Assignment 1 - Q1

September 24, 2023 1:49 PM

Exercise 1: Equiprobability Contour (30 pts)

Let's consider the following two classes:

- Class 1: $N_1 = 5$, $\mu_1 = [4, 7]^T$, $\Sigma_1 = \begin{bmatrix} 9 & 3 \\ 3 & 10 \end{bmatrix}$
- Class 2: $N_2 = 5$, $\mu_1 = [5, 10]^T$, $\Sigma_1 = \begin{bmatrix} 7 & 0 \\ 0 & 16 \end{bmatrix}$

1. (1 pt) Use the *numpy.random.normal* function to generate 5 samples each for the above two classes. Find the sample mean and covariance matrices for the two classes by hand.
2. (10 pts) Using the sample means and covariance matrices, find the eigenvalues and eigenvectors of the covariance matrices by hand.
3. (2 pt) Using the above calculations, plot the equiprobability contours for the two classes in Python.
4. (2 pts) Can you comment on how the contours relate to the original cluster samples?
5. (15 pts) Let's change the number of samples for the two classes to 100, and regenerate these samples using the numpy.random.normal function. Based on the new data, find the sample mean, sample covariance, eigenvalues, and eigenvectors using Python. Plot the equiprobability contours for the two classes. Is there a difference between the contours generated for 100 samples and 5 samples?

1.

```
Class 1 Results:
The generated samples are: [[ 3.93 12.93]
 [ 5.31 10.4 ]
 [ 3.64  7.77]
 [ 4.88 14.27]
 [ 5.84  4.82]]
```

Class 1: $N_1 = 5$ $\mu_1 = [4, 7]^T$, $\Sigma_1 = \begin{bmatrix} 9 & 3 \\ 3 & 10 \end{bmatrix}$

* Calculate Sample Mean:

$$\mu_{sample} = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$$

$$\mu_{sample} = \frac{1}{5}\left(\left[3.93 + 5.31 + 3.64 + 4.88 + 5.84,\ 12.93 + 10.4 + 7.77 + 14.27 + 4.82\right]\right)$$

$$\boxed{\mu_{sample} = [4.72\ ,\ 10.04]}$$

* Calculate Sample Covariance:

$$Cov(\underline{x}) = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$$

$$Cov(x,y) = \frac{1}{N-1}\left(\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)\right)$$

$$Cov(x,y) = \frac{1}{5-1}\left((3.93-4.72)(12.93-10.04) + (5.31-4.72)(10.4-10.04) + (3.64-4.72)(7.77-10.04) + (4.88-4.72)(14.27-10.04) + (5.84-4.72)(4.82-10.04)\right)$$

$Cov(x,y) = -1.20$ //

$Cov(y,x) = -1.20$

$$Cov(x,x) = \frac{1}{N-1}\left(\sum_{i=1}^{N}(x_i - \mu_x)^2\right)$$

$$= \frac{1}{4}\left((3.93-4.72)^2 + (5.31-4.72)^2 + (3.64-4.72)^2 + (4.88-4.72)^2 + (5.84-4.72)^2\right)$$

$Cov(x,x) = 0.85$ //

$$Cov(y,y) = \frac{1}{N-1}\left(\sum_{i=1}^{N}(y_i - \mu_y)^2\right)$$

$$= \frac{1}{4}\left((12.93-10.04)^2 + (10.4-10.04)^2 + (7.77-10.04)^2 + (14.27-10.04)^2 + (4.82-10.04)^2\right)$$

$= 14.69$ //

$$\Rightarrow \boxed{Cov = \begin{bmatrix} 0.85 & -1.2 \\ -1.2 & 14.69 \end{bmatrix}}$$ //

Class 2: $N_2 = 5$ $\mu_2 = [5, 10]^T$, $\Sigma_2 = \begin{bmatrix} 7 & 0 \\ 0 & 16 \end{bmatrix}$

```
Class 2 Results:
The generated samples are: [[ 9.04  4.94]
 [ 6.24  6.12]
 [ 5.8   9.6 ]
 [ 9.19  3.1 ]
 [ 2.07 10.54]]
```

* Calculate Sample Mean:

$$\mu_{sample} = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$$

$$\mu_{sample} = \frac{1}{5}\left(\left[9.04 + 6.24 + 5.8 + 9.19 + 2.07,\ 4.94 + 6.12 + 9.6 + 3.1 + 10.54\right]\right)$$

$$\boxed{\mu_{sample} = [6.47,\ 6.86]}$$

* Calculate Sample Covariance:

$$Cov(\underline{x}) = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$$

$$Cov(x,y) = \frac{1}{N-1}\left(\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)\right)$$

$$Cov(x,y) = \frac{1}{5-1}\left[(9.04-6.47)(4.94-6.86) + (6.24-6.47)(6.12-6.86) + (5.8-6.47)(9.6-6.86) + (9.19-6.47)(3.1-6.86) + (2.07-6.47)(10.54-6.86)\right]$$

$Cov(x,y) = -8.25$ //

$Cov(y,x) = -8.25$ //

$$Cov(x,x) = \frac{1}{N-1}\left(\sum_{i=1}^{N}(x_i - \mu_x)^2\right)$$

$$= \frac{1}{4}\left[(9.04-6.47)^2 + (6.24-6.47)^2 + (5.8-6.47)^2 + (9.19-6.47)^2 + (2.07-6.47)^2\right]$$

$Cov(x,x) = 8.47$ //

$$Cov(y,y) = \frac{1}{N-1}\left(\sum_{i=1}^{N}(y_i - \mu_y)^2\right)$$

$$= \frac{1}{4}\left[(4.94-6.86)^2 + (6.12-6.86)^2 + (9.6-6.86)^2 + (3.1-6.86)^2 + (10.54-6.86)^2\right]$$

$= 9.86$ //

$$\Rightarrow \boxed{Cov = \begin{bmatrix} 8.47 & -8.25 \\ -8.25 & 9.86 \end{bmatrix}}$$ //

2. To find the eigenvalues we can use:

$$\det(\Sigma - \lambda I) = 0$$

Class 1:

where $\det\begin{vmatrix} 0.85-\lambda & -1.2 \\ -1.2 & 14.69-\lambda \end{vmatrix} = 0$

$$(0.85-\lambda)(14.69-\lambda) - (-1.2)^2 = 0$$

$$12.49 - 0.85\lambda - 14.69\lambda + \lambda^2 - 1.44 = 0$$

$$\lambda^2 - 15.54\lambda + 11.05 = 0$$

$$\boxed{\begin{matrix}\lambda_1 = 14.79 \\ \lambda_2 = 0.75\end{matrix}}$$ //

- To find the eigenvectors, we use: $(\Sigma - \lambda I)\underline{V}_1 = 0$

$$\Rightarrow \begin{bmatrix} 0.85-\lambda & -1.2 \\ -1.2 & 14.69-\lambda \end{bmatrix}\begin{bmatrix} V_{11} \\ V_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Assume $V_{11} = 1$

$(0.85-\lambda) - 1.2V_{12} = 0$

For $\lambda = 0.75$

$(0.85 - 0.75) - 1.2V_{12} = 0$

$V_{12} = 0.083$ $\wedge$ $V_{11} = -0.996$

For $\lambda = 14.79$ and for $V_{22} = 1$

$-1.2V_{21} + (14.69 - 14.79) = 0$

$V_{21} = -0.083$ $\wedge$ $V_{22} = -0.996$

$$\boxed{V_{C_1} = \begin{bmatrix} -0.996 & 0.083 \\ -0.083 & -0.996 \end{bmatrix}}$$ //

