SYDE 572: Intro to Pattern Recognition
Assigment 1

Cooper Ang (20768006)

October 3, 2022

1 Exercise 1

1.1 Question 1

(15 pts) Implement the k-nearest neighbour classifier on the MNIST 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.

I first flattened the images to 784x1 images, and then applied PCA to lower the dimensions to 2D space. I then implemented a kNN neighbor class that uses the kNN to determine which class a prediction belongs with. Specifically I used a weighted distance metric (reciprocal of distance) to vote on which class the input vector belongs to. I used numpy's meshgrid to calculate the decision boundaries below. You can see as the k-value increases, the decision boundary becomes smoother.

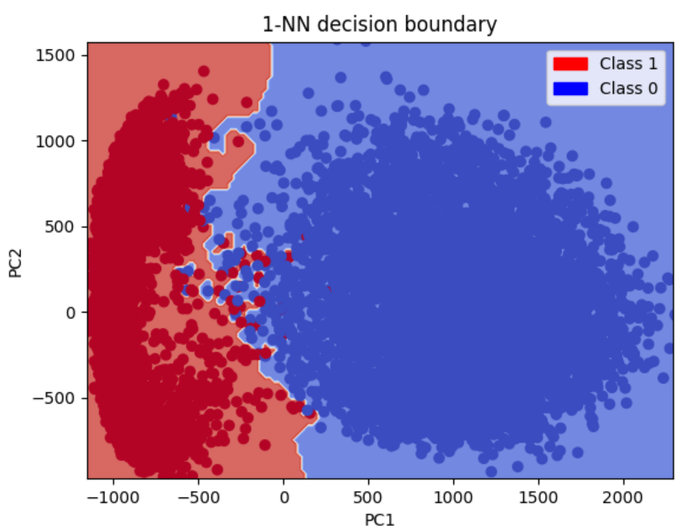

Figure 1: 1-NN Decision Boundary

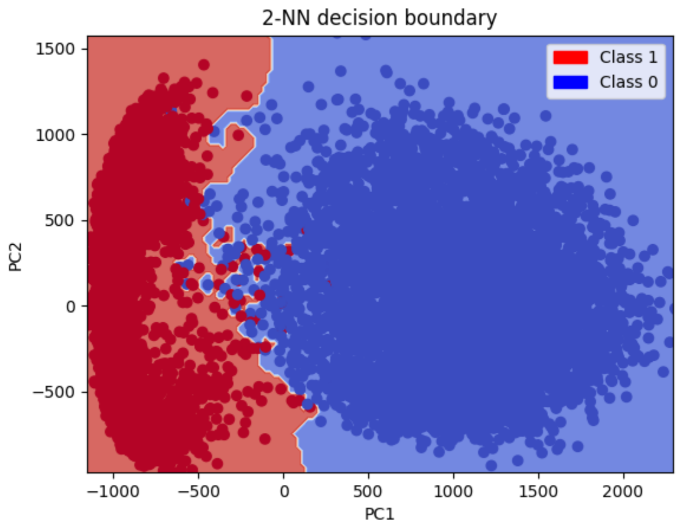

Figure 2: 2-NN Decision Boundary

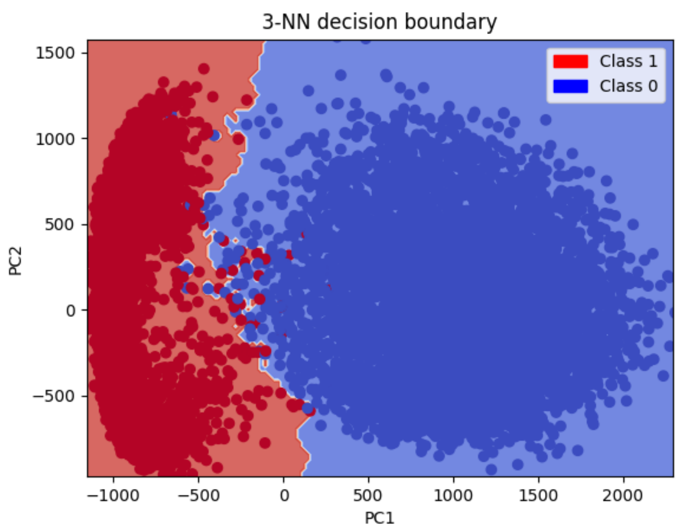

Figure 3: 3-NN Decision Boundary

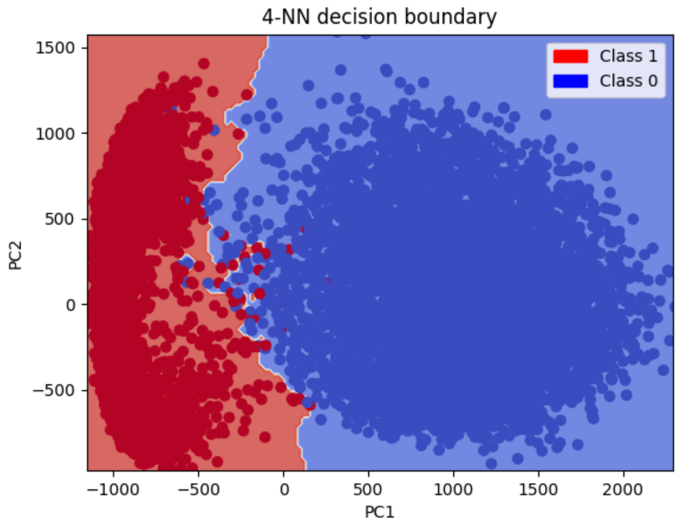

