

# Minimum Risk Bayes Decision Theoretic Classifier

จัดทำโดย

นายปณณวิชญ์ พันธวงศ์ 600610752

เสนอ

รศ.ดร.ศันสนีย์ เอื้อพันธวัริยะกุล

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา

261754

ภาคเรียนที่ 1 ปีการศึกษา 2564

## 1. Theories and Related Method

### 1.1. Covariance Matrices

ความแปรปรวนร่วมเกี่ยว(Covariance) เป็นการวัดปริมาณการเปลี่ยนแปลงของสองตัวแปรว่า จะมีการเปลี่ยนแปลงตามกันมาน้อยเท่าใด ในการคำนวณความแปรปรวนของของตัวแปรสามารถ คำนวณได้จาก

$$\sigma(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

เมทริกความแปรปรวนร่วมเกี่ยว(Covariance Matrix) เป็นเมทริกที่คำนวณ Covariance ที่มี มากกว่า 2 ตัวแปรให้อยู่ในรูปของเมทริก โดย  $\sigma(x_i, y_i) = \sigma(y_i, x_i)$  ซึ่งคำนวณได้ในรูปของเมทริก

$$\mathcal{E} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T$$

โดยผลลัพธ์ที่ได้จะอยู่ในรูปของเมทริกที่รวม Variance ของแต่ละคู่ตัวแปร ดังเมทริกต่อไปนี้

$$\mathcal{E} = \begin{pmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{pmatrix}$$

### 1.2. Multivariate normal distribution

การแจกแจงปกติ(Normal distribution) เป็นการแจกแจงความน่าจะเป็นของค่าของตัวแปร สุ่มที่เป็นค่าแบบต่อเนื่อง โดยที่ค่าของตัวแปรสุ่มมีแนวโน้มที่จะมีค่าอยู่ใกล้ ๆ กับค่า ๆ หนึ่ง กราฟแสดง ค่าฟังก์ชันความหนาแน่น (probability density function) จะเป็นรูปคล้ายระฆังคว่ำ หรือเรียกว่า Gaussian function

การแจกแจงแบบปกติหลายตัวแปร(Multivariate normal distribution) เป็นการแจกแจงความน่าจะเป็นของค่าหลายตัวแปร โดยมีสมการดังนี้

$$f(x_d|\omega) = \frac{\exp(-\frac{1}{2}(x-\mu)^T \mathcal{E}^{-1}(x-\mu))}{\sqrt{(2\pi)^d \|\mathcal{E}\|}}$$

### 1.3. Bayes' Theorem

ทฤษฎีของเบย์(Bayes' theorem) เป็นทฤษฎีที่เป็นส่วนขยายของความน่าจะเป็นแบบมีเงื่อนไข หรืออาจกล่าวว่าเป็นการหาความน่าจะเป็นของส่วนย่อยจากเหตุการณ์ที่สนใจหนึ่งที่เกิดขึ้นแล้ว โดยมีสมการดังนี้

$$P(x_i|\omega) = \frac{P(\omega|x_i) \cdot P(x_i)}{\sum_{i=1}^n P(\omega|x_i) \cdot P(x_i)}$$

กฎของเบย์(Bayes' rule) ที่ใช้ในการทดลองนี้จะเปรียบเทียบว่า sample ที่เข้ามานั้นมีโอกาสอยู่ในคลาสไหนมากที่สุด กล่าวคือจะเปรียบเทียบ  $P(x_i|\omega_m)$  และ  $P(x_i|\omega_n)$  หาก  $P(x_i|\omega_m)$  มีค่ามากกว่า  $P(x_i|\omega_n)$  จะถือว่า  $x_i$  อยู่ใน  $\omega_m$  ในทางกลับกันหาก  $P(x_i|\omega_m)$  มีค่าน้อยกว่า  $P(x_i|\omega_n)$  จะถือว่า  $x_i$  จะอยู่ใน  $\omega_n$