Class 2:

$$\det(\Sigma - \lambda I) = 0$$

where $\det\begin{vmatrix} 8.47-\lambda & -8.25 \\ -8.25 & 9.86-\lambda \end{vmatrix} = 0$

$$(8.47-\lambda)(9.86-\lambda) - (-8.25)^2 = 0$$

$$83.51 - 8.47\lambda - 9.86\lambda + \lambda^2 - 68.06 = 0$$

$$\lambda^2 - 18.27\lambda + 15.45 = 0$$

$$\boxed{\begin{matrix}\lambda_1 = 17.38 \\ \lambda_2 = 0.88\end{matrix}}$$ Eigen values. //

To find the eigen vectors, we need

$$(\Sigma - \lambda I)\underline{V}_1 = 0$$

$$\begin{bmatrix} 8.47-\lambda & -8.25 \\ -8.25 & 9.86-\lambda \end{bmatrix}\begin{bmatrix} V_{11} \\ V_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Assuming $V_{11} = 1$ and for $\lambda = 0.88$

$$\begin{bmatrix} 7.59 & -8.25 \\ -8.25 & 8.98 \end{bmatrix}\begin{bmatrix} 1 \\ V_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$-8.25 + 8.98V_{12} = 0$

$V_{12} = 0.92$ $V_{11} = 0.99$

Assuming $V_{22} = 1$ for $\lambda = 17.38$

$$\begin{bmatrix} -8.91 & -8.25 \\ -8.25 & -7.52 \end{bmatrix}\begin{bmatrix} V_{21} \\ V_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

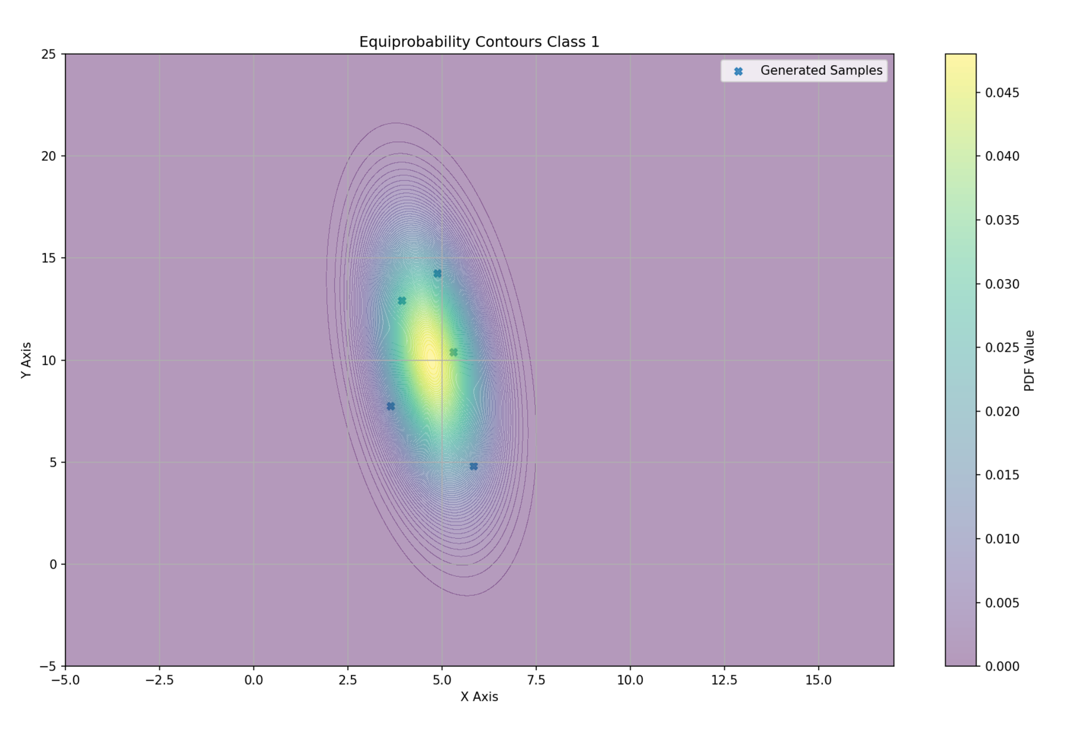
$-8.91V_{21} - 8.25(1) = 0$

$V_{21} = -0.92$

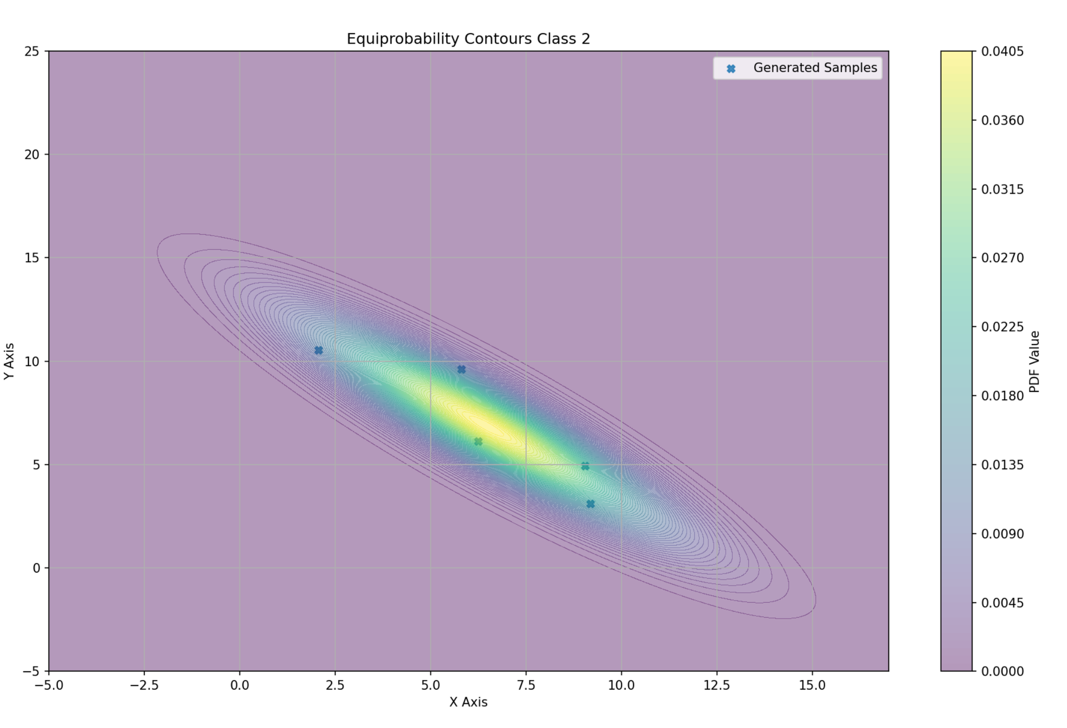
$V_{22} = 0.83$

$$\boxed{V_{C_1} = \begin{bmatrix} 0.99 & 0.92 \\ -0.92 & 0.83 \end{bmatrix}}$$ //

Class 1:



Class 2:



For both class 1 and class 2, the 5 generated samples (for each class) can be found within the contours. Both classes have one sample closer to the center of the countor, being those samples the ones with higher likelihood of experiencing an event with the specified probability. Also, the samples are not uniformly distributed across the contour plot.

