

Machine Learning

Team member

- | | | |
|----|---------------------------------|----------------------|
| 1. | นายคุณานนท์ เจริญดี | 6401012620013 |
| 2. | นายแสง รงยง | 6401012620242 |
| 3. | นายธนชาติ โพธิหลักทรัพย์ | 6401012630051 |
| 4. | นายนรินทร์ ศิริณภัค | 6401012630086 |

Week 1 Linear Regression

1) เขียนโปรแกรมสำหรับสร้างแบบจำลองเชิงเส้นด้วยวิธีลดความชัน
พร้อมทั้งแสดงฟังก์ชันค่าใช้จ่าย ในรูปของคอนทัวร์และแสดงให้เห็น
ถึงขั้นตอนในการปรับพารามิเตอร์

Hypothesis function

```
def hypothesis(x,y):
    w0,w1 = Gradeint_Descent(25,25,x,y)
    return w0 + w1*x
```

Gradient Descent

```
def Gradeint_Descent(w0,w1,x,y):
    alpha = 0.5
    mse = 1000
    while mse > 0.001:
        w0 = w0 - (alpha/len(x)) * Fw0(w0,w1,x)
        w1 = w1 - (alpha/len(x)) * Fw1(w0,w1,x)
        mse = MSE(w0,w1,x,y)
        list_w0.append(w0)
        list_w1.append(w1)
    return w0,w1
```

1) เขียนโปรแกรมสำหรับสร้างแบบจำลองเชิงเส้นด้วยวิธีลดความชัน
พร้อมทั้งแสดงฟังก์ชันค่าใช้จ่าย ในรูปของคอนทัวร์และแสดงให้เห็น
ถึงขั้นตอนในการปรับพารามิเตอร์

Gradient Descent

ปรับค่า w_0 และ w_1

$$w_0 := w_0 - \alpha \frac{1}{n} \sum_{i=1}^n (h(x^{(i)}) - y^{(i)})$$

$$w_1 := w_1 - \alpha \frac{1}{n} \sum_{i=1}^n (h(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

```
def Fw0(w0,w1,x):
    S = 0
    for i in range(len(x)):
        S = S + ((w0 + w1*x[i])-y[i])
    return S

def Fw1(w0,w1,x):
    S = 0
    for i in range(len(x)):
        S = S + ((w0 + w1*x[i])-y[i]) * x[i]
    return S
```

1) เขียนโปรแกรมสำหรับสร้างแบบจำลองเชิงเส้นด้วยวิธีลดความชัน
พร้อมทั้งแสดงฟังก์ชันค่าใช้จ่าย ในรูปของคอนกรีตและแสดงให้เห็น
ถึงขั้นตอนในการปรับพารามิเตอร์

Mean Squared Error: MSE

$$\frac{1}{2n} \sum_{i=1}^n (h(x^{(i)}) - y^{(i)})^2$$

Cost function

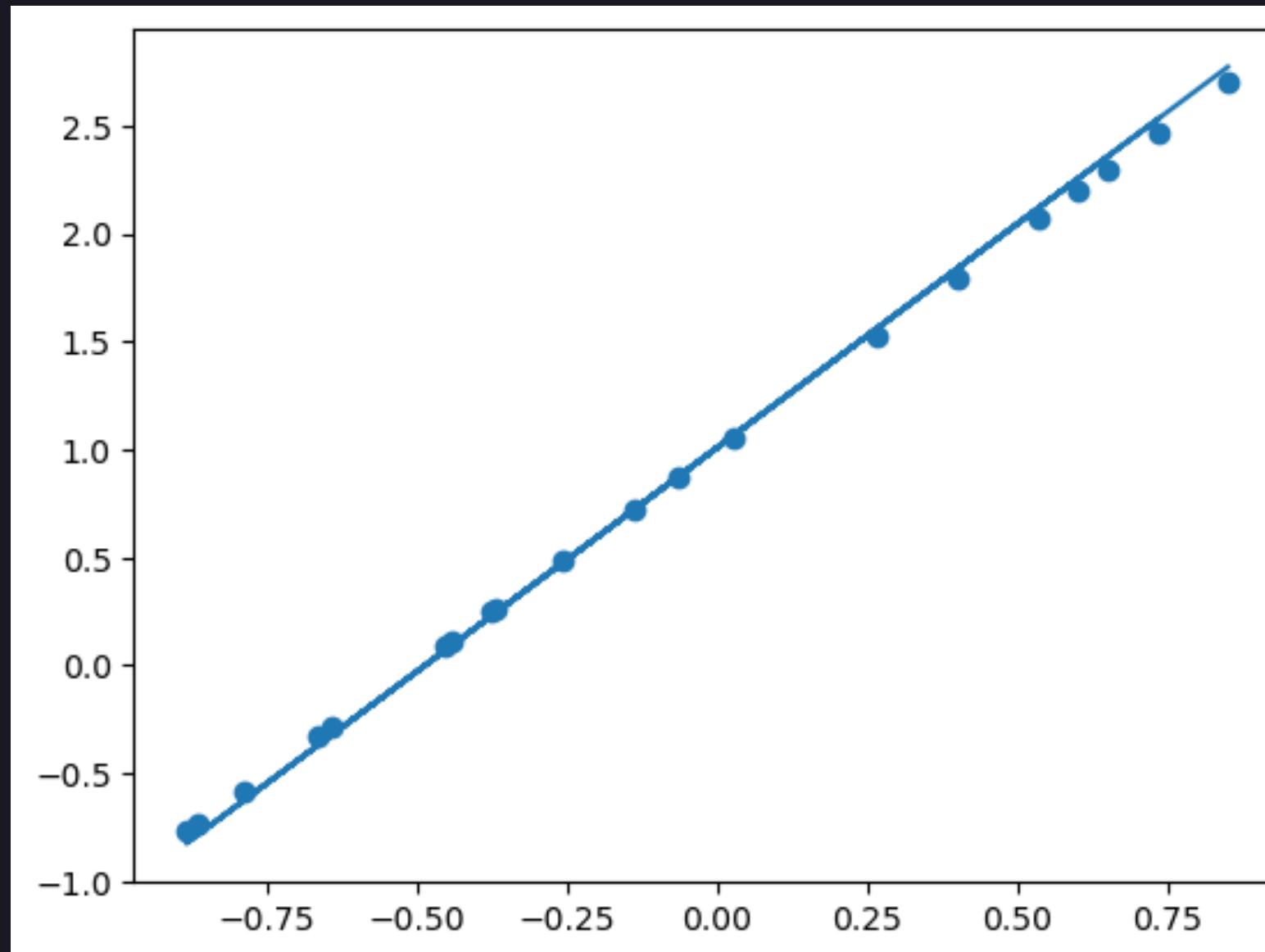
```
def MSE(w0,w1,x,y):
    S = 0
    for i in range(len(x)):
        S = S + ((w0 + w1*x[i])-y[i])**2
    return S/(2*len(x))
```

Input `x = np.random.uniform(-1, 1, size=(20, 1))`

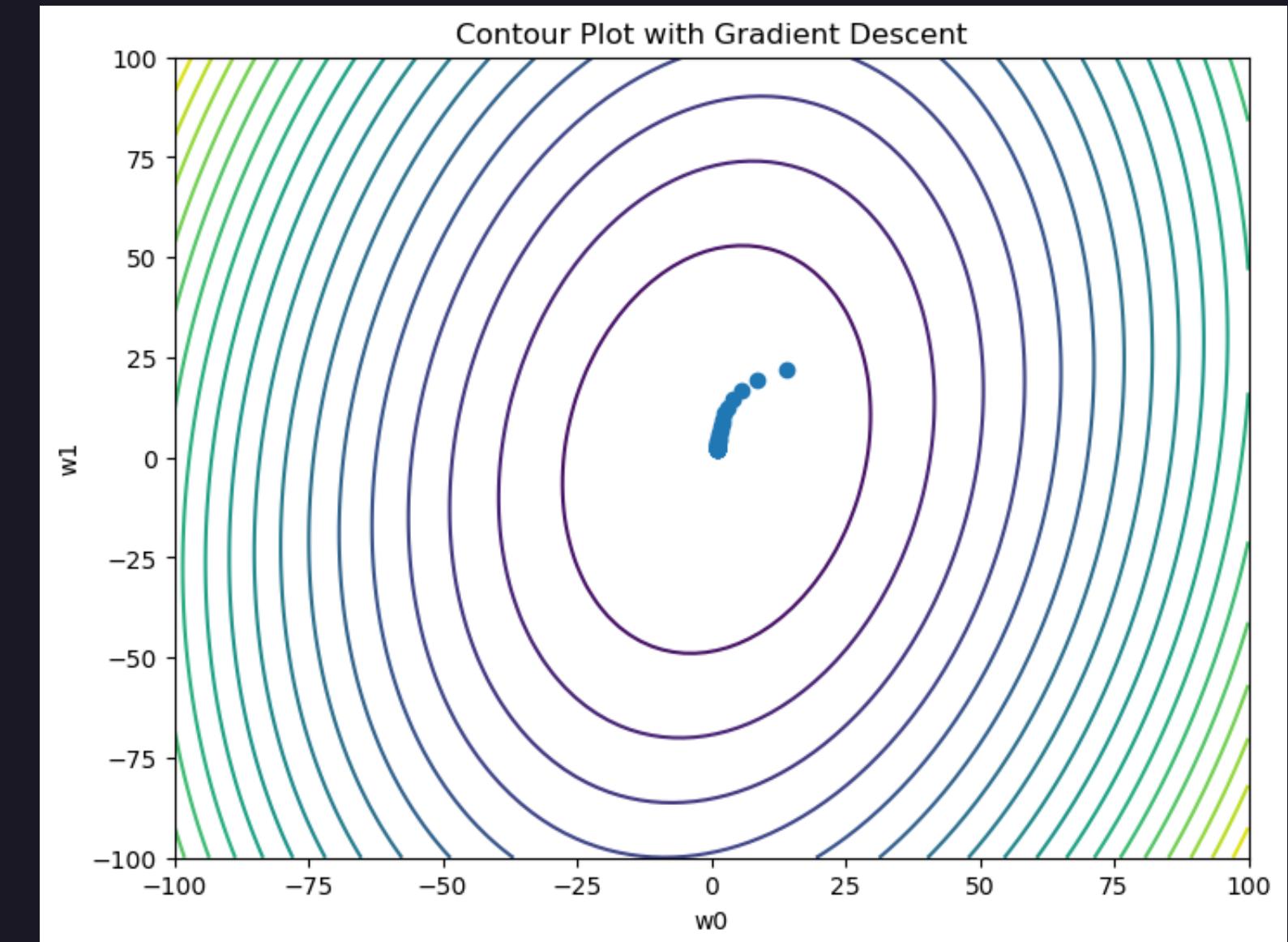
$$y = 2*x + 1$$

Output

Linear Regression



Contour Plot



2) เขียนโปรแกรมสำหรับแสดงผลกราฟต่อการทำงานของวิธีลดตาม
ความชันและพิงค์ชันค่าใช้จ่าย เมื่อตัวแปร x หลายตัวมีค่าต่างกันมาก
และแสดงผลของการปรับปรุงประสิทธิภาพ ด้วยการทำให้เป็นมาตรฐาน

Standardize function

```
def standardization(x):
    x_mean = np.mean(x)
    x_std = np.std(x)
    standardized_x = (x - x_mean) / x_std
    return standardized_x
```

Cost function

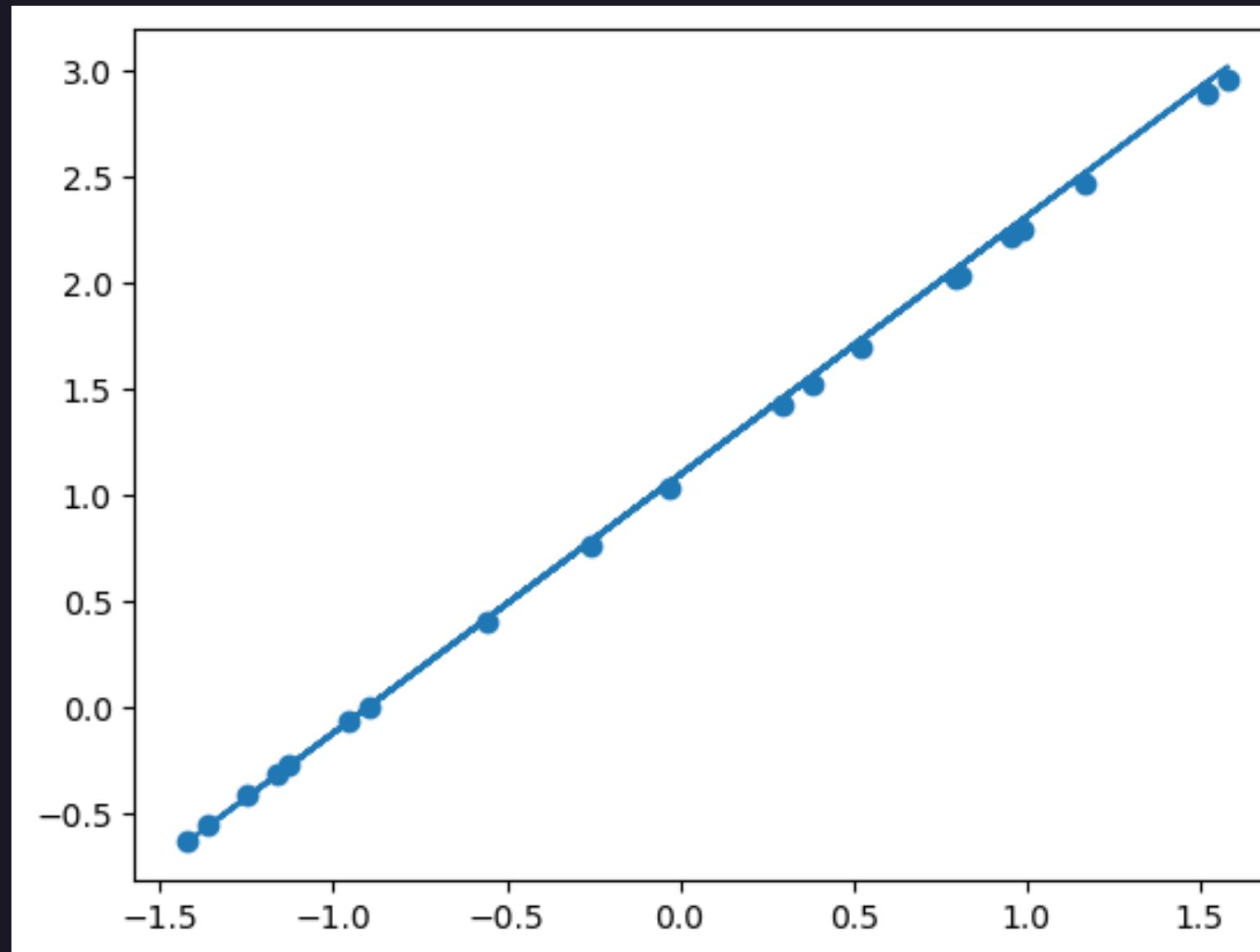
```
def MSE(w0,w1,x,y):
    S = 0
    for i in range(len(x)):
        S = S + ((w0 + w1*x[i])-y[i])**2
    return S/(2*len(x))
```

Input

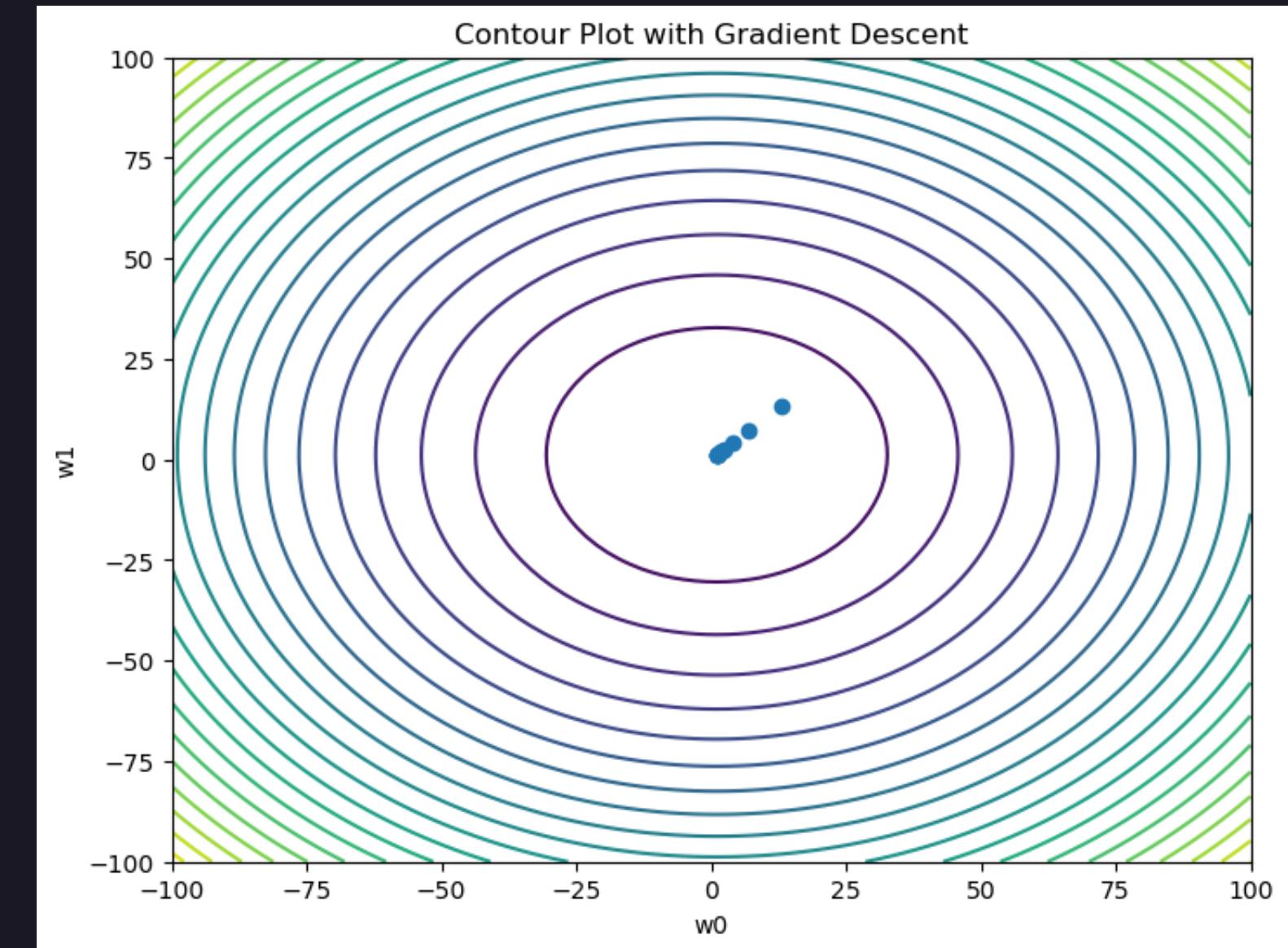
```
x = np.random.uniform(-1, 1, size=(20, 1))  
y = 2*x+1
```

Output

Linear Regression



Contour Plot



2) เขียนโปรแกรมสำหรับแสดงผลกราฟต่อการทำงานของวิธีลดตามความชันและพิงก์ชันค่าใช้จ่าย เมื่อตัวแปร x หลายตัวมีค่าต่างกันมาก และแสดงผลของการปรับปรุงประสิทธิภาพ ด้วยการทำให้เป็นมาตรฐาน

Input

```
X = np.array([[7,7,8,9],  
              [5,3,3,2],  
              [1,2,2,1],  
              [6,6,4,3]])
```

Standardized
Multi X

```
def standardized(X):  
    n = len(X)  
    m = len(X[0])  
    X_std = np.empty((n, m))  
  
    for i in range(n):  
        sigma = np.std(X[i])  
        mean = np.mean(X[i])  
        for j in range(m):  
            X_std[i][j] = (X[i][j] - mean) / sigma  
  
    return X_std
```

Output

```
[[ -0.90453403 -0.90453403  0.30151134  1.50755672]  
 [ 1.60591014 -0.22941573 -0.22941573 -1.14707867]  
 [-1.          1.          1.          -1.          ]  
 [ 0.96225045  0.96225045 -0.57735027 -1.34715063]]
```

3) เขียนโปรแกรมสำหรับแสดงผลของการปรับพารามิเตอร์การเรียนรู้

```
def display_results(learning_rate):
    w0, w1 = Gradient_Descent(1, 0, learning_rate)      #w0,w1 = 1,0 is init value

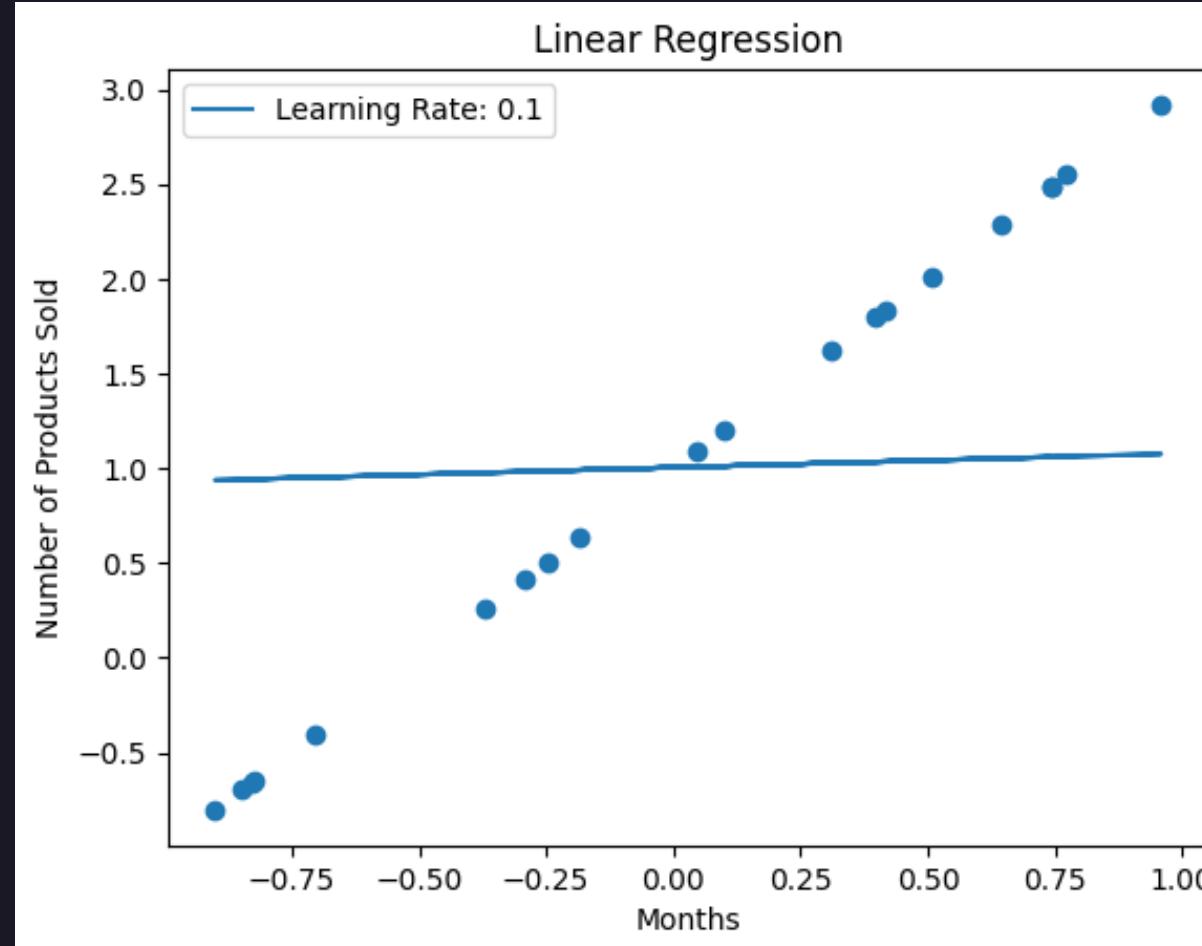
    plt.scatter(x, y)
    plt.plot(x, [w0 + w1 * i for i in x], label=f'Learning Rate: {learning_rate}')
    plt.xlabel('Months')
    plt.ylabel('Number of Products Sold')
    plt.title('Linear Regression')
    plt.legend()
    plt.show()

learning_rates = [0.1, 0.5, 1, 1.5, 1.75, 2]

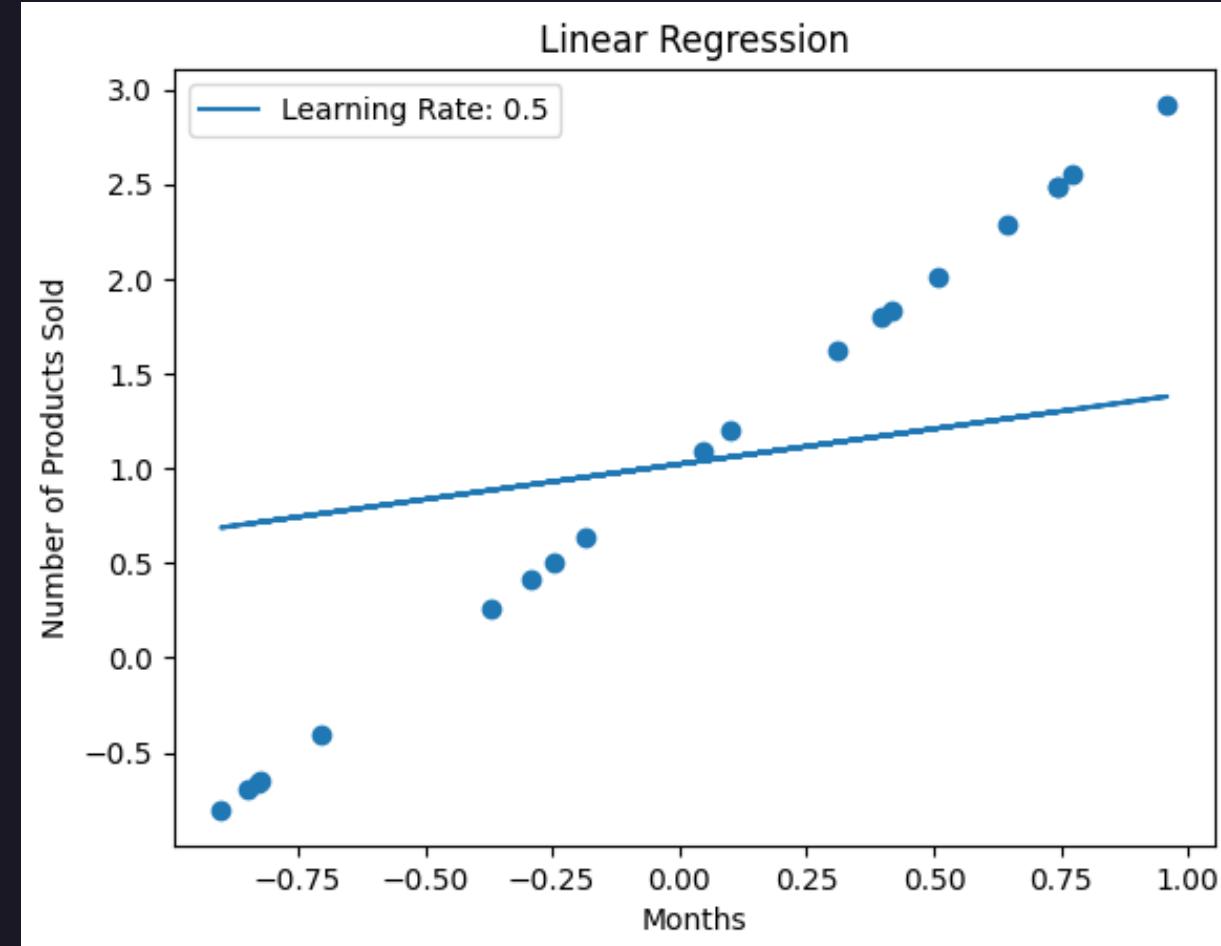
for lr in learning_rates:
    display_results(lr)
```

