**HO CHI MINH UNIVERSITY OF TECHNOLOGY**

**OFFICE FOR INTERNATIONAL STUDY PROGRAMS**

🙞∙∙∙☼∙∙∙🙜



**FACULTY OF MECHANICAL ENGINEERING**

**APPLICATIONS OF ARTIFICIAL INTELLIEGENCE PROJECT**

PROBLEM SET 2

CLASS CC01

STUDENT: NGUYỄN TRẦN VIỆT HƯNG - 2152619

INSTRUCTOR: PHD PHÙNG THANH HUY

**Hồ Chí Minh City – 2023**

# Problem 1:

|  |  |  |  |
| --- | --- | --- | --- |
| **Trial No.** | **Feature 1** | **Feature 2** | **Intended Movement** |
| 1 | 0.56 | 0.21 | Flexion |
| 2 | 0.67 | 0.34 | Flexion |
| 3 | 0.24 | 0.41 | Extension |
| 4 | 0.52 | 0.92 | Extension |
| 5 | 0.34 | 0.64 | Extension |
| 6 | 0.87 | 0.43 | Flexion |
| 7 | 0.33 | 0.82 | Extension |
| 8 | 0.55 | 0.23 | Flexion |
| 9 | 0.62 | 0.44 | Extension |
| 10 | 0.81 | 0.45 | Flexion |
| 11 | 0.3 | 0.51 | Extension |
| 12 | 0.62 | 0.51 | Flexion |

Define Flexion as class 1 and Extension as class 2. We find averages of feature of each class and whole data average:

; ;

Within-class scatter matrix for each class:

Between-class scatter matrix:

Projection matrix:

Find eigen value and eigen vector:

⬄

⬄

⬄

With we have:

* VTR = [

We normalize eigen vector:

LDA model is:

# Problem 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trial** | **Mean** | **Variance** | **Skewness** | **Activity** |
| 1 | 0.56 | 0.21 | 0.45 | RMI |
| 2 | 0.67 | 0.34 | 0.35 | MA |
| 3 | 0.24 | 0.41 | 0.11 | Rest |
| 4 | 0.52 | 0.92 | 0.34 | LMI |
| 5 | 0.34 | 0.64 | 0.43 | MA |
| 6 | 0.87 | 0.43 | 0.46 | RMI |
| 7 | 0.33 | 0.82 | 0.51 | Rest |
| 8 | 0.55 | 0.23 | 0.52 | RMI |
| 9 | 0.62 | 0.44 | 0.51 | LMI |
| 10 | 0.81 | 0.45 | 0.24 | Rest |
| 11 | 0.3 | 0.51 | 0.26 | LMI |
| 12 | 0.62 | 0.51 | 0.41 | MA |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Mean** | **Variance** | **Skewness** |
| 1 | 0.43 | 0.44 | 0.61 |
| 2 | 0.45 | 0.32 | 0.32 |

Because data is continuous, we use Gaussian Naive Bayes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | Mean(X1) | Variance(X1) | Mean(X2) | Variance(X2) | Mean(X3) | Variance(X3) |
|  |
| RMI | 0.66 | 0.0221 | 0.29 | 0.0099 | 0.4767 | 0.001 |  |
| MA | 0.5433 | 0.0211 | 0.4967 | 0.0151 | 0.3967 | 0.0012 |  |
| Rest | 0.46 | 0.0626 | 0.56 | 0.0341 | 0.2867 | 0.0278 |  |
| LMI | 0.48 | 0.0179 | 0.6233 | 0.0448 | 0.37 | 0.0109 |  |

We have:

With m class and n features we have:

* With data test\_1 = [0.43, 0.44, 0.61] has the result:

* test\_1 is classified as Rest
* With data test\_2 = [0.45, 0.32, 0.32] has the result:

* test\_2 is classified as LMI

# Problem 4:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Age(years)** | **Diastolic Blood Pressure (mm Hg)** | **LDL Cholesterol (mg/dL)** | **Body Mass Index (kg/m2)** | **Heart disease** |
| 1 | 25 | 80 | 100 | 23 | No |
| 2 | 65 | 100 | 150 | 28 | Yes |
| 3 | 30 | 110 | 130 | 30 | Yes |
| 4 | 35 | 70 | 100 | 22 | No |
| 5 | 40 | 100 | 130 | 27 | Yes |
| 6 | 25 | 90 | 120 | 25 | No |
| 7 | 32 | 95 | 110 | 28 | Yes |
| 8 | 39 | 90 | 100 | 22 | No |
| 9 | 40 | 100 | 140 | 26 | Yes |
| 10 | 50 | 95 | 80 | 21 | No |
| 11 | 20 | 85 | 75 | 24 | No |
| 12 | 45 | 95 | 145 | 24 | Yes |
| 13 | 48 | 95 | 135 | 26 | Yes |
| 14 | 52 | 90 | 120 | 26 | No |
| 15 | 28 | 85 | 95 | 18 | No |
| 16 | 39 | 90 | 140 | 26 | Yes |
| 17 | 32 | 90 | 85 | 28 | No |
| 18 | 47 | 100 | 175 | 32 | Yes |
| 19 | 41 | 95 | 130 | 29 | Yes |
| 20 | 61 | 95 | 75 | 20 | No |

## **Question 1:**

We choose first data point of dataset for LDA model testing.

Define Yes and No as and .

Define , as average value of Yes and No.

Average vector of Yes:

Similarly with average vector of No, ta have:

Average vector of whole dataset:

Scattering matrix of each class:

Within-class scatter matrix of class:

Between-class scatter matrix:

Projection matrix:

Eigen value and eigen vector:

⬄

⬄

⬄

We choose maximum eigen value , we have:

Use Gaussian method for above matrix, we get the result:

* Eigen vector has form: v = [

Standardize eigen vector:

Convert data (remove the first point) with eigen vector we find above:

Average value of points of class Yes in :

Similarly we can calculate average value of points of class No in :

Average value of points of :

With average value of so if convert data of point data we analyze larger than it belongs to class Yes, if not it belongs to class No.

First point we remove at the beginning:

We convert this data point into:

Because we predict this data point belongs to class No.

Do the same way 19 times with the dataset, for the final result we predict correctly 17/20 times, from now we calculate accuracy of the model:

## **Question 2:**

Because data type is continuous so we use Gaussian Naive Bayes method.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class | Mean (X1) | Variance (X1) | Mean (X2) | Variance (X2) | Mean| (X3) | Variance (X3) | Mean (X4) | Variance (X4) |
|
| No | 36.7 | 166.01 | 87 | 51 | 95 | 245 | 22.9 | 7.89 |
| Yes | 42.7 | 85.61 | 98 | 26 | 138.5 | 255.25 | 27.6 | 4.84 |

With m class and n feature we have:

* With test\_1 = [40,110,150,28] we have:
* test\_1 is classified as Yes
* With test\_2 = [25,90,135,26] we have:
* test\_2 is classified as Yes
* With test\_3 = [65,90,95,22] we have:
* test\_3 is classified as No

# Problem 5:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. N0.** | **Age (years)** | **Diastolic Blood Pressure (mm Hg)** | **LDL Cholesterol (mg/dL)** | **Body Mass Index (kg/m2)** | **Heart Disease** |
| 1 | 25 | 80 | 100 | 23 | No |
| 2 | 65 | 100 | 150 | 28 | Yes |
| 3 | 30 | 110 | 130 | 30 | Yes |
| 4 | 35 | 70 | 100 | 22 | No |
| 5 | 40 | 100 | 130 | 27 | Yes |
| 6 | 25 | 90 | 120 | 25 | No |
| 7 | 32 | 95 | 110 | 28 | Yes |
| 8 | 39 | 90 | 100 | 22 | No |
| 9 | 40 | 100 | 140 | 26 | Yes |
| 10 | 50 | 95 | 80 | 21 | No |

Because the data is continuous so we use Gaussian Naive Bayes method.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class | Mean (X1) | Variance (X1) | Mean (X2) | Variance (X2) | Mean| (X3) | Variance (X3) | Mean (X4) | Variance (X4) |
|  |
| No | 34.8 | 88.16 | 85 | 80 | 100 | 160 | 22.6 | 1.84 |  |
| Yes | 41.4 | 155.84 | 101 | 24 | 132 | 176 | 27.8 | 1.76 |  |

With m class and n feature we have:

* With testing data = [25, 90, 135, 26] sẽ có kết quả:
* test is classified as Yes