$$P(x_i|\omega_m) > P(x_i|\omega_n), \quad x_i \in \omega_m$$

$$P(x_i|\omega_m) < P(x_i|\omega_n), \quad x_i \in \omega_n$$

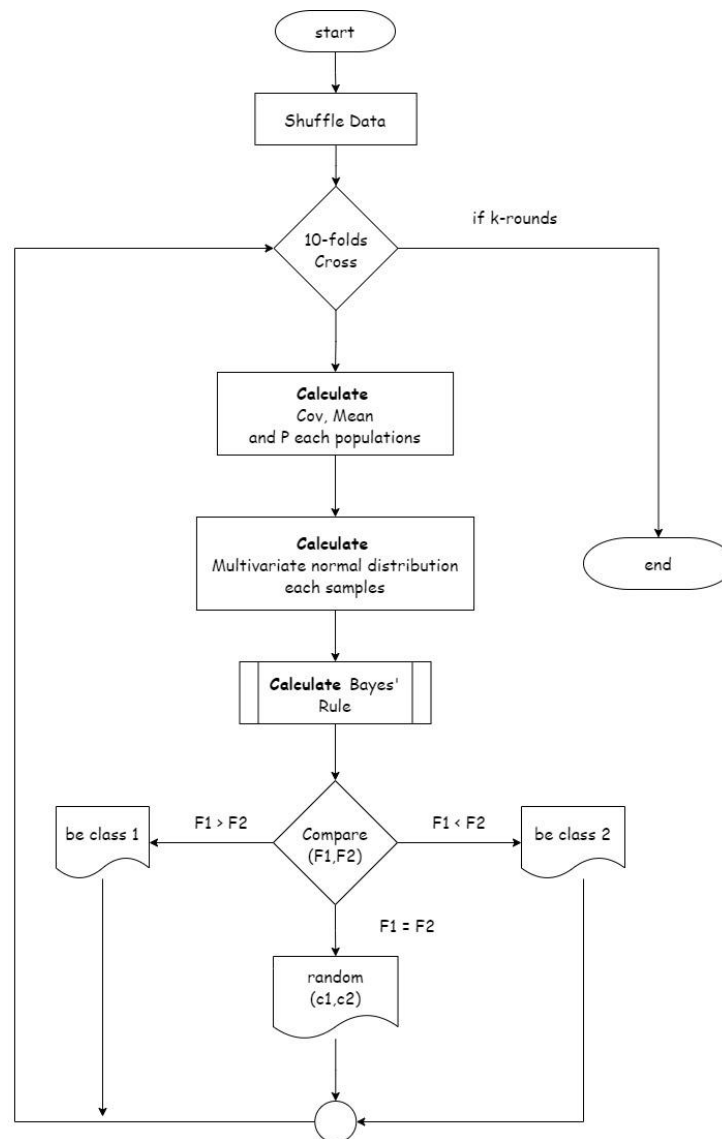
$$P(x_i|\omega_m) = P(x_i|\omega_n), \quad \text{random}(\omega_n, \omega_m)$$

#### 1.4. Cross-validation Test

การวัดประสิทธิภาพด้วยวิธี Cross-validation นี้จะทำการแบ่งข้อมูลออกเป็นหลายส่วน (มักจะแสดงด้วยค่า  $k$ ) เช่น 5-fold cross-validation คือ ทำการแบ่งข้อมูลออกเป็น 5 ส่วน โดยที่แต่ละส่วนมีจำนวนข้อมูลเท่ากัน หรือ 10-fold cross-validation คือ การแบ่งข้อมูลออกเป็น 10 ส่วน โดยที่แต่ละส่วนมีจำนวนข้อมูลเท่ากัน หลังจากนั้นข้อมูลหนึ่งส่วนจะใช้เป็นตัวทดสอบประสิทธิภาพของโมเดล ทำวนไปเช่นนี้จนครบจำนวนที่แบ่งไว้ ซึ่งในการทดลองนี้จะใช้ 10-fold cross-validation

## 2. Algorithm (Flow Chart)

อันดับแรกจะทำการ shuffle ข้อมูลทั้งหมด จากนั้นจะทำการทดลองโดยจะทำการแบ่ง 10-fold cross-validation จากข้อมูล 200 records กล่าวคือ ข้อมูลสำหรับหา Covariance, Mean,  $P(w)$  ของคลาสใดๆ มี 180 และสำหรับทดสอบตามกฎของเบย์ 20 records โดยจะนำข้อมูลสำหรับทดสอบไปหาค่าของการแจกแจงแบบปกติหลายตัวแปรของแต่ละคลาส จากนั้นจะทำการเปรียบเทียบตามกฎของเบย์และวัดผลลัพธ์



