Introduction to Intelligent Systems
Lab 2

Team 15:
Sara Bardají Serra and Albert Sallés Torruella

September 2019

Assignment 1

Let's assume we obtained the class conditional probability of salmon and sea bass, describing the distribution of the length of the two classes. These are given by p_salmon and p_seabass at length l (in cm), all in the file lab3_1.mat.

Firstly, let's see what data we're given:

```
    plot(l, p_salmon)
    hold on
    plot(l, p_seabass)
    title('Class conditional probability of salmon and seabass')
    legend('Salmon', 'Seabass')
    xlabel('Length (cm)')
    ylabel('PPD')
    hold off
```

Code from assignment 1

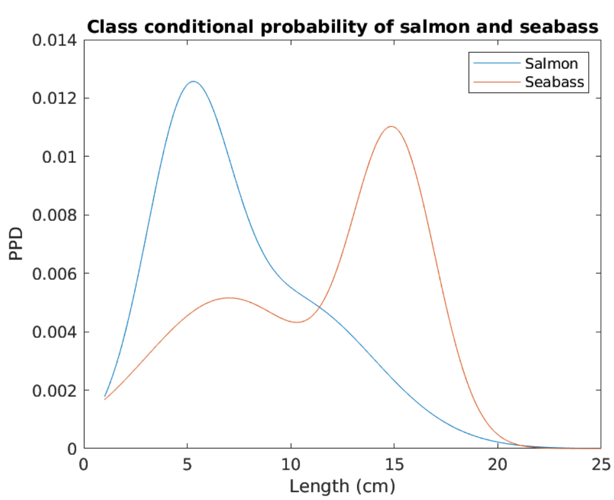

Figure 1: Graph showing our given data

So, what we're given is the class conditional probability density function of both salmon and sea bass, that means p(x—wi).

Exercise 1. Given that sea bass is caught 3 times as often as salmon, calculate the posterior probabilities (don't forget to normalize to 1 for all lengths) and plot them.

```
prior_salmon = 1/4;
prior_seabass = 3/4;

% Calculate the evidence (p(x))
evidence = prior_salmon*p_salmon + prior_seabass*p_seabass;

% Calculate the posterior distribution for both fishes

posterior_salmon = prior_salmon.*p_salmon./evidence;
posterior_seabass = prior_seabass.*p_seabass./evidence;

% Plot the posterior probability distributions
plot(l, posterior_salmon)
hold on
plot(l, posterior_seabass)
title('Posterior probability distribution of salmon and seabass')
legend("Salmon's PPD", "Seabass' PPD")
xlabel('Length (cm)')
ylabel('PPD')
```