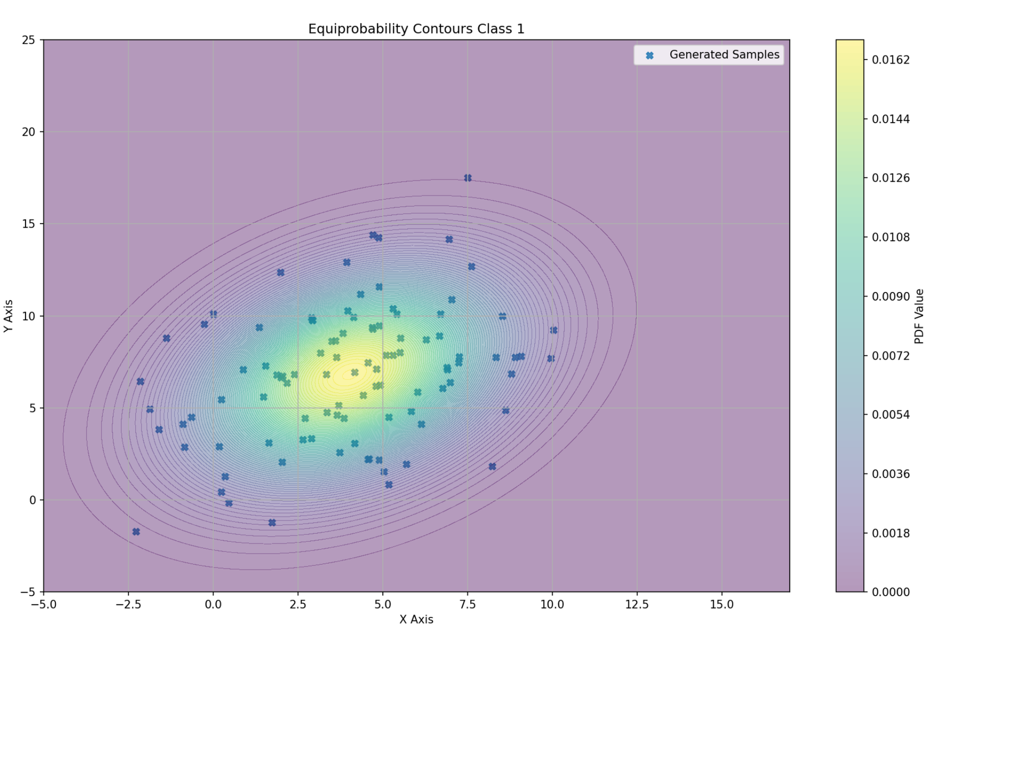
For number of sample = 100

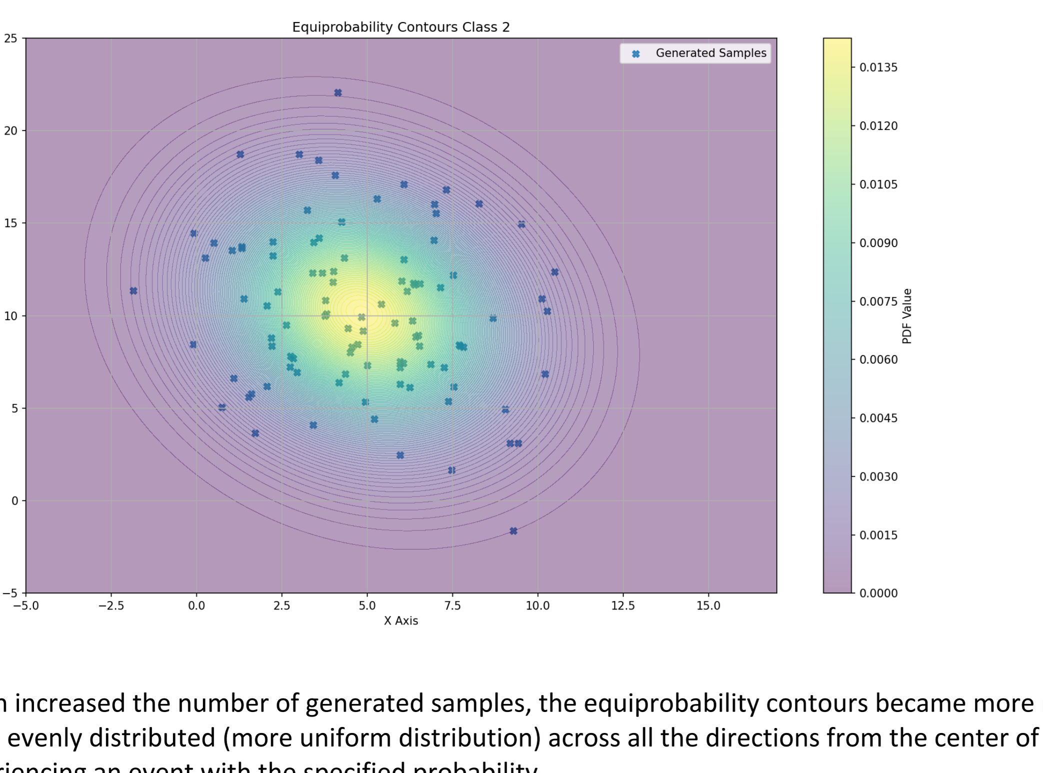
Class 1:

```
Sample Mean: [4.89 6.81]
Sample Covariance: [[ 8.37  3.51]
 [ 3.51 12.61]]
E-value: [ 6.53077483 14.45022597]
E-vector: [[-0.88534613 -0.46493255]
 [ 0.46493255 -0.88534613]]
```

Class 2:

```
Sample Mean: [4.85 10.14]
Sample Covariance: [[ 7.27 -2.05]
 [-2.05 17.75]]
E-value: [ 6.89716358 18.12283642]
E-vector: [[-0.98354895  0.18064141]
 [-0.18064141 -0.98354895]]
```





When increased the number of generated samples, the equiprobability contours became more rounded, and the data points are seemed to be more evenly distributed (more uniform distribution) across all the directions from the center of the contour. This means that it is more likely of experiencing an event with the specified probability.
Overall, it can be concluded that increasing the number of samples in a dataset can provide a more uniform representation of the data leading to the improvement of the accuracy of probability density calculation.

Assignment 1 - Q2

September 30, 2023 9:17 PM

Exercise 2: MED Classifier (40 pts)

In this exercise, you will implement the MED classifier, using the same data you generated in the previous exercise.

1. (15 pts) Let's consider the 5 sample datasets that you generated for the two classes in the previous question. Find the decision boundary for the MED classifier by hand using the sample means and covariances. Plot the boundary and the data.
2. (10 pts) Now, let's consider the 100 sample datasets for the two classes. Find the decision boundary for the MED classifier in Python using the sample means and covariances. Plot the boundary and the data.
3. (10 pts) Generate 50 new samples per class for the two classes. Add some white noise to teach the 50 samples using numpy.random.normal with 0 mean and identity covariance matrix. Using the MED classifiers learned above, predict the labels for the 100 noisy samples generated for the two classes (50 samples per class). Report the classification accuracy of the classifiers using:

$$error = \frac{Number\ of\ correct\ predictions}{total\ number\ of\ data\ points\ in\ the\ test\ set} \quad (1)$$

4. (5 pts) Which of the two classifiers is better, the one learned from 5 samples or the one with 100 samples? Why?