รูปที่ 1 Flow chart ของระบบ

### 3. Analyze data

ข้อมูลจะประกอบไปด้วย 4 features มีทั้งหมด 200 records ซึ่งได้ถูกแบ่งกลุ่ม(class) ไว้ทั้งหมด 2 class ได้แก่ class 1 และ class 2 ดังตารางนี้

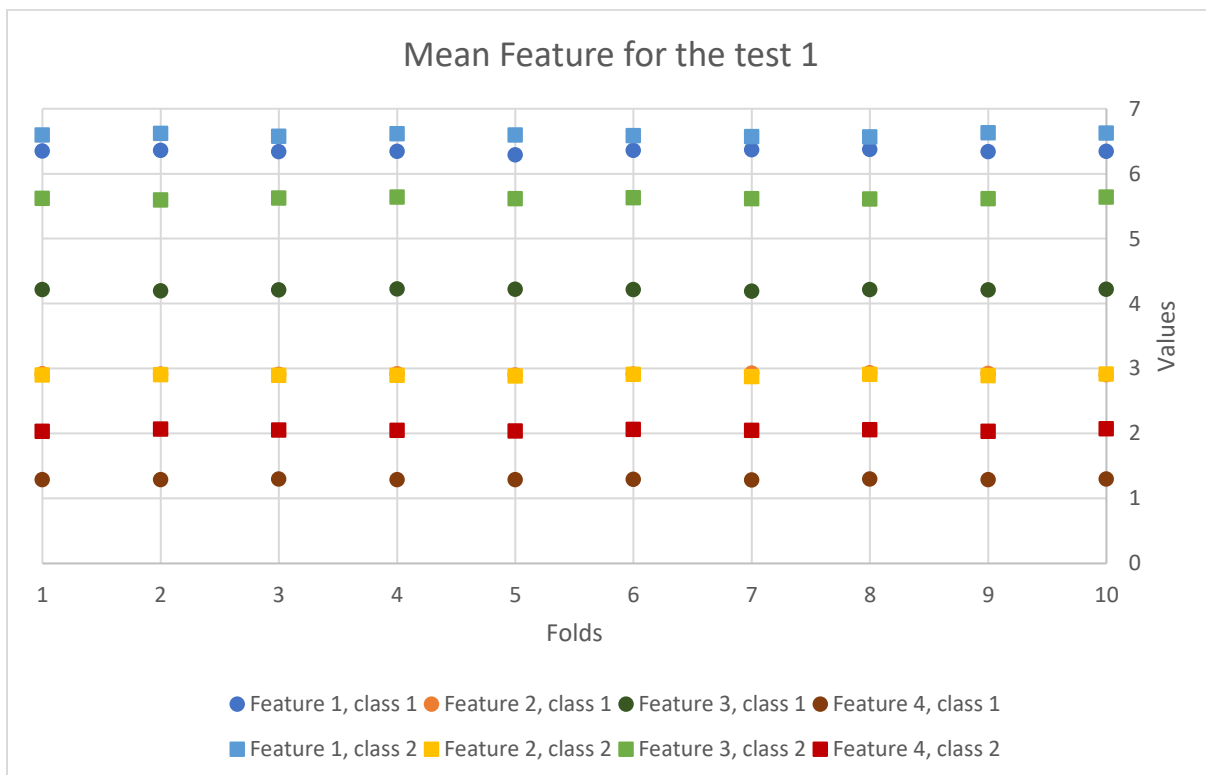
<i>No.</i>	<i>Feature 1</i>	<i>Feature 2</i>	<i>Feature 3</i>	<i>Feature 4</i>	<i>Class</i>
1	5.91	3.09	3.74	1.17	1
...	...	...	...	...	...
200	6.29	3.22	5.77	1.59	2