# Problem 6:

|  |  |  |
| --- | --- | --- |
| **Type** | **Diameter (inches)** | **Weight (grams)** |
| Apple | 3 | 130 |
| Apple | 2.7 | 105 |
| Orange | 2.9 | 160 |
| Orange | 2.5 | 125 |
| Apple | 2.6 | 100 |
| Orange | 2.4 | 120 |
| Apple | 3.1 | 140 |
| Apple | 2.8 | 120 |
| Orange | 2.7 | 140 |
| Orange | 2.6 | 130 |
| Apple | 2.9 | 130 |
| Orange | 2.5 | 122 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Type** | **Diameter (inches)** | **Weight (gms)** |
| 1 | ? | 2.9 | 145 |
| 2 | ? | 2.6 | 103 |
| 3 | ? | 2.8 | 120 |

Because the data is continuous so we use Gaussian Naive Bayes method.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Mean(X1) | Variance(X1) | Mean(X2) | Variance(X2) |
|
| Apple | 2.85 | 0.0292 | 120.8333 | 203.4722 |
| Orange | 2.6 | 0.0267 | 132.8333 | 190.1389 |

We have:

With m class and n feature we have:

* With data test\_1 = [2.9, 145] we have:

* test\_1 is classified as Apple
* With data test\_2 = [2.6, 103] we have:

* test\_2 is classified as Apple
* With data test\_3 = [2.8, 120] we have:

* test\_3 is classified as Apple

# Problem 7:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 202 | 225 | 168 | 23 | 37 | 33 | 0 |
| 0 | 251 | 114 | 63 | 66 | 32 | 56 | 0 |
| 0 | 201 | 40 | 245 | 92 | 213 | 216 | 0 |
| 0 | 70 | 230 | 132 | 2 | 44 | 146 | 0 |
| 0 | 8 | 218 | 171 | 10 | 193 | 48 | 0 |
| 0 | 53 | 33 | 30 | 247 | 53 | 79 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Zero padding for images to perform image convolution on the condition that the output image remains the same size.

Perform convolution with the kernel

|  |  |  |
| --- | --- | --- |
| 0 | -1 | 0 |
| -1 | 5 | -1 |
| 0 | -1 | 0 |

Now, to avoid the case where the value of a pixel is outside the [0,255] area, we let the convolution values ​​CImg(i, j) < 0 have a value of 0, and the convolution value CImg(i, j) > 255 will have the value 255.

We do the same for the remaining elements and we will have the image matrix as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

1. Maxpooling, size = (2,2), stride = 2:

We consider the colored parts:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the maximum number 255.

We move to the next part:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the maximum number 255.

We move to the next part:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the maximum number 97.

We move to the next part:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the maximum number 255.

|  |  |  |
| --- | --- | --- |
| 255 | 255 | 97 |
| 255 | 255 | 255 |
| 255 | 255 | 255 |

We continue to do so until the end of the image matrix. Here are the results:

1. Averagepooling, size = (2,2), stride = 2:

We consider the colored parts:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the average value:

We move to the next part:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the average value:

We move to the next part:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the average value:

We move to the next part:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 255 | 255 | 255 | 0 | 97 | 72 |
| 255 | 0 | 0 | 120 | 0 | 0 |
| 255 | 0 | 255 | 0 | 255 | 255 |
| 0 | 255 | 12 | 0 | 0 | 255 |
| 0 | 255 | 255 | 0 | 255 | 0 |
| 224 | 0 | 0 | 255 | 0 | 255 |

We choose the average value: 127.5

|  |  |  |
| --- | --- | --- |
| 191 | 93 | 42 |
| 127 | 66 | 191 |
| 119 | 127 | 127 |

# Problem 8:

A diagram of a tree

Description automatically generated**Root Node:**

**Left Branch (A = 0):**

Gini Impurity = 0 for both.

**Right Branch (A = 1):**

**Left Branch (B = 0):**

Gini Impurity = 0 for both

Right Branch (B = 1):

Gini Impurity = 0 for both

**Gini Gain at each Split:**

**Split on A:** Gini Gain = Gini Impurity (Parent) - (Weighted Average of Child Gini Impurities)

(perfect split)

**Split on B** (in "Right Branch - A = 1"):

(perfect split)

**Predictions:**

1. A = 1, B = 0, C = 1, predicting Y = 1.
2. A = 0, B = 0, C = 0, predicting Y = 0.