1. Find the decision boundary of MED by hand.

Class 1:

$\underline{x_1} = [3.93, \ 12.93]$

$\underline{x_2} = [5.31, \ 10.4]$

$\underline{x_3} = [3.64, \ 7.77]$

$\underline{x_4} = [4.88, \ 14.27]$

$\underline{x_5} = [5.84, \ 4.82]$

Sample mean (z_1):

$\underline{z_1} = [4.72 \quad 10.04]^T$

Class 2:

$\underline{x_1} = [9.04, \ 4.94]$

$\underline{x_2} = [6.24, \ 6.12]$

$\underline{x_3} = [5.8, \ 9.6]$

$\underline{x_4} = [9.19, \ 3.1]$

$\underline{x_5} = [2.07, \ 10.54]$

Sample mean (z_2):

$\underline{z_2} = [6.47, \ 6.86]^T$

- To find the discriminant function for each class based on MED:

$$d_E(\underline{x}, \underline{z_1}) < d_E(\underline{x}, \underline{z_2})$$

$$[(\underline{x}-\underline{z_1})^T (\underline{x}-\underline{z_1})]^{1/2} < [(\underline{x}-\underline{z_2})^T (\underline{x}-\underline{z_2})]^{1/2}$$

$$(x-z_1)^T (\underline{x}-\underline{z_1}) < (\underline{x}-\underline{z_2})^T (\underline{x}-\underline{z_2})$$

$$-z_1^T\underline{x} + \frac{1}{2} \underline{z_1}^T\underline{z_1} < -\underline{z_2}^T\underline{x} + \frac{1}{2}\underline{z_2}^T\underline{z_2}$$

$$-[4.72 \quad 10.04]^{T^T}[x_1 \ x_2]^T + \frac{1}{2}[4.72 \quad 10.04]^{T^T}[4.72 \quad 10.04]^T < -[6.47 \quad 6.86]^{T^T}[x_1 \ x_2]^T + \frac{1}{2}[6.47 \quad 6.86]^{T^T}[6.47 \quad 6.86]^T$$

$$-[4.72 \quad 10.04][x_1 \quad x_2]^T + \frac{1}{2}[4.72 \quad 10.04][4.72 \quad 10.04]^T < -[6.47 \quad 6.86][x_1 \quad x_2]^T + \frac{1}{2}[6.47 \quad 6.86][6.47 \quad 6.86]^T$$

$$-4.72x_1 - 10.04x_2 + \frac{1}{2}\left((4.72)^2 + (10.04)^2\right) < -6.47x_1 - 6.86x_2 + \frac{1}{2}\left((6.47)^2 + (6.86)^2\right)$$

$$-4.72x_1 - 10.04x_2 + 61.54 < -6.47x_1 - 6.86x_2 + 44.46$$

The discriminant functions are:

$$g_1(x_1, x_2) = -4.72x_1 - 10.04x_2 + 61.54$$

$$g_2(x_1, x_2) = -6.47x_1 - 6.86x_2 + 44.46$$

- To find the decision boundary between C_1 and C_2

$$g(\underline{x}) = (\underline{z_k} - \underline{z_l})^T\underline{x} + \frac{1}{2}(\underline{z_l}^T\underline{z_l} - \underline{z_k}^T\underline{z_k}) = 0$$

$$g(x_1, x_2) = g_1(x_1, x_2) - g_2(x_1 - x_2) = 0$$

Plugging in the discriminant functions g_1, g_2 we have:

$$g(x_1, x_2) = -4.72x_1 - 10.04x_2 + 61.54 + 6.47x_1 + 6.86x_2 - 44.46 = 0$$

$$\boxed{g(x_1, x_2) = 1.75x_1 - 3.18x_2 + 17.08 = 0}$$

Grouping terms:

$$1.75x_1 - 3.18x_2 + 17.08 = 0$$

$$-3.18x_2 = -1.75x_1 - 17.08$$

$$x_2 = \frac{1.75}{3.18}x_1 + \frac{17.08}{3.18}$$

$$x_2 = 0.55x_1 + 5.37$$

The decision boundary is a straight line with a slope of: 0.55 and offset of 5.37.

