

Assignment 2 – SYDE 572

Calvin Tran

Exercise 1

1.

The two classes were generated as follows:

Class 1:

```
[ 5.45681053 12.08163614]
[ 5.41296766 10.10879192]
[ 2.49460888 13.00181466]
[ 2.55850519 11.26749748]
[ 2.20272472  5.14248427]
```

Class 2:

```
[25.31073326 41.94952444]
[18.81649456 28.2918927 ]
[20.68816933 24.29206135]
[23.83854579 34.89750766]
[17.91239753 27.94280608]
```

We can calculate the sample mean for each class:

$$\mu_1 = \begin{bmatrix} \frac{5.45681053 + 5.41296766 + 2.49460888 + 2.55850519 + 2.20272472}{5} \\ \frac{12.08163614 + 10.10879192 + 13.00181466 + 11.26749748 + 5.14248427}{5} \end{bmatrix} = \begin{bmatrix} 3.63 \\ 10.32 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} \frac{25.31073326 + 18.81649456 + 20.68816933 + 23.83854579 + 17.91239753}{5} \\ \frac{41.94952444 + 28.2918927 + 24.29206135 + 34.89750766 + 27.94280608}{5} \end{bmatrix} = \begin{bmatrix} 21.31 \\ 31.47 \end{bmatrix}$$

We can calculate the covariance matrix for each class. We will assume that the sample mean will be used to calculate the covariance matrices. Starting with class 1:

$$\Sigma_1 = \begin{bmatrix} \text{Var}(x_1) & \text{Cov}(x_1, x_2) \\ \text{Cov}(x_2, x_1) & \text{Var}(x_2) \end{bmatrix}$$

$$\begin{aligned} & Var(x_1) \\ &= \frac{(5.46 - 3.63)^2 + (5.41 - 3.63)^2 + (2.49 - 3.63)^2 + (2.56 - 3.63)^2 + (2.20 - 3.63)^2}{5 - 1} \\ &= 2.75 \end{aligned}$$

$$\begin{aligned} & Var(x_2) \\ &= \frac{(12.08 - 10.32)^2 + (10.11 - 10.32)^2 + (13.00 - 10.32)^2 + (11.27 - 10.32)^2 + (5.14 - 10.32)^2}{5 - 1} \\ &= 9.51 \end{aligned}$$

$$\begin{aligned} & Cov(x_1, x_2) = Cov(x_2, x_1) \\ &= \frac{(5.46 - 3.63)(12.08 - 10.32) + (5.41 - 3.63)(10.11 - 10.32) + (2.49 - 3.63)(13.00 - 10.32) + (2.56 - 3.63)(11.27 - 10.32) + (2.20 - 3.63)(5.14 - 10.32)}{5 - 1} \\ &= 1.54 \end{aligned}$$

$$\Sigma_1 = \begin{bmatrix} 2.75 & 1.54 \\ 1.54 & 9.51 \end{bmatrix}$$

For class 2:

$$\left[\begin{array}{c} \frac{25.31073326 + 18.81649456 + 20.68816933 + 23.83854579 + 17.91239753}{5} \\ \frac{41.94952444 + 28.2918927 + 24.29206135 + 34.89750766 + 27.94280608}{5} \end{array} \right]$$

$$\Sigma_2 = \begin{bmatrix} Var(x_1) & Cov(x_1, x_2) \\ Cov(x_2, x_1) & Var(x_2) \end{bmatrix}$$

$$\begin{aligned} & Var(x_1) \\ &= \frac{(25.31 - 21.31)^2 + (18.82 - 21.31)^2 + (20.69 - 21.31)^2 + (23.84 - 21.31)^2 + (17.91 - 21.31)^2}{5 - 1} \\ &= 10.14 \end{aligned}$$

$$\begin{aligned} & Var(x_2) \\ &= \frac{(41.95 - 31.47)^2 + (28.29 - 31.47)^2 + (24.29 - 31.47)^2 + (34.90 - 31.47)^2 + (27.94 - 31.47)^2}{5 - 1} \\ &= 48.91 \end{aligned}$$

$$\begin{aligned} & Cov(x_1, x_2) = Cov(x_2, x_1) \\ &= \frac{(25.31 - 21.31)(41.95 - 31.47) + (18.82 - 21.31)(28.29 - 31.47) + (20.69 - 21.31)(24.29 - 31.47) + (23.84 - 21.31)(34.90 - 31.47) + (17.91 - 21.31)(27.94 - 31.47)}{5 - 1} \\ &= 18.74 \end{aligned}$$

$$\Sigma_2 = \begin{bmatrix} 10.13 & 18.74 \\ 18.74 & 48.91 \end{bmatrix}$$

2.

We first determine the eigenvalues and eigenvectors of class 1:

$$\Sigma_1 = \begin{bmatrix} 2.75 & 1.54 \\ 1.54 & 9.51 \end{bmatrix}$$

$$\begin{aligned} \det(\lambda I - A) &= 0 \\ \det\left(\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 2.75 & 1.54 \\ 1.54 & 9.51 \end{bmatrix}\right) &= 0 \\ \det \begin{bmatrix} \lambda - 2.75 & -1.54 \\ -1.54 & \lambda - 9.51 \end{bmatrix} &= 0 \\ \lambda^2 - 12.26\lambda + 23.78 &= 0 \end{aligned}$$

$$\lambda_1 = 9.84, \lambda_2 = 2.42$$

For $\lambda_1 = 9.84$:

$$\left(9.84 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 2.75 & 1.54 \\ 1.54 & 9.51 \end{bmatrix}\right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 7.09 & -1.54 \\ -1.54 & 0.33 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 7.09 & -1.54 & | & 0 \\ -1.54 & 0.33 & | & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -0.22 & | & 0 \\ 0 & 0 & | & 0 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 0.22 \\ 1 \end{bmatrix}$$

For $\lambda_2 = 2.42$:

$$\left(2.42 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 2.75 & 1.54 \\ 1.54 & 9.51 \end{bmatrix}\right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -0.33 & -1.54 \\ -1.54 & -7.09 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -0.33 & -1.54 & | & 0 \\ -1.54 & -7.09 & | & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4.67 & | & 0 \\ 0 & 0 & | & 0 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} -4.67 \\ 1 \end{bmatrix}$$

Thus, for class 1, the eigenvalues are 9.84 and 2.42. Their respective eigenvectors are $\begin{bmatrix} 0.22 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} -4.67 \\ 1 \end{bmatrix}$.