Output

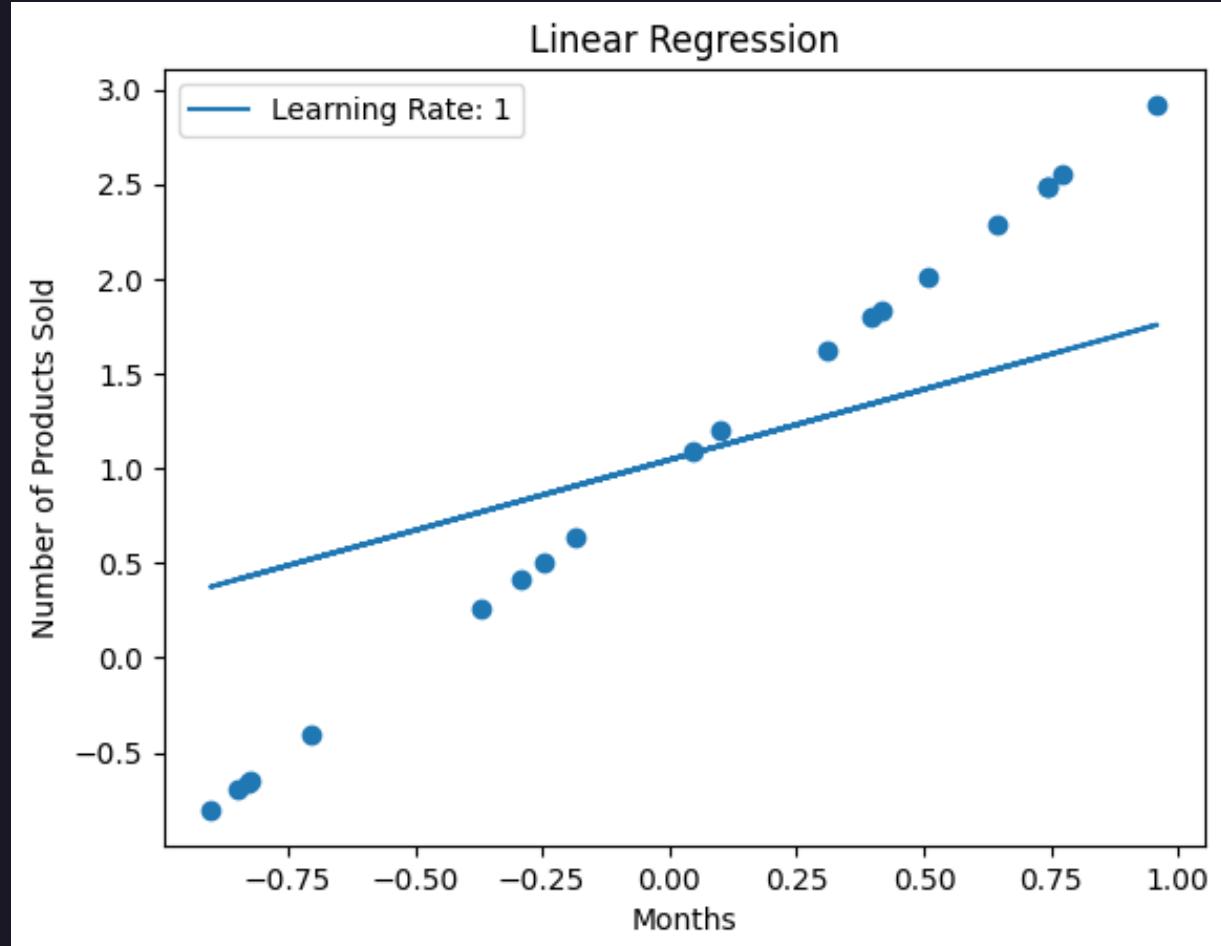
Learning rate: 0.1



Learning rate: 0.5

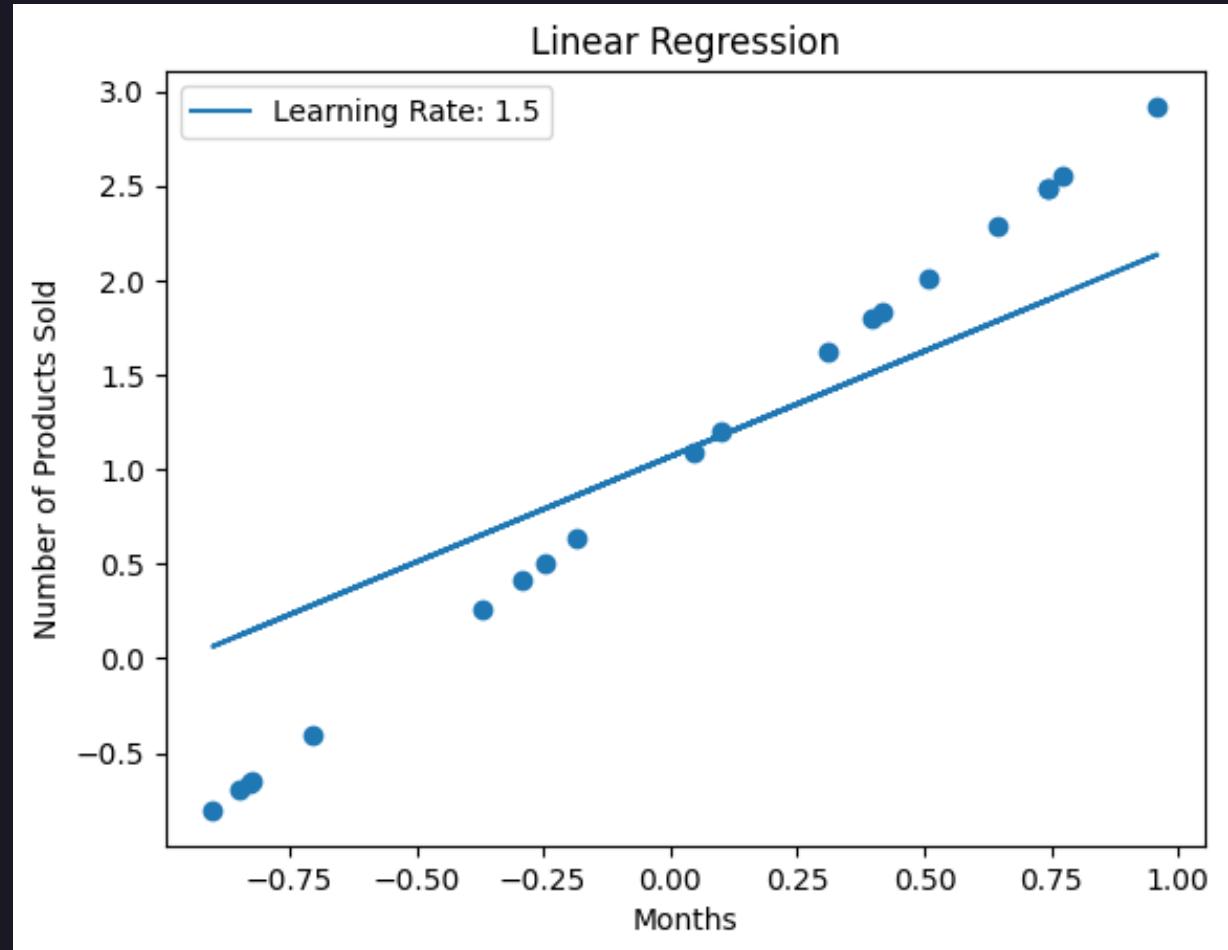


Learning rate: 1

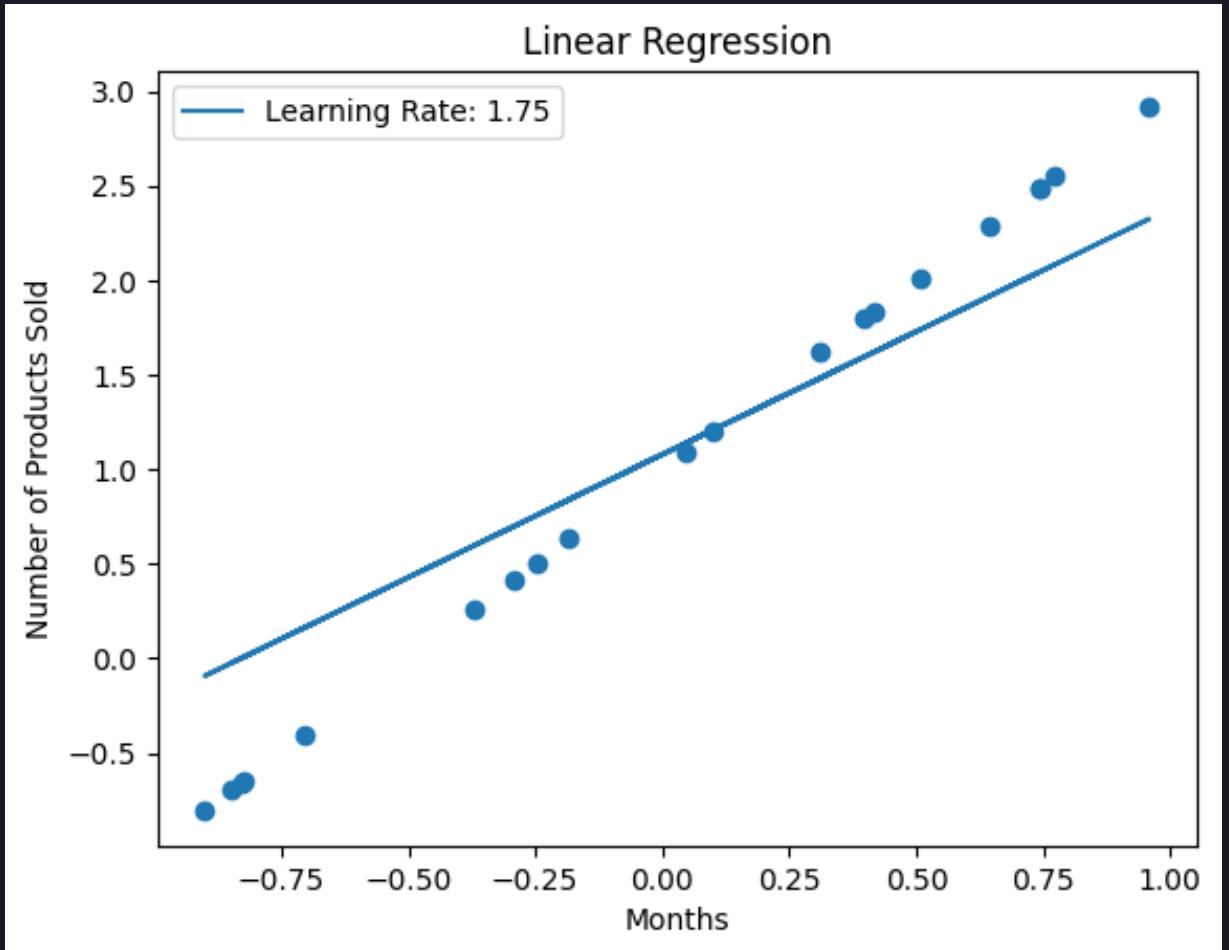


Output

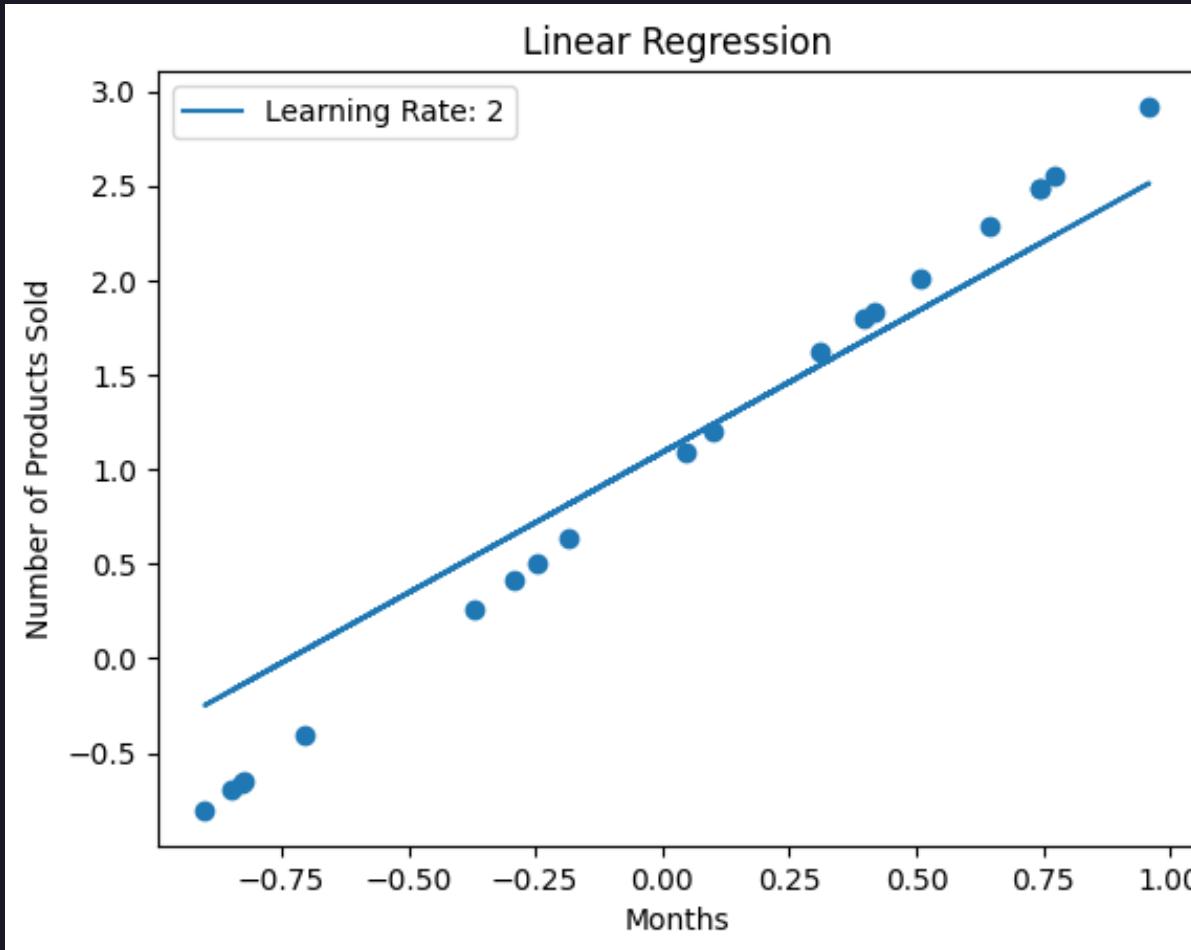
Learning rate: 1.5



Learning rate: 1.75



Learning rate: 2



4) เขียนโปรแกรมสำหรับเปรียบเทียบผลลัพธ์ที่ได้จากวิธี สมการปกติ และ วิธีลดตามความชัน

$$\mathbf{w} := \mathbf{w} - \frac{\alpha}{n} \mathbf{X}^T (\mathbf{X}\mathbf{w} - \mathbf{y})$$

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Gradient Descent function

```
def Gradeint_Descent(X,Y):
    alpha = 0.001
    m = len(Y)
    n = X.shape[1]
    w = np.zeros(n)
    mse = 100000
    while mse > 1 :
        y_pred = np.dot(X,w)
        mse = MSE(Y,y_pred)
        gradients = (1/m) * np.dot(X.T,(y_pred - Y))
        w = w - alpha * gradients
    return w
```

Normal Equation function

```
def Normal_Equation(X,Y):
    A = np.dot(X.T, X)
    B = np.dot(X.T,Y)
    A_I = np.linalg.inv(A)

    W = np.dot(A_I,B)
    return W
```

4) เขียนโปรแกรมสำหรับเปรียบเทียบผลลัพธ์ที่ได้จากวิธี สมการปกติ และ วิธีลดตามความชัน

Mean Squared Error: MSE

$$\frac{1}{2n} \sum_{i=1}^n (h(x^{(i)}) - y^{(i)})^2$$

```
def MSE (y,y_pred):  
    m = len(y)  
    mse = (1/2*m) * np.sum(y_pred-y)**2  
    return mse
```

Input $x = [[5, 3, 3, 2], [1, 2, 2, 1], [6, 6, 4, 3]]$

$y = [460, 232, 315, 178]$

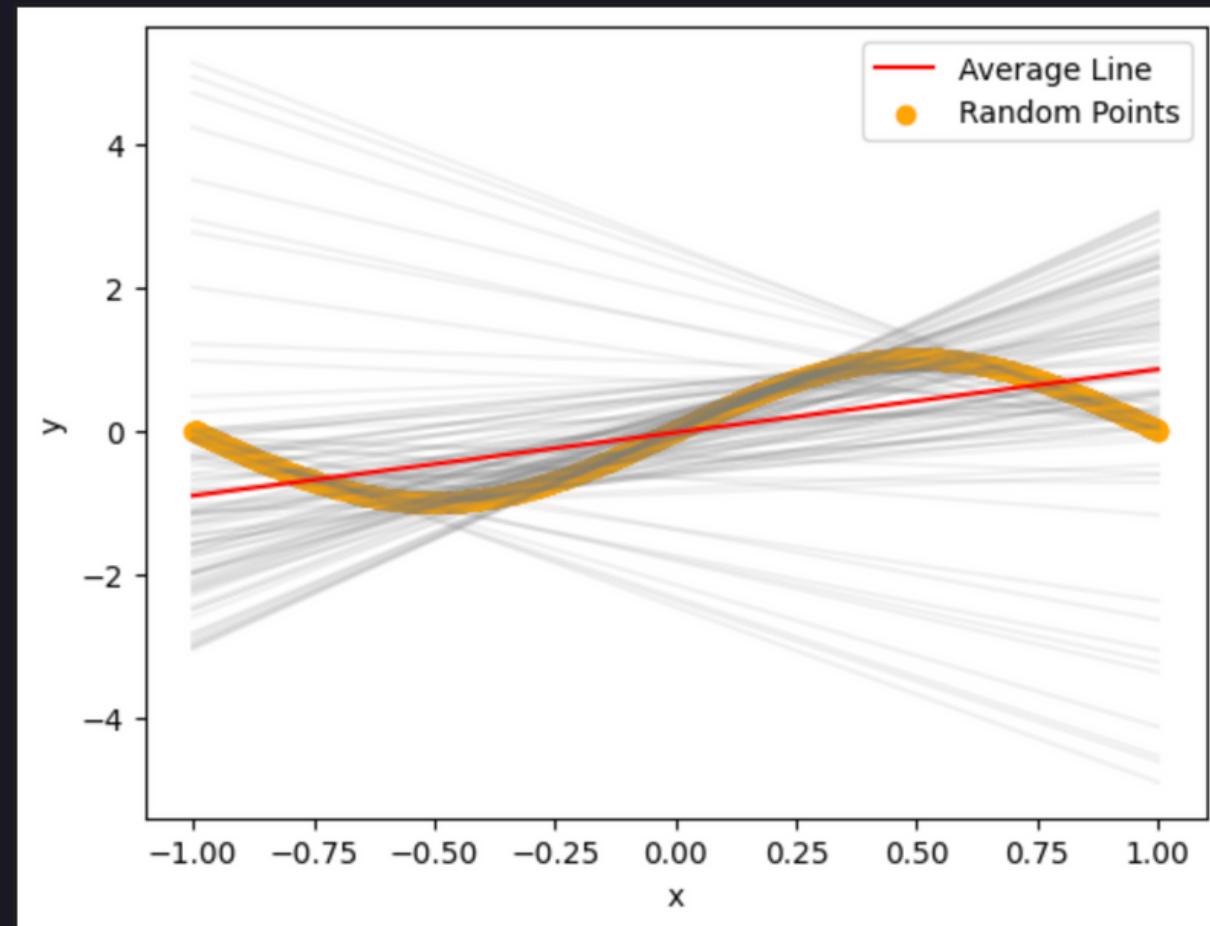
Output

```
Form Gradient Descent W =  [-48.24373645  -57.5743439   309.7968312  -141.95226048]  
Form Normal Equation W =  [-48.34328358  -58.01492537  310.98507463  -142.59701493]
```

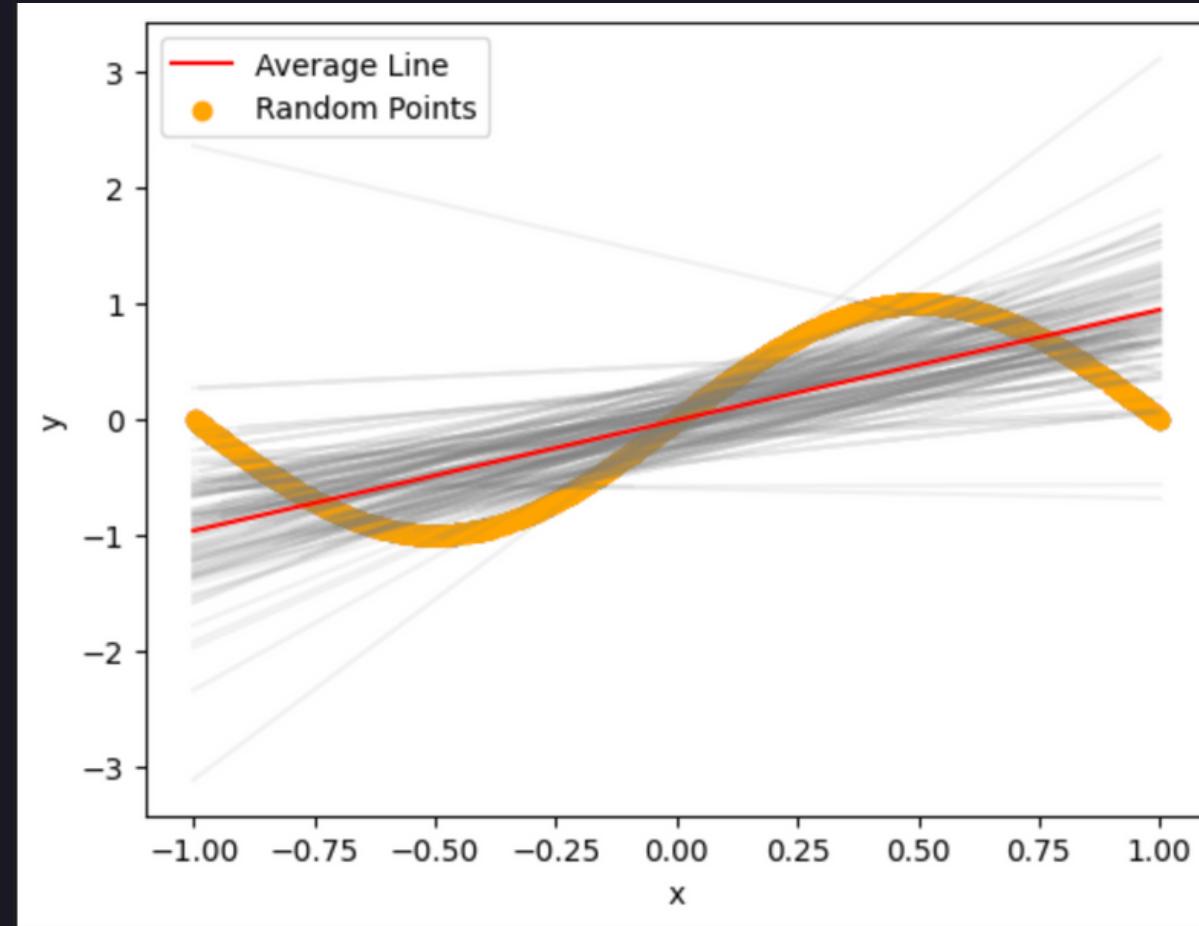
Week 2 Generalization

Linear model

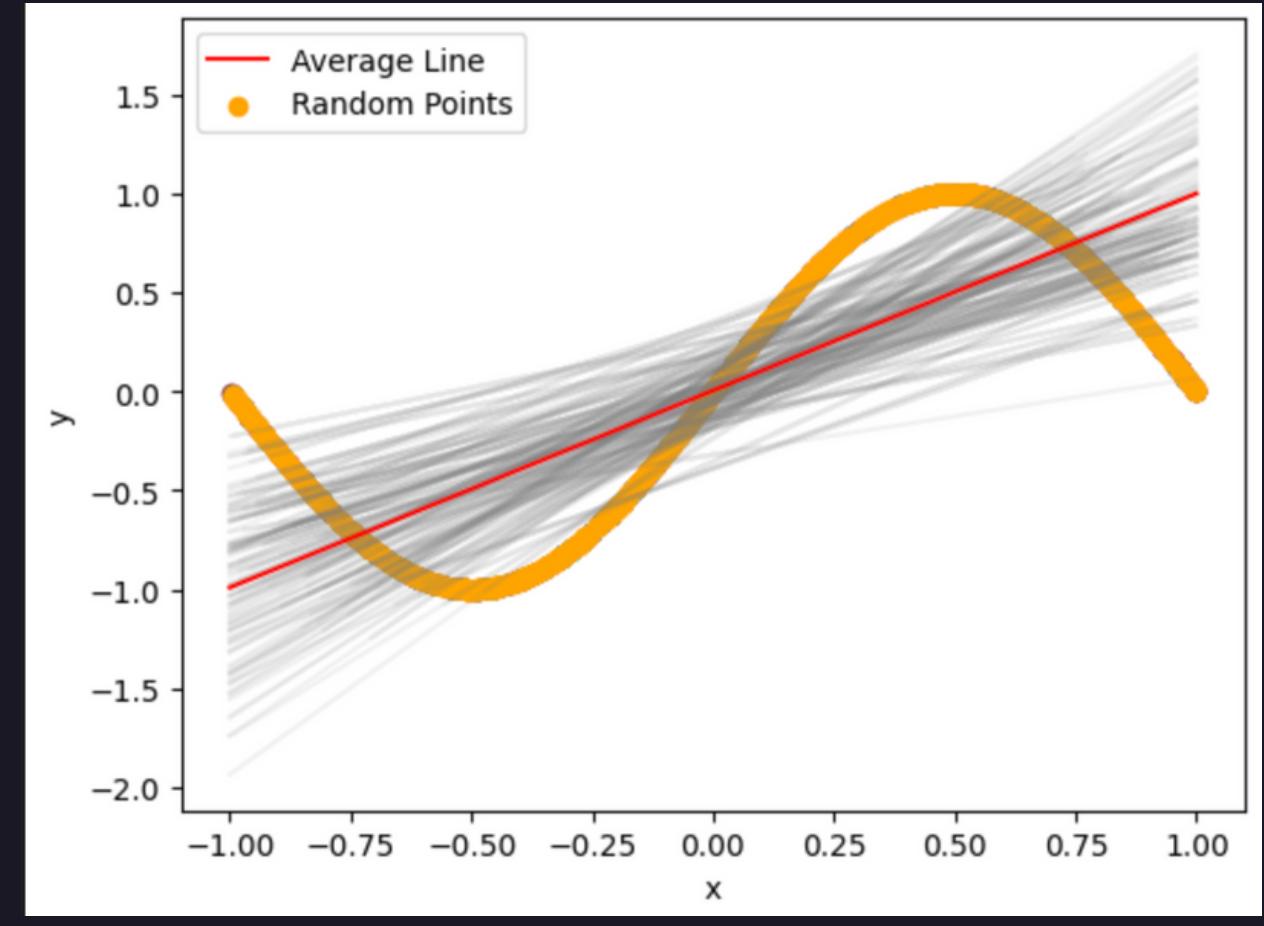
$n = 2$



$n = 4$



$n = 6$



1.1 เมื่อกำหนดให้ฟังก์ชันเป้าหมายคือ $\sin(\pi * x)$ และสุ่มข้อมูลด้วยการแจกแจงแบบเอกรูป ออกมา 2 ตัวอย่างในช่วง $[-1,1]$

Result of Analytic Method

gD(x) function

$$\sin(\pi x_1) + \frac{(-x + x_1) \sin(\pi x_1) + (x - x_1) \sin(\pi x_2)}{-x_1 + x_2}$$

Varaince is 1.6762875
Bias is 0.2067167

Linear Regression

$$\frac{x(x_1 \sin(\pi x_1) + x_2 \sin(\pi x_2))}{x_1^2 + x_2^2}$$

Varaince is 0.23657625
Bias is 0.27064

Through the origin

$$\frac{\sin(\pi x_1)}{2} + \frac{\sin(\pi x_2)}{2}$$

Varaince is 0.25
Bias is 0.5

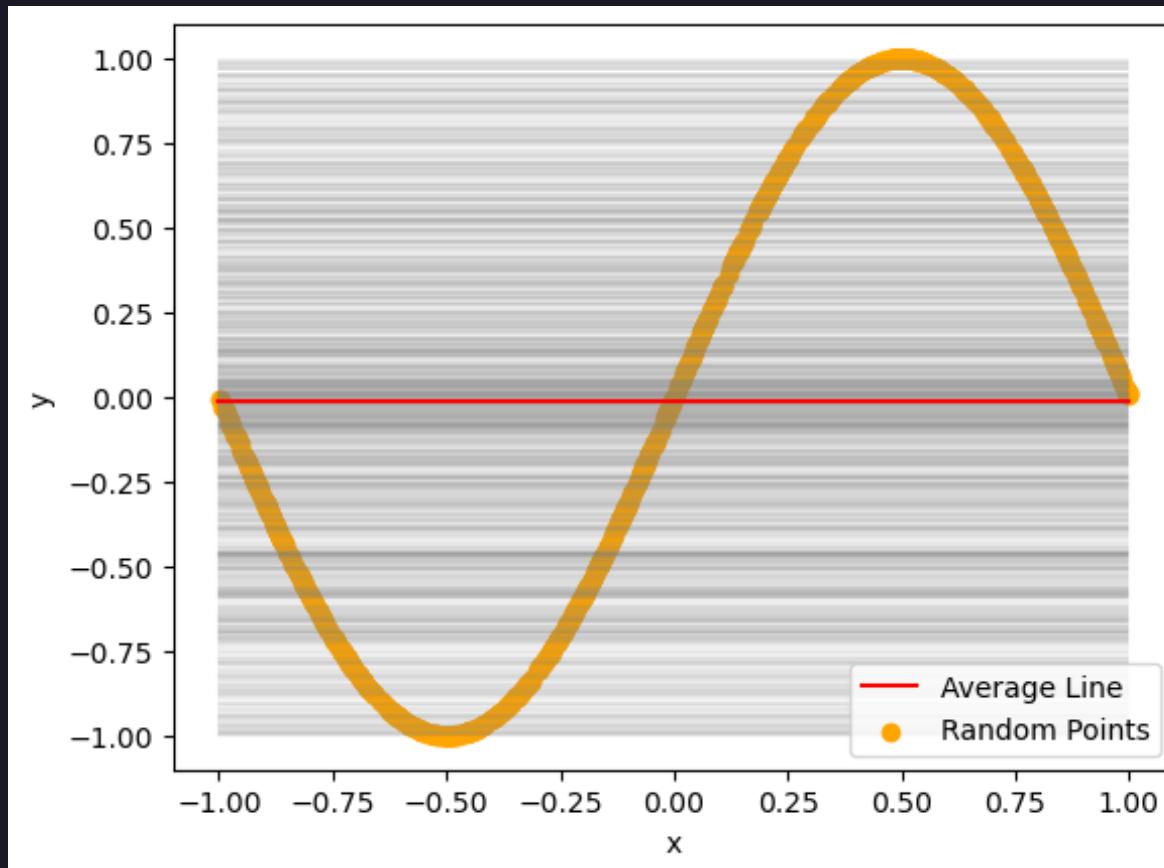
Constant Model

1.1 เมื่อกำหนดให้ฟังก์ชันเป้าหมายคือ $\sin(\pi \cdot x)$ และสุ่มข้อมูลด้วยการแจกแจงแบบเอกรูปอوكมา 2 ตัวอย่างในช่วง $[-1,1]$

Result of Simulation

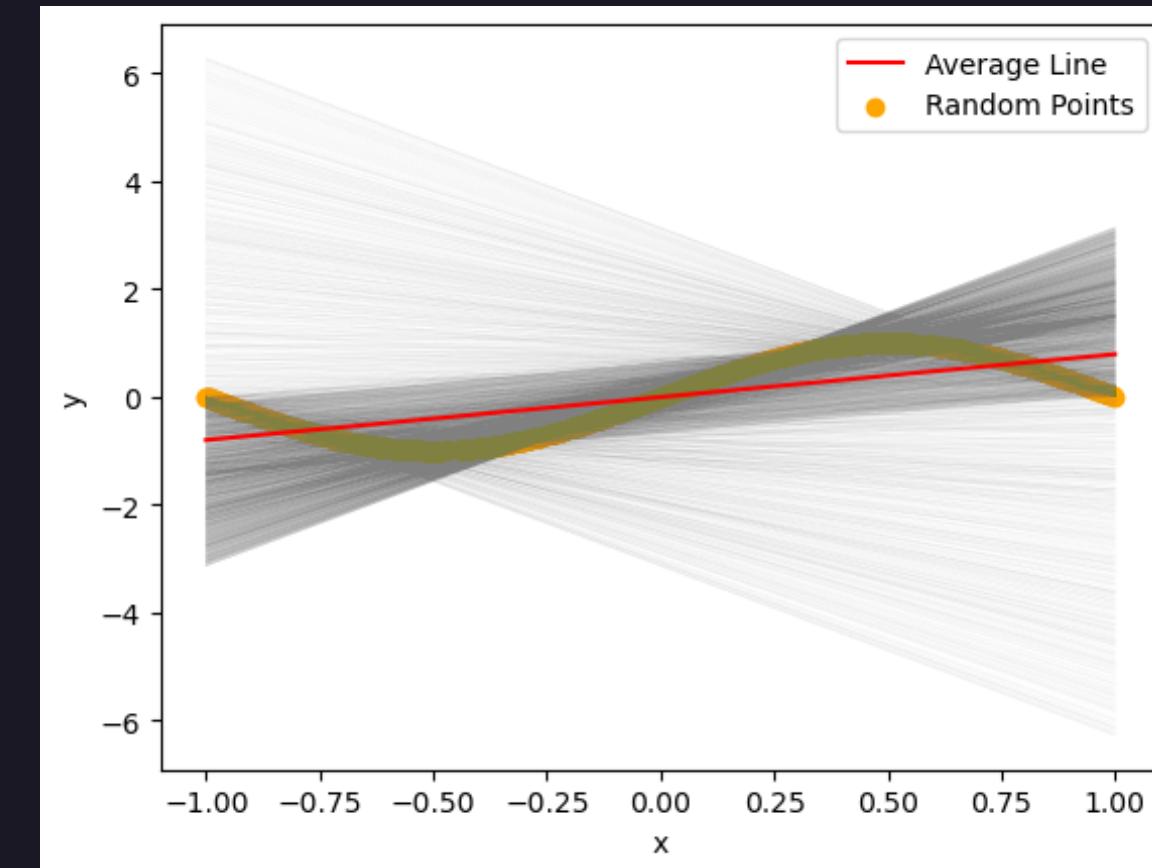
Constant Model

Bias value: 0.4997
variant value: 0.2431



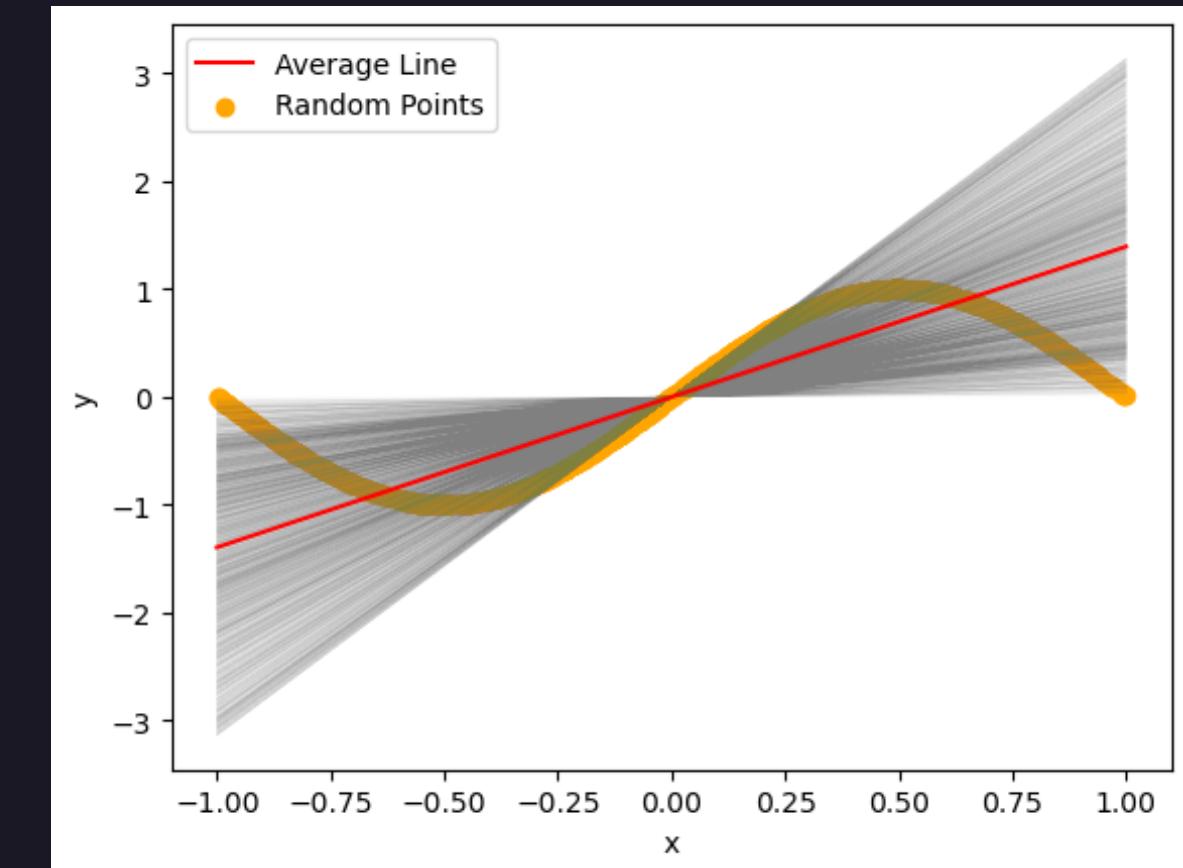
Linear Regression

Bias value: 0.2044
variant value: 1.6352



Through the origin

Bias value: 0.2871
variant value: 0.2327



1.2 เมื่อกำหนดให้ฟังก์ชันเป้าหมายคือ x^2 และสุ่มข้อมูลด้วยการแจกแจงแบบเอกรูปอ กมา 2 ตัวอย่างในช่วง $[-1,1]$

Result of Analytic Method

gD(x) function

$$x_1^2 + \frac{-x_1^2(x - x_1) + x_2^2(x - x_1)}{-x_1 + x_2}$$

Varaince is 0.33333375
Bias is 0.2

Linear Regression

$$\frac{x(x_1^3 + x_2^3)}{x_1^2 + x_2^2}$$

Varaince is 0.11492125
Bias is 0.2

Through the origin

$$\frac{x_1^2}{2} + \frac{x_2^2}{2}$$

Varaince is 0.04444
Bias is 0.08888

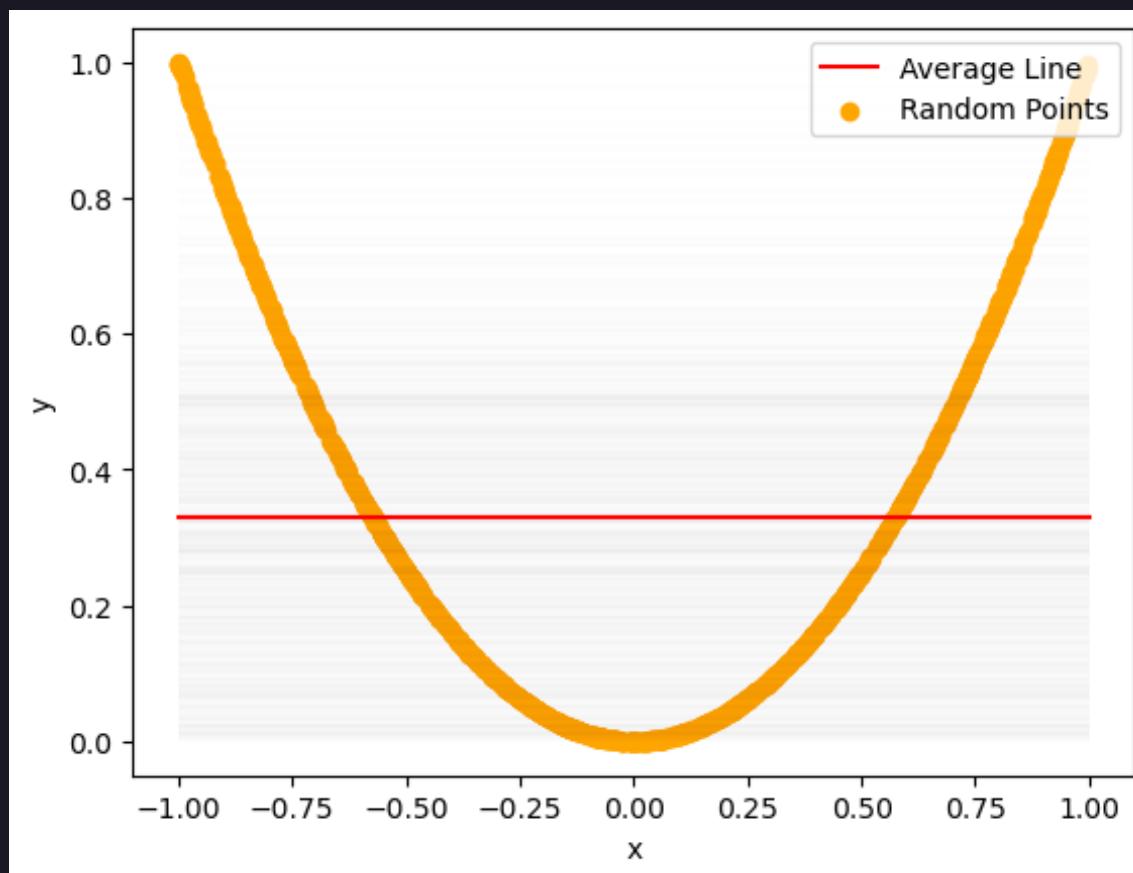
Constant Model

1.2 เมื่อกำหนดให้ฟังก์ชันเป้าหมายคือ x^2 และสุ่มข้อมูลด้วยการแจกแจงแบบเอกรูปอوكมา 2 ตัวอย่างในช่วง $[-1,1]$

