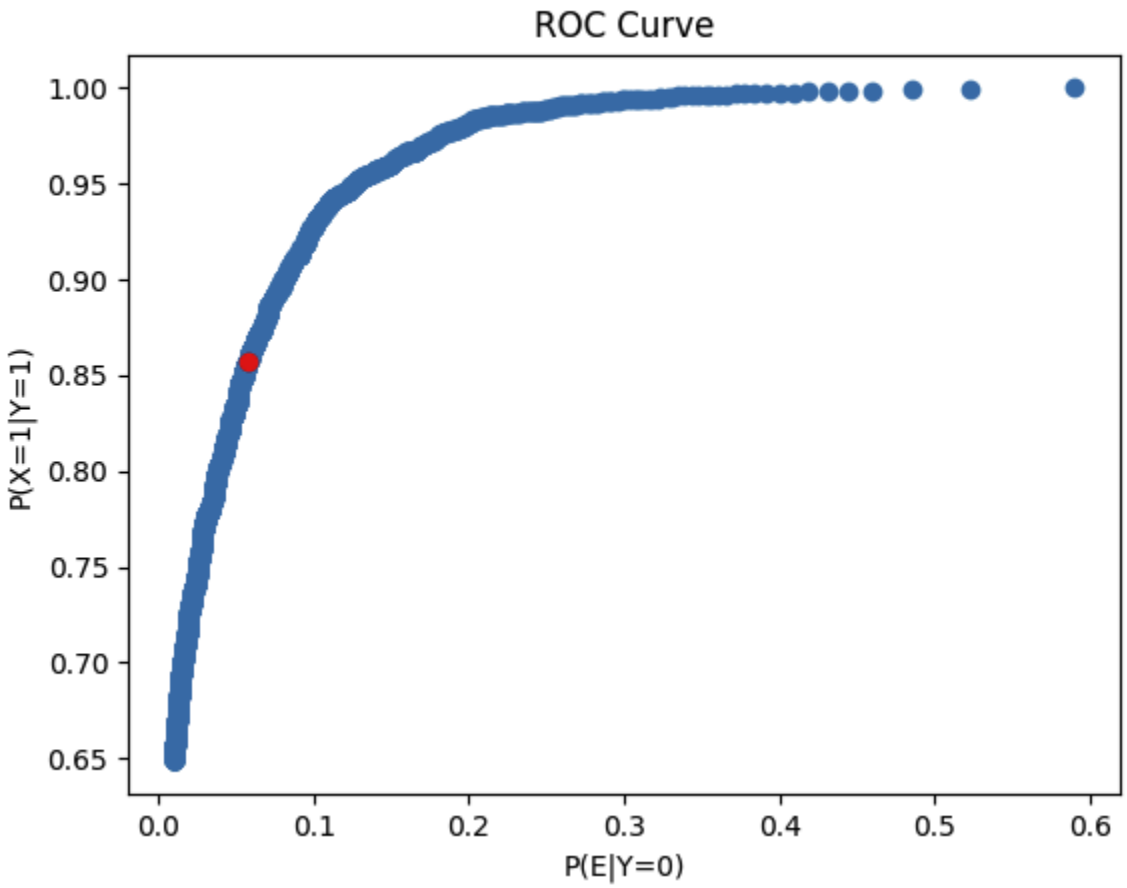
Q1

How optimal decision rule was found:



Q2
ROC Curve for experimental gammas from 0.01 to 10, incremented slightly. Each gamma made inferences for 10k samples.