If $x_1 = 0$, $x_2 = 5.37$
$x_2 = 0$, $x_1 = -9.76$

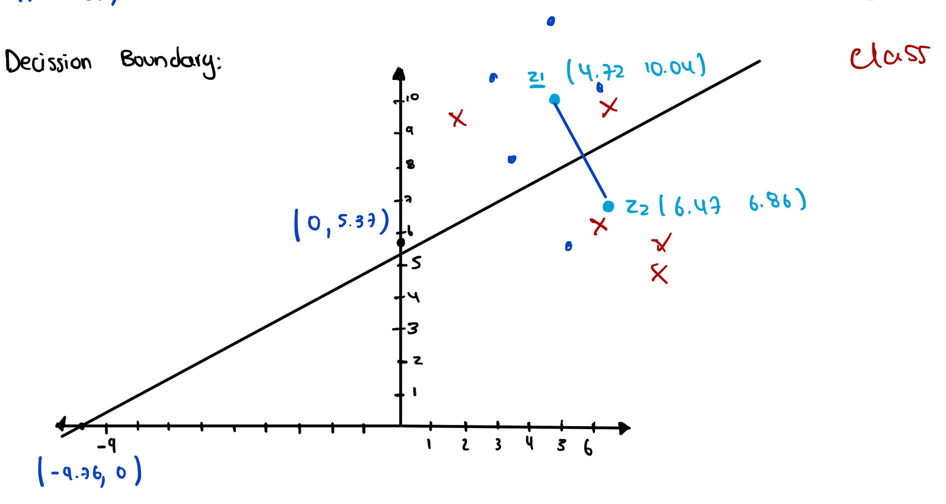
$\underline{z_1} = (4.72 \quad 10.04)$

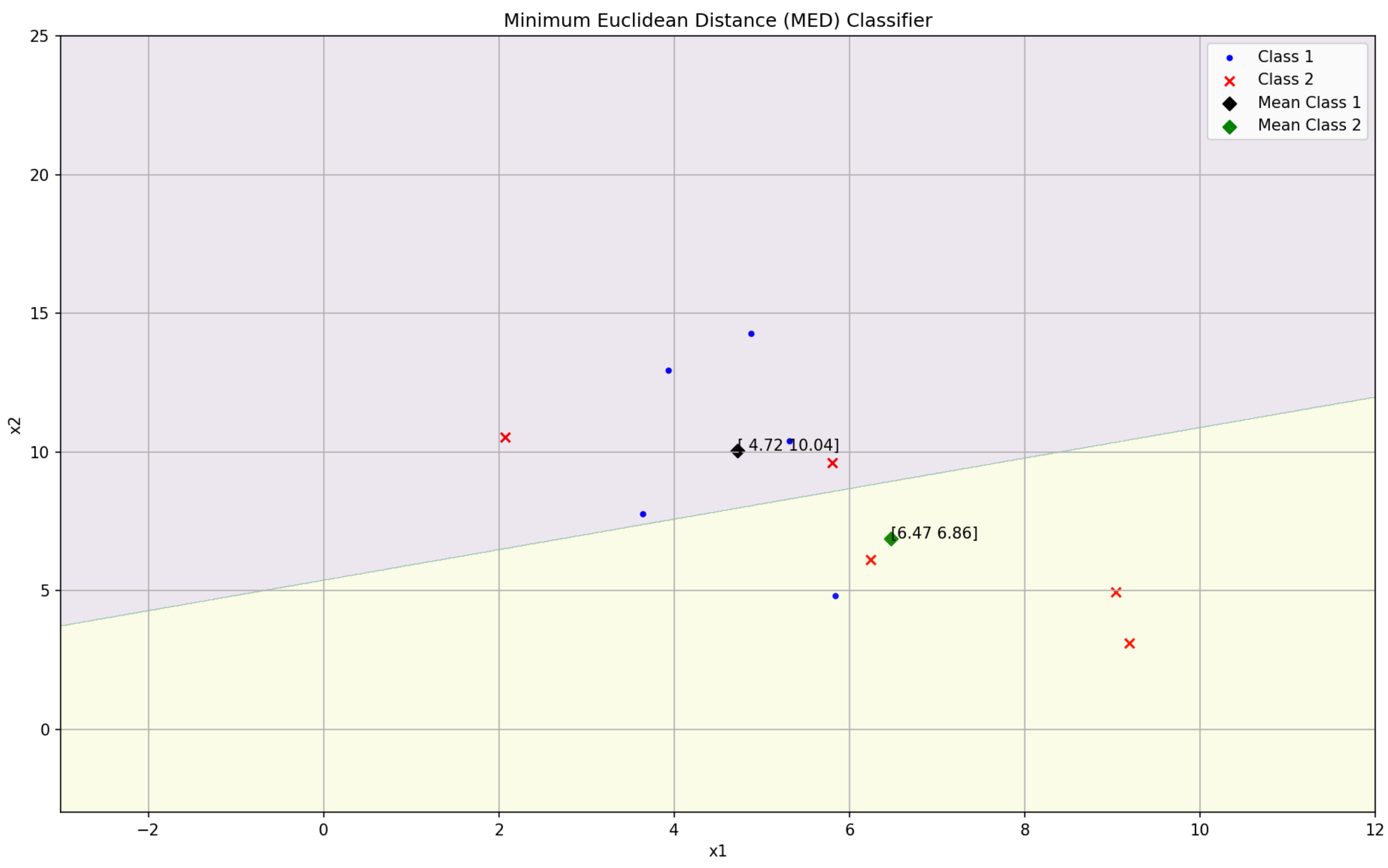
$\underline{z_2} = (6.47 \quad 6.86)$

Class 1 •

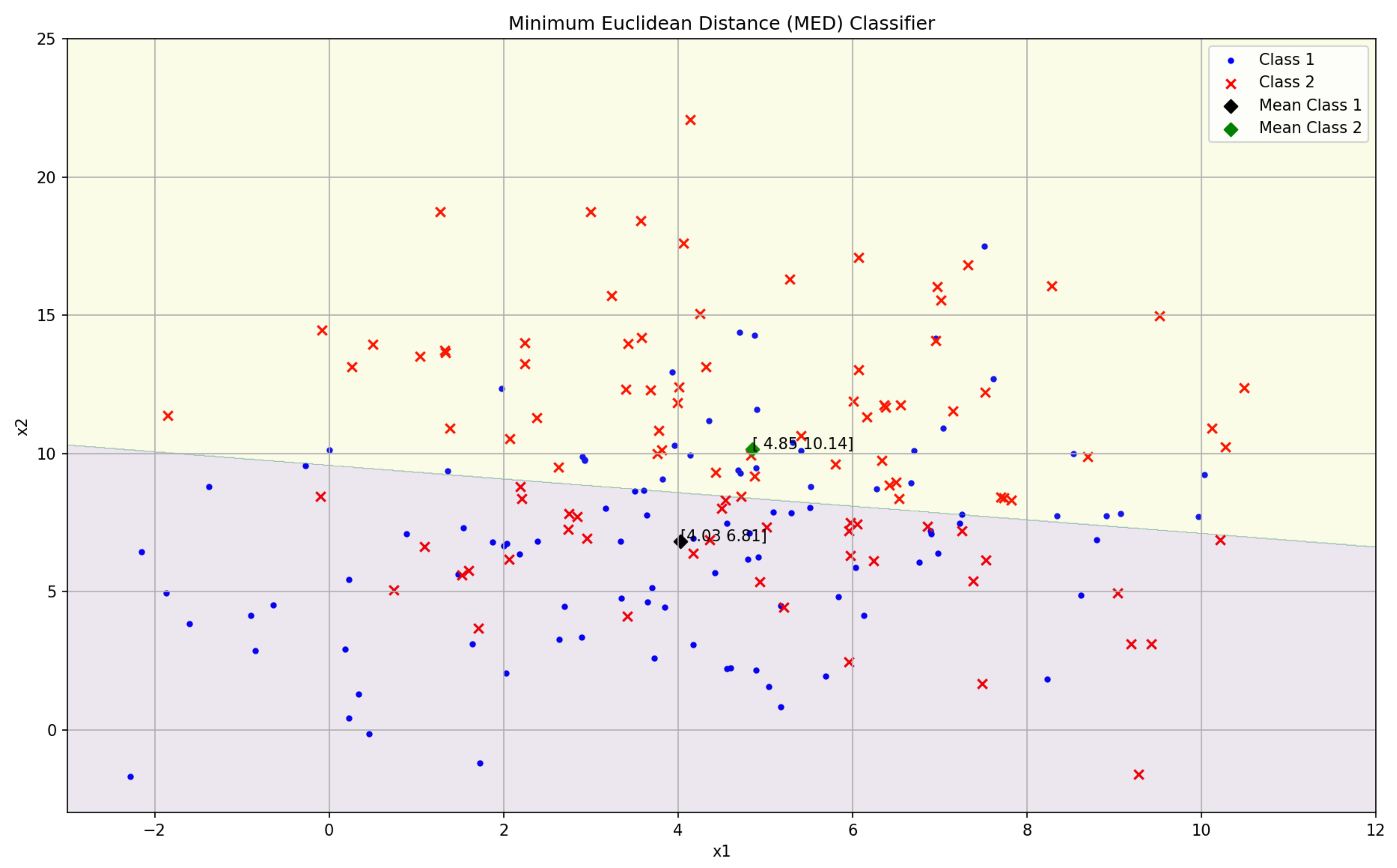
class 2 ×

Decission Boundary:






```
The Decision Boundary Equation is a straight line with a slope of -0.25 and an offset of 9.56
The line equation is: y = -0.25x + 9.56
```




```
The predicted labels using MED for 100 samples are: [2 2 1 2 1 1 2 1 1 2 1 2 2 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2
 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 1 2 2 1 2 2 1 2 1 2 1 2 2 2 2
 1 2 2 2 2 1 1 1 1 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2]
The error for the MED classifier trained with 100 samples is: 0.68
The predicted labels using MED for 5 samples are: [2 2 1 2 1 1 2 1 1 2 1 2 2 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2
 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 1 2 2 1 2 2 1 2 1 2 1 2 2 2 2
 1 2 2 2 2 1 1 1 1 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2]
The error for the MED classifier trained with 5 samples is: 0.31
```

The classifier that was trained with 100 samples led to a higher accuracy (68%) compared to the one trained with 5 samples (31%). These results make sense since having more diversed samples in the train_dataset can help in calculating more accurate classes' averages, which are the reference points for the classification of new test samples.

