

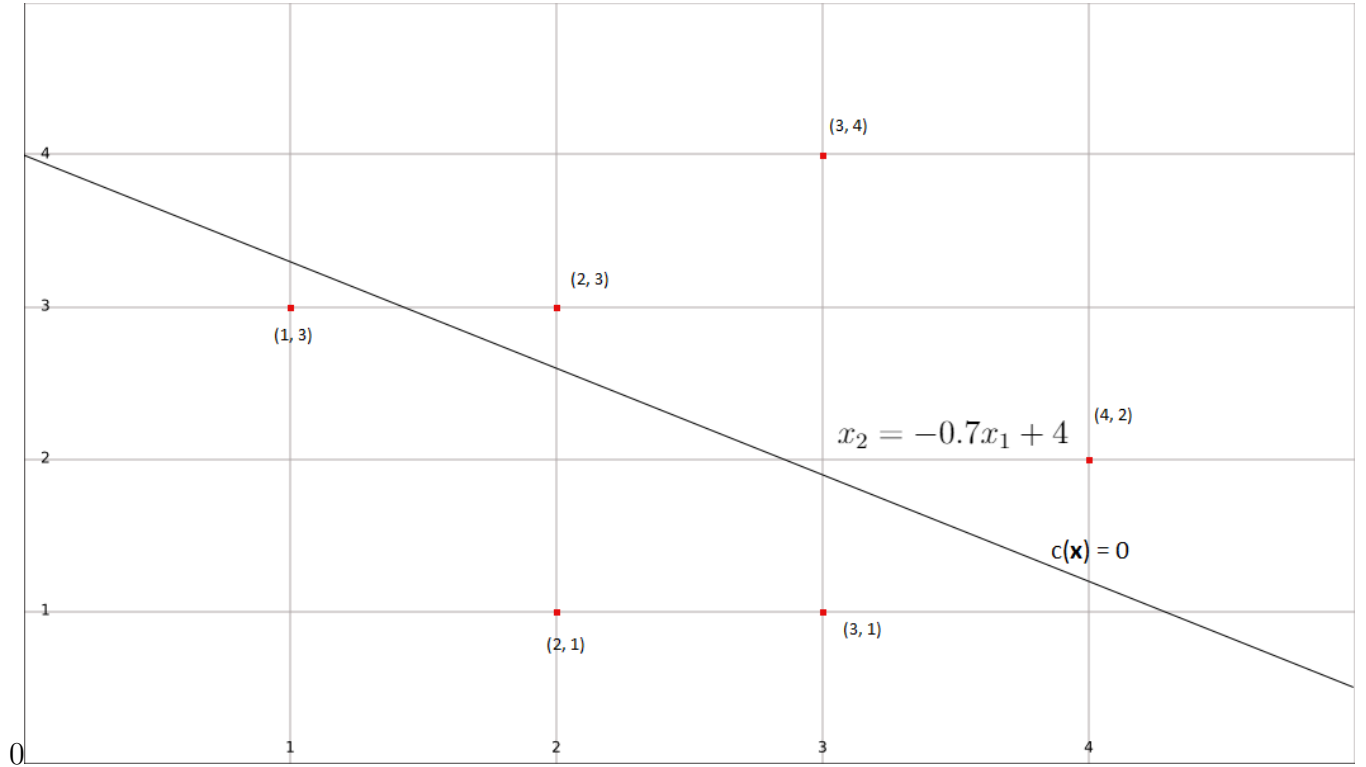
Portfolio Assignment 2

Candidate 25

Problem 1

(1a)