## 4. Experimental

ผู้จัดทำได้ทำการทดลอง 2 แบบ ซึ่งแต่ละการทดลองจะทำการแบ่งเทรน(Train) กับเทส(Test) โดยใช้ 10-folds cross validation โดยทำตามอัลกอริทึมตามข้อ 2 โดยการทดลองที่ 1 จะใช้ 4 Features และการทดลองที่ 2 จะใช้ 2 Features โดยเป็น Feature1 และ Feature2 ทำการทดลอง

### 4.1. การทดลองที่ 1

สำหรับการทดลองที่ 1 ใช้ 4 features ในการทดลอง โดยได้ผลลัพธ์ดังนี้



รูปที่ 2 ค่าเฉลี่ยของ Features ในแต่ละ Fold

จากรูปที่ 2 สังเกตได้ว่า ค่าเฉลี่ยของ Feature3(สีเขียว) และ Feature4(สีแดง) ของแต่ละ class มีค่าที่ต่างกันเห็นได้อย่างชัดเจน และสำหรับ Feature1(สีฟ้า) มีค่าเฉลี่ยของแต่ละคลาสต่างกันเล็กน้อย แต่สำหรับ Feature2(สีเหลือง) มีค่าที่ใกล้เคียงกันมากๆจนแยกไม่ออก

Fold 1	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 2	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.314	-0.023	0.021	-0.011	Fea. 1	0.312	-0.022	0.017	-0.012
Fea. 2	-0.024	0.116	-0.009	0.003	Fea. 2	-0.02	0.118	-0.012	0.005
Fea. 3	0.021	-0.007	0.166	0.016	Fea. 3	0.018	-0.013	0.148	0.014
Fea. 4	-0.009	0.004	0.017	0.04	Fea. 4	-0.011	0.004	0.014	0.04
Fold 3	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 4	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.313	-0.007	0.026	-0.006	Fea. 1	0.301	-0.023	0.015	-0.015
Fea. 2	-0.009	0.114	-0.005	0.002	Fea. 2	-0.023	0.119	-0.008	0.006
Fea. 3	0.025	-0.004	0.162	0.016	Fea. 3	0.015	-0.009	0.152	0.01
Fea. 4	-0.009	0.001	0.015	0.038	Fea. 4	-0.015	0.006	0.01	0.038
Fold 5	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 6	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.282	-0.032	0.019	-0.013	Fea. 1	0.324	-0.02	0.032	-0.011
Fea. 2	-0.033	0.113	-0.011	0.006	Fea. 2	-0.019	0.119	-0.005	0.006
Fea. 3	0.017	-0.011	0.145	0.018	Fea. 3	0.033	-0.005	0.143	0.011
Fea. 4	-0.014	0.006	0.018	0.041	Fea. 4	-0.009	0.007	0.012	0.041
Fold 7	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 8	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.317	-0.023	0.043	-0.015	Fea. 1	0.306	-0.014	0.038	-0.012
Fea. 2	-0.023	0.11	0	0.011	Fea. 2	-0.013	0.104	-0.001	0.003
Fea. 3	0.044	0	0.144	0.017	Fea. 3	0.04	-0.002	0.158	0.018
Fea. 4	-0.016	0.01	0.017	0.036	Fea. 4	-0.013	0.002	0.017	0.039
Fold 9	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 10	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.326	-0.017	0.024	-0.015	Fea. 1	0.276	-0.016	0.031	-0.006
Fea. 2	-0.02	0.099	-0.018	0.01	Fea. 2	-0.014	0.117	-0.007	0.006
Fea. 3	0.025	-0.015	0.157	0.017	Fea. 3	0.034	-0.007	0.162	0.015
Fea. 4	-0.013	0.011	0.017	0.04	Fea. 4	-0.008	0.005	0.013	0.039

