Assignment 1

MIT 801

Connor McDonald
u16040725

Introduction To Machine And Statistical Learning

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
University of Pretoria
7 May 2021

1

1.1

$$Entropy(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

$$Entropy(S) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$$

Using (1) we calculate the following for attribute x_1:

$$\begin{aligned} S &= [2+, 3-] = 0.97 \\ S_+ &= [2+, 2-] = 1 \\ S_- &= [0+, 1-] = 0 \end{aligned}$$

$$Gain(S, A) = Entropy(S) - \sum_{v \epsilon Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2}$$

$$\begin{aligned} Gain(S, x_1) &= \mathrm{S} - \frac{4}{5}Entropy(S_+) - \frac{1}{5}Entropy(S_-) \\ &= 0.97 - \frac{4}{5}(1) - -\frac{1}{5}(0) \\ &= 0.17 \end{aligned}$$

Repeat for x_2:

$$\begin{aligned} S &= [2+, 3-] = 0.97 \\ S_+ &= [1+, 2-] = 0.92 \\ S_- &= [1+, 1-] = 1 \end{aligned}$$

$$\begin{aligned} Gain(S, x_2) &= \mathrm{S} - \frac{3}{5}Entropy(S_+) - \frac{2}{5}Entropy(S_-) \\ &= 0.97 - \frac{3}{5}(0.92) - -\frac{2}{5}(1) \\ &= 0.02 \end{aligned}$$

Repeat for x_3:

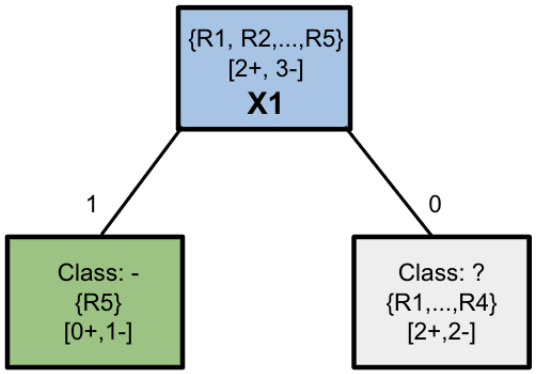
$$\begin{aligned} S &= [2+, 3-] = 0.97 \\ S_+ &= [1+, 2-] = 0.92 \\ S_- &= [1+, 1-] = 1 \end{aligned}$$

$$\begin{aligned} Gain(S, x_3) &= \mathrm{S} - \frac{3}{5} Entropy(S_+) - \frac{2}{5} Entropy(S_-) \\ &= 0.97 - \frac{3}{5}(0.92) - -\frac{2}{5}(1) \\ &= 0.02 \end{aligned}$$

$$\therefore x_1 \text{ has the highest information gain at } 0.17 \longrightarrow$$

1.2

To construct the decision tree we begin by identifying the attribute with the highest information gain, in this case it is x_1, this will be our root node. The start of the decision tree is drawn as follows:



Now that we have identified the root node we will be using the subset of the original data shown below to draw the rest of the tree.

Record	x_2	x_3	y
R1	0	0	+
R2	0	1	-
R3	1	0	-
R4	1	1	+

With this new subset we will calculate the information gain of the remaining attributes.

$$S_{sub} = [2+, 2-]$$
$$Entropy(S_{sub}) = -\frac{2}{2}\log_2\frac{2}{2} - \frac{2}{2}\log_2\frac{2}{2} = 1$$

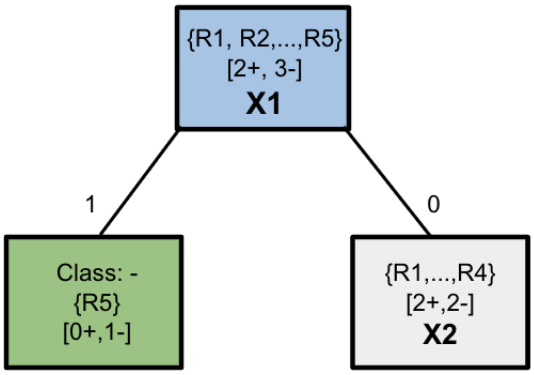
for x_2:

$$\begin{aligned} S_{sub+} &= [1+, 1-] = 1 \\ S_{sub-} &= [1+, 1-] = 1 \\ Gain(S_{sub}, x_2) &= S_{sub} - \frac{2}{4}Entropy(S_{sub+}) - \frac{2}{4}Entropy(S_{sub-}) \\ &= 1 - \frac{1}{2}(1) - -\frac{1}{2}(1) \\ &= 0 \end{aligned}$$