Figure 1: Decision boundary of a logistic discrimination classifier



In the plot above we see the points from the training set: $\{(\mathbf{x}^i, y^i)\}_{i=1}^6$ as red points. \mathbf{x}^i is the training data, and y^i is the ground truth. The line going somewhat diagonally across the plot is the decision boundary. This is used to classify the points above the line as $y^i = 0$, and points underneath the decision boundary as $y^i = 1$. This decision boundary is given by the equation:

$$C(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} + w_0 \quad (1)$$

Which put simply, assigns a class to an vector \mathbf{x}^i . If the vector after being put into $C(\mathbf{x}) > 0 \implies \mathbf{x} \rightarrow C^1$ or the opposite case where $C(\mathbf{x}) < 0 \implies \mathbf{x} \rightarrow C^2$. Which means at $C(\mathbf{x}) = 0$ we are on the decision boundary. From this we can derive the equation of the decision

boundary, $x_2 = -\frac{w_1}{w_2} \cdot x_1 - \frac{w_0}{w_2}$. In my guess for the decision boundary i used $w_1 = -0.7, w_2 = 1$ and $w_0 = 4$ to give the decision boundary of $x_2 = -0.7 \cdot x_1 + 4$. Here we see that the weights, \mathbf{w} of the classifier determines the slope of the decision boundary, whereas the bias, w_0 determines the intercept with the second axis, x_2 .

To figure out the distance between the origin, \mathbf{O} , and the decision boundary we can calculate this with the formula,

$$d = \frac{w_0}{\|\mathbf{w}\|} = \frac{4}{\sqrt{(-0.7)^2 + 1^2}} \approx \underline{\underline{3,277}}$$

Now i shall demonstrate how this decision boundary could give us a decision rule, based on the estimated weights and biases. Consider a test point,

$$\mathbf{x}^t = \begin{bmatrix} x_1^t \\ x_2^t \end{bmatrix}$$

this point will from what I stated earlier follow the decision rule,

$$\begin{cases} C(\mathbf{x}^t) > 0, & y^t = 0 \\ C(\mathbf{x}^t) \leq 0, & y^t = 1 \end{cases}$$

Here i have also made the assumption that if $C(\mathbf{x}^t) = 0$ then \mathbf{x}^t is assigned to $y^t = 1$, although in reality this is quite rare.

(1b)

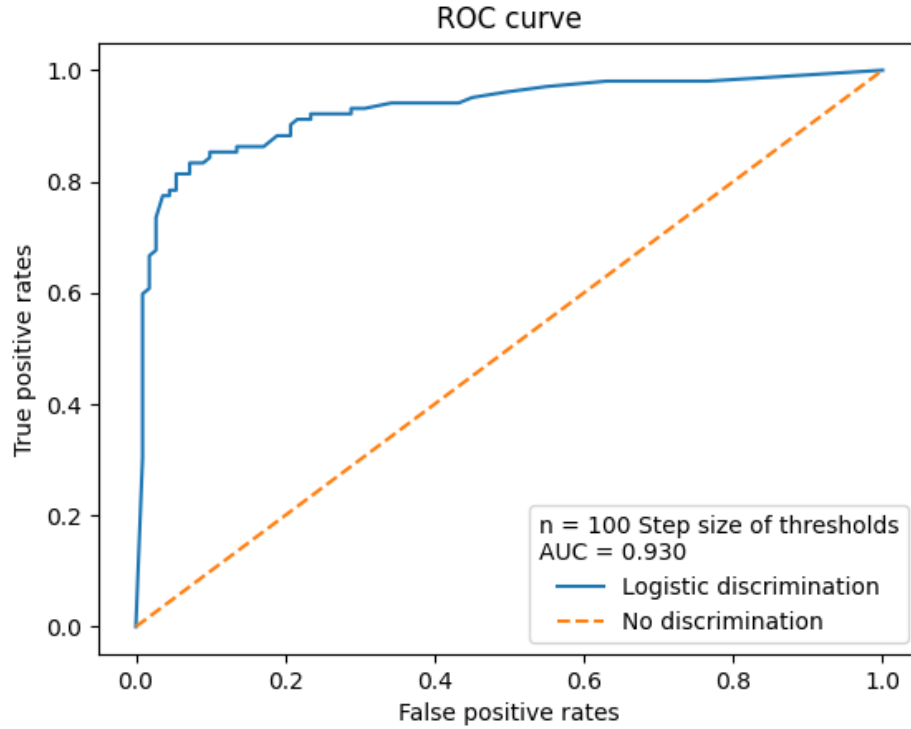
After having trained my logistic discrimination model on the *seals_train.csv* file, and the used the trained weights to classify the test data. I produce with the *seals_test.csv* file the confusion matrix:

$$\text{confusion matrix} = \begin{bmatrix} 87 & 15 \\ 11 & 100 \end{bmatrix}, \text{ and accuracy} \approx \underline{\underline{0.878}}$$

Which means the model classified 87 class 0's correctly, 11 class 1's as class 0's, 15 class 0's as class 1's and 100 class 1's correctly.

(1c)

Figure 2: The Receiver Operating Characteristic



The ROC curve represents

For the Area Under the Curve score i got $AUC \approx 0.930$ by using *sklearn.metrics* built-in function for calculating the AUC score.

(1d)

For this task I have created a function which classifies seal images corresponding to a given index, called *show_image*. Furthermore I use this function to show 5 random images which I know my classifier correctly classified in task (1b), and 5 random images I know are incorrectly classified.

Figure 3: Pictures of correctly classified seals

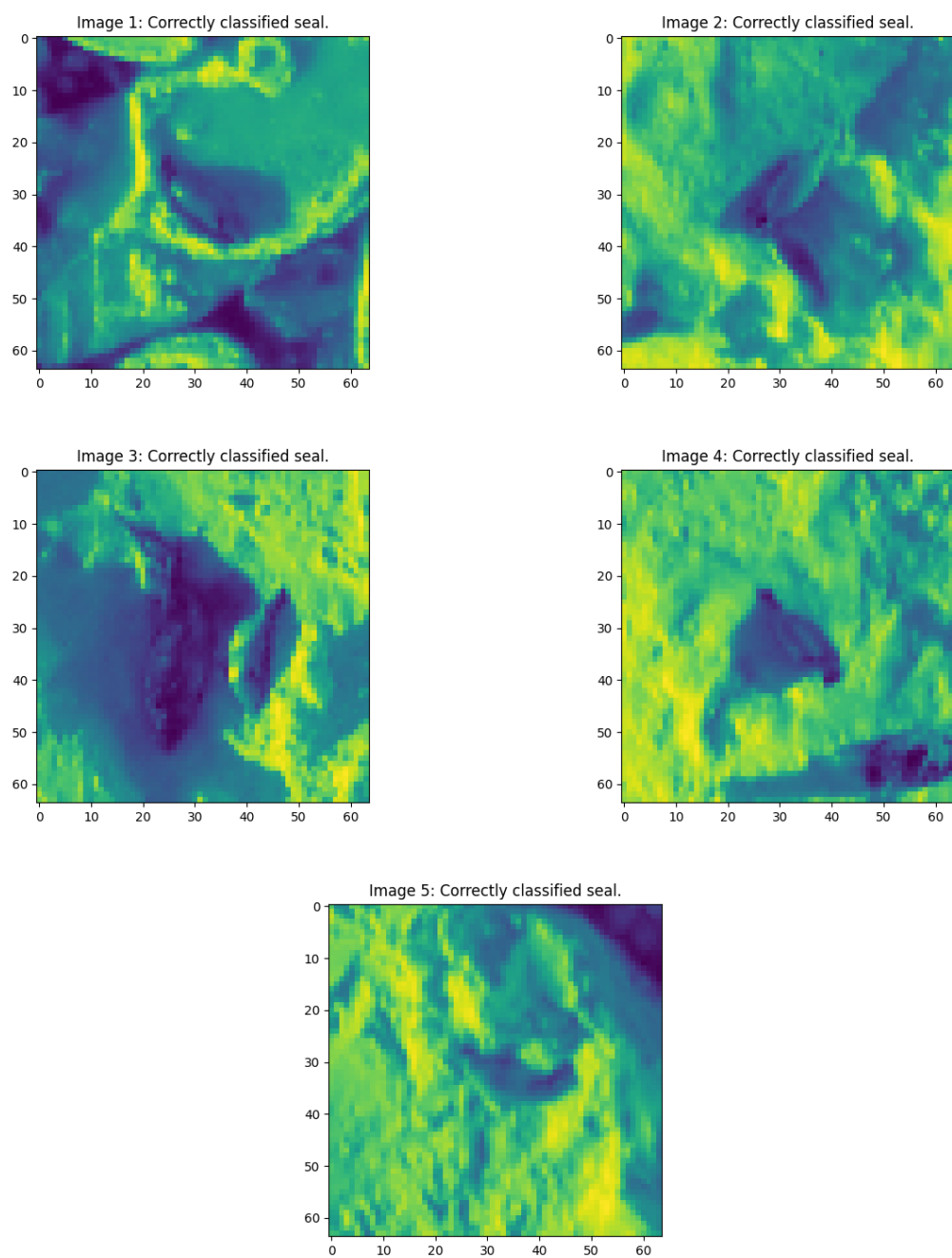
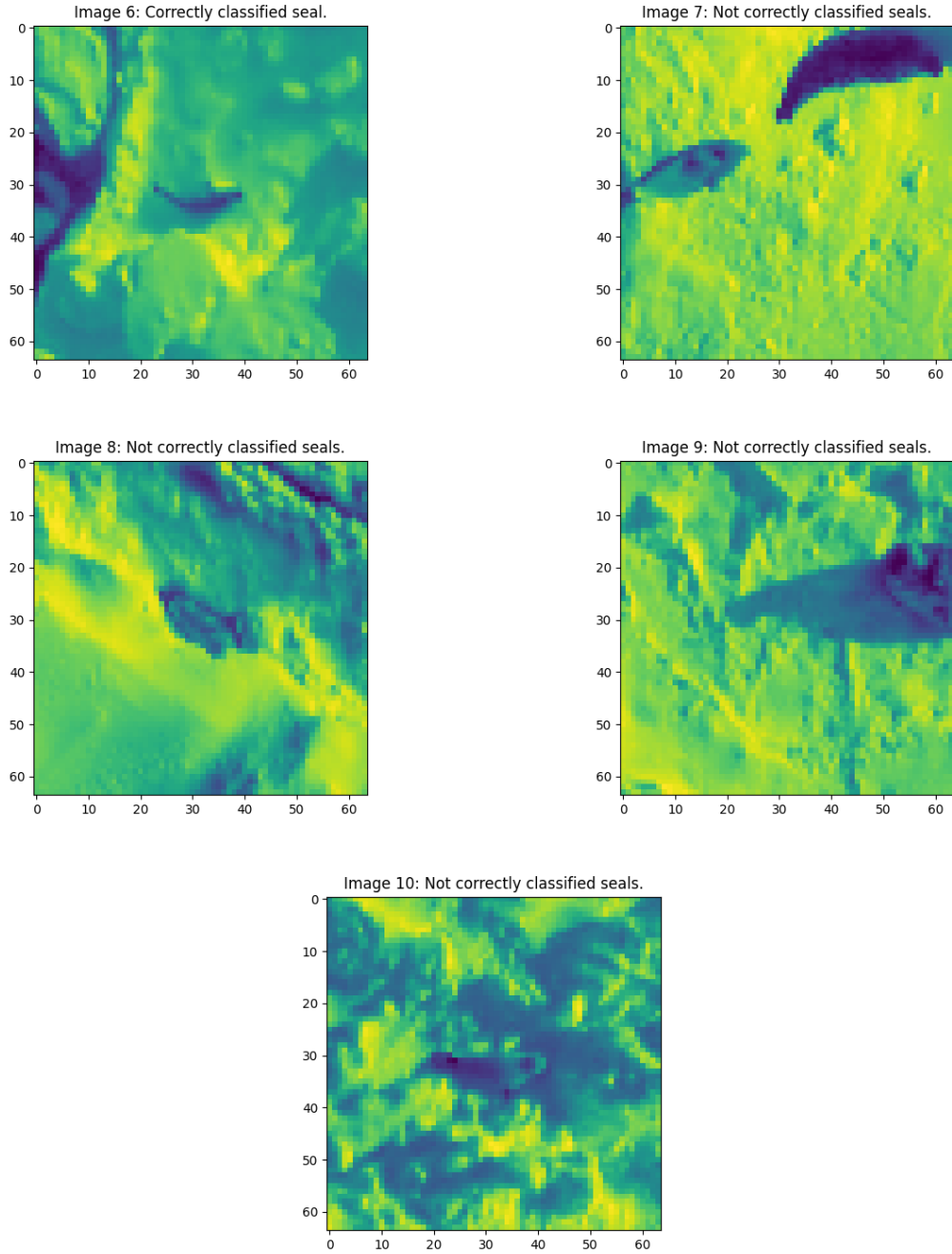