Figure 4: 4-NN Decision Boundary

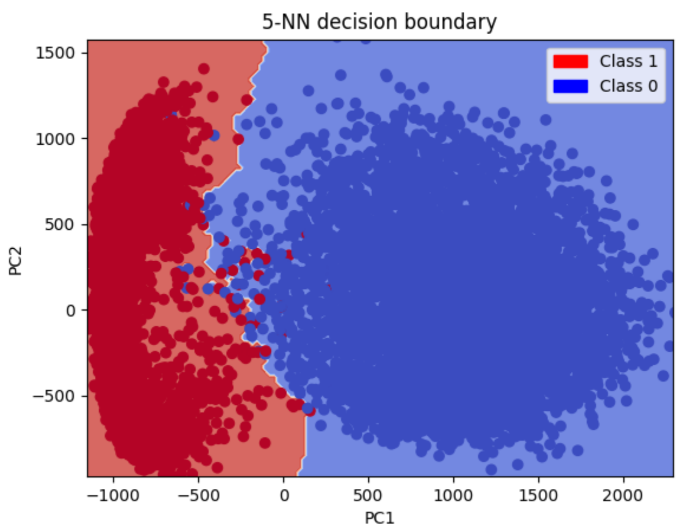

Figure 5: 5-NN Decision Boundary

1.2 Question 2

(15 pts) Use the test set of the two classes in the MNIST dataset, and make label predictions for all the test vectors for the two classes using your kNN classifier. Find the prediction error for your classifier as:

$$\text{error} = \frac{\text{Number of incorrect predictions}}{\text{total number of data points in the test set.}}$$

Plot the test set error for each value of k.

The classifier was run multiple times with different k-values yielding the bar graph in Figure 6. The numerical values of the errors are as follows:

kNN with k=1 has an error of 0.003783
kNN with k=2 has an error of 0.003783
kNN with k=3 has an error of 0.002364
kNN with k=4 has an error of 0.002837
kNN with k=5 has an error of 0.002364

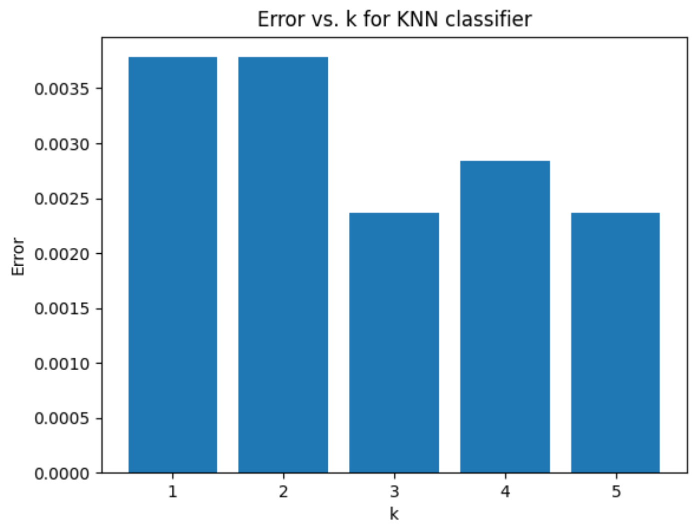

Figure 6: Error vs k for kNN classifier

1.3 Question 3

(5 pts) Which k value seems to be producing the best results? Why?