Result of Simulation

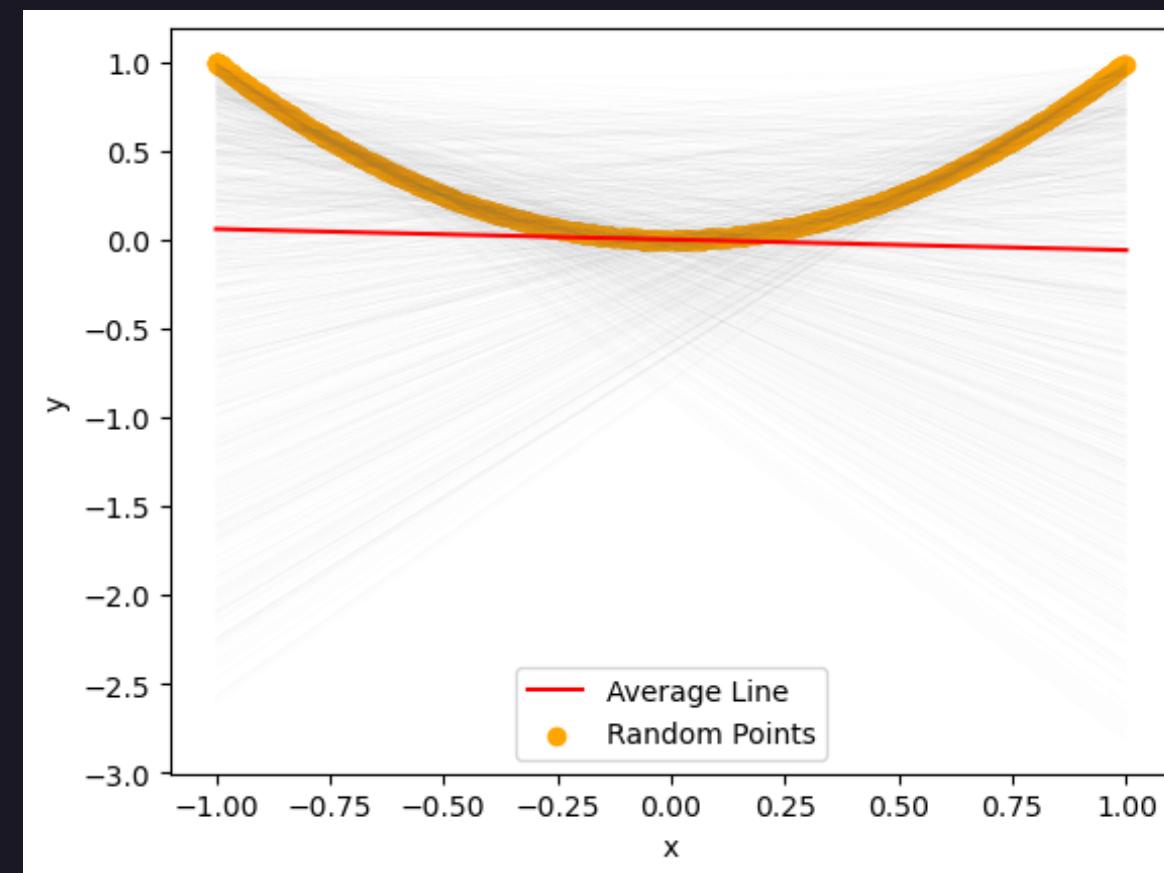
Constant Model

Bias value: 0.0894
variant value: 0.0393



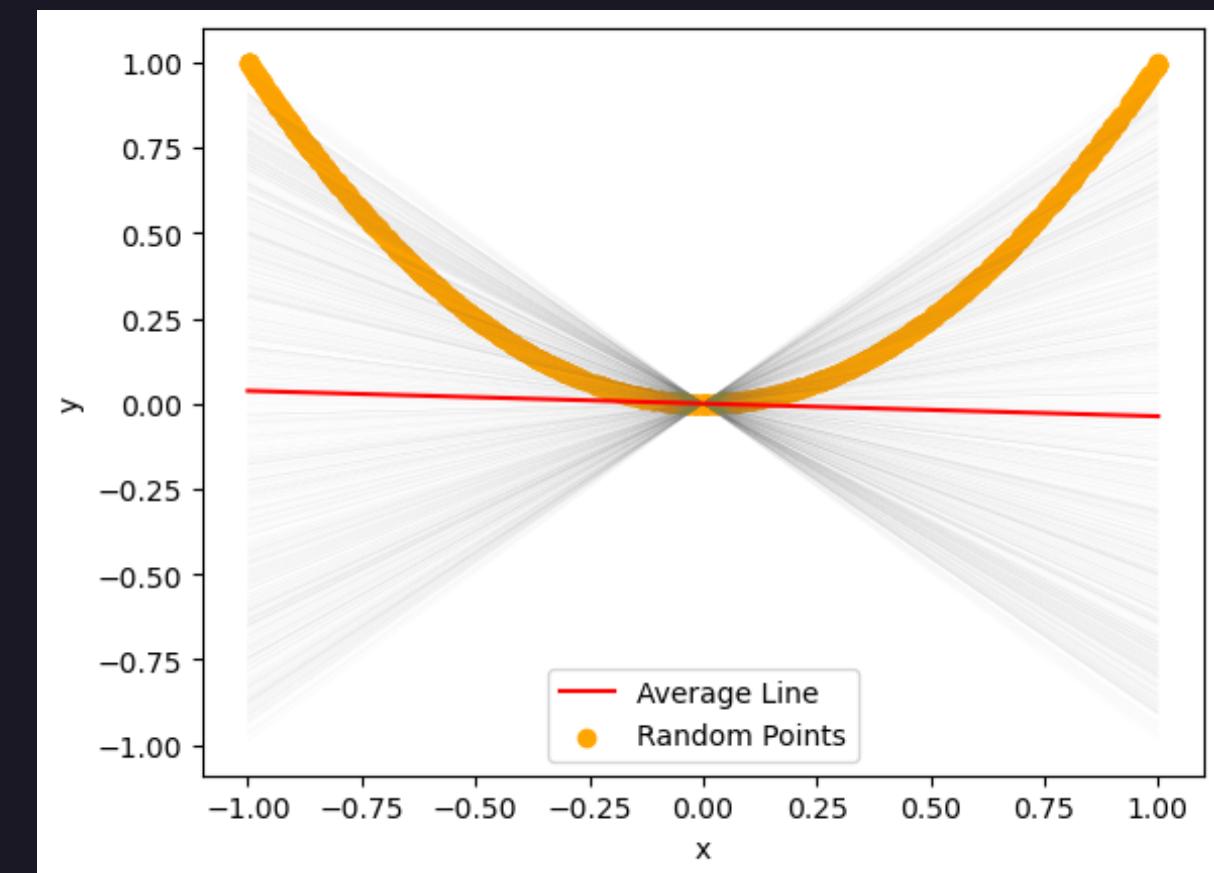
Linear Regression

Bias value: 0.1973
variant value: 0.3224



Through the origin

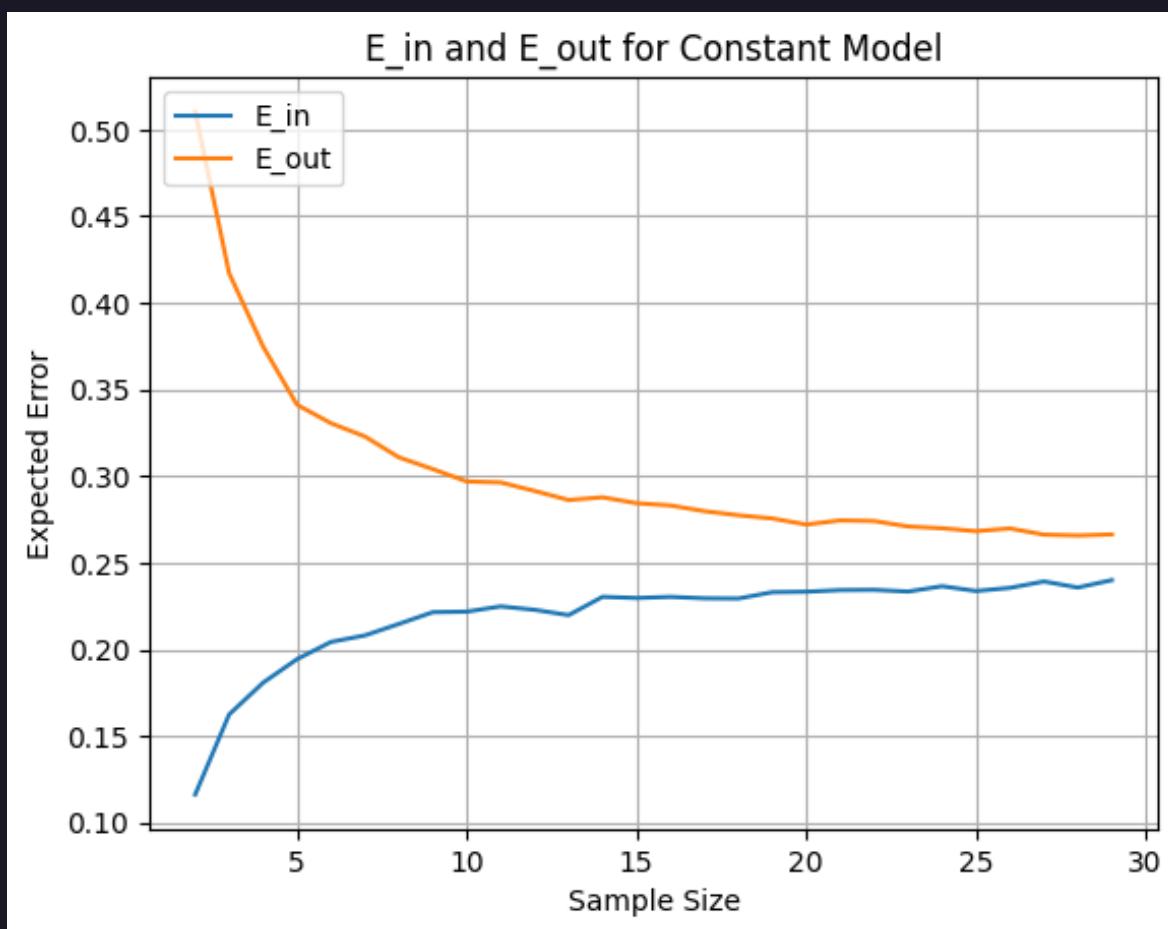
Bias value: 0.2008
variant value: 0.1104



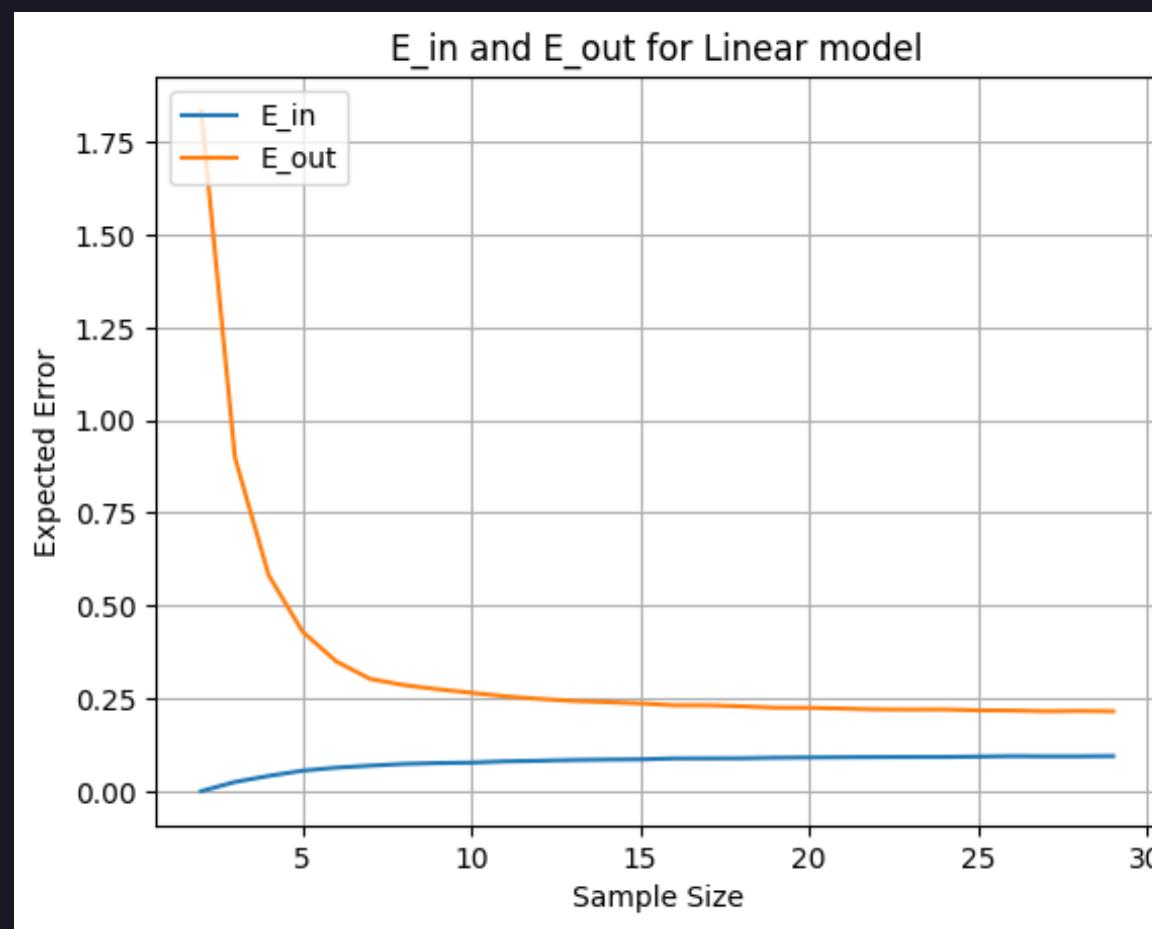
2.เขียนโปรแกรมสำหรับเส้นโค้งการเรียนรู้เปรียบเทียบระหว่างแบบจำลองค่าคงที่ แบบจำลองเชิงเส้น แบบจำลองเชิงเส้นผ่านจุดกำเนิดและทดลองเพิ่มเติมด้วยการใส่สัญญาณรบกวน

Result of Simulation

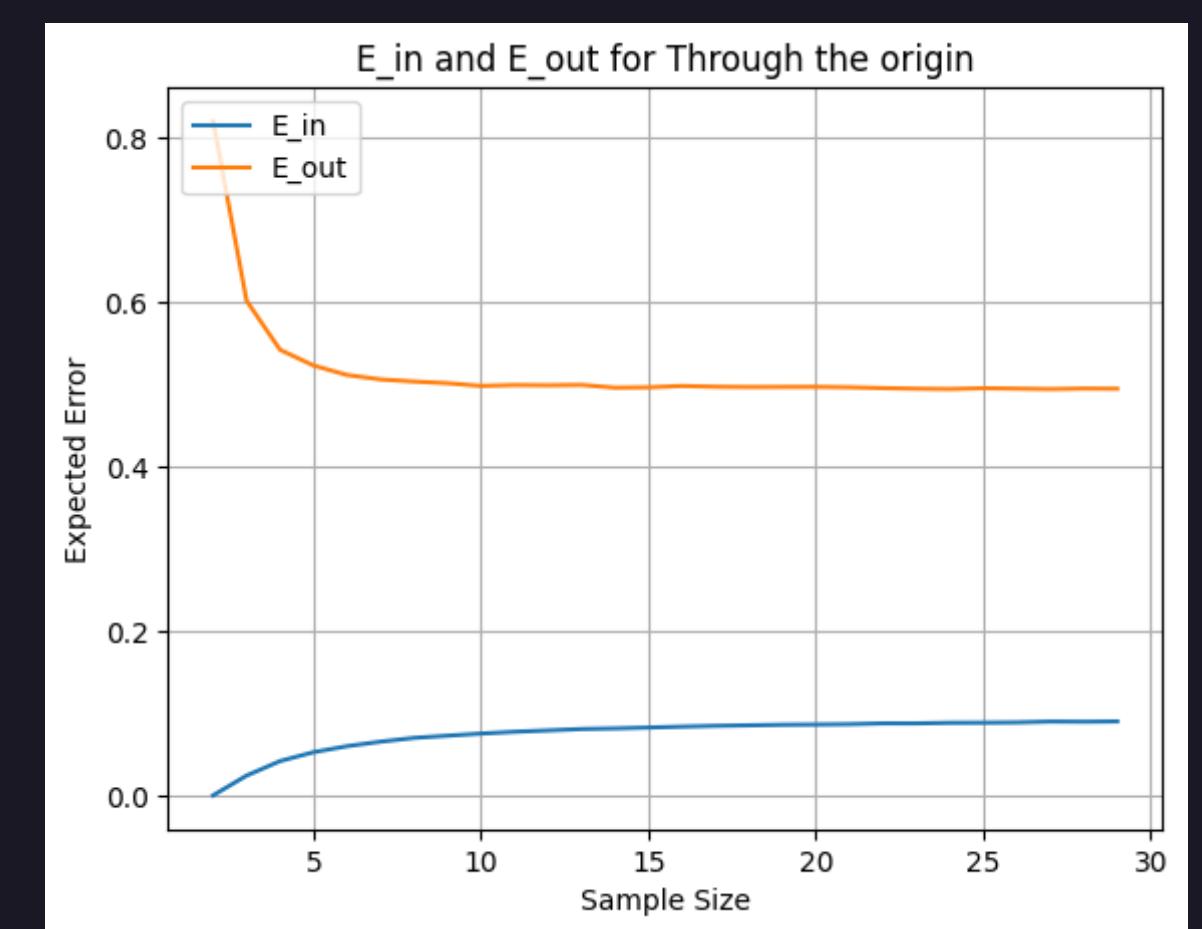
Constant Model



Linear model

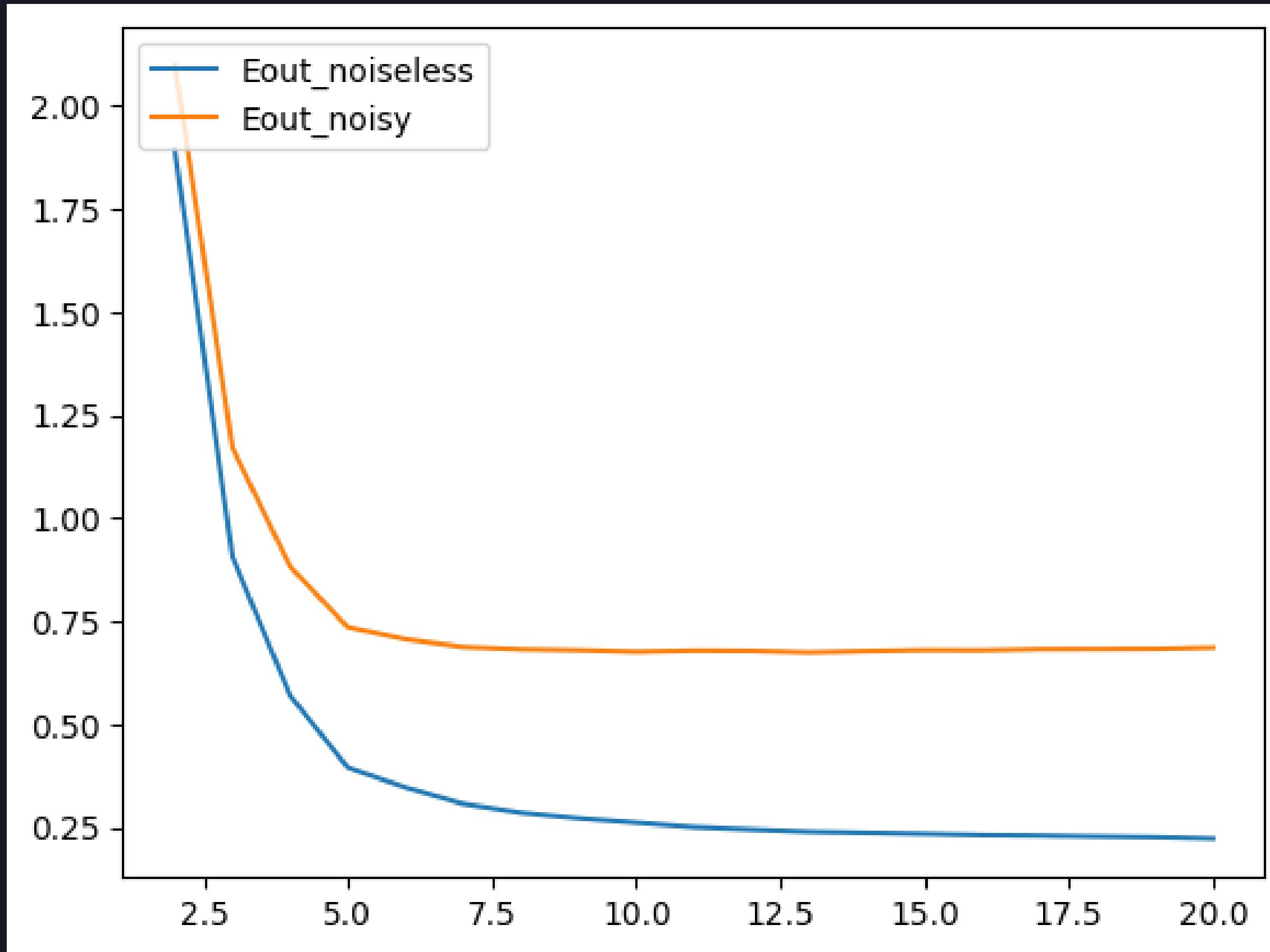


Through the origin



*function : $y = \sin(\pi \cdot x)$

Noiseless Vs Noisy

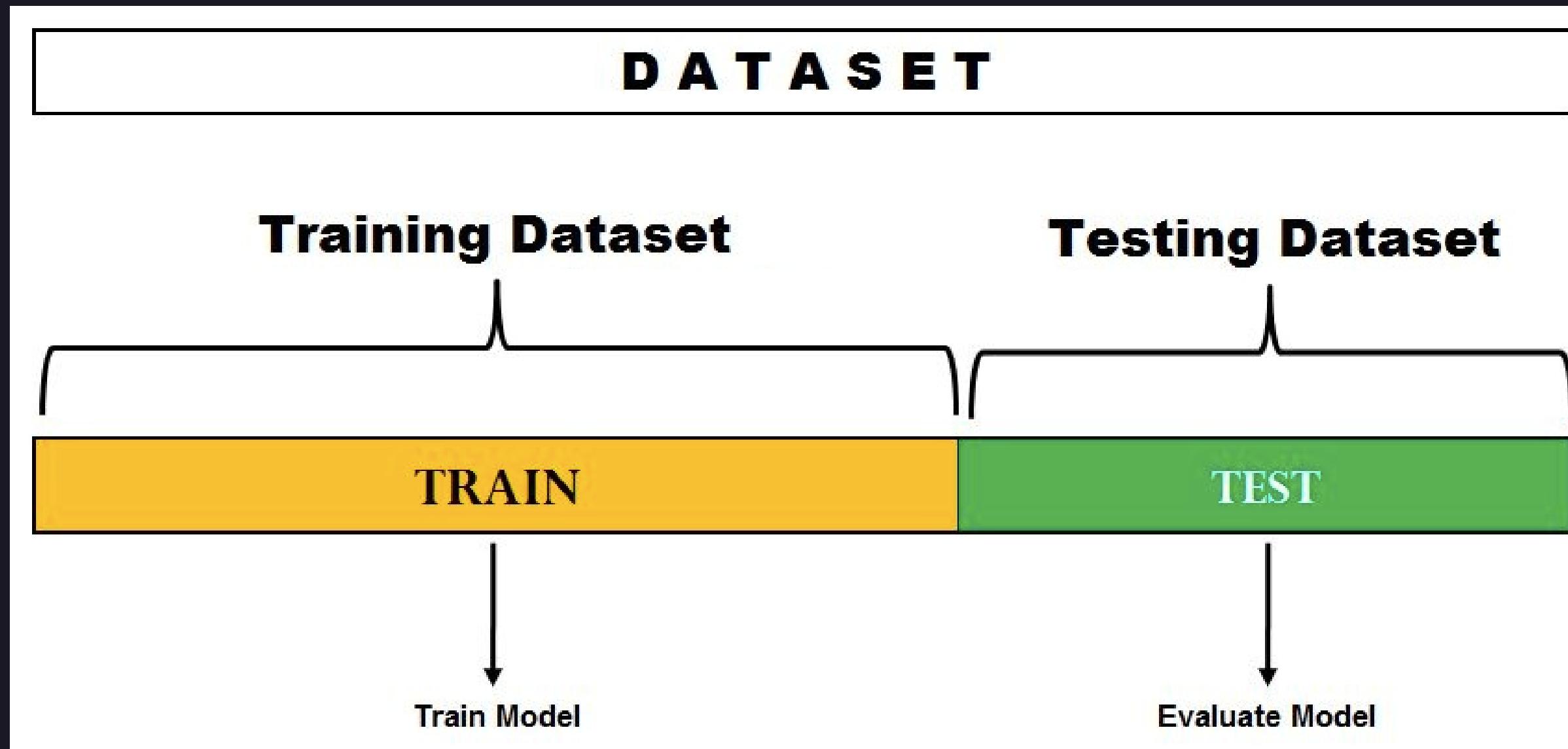


*function : $y = \sin(\pi \cdot x)$

Week 3

Performance Estimation

1.1 การทดสอบความเที่ยงตรง (Precision) ของวิธี Holdout



1.1 การทดสอบความเที่ยงตรง (Precision) ของวิธี Holdout

- HeightWeight20.csv (20 samples) ด้วย seed 1-500

Value from Weka

seed	90	80	50	20	10
1	8.31	7.912	8.48	8.72	18.35
2	8.95	6.99	7.25	8.76	19.25
3	7.25	5.9	6.83	9.23	30.4
4	9.56	8.41	6.7	7.33	21.92
5	5.8	6.74	6.29	14.92	32.27
6	11.49	8.62	6.6	8.4	18.64
7	6.5543	8.73	7.99	9.06	18.6
8	6.62	5.64	7.923	20.69	19.27
9	8.14	6.89	7.4	8.81	18.23
10	8.58	8.72	8.37	14.22	23.98
avg	8.12543	7.4552	7.3833	11.014	22.091
sd	1.587700275	1.116372411	0.7372679364	4.022019891	4.952547728

*ที่ 50% ให้ค่า Min Std

Training set

seed	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%	30%	25%	20%	15%	10%
1	7.847	6.198	5.563	5.554	5.055	6.403	6.91	6.388	5.968	5.998	5.671	9.166	9.937	8.417	7.775	6.572	11.238	11.198
2	8.52	8.607	8.11	8.885	9.733	8.887	9.716	9.554	8.781	8.271	7.81	7.974	8.554	8.537	7.798	7.466	7.243	75.949
3	8.156	9.048	9.664	8.438	7.553	6.912	6.371	6.35	7.164	7.545	8.559	8.054	7.683	8.575	8.255	9.728	9.328	9.135
499	7.847	7.225	6.951	7.596	7.554	7.022	7.481	7.808	7.325	6.877	7.049	6.869	6.81	7.3	6.928	6.713	12.93	10.083
500	11.879	8.46	8.497	7.531	9.542	11.469	10.666	10.416	10.683	10.76	9.924	9.692	9.373	8.43	8.932	8.653	11.517	10.083
Average	6.798	7.173	7.307	7.388	7.484	7.524	7.648	7.717	7.785	7.883	7.954	8.067	8.272	8.486	9.017	9.678	12.632	163.121
Std	2.833	2.013	1.655	1.448	1.306	1.209	1.237	1.214	1.207	1.206	1.187	1.319	1.435	1.544	2.364	3.13	15.464	1392.251

*ที่ 45% ให้ค่า Min Std

1.1 การทดสอบความเที่ยงตรง (Precision) ของวิธี Holdout

- HeightWeight20.csv (20 samples)

Seed 1-100

seed	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%	30%	25%	20%	15%	10%
1	7.847	6.198	5.563	5.554	5.055	6.403	6.91	6.388	5.968	5.998	5.671	9.166	9.937	8.417	7.775	6.572	11.238	11.198
2	8.52	8.607	8.11	8.885	9.733	8.887	9.716	9.554	8.781	8.271	7.81	7.974	8.554	8.537	7.798	7.466	7.243	75.949
3	8.156	9.048	9.664	8.438	7.553	6.912	6.371	6.35	7.164	7.545	8.559	8.054	7.683	8.575	8.255	9.728	9.328	9.135
99	11.879	11.492	9.453	8.884	8.745	8.078	7.695	7.386	7.273	6.976	6.74	7.099	7.853	7.66	7.119	9.352	7.76	30.974
100	4.284	4.337	8.821	9.156	9.593	8.738	8.049	8.695	7.887	7.16	7.179	7.153	8.636	8.142	12.002	12.63	8.112	8.949
Averages	6.968	7.156	7.255	7.401	7.435	7.464	7.64	7.788	7.783	7.782	7.859	7.933	8.119	8.309	8.991	9.637	12.524	36.941
Std	2.841	1.974	1.714	1.58	1.306	1.211	1.357	1.306	1.218	1.201	1.211	1.27	1.265	1.257	2.183	3.231	8.286	67.913

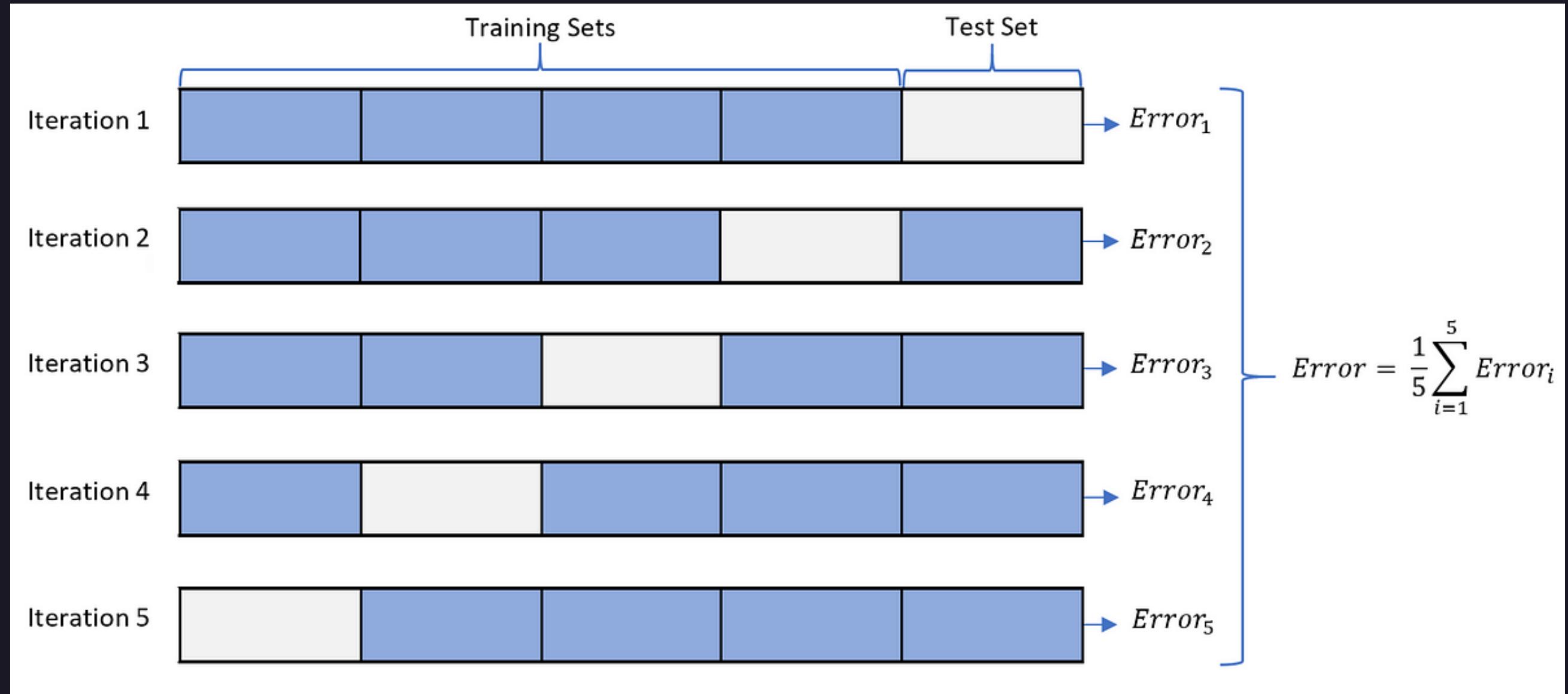
*ที่ 50% ให้ค่า Min Std

Seed 1-500

seed	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%	30%	25%	20%	15%	10%
1	7.847	6.198	5.563	5.554	5.055	6.403	6.91	6.388	5.968	5.998	5.671	9.166	9.937	8.417	7.775	6.572	11.238	11.198
2	8.52	8.607	8.11	8.885	9.733	8.887	9.716	9.554	8.781	8.271	7.81	7.974	8.554	8.537	7.798	7.466	7.243	75.949
3	8.156	9.048	9.664	8.438	7.553	6.912	6.371	6.35	7.164	7.545	8.559	8.054	7.683	8.575	8.255	9.728	9.328	9.135
499	7.847	7.225	6.951	7.596	7.554	7.022	7.481	7.808	7.325	6.877	7.049	6.869	6.81	7.3	6.928	6.713	12.93	10.083
500	11.879	8.46	8.497	7.531	9.542	11.469	10.666	10.416	10.683	10.76	9.924	9.692	9.373	8.43	8.932	8.653	11.517	10.083
Average	6.798	7.173	7.307	7.388	7.484	7.524	7.648	7.717	7.785	7.883	7.954	8.067	8.272	8.486	9.017	9.678	12.632	163.121
Std	2.833	2.013	1.655	1.448	1.306	1.209	1.237	1.214	1.207	1.206	1.187	1.319	1.435	1.544	2.364	3.13	15.464	1392.251