Figure 4: Pictures of not correctly classified seals



In Figure 3. the pictures are of either *Harp seal pups* or *Hooded seal pups*, where the algorithm has classified these seals correctly. Whereas in Figure 4. the pictures are of either *Harp seal pups* classified as *Hooded seal pups*, or *Hooded seal pups* classified as *Harp seal pups*.

This leads to a question, what are the possible reasons as to why the pictures in Figure 4. are misclassified. By studying these example photos I believe one reason to this comes from not the whole seal being present in the image. An example of this we see in Image 9. The seal is located between approximately pixel 50 to pixel 64 along the first axis, and pixels 10 to 30 along the second axis. Where not the whole seal is present, although we tell the classifier there is a seal in the image.

This leads to making a sort of educated guess, since not all of the features necessary to make a correct classification are present.

In image 8, it seems as though the lack of a good image resolution may be hiding a lot of the features the classifier needs to make a correct decision, and hence it misclassifies the seal.

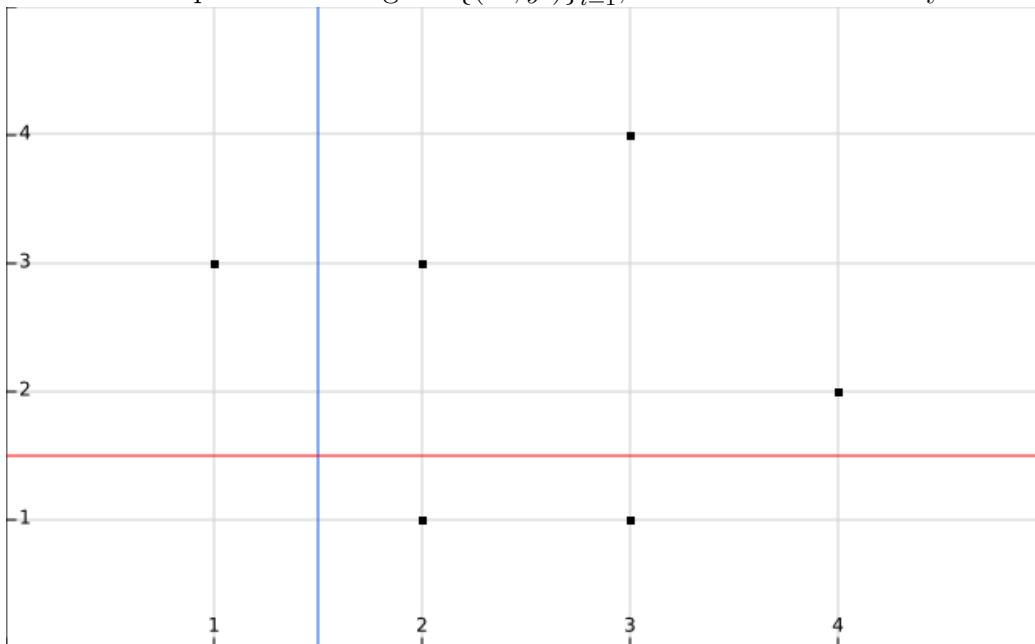
Image 10, demonstrates a noisy image where a lot of the features of the seal are hidden due to lots of darker areas, and very bright areas making it hard to pinpoint where the seal begins and ends.