Code from exercise 1

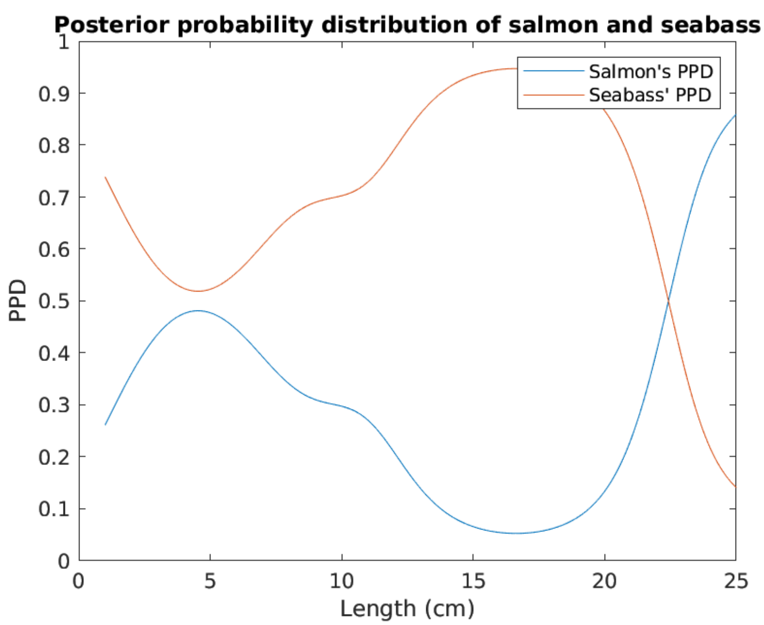

Figure 2: Plot of the posterior probabilities of our data

Exercise 2. Now a new fish is measured, which turns out to have a length of 8 cm. According to your posterior probabilities, how would you classify this fish? And what if it's length is 20 cm?

```matlab
        % 8cm fish
        % Add a line to the plot to have a graphic idea
        line([8, 8], ylim, 'LineWidth', 2, 'Color', 'y')
        % Find the index in the length array for 8cm
        index = find(l ==8);
        % Find the posterior probability for both salmon and seabass at 8 cm
        prob_salmon1 = posterior_salmon(index);
        prob_seabass1 = posterior_seabass(index);

        % 20cm fish
        % Add a line to the plot to have a graphic idea
        line([20, 20], ylim, 'LineWidth', 2, 'Color', 'g')
        legend("Salmon's PPD", "Seabass' PPD", "Criterion at 8cm length", "Criterion at 20cm length")
        % Find the index in the length array for 20cm
        index = find(l ==20);
        % Find the posterior probability for both salmon and seabass at 20 cm
        prob_salmon2 = posterior_salmon(index);
        prob_seabass2 = posterior_seabass(index);

        % Results
        fprintf('Case 8cm long fish:')
        fprintf('Posterior probability for salmon: %.4f', prob_salmon1)
        fprintf('Posterior probability for seabass: %.4f', prob_seabass1)
        fprintf('Case 20cm long fish:')
        fprintf('Posterior probability for salmon: %.4f', prob_salmon2)
        fprintf('Posterior probability for seabass: %.4f', prob_seabass2)
```

Code from exercise 2

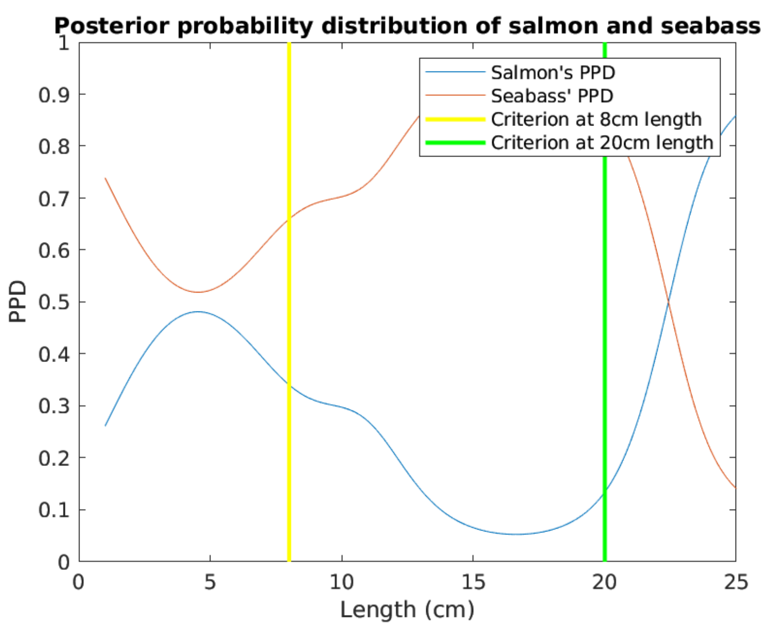

Figure 3: Plot of the posterior probabilities of our data with criteria on both fish's lengths

The output we obtained from exercise 2 is as follows:

```
Case 8cm long fish:
Posterior probability for salmon: 0.3403
Posterior probability for seabass: 0.6597
Case 20cm long fish:
Posterior probability for salmon: 0.1342
Posterior probability for seabass: 0.8658
```

For a fish 8cm long, given that the posterior probability for seabass is greater than the one for salmon, we would classify this fish as seabass. For a fish 20cm long, the posterior probability for seabass is also greater than the on for salmon, so we would also classify this fish as seabass

Assignment 2

Consider the following two sets of observed 1D data (feature values) coming from objects of two different classes (read the data from the file lab3_1.mat): S1, S2; and the following test set of data points that need to be classified: T.