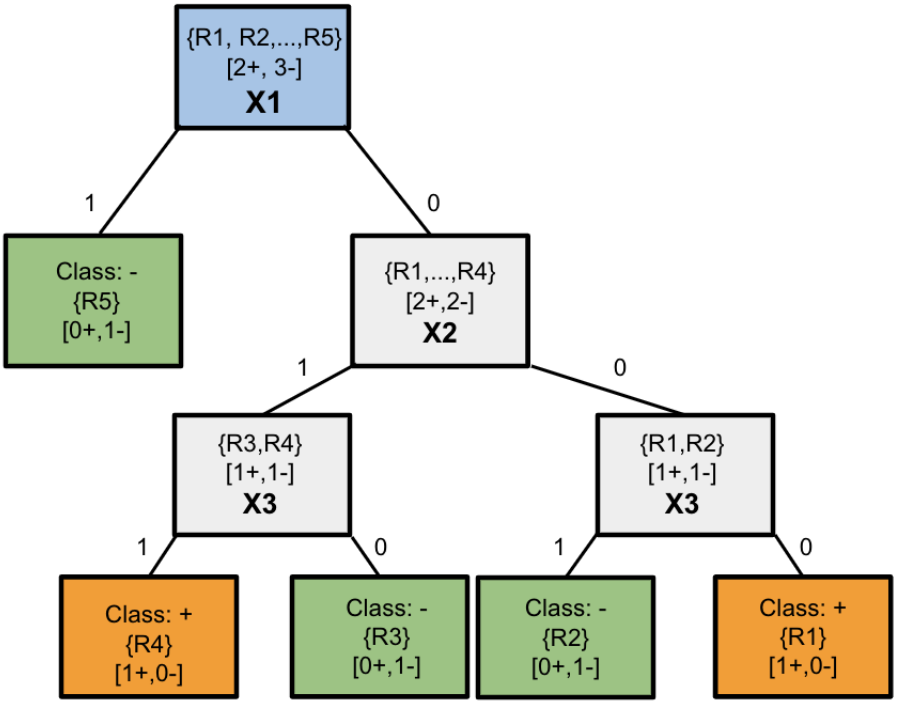
for x_3:

$$\begin{aligned} S_{sub+} &= [1+, 1-] = 1 \\ S_{sub-} &= [1+, 1-] = 1 \\ Gain(S_{sub}, x_3) &= S_{sub} - \frac{2}{4}Entropy(S_{sub+}) - \frac{2}{4}Entropy(S_{sub-}) \\ &= 1 - \frac{1}{2}(1) - -\frac{1}{2}(1) \\ &= 0 \end{aligned}$$

Since neither attribute provides any further information gain we will choose x_2 as our next node for the sake of numerical order:



x_3 is the only remaining attribute, and since it is a binary attribute we can draw the remainder of the tree with the blue box as the root node, grey boxes as decision nodes, green boxes as the negative class and orange boxes as the positive class.



1.3

No, we cannot construct a decision tree with 100% accuracy. Since x_1 was proved to have the highest information gain in 1.1, our model uses this attribute as the root node. If we look at R5 specifically, it is observed that this is the only row where the observation equals 1 for the x_1 attribute, this means that in 100% of the cases where the value 1 is observed in the x_1 attribute, the target variable is observed to be negative. Since the model was trained on this data, it will likely predict the target variable to be negative when the value 1 is observed in the x_1 attribute of the validation set.

Looking at R1 of the validation set, we see that the value observed for x_1 is 1, which would lead the model to predict the target variable as negative. However, the observed value of the target variable is positive, which would lead to an inaccurate prediction by the model.

This is a case of overfitting likely caused by an insufficent amount of training data for the model.

2

2.1

Table 1:

x_1	x_2
2.20	2.20
4.23	3.64
1.91	2.07
2.19	2.12
1.92	1.82
1.67	1.49
3.84	3.89
2.34	1.66
2.34	1.78
1.63	1.86

$$\vec{\mu} = \frac{\sum_{i=1}^{N} \boldsymbol{x}_i}{N} \tag{3}$$

Using equation (3), we can calculate the mean vector $\vec{\mu}$ as follows:

$$\begin{aligned} \vec{\mu}_1 &= \frac{\sum \boldsymbol{x}_1}{N} \\ &= \frac{24.27}{10} \\ &= 2.427 \end{aligned} \qquad \begin{aligned} \vec{\mu}_2 &= \frac{\sum \boldsymbol{x}_2}{N} \\ &= \frac{22.53}{10} \\ &= 2.253 \end{aligned}$$

$$\therefore \vec{\mu} = \begin{bmatrix} 2.43 & 2.25 \end{bmatrix} \longrightarrow$$

To calculate the covariance matrix ($\boldsymbol{\Sigma}$) we will use equation (4).