รูปที่ 3 Covariance Matrix ของ Class 1 ในแต่ละ Fold



Fold 1	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 2	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.353	0.022	0	0.002	Fea. 1	0.369	-0.002	-0.01	-0.009
Fea. 2	0.02	0.097	0.011	-0.001	Fea. 2	-0.001	0.106	0.009	-0.006
Fea. 3	0.001	0.012	0.187	0.028	Fea. 3	-0.007	0.01	0.182	0.025
Fea. 4	0.003	0	0.028	0.078	Fea. 4	-0.006	-0.005	0.027	0.075
Fold 3	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 4	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.369	0.004	-0.003	0	Fea. 1	0.373	0.01	-0.02	-0.012
Fea. 2	0.007	0.105	0.014	-0.002	Fea. 2	0.012	0.105	0.016	-0.004
Fea. 3	0.002	0.014	0.194	0.025	Fea. 3	-0.017	0.016	0.2	0.031
Fea. 4	0.003	-0.001	0.026	0.079	Fea. 4	-0.011	-0.004	0.031	0.08
Fold 5	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 6	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.389	0.005	-0.013	0	Fea. 1	0.387	0	-0.011	-0.009
Fea. 2	0.005	0.107	0.011	-0.006	Fea. 2	0.001	0.104	0.003	-0.008
Fea. 3	-0.012	0.011	0.197	0.03	Fea. 3	-0.009	0.003	0.175	0.024
Fea. 4	-0.001	-0.006	0.029	0.078	Fea. 4	-0.007	-0.007	0.025	0.081
Fold 7	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 8	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.366	0.001	-0.01	-0.001	Fea. 1	0.338	0.02	-0.031	-0.011
Fea. 2	0.001	0.103	0.013	-0.004	Fea. 2	0.02	0.096	0.015	-0.002
Fea. 3	-0.013	0.012	0.19	0.024	Fea. 3	-0.028	0.016	0.168	0.033
Fea. 4	-0.003	-0.005	0.023	0.081	Fea. 4	-0.012	-0.003	0.031	0.078
Fold 9	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fold 10	Fea. 1	Fea. 2	Fea. 3	Fea. 4
Fea. 1	0.36	-0.002	0.007	0	Fea. 1	0.383	-0.012	-0.022	-0.007
Fea. 2	-0.005	0.097	0.004	-0.009	Fea. 2	-0.01	0.095	0.017	0.004
Fea. 3	0.006	0.006	0.181	0.025	Fea. 3	-0.023	0.016	0.195	0.021
Fea. 4	0.001	-0.007	0.027	0.08	Fea. 4	-0.008	0.003	0.02	0.073

รูปที่ 4 Covariance Matrix ของ Class 2 ในแต่ละ Fold

Fold 1	Predict		Acc	Fold 2	Predict		Acc
Actual	10	0	100	Actual	7	1	95
	0	10			0	12	
Fold 3	Predict		Acc	Fold 4	Predict		Acc
Actual	12	0	100	Actual	10	0	100
	0	8			0	10	
Fold 5	Predict		Acc	Fold 6	Predict		Acc
Actual	11	0	100	Actual	9	0	95
	0	9			1	10	
Fold 7	Predict		Acc	Fold 8	Predict		Acc
Actual	9	2	90	Actual	13	0	100
	0	9			0	7	
Fold 9	Predict		Acc	Fold 10	Predict		Acc
Actual	9	0	100	Actual	7	0	95
	0	11			1	12	
Mean Acc.			97.5				