Exercise 1. Plot the elements of the two sets S1 and S2 on the x-axis of a Fig. 1: the elements of S1 as blue circles (bo) and the elements of the set S2 as red circles (ro). Plot in the same Fig 1. the points of the test set T as black squares (ks).

```
ps1 = plot(S1, zeros(1,numel(S1)), 'bo');
hold on
ps2 = plot(S2, zeros(1,numel(S2)), 'ro');
pt = plot(T, zeros(1,numel(T)), 'ks');
title('Plot of elements of S1, S2 and T')
legend([ps1 ps2 pt],'Elements of S1', 'Elements of S2', 'Elements of T')
```

Code from exercise 1

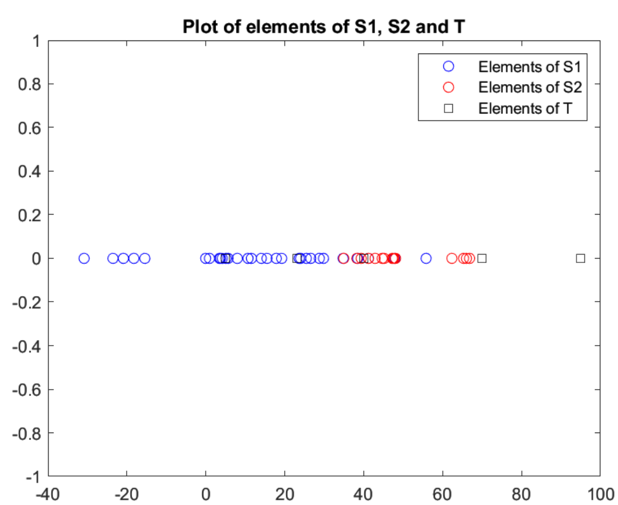

Figure 4: Plot of S1, S2 and T sets

Exercise 2. Let's assume S1 and S2 are drawn from normal distributions. Compute the parameters (mean and standard deviation) of the two distributions, using maximum estimation. Create a Fig.2 in which you plot the two Gaussian functions, one in blue the other in red, together with the points of the two training data sets S1 and S2. The two functions are the class conditional probability densities $p(x|\omega_1)$ and $p(x|\omega_2)$ that the two classes produce a value x of the considered feature. Comment on the use of the functions, in comparison to using the rugged data.

```
% Get the limits to establish a range
xlimits = xlim;
xrange = xlimits(1)-20:xlimits(2);
% Compute the mean and standard deviation
mean_S1 = mean(S1);
std_S1 = std(S1);
% Plot the normal distribution with the parameters previously computed
norm_S1 = normpdf(xrange, mean_S1, std_S1);
pns1 = plot(xrange, norm_S1);
% Do the same for set S2
mean_S2 = mean(S2);
std_S2 = std(S2);
norm_S2 = normpdf(xrange, mean_S2, std_S2);
pns2 = plot(xrange, norm_S2);

title('Normal distributions of sets S1 and S2')
legend([ps1 ps2 pt pns1 pns2], 'Elements of S1', 'Elements of S2', '
    Elements of T', 'Gaussian function of S1', 'Gaussian function of S2
    ', 'Location', 'northwest')
hold off
```

Code from exercise 2

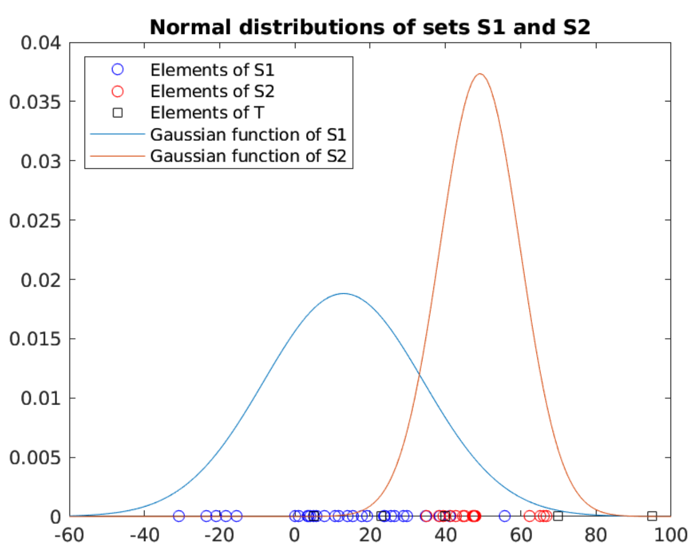

Figure 5: Plot of the normal distributions of S1 and S2 sets

We can use the function to calculate the probability of any point whether we have it in our data set or not.