$$\boldsymbol{\Sigma} = \frac{\sum_{i=1}^{N} (\boldsymbol{x}_i - \vec{\mu}) \times (\boldsymbol{x}_i - \vec{\mu})^T}{N - 1} \tag{4}$$

$$(\boldsymbol{x}_i - \vec{\mu}) = \begin{bmatrix} -0.23 & 1.80 & -0.52 & -0.24 & -0.51 & -0.76 & 1.41 & -0.09 & -0.09 & -0.80 \\ -0.05 & 1.39 & -0.18 & -0.13 & -0.43 & -0.76 & 1.64 & -0.59 & -0.47 & -0.39 \end{bmatrix}$$

$$\Rightarrow (\boldsymbol{x}_i - \vec{\mu})^T = \begin{bmatrix} -0.23 & -0.05 \\ 1.80 & 1.39 \\ -0.52 & -0.18 \\ -0.24 & -0.13 \\ -0.51 & -0.43 \\ -0.76 & -0.76 \\ 1.41 & 1.64 \\ -0.09 & -0.59 \\ -0.09 & -0.47 \\ -0.80 & -0.39 \end{bmatrix}$$

$$\frac{(\boldsymbol{x}_i - \vec{\mu}) \times (\boldsymbol{x}_i - \vec{\mu})^T}{N-1} = \frac{\begin{bmatrix} 7.10 & 6.16 \\ 6.16 & 6.16 \end{bmatrix}}{10-1} = \begin{bmatrix} 0.79 & 0.68 \\ 0.68 & 0.68 \end{bmatrix}$$

$$\therefore \text{Covariance Matrix } (\boldsymbol{\Sigma}) = \begin{bmatrix} 0.79 & 0.68 \\ 0.68 & 0.68 \end{bmatrix} \longrightarrow$$

Next we will find the Principal Component Matrix $\mathbf{P}$ by finding the eigenvalues (λ_1, λ_2) and the corresponding unit eigenvectors $[\vec{p_1}, \vec{p_2}]$.

$$\boldsymbol{\Sigma} - \lambda \mathbf{I} = \begin{bmatrix} 0.79 - \lambda & 0.68 \\ 0.68 & 0.68 - \lambda \end{bmatrix}$$

Let $(\boldsymbol{\Sigma} - \lambda \mathbf{I}) = 0$ and solve for λ:

$$\boldsymbol{\Sigma} - \lambda \mathbf{I} = \begin{vmatrix} 0.79 - \lambda & 0.68 \\ 0.68 & 0.68 - \lambda \end{vmatrix} = \lambda^2 - 1.47\lambda + 0.0748$$

$$\lambda^2 - 1.47\lambda + 0.0748 = 0$$

$$\therefore \lambda_1 = 1.42, \quad \lambda_2 = 0.05$$

Let $(\mathbf{\Sigma}-\lambda_1\mathbf{I}) = \mathbf{B}$ and solve $\mathbf{B}\vec{v} = \bar{0}$, where $\vec{v}$ is the eigenvector of the respective eigenvalue(λ_1):

$$\begin{bmatrix} 0.79-\lambda_1 & 0.68 \\ 0.68 & 0.68-\lambda_1 \end{bmatrix} = \begin{bmatrix} -0.63 & 0.68 \\ 0.68 & -0.74 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} -0.63 & 0.68 \\ 0.68 & -0.74 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow -0.63v_1 + 0.68v_2 = 0$$

$$v_1 = \frac{68v_2}{63} = t$$

$$\text{Let t} = 1\text{:} \Rightarrow \vec{v} = \begin{bmatrix} 1 \\ \frac{68}{63} \end{bmatrix}$$

$$\text{Compute the unit vector: } \begin{bmatrix} \frac{1}{\sqrt{1^2+(\frac{68}{63})^2}} \\ \frac{\frac{68}{63}}{\sqrt{1^2+(\frac{68}{63})^2}} \end{bmatrix} = \begin{bmatrix} 0.73 \\ 0.68 \end{bmatrix}$$

$$\therefore \text{The eigenvector corresponding to the eigenvalue } 1.42 = \begin{bmatrix} 0.73 \\ 0.68 \end{bmatrix} \quad = \vec{p_1}$$

Repeat for λ_2: Let $(\mathbf{\Sigma} - \lambda_2\mathbf{I}) = \mathbf{B}$ and solve $\mathbf{B}\vec{v} = \bar{0}$, where $\vec{v}$ is the eigenvector of the respective eigenvalue(λ_2):

$$\begin{bmatrix} 0.79-\lambda_2 & 0.68 \\ 0.68 & 0.68-\lambda_2 \end{bmatrix} = \begin{bmatrix} 0.74 & 0.68 \\ 0.68 & 0.63 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0.74 & 0.68 \\ 0.68 & 0.63 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow 0.74v_1 + 0.68v_2 = 0$$

$$v_2 = -\frac{68v_1}{74} = t$$

$$\text{Let t} = 1\text{:} \Rightarrow \vec{v} = \begin{bmatrix} -\frac{68}{74} \\ 1 \end{bmatrix}$$

$$\text{Compute the unit vector: } \begin{bmatrix} \frac{-\frac{68}{74}}{\sqrt{1^2+(-\frac{68}{74})^2}} \\ \frac{1}{\sqrt{1^2+(\frac{68}{63})^2}} \end{bmatrix} = \begin{bmatrix} -0.68 \\ 0.73 \end{bmatrix}$$

$$\therefore \text{ The eigenvector corresponding to the eigenvalue } 0.05 = \begin{bmatrix} -0.68 \\ 0.73 \end{bmatrix} \quad = \vec{p_2}$$

$$\therefore \text{ The Principal Component Matrix } \mathbf{P} = \begin{bmatrix} \vec{p_1} & \vec{p_2} \end{bmatrix} = \begin{bmatrix} 0.73 & -0.68 \\ 0.68 & 0.73 \end{bmatrix} \longrightarrow$$