The k-value of 3 and 5 produce the best results with an error of 0.002364 or an accuracy of 99.764%. Generally, it seems like a higher k-value results in a lower error. This is cause that a higher k-value uses more points of the surrounding data in the vector space and thus outliers have a lesser impact. In other words, the decision boundary of higher k-values is smoother leading to better generalization of the pattern recognition task and reducing the impact of noise. There is a limit to how high of a k-value is suitable, especially when taking into account the number of training data per class. In this case, it seems k=5 is the best.

2 Exercise 2

2.1 Question 1

(20 pts) Implement the MED and GED classifiers on the MNIST dataset. Determine the decision boundaries for the two classifiers. Can you plot this decision boundary? Explain.

Using the same flattened image matrices, I applied PCA to turn the 784x1 matrices into 20x1 matrices. The decision boundaries were then found analytically through the code and the resulting decision boundary equations are shown below. Because the matrices are in 20D space the decision boundaries of both the MED and GED classifiers lie in 20D space, thus making it impossible to visualize.

In the case of two classes, the MED classifier decision boundary is determined by setting the discriminant function of both prototypes to equal each other. In other words, I can find the decision boundary by finding the equidistant euclidean distance between the two prototypes as seen in Equation 1. Letting $g(\underline{x}_0)$ be the euclidean distance to prototype 0 and $g(\underline{x}_1)$ be the euclidean distance to prototype

1:

$$g(\underline{x}) = g_0(\underline{x}) - g_1(\underline{x}) = 0 \tag{1}$$

Plugging in the discriminant function (Equation 2) with z_0 as prototype 0 and z_1 as prototype 1, and simplifying results in a hyperplane as defined by Equation 3.

$$-\underline{z}_0^\top \underline{x} + \frac{1}{2}\underline{z}_0^\top \underline{z}_0 = -\underline{z}_1^\top \underline{x} + \frac{1}{2}\underline{z}_1^\top \underline{z}_1 \tag{2}$$

$$g(\underline{x}) = \underline{w}^\top \underline{x} + w_0 = 0 \tag{3}$$

Calculating the values using the 20D data results in:

$$\underline{w} = \begin{bmatrix} -1923.575 \\ -32.34089 \\ 34.45161 \\ -64.62583 \\ 23.66002 \\ 15.39483 \\ -71.55443 \\ -57.20006 \\ 7.339949 \\ -35.55167 \\ -10.15746 \\ -6.077976 \\ -13.90665 \\ -11.8269 \\ -4.17517 \\ -1.241307 \\ 6.437268 \\ 5.120736 \\ -9.913559 \\ 12.03327 \end{bmatrix}$$

$$w_0 = 120210.3754$$

The GED classifier decision boundary is determined by the intersections of the corresponding equidistant contours around the classes. This can be determined by the Equation 4. If you expand and simplify it leads to the general surface defined by the matrices Q_0, Q_1, Q_2 in Equation 5

$$(\underline{x} - m_0)^\top \Sigma_0^{-1}(\underline{x} - m_0) = (\underline{x} - m_1)^\top \Sigma_1^{-1}(\underline{x} - m_1) \tag{4}$$

$$\underline{x}^\top Q_0 \underline{x} \ + \ Q_1 \underline{x} \ + \ Q_2 \underline{x} = 0 \tag{5}$$