*ที่ 45% ให้ค่า Min Std

1.2 การทดสอบความเที่ยงตรง (Precision) ของวิธี k-fold Cross-Validation



1.2 การทดสอบความเที่ยงตรง (Precision) ของวิธี k-fold Cross-Validation

- HeightWeight20.csv (20 samples)
- Seed 1-500

Value from Weka

seed	10-fold	5-fold	3-fold	2-fold
1	7.78	9.09	9.1	8.1
2	7.55	7.25	7.05	6.86
3	7.46	7.64	8.81	9.02
4	7.3	6.78	7.03	6.76
5	7.42	8.36	7.96	7.15
6	7.71	7.28	6.86	7.06
7	7.3476	7.63	7.7999	7.588
8	7.61	7.168	7.992	7.83
9	7.33	7.42	7.85	8.91
10	7.43	7.17	6.91	7.29
avg	7.49376	7.5788	7.73619	7.6568
sd	0.1554787394	0.6400726209	0.7460192256	0.7646359657

ที่ 10 folds

ให้ค่า Min Std

Seed	15 fold	10 fold	8 fold	5 fold	4 fold	3 fold	2 fold
1	6.691	7.185	7.221	7.522	7.605	7.546	8.351
2	6.347	7.397	6.79	7.914	8.441	7.875	8.387
3	6.743	7.062	7.109	7.065	7.649	6.869	6.782
4	6.493	7.64	7.526	8.247	6.881	7.881	6.787
5	6.925	7.387	7.615	7.269	8.332	8.073	8.873
6	6.577	7.074	6.739	7.287	7.77	7.276	7.311
7	6.71	7.028	6.864	7.277	8.099	8.012	8.223
8	6.904	7.012	6.565	6.742	7.069	6.973	7.48
9	6.501	7.294	7.043	7.27	7.4	7.959	8.044
10	6.489	7.437	7.187	7.851	7.657	8.006	8.119
11	7.438	7.217	7.429	7.547	7.429	7.985	8.841
12	6.658	7.18	6.969	7.936	7.849	8.463	10.142
13	6.841	7.049	6.944	7.225	7.493	7.28	6.83
14	7.308	6.698	7.035	7.042	7.277	6.952	7.777
15	7.17	7.482	7.57	7.207	7.344	7.2	7.427
16	6.227	7.243	6.403	7.798	8.248	8.101	7.722
17	6.979	6.698	6.664	7.242	7.868	7.252	7.164
18	7.127	7.411	7.443	7.591	7.472	7.21	8.026
19	7.231	6.898	6.918	6.855	7.096	6.945	6.74
20	6.987	7.316	7.375	8.711	8.585	9.213	9.602
⋮							
495	7.244	7.412	7.872	7.514	8.328	7.623	8.36
496	7.012	7.217	6.952	7.674	7.579	7.057	8.674
497	7.201	6.939	7.459	6.799	7.195	6.966	6.814
498	7.011	7.559	7.966	7.685	7.98	7.782	8.028
499	6.576	6.97	6.616	7.311	7.485	7.033	7.273
500	7.317	7.247	7.714	7.064	7.346	6.938	7.825
Average	6.907	7.226	7.213	7.432	7.491	7.568	7.8
Standard Deviation	0.391	0.205	0.391	0.372	0.479	0.601	0.923

1.3 การทดสอบความเที่ยงตรง (Precision) และความแม่นยำ (Accuracy) เมื่อจำนวนข้อมูลมากขึ้น

Holdout

HeightWeight20.csv (20 samples)

seed	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%	30%	25%	20%	15%	10%
1	7.847	6.198	5.563	5.554	5.055	6.403	6.91	6.388	5.968	5.998	5.671	9.166	9.937	8.417	7.775	6.572	11.238	11.198
2	8.52	8.607	8.11	8.885	9.733	8.887	9.716	9.554	8.781	8.271	7.81	7.974	8.554	8.537	7.798	7.466	7.243	75.949
3	8.156	9.048	9.664	8.438	7.553	6.912	6.371	6.35	7.164	7.545	8.559	8.054	7.683	8.575	8.255	9.728	9.328	9.135
499	7.847	7.225	6.951	7.596	7.554	7.022	7.481	7.808	7.325	6.877	7.049	6.869	6.81	7.3	6.928	6.713	12.93	10.083
500	11.879	8.46	8.497	7.531	9.542	11.469	10.666	10.416	10.683	10.76	9.924	9.692	9.373	8.43	8.932	8.653	11.517	10.083
Average	6.798	7.173	7.307	7.388	7.484	7.524	7.648	7.717	7.785	7.883	7.954	8.067	8.272	8.486	9.017	9.678	12.632	163.121
Std	2.833	2.013	1.655	1.448	1.306	1.209	1.237	1.214	1.207	1.206	1.187	1.319	1.435	1.544	2.364	3.13	15.464	1392.251

HeightWeight100.csv (100 samples)

seed	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%	30%	25%	20%	15%	10%
1	7.003	7.608	6.743	6.63	6.206	5.903	6.032	5.867	5.931	6.136	5.966	5.766	5.606	5.896	5.662	5.581	5.77	6.806
2	6.409	4.86	5.273	5.708	5.317	5.354	5.517	6.227	6.227	6.029	6.036	6.03	5.919	5.778	5.735	5.673	6.022	6.177
3	4.858	6.481	6.317	6.864	6.59	6.634	6.378	6.101	5.927	5.958	5.744	5.929	5.912	5.955	6.083	6.12	6.043	5.905
499	4.619	6.501	5.627	5.57	5.419	5.986	5.88	5.907	6.056	6.224	6.042	5.824	5.872	5.709	5.72	5.923	6.029	6.149
500	3.97	5.328	6.295	6.639	6.33	5.974	6.632	6.729	6.716	6.517	6.182	6.558	6.349	6.306	6.133	5.879	5.793	6.542
Average	5.483	5.695	5.712	5.775	5.791	5.828	5.857	5.867	5.866	5.897	5.921	5.935	5.948	5.971	6.003	6.073	6.139	6.382
Std	1.665	1.143	0.932	0.81	0.673	0.596	0.552	0.5	0.447	0.413	0.37	0.35	0.331	0.315	0.315	0.346	0.371	0.588

*ค่า Std และ RMSE มีค่าต่ำลงเมื่อจำนวนข้อมูลมีค่ามากขึ้น

1.3 การทดสอบความเที่ยงตรง (Precision) และความแม่นยำ (Accuracy) เมื่อจำนวนข้อมูลมากขึ้น

Holdout

HeightWeight.csv (10000 samples)

seed	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%	30%	25%	20%	15%	10%
1	5.353	5.4	5.438	5.403	5.447	5.461	5.474	5.466	5.491	5.534	5.552	5.526	5.54	5.544	5.528	5.542	5.55	5.553
2	5.57	5.639	5.588	5.581	5.543	5.575	5.547	5.528	5.54	5.524	5.543	5.565	5.577	5.581	5.571	5.576	5.562	5.558
3	5.468	5.539	5.561	5.602	5.609	5.612	5.582	5.613	5.623	5.623	5.596	5.594	5.564	5.558	5.553	5.552	5.543	5.548
499	5.632	5.676	5.552	5.545	5.484	5.486	5.495	5.492	5.513	5.523	5.531	5.524	5.522	5.539	5.569	5.578	5.56	5.57
500	5.418	5.522	5.523	5.5	5.477	5.473	5.461	5.479	5.497	5.53	5.533	5.535	5.545	5.539	5.542	5.556	5.564	5.559
Averages	5.532	5.537	5.539	5.537	5.537	5.537	5.54	5.539	5.542	5.542	5.543	5.544	5.544	5.545	5.546	5.546	5.547	5.548
Std	0.171	0.12	0.09	0.077	0.067	0.06	0.053	0.049	0.046	0.041	0.037	0.033	0.03	0.026	0.023	0.02	0.016	0.014

seed	90	80	50	20	10
1	6.2987	5.9326	5.098	6.076	6.093
2	6.8886	5.9	6.0855	6.405	8.47
3	5.547	6.5237	6.272	6.511	6.157
4	5.995	5.606	6.029	6.04	6.9555
5	5.443	5.1409	6.188	6.537	6.7449
6	5.92	6.23	6.03	5.78	5.94
7	6.14	5.73	5.56	6.71	6.63
8	5.77	6.83	6.81	6.33	6.75
9	5.28	4.68	5.86	6.3	6.77
10	5.72	6.36	5.87	6.06	6.32
avg	5.90023	5.89332	5.98025	6.2749	6.68304
std	0.4679119244	0.6459262975	0.4489843631	0.2828802063	0.7145253159

Value from Weka

1.3 การทดสอบความเที่ยงตรง (Precision) และความแม่นยำ (Accuracy) เมื่อจำนวนข้อมูลมากขึ้น

Cross validation

HeightWeight20.csv (20 samples)

Seed	15 fold	10 fold	8 fold	5 fold	4 fold	3 fold	2 fold
1	6.691	7.185	7.221	7.522	7.605	7.546	8.351
2	6.347	7.397	6.79	7.914	8.441	7.875	8.387
3	6.743	7.062	7.109	7.065	7.649	6.869	6.782
4	6.493	7.64	7.526	8.247	6.881	7.881	6.787
5	6.925	7.387	7.615	7.269	8.332	8.073	8.873
6	6.577	7.074	6.739	7.287	7.77	7.276	7.311
7	6.71	7.028	6.864	7.277	8.099	8.012	8.223
8	6.904	7.012	6.565	6.742	7.069	6.973	7.48
9	6.501	7.294	7.043	7.27	7.4	7.959	8.044
10	6.489	7.437	7.187	7.851	7.657	8.006	8.119
11	7.438	7.217	7.429	7.547	7.429	7.985	8.841
12	6.658	7.18	6.969	7.936	7.849	8.463	10.142
13	6.841	7.049	6.944	7.225	7.493	7.28	6.83
14	7.308	6.698	7.035	7.042	7.277	6.952	7.777
15	7.17	7.482	7.57	7.207	7.344	7.2	7.427
16	6.227	7.243	6.403	7.798	8.248	8.101	7.722
17	6.979	6.698	6.664	7.242	7.868	7.252	7.164
18	7.127	7.411	7.443	7.591	7.472	7.21	8.026
19	7.231	6.898	6.918	6.855	7.096	6.945	6.74
20	6.987	7.316	7.375	8.711	8.585	9.213	9.602

⋮

495	7.244	7.412	7.872	7.514	8.328	7.623	8.36
496	7.012	7.217	6.952	7.674	7.579	7.057	8.674
497	7.201	6.939	7.459	6.799	7.195	6.966	6.814
498	7.011	7.559	7.966	7.685	7.98	7.782	8.028
499	6.576	6.97	6.616	7.311	7.485	7.033	7.273
500	7.317	7.247	7.714	7.064	7.346	6.938	7.825
Average	6.907	7.226	7.213	7.432	7.491	7.568	7.8
Standard Deviation	0.391	0.205	0.391	0.372	0.479	0.601	0.923

1.3 การทดสอบความเที่ยงตรง (Precision) และความแม่นยำ (Accuracy) เมื่อจำนวนข้อมูลมากขึ้น

Cross validation

HeightWeight100.csv (100 samples)

Seed	50 fold	40 fold	25 fold	20 fold	15 fold	10 fold	5 fold	4 fold	3 fold	2 fold
1	5.261	5.13	5.504	5.626	5.554	5.818	5.951	5.798	5.633	5.814
2	5.182	5.328	5.434	5.614	5.713	5.758	5.774	5.926	5.755	5.891
3	5.183	5.038	5.512	5.601	5.577	5.82	5.872	5.987	5.905	5.827
4	5.168	5.561	5.545	5.546	5.805	5.64	5.772	5.885	5.952	6.425
5	5.1	5.21	5.392	5.468	5.629	5.691	5.758	5.778	5.861	5.876
6	5.016	4.75	5.444	5.501	5.394	5.734	5.781	5.783	5.885	5.771
7	5.178	5.105	5.377	5.494	5.401	5.605	5.786	5.767	5.749	5.765
8	5.224	5.173	5.594	5.68	5.544	5.767	5.788	5.827	5.814	5.869
9	5.23	4.929	5.522	5.6	5.388	5.704	5.871	5.889	5.929	6.284
10	5.215	5.101	5.58	5.633	5.788	5.758	5.821	5.853	5.759	5.943
11	5.125	4.885	5.415	5.565	5.302	5.694	5.719	5.774	5.801	5.716
12	5.262	5.404	5.544	5.561	5.856	5.76	5.853	5.754	5.92	5.744

▪
▪
▪

495	5.204	5.321	5.603	5.586	5.737	5.79	5.862	5.817	5.886	5.793
496	5.319	5.183	5.607	5.766	5.883	5.803	5.814	5.849	5.783	5.852
497	5.086	4.934	5.55	5.651	5.609	5.735	5.755	5.742	5.658	5.754
498	5.191	5.097	5.517	5.59	5.686	5.74	5.863	5.927	6.092	5.756
499	5.138	5.012	5.457	5.515	5.518	5.654	5.663	5.808	5.853	5.77
500	5.078	5.002	5.433	5.505	5.66	5.725	5.75	5.734	5.802	5.793
Average	5.179	5.189	5.504	5.571	5.626	5.716	5.796	5.814	5.834	5.883
Standard Deviation	0.064	0.206	0.066	0.07	0.149	0.064	0.071	0.082	0.103	0.172

1.3 การทดสอบความเที่ยงตรง (Precision) และความแม่นยำ (Accuracy) เมื่อจำนวนข้อมูลมากขึ้น

Cross validation

HeightWeight.csv (10000 samples)

seed		50 fold	40 fold	25 fold	20 fold	15 fold	10 fold	5 fold	4 fold	3 fold	2 fold
	1	5.538	5.536	5.541	5.541	5.544	5.543	5.542	5.543	5.543	5.543
	2	5.534	5.538	5.54	5.541	5.543	5.543	5.543	5.543	5.544	5.543
	3	5.537	5.538	5.541	5.54	5.54	5.542	5.543	5.543	5.543	5.543
	4	5.537	5.537	5.541	5.541	5.541	5.542	5.542	5.542	5.543	5.545
	5	5.538	5.54	5.54	5.54	5.542	5.543	5.544	5.544	5.544	5.543
	6	5.536	5.537	5.539	5.54	5.541	5.543	5.543	5.542	5.545	5.544
	7	5.537	5.539	5.539	5.54	5.54	5.543	5.542	5.543	5.543	5.545
	8	5.539	5.539	5.54	5.542	5.542	5.541	5.543	5.543	5.541	5.543
	9	5.537	5.538	5.541	5.541	5.54	5.543	5.546	5.544	5.543	5.547
	10	5.537	5.539	5.541	5.538	5.54	5.543	5.542	5.542	5.543	5.548
	11	5.535	5.537	5.54	5.541	5.541	5.542	5.543	5.544	5.542	5.542

1.4 การทดสอบความแม่นยำ (Accuracy) ของเครื่องมือ วัดประสิทธิภาพของแบบจำลอง

- Seed 1-500
- Sample size = 20

RMSE from Calculation

Reference RMSE = 5.542

เลือก cross validation ที่ k = 3

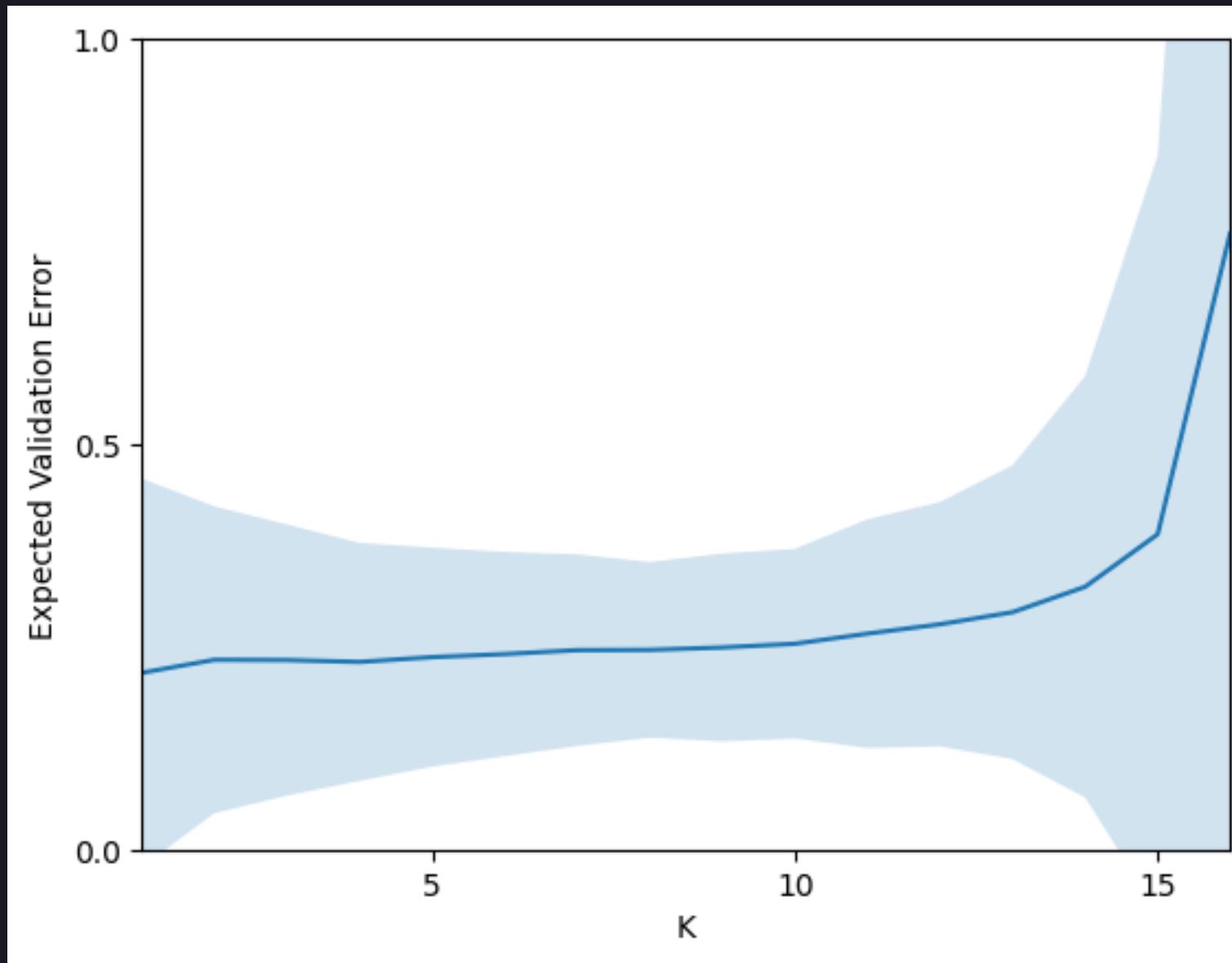
Seed	CV					Training	Holdout				
	20 fold	10 fold	5 fold	3 fold	2 fold		0.9 holdout	0.8 holdout	0.5 holdout	0.2 holdout	0.1 holdout
1	5.474	5.779	6.089	5.818	6.496	6.073	8.598	8.545	7.488	7.293	6.621
2	4.26	4.577	5.036	5.317	6.024	4.682	2.901	4.722	7.797	6.066	8.468
3	5.254	5.677	5.934	5.583	5.929	5.254	6.44	6.837	5.243	7.41	12.93
4	5.248	5.6	6.282	6.249	6.892	6.144	2.252	8.642	5.234	6.57	15.285
5	4.367	5.127	5.27	5.778	5.587	4.93	4.365	8.128	5.513	7.538	7.052
6	4.789	5.446	6.251	7.805	9.77	5.458	8.658	5.512	7.225	5.95	8.398
7	3.745	4.177	4.634	4.162	3.962	3.935	4.96	4.585	3.768	5.785	9.929
8	5.369	5.912	6.138	5.856	6.857	6.134	6.687	9.059	6.038	7.625	27.675
9	3.185	3.609	3.813	3.998	4.101	3.892	5.167	2.912	4.277	5.229	4.124
10	4.245	4.732	5.124	5.596	6.543	4.687	2.469	5.255	4.489	5.919	8.26
11	4.141	4.612	5.102	4.79	5.221	4.364	3.194	5.654	4.912	6.908	11.452
12	5.277	5.6	5.714	5.971	6.349	5.464	3.455	6.691	6.349	7.237	7.488
13	5.109	5.514	6.158	5.876	5.785	5.272	4.431	4.652	8.211	6.745	6.304
14	5.552	5.976	6.194	5.802	7.113	6.453	6.004	5.721	8.296	7.523	20.344
15	5.644	6.295	6.769	7.164	6.435	6.303	11.372	6.769	9.103	6.662	8.925
16	5.354	5.588	6.493	6.691	7.5	5.561	3.081	7.81	6.902	6.986	15.765
17	4.319	4.517	4.838	5.032	4.793	4.447	3.448	5.167	7.522	9.96	5.62
18	3.35	3.734	3.703	3.699	3.765	3.681	2.958	4.579	7.165	4.741	37.291
19	3.636	4.162	4.388	4.473	5.103	4.375	3.447	1.915	3.932	4.913	14.623
20	3.266	3.744	4.215	4.77	5.259	3.796	8.27	3.58	4.309	4.396	4.876
⋮											

495	4.552	5.091	5.267	5.847	5.685	4.959	0.996	6.959	5.63	5.682	11.3
496	5.232	6.317	7.223	6.666	6.996	6.323	4.385	1.802	7.113	6.573	13.569
497	5.497	5.658	6.235	6.943	6.647	5.941	7.107	7.537	6.14	12.958	6.294
498	4.938	5.348	5.294	5.776	5.378	5.608	1.127	8.846	6.74	6.116	8.863
499	5.344	5.719	5.977	6.578	6.65	5.94	5.457	4.482	5.608	7.441	21.416
500	4.669	5.109	4.985	5.166	5.143	4.771	5.061	5.775	6.06	4.631	6.853
Average	4.665	5.187	5.593	5.716	6.023	5.196	5.283	5.488	6.031	7.715	44.408
Standard Deviation	0.806	0.874	0.953	1.051	1.259	0.858	2.624	1.945	1.471	3.541	309.992

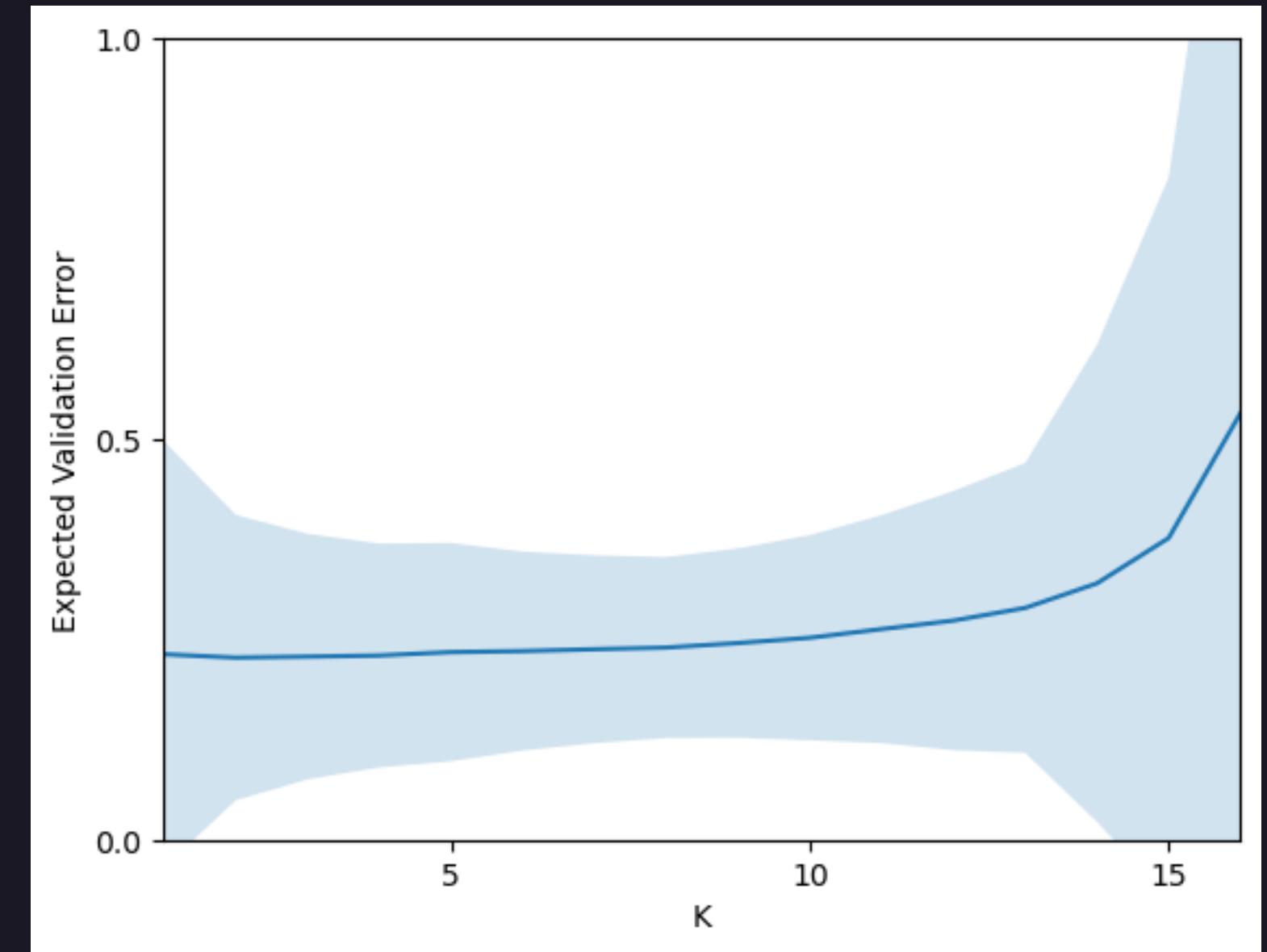
2. Generate Data ប៉ុណ្ណោះ មិន noiseless ទេ noisy ខាងក្រោម សង្ឃឹមថា សរុបនៅក្នុងក្រឡាតាំងនេះ គឺជាបន្ទាត់នៅក្នុងក្រឡាតាំងនេះ

Noiseless

Seed 1 - 500



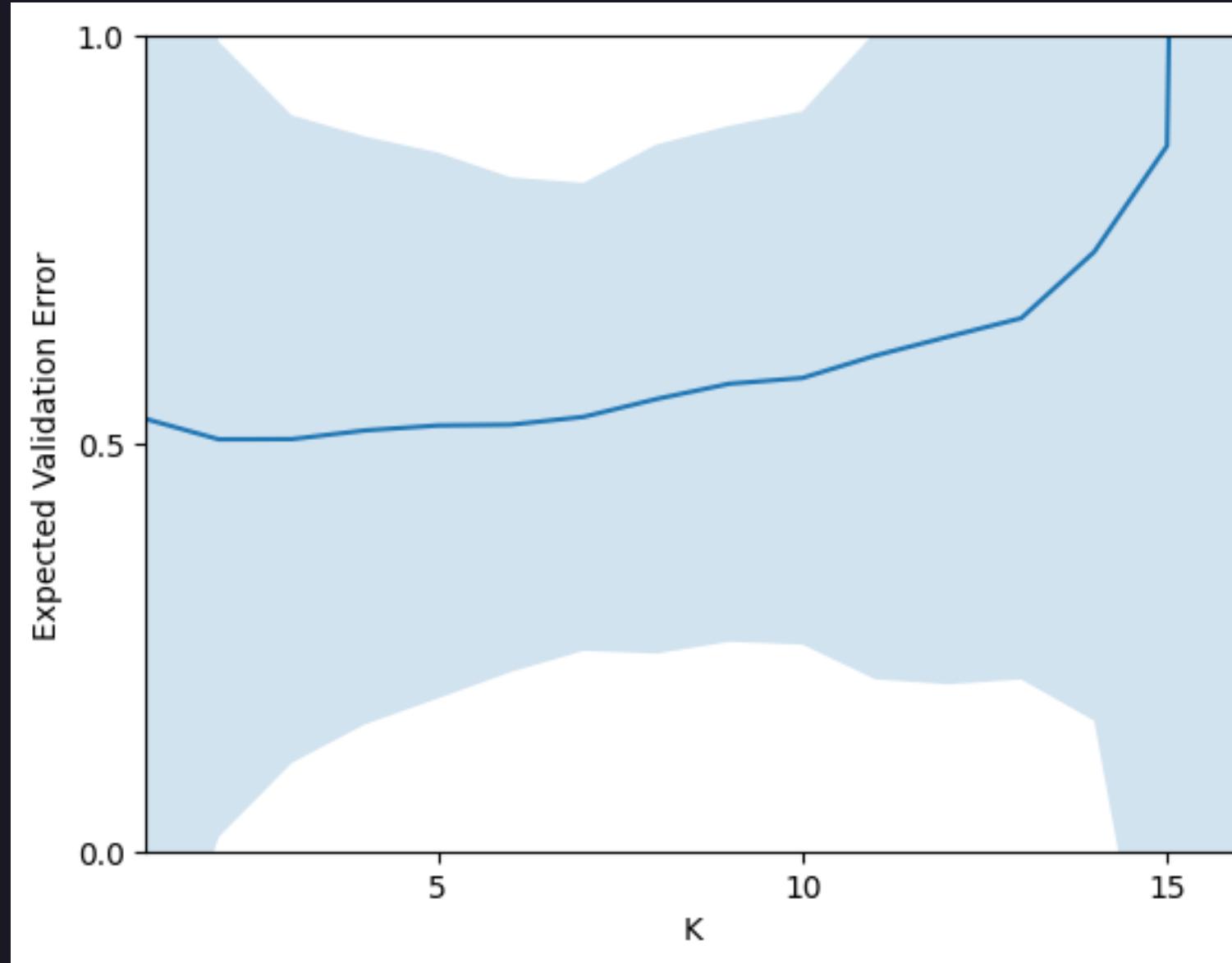
Seed 1 - 2000



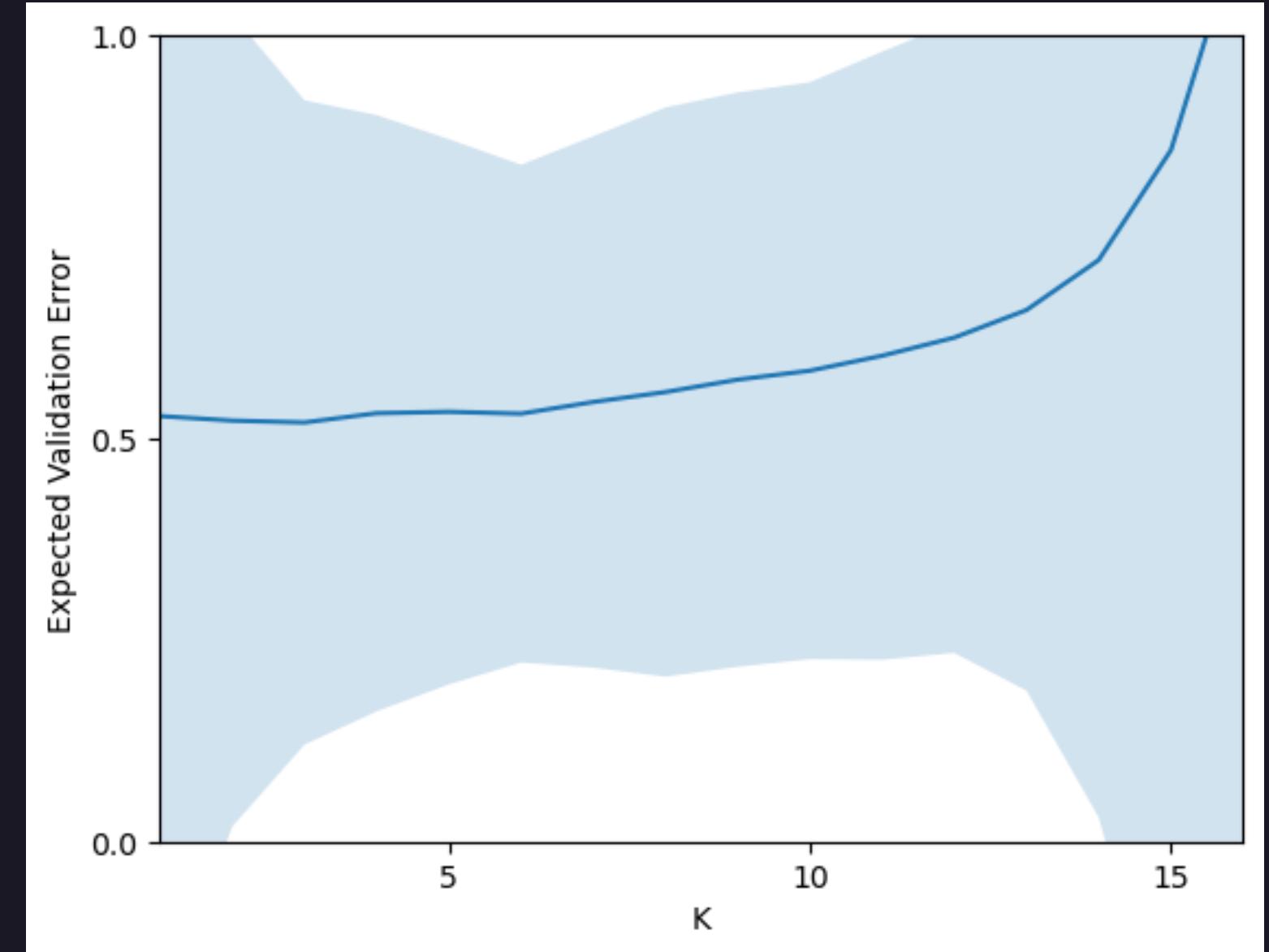
2. Generate Data ប៊ូនមានឈង ឬជារឿយ noiseless ឬជារឿយ noisy ទូទៅអង្គភាព sin ដើម្បីរាយការណ៍
Expected Validation Error តាម lecture លីតី 60