Assignment 1 - Q3

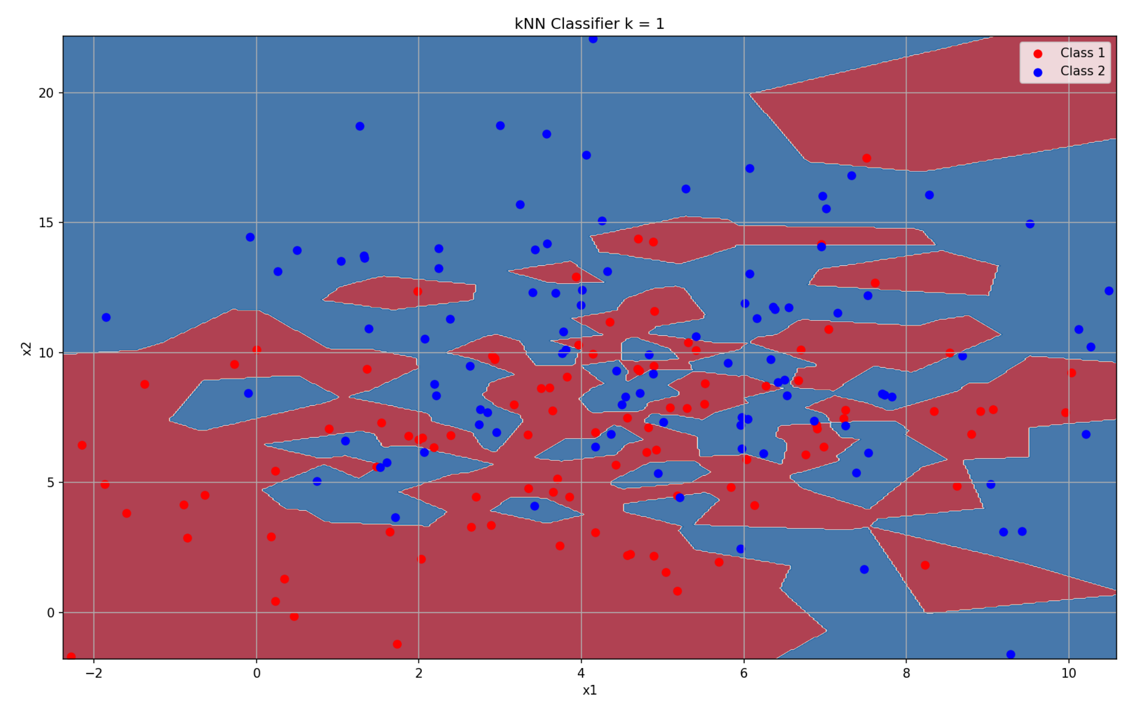
October 2, 2023 11:01 PM

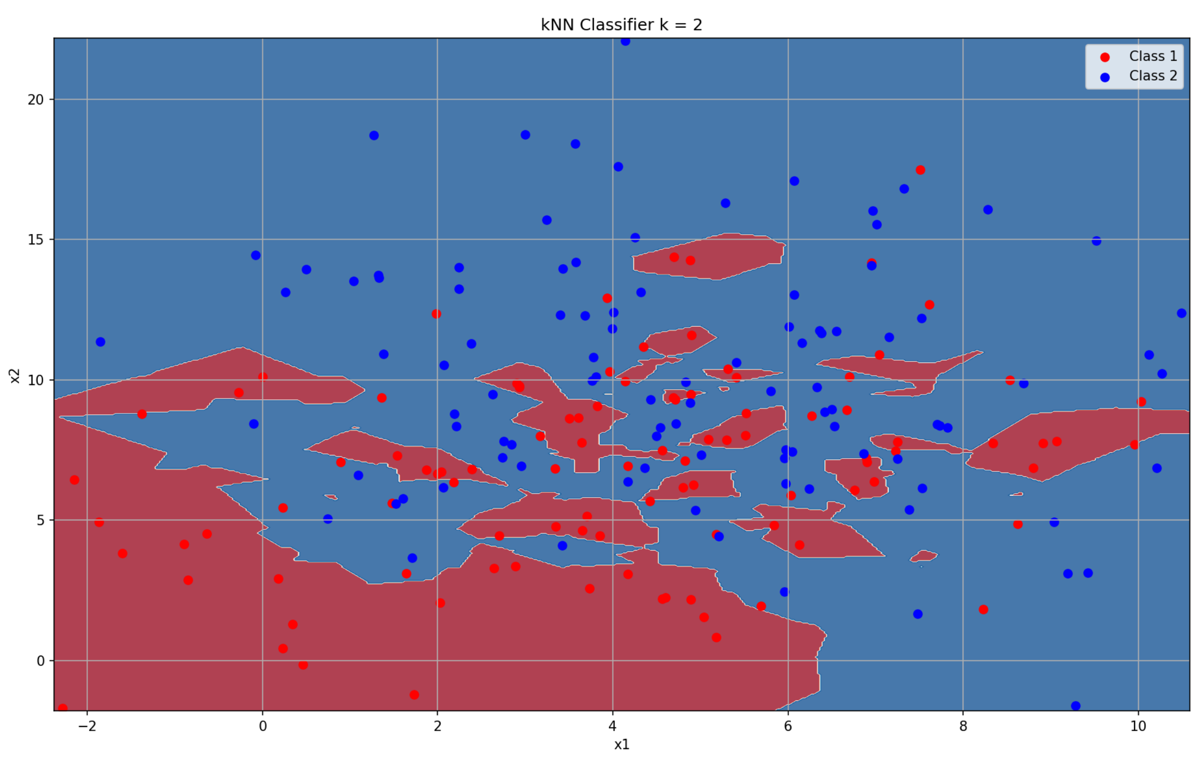
Exercise 3: Nearest Neighbor Classifier (40 pts)