Exercise 3. Estimate the prior probabilities $\mathrm{P}(\omega_1)$ and $\mathrm{P}(\omega_2)$.

```
    el_S1 = numel(S1);
    el_S2 = numel(S2);
    % Calculate the prior probabilities
    p_w1 = el_S1./(el_S1 + el_S2);
    p_w2 = el_S2./(el_S1 + el_S2);

    fprintf("The estimated probability for w1 is: %.4f", p_w1)
    fprintf("The estimated probability for w2 is: %.4f", p_w2)
```

Code from exercise 3

The output we obtained from exercise 3 is as follows:

```
The estimated probability for w1 is: 0.6667
The estimated probability for w2 is: 0.3333
```

Exercise 4. Create a Fig. 3 in which you plot the two products $P(\omega_1)p(x|\omega_1)$ and $P(\omega_2)p(x|\omega_2)$, together with the points of the two training data sets S1 and S2 and the test set T.

```
    % First plot the values of S1, S2, and T on the x-axis
    ps1 = plot(S1, zeros(1,numel(S1)), 'bo');
    hold on
    ps2 = plot(S2, zeros(1,numel(S2)), 'ro');
    pt = plot(T, zeros(1,numel(T)), 'ks');
    % Then plot the normal distributions times the prior probabilities
    p1 = p_w1.*norm_S1;
    p2 = p_w2.*norm_S2;
    pjs1 = plot(xrange, p1);
    pjs2 = plot(xrange, p2);
    title('Joint probability of sets S1 and S2')
    legend([ps1 ps2 pt pjs1 pjs2], 'Elements of S1', 'Elements of S2', '
       Elements of T', 'Joint probability S1', 'Joint probability S2', '
       Location', 'northwest')
    ylim([0 0.02])
    hold off
```

Code from exercise 4

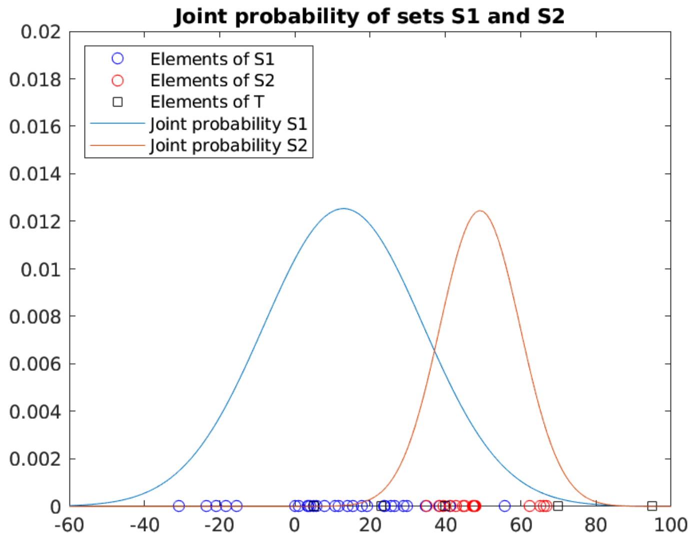

Figure 6: Plot of the joint probability of S1 and S2 sets

Exercise 5. Substitute the values of the prior probabilities and the class conditional probability densities determined above in the equation $P(\omega_1)p(x|omega_1) = P(\omega_2)p(x|\omega_2)$ and solve the resulting equation for x in order to determine the value(s) of the decision criterion that we should use for classication.

```
% Solve the equation P(w1)p(x|w1) = P(w2)p(x|w2)
sympref('FloatingPointOutput', true);
syms x
eqn = p_w1*(1/(std_S1*sqrt(2*pi)))*exp(-(x-mean_S1)^2/(2*std_S1^2)) == p_w2*(1/(std_S2*sqrt(2*pi)))*exp(-(x-mean_S2)^2/(2*std_S2^2));
xSol = solve(eqn, x);
% Obtain both solutions
x1 = xSol(1);
x2 = xSol(2);
% Plot them in the graph as a line
pdc1 = line([x1, x1], ylim, 'LineWidth', 2, 'Color', 'y');
pdc2 = line([x2, x2], ylim, 'LineWidth', 2, 'Color', 'y');
legend([ps1 ps2 pt pjs1 pjs2 pdc1], 'Elements of S1', 'Elements of S2', 'Elements of T', 'Joint probability S1', 'Joint probability S2', 'Decision criteria', 'Location', 'northwest')
ylim([0 0.02])

fprintf('First decision criterion: %.4f', x1)
fprintf('Second decision criterion: %.4f', x2)
```

