Group report for Assignment 1

Group number: **XX**
Name of the group members with student numbers:

- s0000000, first name last name
- s0000001, first name last name
- s0000002, first name last name
- s0000003, first name last name

General guidelines

Please add your answer to the places indicated. If there is no clear place indicated make sure that your answers are clear and structured according to the instructions in the assignment.

Task 1: Pairwise correlation coefficients

Maximum obtainable points: **05**

1. The correlation coefficients are shown in Table 1. To obtain the correlation coefficients, the following code was used:

```
corr_mat = [0 0 0; 0 0 0; 0 0 0];

for r = 1:3
    for c = 1:3
        corr_mat(r, c) = coeff(lab1_1(:,r), lab1_1(:,
            c));
    end
end

function corr_coeff = coeff(f1, f2)
    covar = sum((f1 - mean(f1)) .* (f2 - mean(f2)))/(
        length(f1) - 1);
    corr_coeff = covar/sqrt(var(f1) * var(f2));
end
```

2. Add the two scatter plots here.

   Conclusions:

   - Figure 1 shows the scatter plot of the height and body weight features. The figure displays an increase in height with an increase in body weight, indicating a positive linear correlation.

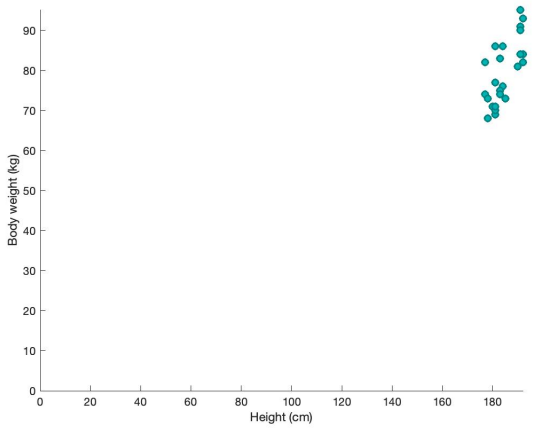

Figure 1: 2D scatter plot of Height (cm) and Body weight (kg) with corr = 0.7156.

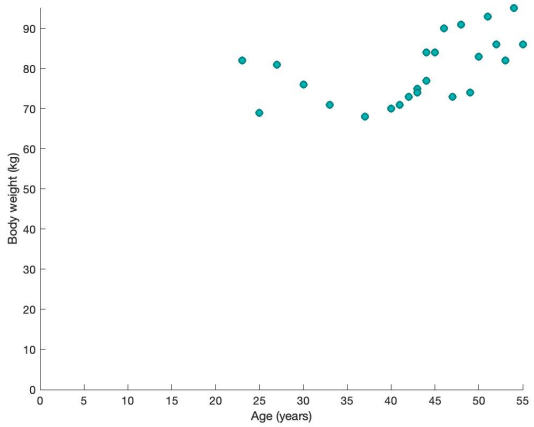

Figure 2: 2D scatter plot of age (years) and Body weight (kg) with corr = 0.5142.

	1	*2*	*3*
1	1	-0.0615	0.7156
2	-0.0615	1	0.5142
3	0.7156	0.5142	1

Table 1: Correlation coefficients.

- Figure 2 shows the scatter plot of the two features with the second largest correlation coefficient (age and body weight). The figure shows a decrease in body weight with an increase in age, after which it shows an increase in body weight with an increase in age. Therefore, the correlation between age and body weight features is non-linear.

Task 2: Hamming distance

Maximum obtainable points: **15**

1. To sample from the files, a string array is created containing all file names. To create the S group, a file is randomly sampled from the string array, from which two rows are randomly sampled without replacement.
   To create the D group, two files are randomly sampled without replacement, and a row is selected. The Hamming distance for both the S and D groups is computed with the `pdist` function.
   The code used to obtain the groups S and D is shown below.

```
load("Assignment1-files/Task_2/person01.mat");
path = "Assignment1-files/Task_2/";

%1
filenames = [];
for i = 1:20
    h = sprintf('person%02d.mat',i);
    filenames = [filenames; char(h)];
end

% a)
reps = 1000;
HD_S = zeros(1, reps);

for rep = 1:reps
    file = datasample(filenames, 1);
    load(path + file);
    y = datasample(1:20,2,'Replace',false);
    r1 = iriscode(y(1), :);
    r2 = iriscode(y(2), :);
    HD_S(1, rep) = pdist([r1;r2], 'hamming');
```