We do the same for class 2:

$$\Sigma_2 = \begin{bmatrix} 10.13 & 18.74 \\ 18.74 & 48.91 \end{bmatrix}$$

$$\begin{aligned} \det(\lambda I - A) &= 0 \\ \det\left(\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 10.13 & 18.74 \\ 18.74 & 48.91 \end{bmatrix}\right) &= 0 \\ \det\begin{bmatrix} \lambda - 10.13 & -18.74 \\ -18.74 & \lambda - 48.91 \end{bmatrix} &= 0 \\ \lambda^2 - 59.04\lambda + 144.27 &= 0 \end{aligned}$$

$$\lambda_1 = 56.49, \lambda_2 = 2.55$$

For $\lambda_1 = 56.49$:

$$\left(56.49 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 10.13 & 18.74 \\ 18.74 & 48.91 \end{bmatrix}\right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 46.36 & -18.74 \\ -18.74 & 7.58 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 46.36 & -18.74 & | & 0 \\ -18.74 & 7.58 & | & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -0.404 & | & 0 \\ 0 & 0 & | & 0 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 0.404 \\ 1 \end{bmatrix}$$

For $\lambda_2 = 2.55$:

$$\left(2.55 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 10.13 & 18.74 \\ 18.74 & 48.91 \end{bmatrix}\right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -7.58 & -18.74 \\ -18.74 & -46.36 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -7.58 & -18.74 & | & 0 \\ -18.74 & -46.36 & | & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2.47 & | & 0 \\ 0 & 0 & | & 0 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} -2.47 \\ 1 \end{bmatrix}$$

The eigenvalues for class 2 are 56.49 and 2.55. Their respective eigenvectors are $\begin{bmatrix} 0.404 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} -2.47 \\ 1 \end{bmatrix}$.

3.

We first need to find the inverse of the covariance matrices for each class. We currently have:

$$S_1 = \begin{bmatrix} 2.75 & 1.54 \\ 1.54 & 9.51 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 10.13 & 18.74 \\ 18.74 & 48.91 \end{bmatrix}$$

To find the inverse, we can use the following formula for a 2x2 matrix:

$$S^{-1} = \phi^T \Lambda^{-1} \phi$$

For class 1:

$$\begin{aligned} \Lambda^{-1} &= \begin{bmatrix} \frac{1}{9.846} & 0 \\ 0 & \frac{1}{2.412} \end{bmatrix} \\ &= \begin{bmatrix} 0.1016 & 0 \\ 0 & 0.4146 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \phi &= \begin{bmatrix} 0.2149 & -0.9778 \\ 0.9767 & 0.2094 \end{bmatrix} \\ \phi^T &= \begin{bmatrix} 0.2149 & 0.9767 \\ -0.9778 & 0.2094 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} S_1^{-1} &= \begin{bmatrix} 0.2149 & 0.9767 \\ -0.9778 & 0.2094 \end{bmatrix} \begin{bmatrix} 0.1016 & 0 \\ 0 & 0.4146 \end{bmatrix} \begin{bmatrix} 0.2149 & -0.9778 \\ 0.9767 & 0.2094 \end{bmatrix} \\ &= \begin{bmatrix} 0.3999 & -0.06476 \\ -0.06476 & 0.1156 \end{bmatrix} \end{aligned}$$

For class 2:

$$\begin{aligned}
\Lambda^{-1} &= \begin{bmatrix} \frac{1}{56.49} & 0 \\ 0 & \frac{1}{2.55} \end{bmatrix} \\
&= \begin{bmatrix} 0.01770 & 0 \\ 0 & 0.3922 \end{bmatrix} \\
\phi &= \begin{bmatrix} 0.3746 & -0.9272 \\ 0.9272 & 0.3746 \end{bmatrix} \\
\phi^T &= \begin{bmatrix} 0.3746 & 0.9272 \\ -0.9272 & 0.3746 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
S_2^{-1} &= \begin{bmatrix} 0.3746 & 0.9272 \\ -0.9272 & 0.3746 \end{bmatrix} \begin{bmatrix} 0.01770 & 0 \\ 0 & 0.3922 \end{bmatrix} \begin{bmatrix} 0.3746 & -0.9272 \\ 0.9272 & 0.3746 \end{bmatrix} \\
&= \begin{bmatrix} 0.3390 & -0.1299 \\ -0.1299 & 0.07022 \end{bmatrix}
\end{aligned}$$

We also know:

$$\mu_1 = \begin{bmatrix} 3.63 \\ 10.32 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 21.31 \\ 31.47 \end{bmatrix}$$

We can then find the decision boundary.

$$\begin{aligned}
Q_0 &= S_1^{-1} - S_2^{-1} \\
&= \begin{bmatrix} 0.3999 & -0.06476 \\ -0.06476 & 0.1156 \end{bmatrix} - \begin{bmatrix} 0.3390 & -0.1299 \\ -0.1299 & 0.07022 \end{bmatrix} \\
&= \begin{bmatrix} 0.0609 & 0.06514 \\ 0.06514 & 0.04538 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
Q_1 &= 2[\mu_2^T S_2^{-1} - \mu_1^T S_1^{-1}] \\
&= 2 \left[\begin{bmatrix} 21.31 & 31.47 \end{bmatrix} \begin{bmatrix} 0.3390 & -0.1299 \\ -0.1299 & 0.07022 \end{bmatrix} - \begin{bmatrix} 3.63 & 10.32 \end{bmatrix} \begin{bmatrix} 0.3999 & -0.06476 \\ -0.06476 & 0.1156 \end{bmatrix} \right] \\
&= 2 \left[\begin{bmatrix} 3.136 & -0.5583 \end{bmatrix} - \begin{bmatrix} 0.7833 & 0.9579 \end{bmatrix} \right] \\
&= 2 \begin{bmatrix} 2.3527 & -1.5162 \end{bmatrix} \\
&= \begin{bmatrix} 4.705 & -3.032 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
Q_2 &= \mu_1^T S_1^{-1} \mu_1 - \mu_2^T S_2^{-1} \mu_2 \\
&= \begin{bmatrix} 3.63 & 10.32 \end{bmatrix} \begin{bmatrix} 0.3999 & -0.06476 \\ -0.06476 & 0.1156 \end{bmatrix} \begin{bmatrix} 3.63 \\ 10.32 \end{bmatrix} \\
&\quad - \begin{bmatrix} 21.31 & 31.47 \end{bmatrix} \begin{bmatrix} 0.3390 & -0.1299 \\ -0.1299 & 0.07022 \end{bmatrix} \begin{bmatrix} 21.31 \\ 31.47 \end{bmatrix} \\
&= 12.73 - 49.27 \\
&= -36.54
\end{aligned}$$