Code from exercise 5

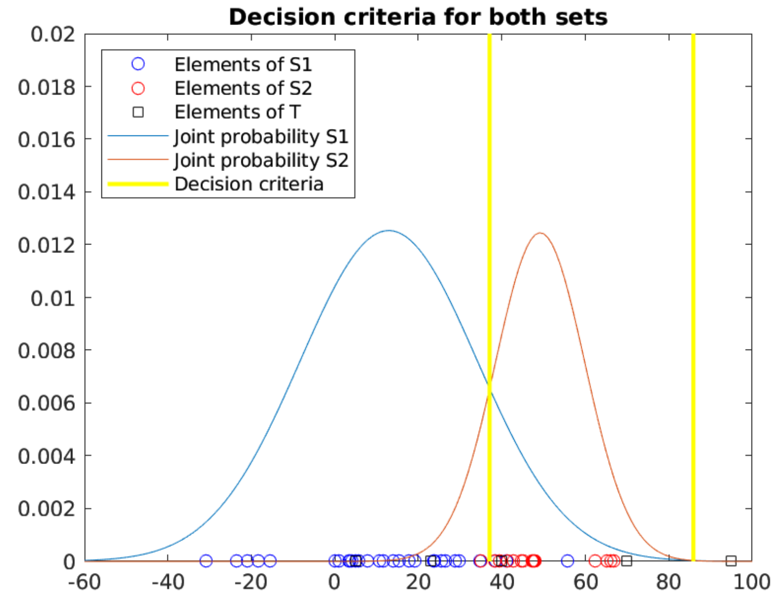

Figure 7: Plot with the decision criteria of both sets S1 and S2

The output we obtained from exercise 5 is as follows:

```
First decision criterion: 37.0697
Second decision criterion: 85.8863
```

Exercise 6. Using the thus obtained value(s) of the decision criterion, determine the classes of the points in the test set [5 23 40 70 95]. Create a Fig. 4 which copies Fig. 3 and adds to it a specification of which domain on the x-axis belongs to class 1 (blue) and which complementary domain corresponds to class 2 (red).

```
% Paint the areas separated by the decision criteria
ylimits = ylim;
hold on
areaS1 = area([xrange(1) x1], [ylimits(2) ylimits(2)], 'FaceColor', 'b', 'FaceAlpha', 0.3);
areaS2 = area([x1 x2], [ylimits(2) ylimits(2)], 'FaceColor', 'r', 'FaceAlpha', 0.3);
area([x1 xrange(end)], [ylimits(2) ylimits(2)], 'FaceColor', 'b', 'FaceAlpha', 0.3);
title('Areas for the decision criteria')
legend([ps1 ps2 pt pjs1 pjs2 pdc1 areaS1 areaS2], 'Elements of S1', 'Elements of S2', 'Elements of T', 'Joint probability S1', 'Joint probability S2', 'Decision criteria', 'Area S1', 'Area S2', 'Location', 'northwest')
hold off

% Classify the set T into both classes
set_S1 = T(T < x1 | x2 < T);
set_S2 = T(x1 < T & T < x2);

```

```
fprintf(strcat(['The points that belong to S1 are:', ' ', num2str(set_S1)]))
fprintf(strcat(['The points that belong to S2 are:', ' ', num2str(set_S2)]))
```