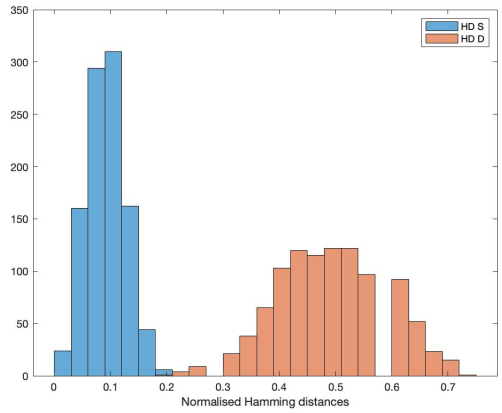

Figure 3: Histograms of S and D groups of Hamming distances.

```
end

% b)
HD_D = zeros(1, reps);

for rep = 1:reps
    files = datasample(filenames, 2, 'Replace', false
        );
    load(path + files(1, :));
    r1 = iriscode(datasample(1:20, 1), :);
    load(path + files(2, :));
    r2 = iriscode(datasample(1:20, 1), :);
    HD_D(1, rep) = pdist([r1;r2], 'hamming');
end
```

2. *How much do the two histograms overlap?*
   The histogram is displayed in Figure 3, which shows that the two histograms slightly overlap at the normalized Hamming distance 0.2.

3. The means and variances of groups S and D are displayed in Table 2. They were computed using Matlab's `mean` and `var` functions.

	S	D
Mean	0.0861	0.4889
Variance	0.0015	0.0092

Table 2:

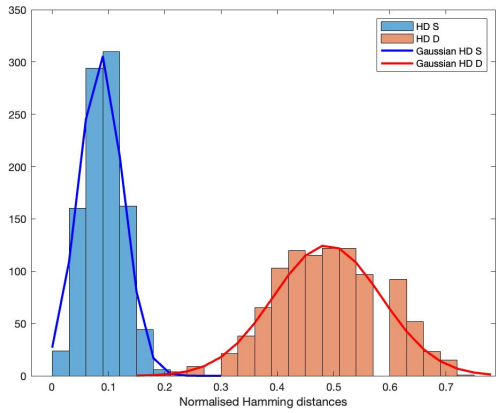

Figure 4: Gaussian distributions over histograms of S and D groups.

4. The histograms are shown in Figure 4. The Gaussian distributions were scaled by multiplying with the bin width and the number of samples so that the area under the curve would resemble that of the histograms. The Gaussian distributions were obtained through Matlab's `normpdf` function after calculating the standard deviations.

5. (a) Decision criterion = 0.19. The decision criterion was estimated by iterating over the values of the x-axis and obtaining the x-value for which the cumulative distribution function reached a value equal to or higher than the false acceptance error.

   (b) False rejection rate = 0.0038. The false rejection rate was computed through subtracting the cdf value at the decision criterion of group S from the cdf at the maximum hamming distance of group S.

6. Person 5. By taking the mean Hamming distances of the test person's iriscode to each other person's iriscodes using the mask, results in a minimum mean distance for person 5, making them the most likely to be the test person.

Task 3: Covariance matrix

Maximum obtainable points: **05**

1. Means: [5.8 5.0 6.2]
   Covariance matrix:

   The code used to compute the covariance matrix and the means is listed below.

3.2000	0.2500	-0.4500
0.2500	2.5000	-3.7500
-0.4500	-3.7500	5.7000

```
feat_vec = [4 5 6; 6 3 9; 8 7 3; 7 4 8; 4 6 5];

cov_mat = get_cov(feat_vec);

means = [0 0 0];
for i = 1:3
    means(i) = sum(feat_vec(:, i))/length(feat_vec(:,
        i));
end

mvnpdf([5 5 6], means, cov_mat)
mvnpdf([3 5 7], means, cov_mat)
mvnpdf([4 6.5 1], means, cov_mat)

function cov_mat = get_cov(feat_vec)
    cov_mat = [0 0 0; 0 0 0; 0 0 0];

    for r = 1:3
        f1 = feat_vec(:, r);
        mean_f1 = sum(f1)/length(f1);
        for c = 1:3
            f2 = feat_vec(:, c);
            mean_f2 = sum(f2)/length(f2);
            cov_mat(r, c) = sum((f1 - mean_f1) .* (f2
                - mean_f2))/(length(f1) - 1);
        end
    end
end
```