$$x^T Q_0 x + Q_1 x + Q_2 = 0$$

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 0.0609 & 0.06514 \\ 0.06514 & 0.04538 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 4.705 & -3.032 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 36.54 = 0$$

$$\begin{bmatrix} 0.0609x_1 + 0.06514x_2 & 0.06514x_1 + 0.04538x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 4.705x_1 - 3.032x_2 - 36.54 = 0$$

$$0.0609x_1^2 + 0.06514x_1x_2 + 0.06514x_1x_2 + 0.04538x_2^2 + 4.705x_1 - 3.032x_2 - 36.54 = 0$$

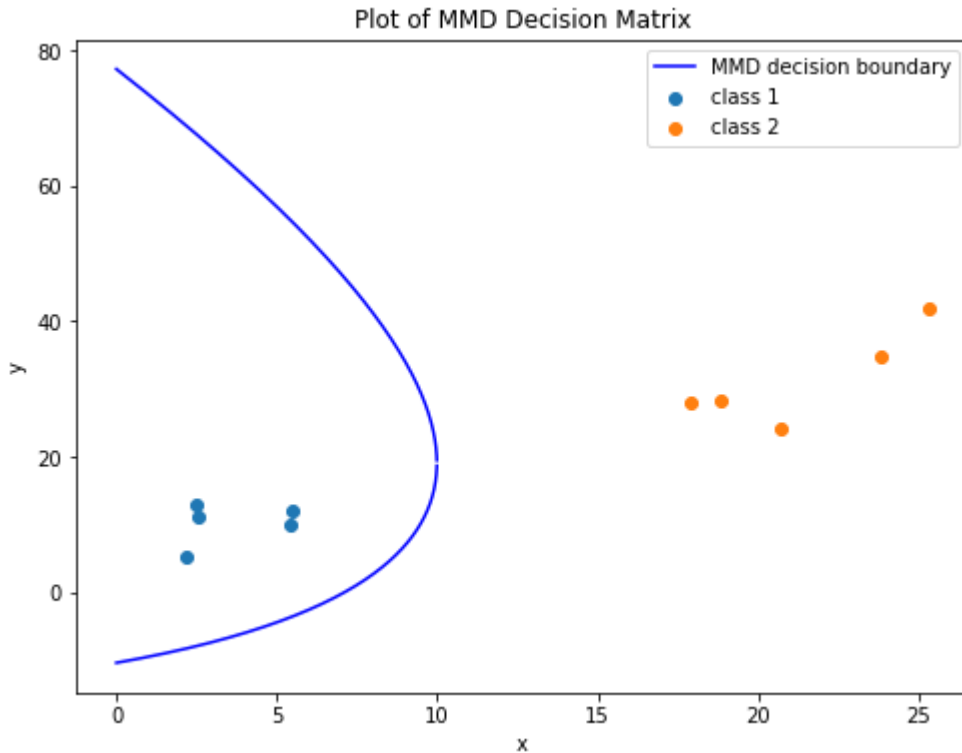
$$0.0609x_1^2 + 0.04538x_2^2 + 0.13028x_1x_2 + 4.705x_1 - 3.032x_2 - 36.54 = 0$$

$$0.04538x_2^2 + (0.13028x_1 - 3.032)x_2 + (0.0609x_1^2 + 4.705x_1 - 36.54) = 0$$

Solving for x_2 in the above equation using the quadratic formula gives:

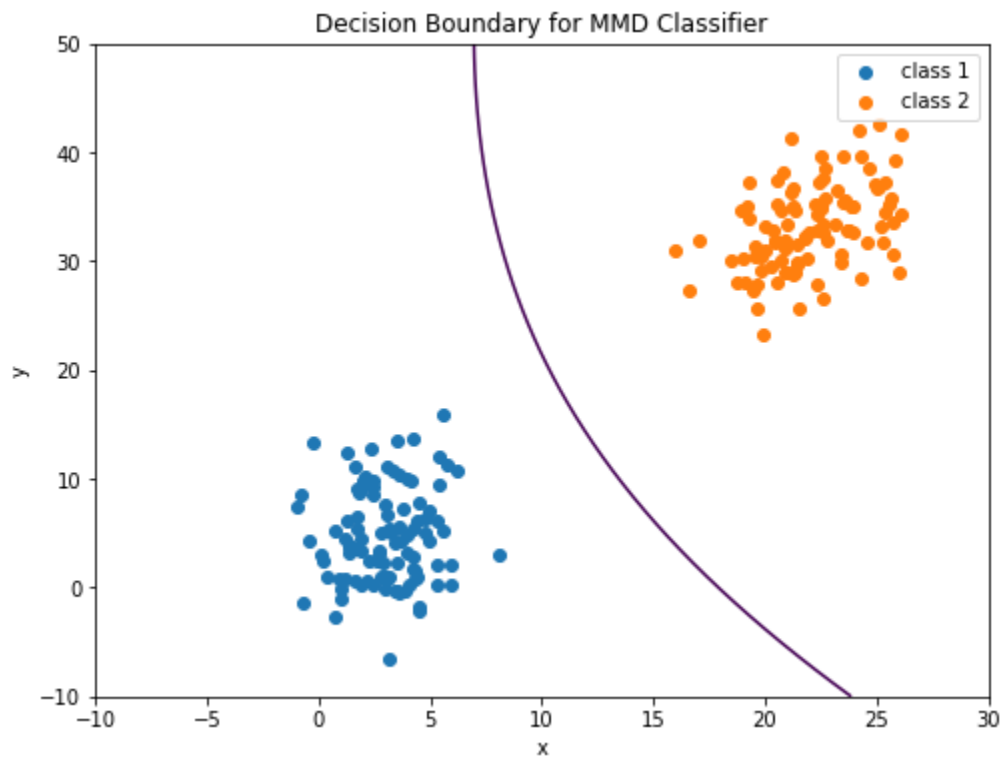
$$x_2 = \frac{-(0.13028x_1 - 3.032) \pm \sqrt{0.005918x_1^2 - 1.644x_1 + 15.83}}{0.09076}$$

We can then generate the following plot for the decision boundary using Python's matplotlib library.