Calculating the matrices Q_0, Q_1, Q_2 for the 20D data leads to:

$$Q_0 = \begin{pmatrix}
-0.000266 & -6.99e-06 & -1.53e-05 & 0.000101 & -4.5e-05 & 8.84e-05 & 1.76e-05 & 8.28e-05 & 5.97e-05 & 0.000269 \\
5.65e-05 & 8.23e-05 & 0.000155 & -7.57e-05 & 4.85e-05 & -4.87e-05 & 0.000104 & 4.33e-05 & -8.66e-06 & 4.49e-05 \\
-6.99e-06 & -0.000101 & 0.000334 & -0.000162 & 8.96e-05 & 0.000104 & -4.97e-05 & 0.000115 & 3.38e-05 & 0.000193 \\
-5.27e-05 & 6.58e-05 & 2.32e-05 & -0.000146 & -2.33e-05 & 3.47e-06 & -9.62e-06 & -5.7e-05 & 5.01e-06 & 2.26e-05 \\
-1.53e-05 & 0.000334 & -0.000552 & 0.000432 & -0.000167 & -0.000259 & 9.89e-05 & -0.000241 & -8.71e-05 & -0.000314 \\
7.05e-05 & -0.000134 & -6.6e-05 & 0.000262 & 1.78e-05 & -2.58e-05 & 6.56e-05 & 9.36e-05 & -4.44e-05 & -4.88e-06 \\
0.000101 & -0.000162 & 0.000432 & -0.000958 & 7.02e-06 & 0.000196 & -0.000128 & 8.17e-05 & 6.05e-05 & 0.000426 \\
-4.5e-05 & 0.000109 & 0.000183 & -6.73e-05 & -2.04e-05 & 5.46e-05 & -0.000125 & -0.000142 & 5.79e-05 & -0.000122 \\
-4.5e-05 & 8.96e-05 & -0.000167 & 7.02e-06 & -7.35e-05 & -0.000186 & -3.94e-05 & -8.51e-05 & 4.9e-06 & -2.71e-05 \\
-2.56e-05 & -1.82e-05 & 8.46e-05 & 9.11e-05 & -1.93e-06 & 3.03e-05 & 3.07e-06 & -2.43e-05 & 4.5e-06 & -2.14e-05 \\
8.84e-05 & 0.000104 & -0.000259 & 0.000196 & -0.000186 & -0.000178 & -0.000121 & -0.000199 & -4.94e-05 & -0.000265 \\
-5.27e-05 & -8.06e-05 & -6.79e-05 & 0.000104 & 9.58e-06 & 2.82e-05 & -5.75e-05 & 2.17e-05 & 1.9e-05 & -5.07e-05 \\
1.76e-05 & -4.97e-05 & 9.89e-05 & -0.000128 & -3.94e-05 & -0.000121 & -0.00015 & 3.18e-06 & -1.51e-05 & 6.77e-05 \\
-7.54e-05 & 4.98e-05 & 6.53e-05 & -5.05e-05 & 2.22e-06 & 1.5e-05 & -3.48e-05 & -8.13e-05 & 5.11e-05 & 5.83e-06 \\
8.28e-05 & 0.000115 & -0.000241 & 8.17e-05 & -8.51e-05 & -0.000199 & 3.18e-06 & -0.000347 & -0.000141 & -0.000226 \\
7.21e-07 & 2.5e-05 & -9.07e-05 & 0.00023 & -2.01e-05 & 8.27e-05 & -8.91e-05 & -7.17e-06 & 5.89e-05 & -2.74e-05 \\
5.97e-05 & 3.38e-05 & -8.71e-05 & 6.05e-05 & 4.9e-06 & -4.94e-05 & -1.51e-05 & -0.000141 & -0.000235 & -0.000188 \\
-1.17e-05 & 7.66e-06 & -8.02e-05 & 4.94e-05 & 4.79e-05 & -7.43e-05 & -0.000107 & -1.68e-05 & 2.49e-05 & -4.77e-06 \\
0.000269 & 0.000193 & -0.000314 & 0.000426 & -2.71e-05 & -0.000265 & 6.77e-05 & -0.000226 & -0.000188 & -0.000738 \\
-5.79e-05 & -0.000317 & -0.000341 & 0.0003 & -3.08e-05 & -5.37e-05 & -0.000153 & 2.17e-05 & 4.68e-05 & -1.18e-05 \\