2. Probability densities of: [5 5 6] = 0.0543, [3 5 7] = 6.1287e-04 , [4 6.5 1] = 7.0300e-29.

Task 4: 2D Gaussian

Maximum obtainable points: **05**

1. The Gaussian distribution is displayed in Figure 5, for which the code below was used.

```
x = -10:.1:10;
```

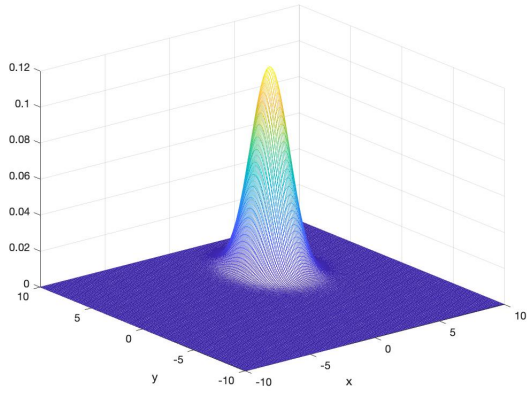

Figure 5: 3D plot of 2D Gaussian distribution.

```
[X1, X2] = meshgrid(x, x);
X = [X1(:) X2(:)];
sigma = [1 0; 0 2];
mean = [3 4];

gauss = mvnpdf(X, mean, sigma);
gauss = reshape(gauss,length(x),length(x));

mesh(-10:.1:10, -10:.1:10, gauss).
```

2. Please give the Mahalanobis distance between the points and the mean here. It is recommended to use code to calculate this, however you are also free to calculate it manually. Please include the code or the calculations as well.: [10 10]' = 8.1854, [0 0]' = 4.1231, [3 4]' = 0, [6 8]' = 4.1231

   The code with which the distances were computed is listed below.

```
points = [10 10; 0 0; 3 4; 6 8];

for i = 1:length(points)
    mahalanobis = mahal_dis(points(i,:)', mean', sigma
        );
end

function mahalanobis = mahal_dis(x, y, cov_mat)
     mahalanobis = sqrt((x - y)' * inv(cov_mat) * (x -
         y));
end
```

Task 5: Naive Bayesian rule

Maximum obtainable points: **05**

1. (a) No Spam. The probability of $P(Spam|Customers, Watches) = 1.350 * 10^{-6}$, and the probability of $P(NoSpam|Customers, Watches) = 1.400 * 10^{-5}$. Therefore, $P(NoSpam|Customers, Watches) > P(Spam|Customers, Watches)$.

   (b) Spam. The probability of $P(Spam|Fun, Vacation) = 3.375 * 10^{-8}$, and the probability of $P(NoSpam|Fun, Vacation) = 9.800 * 10^{-9}$. Therefore, $P(Spam|Fun, Vacation) > P(NoSpam|Fun, Vacation)$.

The probabilities were computed with the code below.

```
spam_prior = 0.9;
nospam_prior = 0.1;

customers_spam = 0.005;
customers_nospam = 0035;

watches_spam = 0.0003;
watches_nospam = 0.000004;

fun_spam = 0.00015;
fun_nospam = 0.0007;

vacation_spam = 0.00025;
vacation_nospam = 0.00014;

%a)
spam_prob = spam_prior * customers_spam *
    watches_spam
nospam_prob = nospam_prior * customers_nospam *
    watches_nospam

%b)
spam_prob = spam_prior * fun_spam * vacation_spam
nospam_prob = nospam_prior * fun_nospam *
    vacation_nospam
```

Task 6: Decision Tree

Maximum obtainable points: **05**

The decision tree is shown in Figure 6.

For both features, size and colour, the gini impurities were computed in order to determine the root node.