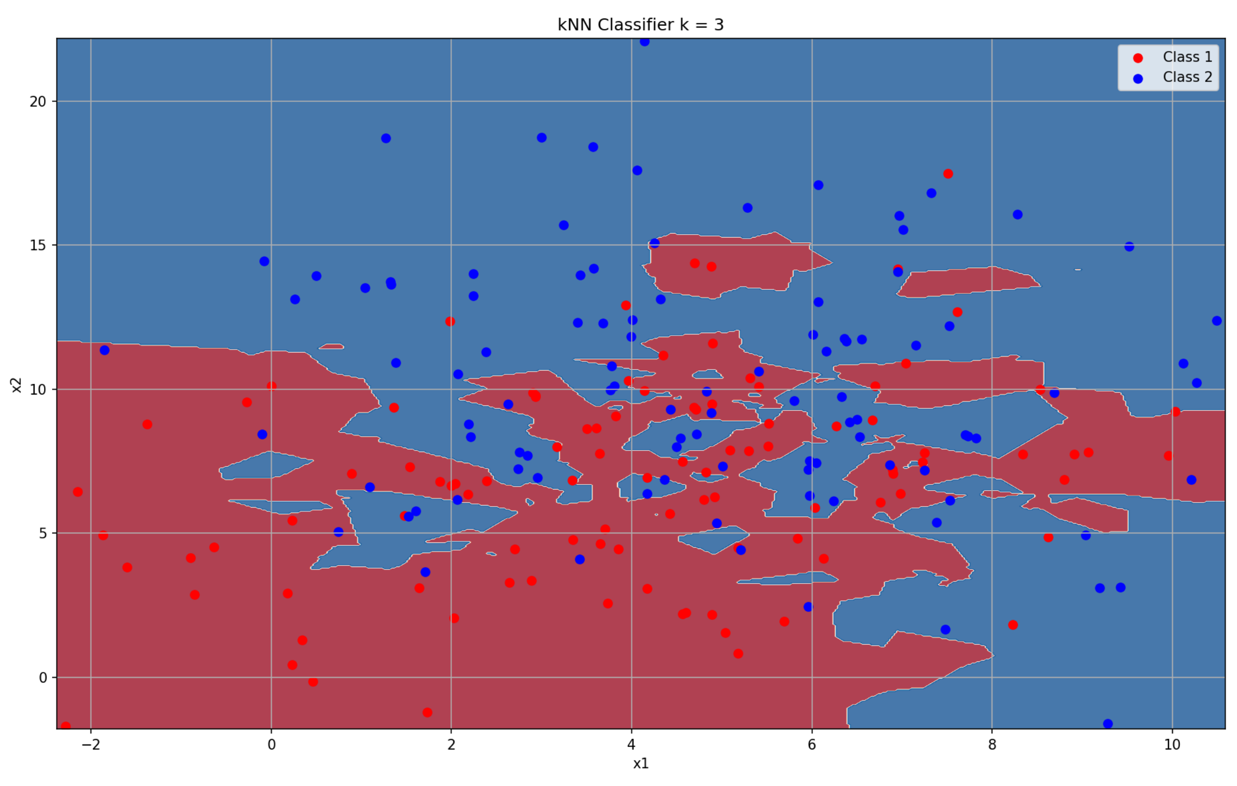
In this exercise, you will use the 100 sample datasets for two classes for a kNN classifier.

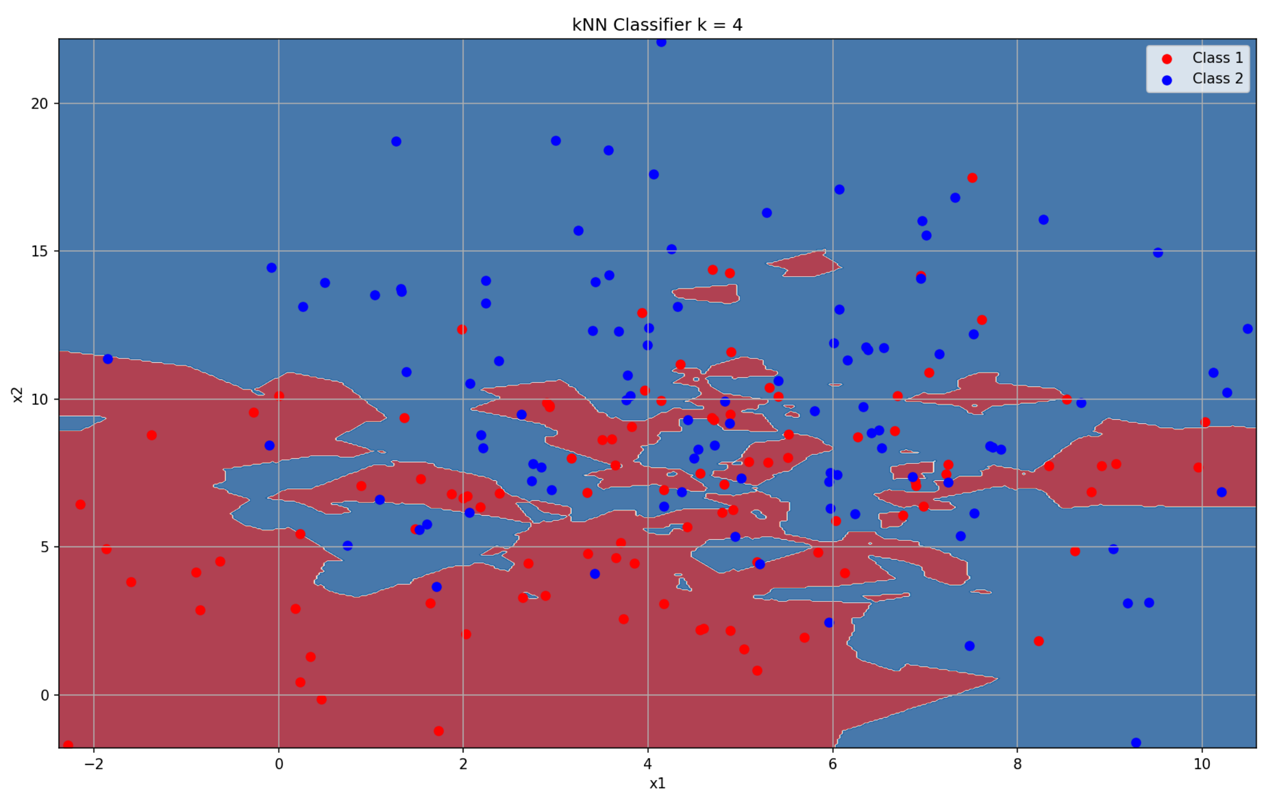
1. (15 pts) Implement the k-nearest neighbor classifier using the dataset. Use Euclidean distance as the distance metric. Compute the kNN solution for each integer k from 1 to 5. Plot the classification boundaries between the two classes for the kNN classifier for each value of k between 1 and 5.
2. (15 pts) Use the noisy data that you generated for the two classes in the previous exercise and find the classification accuracy for the kNN classifiers. Plot the accuracy for each value of k.
3. (5 pts) Which k value seems to be producing the best results? Why?
4. (5 pts) How does the kNN classifier compare against the MED classifier in the previous exercise?