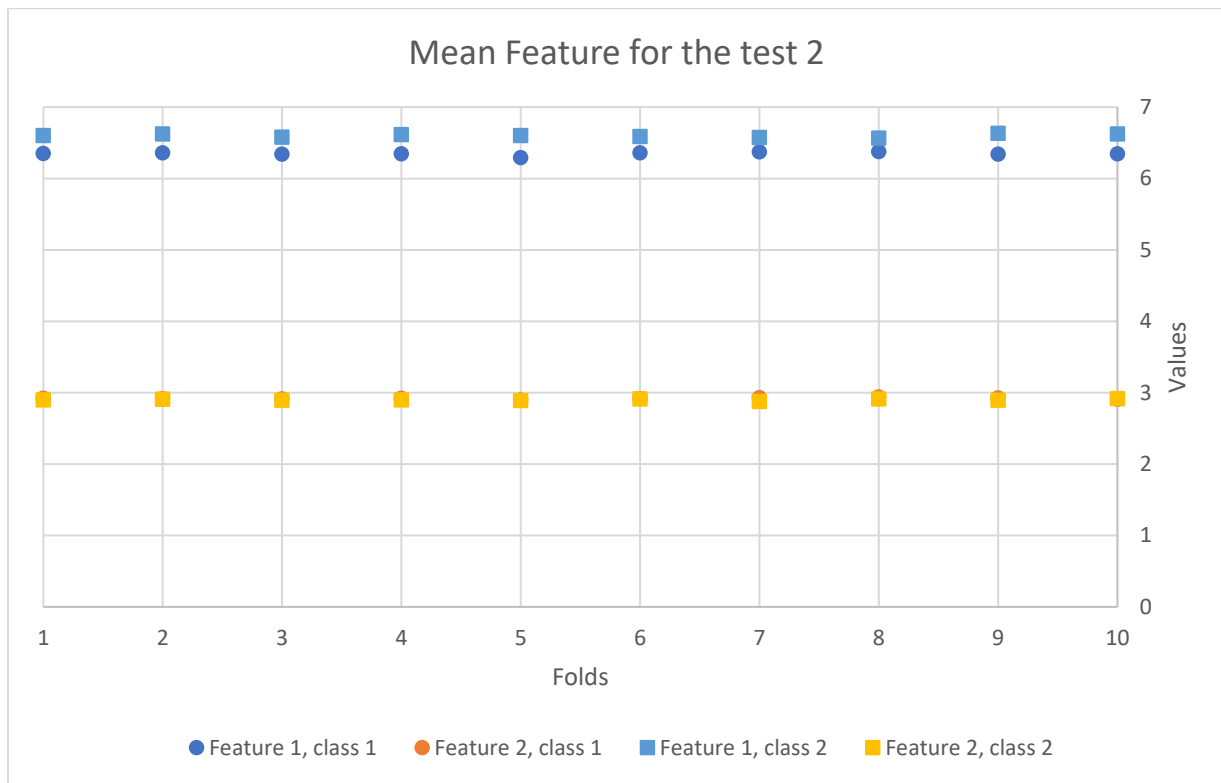
รูปที่ 5 Confusion Matrix Test 1 ในแต่ละ fold

จากรูปที่ 5 พบว่าโดยเฉลี่ยทั้ง 10 folds มีค่าความแม่นยำถึง 97.5% โดยค่าความแม่นยำ มีค่าต่ำที่สุดอยู่ใน fold 7 ซึ่งมีค่า 90%

## 4.2. การทดลองที่ 2

สำหรับการทดลองที่ 1 ใช้ 2 features ในการทดลอง โดยเลือกใช้ Feature1 และ Feature2 ได้

ผลลัพธ์ดังนี้



รูปที่ 6 ค่าเฉลี่ยของ Features ในแต่ละ Fold

จากรูปที่ 6 สังเกตได้ว่า ค่าเฉลี่ยของทั้ง 2 Features ในแต่ละคลาส มีค่าใกล้เคียงกันมากๆ โดยมีเพียงแค่ Feature1 ที่มีค่าต่างกันเล็กน้อย

Fold 1	Fea. 1	Fea. 2	Fold 2	Fea. 1	Fea. 2
Fea. 1	0.314	-0.023	Fea. 1	0.312	-0.022
Fea. 2	-0.024	0.116	Fea. 2	-0.02	0.118
Fold 3	Fea. 1	Fea. 2	Fold 4	Fea. 1	Fea. 2
Fea. 1	0.313	-0.007	Fea. 1	0.301	-0.023
Fea. 2	-0.009	0.114	Fea. 2	-0.023	0.119
Fold 5	Fea. 1	Fea. 2	Fold 6	Fea. 1	Fea. 2
Fea. 1	0.282	-0.032	Fea. 1	0.324	-0.02
Fea. 2	-0.033	0.113	Fea. 2	-0.019	0.119
Fold 7	Fea. 1	Fea. 2	Fold 8	Fea. 1	Fea. 2
Fea. 1	0.317	-0.023	Fea. 1	0.306	-0.014
Fea. 2	-0.023	0.11	Fea. 2	-0.013	0.104
Fold 9	Fea. 1	Fea. 2	Fold 10	Fea. 1	Fea. 2
Fea. 1	0.326	-0.017	Fea. 1	0.276	-0.016
Fea. 2	-0.02	0.099	Fea. 2	-0.014	0.117