Noisy

Seed 1 - 500



Seed 1 - 2000



Week 4 Model Selection

1. ทดลองช้าและการปรับพารามิเตอร์ให้มากกว่าใน lecture หรือลอง generate date ที่แตกต่างจากที่เรียน

Table of relation Degree and MSE

Noiseless

Sample size	Degree	Training error	CV Error
10	1	0.476717121	0.496119605
10	3	0.063022001	0.15073283
10	8	8.78E-08	3.51E-06
20	1	0.420128376	0.218308392
20	3	0.063942693	0.007727745
20	8	7.63E-05	2.27E-06
40	1	0.458104891	0.230321692
40	3	0.06206638	0.005473197
40	8	0.000139322	1.46E-07
80	1	0.444214019	0.208756516
80	3	0.069380117	0.005308549
80	8	0.000170674	1.03E-07

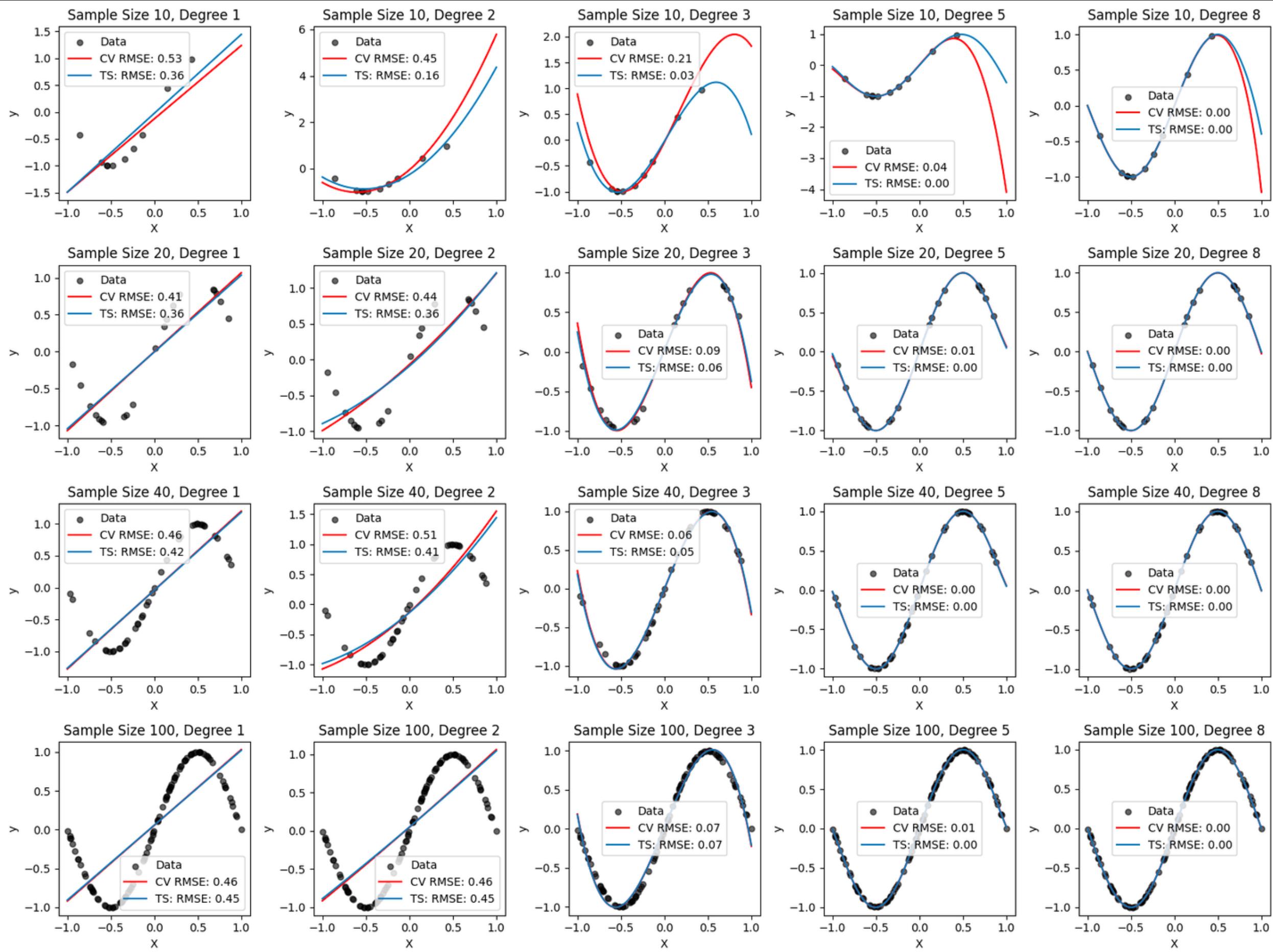
Noisy

Sample size	Degree	Training error	CV Error
10	1	0.456059908	0.451844318
10	3	0.102773354	0.023446589
10	8	0.016402279	123420.9972
20	1	0.488479477	0.28702981
20	3	0.100741488	0.017285796
20	8	0.070199756	0.014352879
40	1	0.421140472	0.197139558
40	3	0.091746172	0.012398525
40	8	0.061897133	0.325839514
80	1	0.454278986	0.217350881
80	3	0.123324762	0.017338276
80	8	0.104035515	0.013128631

Noiseless

Function: $y = \sin(\pi X)$

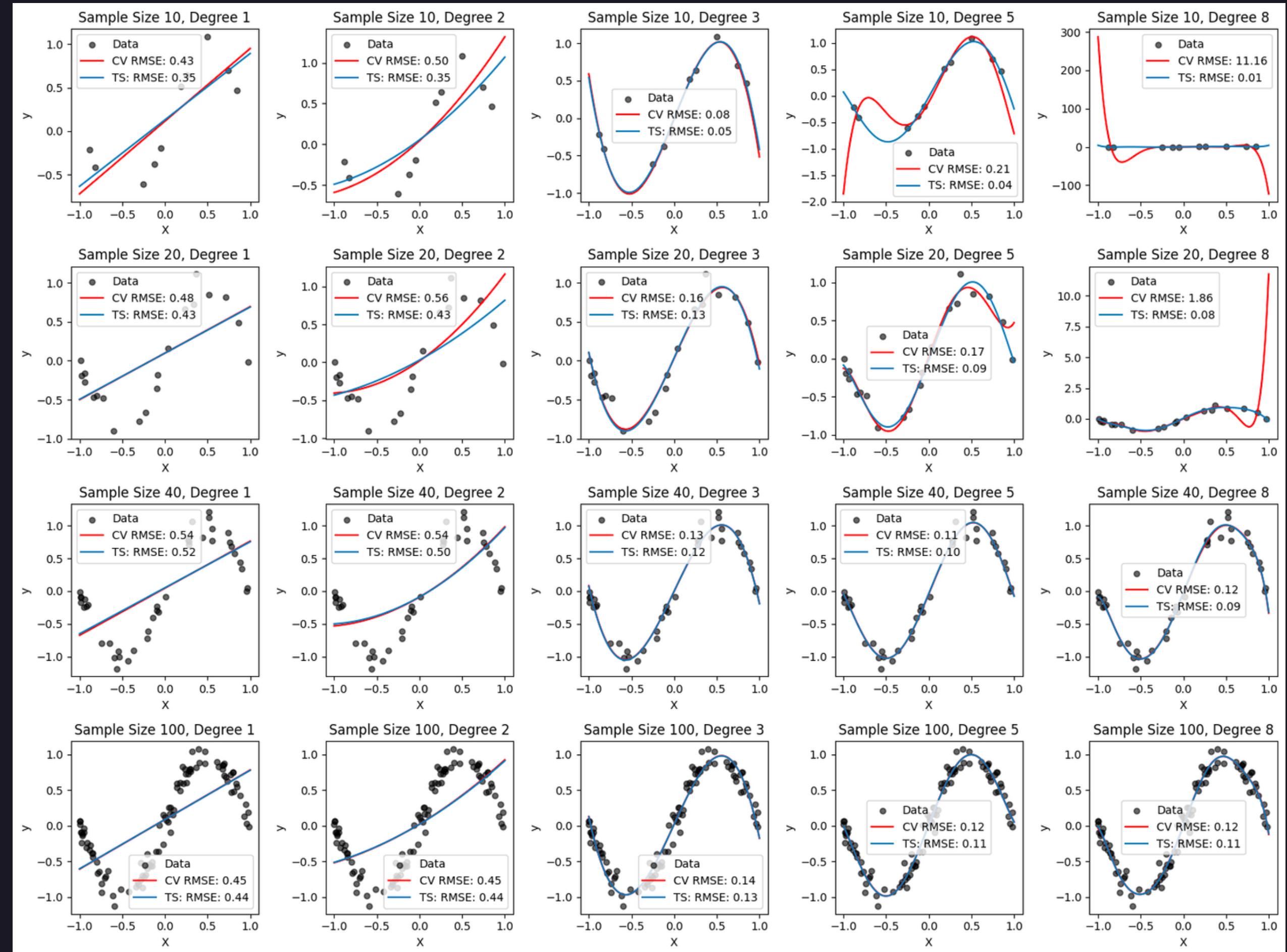
Data Set: Generated data



Noisy

Function: $y = \sin(\pi X)$

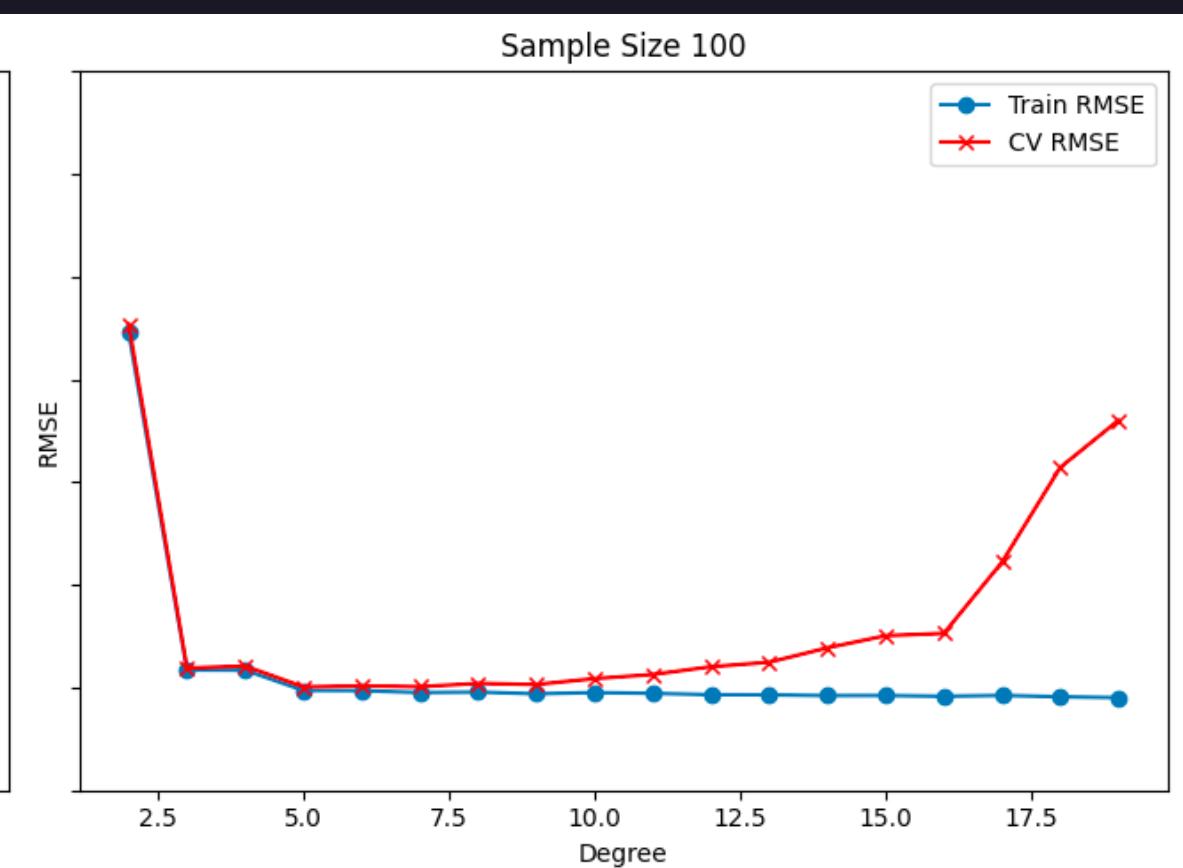
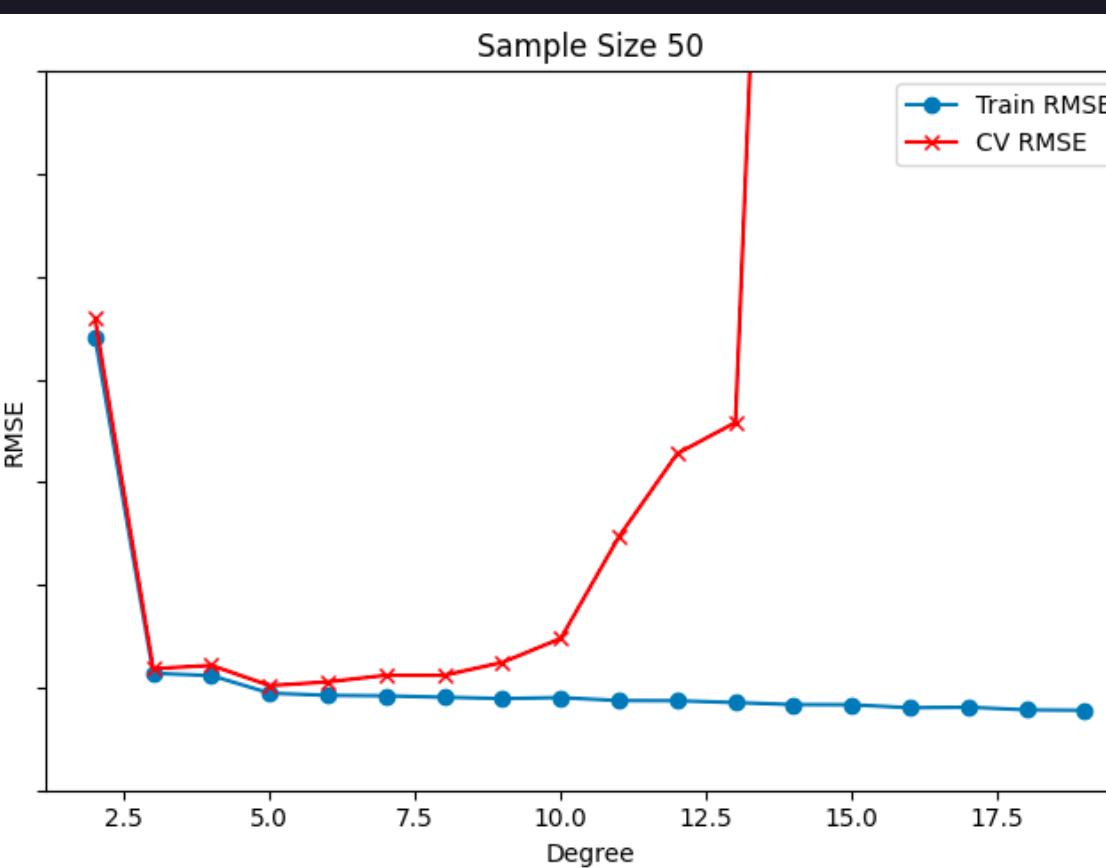
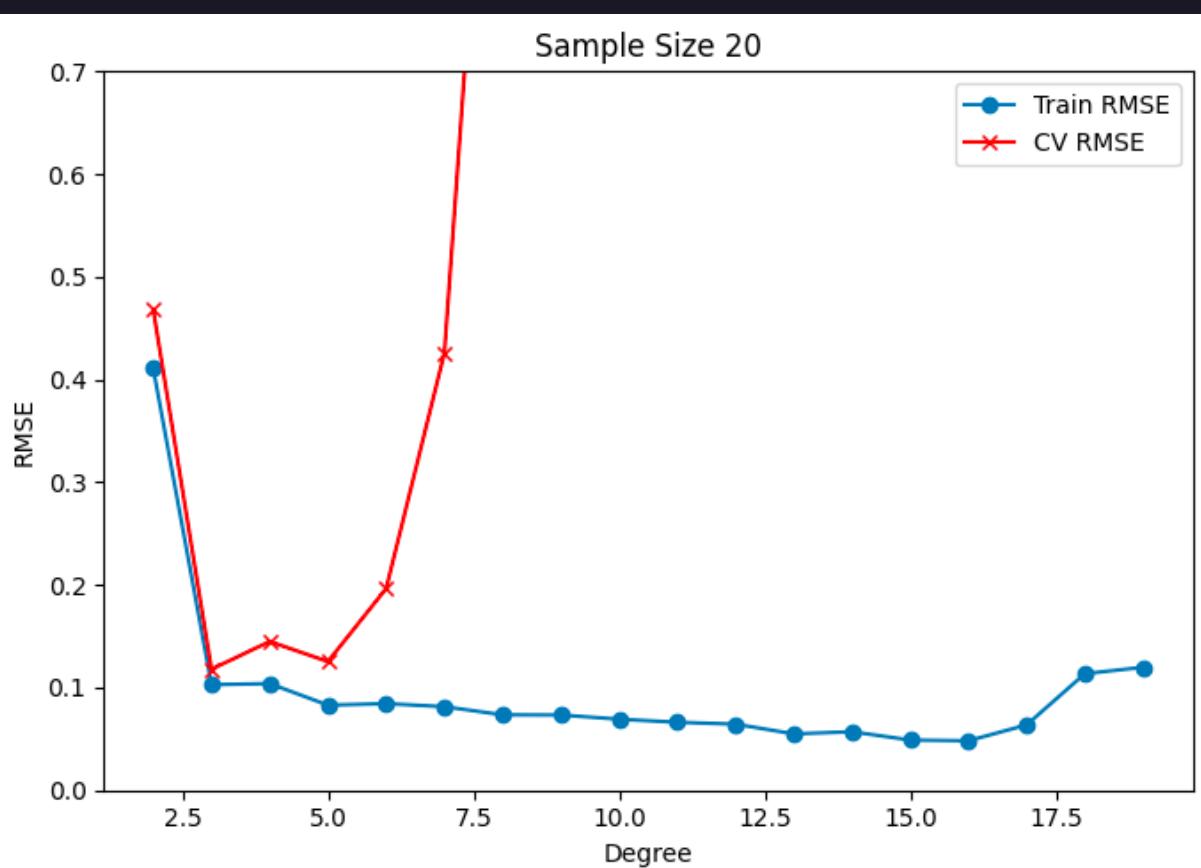
Data Set: Generated data



Compare CV vs Training

Bias vs. Variance

Sample Size: 20



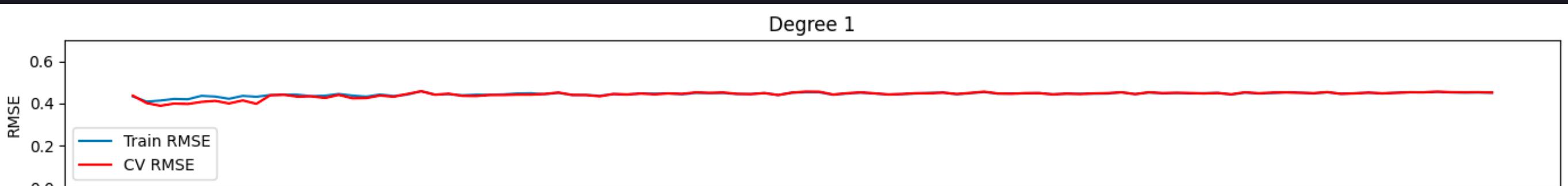
Function: $y = \text{Sin}(\pi * x)$

*Small Sample Size จะมีช่องว่างระหว่าง Train และ CV มาก

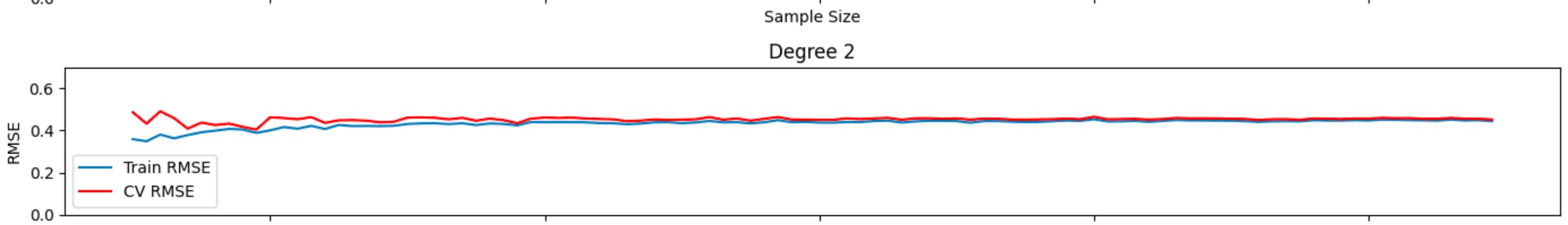
Compare CV vs Training

learning curve

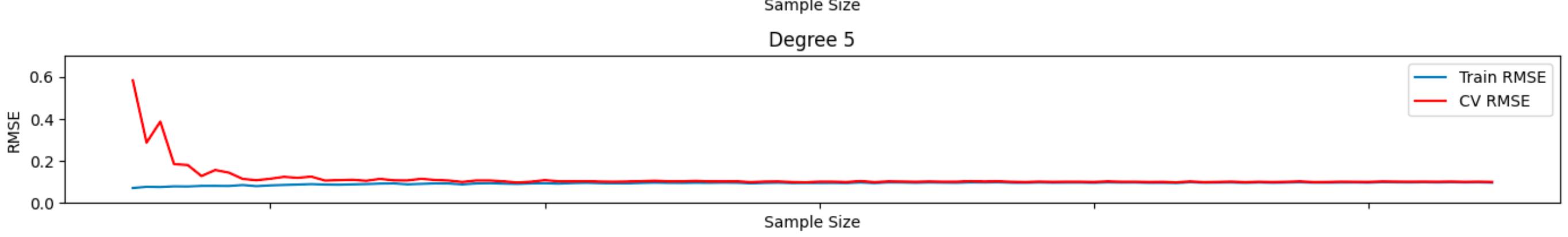
Degree 1



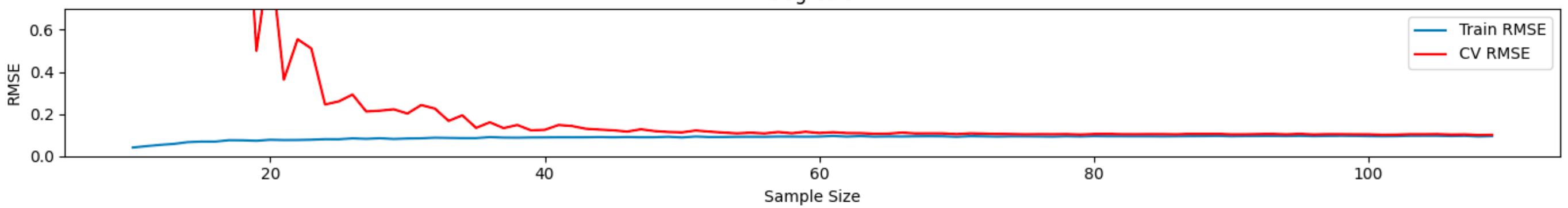
Degree 2



Degree 5



Degree 8



2. เขียนโปรแกรมสำหรับการทำ **Nested Cross-Validation** และออกแบบการทดลองเพื่อแสดงให้เห็นถึงความจำเป็นของ การทำสองลูปแทนที่จะทำเพียงแค่ลูปเดียว

Nested Loop

sample size	Degree Noisy	Rmse Noisy 2Loop	Degree Noiseless	Rmse Noiseless 2Loop
0	30	3	0.378712	11
1	40	7	0.334815	13
2	50	1	0.446815	13
3	60	3	0.290470	13
4	70	3	0.288234	13
5	80	3	0.276193	13
6	90	3	0.341611	13
7	100	3	0.342355	13
8	110	3	0.326383	13
9	120	3	0.377842	13
10	130	3	0.338439	13
11	140	5	0.379046	13
12	150	5	0.417528	13
13	160	3	0.416439	13
14	170	3	0.418448	13
15	180	5	0.456068	13
16	190	3	0.456433	14
17	200	3	0.428902	13

Function: $y = \sin(\pi X)$

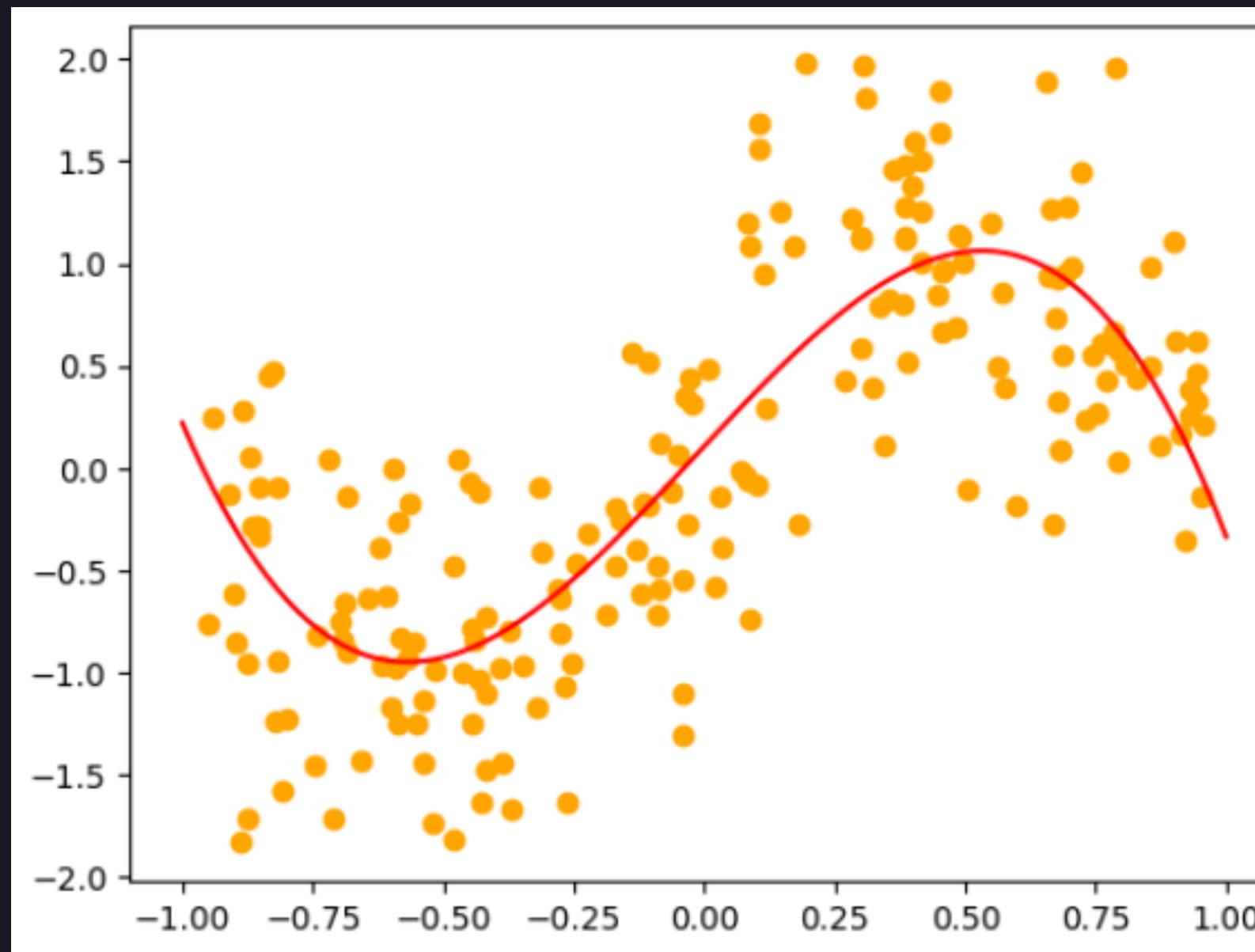
Data Set: Generated data

Sample size = 30 - 200

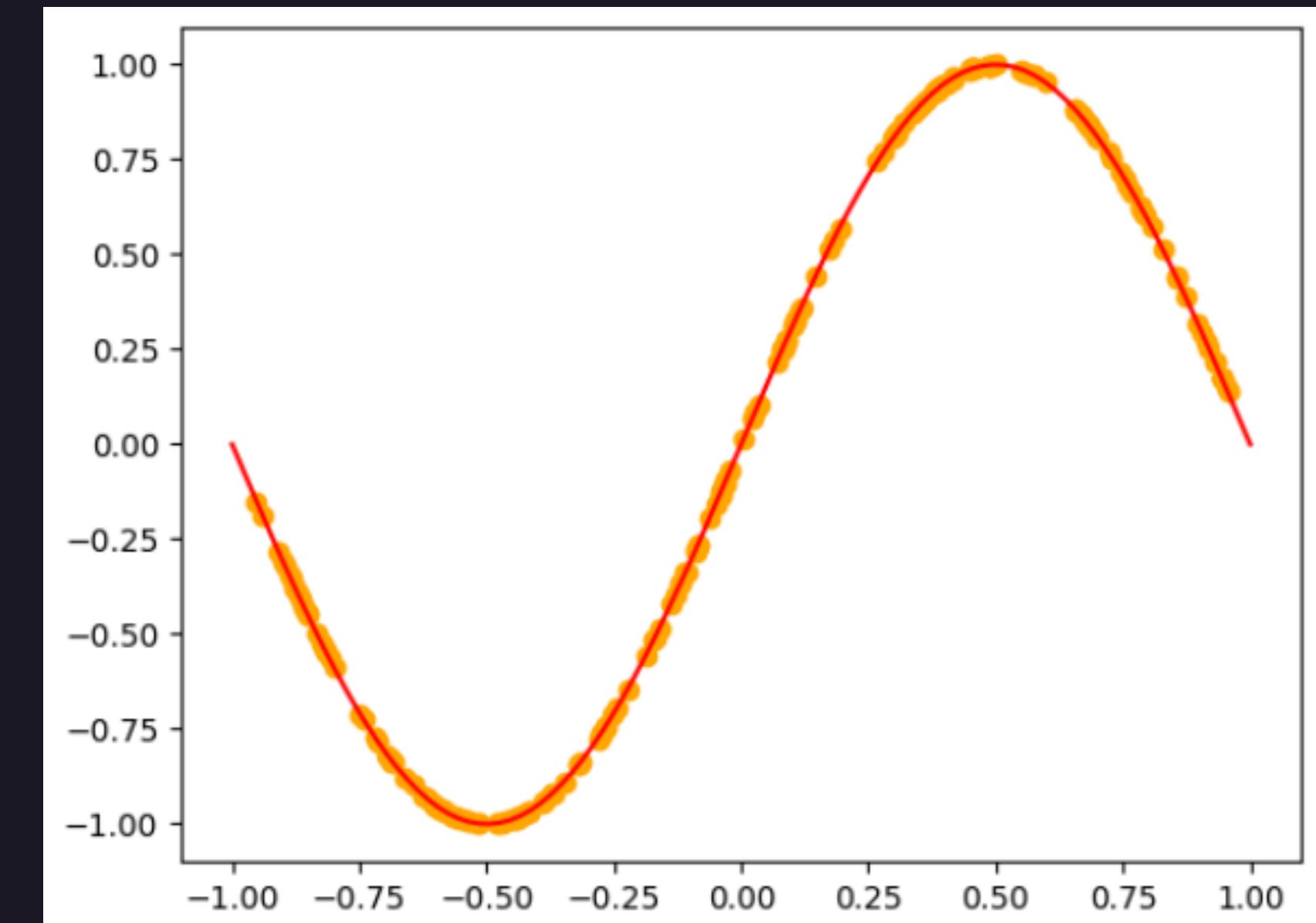
Nested Loop

Function: $y = \sin(\pi x)$
Data Set: Generated data

Noisy



Noiseless



Nested Loop vs 1Loop

Function: $y = \sin(\pi X)$

Data Set: Generated data

Degree = 13 for single loop

sample size	Rmse Noisy 2Loop	Rmse Noisy 1Loop	Rmse Noiseless 2Loop	Rmse Noiseless 1Loop
30	0.378711889	74.09394155	8.60E-09	2.05E-07
40	0.334814617	36.88945614	2.98E-09	3.43E-08
50	0.446815156	4.574649982	2.00E-09	4.89E-09
60	0.290469778	14.64639807	8.65E-10	3.73E-08
70	0.288233889	0.504564136	8.83E-10	1.73E-09
80	0.276192702	0.498427852	1.20E-09	2.10E-09
90	0.341611102	0.645978835	7.16E-10	2.99E-09
100	0.342355158	0.539790883	8.28E-10	2.29E-09
110	0.326383471	1.434346409	9.86E-10	2.61E-09
120	0.377841765	0.521579682	9.71E-10	1.42E-09
130	0.338438982	0.560455427	7.57E-10	1.25E-09
140	0.379045722	0.692748037	1.05E-09	2.61E-09
150	0.417528125	0.703639752	6.95E-10	2.22E-09
160	0.416439375	0.755787494	9.13E-10	2.32E-09
170	0.418447764	0.561466224	6.88E-10	1.68E-09
180	0.456068331	0.560138114	7.32E-10	1.72E-09

Nested Loop

sample size	Degree Noisy	Rmse Noisy 2Loop	Degree Noiseless	Rmse Noiseless 2Loop
0	30	1	0.378597	2
1	40	1	0.333185	2
2	50	1	0.417410	2
3	60	1	0.299742	2
4	70	1	0.288235	2
5	80	3	0.268040	2
6	90	1	0.337797	2
7	100	1	0.340204	2
8	110	3	0.322237	2
9	120	1	0.387811	2
10	130	2	0.338744	2
11	140	2	0.371208	2
12	150	2	0.410030	2
13	160	2	0.409766	2
14	170	2	0.389583	2
15	180	2	0.450165	2
16	190	2	0.452226	2
17	200	2	0.434684	2