Code from exercise 6

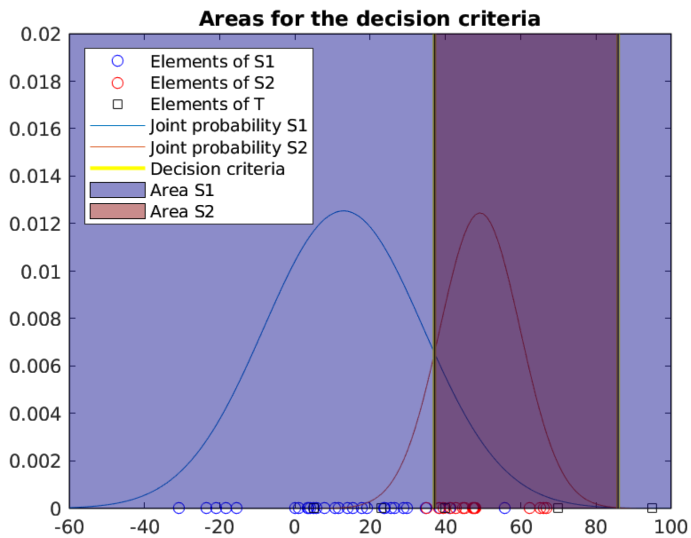

Figure 8: Plot of the decision criteria area for S1 and S2 sets

The output we obtained from exercise 6 is as follows:

```
The points that belong to S1 are: 5   23   95
The points that belong to S2 are: 40   70
```

Exercise 7. Evaluate the misclassification rate of the created classifier for each of the two classes. (For this, you can for instance use the error function erf, familiar from the iris recognition assignment in the first week.)

```
% Plot all the graphs previously computed
ps1 = plot(S1, zeros(1,numel(S1)), 'bo');
hold on
ps2 = plot(S2, zeros(1,numel(S2)), 'ro');
pt = plot(T, zeros(1,numel(T)), 'ks');
ylim([0 0.02])
p1 = p_w1.*norm_S1;
p2 = p_w2.*norm_S2;
pjs1 = plot(xrange, p1);
pjs2 = plot(xrange, p2);
pdc1 = line([x1, x1], ylim, 'LineWidth', 2, 'Color', 'k');
pdc2 = line([x2, x2], ylim, 'LineWidth', 2, 'Color', 'k');
title('Misclassification area for both sets')
% Paint the areas we want to calculate. To do this, we use the x-axis range
```

```
    % used to plot the graphs and take all the values that need to be painted,
    % then use the joint probability function to determine its Y value and
    % paint the area
    areaS2 = area(xrange(xrange < x1), p2(xrange(xrange < x1) + 61), 'FaceColor', 'r');
    areaS1 = area(xrange(x1 < xrange & xrange < x2), p1(xrange(x1 < xrange & xrange < x2) + 61), 'FaceColor', 'b');
    area(xrange(x2 < xrange), p2(xrange(x2 < xrange) + 61), 'FaceColor', 'r');
    legend([ps1 ps2 pt pjs1 pjs2 pdc1 areaS1 areaS2], 'Elements of S1', 'Elements of S2', 'Elements of T', 'Joint probability S1', 'Joint probability S2', 'Decision criteria', 'Error area S1', 'Error area S2', 'Location', 'northwest')
    hold off
    % Finally, calculate the areas using the CDF of the normal distributions
    % times the prior probabilities
    errArea_S1 = normcdf(x2, mean_S1, std_S1) - normcdf(x1, mean_S1, std_S1);
    errArea_S1 = errArea_S1 * p_w1;
    errArea_S2 = normcdf(x1, mean_S2, std_S2) + 1 - normcdf(x2, mean_S2, std_S2);
    errArea_S2 = errArea_S2 * p_w2;
    fprintf('S1 error is: %.4f', errArea_S1)
    fprintf('S2 error is: %.4f', errArea_S2)
```