รูปที่ 7 Covariance Matrix ของ Class 1 ในแต่ละ Fold

Fold 1	Fea. 1	Fea. 2	Fold 2	Fea. 1	Fea. 2
Fea. 1	0.353	0.022	Fea. 1	0.369	-0.002
Fea. 2	0.02	0.097	Fea. 2	-0.001	0.106
Fold 3	Fea. 1	Fea. 2	Fold 4	Fea. 1	Fea. 2
Fea. 1	0.369	0.004	Fea. 1	0.373	0.01
Fea. 2	0.007	0.105	Fea. 2	0.012	0.105
Fold 5	Fea. 1	Fea. 2	Fold 6	Fea. 1	Fea. 2
Fea. 1	0.389	0.005	Fea. 1	0.387	0
Fea. 2	0.005	0.107	Fea. 2	0.001	0.104
Fold 7	Fea. 1	Fea. 2	Fold 8	Fea. 1	Fea. 2
Fea. 1	0.366	0.001	Fea. 1	0.338	0.02
Fea. 2	0.001	0.103	Fea. 2	0.02	0.096
Fold 9	Fea. 1	Fea. 2	Fold 10	Fea. 1	Fea. 2
Fea. 1	0.36	-0.002	Fea. 1	0.383	-0.012
Fea. 2	-0.005	0.097	Fea. 2	-0.01	0.095

รูปที่ 8 Covariance Matrix ของ Class 1 ในแต่ละ Fold

Fold 1	Predict		Acc	Fold 2	Predict		Acc
Actual	6	4	50	Actual	6	2	65
	6	4			5	7	
Fold 3	Predict		Acc	Fold 4	Predict		Acc
Actual	7	5	60	Actual	7	3	50
	3	5			7	3	
Fold 5	Predict		Acc	Fold 6	Predict		Acc
Actual	5	6	45	Actual	7	2	65
	5	4			5	6	
Fold 7	Predict		Acc	Fold 8	Predict		Acc
Actual	7	4	65	Actual	8	5	65
	3	6			2	5	
Fold 9	Predict		Acc	Fold 10	Predict		Acc
Actual	6	3	55	Actual	4	3	30
	6	5			11	2	
Mean Acc.		55					

รูปที่ 9 Confusion Matrix Test 2 ในแต่ละ fold

จากรูปที่ 9 สำหรับการทดลอง 2 พบว่า มีความแม่นยำเพียงแค่ 55% โดย Fold ที่ 10 มีความแม่นยำต่ำที่สุดเพียงแค่ 30% เท่านั้น

## 5. Analyze the experiment

จากการทดลองพบว่า การทดลองที่ 1 มีความแม่นยำโดยเฉลี่ยถึง 90% และการทดลองที่ 2 มีความแม่นยำ 55% ซึ่งสามารถสังเกตได้จาก ค่าเฉลี่ยของ Feature ในแต่ละ Folds ในการทดลองที่ 1 พบว่า Feature3 และ Feature4 มีค่าที่ต่างกันเห็นได้อย่างชัดเจนในแต่ละ Class ลำดับถัดมาคือ Feature1 มีค่าต่างกันเพียงเล็กน้อย และลำดับสุดท้าย Feature2 มีค่าเฉลี่ยที่แทบจะคล้ายคลึงกัน ดังนั้น Feature ที่เหมาะสมคือ Feature3 และ Feature4 ซึ่งการทดลองที่ 2 ผู้ทดลองได้ทดลองใช้ Feature1 และ Feature2 จึงทำให้ผลลัพธ์ออกมาไม่แม่นยำ