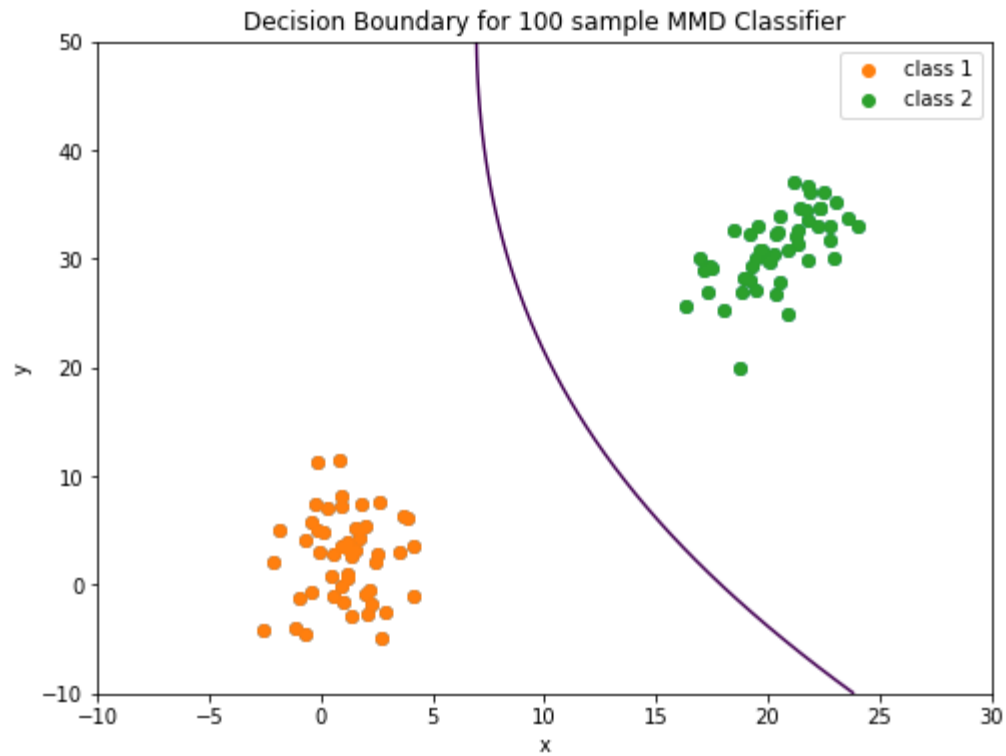
4.

Using Python, we can create a function that determines the MMD, given a point (x, y) . As seen in the code for this part, we have a function called `MMD`, which is applied to all points in an XY meshgrid. Then, we plot the contour for when $Z = 0$, which gives the following decision boundary and plot:

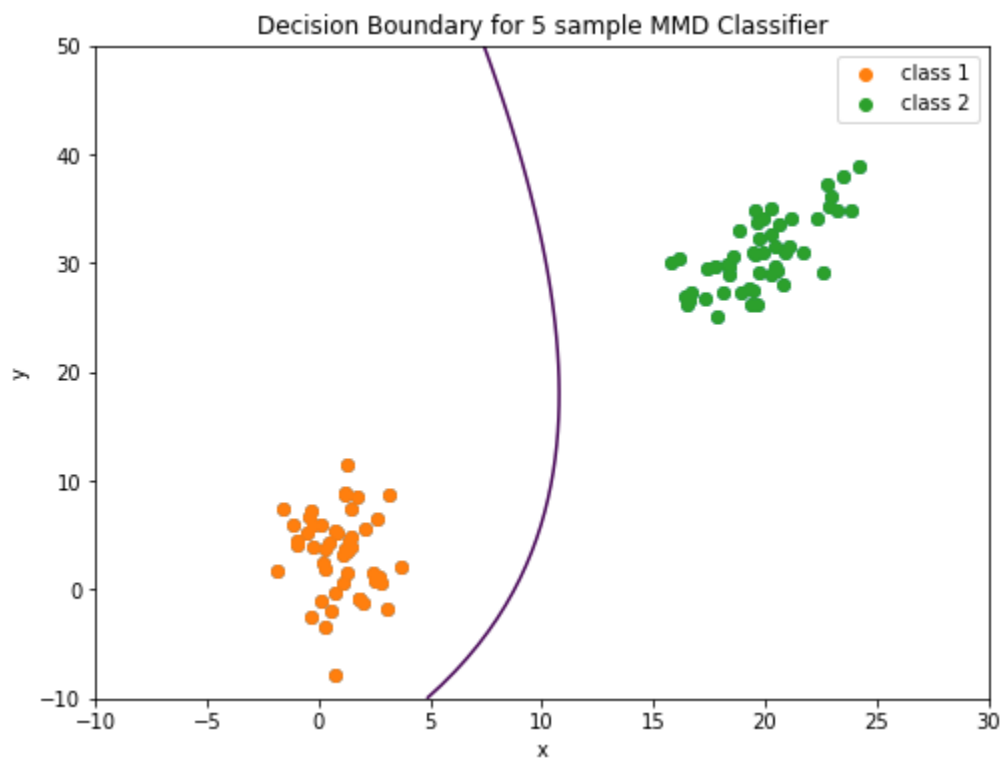


5.

The code in Part 5 of Exercise 1 in `Assignment_2.ipynb` is used to determine the accuracy of both MMDs. The MMD trained on 100 samples for each class had an accuracy of 100% and the resulting plot looks like so:



The MMD trained on 5 samples for each class also had an accuracy of 100% and the resulting plot looks like so:



6.