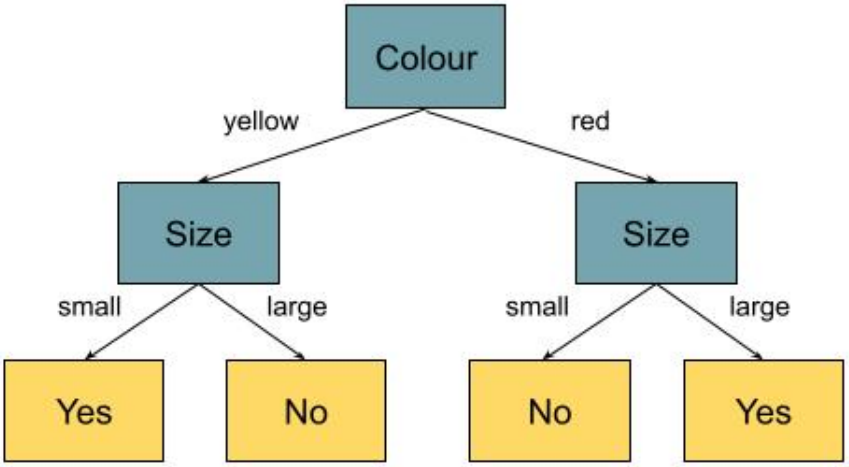

Figure 6: Decision tree best fitting the given data.

The size feature:

4 yes and 2 no corresponded to the small option, resulting in a gini impurity of $1 - (\frac{4}{6})^2 - (\frac{2}{6})^2 = \frac{4}{9}$.
1 yes and 1 no corresponded to the large option, resulting in a gini impurity of $1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = \frac{1}{2}$.

The total gini impurity is then:
$\frac{6}{8} \cdot \frac{4}{9} + \frac{2}{8} \cdot \frac{1}{2} = \frac{11}{24}$.

The colour feature:

4 yes and 1 no corresponded to the colour yellow, resulting in a gini impurity of $1 - (\frac{4}{5})^2 - (\frac{1}{5})^2 = \frac{8}{25}$.
1 yes and 2 no corresponded to the colour red, resulting in a gini impurity of $1 - (\frac{1}{3})^2 - (\frac{2}{3})^2 = \frac{4}{9}$.

The total gini impurity for the colour feature is then:
$\frac{5}{8} \cdot \frac{8}{25} + \frac{3}{8} \cdot \frac{4}{9} = \frac{11}{30}$.

Since $\frac{11}{30} < \frac{11}{24}$, the colour feature has the lowest total gini impurity, and thus is the root node of the decision tree.

In both the right and left branches are the nodes for size. For the left branch, the small option led to only yes, and the large option to only no, hence the nodes correspond to these findings in the data. Likewise the small decision for the size node in the right branch led to only no, and the large decision only yes.

Task 7: Learning Vector Quantization

Maximum obtainable points: **10**

1. Please add scatter plots of the datasets and your choice and (brief) reasoning for the number of prototypes.
2.
```
1 Add your implementation of LVQ here.
```

3. Please include 1 plot containing the error curves for the four cases and four scatter plots with the different prototypes here. Note that the scatter plots should contain the labels assigned by LVQ to all points.

Task 8: Cross-Validation

Maximum obtainable points: **05**

1.
```
Add your code for 10-fold cross validation here.
```

2. Test error:

```
Your code to compute the test error here (or
    alternatively include it in the code for the
    cross validation)
```

3. Please add the bar plot here.

Task 9: ROC

Maximum obtainable points: **05**

1. Your answer to part 1 here.
2. You answer to part 2 here.

Task 10: Transforms

Maximum obtainable points: **15**
Part 1:

Part 2:

Part 3:

Part 4:

Task 11: K-Nearest Neighbor

Maximum obtainable points: **10**

1.
```
Your implementation of k-nn here.
```

2. Please add scatter plots where the points are colored as the labels that K-nn (for k = 1, 3, 5, 7) assigns them here.
3. Please add your answer here.
4. Please add your answer here.

Task 12: K-means

Maximum obtainable points: **15**

1. Please add your answer to part 1.
2. Please add your answer to part 1.

Individual contributions

- who did what?
- or, How the tasks were divided?
- or, How the load was distributed?