5.65e-05 & -5.27e-05 & 7.05e-05 & -4.5e-05 & -2.56e-05 & -5.27e-05 & -7.54e-05 & 7.21e-07 & -1.17e-05 & -5.79e-05 \\
-0.000441 & -8.2e-06 & 0.000168 & -0.000207 & 4.34e-05 & 0.000142 & -0.000129 & -7.36e-05 & -3.03e-05 & -5.4e-05 \\
8.23e-05 & 6.58e-05 & -0.000134 & 0.000109 & -1.82e-05 & -8.06e-05 & 4.98e-05 & 2.5e-05 & 7.66e-06 & -0.000317 \\
-8.2e-06 & -7.8e-05 & 2.86e-05 & 0.000165 & -4.01e-05 & -8.11e-05 & -1.59e-05 & 4.7e-05 & -1.99e-05 & -4.32e-05 \\
0.000155 & 2.32e-05 & -6.6e-05 & 0.000183 & 8.46e-05 & -6.79e-05 & 6.53e-05 & -9.07e-05 & -8.02e-05 & -0.000341 \\
0.000168 & 2.86e-05 & -0.000587 & 0.000136 & -2.32e-05 & 8.61e-05 & -0.00027 & -8.99e-05 & 4.65e-05 & 0.000109 \\
-7.57e-05 & -0.000146 & 0.000262 & -6.73e-05 & 9.11e-05 & 0.000104 & -5.05e-05 & 0.00023 & 4.94e-05 & 0.0003 \\
-0.000207 & 0.000165 & 0.000136 & -0.000685 & -2.71e-05 & -2.98e-05 & 0.000322 & -0.000133 & 1.61e-06 & 5.39e-05 \\
4.85e-05 & -2.33e-05 & 1.78e-05 & -2.04e-05 & -1.93e-06 & 9.58e-06 & 2.22e-06 & -2.01e-05 & 4.79e-05 & -3.08e-05 \\
4.34e-05 & -4.01e-05 & -2.32e-05 & -2.71e-05 & -0.000173 & 8.98e-05 & -6.63e-05 & -8.86e-07 & 5.97e-05 & -6.19e-06 \\
-4.87e-05 & 3.47e-06 & -2.58e-05 & 5.46e-05 & 3.03e-05 & 2.82e-05 & 1.5e-05 & 8.27e-05 & -7.43e-05 & -5.37e-05 \\
0.000142 & -8.11e-05 & 8.61e-05 & -2.98e-05 & 8.98e-05 & -0.0003 & 9.55e-05 & 7.18e-05 & 2.8e-05 & 4.5e-05 \\
0.000104 & -9.62e-06 & 6.56e-05 & -0.000125 & 3.07e-06 & -5.75e-05 & -3.48e-05 & -8.91e-05 & -0.000107 & -0.000153 \\
-0.000129 & -1.59e-05 & -0.00027 & 0.000322 & -6.63e-05 & 9.55e-05 & -0.000667 & -0.000112 & 3.4e-05 & 3.94e-06 \\
4.33e-05 & -5.7e-05 & 9.36e-05 & -0.000142 & -2.43e-05 & 2.17e-05 & -8.13e-05 & -7.17e-06 & -1.68e-05 & 2.17e-05 \\
-7.36e-05 & 4.7e-05 & -8.99e-05 & -0.000133 & -8.86e-07 & 7.18e-05 & -0.000112 & -0.000176 & 5.97e-05 & -0.000113 \\
-8.66e-06 & 5.01e-06 & -4.44e-05 & 5.79e-05 & 4.5e-06 & 1.9e-05 & 5.11e-05 & 5.89e-05 & 2.49e-05 & 4.68e-05 \\
-3.03e-05 & -1.99e-05 & 4.65e-05 & 1.61e-06 & 5.97e-05 & 2.8e-05 & 3.4e-05 & 5.97e-05 & -7.78e-05 & 1.77e-05 \\
4.49e-05 & 2.26e-05 & -4.88e-06 & -0.000122 & -2.14e-05 & -5.07e-05 & 5.83e-06 & -2.74e-05 & -4.77e-06 & -1.18e-05 \\
-5.4e-05 & -4.32e-05 & 0.000109 & 5.39e-05 & -6.19e-06 & 4.5e-05 & 3.94e-06 & -0.000113 & 1.77e-05 & -0.000138
\end{pmatrix}$$