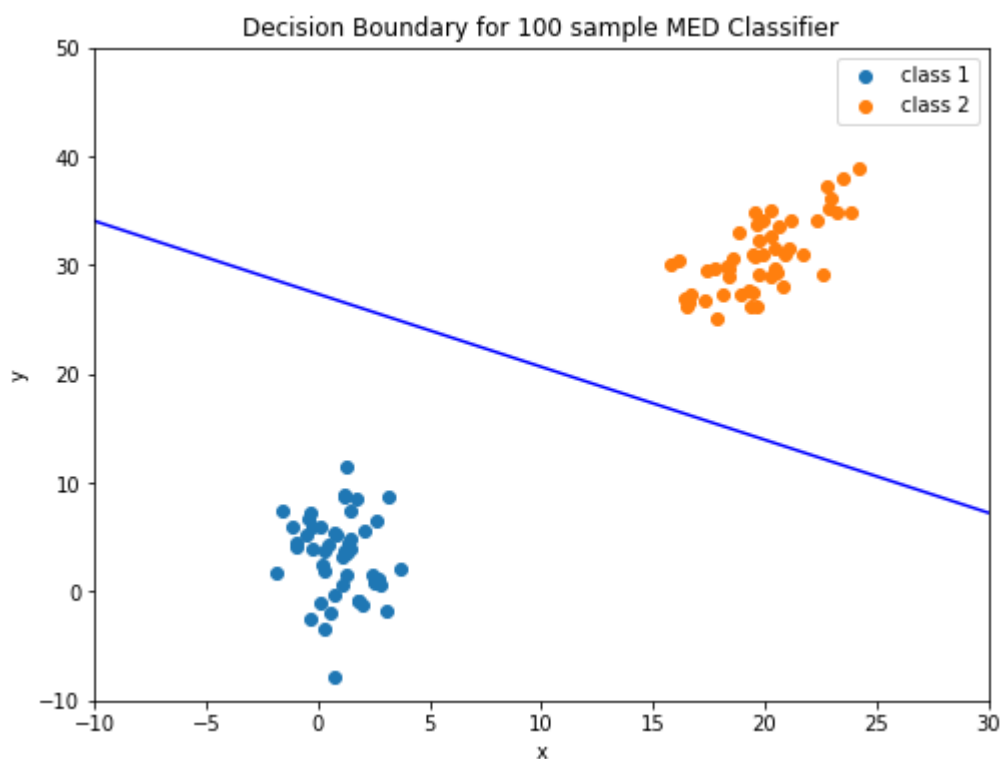
Both classifiers had the same accuracy of 100% when predicting on the new 100 samples. However, upon inspection of the decision boundaries for each classifier, we see that the average distance between the decision boundary trained on 5 samples and the test points is smaller than that of the decision boundary trained on 100 samples and the test points. Because of this, it becomes clear that decision boundary trained on 100 samples per class is better.

7.

The 100-sample training dataset was used to train the MED classifier from Assignment 1. The code to do this can be found in the Jupyter notebook. The following linear equation was found as a result of the classifier:

$$y = -0.67x + 27.35$$

This classifier also had a 100% accuracy on the same test data. The following plot was created for the MED classifier:



While the MED classifier is simpler to implement and also provides an accuracy of 100% in this case, we see that the MED classifier does not take into account the statistical distributions of each class when determining the classification, as it compares Euclidean distance, instead of Mahalanobis distance. This makes the MMD classification more accurate, especially when the points in each class overlap.

Exercise 2:

1.

To estimate the probability distributions of the two classes, we can use the following formula:

$$P(x|c_k) = \frac{e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}}$$

For class 1:

$$\mu_1 = \begin{bmatrix} 2.983 \\ 4.748 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 3.032 & 0.4919 \\ 0.4919 & 19.30 \end{bmatrix}$$

$$\Sigma_1^{-1} = \begin{bmatrix} 0.33122 & -0.008443 \\ -0.008443 & 0.05204 \end{bmatrix}$$

$$n = 2$$

Then, we can calculate the PDF of class 1:

$$P(x | c_1) = \frac{\exp\left(-\frac{1}{2} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 2.983 \\ 4.748 \end{bmatrix}\right)^T \begin{bmatrix} 0.33122 & -0.008443 \\ -0.008443 & 0.05204 \end{bmatrix} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 2.983 \\ 4.748 \end{bmatrix}\right)\right)}{(2\pi)^{\frac{1}{2}} \begin{vmatrix} 3.032 & 0.4919 \\ 0.4919 & 19.30 \end{vmatrix}^{\frac{1}{2}}}$$

Similarly, for class 2:

$$\mu_2 = \begin{bmatrix} 22.04 \\ 33.15 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 5.135 & 3.876 \\ 3.876 & 15.29 \end{bmatrix}$$