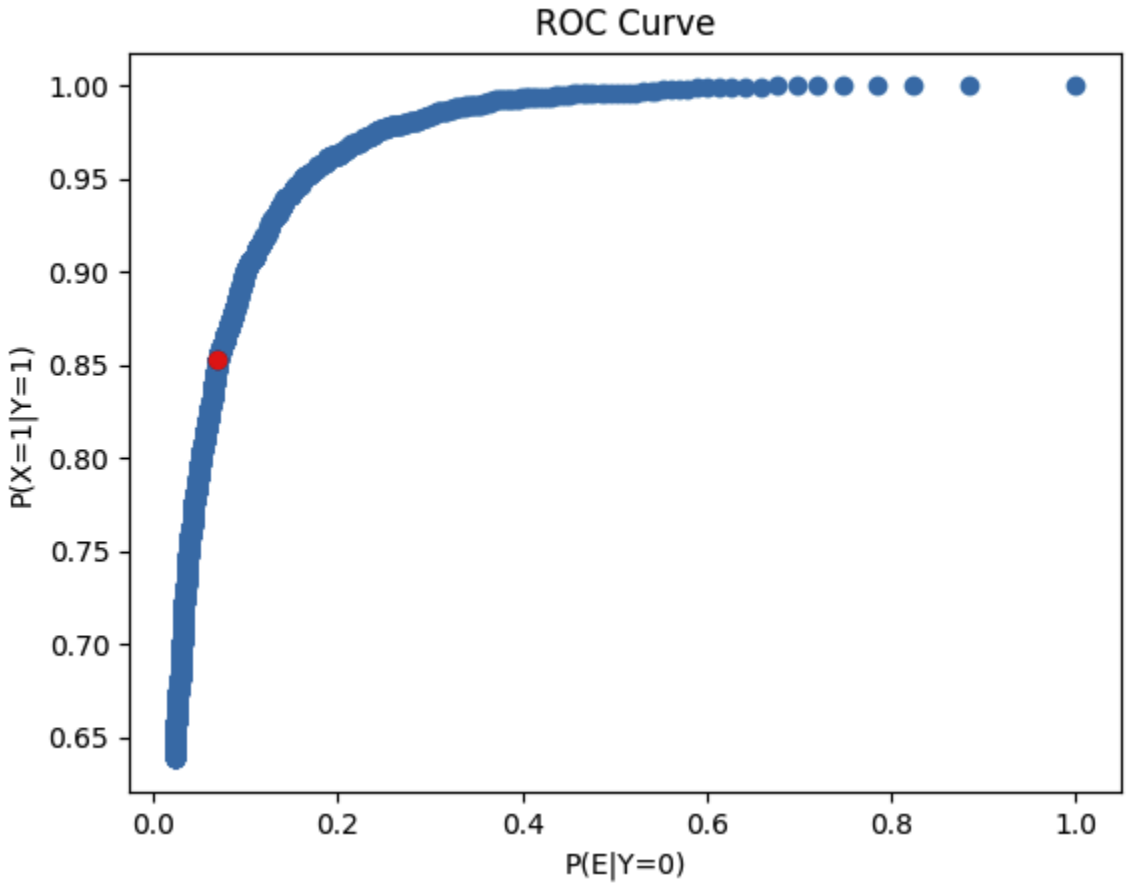
Q3



Threshold value that achieves minimum error rate: 2.327
True Positive Rate: 0.857
False Positive Rate: 0.057

Optimal experimental and theoretical thresholds are close. It makes sense that we'd receive the given experimental threshold, since it was proven theoretically that the optimal threshold depends only on the ratio of prior probabilities (for 0-1 loss). The # of hypothesis 0 and hypothesis 1 samples were generated randomly w/ respective prior probs, but their ratio always hovered close to 2.33 for 10k samples, as does the experimental threshold.

Part B:
Premise: samples come from class-conditional distributions that behave differently than what I've modeled them to behave like. This will probably lead my model to make more incorrect inferences and increase the minimum probability of error threshold.



Threshold value that achieves minimum error rate: 2.506
True Positive Rate: 0.853
False Positive Rate: 0.0696

Not much change, likely because dependence between gaussian terms in sampled class-conditional pdfs is weak, so the no-correlation assumed pdfs were not a bad approximation. Minimum achievable error did rise slightly, as expected.

Problem 1.2
Part A:
Q1
Done, class conditional distributions in code.

Q2

Decision rule utilized: MAP rule for multi-dimensional distributions and multiple hypotheses.

0-1 loss func, multiple hypotheses

$\hat{\alpha}(x) = \underset{i \in Y}{\text{argmax}}\ g_i(x)$ ← multi-dimensional MAP

where $g_i(x)$ (when every hypothesis has unique cov matrix) is:

$$g_i(x) = -\frac{1}{2}(\vec{x} - \vec{M}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{M}_i) - \frac{1}{2}\ln(\det \Sigma_i) + \ln P_r(i)$$