$$Q_1 = \begin{pmatrix} -0.5357 \\ -0.0406 \\ 0.0159 \\ 0.0943 \\ -0.0127 \\ 0.1328 \\ -0.0267 \\ 0.1171 \\ 0.1177 \\ 0.4602 \\ 0.0821 \\ 0.1547 \\ 0.2729 \\ -0.1605 \\ 0.0712 \\ -0.0751 \\ 0.1734 \\ 0.0597 \\ -0.0448 \\ 0.0894 \end{pmatrix}$$

$$Q_2 = [-192.8011]$$

2.2 Question 2

(10 pts) Use the test set of the two classes in the MNIST dataset, and make label predictions for all the test vectors for the two classes using the MED and GED classifiers. Plot the test set error for both your classifiers in two separate plots. The MED and GED classifiers were implemented and used on the 20D test set with the same equation used to calculate the kNN test set error. The resulting error and bar graph:

MED classifier has an error of 0.004728
GED classifier has an error of 0.165957

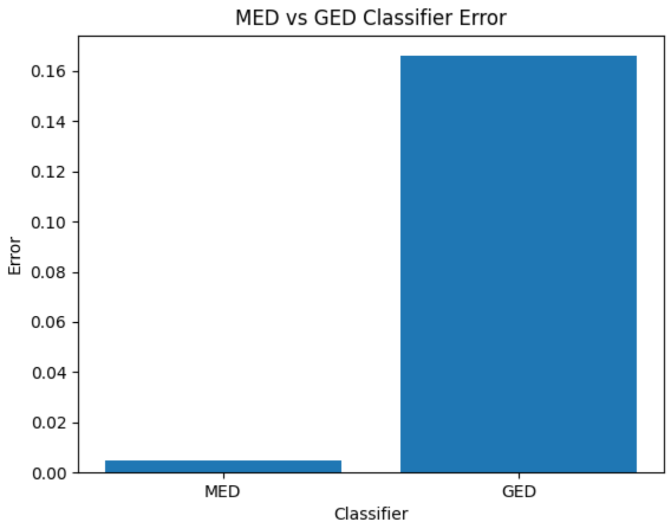

Figure 7: MED vs GED classifier error (20D test set)

2.3 Question 3

(5 pts) Which classifier is better? Why?

In the 20D space, it seems like the MED classifier is a much better with a test set error of 0.004728 versus the GED classifier with a test set error of 0.18156. This is a large discrepancy in error, and it a weird result because GED is supposed to correct for features that are correlated when compared to MED classifier. The GED classifier is supposed to transform each of the patterns relative to the statistics of the class and thus perform better. The experimental results contradict the advantage, and this could be because that GED classifiers usually do better at handling class distributions that can be modeled by Gaussian models. It could be that the classes in the 20D space have more complex class distributions. In this case the MED classifier is better just because the experimental results on this test set have much better performance, although the GED classifier may perform better in other cases.

2.4 Question 4

(15) Convert the training set images for the two classes in MNIST to 2×1 vectors, and plot the decision boundaries for the MED and GED classifiers. Do you think MED and GED classifiers are better than the kNN classifier? Explain.

For the MED Classifier in 2D space I applied Equation 2 with the following prototypes:

$$z_0 = \begin{bmatrix} 1023.98 & 17.22 \end{bmatrix}^\top$$

$$z_1 = \begin{bmatrix} -899.59 & -15.125 \end{bmatrix}^\top$$

Which resulted in the decision boundary:

$$1923.575x_1 + 32.341x_2 - 119671.235 = 0$$

Plotting the function and overlaying the training data results in the decision boundary in Figure 8

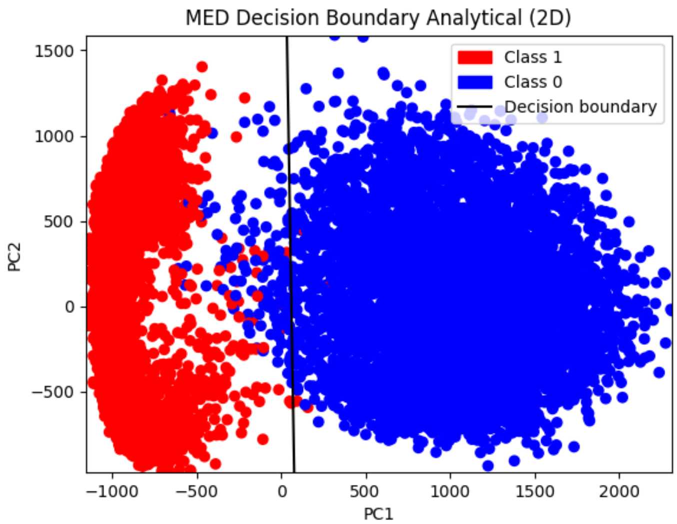

Figure 8: MED Classifier Decision Boundary

For the GED Classifier in 2D space I first computed the covariance matrix of class 0:

$$m_0 = \begin{bmatrix} 1023.98 & 17.22 \end{bmatrix}^\top, \quad \Sigma_0 = \begin{bmatrix} 215286.025 & -23351.239 \\ -23351.239 & 190498.533 \end{bmatrix}$$

I then applied the eigenvalue decomposition on Σ_0:

$$det \left(\begin{bmatrix} 215286.025 & -23351.239 \\ -23351.239 & 190498.533 \end{bmatrix} - \quad \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0$$

Solving above for λ using the quadratic formula yields:

$$\lambda_0 = 229328.72$$

$$\lambda_1 = 176455.84$$

Using the eigenvalues, the eigenvectors are determined:
$\lambda_0 = 229328.72$:

$$\begin{bmatrix} 215286.025 & -23351.239 \\ -23351.239 & 190498.533 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \quad = \quad 229328.72 \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

$$\phi_0 = \begin{bmatrix} 0.8569752 \\ -0.51536 \end{bmatrix}$$

$\lambda_1 = 176455.842$:

$$\begin{bmatrix} 22049.687 & -8583.773 \\ -8583.773 & 380576.598 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \quad = \quad 176455.84 \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

$$\phi_1 = \begin{bmatrix} 0.51536 \\ 0.856975 \end{bmatrix}$$

Combining ϕ_0 and ϕ_1 into a single matrix results in:

$$\phi = \begin{bmatrix} 0.8569752 & 0.515357 \\ -0.5153573897 & 0.8569753561 \end{bmatrix}$$

Placing the eigenvalues along the diagonal yields:

$$\Lambda_0 = \begin{bmatrix} 229328.7183 & 0 \\ 0 & 176455.8401 \end{bmatrix}$$

A weight matrix W can be calculated using Equation 6 and then the weight matrix can be used to determine the inverse covariance matrix in Equation 7

$$W = \Lambda_0^{-1/2}\phi^\top \tag{6}$$

$$\Sigma^{-1} = W^\top W \tag{7}$$

$$W_0 = \begin{bmatrix} 0.00208819 & 0 \\ 0 & 0.00238058 \end{bmatrix} \begin{bmatrix} 0.856975 & -0.515357 \\ 0.515357 & 0.8569753 \end{bmatrix} = \begin{bmatrix} 0.0017895 & -0.001076166 \\ 0.00122684 & 0.00204009 \end{bmatrix}$$

$$\Sigma_0^{-1} = \begin{bmatrix} 4.7076e-06 & 5.7705e-07 \\ 5.7705e-07 & 5.3201e-06 \end{bmatrix}$$

Similar to above, for class 1, Σ_1^{-1} is calculated. As steps are the same less detail is included:

$$m_1 = \begin{bmatrix} -899.59 & -15.125 \end{bmatrix}^\top, \ \ \Sigma_1 = \begin{bmatrix} 22049.69 & -8583.776 \\ -8583.776 & 380576.6 \end{bmatrix}$$

$$det\left(\begin{bmatrix} 22049.69 & -8583.776 \\ -8583.776 & 380576.6 \end{bmatrix} - \ \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