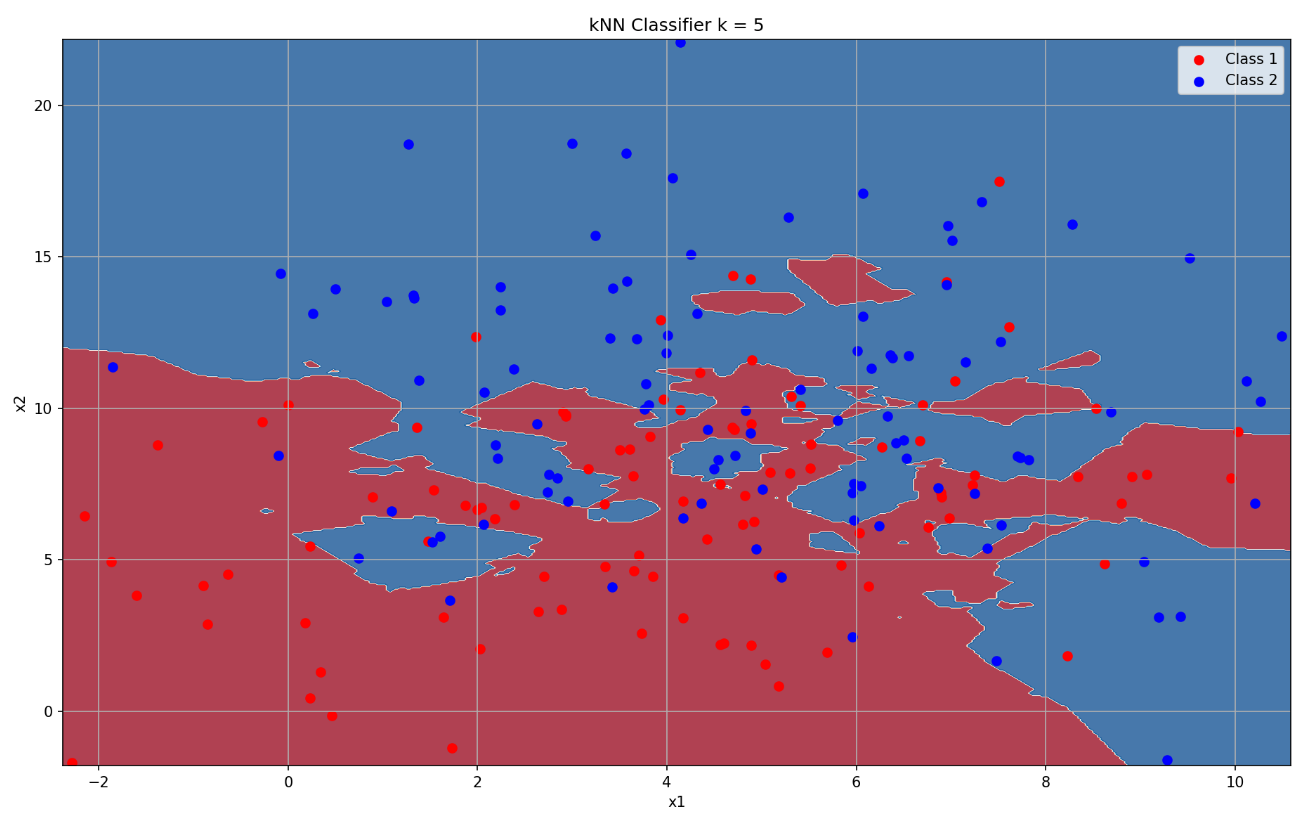
1. Classification boundaries between the two classes k between 1 and 5



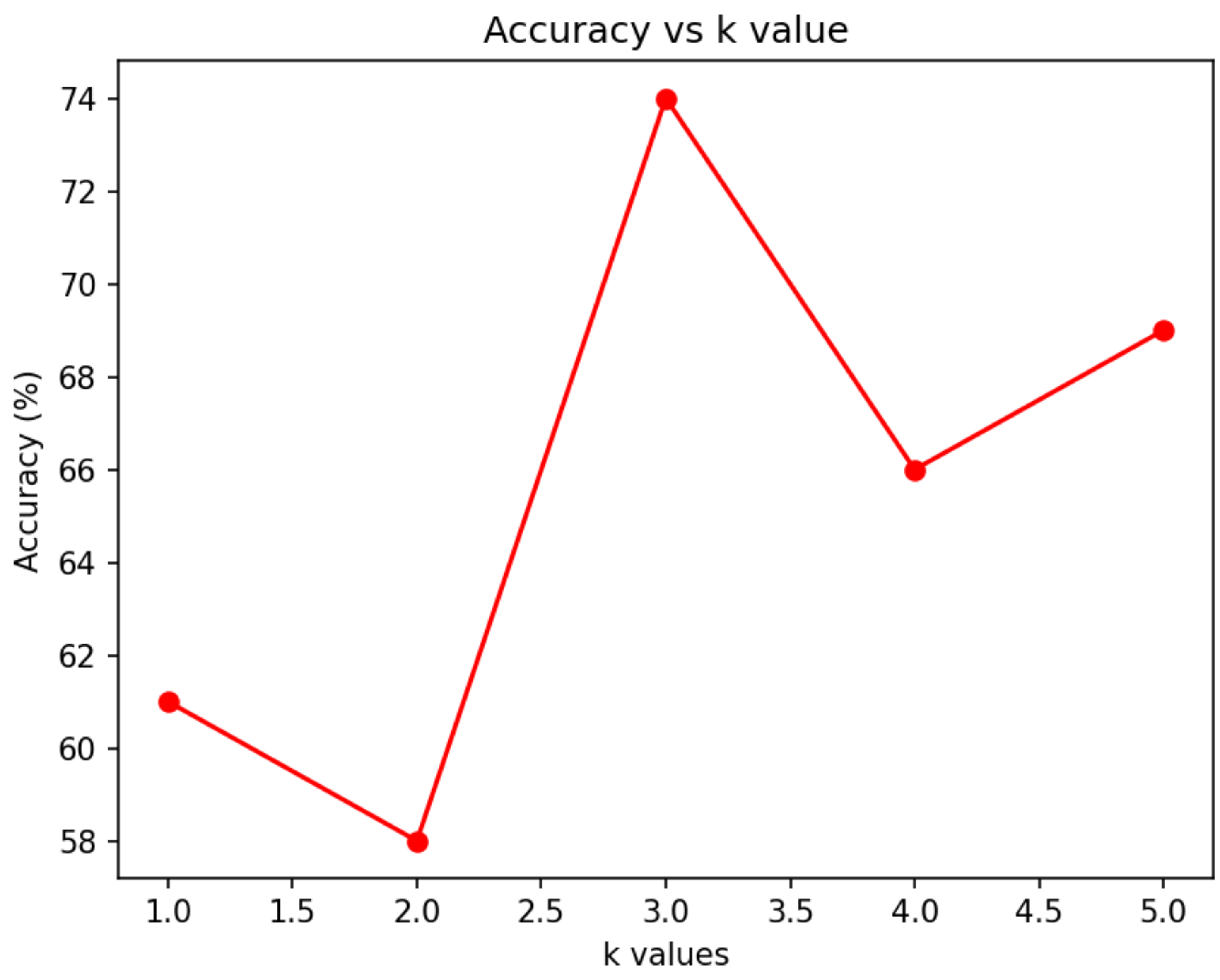








2. Find the classification accuracy for the kNN classifiers using the naisy data. Plot the accuracy for each k value




```
The accuracy for the kNN classifier with k= 1 is: 61.0 %
The accuracy for the kNN classifier with k= 2 is: 58.0 %
The accuracy for the kNN classifier with k= 3 is: 74.0 %
The accuracy for the kNN classifier with k= 4 is: 66.0 %
The accuracy for the kNN classifier with k= 5 is: 69.0 %
```

3. For this dataset, the k = 3 value led to a higher accuracy even though the classifier using k =1 seemed to have a better decision boundary when trained. This could have happened since k1 might have led to overfitting (the model was trained to well) which impacted the accuracy when predicting new data. And, for the classifiers using k =2, 4, and 5, there were more outliers than for k =3. Maybe, this led to lower accuracies compared to k=3.

4. The kNN leads to a slightly higher accuracy with 74% compared to the 100 samples trained MED classifier that had an accuracy of 68%. This is given since the kNN classifier can capture complex decision boundaries where the data does not necessarily have Gaussian-like class distributions or does not have a linear pattern. Therefore, kNN can generate more accurate predictions based on observable data similarities and distance (non-linear).