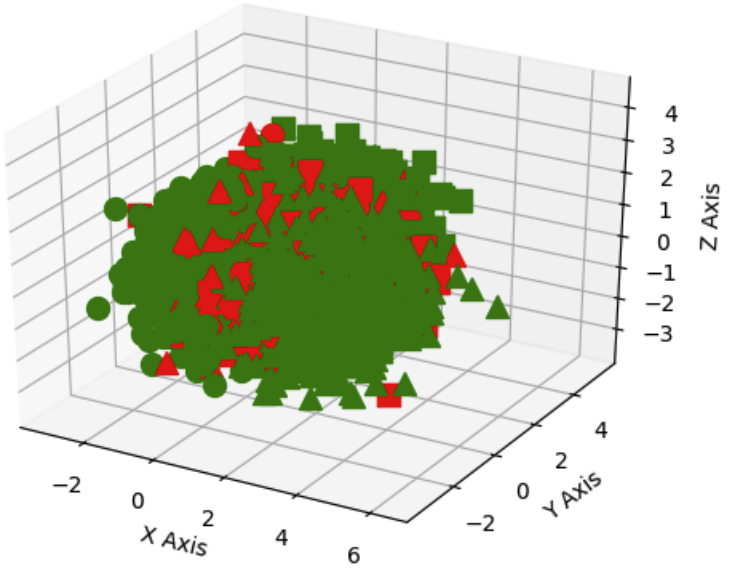
Confusion matrix with 0-1 Loss:

(row i, col j) corresponds to P(alpha = i+1, truth = j+1)

[[0.18406825 0.12323828 0.03100186]
[0.0266937 0.71779744 0.03547897]
[0.04291265 0.15896427 0.26685251]]

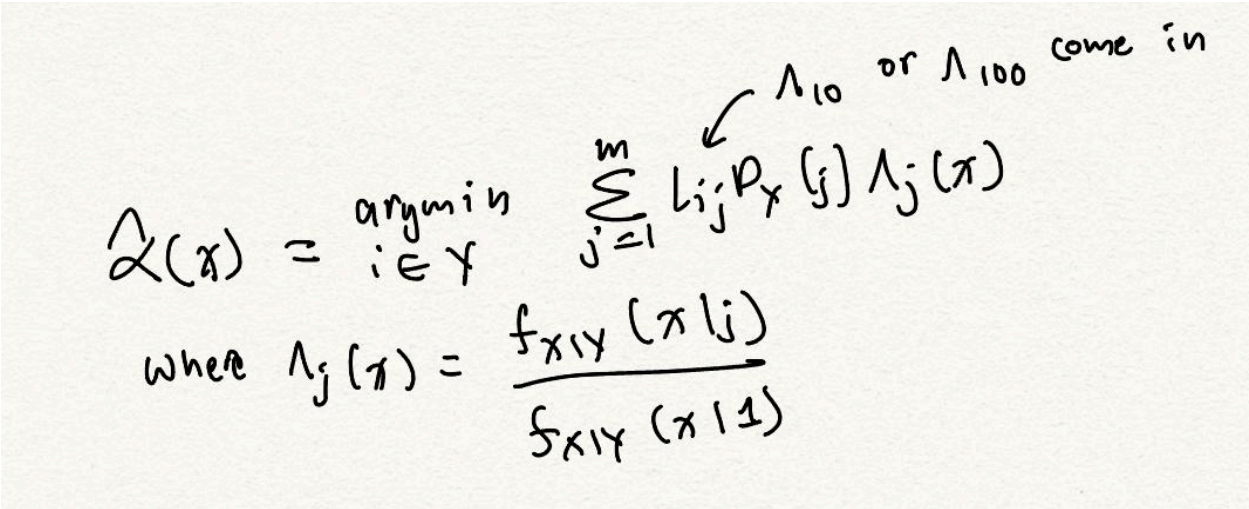
Q3

Visualization:



Part B

Decision rule is now more general / deals with any error-cost (correct-cost still must be 0):

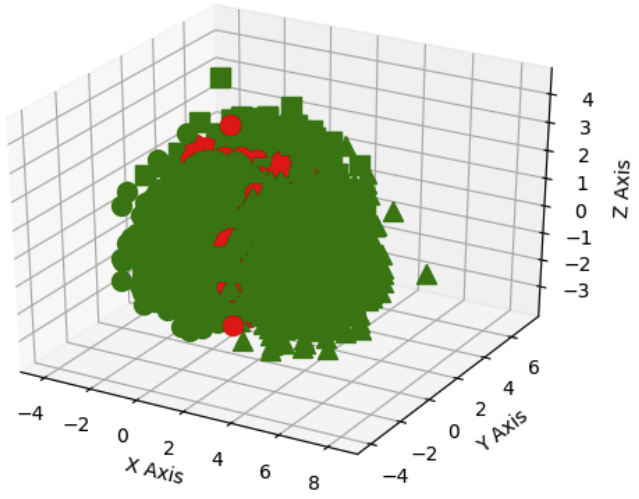


Confusion Matrix with Loss Matrix A10:

(row i, col j) corresponds to P(alpha = i+1, truth = j+1)

[[2.04167700e-01 1.80602007e-02 2.06731167e-03]
 [6.20193500e-03 9.69899666e-01 2.48077400e-04]
 [3.59712230e-02 1.20401338e-02 3.31017944e-01]]

Visualization:



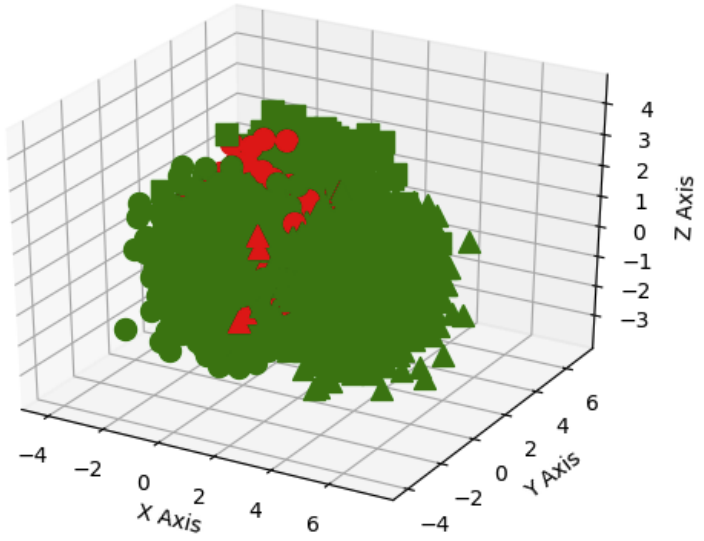
Confusion Matrix with A100:

(row i, col j) corresponds to P(alpha = i+1, truth = j+1)

[[1.61677659e-01 1.34558582e-02 1.69118891e-04]
 [4.14341282e-03 9.54053167e-01 0.00000000e+00]

[8.87874176e-02 3.24909747e-02 3.33164214e-01]]

Visualization:



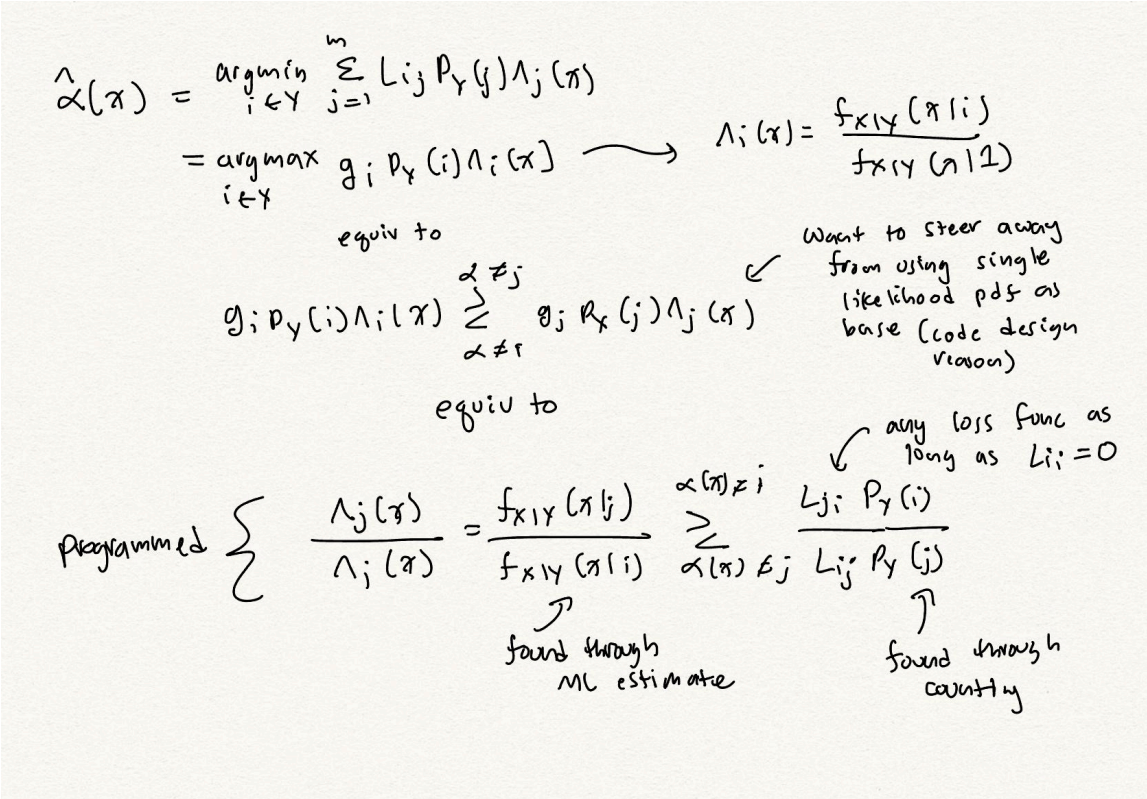
Interesting insights:
The new decision rule is much more accurate than in part A! Not sure why.
Both loss matrices make (1, 3) and (2, 3) decisions rarer, with A100 having the most impact. It also seems like the probability of 2 and 3 being chosen has decreased in general. This is likely because the expected loss term when deciding either 1 or 2 has increased thanks to the loss matrices, while 3's risk is unchanged.