Function: $y = X^2$

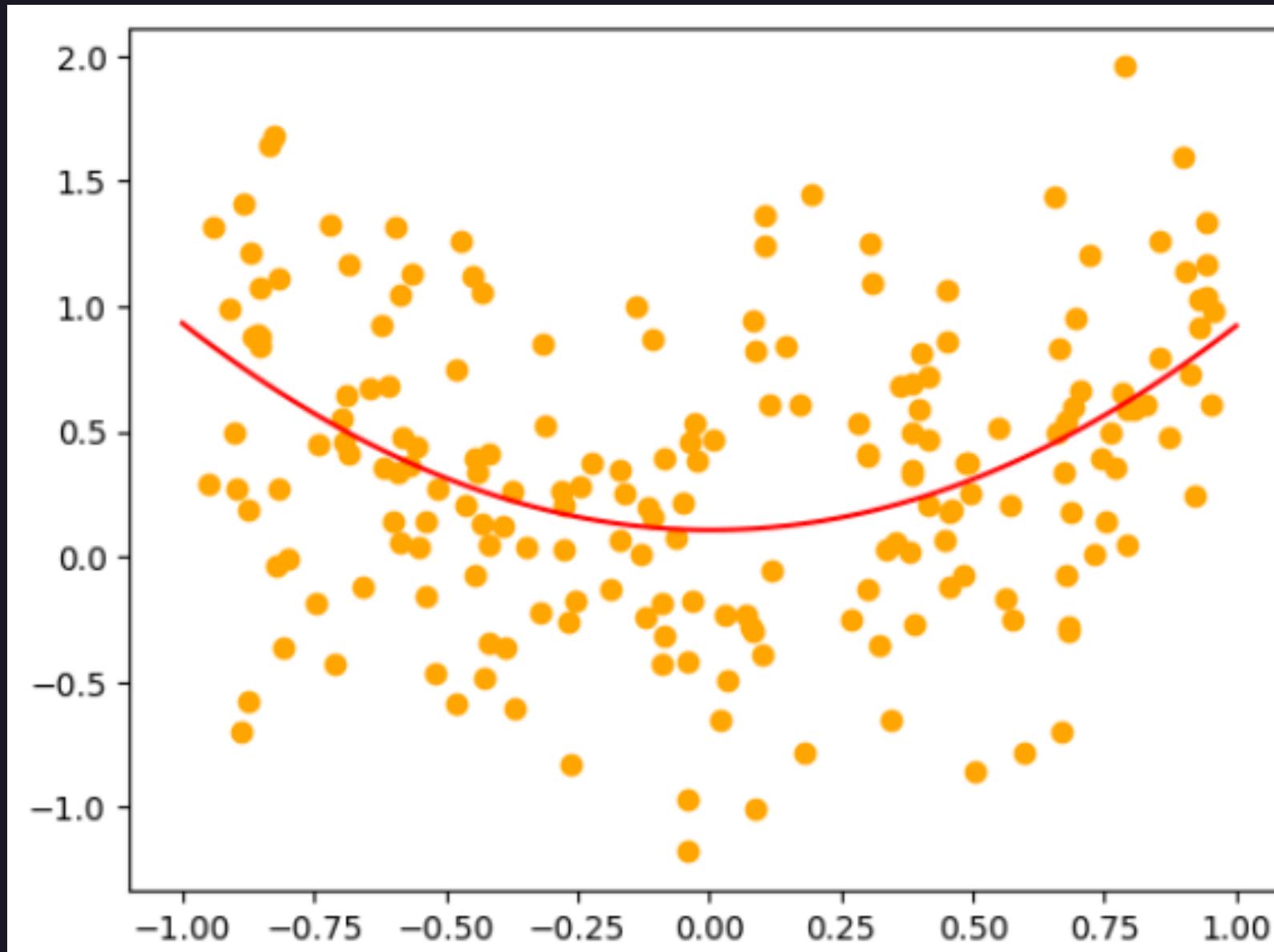
Data Set: Generated data

Sample size = 30 - 200

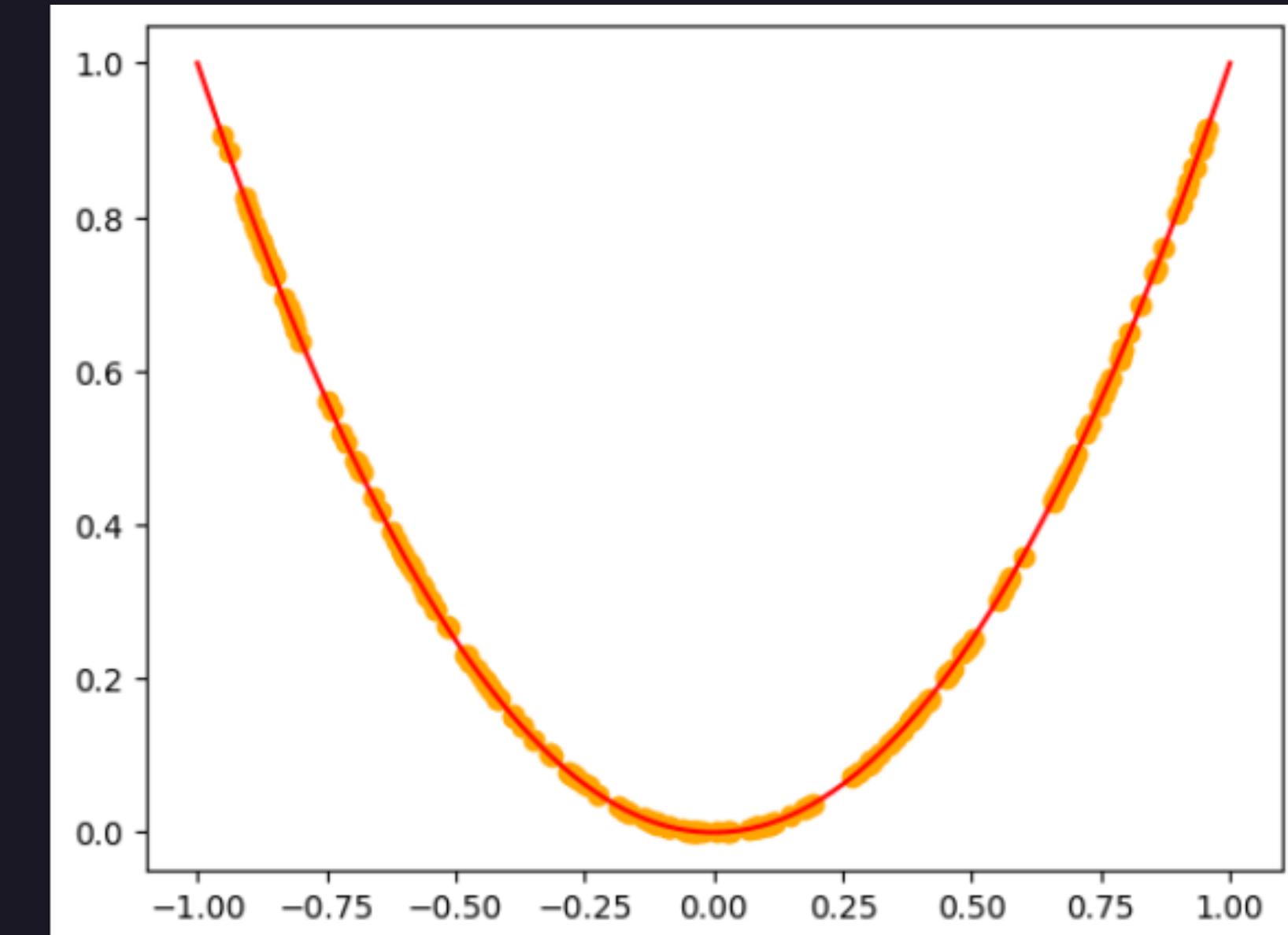
Nested Loop

Function: $y = X^2$
Data Set: Generated data

Noisy



Noiseless



Nested Loop vs 1Loop

Function: $y = x^2$

Data Set: Generated data

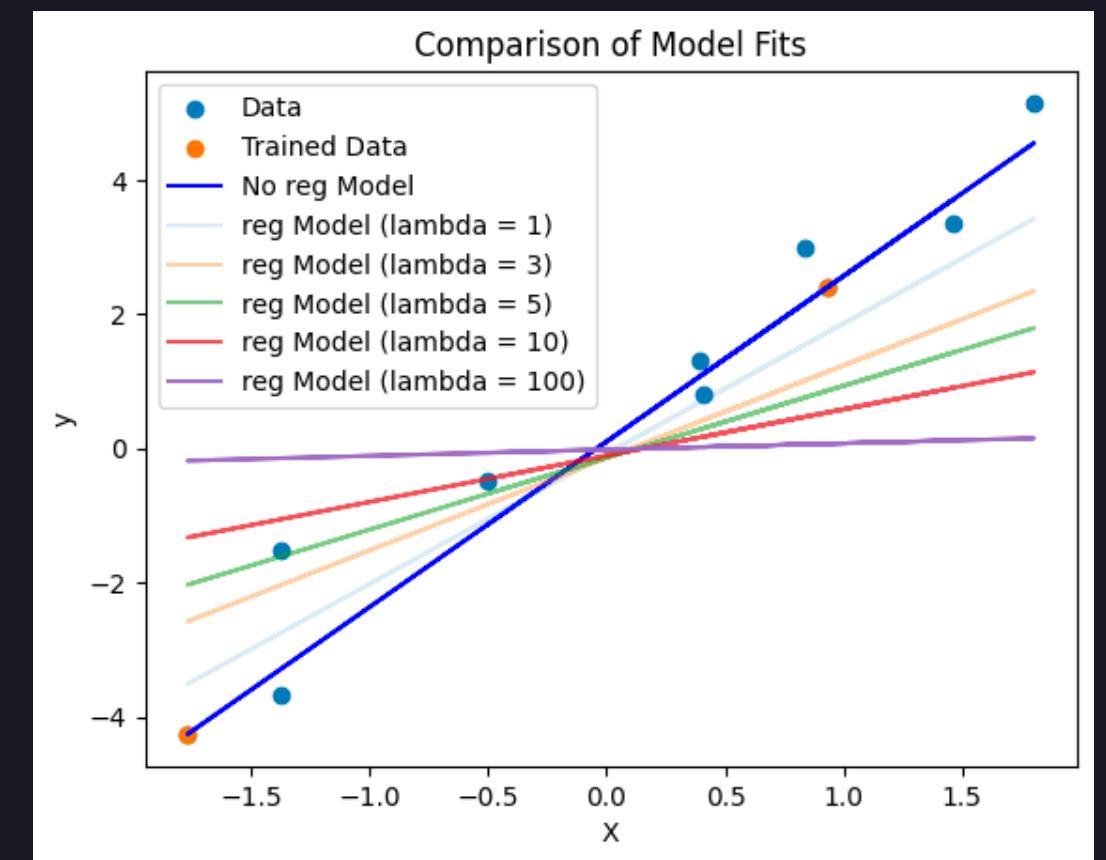
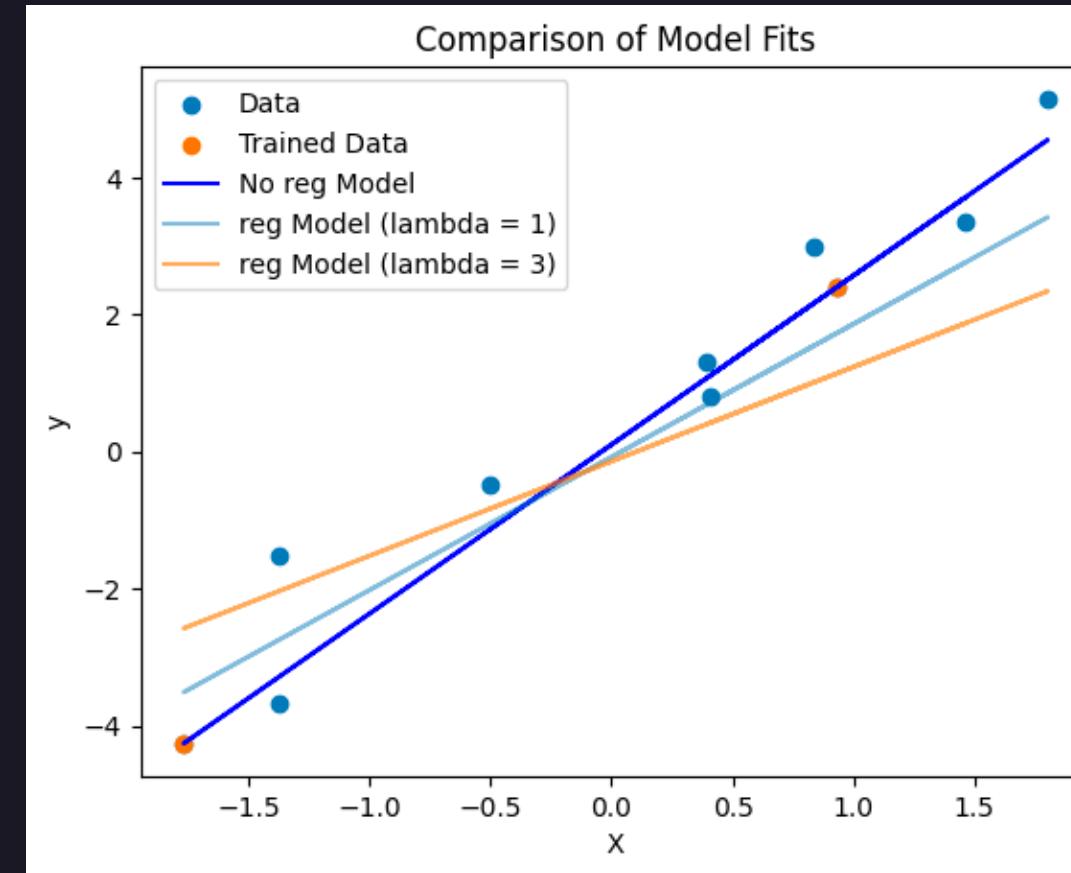
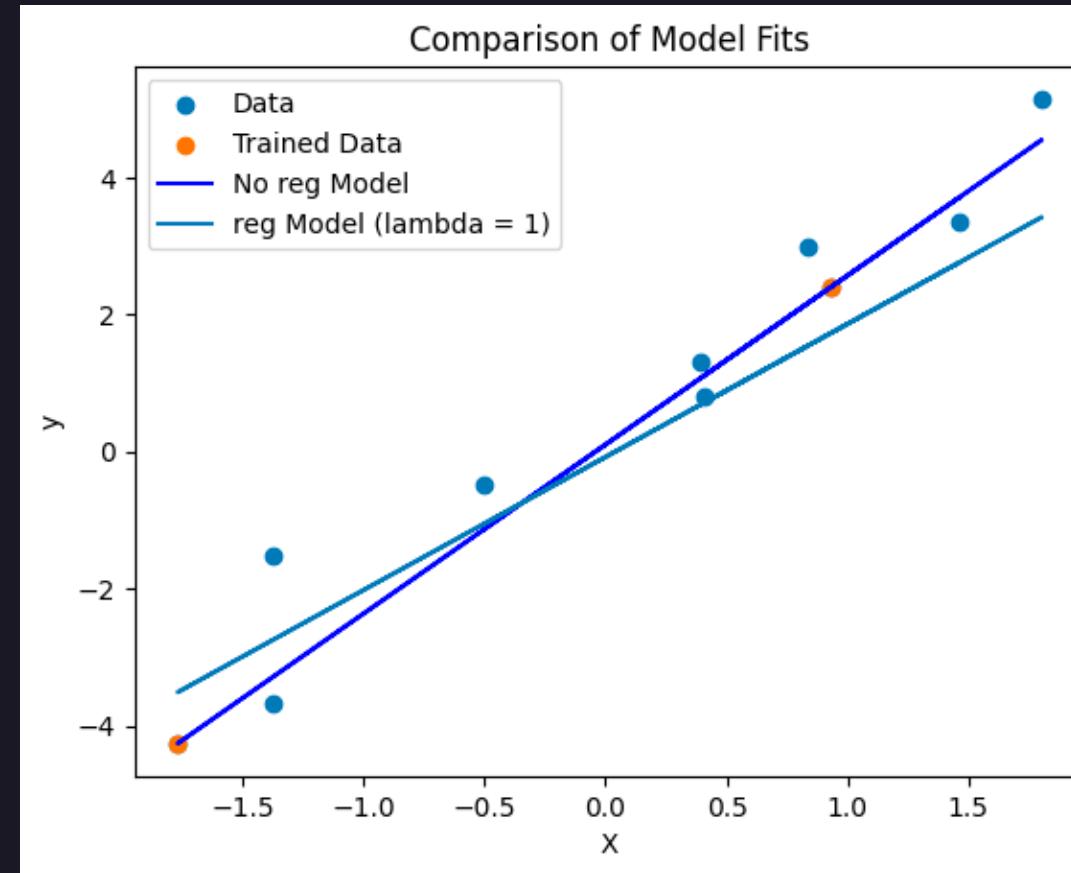
Degree = 2 for single loop

sample size	Rmse Noisy 2Loop	Rmse Noisy 1Loop	Rmse Noiseless 2Loop	Rmse Noiseless 1Loop
30	0.378711889	74.09394125	8.60E-09	3.75E-07
40	0.334814617	36.88945612	2.98E-09	1.82E-08
50	0.446815156	4.574649975	2.00E-09	6.97E-09
60	0.290469778	14.64639804	8.65E-10	4.48E-09
70	0.288233889	0.504564136	8.83E-10	1.72E-09
80	0.276192702	0.498427853	1.20E-09	1.64E-09
90	0.341611102	0.645978835	7.16E-10	1.51E-09
100	0.342355158	0.539790882	8.28E-10	1.42E-09
110	0.326383471	1.434346408	9.86E-10	1.46E-09
120	0.377841765	0.521579682	9.71E-10	7.94E-10
130	0.338438982	0.560455427	7.57E-10	6.83E-10
140	0.379045722	0.692748037	1.05E-09	7.77E-10
150	0.417528125	0.703639753	6.95E-10	7.61E-10
160	0.416439375	0.755787492	9.13E-10	7.89E-10
170	0.418447764	0.561466225	6.88E-10	9.84E-10
180	0.456068331	0.560138114	7.32E-10	7.88E-10
190	0.4564325	0.578358362	8.32E-10	1.08E-09
200	0.428902275	0.615946493	8.94E-10	9.19E-10

Week 5

Linear Model Selection and Regularization

1. เขียนโปรแกรมแสดงความสัมพันธ์ระหว่าง
model และ cost function ของ ridge regression

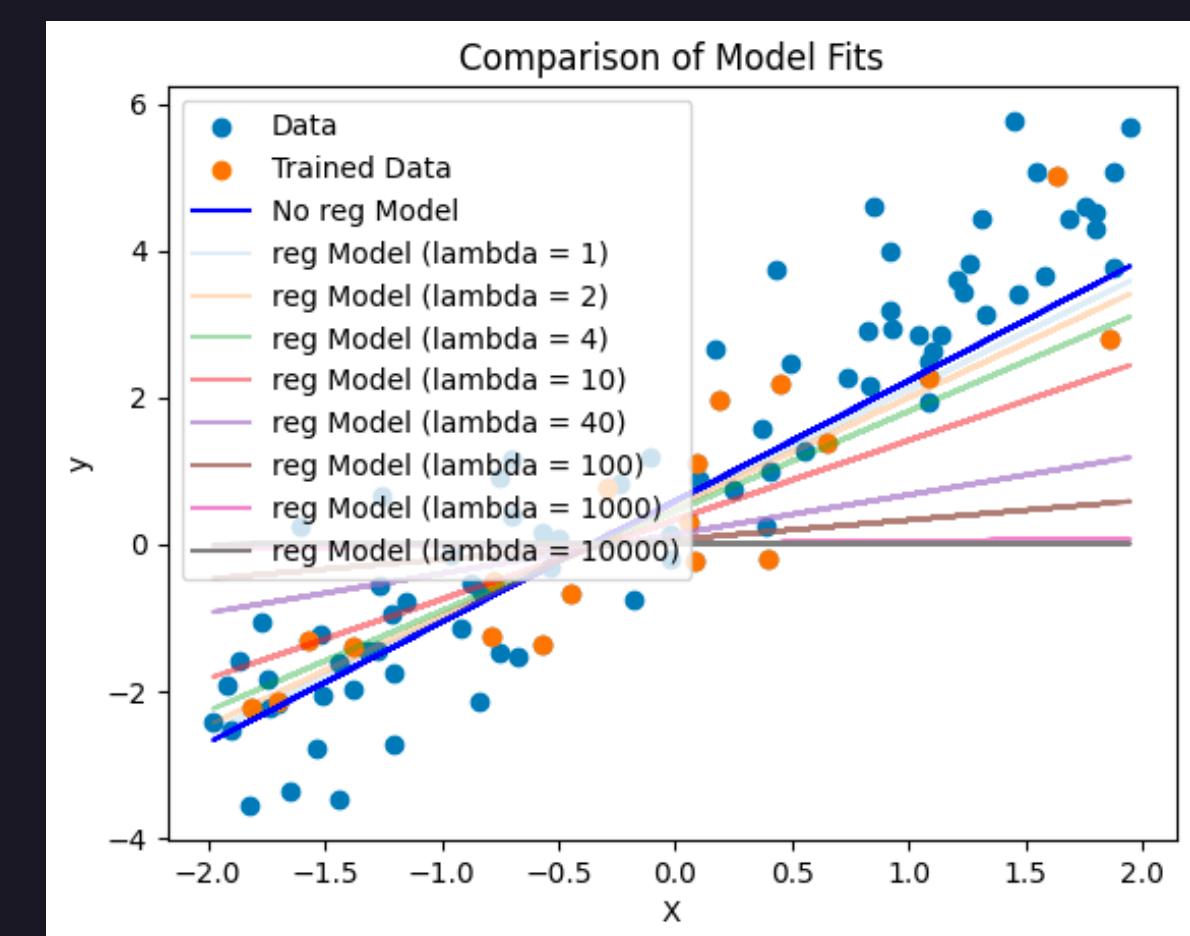
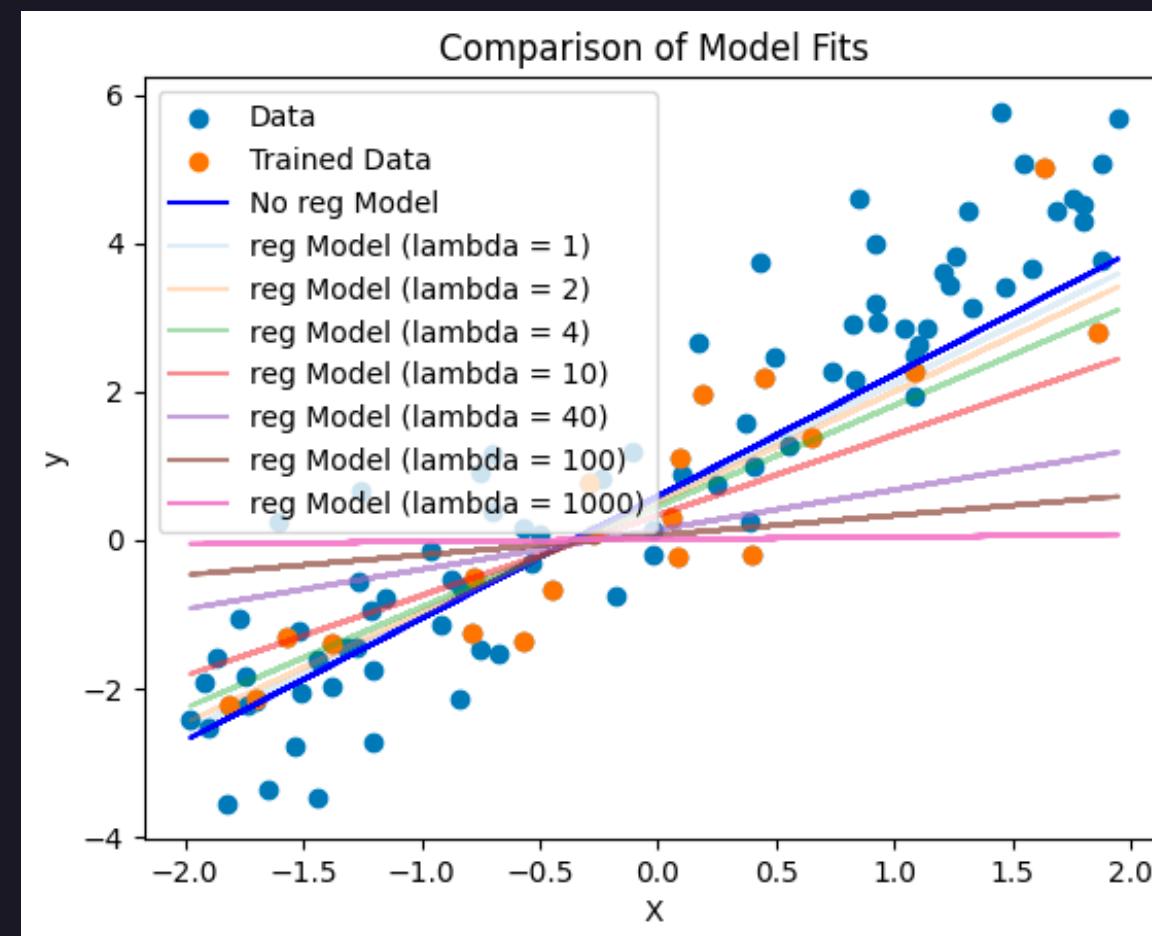
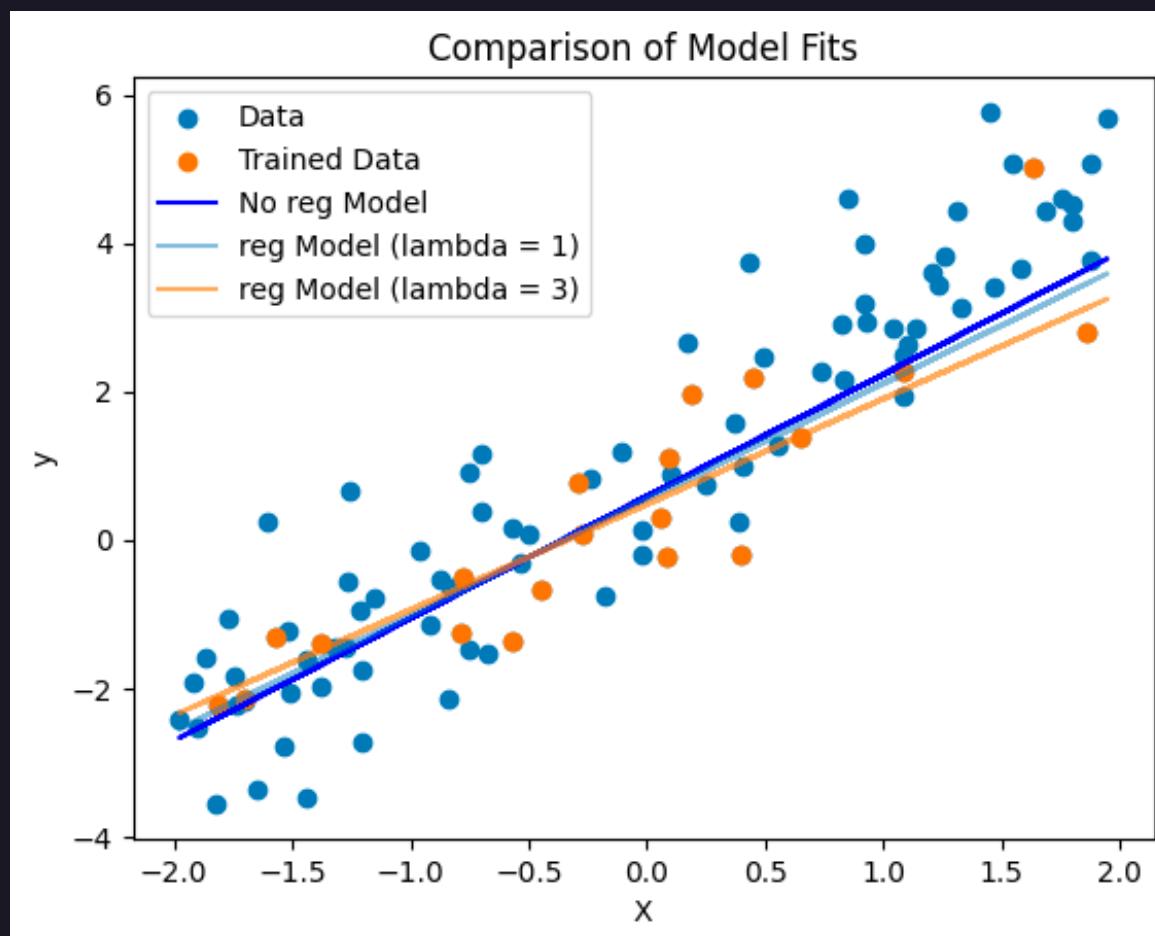


Data:10 ,Trained Data :2

*Slope มีค่าเข้าใกล้ 0 เรื่อยๆ เมื่อ λ ทีค่าเพิ่มมากขึ้น

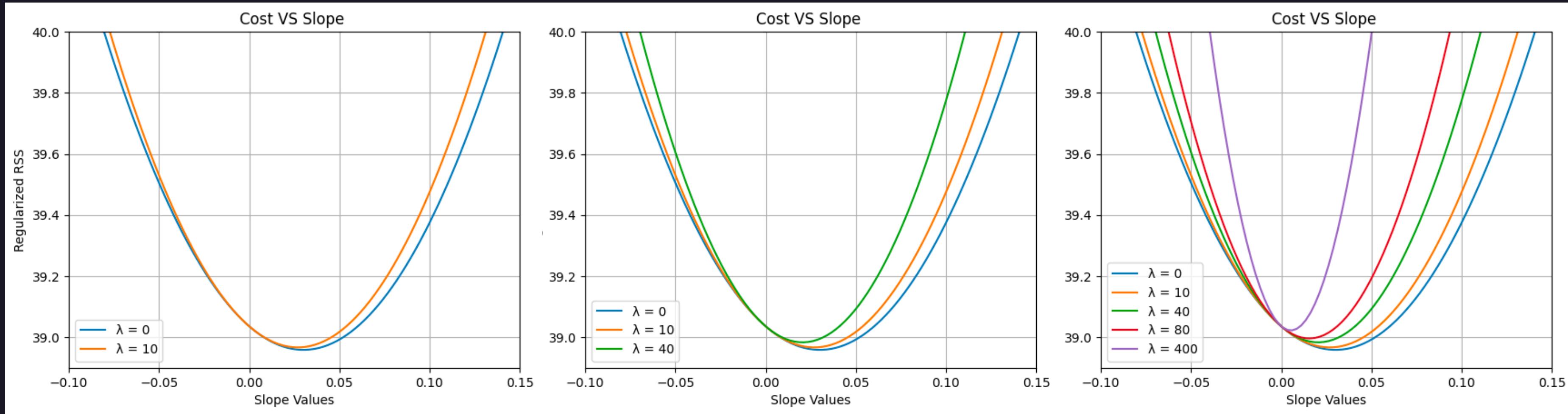
การทดลองเพิ่มจำนวนข้อมูลเป็น 100 ตัว

Data: 100, Trained Data: 20



*Slope มีค่าเข้าใกล้ 0 เรื่อยๆ เมื่อ λ ที่ค่าเพิ่มมากขึ้น

กราฟความสัมพันธ์ระหว่าง RSS และ Slope Value เมื่อเพิ่ม λ ให้มากขึ้น

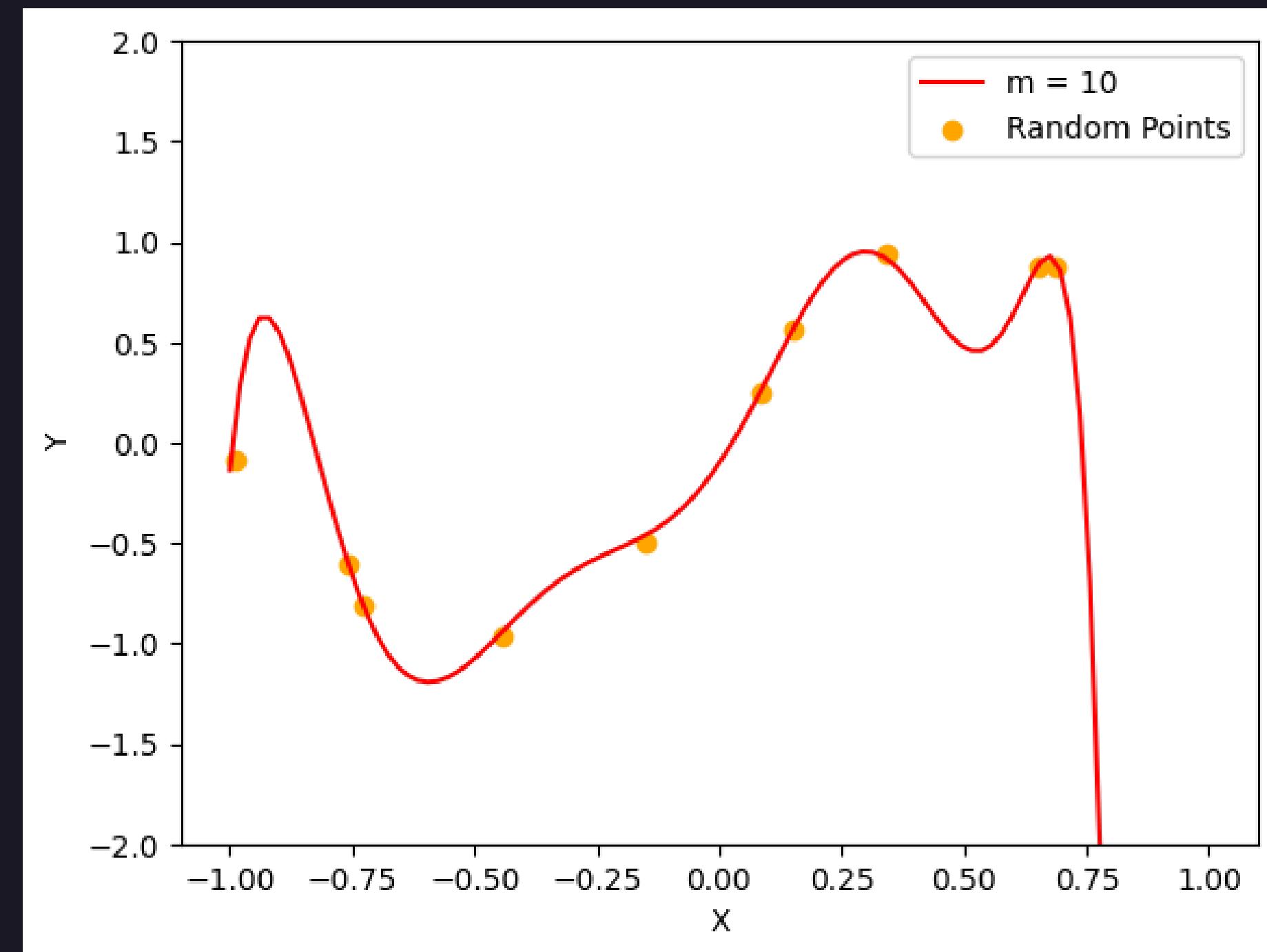
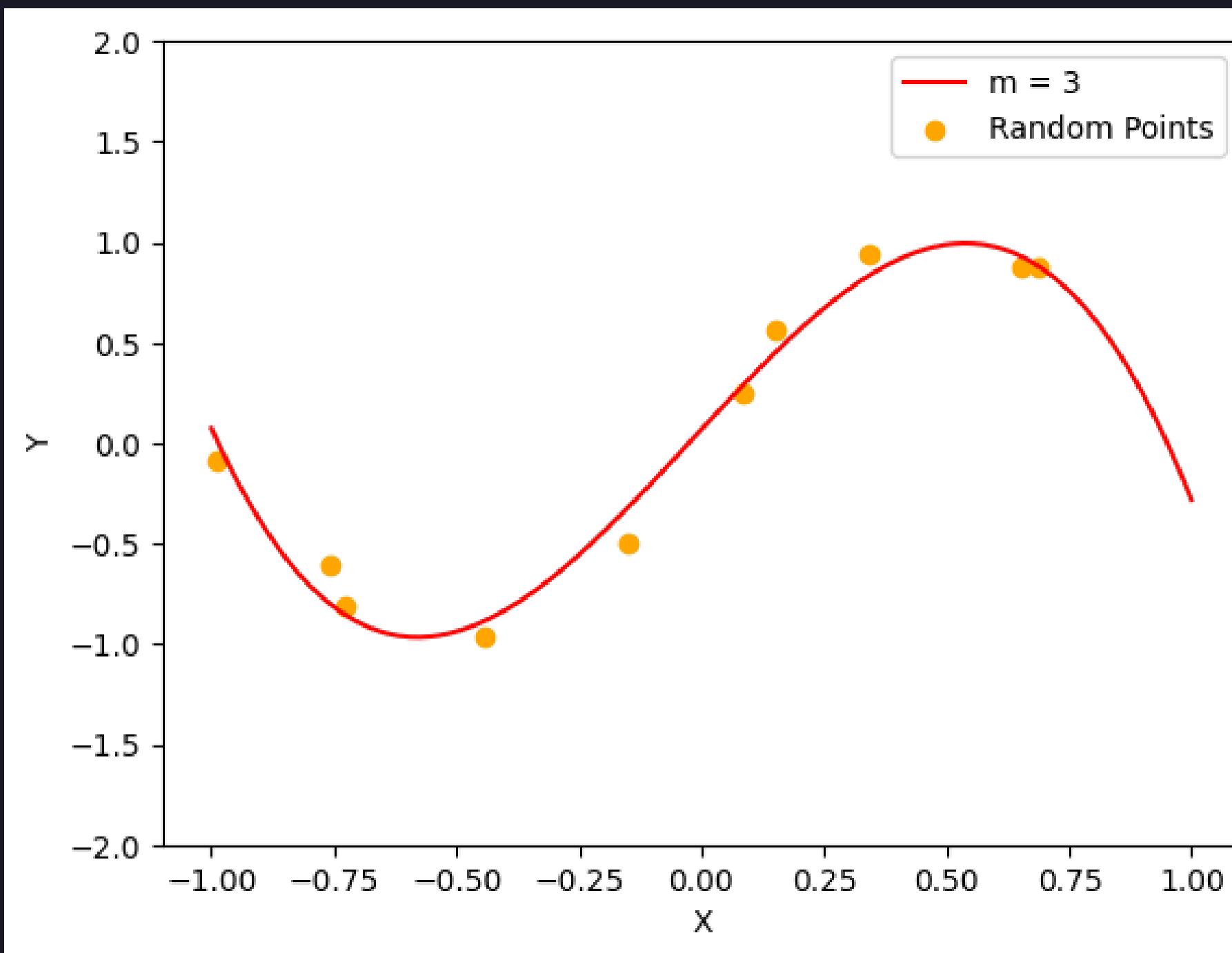