Solving above for λ using the quadratic formula yields:

$$\lambda_0 = 21844.294$$

$$\lambda_1 = 380782.01$$

$\lambda_0 = 21844.294$:

$$\begin{bmatrix} 22049.69 & -8583.776 \\ -8583.776 & 380576.6 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = 21844.294 \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

$$\phi_0 = \begin{bmatrix} -0.99971 \\ -0.023921 \end{bmatrix}$$

$\lambda_1 = 380782.01$:

$$\begin{bmatrix} 22049.69 & -8583.776 \\ -8583.776 & 380576.6 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = 380782.01 \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

$$\phi_1 = \begin{bmatrix} 0.023921 \\ -0.99971 \end{bmatrix}$$

Again using Equation 6 to calculate the W matrix, and then subbing W into Equation 7 it to find the inverse of covariance matrix leads to Σ_1^{-1}

$$W_1 = \begin{bmatrix} 0.006766 & 0 \\ 0 & 0.0016205 \end{bmatrix} \begin{bmatrix} -0.99971 & -0.023921 \\ 0.023921 & -0.99971 \end{bmatrix} = \begin{bmatrix} -0.006764 & -0.00016185 \\ 3.8765e-05 & -0.0016201 \end{bmatrix}$$

$$\Sigma_1^{-1} = \begin{bmatrix} 4.5754e-05 & 1.032e-06 \\ 1.032e-06 & 2.6509e-06 \end{bmatrix}$$

With both inverse covariance matrices I can use Equation 4 and then simplify to find the matrices Q_0, Q_1, Q_2 in Equation 5:

$$Q_0 = \begin{bmatrix} -4.104628e-05 & -4.549105e-07 \\ -4.549105e-07 & 2.669251e-06 \end{bmatrix}$$

$$Q_1 = \begin{bmatrix} -0.09201167 \\ -0.00330184 \end{bmatrix}$$

$$Q_0 = \begin{bmatrix}-32.0977269\end{bmatrix}$$

$$(-4.105e{-}05)x_1^2 - (9.01e{-}07)x_1x_2 + (2.67e{-}06)x_2^2 - 0.09201166x_1 - 0.00330184x_2 - 32.09772685 \ = \ 0$$

Plotting the equation above yields a quadratic decision boundary seen as the black curve in Figure 9

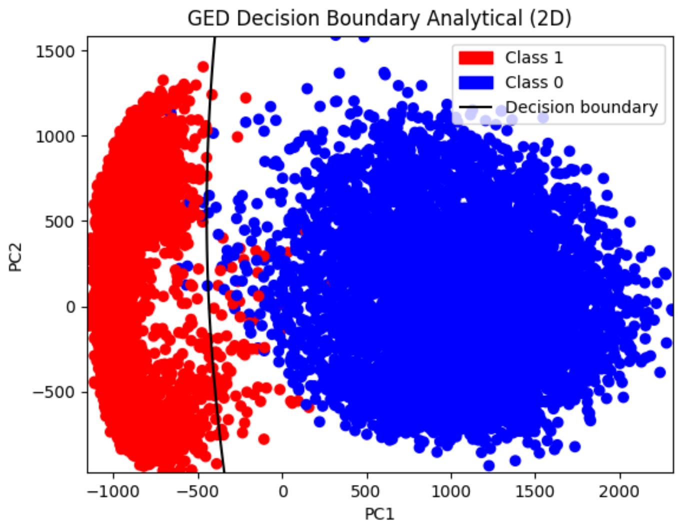

Figure 9: GED Classifier Decision Boundary

3 Question 5

(15 pts) Find the confusion matrices for the MED, GED, and kNN classifiers (k=1 and k=5). Find the confusion matrices using the test set of the two MNIST classes. Use the 2×1 vectors for the data.

Confusion matrices are an important tool to go past only looking at error. It allows us analyze the precision and recall of the classifier. It also is a sanity check to see if the test set is balanced, if not a poor binary classifier could end up with a very high accuracy if there were to be a big class imbalance and the classifier would always just predict the most common class.

Confusion matrices are in the format:

$$\begin{bmatrix}\text{True Positive} & \text{False Negative}\\ \text{False Positive} & \text{True Negative}\end{bmatrix}$$

$$\text{Confusion Matrix for MED Classifier: } \begin{bmatrix}968 & 12\\ 0 & 1135\end{bmatrix}$$

$$\text{Confusion Matrix for GED Classifier: } \begin{bmatrix}980 & 0\\ 7 & 1128\end{bmatrix}$$

$$\text{Confusion Matrix for kNN Classifier (k=1): } \begin{bmatrix}975 & 5\\ 3 & 1132\end{bmatrix}$$

$$\text{Confusion Matrix for kNN Classifier (k=5): } \begin{bmatrix}979 & 1\\ 4 & 1131\end{bmatrix}$$