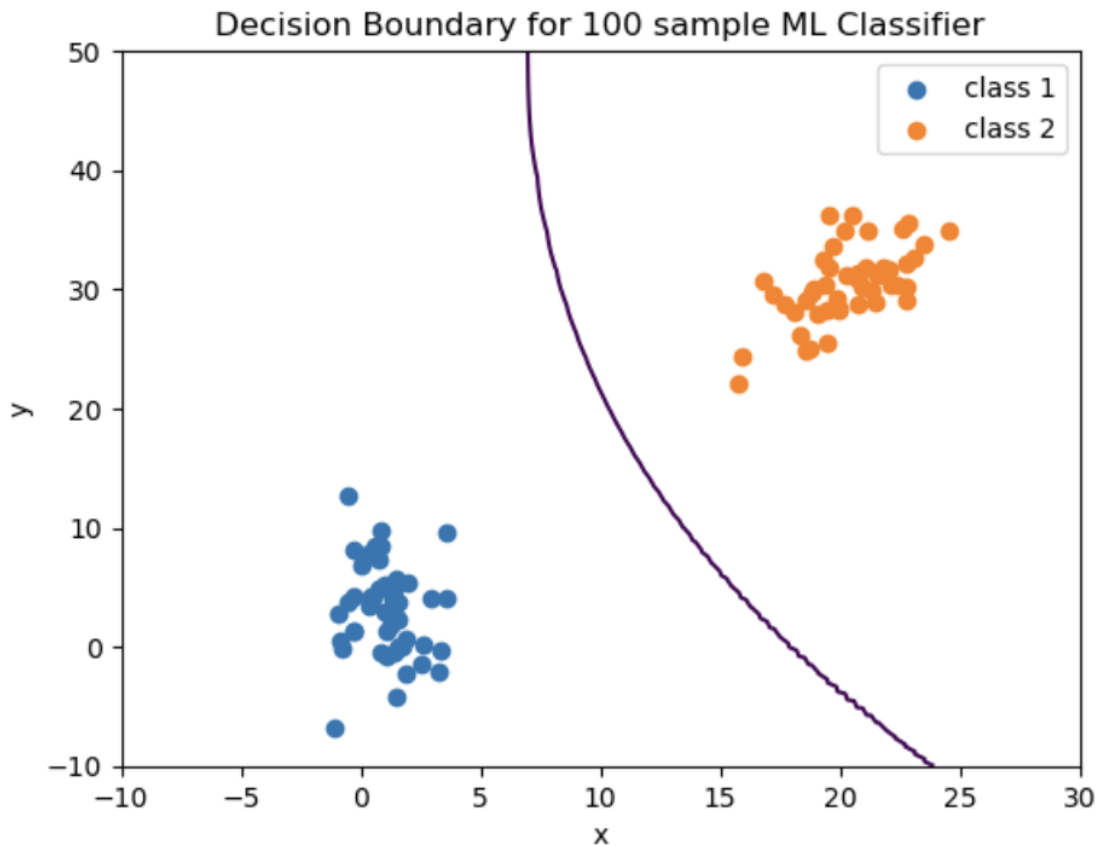
$$\Sigma_2^{-1} = \begin{bmatrix} 0.240814 & -0.06105 \\ -0.06105 & 0.08087 \end{bmatrix}$$

$$n = 2$$

Then, we can calculate the PDF of class 2:

$$P(x | c_2) = \frac{\exp\left(-\frac{1}{2}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 22.04 \\ 33.15 \end{bmatrix}\right)^T \begin{bmatrix} 0.240814 & -0.06105 \\ -0.06105 & 0.08087 \end{bmatrix} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 22.04 \\ 33.15 \end{bmatrix}\right)\right)}{(2\pi)^{\frac{1}{2}} \begin{vmatrix} 5.135 & 3.876 \\ 3.876 & 15.29 \end{vmatrix}^{\frac{1}{2}}}$$

Both of these PDFs were created in Python as a function in the code for Exercise 2. The classifier was created as a function called ML_classify. Upon testing on the 100 test points created earlier, we see that the accuracy is also 100%. The following plot was made to show the decision boundary created by the ML classifier:



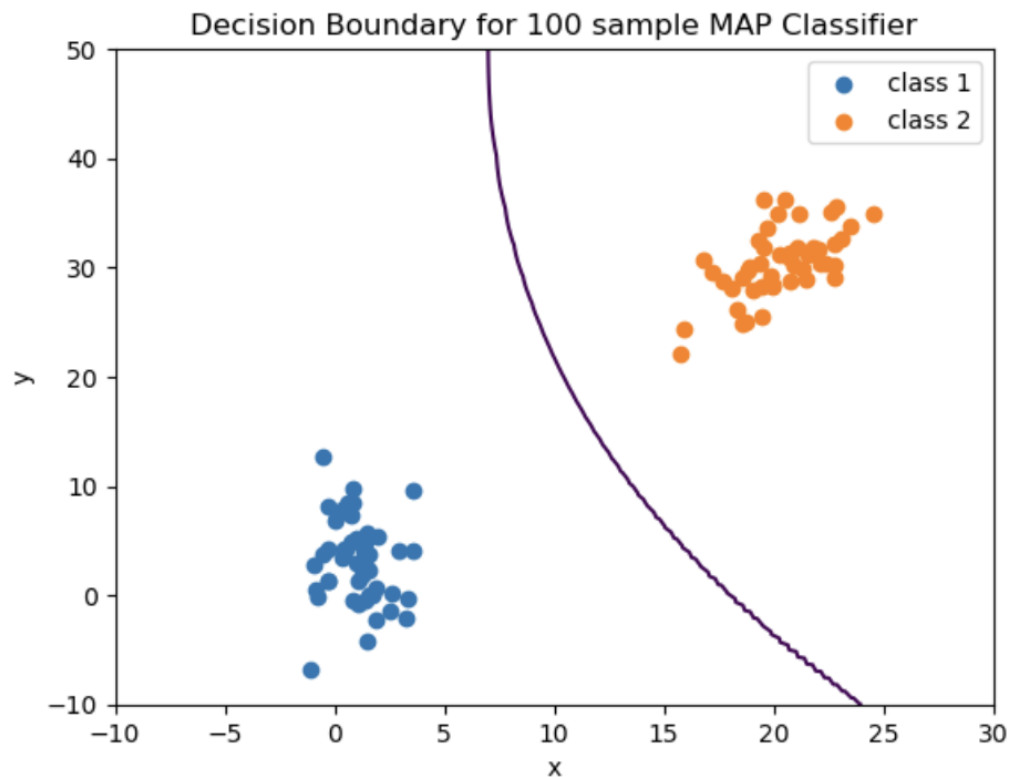
2.

To develop a MAP classifier, we multiply the PDFs for each class by that class's prior probability. Then, we get:

$$P(x | c_1) = 0.58 * \frac{\exp\left(-\frac{1}{2}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 2.983 \\ 4.748 \end{bmatrix}\right)^T \begin{bmatrix} 0.33122 & -0.008443 \\ -0.008443 & 0.05204 \end{bmatrix} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 2.983 \\ 4.748 \end{bmatrix}\right)\right)}{(2\pi)^{\frac{1}{2}} \begin{vmatrix} 3.032 & 0.4919 \\ 0.4919 & 19.30 \end{vmatrix}^{\frac{1}{2}}}$$