Note: One of the properties of eigenvectors is that they are ambiguous in sign, which implies that the opposite sign of these eigenvectors are also valid eigenvectors for eigenvalues $\lambda = 1.42$ and $\lambda = 0.05$. Since this is a hypothetical question, I have reversed the sign of my $\vec{p_1}$ vector to yield mainly positive components in my $\vec{\omega_1}$ projection coefficient and therefore resulting in 8 'Normal' and 2 'Anomaly' classes.

To find the projection coefficients $\boldsymbol{\omega} = [\omega_1, \omega_2]$. We use equation (5) shown below.

$$\vec{\omega_i} = [(\boldsymbol{x} - \vec{\mu})^T \vec{p_i}] \tag{5}$$

$$\vec{w_1} = \begin{bmatrix} -0.23 & -0.05 \\ 1.80 & 1.39 \\ -0.52 & -0.18 \\ -0.24 & -0.13 \\ -0.51 & -0.43 \\ -0.76 & -0.76 \\ 1.41 & 1.64 \\ -0.09 & -0.59 \\ -0.09 & -0.47 \\ -0.80 & -0.39 \end{bmatrix} \times \begin{bmatrix} -0.73 \\ -0.68 \end{bmatrix}$$

$$\vec{w_1} = \begin{bmatrix} 0.20 \\ -2.27 \\ 0.50 \\ 0.26 \\ 0.67 \\ 1.07 \\ -2.15 \\ 0.47 \\ 0.39 \\ 0.85 \end{bmatrix}$$

Repeat this process for $\vec{\omega_2}$:

$$\vec{w_2} = \begin{bmatrix} -0.23 & -0.05 \\ 1.80 & 1.39 \\ -0.52 & -0.18 \\ -0.24 & -0.13 \\ -0.51 & -0.43 \\ -0.76 & -0.76 \\ 1.41 & 1.64 \\ -0.09 & -0.59 \\ -0.09 & -0.47 \\ -0.80 & -0.39 \end{bmatrix} \times \begin{bmatrix} -0.68 \\ 0.73 \end{bmatrix}$$

$$\vec{w_2} = \begin{bmatrix} 0.12 \\ -0.21 \\ 0.22 \\ 0.06 \\ 0.02 \\ -0.05 \\ 0.24 \\ -0.38 \\ -0.29 \\ 0.25 \end{bmatrix}$$

The question defines the class of each record as: all records with projection coefficients on the first principal component (or the principal component with the largest eigenvalue) greater than or equal to 0 are from the 'normal' class, otherwise they are from the 'anomaly' class.

Using this definition we arrive at the final answer shown in the table below

Record#	$\vec{\omega_1}$	$\vec{\omega_2}$	class
1	0.20	0.12	Normal
2	-2.27	-0.21	Anomaly
3	0.50	0.22	Normal
4	0.26	0.06	Normal
5	0.67	0.02	Normal
6	1.07	-0.05	Normal
7	-2.15	0.24	Anomaly
8	0.47	-0.38	Normal
9	0.39	-0.29	Normal
10	0.85	0.25	Normal

Code results for question 2.1

```
Mean Vector:  [2.427, 2.253]

Covariance Matrix:
[[0.78920111 0.68389889]
 [0.68389889 0.68411222]]

Eigenvalues:  [1.4225711  0.05074223]

Eigenvectors for eigenvalue = 1.42:  [0.733691   0.67948328]

Eigenvectors for eigenvalue = 0.05:  [-0.67948328  0.733691  ]

Principal Component Matrix(P):
 [[ 0.733691   -0.67948328]
 [ 0.67948328  0.733691  ]]

Projection Coefficients (using negative sign of p1):
          w1        w2    Class
0  0.202560  0.115357   Normal
1 -2.265288 -0.207479  Anomaly
2  0.503664  0.217027   Normal
3  0.264256  0.063457   Normal
4  0.666198  0.026810   Normal
5  1.073850 -0.045437   Normal
6 -2.149020  0.240942  Anomaly
7  0.466765 -0.375964   Normal
8  0.385227 -0.287921   Normal
9  0.851789  0.253208   Normal
```

2.2

2.2.1

For the first part of this question we will make use of equation (3) and equation (4) to compute $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ used to represent the multivariate Gaussian distribution of the 'Normal' class.