*จุดต่ำสุด มีค่าเข้าใกล้ 0 เรื่อยๆ เมื่อ λ ทิค่าเพิ่มมากขึ้น

2. เขียนโปรแกรมแสดงความสัมพันธ์ระหว่าง complexity
ของ model และ E_{train} , E_{test} ของ ridge regression

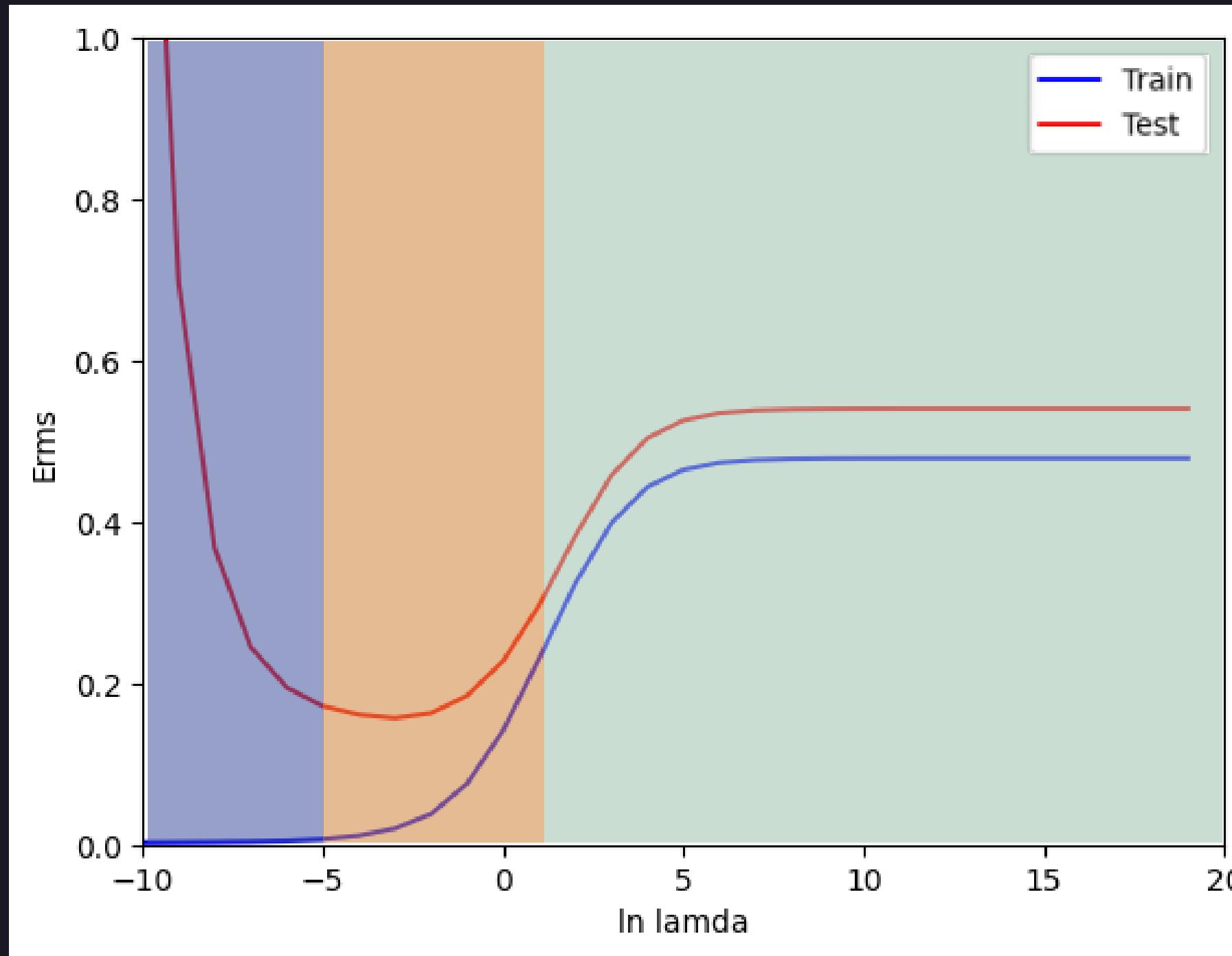
กราฟความสัมพันธ์ระหว่าง X และ Y โดย Degree ที่มากขึ้น

Sample Size = 10, Degree = 3 , 10
Noise = 0 - 0.1



Degree ที่ 10 จะเกิด overfitting

กราฟความสัมพันธ์ระหว่าง E_{train} และ E_{test}



Sample Size: 10
Degree: 10
Noise: 0 - 0.1

Overfitting

Goodfit/robust

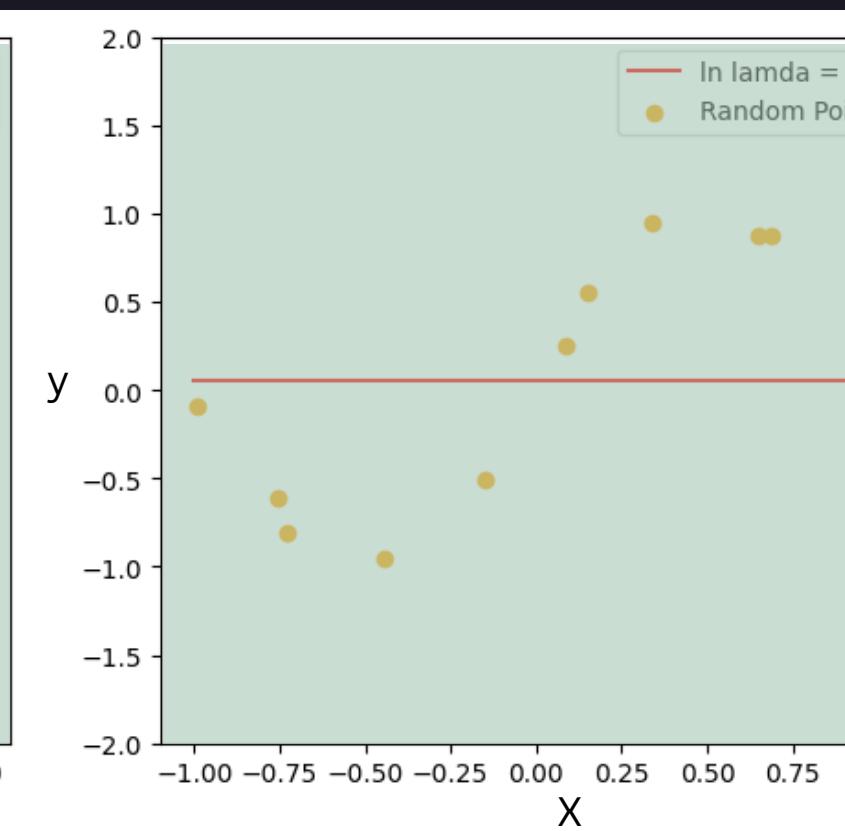
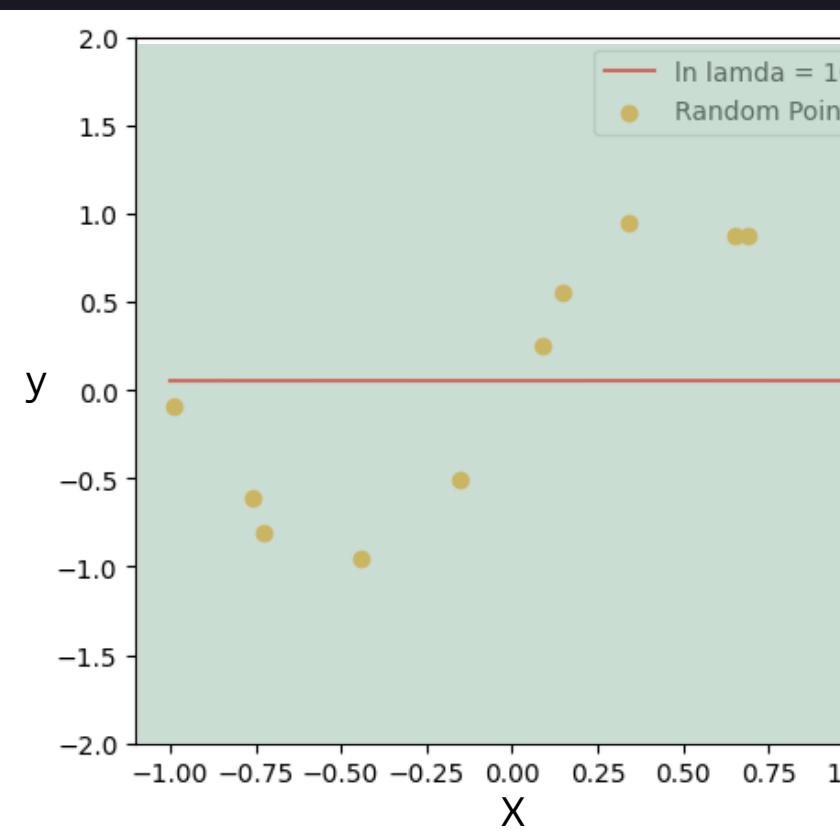
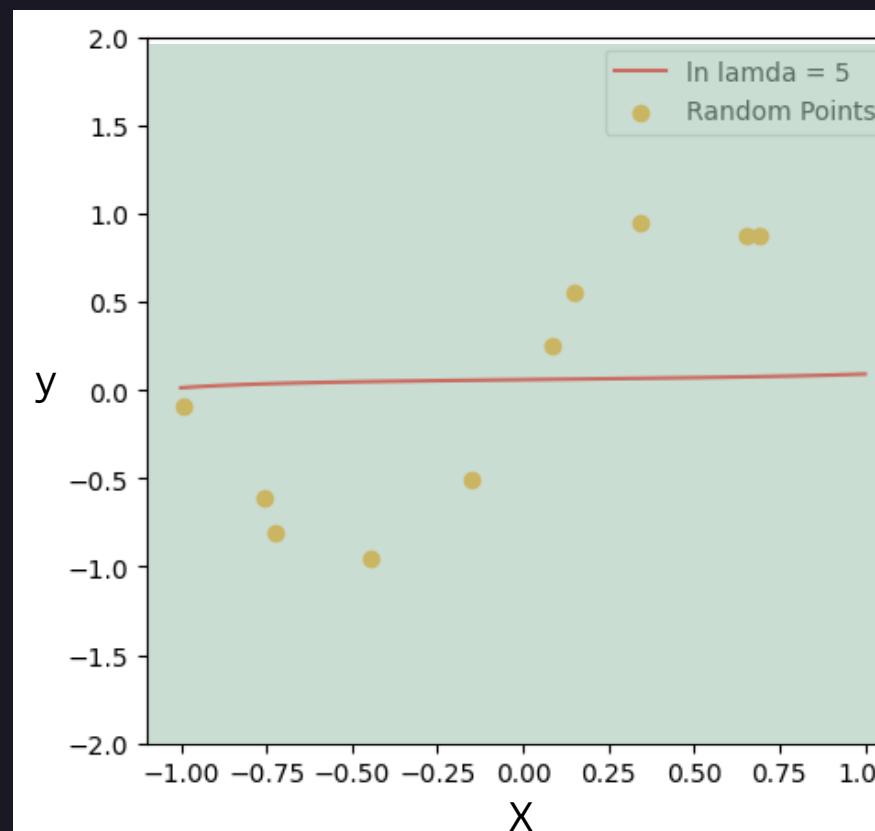
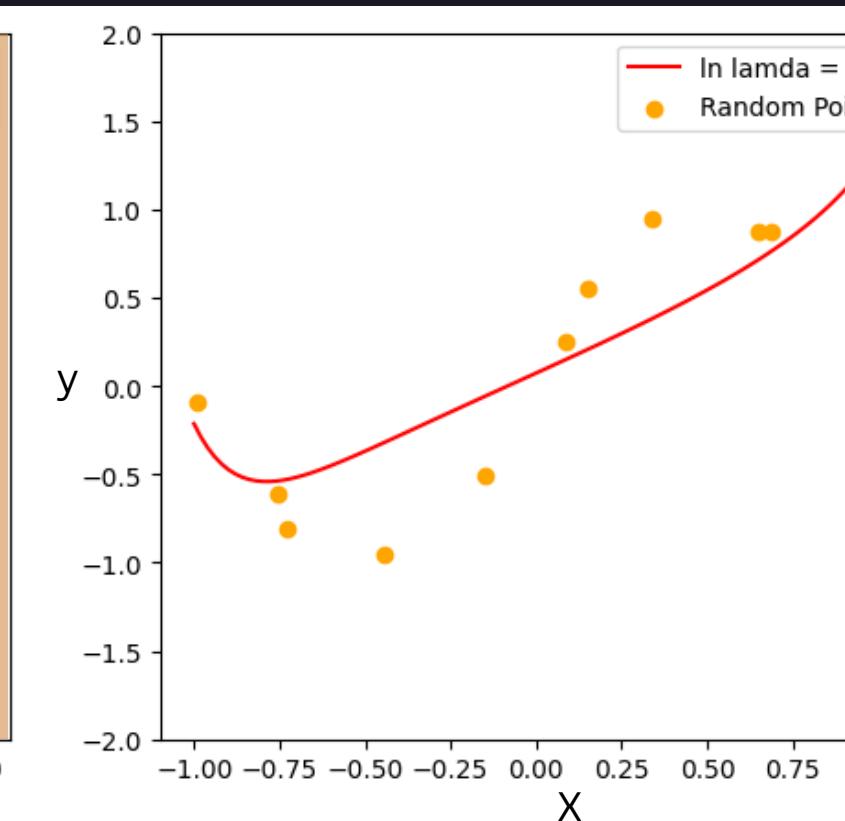
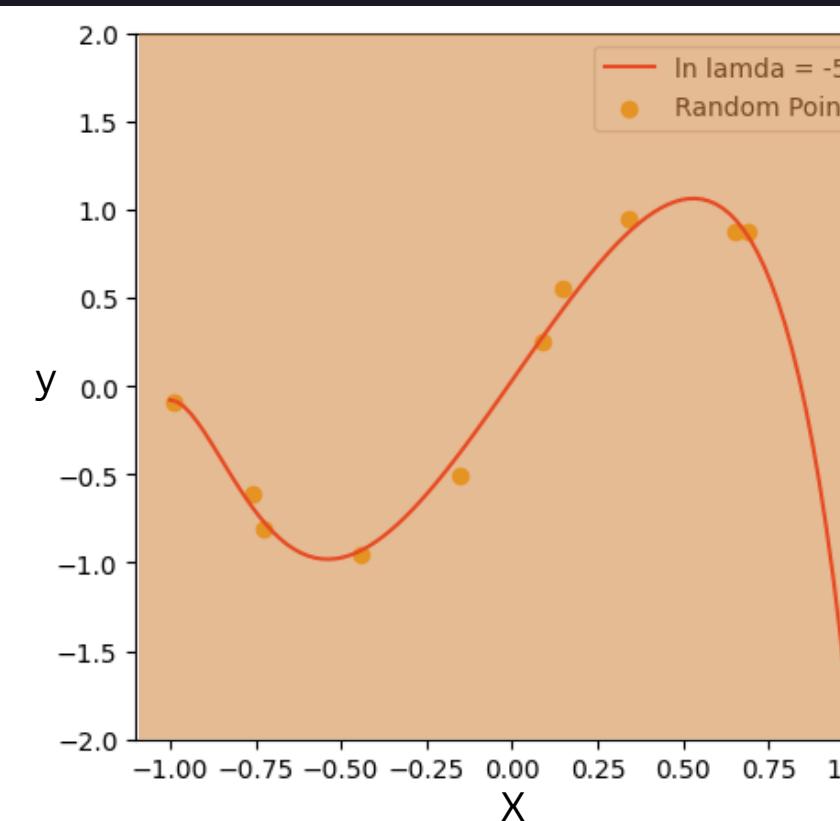
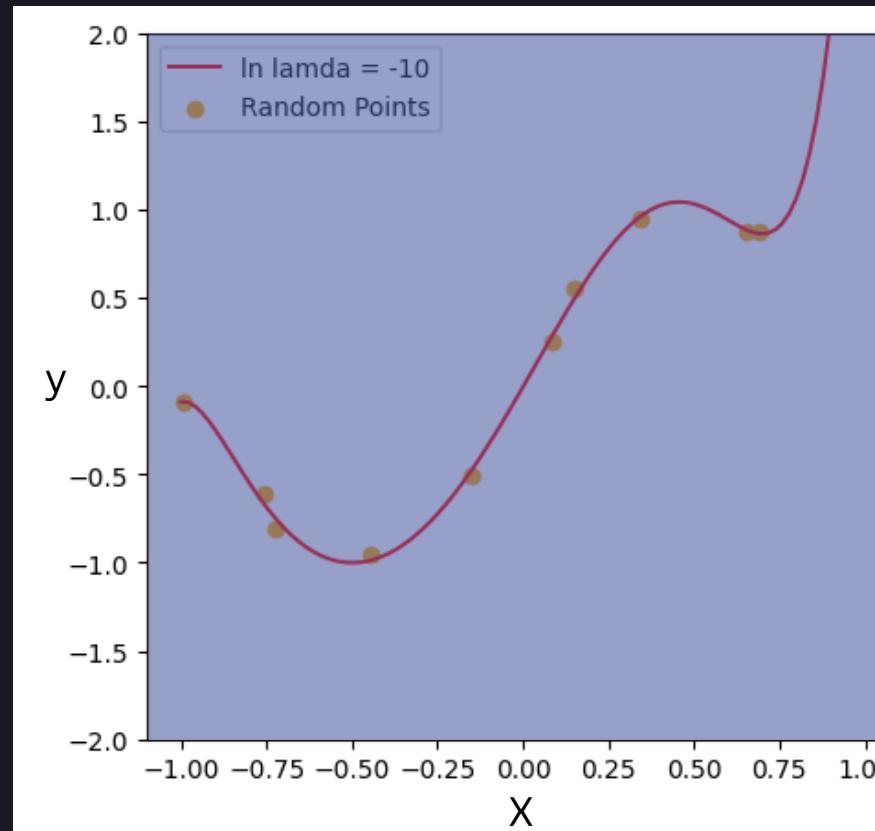
Underfit

$E_{in} = \text{train}$, $E_{out} = \text{test}$

Overfitting

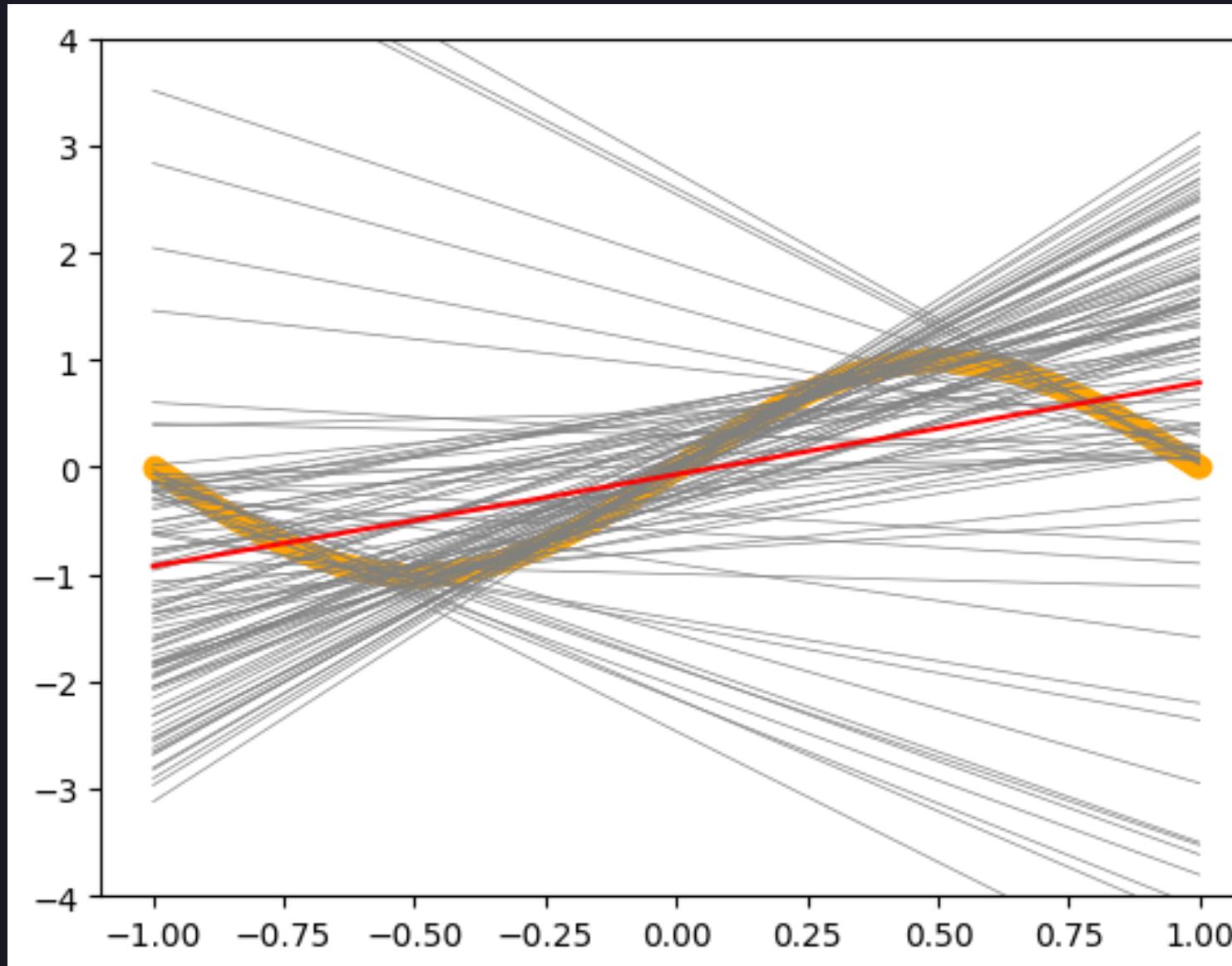
Goodfit/robust

Underfit



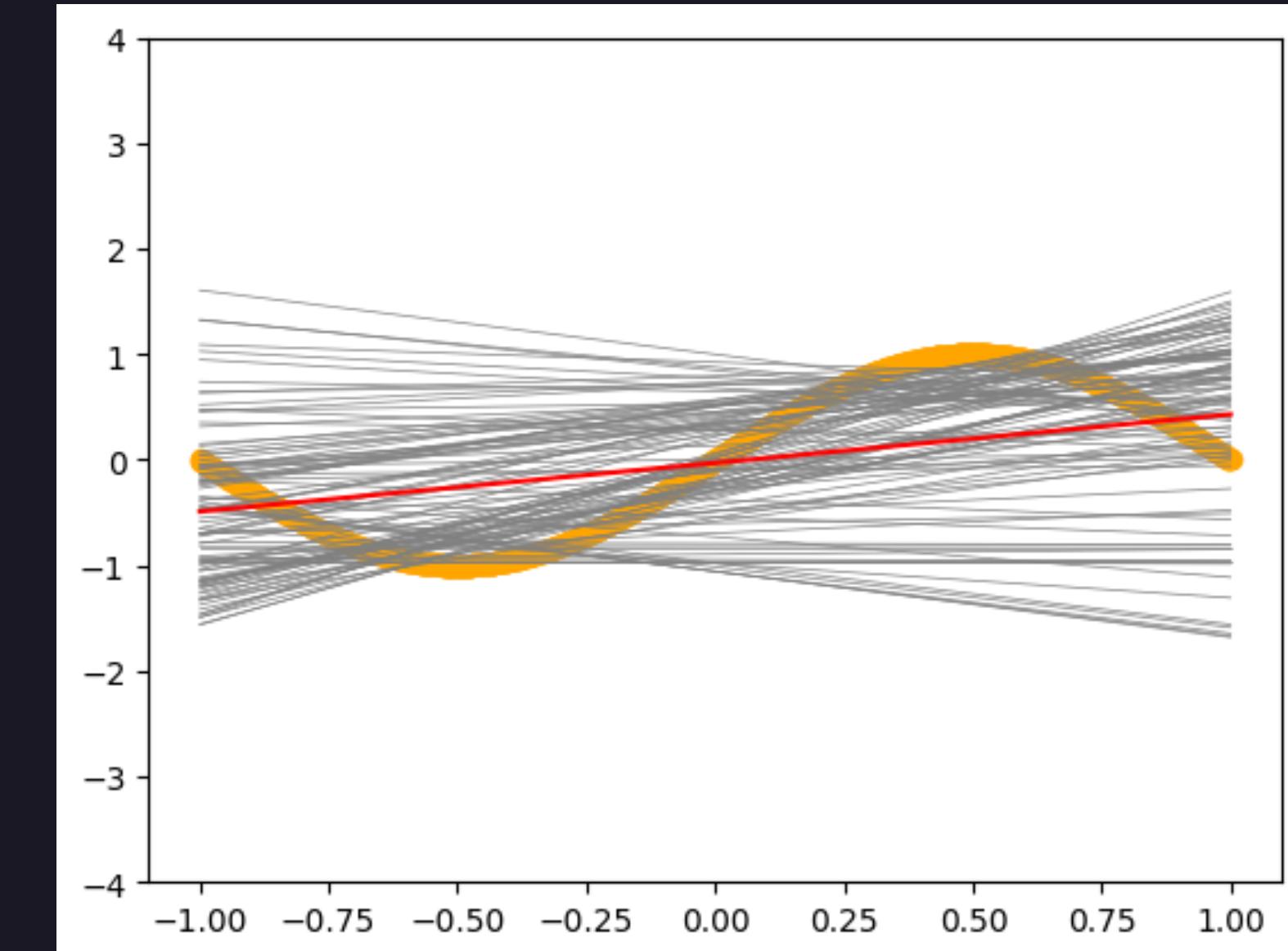
**3.เขียนโปรแกรมสำหรับเปรียบเทียบค่าคลาดเคลื่อนนอก
ตัวอย่างของแบบจำลองเชิงเส้นที่มีและไม่มีการทำให้เป็นปกติ**

Linear Regression



Without regularization

Ridge Regression



With regularization

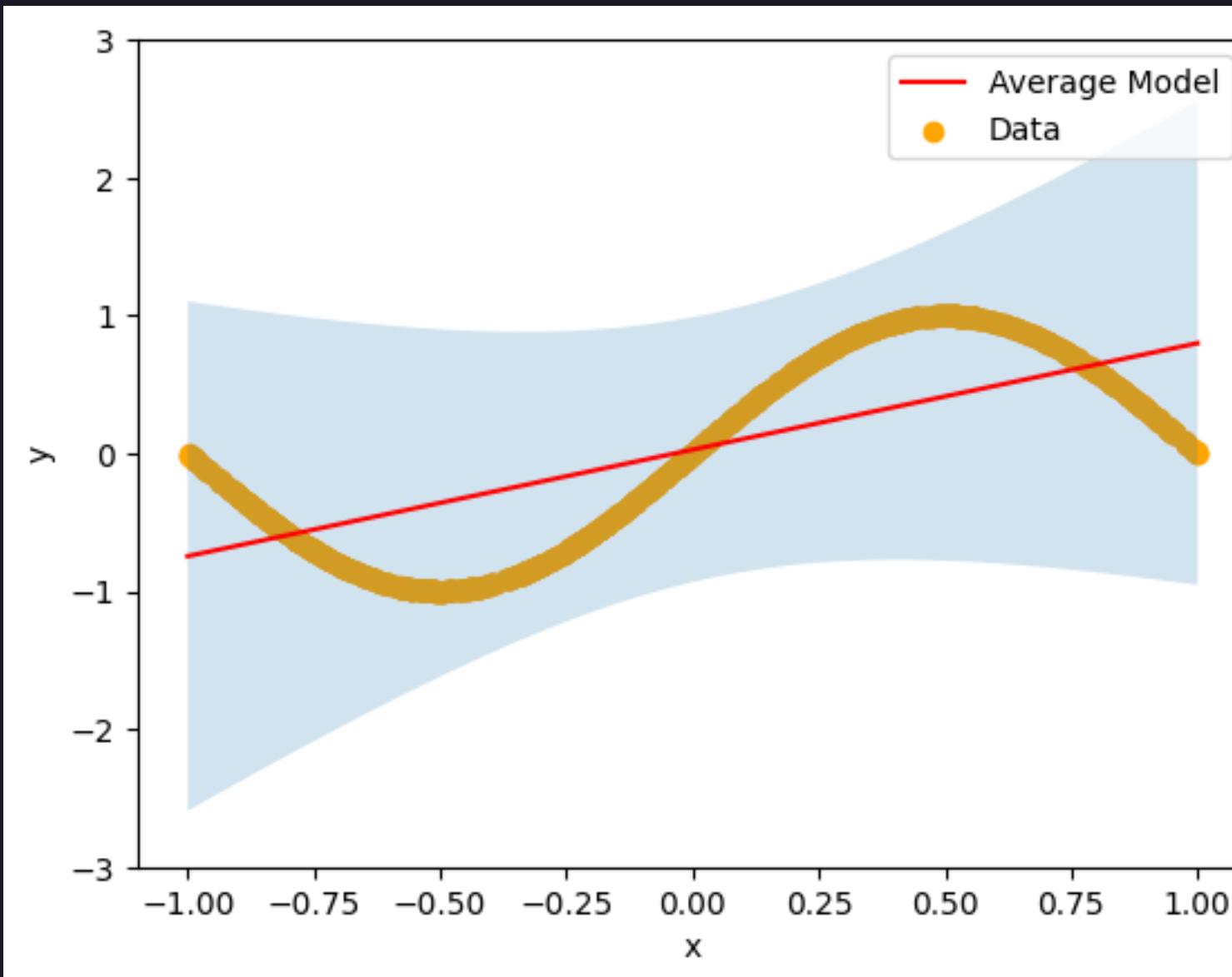
*Data info: $X = \text{uniform}(-1, 1)$ $y = \sin(\pi X)$