In conclusion, the reason as to why the classifier fails to make a correct decision I believe arise when the image hides the distinct features of the seal.

Problem 2

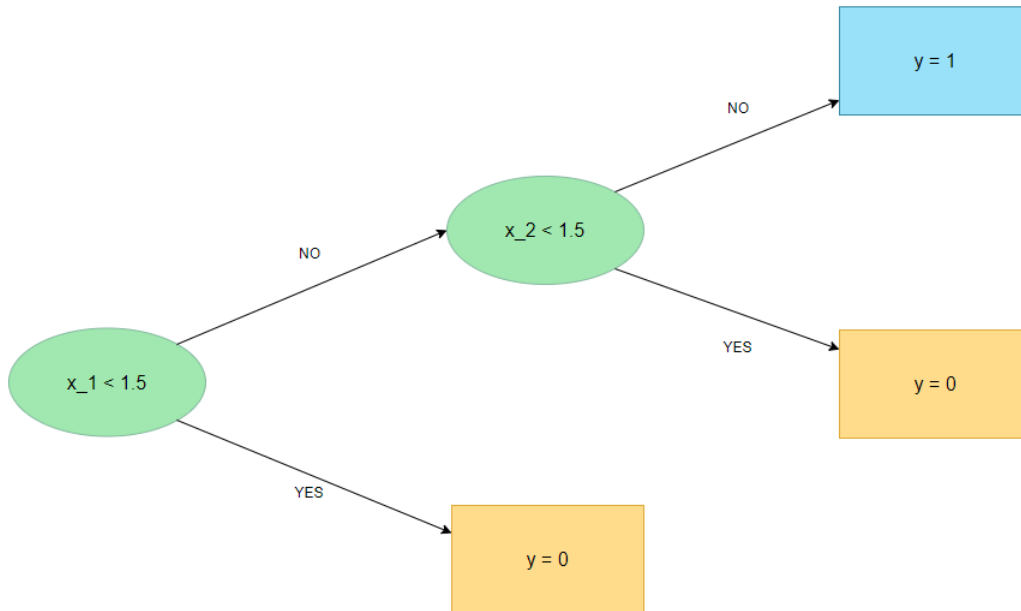
(2a)

Figure 5: 2-dimensional plot of training set $\{(\mathbf{x}^i, y^i)\}_{i=1}^6$, and decision boundary of a decision tree.



This plot illustrates the two decision boundaries needed to classify the training set with zero errors. Here I have used two boundaries. The first going from $x_1 = 1.5$ spanning the second axis, and the second going from $x_2 = 1.5$ spanning the first axis.

Figure 6: Decision tree for training set.



From this decision tree we see that we can create a IF-THEN rule which describes how the tree would classify the datapoint:

$$\mathbf{x}^t = \begin{bmatrix} x_1^t \\ x_2^t \end{bmatrix}$$

This means that,

IF: $x_1^t < 1.5$, THEN $\mathbf{x}^t \rightarrow y^t = 0$

IF: $x_1^t \geq 1.5$ AND $x_2^t < 1.5$ THEN $y^t = 0$

IF: $x_1^t \geq 1.5$ AND $x_2^t \geq 1.5$ THEN $y^t = 1$

Problem 3

(3a)

3a