x_1	x_2
2.20	2.20
1.91	2.07
2.19	2.12
1.92	1.82
1.67	1.49
2.34	1.66
2.34	1.78
1.63	1.86

$$\vec{\mu}_1 = \frac{\sum \boldsymbol{x}_1}{N} = \frac{16.20}{8} = 2.025 \qquad \vec{\mu}_2 = \frac{\sum \boldsymbol{x}_2}{N} = \frac{15.00}{8} = 1.875$$

$$\xrightarrow{\therefore \vec{\mu} = \begin{bmatrix} 2.03 & 1.88 \end{bmatrix}}$$

$$(\boldsymbol{x}_i - \vec{\mu}) = \begin{bmatrix} 0.18 & -0.12 & 0.17 & -0.10 & -0.36 & 0.31 & 0.31 & -0.40 \\ 0.33 & 0.19 & 0.25 & -0.05 & -0.38 & -0.22 & -0.09 & -0.01 \end{bmatrix}$$

$$\Rightarrow (\boldsymbol{x}_i - \vec{\mu})^T = \begin{bmatrix} 0.18 & 0.33 \\ -0.12 & 0.19 \\ 0.17 & 0.25 \\ -0.10 & -0.05 \\ -0.36 & -0.38 \\ 0.31 & -0.22 \\ 0.31 & -0.09 \\ -0.40 & -0.01 \end{bmatrix}$$

$$\frac{(\boldsymbol{x}_i - \vec{\mu}) \times (\boldsymbol{x}_i - \vec{\mu})^T}{N-1} = \frac{\begin{bmatrix} 0.56 & 0.13 \\ 0.13 & 0.41 \end{bmatrix}}{8-1} = \begin{bmatrix} 0.08 & 0.02 \\ 0.02 & 0.06 \end{bmatrix}$$

$$\therefore (\mathbf{\Sigma}) = \begin{bmatrix} 0.08 & 0.02 \\ 0.02 & 0.06 \end{bmatrix} \longrightarrow$$

2.2.2

To compute the approximation to $\mathrm{p}(\mathbf{x}^{(t)})$ we will use the formula shown below in (6), where $\boldsymbol{x}^{(t)}$ represents each record in the original dataset shown in table 1.

$$p(\boldsymbol{x}^{(t)}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_i - \vec{\mu})^T\right) \tag{6}$$

First, we calculate the determinant and inverse of Σ.

$$|\Sigma| = (0.08 \times 0.06) - (0.02 \times 0.02) = 0.0044$$
$$\Sigma^{-1} = \begin{bmatrix} 0.08 & 0.02 \\ 0.02 & 0.06 \end{bmatrix}^{-1} = \begin{bmatrix} 13.63 & -4.54 \\ -4.54 & 18.18 \end{bmatrix}$$

Next we will calculate the constant term of equation (6).

$$\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} = \frac{1}{(2\pi)^{\frac{2}{2}}0.0044^{\frac{1}{2}}} = 2.40$$

Now we will calculate the variable term of equation (6) for each record in table 1.

$$R_1 = -\frac{1}{2}(\boldsymbol{x}_1 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_1 - \vec{\mu})^T = -0.94$$
$$R_2 = -\frac{1}{2}(\boldsymbol{x}_2 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_2 - \vec{\mu})^T = -45.07$$
$$R_3 = -\frac{1}{2}(\boldsymbol{x}_3 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_3 - \vec{\mu})^T = -0.53$$
$$R_4 = -\frac{1}{2}(\boldsymbol{x}_4 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_4 - \vec{\mu})^T = -0.57$$
$$R_5 = -\frac{1}{2}(\boldsymbol{x}_5 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_5 - \vec{\mu})^T = -0.08$$
$$R_6 = -\frac{1}{2}(\boldsymbol{x}_6 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_6 - \vec{\mu})^T = -1.64$$
$$R_7 = -\frac{1}{2}(\boldsymbol{x}_7 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_7 - \vec{\mu})^T = -44.22$$

$$R_8 = -\frac{1}{2}(\boldsymbol{x}_8 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_8 - \vec{\mu})^T = -1.36$$
$$R_9 = -\frac{1}{2}(\boldsymbol{x}_9 - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_9 - \vec{\mu})^T = -0.87$$
$$R_{10} = -\frac{1}{2}(\boldsymbol{x}_{10} - \vec{\mu})\Sigma^{-1}(\boldsymbol{x}_{10} - \vec{\mu})^T = -1.02$$

Lastly, we multiply the constant term by the exponential of the variable term to arrive at the following approximation results for each record:

$$p(\boldsymbol{x}^{(1)}) = 2.40 \exp(-0.94) = 0.94$$
$$p(\boldsymbol{x}^{(2)}) = 2.40 \exp(-45.07) \approx 0$$
$$p(\boldsymbol{x}^{(3)}) = 2.40 \exp(-0.53) = 1.41$$
$$p(\boldsymbol{x}^{(4)}) = 2.40 \exp(-0.57) = 1.36$$
$$p(\boldsymbol{x}^{(5)}) = 2.40 \exp(-0.08) = 2.22$$
$$p(\boldsymbol{x}^{(6)}) = 2.40 \exp(-1.64) = 0.47$$
$$p(\boldsymbol{x}^{(7)}) = 2.40 \exp(-44.22) \approx 0$$
$$p(\boldsymbol{x}^{(8)}) = 2.40 \exp(-1.36) = 0.62$$
$$p(\boldsymbol{x}^{(9)}) = 2.40 \exp(-0.87) = 1.01$$
$$p(\boldsymbol{x}^{(10)}) = 2.40 \exp(-1.02) = 0.87$$