## 6. Appendix

```
import numpy as np

from numpy import linalg as LA
import pandas as pd

# mean
def mean(x): return np.round(x.mean(axis=0),3)

# std
def std(x): return np.round(x.std(axis=0),3)

# covarian-matrix
def cov_matrix(x):
    fact = x.shape[0] - 1
    return np.round(np.dot((x-mean(x)).T, (x-std(x))) * (1/fact), 3)

# multivariate normal distribution
def multi_distribution(X, cov, mean):
    const = ((2*np.pi)**(cov.shape[1]/2))
    cov_norm = LA.norm(cov)**(0.5)

    exp = np.array(list(map(lambda x: np.exp(-0.5*np.dot(np.dot((x-mean), LA.inv(cov)), (x-mean).T)), X)))
    return ((1/(const*cov_norm))*exp)
```

```

# cross_validations
def cross_validations_split(shape,folds):
    fold_size = int(shape * folds/100)
    k = 0
    index = []
    for i in range(1,folds+1):
        index.append([k,i*fold_size]) if (i < folds) else
index.append([k,shape])
        k = i*fold_size
    return index

# probability of Wi
def prob_of_p(n,N):
    return n/N

# for 2 classes
def bayes_rules(f1,f2,p1,p2):
    likelihood_ratio = f1/f2
    threshold = p2/p1
    decision_matrix = (likelihood_ratio > threshold)

    return np.where(decision_matrix,np.float64(1),np.float64(2)).reshape(-1)

# confusion matrix
def confusion_matrix(y_pred,y_true,err = False):

    if y_true.shape != y_pred.shape : return

    def _condition(y_pred,y_true):
        if y_pred == y_true and y_true == 1:
            return "TN"
        elif y_pred != y_true and y_true == 2:
            return "FP"
        elif y_pred != y_true and y_true == 1:
            return "FN"
        return "TP"

    matrix = np.array([[0, 0], [0, 0]])

    for i in range(y_true.shape[0]):
        result = _condition(y_pred[i],y_true[i])
        if result == "TN":
            matrix[0][0] += 1
        elif result == "FN":
            matrix[0][1] += 1
        elif result == "FP":
            matrix[1][0] += 1
        else:
            matrix[1][1] += 1

    if err:
        return matrix,100-(matrix[0][0]+matrix[1][1])*100/y_true.shape[0]
    return matrix

```

```

def preprocess_data(data,i,j):

    population = np.concatenate((data[:,i],data[:,j]))
    samples = data[i:j]

    x_class1 = population[population[:, -1] == 1][:, :-1]
    x_class2 = population[population[:, -1] == 2][:, :-1]

    # calculate P(Wi)
    p1 = prob_of_p(population[population[:, -1] == 1][:, :-1]
1].shape[0],population.shape[0])
    p2 = prob_of_p(population[population[:, -1] == 2][:, :-1]
1].shape[0],population.shape[0])

    # calculate COV(Wi)
    cov_1 = cov_matrix(x_class1)
    cov_2 = cov_matrix(x_class2)

    # calculate mean(Wi)
    mean_1 = mean(x_class1)
    mean_2 = mean(x_class2)

    pre_data = {
        'population' : population,
        'x_sample' : samples[:, :-1],
        'x_class1' : x_class1, # separate the data to class 1
        'x_class2' : x_class2, # separate the data to class 2
        'p1': p1,
        'p2': p2,
        'y_sample': samples[:, -1],
        'cov1': cov_1,
        'cov2': cov_2,
        'mean1': mean_1,
        'mean2': mean_2,
    }
    return pre_data

```



```

# main
if __name__ == "__main__":

    # split features and classes to two classes
    data = np.genfromtxt('TWOCLASS.csv',delimiter=',')[1:,:]
    np.random.shuffle(data) # shuffle data

    k = 1
    for i,j in cross_validations_split(data.shape[0],10):

        # * ----- preprocess data -----
        x1 = preprocess_data(data,i,j) # for test 1
        x2 = preprocess_data(data[:,[0,1,-1]],i,j) # for test 2

        # calculate multivariate normal distribution test 1
        fx1_1 = multi_distribution(x1['x_sample'],x1['cov1'],x1['mean1'])
        fx1_2 = multi_distribution(x1['x_sample'],x1['cov2'],x1['mean2'])

        # calculate multivariate normal distribution test 2
        fx2_1 = multi_distribution(x2['x_sample'],x2['cov1'],x2['mean1'])
        fx2_2 = multi_distribution(x2['x_sample'],x2['cov2'],x2['mean2'])

        # evaluate test 1
        y_pred1 = bayes_rules(fx1_1,fx1_2,x1['p1'],x1['p2'])
        y_true1 = x1['y_sample']

        # evaluate test 2
        y_pred2 = bayes_rules(fx2_1,fx2_2,x2['p1'],x2['p2'])
        y_true2 = x2['y_sample']

    print("##### K=", k , " #####")
    k+=1

```