Code from exercise 7

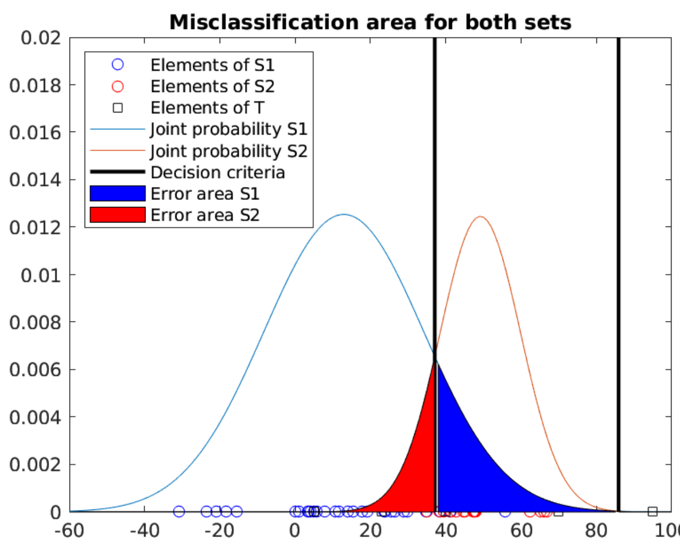

Figure 9: Plot of the misclassification area for S1 and S2 sets

The output we obtained from exercise 7 is as follows:

```
S1 error is: 0.0847
S2 error is: 0.0429
```

Assignment 3

Exercise 1. Let us consider two one-dimensional normal distributions with mean $m_1 = 1$ and $m_2 = -3$, with standard deviations $s_1 = s_2 = \sqrt{2}$. The two distributions have same prior probability $P(\omega_1) = P(\omega_2) = 0.5$. Compute the value of the decision criterion $x*$. Briefly comment on the approach that you take and report the main steps of the calculations.

```
% Given data
m1 = 1;
m2 = -3;
s1 = sqrt(2);
s2 = sqrt(2);
p_w1 = 0.5;
p_w2 = 0.5;

% Solve the equation P(w1)p(x|w1) = P(w2)p(x|w2)
sympref('FloatingPointOutput', true);
syms x
eqn = p_w1*(1/(s1*sqrt(2*pi)))*exp(-(x-m1)^2/(2*s1^2)) == p_w2*(1/(s2*sqrt(2*pi)))*exp(-(x-m2)^2/(2*s2^2));
xSol = solve(eqn, x);
fprintf("The value of the decision criterion x* is %i.", xSol);

% Let's make a visual example
% Compute normal distributions
xrange = -10:0.1:10;
norm1 = normpdf(xrange, m1, s1);
norm2 = normpdf(xrange, m2, s2);
% Compute joint probability of d1 and d2
p1 = norm1 * p_w1;
p2 = norm2 * p_w2;
% Plot the decision criterion in the graph as a line
plot_norm1 = plot(xrange, norm1);
hold on
plot_norm2 = plot(xrange, norm2);
pdc = line([xSol, xSol], ylim, 'LineWidth', 2, 'Color', 'y');
title('Plot of two normal distributions');
legend([plot_norm1, plot_norm2, pdc], 'Normal distribution 1', 'Normal distribution 2', 'Decision criterion');
hold off
```

Code from exercise 1

The output we obtained from exercise 1 is as follows:

```
The value of the decision criterion x* is -1.
```

We want to classify into two different classes using Bayes decision rule, so we know that given two normal distributions, we have to find the point in which both joint probabilities are equal, therefore, the solution to the equation $P(\omega_1)p(x|omega_1) = P(\omega_2)p(x|\omega_2)$. Below, we can see the example generated in the code.

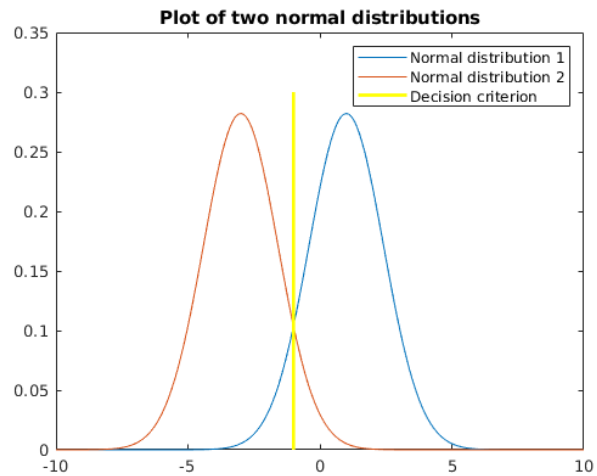

Figure 10: Plot of the visual example for exercise 1

Work division

During the realization of the assignments, the both of us have been working together and we've then divided the writing of the report equally. All the exercises have been discussed and checked by the both of us.