2.2.3

Using an ϵ value of 0.5, and classifying anomalies according to $p(\boldsymbol{x}^{(t)}) < \epsilon$, we can classify the records as follows:

Record	x_1	x_2	$p(\boldsymbol{x}^{(t)})$	Class
1	2.20	2.20	0.94	Normal
2	4.23	3.64	0	Anomaly
3	1.91	2.07	1.41	Normal
4	2.19	2.12	1.36	Normal
5	1.92	1.82	2.22	Normal
6	1.67	1.49	0.47	Anomaly
7	3.84	3.89	0	Anomaly
8	2.34	1.66	0.62	Normal
9	2.34	1.78	1.01	Normal
10	1.63	1.86	0.87	Normal

We can plot the approximation with the μ and Σ parameters calculated above as a 3D surface and a contour map to get a better understanding of our results. 3 points fall in the darkest region of the plot and are coloured white, these are the 'anomalies'. The red points all fall above the darkest region, these are the 'normal' classes. This supports the choice of using the negative eigenvector in question (2.1) because if the positive eigenvector was used instead our kernel would've been centered around the two anomalies in the top right of figure 2.

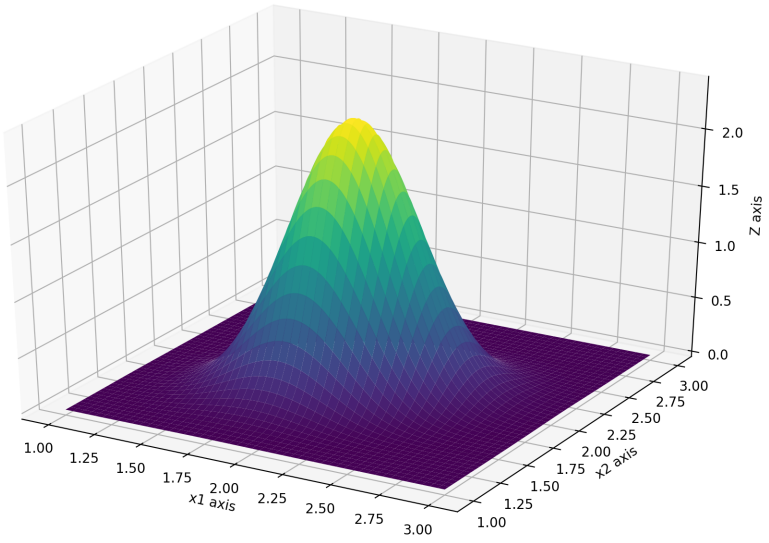


Figure 1: 3D Plot Of Multivariate Gaussian Distribution

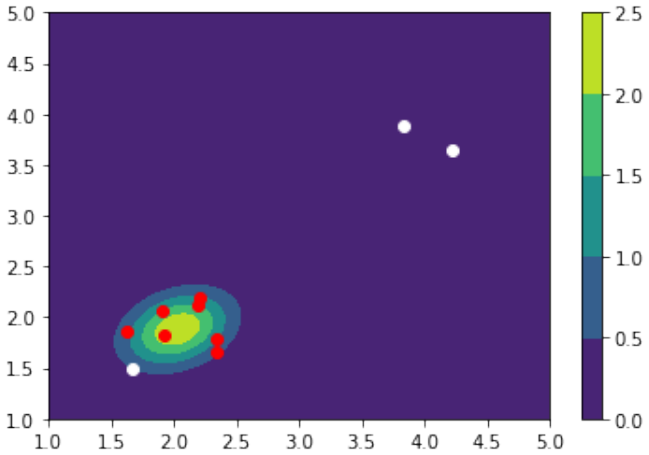


Figure 2: Contour Map Of Multivariate Gaussian Distribution

Code results for question 2.2

```
Question 2.2.1
Mean Vector:  [2.025, 1.875]
Covariance Matrix:
 [[0.08037143 0.01794286]
 [0.01794286 0.05862857]]


Question 2.2.2
Determinant of Sigma:
 0.004390115918367348

Inverse of Sigma:
 [[13.35467503 -4.08710327]
 [-4.08710327 18.30735909]]

Constant Term: 2.4020505230155385

Variable terms:
-0.9388968647008912
-45.07486259253626
-0.5280297442752655
-0.5660189787187994
-0.07770450531546556
-1.6397107709675087
-44.21519443631444
-1.3624867210981555
-0.8674773382395927
-1.0196750766843223

P(xt) approximations:
0.9393434478056931
6.379961255353182e-20
1.4166472848463374
1.3638393562105182
2.222467914984923
0.46608464775823216
1.5071871329339137e-19
0.614980964348691
1.0088846552452468
0.8664487483068845

Classes:
Normal
Anomaly
Normal
Normal
Normal
Anomaly
Anomaly
Normal
Normal
Normal
```

3

3.1

- Clustering algorithms are a form of unsupervised supervised learning, typically used to identify natural grouping within data. Clustering algorithms can be used to segment data into groups such as the market segmentation analysis, which is often used in the retail field, furthermore, clustering algorithms can even be used in outlier detection such as credit card fraud.