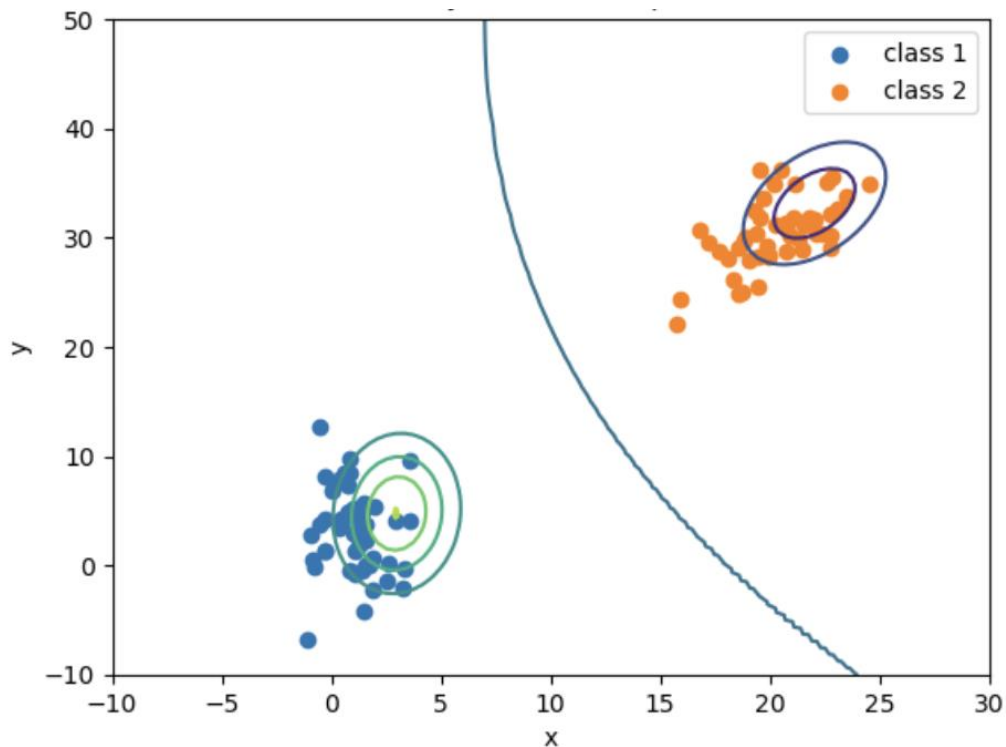
$$P(x | c_2) = 0.42 * \frac{\exp\left(-\frac{1}{2}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 22.04 \\ 33.15 \end{bmatrix}\right)^T \begin{bmatrix} 0.240814 & -0.06105 \\ -0.06105 & 0.08087 \end{bmatrix} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 22.04 \\ 33.15 \end{bmatrix}\right)\right)}{(2\pi) \begin{vmatrix} 5.135 & 3.876 \\ 3.876 & 15.29 \end{vmatrix}}^{\frac{1}{2}}$$

We can then create a MAP classifier in Python. This was implemented in a function called `MAP_classify`. We then see that the accuracy of the MAP classifier is also 100%. The following plot was created to display the decision boundary:



3.

Based on the results, we can conclude that it was correct to assume the probability distributions of the two classes were Gaussian. Taking a look at the contours for each class's estimated PDF on top of the test points, we get the following graph:



As seen above, the contours of the estimated PDF for each class matches the distribution of the test points closely. The maximum of the PDF for each class is somewhat off due to the noise that was added to the training data, but because the two classes are different enough, both the ML and MAP classifiers are able to correctly classify all the test points.

However, there are real-world cases where data cannot be fit accurately to a Gaussian distribution. In those cases, other types of classifiers, like MED and MMD classifiers, may be a better choice.

If no noise was added to the training data, the classifiers would likely perform worse when testing with real-world data. This is because real-world data is often noisy (depending on the scenario). According to the bias-variance tradeoff, if we were to remove the noise from the training data, it would make the classifier more accurate to the training data, but less resilient to noisy data. In the case that the test data follows the exact same distribution as the training data, then removing the noise may actually make the classifiers better, but in most real-world scenarios, this will not make the classifiers better.

4.

Based on classification accuracy alone, all 4 classifiers have the same classification accuracy of 100%. However, classification accuracy should not be the only metric used to measure the performance of a classifier.

For this particular data, where we know both the training and test data is characterized by a Gaussian distribution, the ML and MAP classifiers would be better classifiers than the MED and MMD classifiers. Furthermore, since we are given the prior probabilities for each class, this extra information should make the MAP classifier the most accurate on test data.

Nevertheless, there are scenarios where different classifiers may perform better. For example, if we were in a scenario where prior probabilities were not given, or prior probabilities are not accurate (such as with small datasets), then the ML classifier becomes the best option.

Deterministic classifiers, like MMD and MED classifiers, are typically better when the classes are well-defined and separate. They would also likely perform better when the datasets cannot be estimated with a Gaussian distribution. MMD and MED classifiers are also much less computationally intensive, so when dealing with large datasets, being able to choose simpler classifiers like the MED classifier may be beneficial.

However, when there could be overlap in the data points, then probabilistic classifiers, like MAP and ML classifiers, can provide more insight. They do this by providing the probabilities that a data point belongs to a class, rather than simply choosing one class and classifying the point as such.

Overall, there is no classifier that performs the best in all scenarios. Rather, different classifiers will be appropriate in different scenarios. It is a matter of analyzing the data and scenario in order to determine the best classifier.

Exercise 3

1.

The exponential distribution is characterized by the equation:

$$p(x) = \lambda e^{-\lambda x} \text{ for } x > 0$$

We have $\theta = \lambda$, so the equation becomes:

$$p(\{x_i\} | \theta) = \prod_{i=1}^N \theta e^{-\theta x_i}$$

Taking the natural log gives us:

$$\begin{aligned} l(\theta) &= \ln \left[\prod_{i=1}^N \theta e^{-\theta x_i} \right] \\ &= \sum_{i=1}^N \ln \theta - \theta x_i \end{aligned}$$

$$= N \ln \theta - \sum_{i=1}^N \theta x_i$$

Taking the derivative:

$$\frac{\partial}{\partial \theta} l(\theta) = \frac{N}{\theta} - \sum_{i=1}^N x_i$$

Setting it to zero:

$$\begin{aligned} \frac{N}{\theta} - \sum_{i=1}^N x_i &= 0 \\ \frac{N}{\theta} &= \sum_{i=1}^N x_i \\ \hat{\theta} &= \frac{N}{\sum_{i=1}^N x_i} \end{aligned}$$

Thus, λ is just the reciprocal of the sample mean.