Problem 1.3

I developed a SimpleClassifier class that works on any standardized labeled-feature dataset. I utilized it to make inferences on both the wine-quality and human-activity datasets.

SimpleClassifier assumes that all class-specific distributions can be represented by a gaussian distribution. It also assumes that the loss for a correct decision must be 0.

With those two assumptions, I was able to derive the following decision process:



The expressed decision rule requires two vital pieces of information: the prior probabilities and likelihoods of each class.

Finding the prior probabilities simplifies to a counting problem, the ratio of the frequency of a class divided by the total number of labeled-samples.

The class-conditional pdfs were found through ML estimation. ML estimation intuitively wants to find parameters of a chosen distribution (gaussian in our case) that maximize the likelihood of reproducing given data. I used numpy's mean and covariance functions to find those gaussian-relevant parameters for the unique sequence of data collected for every class. Those parameters derived from given data-points naturally maximize the likelihood of reproducing said data-points (ML estimation performed).

Now, with both prior and likelihood probabilities found for every class, inference on labeled-features was simply a matter of executing the above decision rule in code.

Results:
Confusion Matrix for wine-quality dataset:
(row i, col j) corresponds to P(alpha = i, truth = j)

```
results:
 [[0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.3333 0.0324 0.0065 0.0049 0.0019 0.     0.    ]
 [0.     0.     0.     0.0667 0.088  0.0201 0.0092 0.0028 0.0052 0.    ]
 [0.     0.     0.     0.3333 0.463  0.5374 0.2091 0.0306 0.0155 0.    ]
 [0.     0.     0.     0.1667 0.3333 0.3616 0.4799 0.3012 0.2487 0.2   ]
 [0.     0.     0.     0.0667 0.0694 0.0688 0.2846 0.6423 0.6425 0.8   ]
 [0.     0.     0.     0.0333 0.0093 0.0009 0.0099 0.0213 0.0881 0.    ]
 [0.     0.     0.     0.     0.0046 0.0047 0.0025 0.     0.     0.    ]]
p of error: 0.4999230414037272
```

P of error: 0.499

The high probability of error suggests that assuming that the true likelihood pdfs could be approximated with a gaussian distribution was an incorrect hypothesis. If the data-generated pdf does not approximate the underlying, actual pdf well, your decision rule is based on flawed information and thus can never be accurate.

Confusion Matrix for human-activity x and y_train dataset:
(row i, col j) corresponds to P(alpha = i, truth = j)

```
all features + labels in X_train y_train
results:
 [[0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.9715 0.0354 0.0923 0.     0.     0.    ]
 [0.     0.0261 0.9627 0.1298 0.0008 0.     0.0007]
 [0.     0.0024 0.0019 0.7779 0.     0.     0.    ]
 [0.     0.     0.     0.     0.6633 0.0146 0.    ]
 [0.     0.     0.     0.     0.3305 0.9854 0.    ]
 [0.     0.     0.     0.     0.0054 0.     0.9993]]
p of error: 0.061479869423285836
```

P of error: 0.061

Here, it is evident that choosing gaussian distributions to represent class-conditional pdfs was a good idea, supported by the low probability of error.