- We can optimize the number of clusters used in the K-Means algorithm by calculating and plotting the Within Cluster Sum of Squares (WCSS), this value will tend towards 0 as the number of clusters approaches the number of points, when we plot the WCSS we can identify the 'Elbow' which is when the change in WCSS between two numbers of clusters sharply reduces, this is identified as the optimal number of clusters, as anymore than this will have negligible improvement on the accuracy. A visualisation of this method is shown in Figure 3.

  Another popular method is the Average Silhouette method which measures the quality of clustering, the goal of this algorithm is to maximise the silhouette width by iterating over different values of k (number of clusters), the value of k used to achieve the maximum silhouette width corresponds to the optimal number of clusters.

- I would define clustering as categorizing data points into groups based on similarities between the points. Some examples of clustering algorithms are, agglomerative hierarchical clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and the Gaussian Mixture Model algorithm.

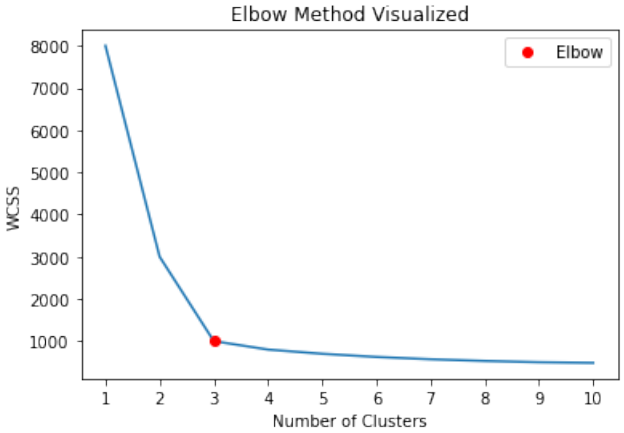


Figure 3: Plot of WCSS Vs. Number of Clusters

3.2

$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{n} (\boldsymbol{q_i} - \boldsymbol{p_i})^2} \tag{7}$$

For this section the record numbers where identified with python. Only the relevant points were considered when calculating the distances instead of calculating the distance for all 40 points. This was done with the Euclidean Distance formula (7) shown above .

3.2.1

$$x = \begin{bmatrix} 5.03 \\ -4.97 \\ 3.50 \end{bmatrix} \qquad R_{11} = \begin{bmatrix} 4.17 \\ -3.45 \\ 3.64 \end{bmatrix}$$

$$d(\boldsymbol{x}, \boldsymbol{R_{11}}) = \sqrt{(5.03 - 4.17)^2 + (-4.97 - -3.45)^2 + (3.50 - 3.64)^2} = 1.75$$

$$\text{Label}(\boldsymbol{x}) = \text{Max Proportion of labels} = 0$$

Record #	Distance from the record	Label of the record
11	1.75	0
Prediction	-	0

3.2.2

$$x = \begin{bmatrix} -3.11 \\ -6.84 \\ 10.84 \end{bmatrix} \qquad R_{28} = \begin{bmatrix} -3.76 \\ -8.29 \\ 10.30 \end{bmatrix}$$

$$R_{25} = \begin{bmatrix} -4.50 \\ -5.81 \\ 10.89 \end{bmatrix} \qquad R_{27} = \begin{bmatrix} -4.02 \\ -8.30 \\ 12.56 \end{bmatrix}$$

$$d(\boldsymbol{x}, \boldsymbol{R_{28}}) = \sqrt{(-3.11 - -3.76)^2 + (-6.84 - -8.29)^2 + (10.84 - 10.30)^2} = 1.64$$

$$d(\boldsymbol{x}, \boldsymbol{R_{25}}) = \sqrt{(-3.11 - -4.50)^2 + (-6.84 - -5.81)^2 + (10.84 - 10.89)^2} = 1.74$$

$$d(\boldsymbol{x}, \boldsymbol{R_{27}}) = \sqrt{(-3.11 - -4.02)^2 + (-6.84 - -8.30)^2 + (10.84 - 12.56)^2} = 2.53$$

$$\text{Label}(\boldsymbol{x}) = \text{Max Proportion of labels} = 1$$

Record #	Distance from the record	Label of the record
28	1.64	1
25	1.74	1
27	2.53	1
Prediction	-	1