2.

The uniform distribution is given by the equation:

$$p(x) = \frac{1}{b-a} \text{ for } a \leq x \leq b$$

We have $\theta_1 = a$ and $\theta_2 = b$, so the equation becomes:

$$p(\{x_i\} | \theta) = \prod_{i=1}^N \frac{1}{\theta_2 - \theta_1}$$

Taking the natural log gives us:

$$\begin{aligned} l(\theta) &= \ln \left[\prod_{i=1}^N \frac{1}{\theta_2 - \theta_1} \right] \\ &= \sum_{i=1}^N \ln \left(\frac{1}{\theta_2 - \theta_1} \right) \\ &= \sum_{i=1}^N \ln(\theta_2 - \theta_1)^{-1} \\ &= \ln(\theta_2 - \theta_1)^{-N} \\ &= -N \ln(\theta_2 - \theta_1) \end{aligned}$$

Taking the derivative:

$$\frac{\partial}{\partial \theta_1} l(\theta) = \frac{N}{\theta_2 - \theta_1}$$

$$\frac{\partial}{\partial \theta_2} l(\theta) = -\frac{N}{\theta_2 - \theta_1}$$

We cannot make these functions equal to 0, as N will be nonzero. To minimize these two functions, we simply choose θ_2 , or b , to be as large as possible and θ_1 , or a , to be as small as possible.

3.

The distribution given is:

$$p(x | \theta) = \frac{1}{2} [N(x | \theta_1, \theta_2) * f(x; \lambda)]$$

We know:

$$N(x | \theta_1, \theta_2) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\theta_2}} \exp \left[-\frac{1}{2} \frac{(x_i - \theta_1)^2}{\theta_2} \right]$$

$$f(x | \theta_3) = \prod_{i=1}^N \theta_3 e^{-\theta_3 x_i}$$

This gives us:

$$p(\{x_i\} | \theta_1, \theta_2, \theta_3) = \prod_{i=1}^N \frac{1}{2} \left[\frac{1}{\sqrt{2\pi\theta_2}} \exp \left[-\frac{1}{2} \frac{(x_i - \theta_1)^2}{\theta_2} \right] * \theta_3 e^{-\theta_3 x_i} \right]$$

Taking the log gives us:

$$\begin{aligned} l(\theta) &= \ln \left[\prod_{i=1}^N \frac{1}{2} \left[\frac{1}{\sqrt{2\pi\theta_2}} \exp \left[-\frac{1}{2} \frac{(x_i - \theta_1)^2}{\theta_2} \right] * \theta_3 e^{-\theta_3 x_i} \right] \right] \\ &= \sum_{i=1}^N \ln \left(\frac{1}{\sqrt{2\pi\theta_2}} \exp \left[-\frac{1}{2} \frac{(x_i - \theta_1)^2}{\theta_2} \right] \right) + \sum_{i=1}^N \ln \left(\frac{1}{2} \theta_3 e^{-\theta_3 x_i} \right) \\ &= \sum_{i=1}^N \left[-\frac{1}{2} \frac{(x_i - \theta_1)^2}{\theta_2} - \ln(\sqrt{2\pi\theta_2}) \right] + \sum_{i=1}^N \ln(\theta_3) - \theta_3 x_i - \ln(2) \\ &= \sum_{i=1}^N \left[-\frac{1}{2} \frac{(x_i - \theta_1)^2}{\theta_2} - \ln(\sqrt{2\pi\theta_2}) \right] + N \ln(\theta_3) - N \ln(2) - \sum_{i=1}^N \theta_3 x_i \end{aligned}$$

We can then take each of the partial derivatives and set them to 0. For θ_1 :

$$\begin{aligned}\frac{\partial l(\theta)}{\partial \theta_1} &= \sum_{i=1}^N \frac{x_i - \theta_1}{\theta_2} \\ \sum_{i=1}^N \frac{x_i - \theta_1}{\theta_2} &= 0 \\ \sum_{i=1}^N x_i &= N\theta_1 \\ \hat{\theta}_1 &= \frac{1}{N} \sum_{i=1}^N x_i\end{aligned}$$

For θ_2 :

$$\begin{aligned}\frac{\partial l(\theta)}{\partial \theta_2} &= \frac{1}{2} \sum_{i=1}^N \frac{(x_i - \theta_1)^2}{\theta_2^2} - \frac{N}{2\theta_2} \\ \frac{1}{2} \sum_{i=1}^N \frac{(x_i - \hat{\theta}_{1,ML})^2}{\theta_2^2} - \frac{N}{2\theta_2} &= 0 \\ \sum_{i=1}^N \frac{(x_i - \hat{\theta}_{1,ML})^2}{\theta_2^2} &= \frac{N}{\theta_2} \\ \sum_{i=1}^N (x_i - \hat{\theta}_{1,ML})^2 &= N\theta_2 \\ \hat{\theta}_{2,ML} &= \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\theta}_{1,ML})^2\end{aligned}$$

For θ_3 :

$$\begin{aligned}\frac{\partial l(\theta)}{\partial \theta_3} &= \frac{N}{\theta_3} - \sum_{i=1}^N x_i \\ \frac{N}{\theta_3} - \sum_{i=1}^N x_i &= 0 \\ \frac{N}{\theta_3} &= \sum_{i=1}^N x_i \\ \hat{\theta}_{3,ML} = \hat{\lambda} &= \frac{N}{\sum_{i=1}^N x_i}\end{aligned}$$

4.

Each of the three distribution classifiers were created and trained in Python and then tested on 100 noisy samples, with 50 samples generated per class. This process was done 100 times and the average accuracy for each distribution was reported. The accuracies for each distribution were as follows:

- Exponential classifier: 77%
- Uniform classifier: 76%
- Gaussian-exponential classifier: 83%

However, it is important to note that subsequent runs give relatively different accuracies for each classifier. In most cases, we see that the Gaussian-exponential classifier is the most accurate, which makes sense, as the data used to both train and test the model were generated using a Gaussian distribution. Meanwhile, since we know the data cannot be accurately described by an exponential or uniform distribution, it makes sense that their corresponding classifiers are usually not as accurate.