3.2.3

$$x = \begin{bmatrix} -1.62 \\ -6.34 \\ 4.66 \end{bmatrix} \qquad R_{36} = \begin{bmatrix} -0.81 \\ -5.74 \\ 4.39 \end{bmatrix}$$

$$R_{39} = \begin{bmatrix} -0.72 \\ -6.76 \\ 5.84 \end{bmatrix} \qquad R_{29} = \begin{bmatrix} -0.55 \\ -7.92 \\ 6.72 \end{bmatrix}$$

$$R_{30} = \begin{bmatrix} -1.28 \\ -2.41 \\ 4.57 \end{bmatrix} \qquad R_{4} = \begin{bmatrix} 1.04 \\ -6.93 \\ 8.29 \end{bmatrix}$$

$$d(\boldsymbol{x}, \boldsymbol{R_{36}}) = \sqrt{(-1.62 - -0.81)^2 + (-6.34 - -5.74)^2 + (4.66 - 4.39)^2} = 1.04$$

$$d(\boldsymbol{x}, \boldsymbol{R_{39}}) = \sqrt{(-1.62 - -0.72)^2 + (-6.34 - -6.76)^2 + (4.66 - 5.84)^2} = 1.54$$

$$d(\boldsymbol{x}, \boldsymbol{R_{29}}) = \sqrt{(-1.62 - -0.55)^2 + (-6.34 - -7.92)^2 + (4.66 - 6.72)^2} = 2.81$$

$$d(\boldsymbol{x}, \boldsymbol{R_{30}}) = \sqrt{(-1.62 - -1.28)^2 + (-6.34 - -2.41)^2 + (4.66 - 4.57)^2} = 3.95$$

$$d(\boldsymbol{x}, \boldsymbol{R_{4}}) = \sqrt{(-1.62 - 1.04)^2 + (-6.34 - -6.93)^2 + (4.66 - -8.29)^2} = 4.54$$

$$\text{Label}(\boldsymbol{x}) = \text{Max Proportion of labels} = 1$$

Record #	Distance from the record	Label of the record
36	1.04	1
39	1.54	1
29	2.81	1
30	3.95	1
4	4.54	0
Prediction	-	1

3.2.4

The training data and test data were run through various K-NN models to arrive at the intermediate table shown below. The results from this table were then used to calculate error rates show in the results table on the next page.

K-NN	Correct Predictions (Train)	Correct Predictions (Test)
1-NN	40	39
3-NN	40	39
5-NN	38	38
7-NN	38	38
9-NN	38	38
11-NN	38	36
13-NN	37	39
Prediction	-	-

Next, equation (8) was used with the data above to calculate the error rates.

$$\text{Error Rate} = 1 - \frac{\text{Correct Predictions}}{\text{Total Number Of Records}} \tag{8}$$

$$\text{Train error}_{1NN} = 1 - \frac{40}{40} = 0.00$$
$$\text{Train error}_{3NN} = 1 - \frac{40}{40} = 0.00$$
$$\text{Train error}_{5NN} = 1 - \frac{38}{40} = 0.05$$
$$\text{Train error}_{7NN} = 1 - \frac{38}{40} = 0.05$$
$$\text{Train error}_{9NN} = 1 - \frac{38}{40} = 0.05$$
$$\text{Train error}_{11NN} = 1 - \frac{38}{40} = 0.05$$
$$\text{Train error}_{13NN} = 1 - \frac{37}{40} = 0.08$$

$$\text{Test error}_{1NN} = 1 - \frac{39}{40} = 0.03$$
$$\text{Test error}_{3NN} = 1 - \frac{39}{40} = 0.03$$
$$\text{Test error}_{5NN} = 1 - \frac{38}{40} = 0.05$$
$$\text{Test error}_{7NN} = 1 - \frac{38}{40} = 0.05$$
$$\text{Test error}_{9NN} = 1 - \frac{38}{40} = 0.05$$
$$\text{Test error}_{11NN} = 1 - \frac{36}{40} = 0.10$$
$$\text{Test error}_{13NN} = 1 - \frac{39}{40} = 0.03$$

Below is the table of results:

K-NN	Training Error Rate	Test Error Rate
1-NN	0.00	0.03
3-NN	0.00	0.03
5-NN	0.05	0.05
7-NN	0.05	0.05
9-NN	0.05	0.05
11-NN	0.05	0.10
13-NN	0.08	0.03
Prediction	-	-

Code results for question 3.2

```
Results for 3.2.1
Class Prediction:  [0]
Distances and indexs of points:  (array([[1.7520274]]), array([[10]]))

Results for 3.2.2
Class Prediction:  [1]
Distances and indexs of points:  (array([[1.64106673, 1.7393677 , 2.5263
0164]]), array([[27, 24, 26]]))

Results for 3.2.3
Class Prediction:  [1]
Distances and indexs of points:  (array([[1.04355163, 1.54233589, 2.8080
0641, 3.94570653, 4.53878838]]), array([[35, 38, 28, 29,  3]]))

Results for 3.2.4
training error(1-NN):  0.0
test error(1-NN):  0.025

training error(3-NN):  0.0
test error(3-NN):  0.025

training error(5-NN):  0.05
test error(5-NN):  0.05

training error(7-NN):  0.05
test error(7-NN):  0.05

training error(9-NN):  0.05
test error(9-NN):  0.05

training error(11-NN):  0.05
test error(11-NN):  0.1

training error(13-NN):  0.075
test error(13-NN):  0.025
```