Linear Regression

Bias : 0.208

Variance : 1.697

Eout : 1.905



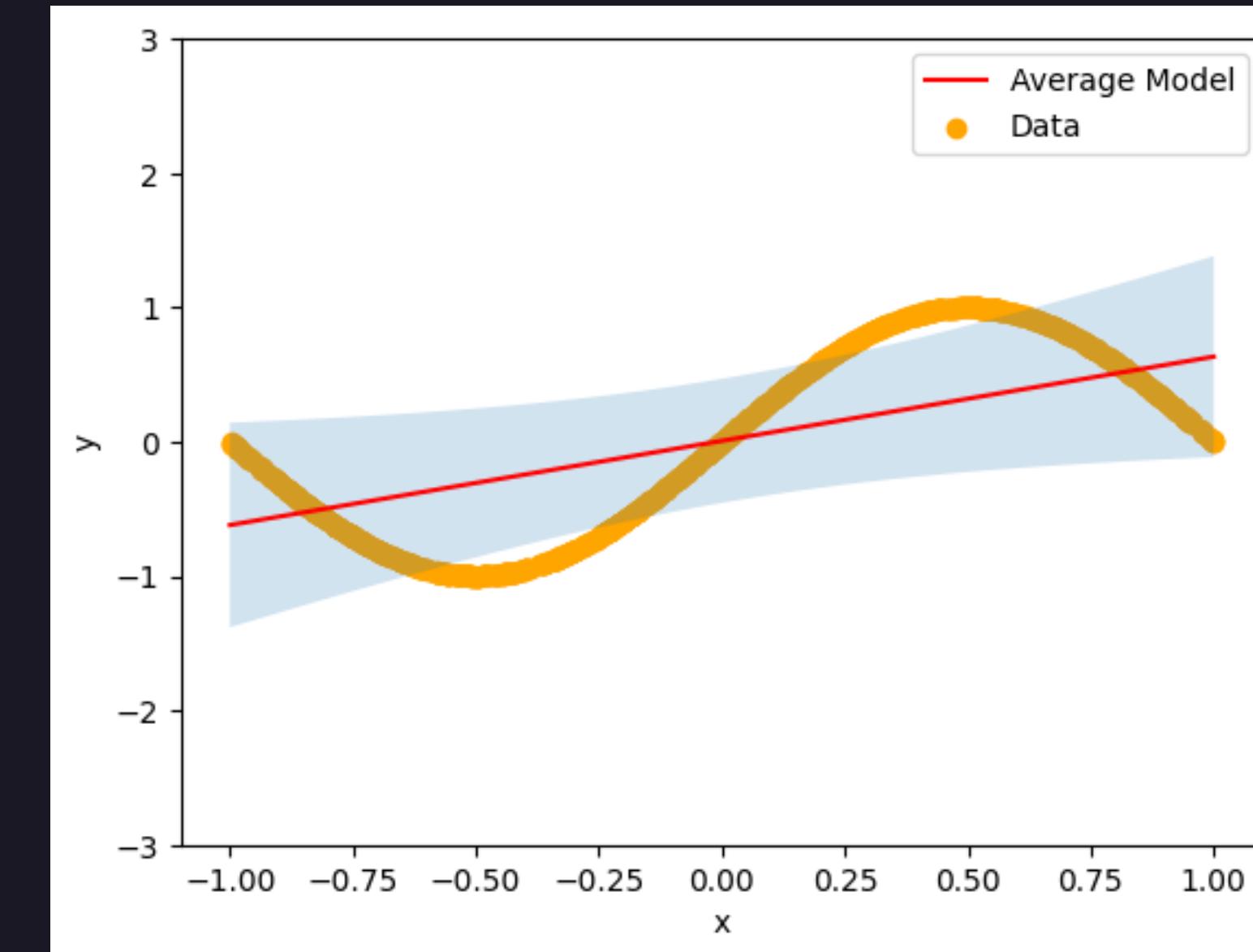
Without regularization

Ridge Regression, Where $\lambda = 0.1$

Bias : 0.232

Variance : 0.332

Eout : 0.564



With regularization

*Data info: X = uniform(-1, 1) y = sin(pi X)

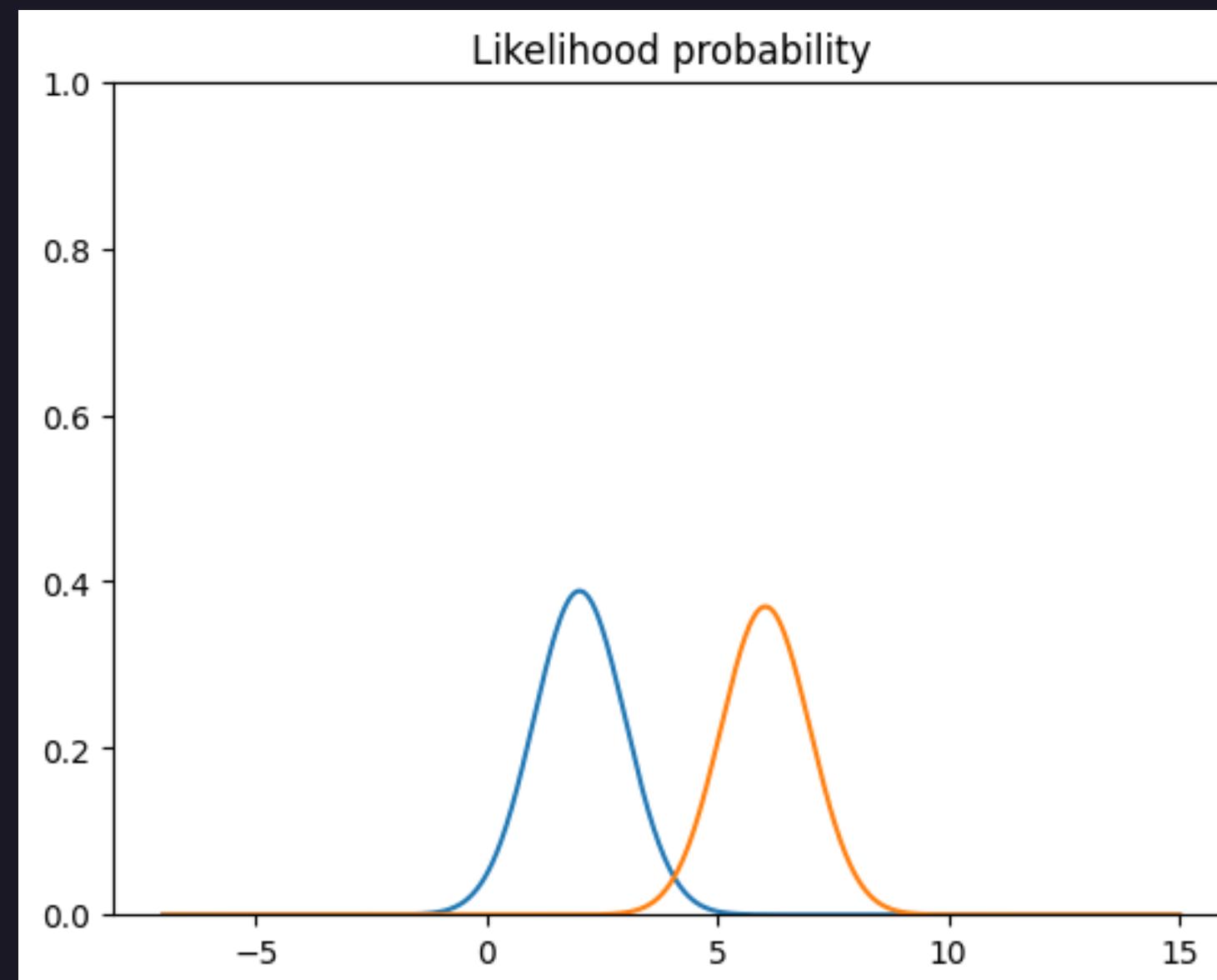
แบบฝึกหัดเขียนโปรแกรมครั้งที่ 6

ตัวจำแนกแบบเบส

1. เขียนโปรแกรมสำหรับสร้างตัวจำแนกแบบเบสสำหรับการแยกแจงประเภทตัวแปรเดียว กรณีที่ความแปรปรวนของทั้งสองคลาสเท่ากัน

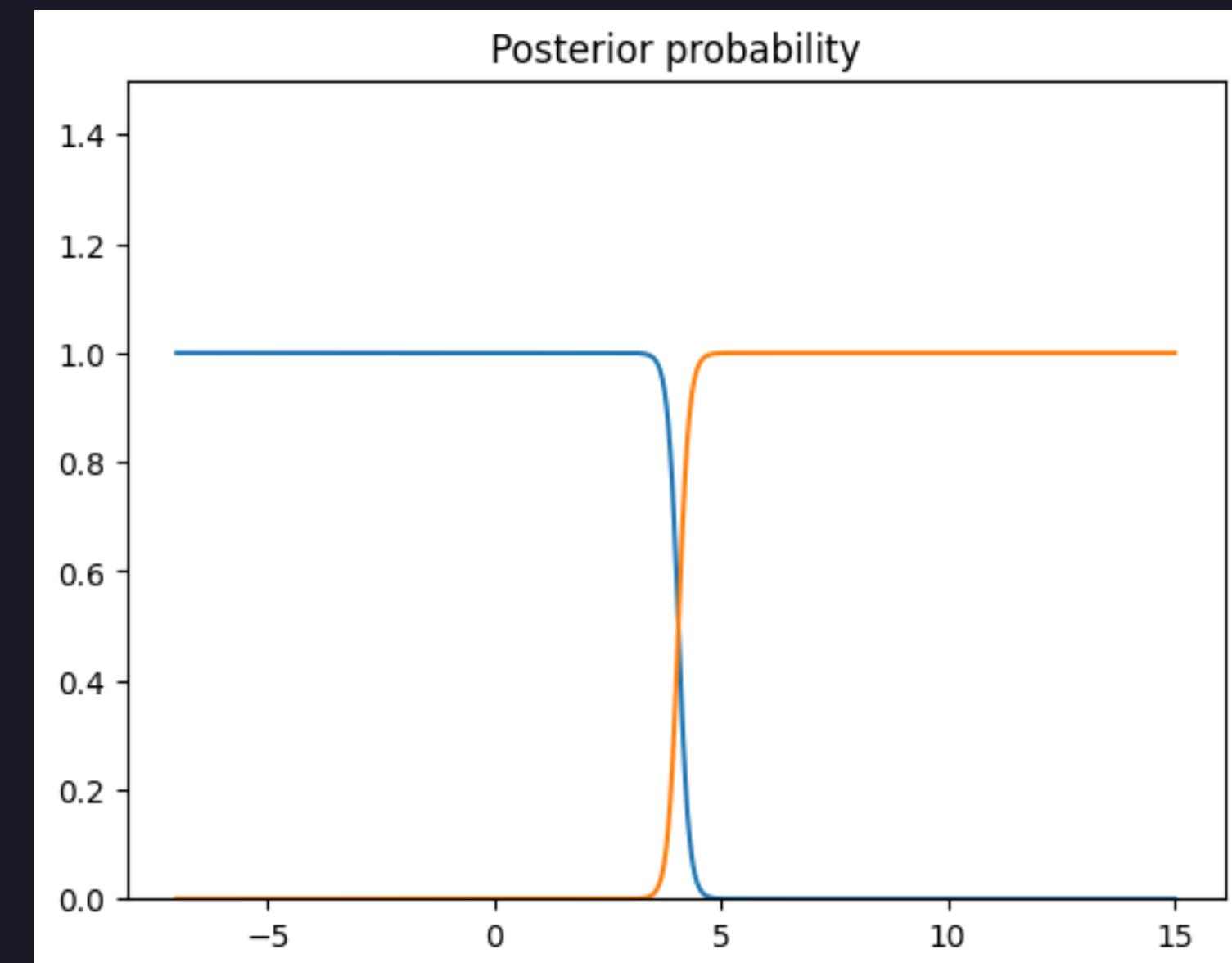
ไม่ Fix Distribution

Likelihood probability



case 1: $\text{var1} = \text{var2}$

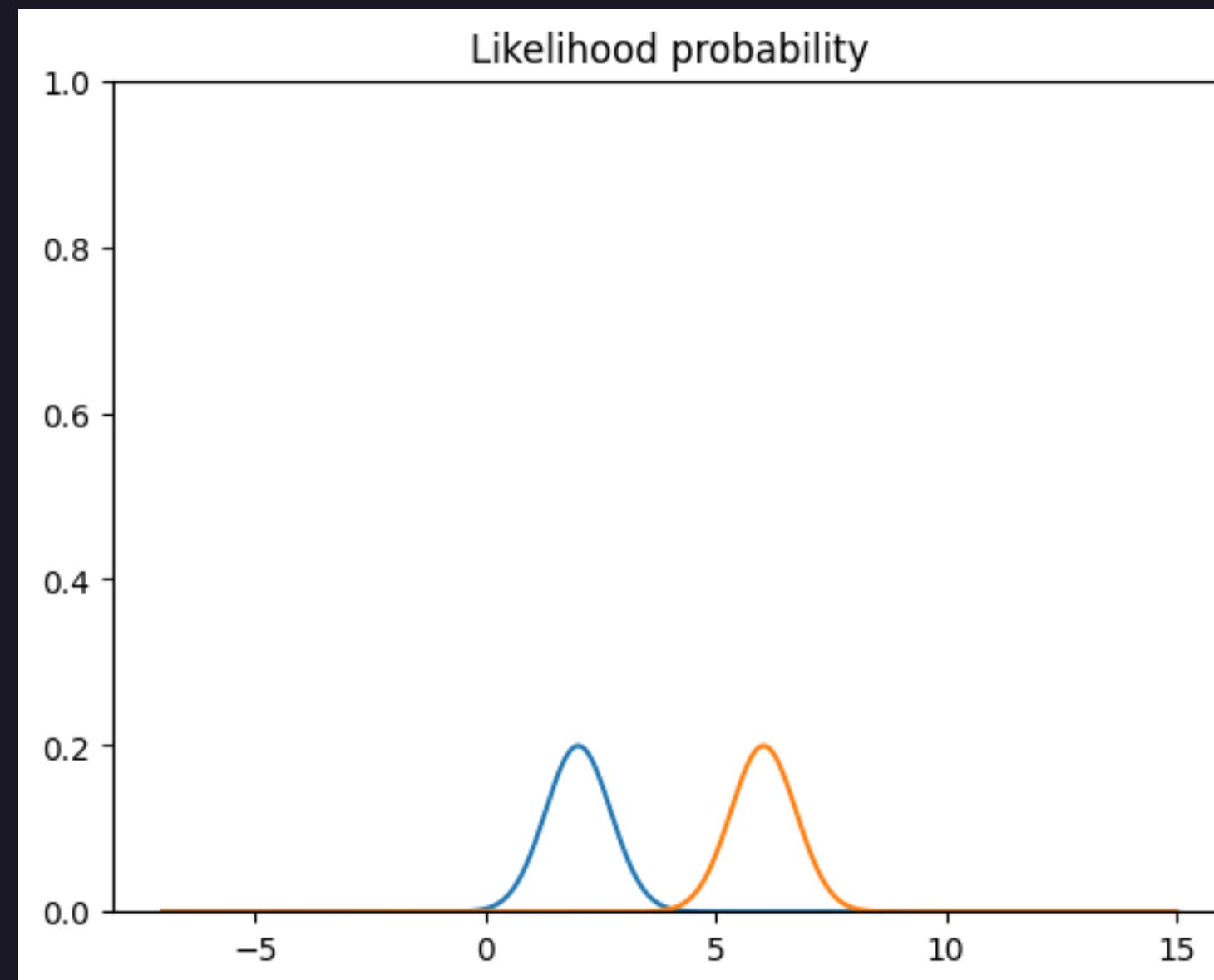
Posterior probability



1. เขียนโปรแกรมสำหรับสร้างตัวจำแนกแบบเบสสำหรับการแยกแจงประเภทตัวแปรเดียว กรณีที่ความแปรปรวนของทั้งสองคลาสเท่ากัน

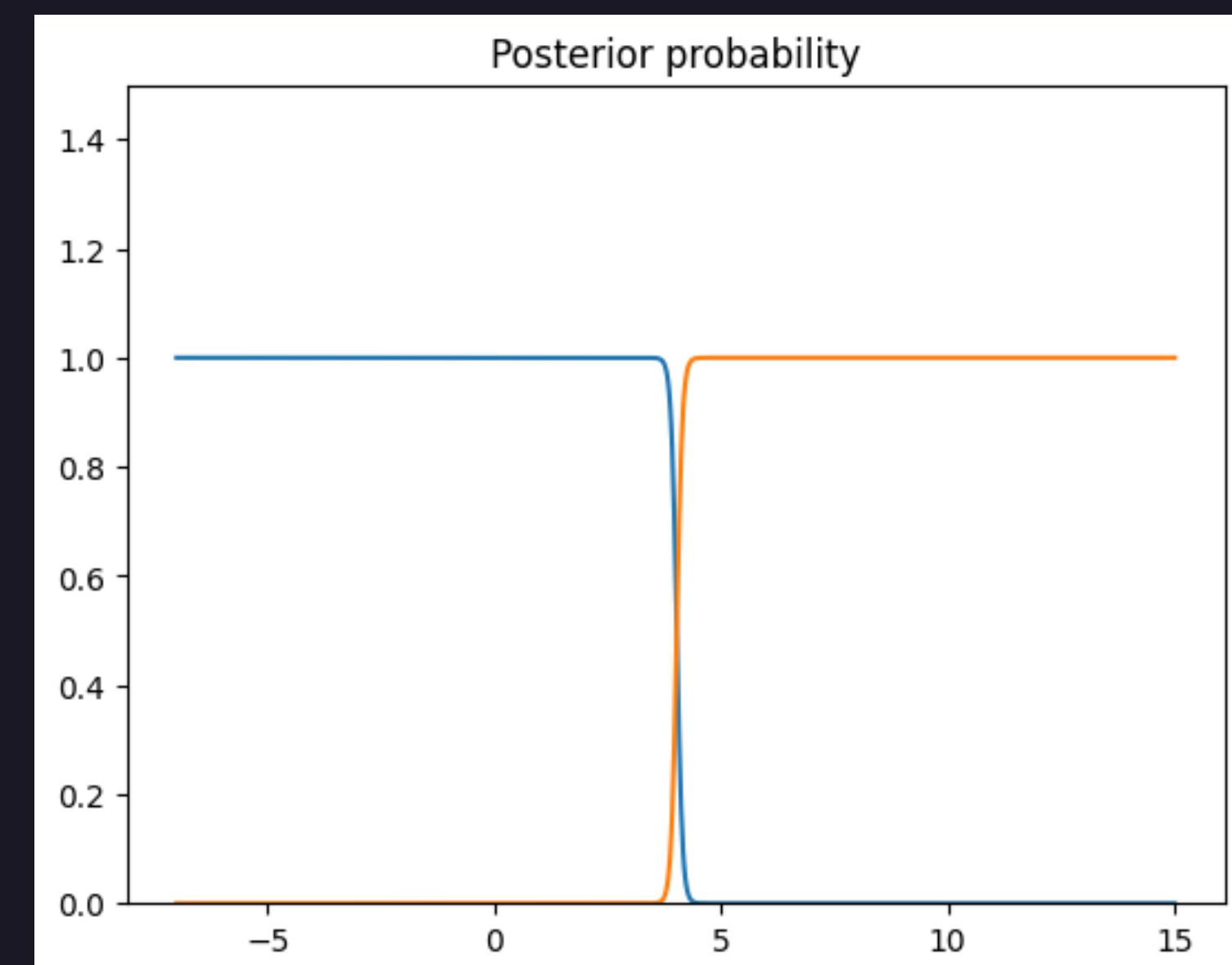
Fix Distribution

Likelihood probability



case 1: $\text{var1} = \text{var2}$

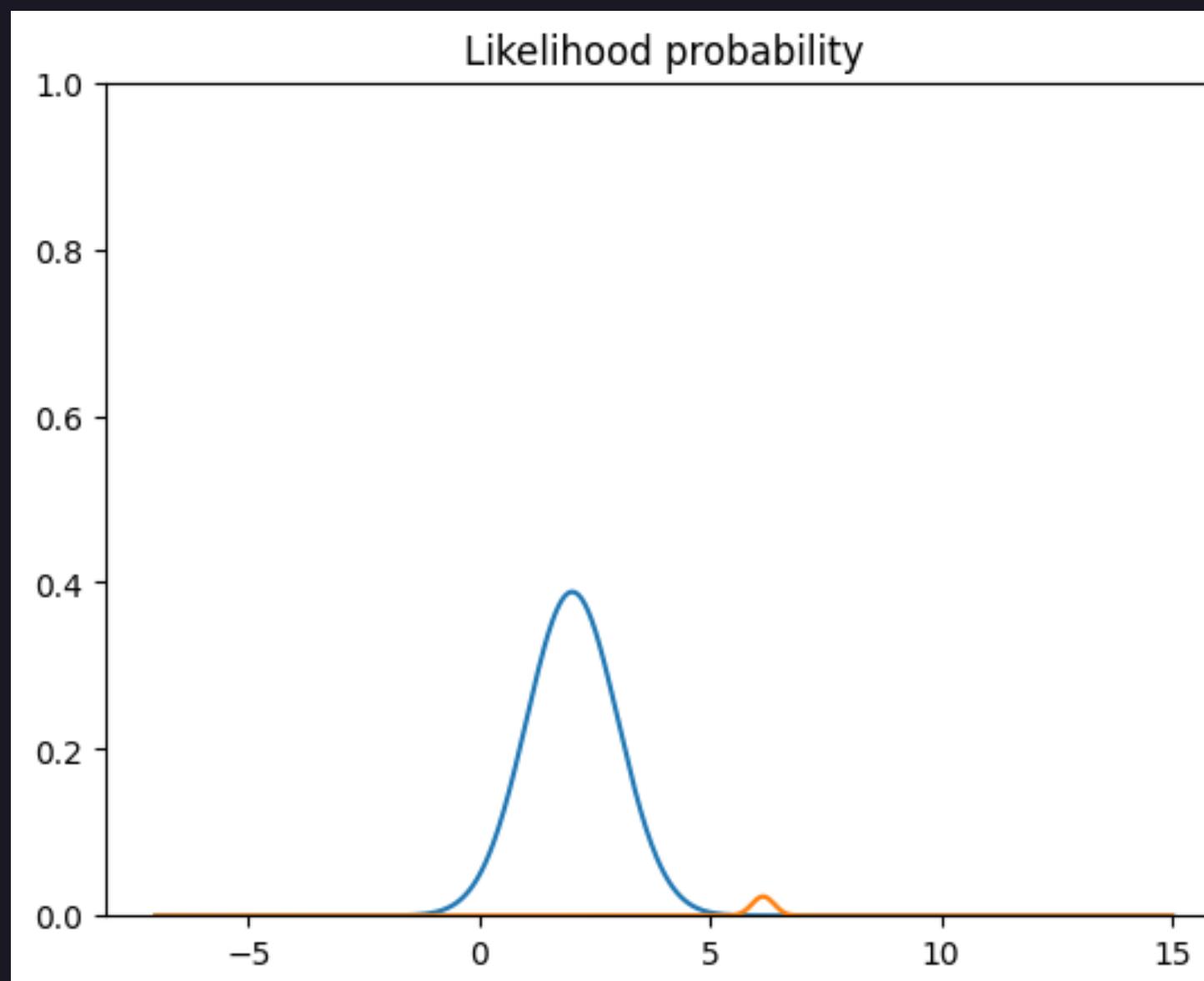
Posterior probability



2. เขียนโปรแกรมสำหรับสร้างตัวจำแนกแบบเบสสำหรับการแยกแจงประเภทตัวแปรเดียว กรณีที่ความแปรปรวนของทั้งสองคลาสไม่เท่ากัน

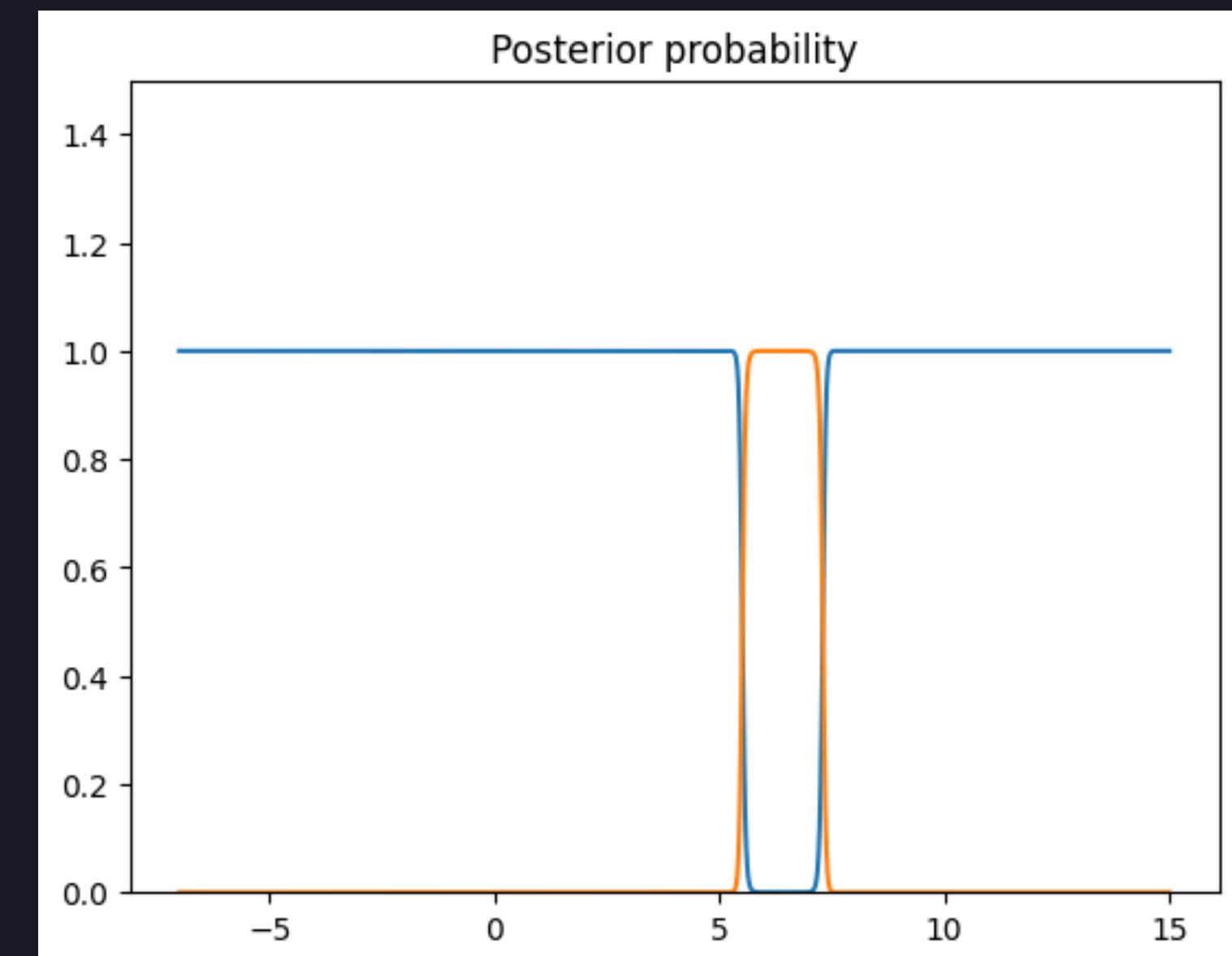
ไม่ Fix Distribution

Likelihood probability



case 2: $\text{var1} \neq \text{var2}$

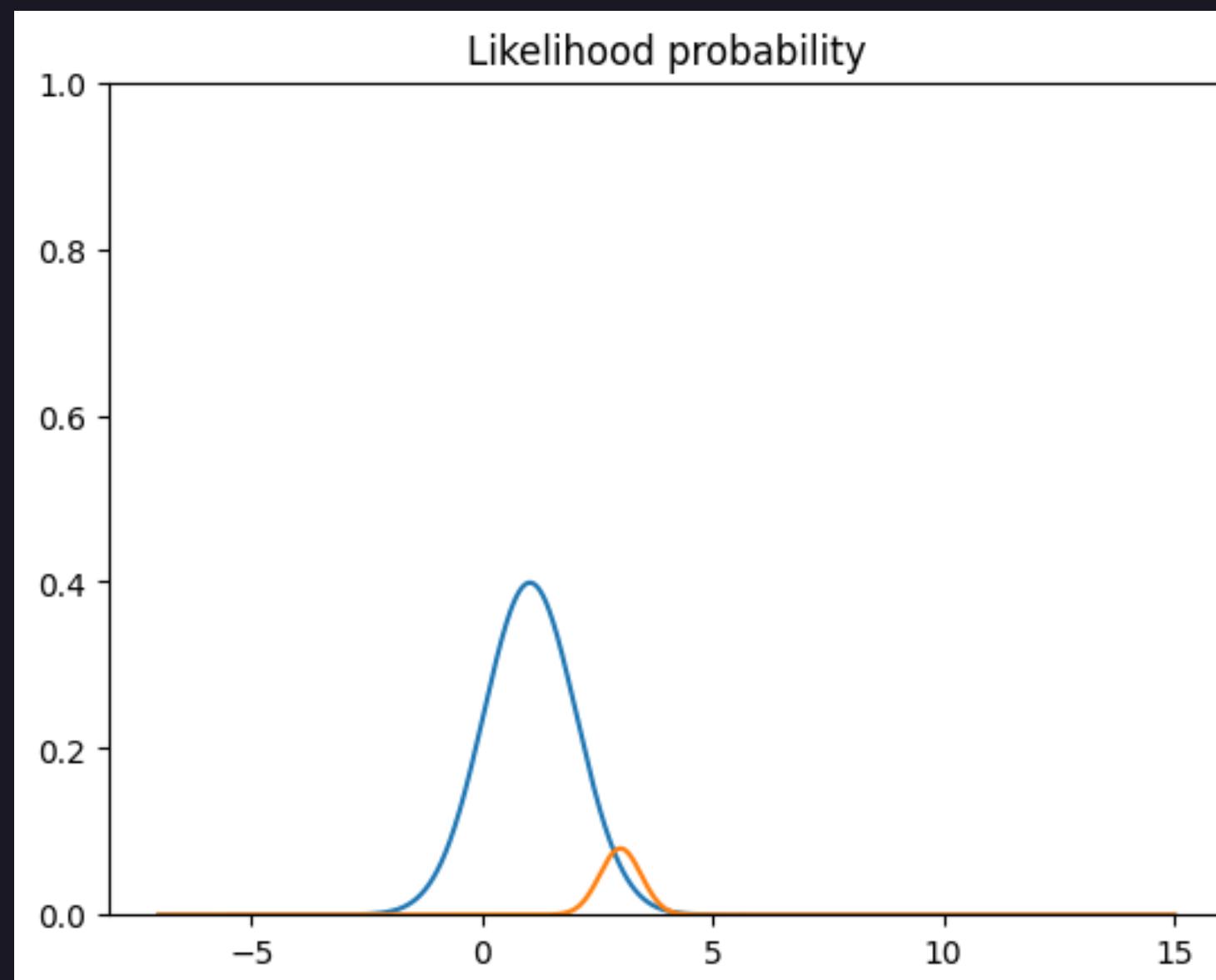
Posterior probability



2. เขียนโปรแกรมสำหรับสร้างตัวจำแนกแบบเบสสำหรับการแยกแจงประเภทตัวแปรเดียว กรณีที่ความแปรปรวนของทั้งสองคลาสไม่เท่ากัน

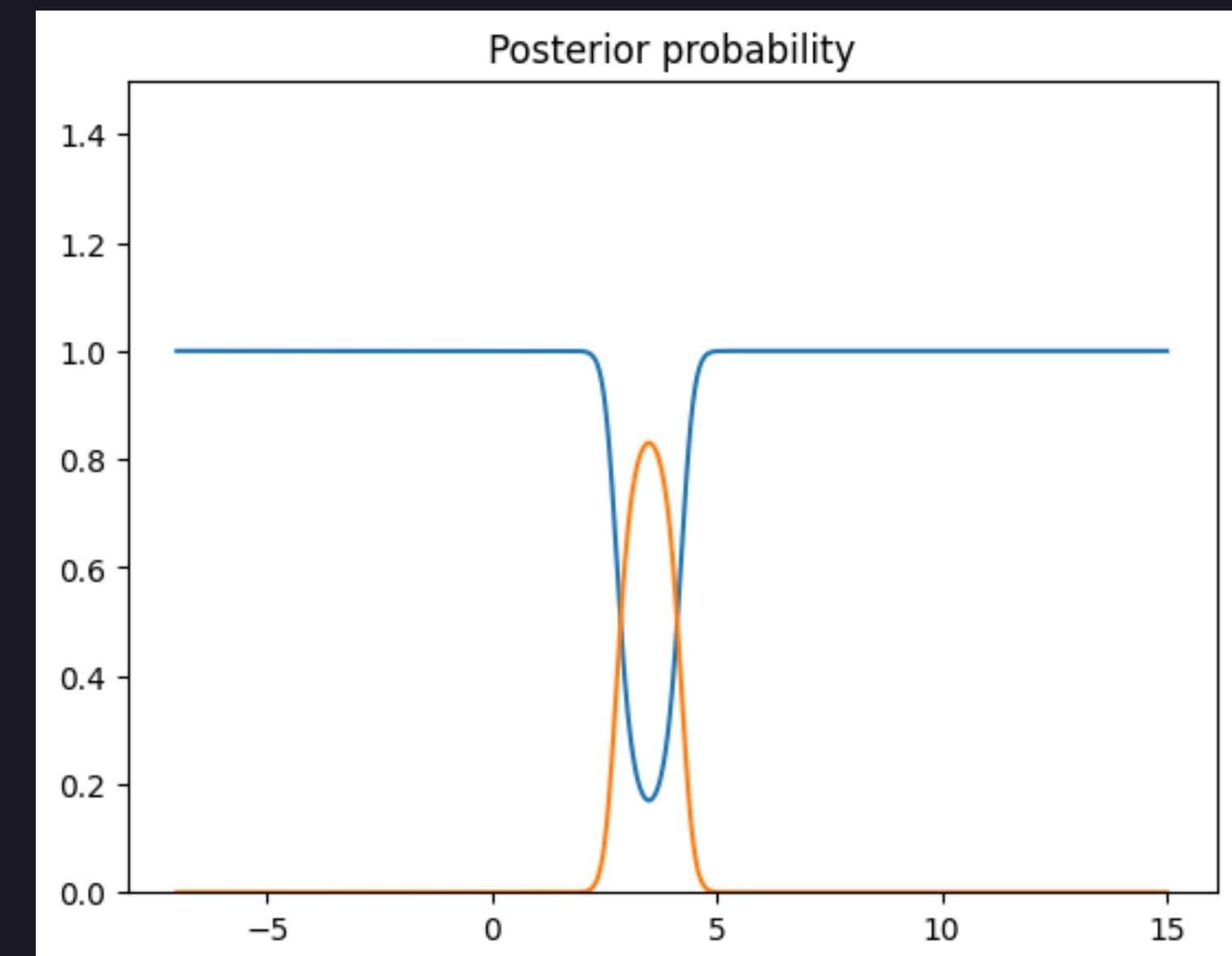
Fix Distribution

Likelihood probability



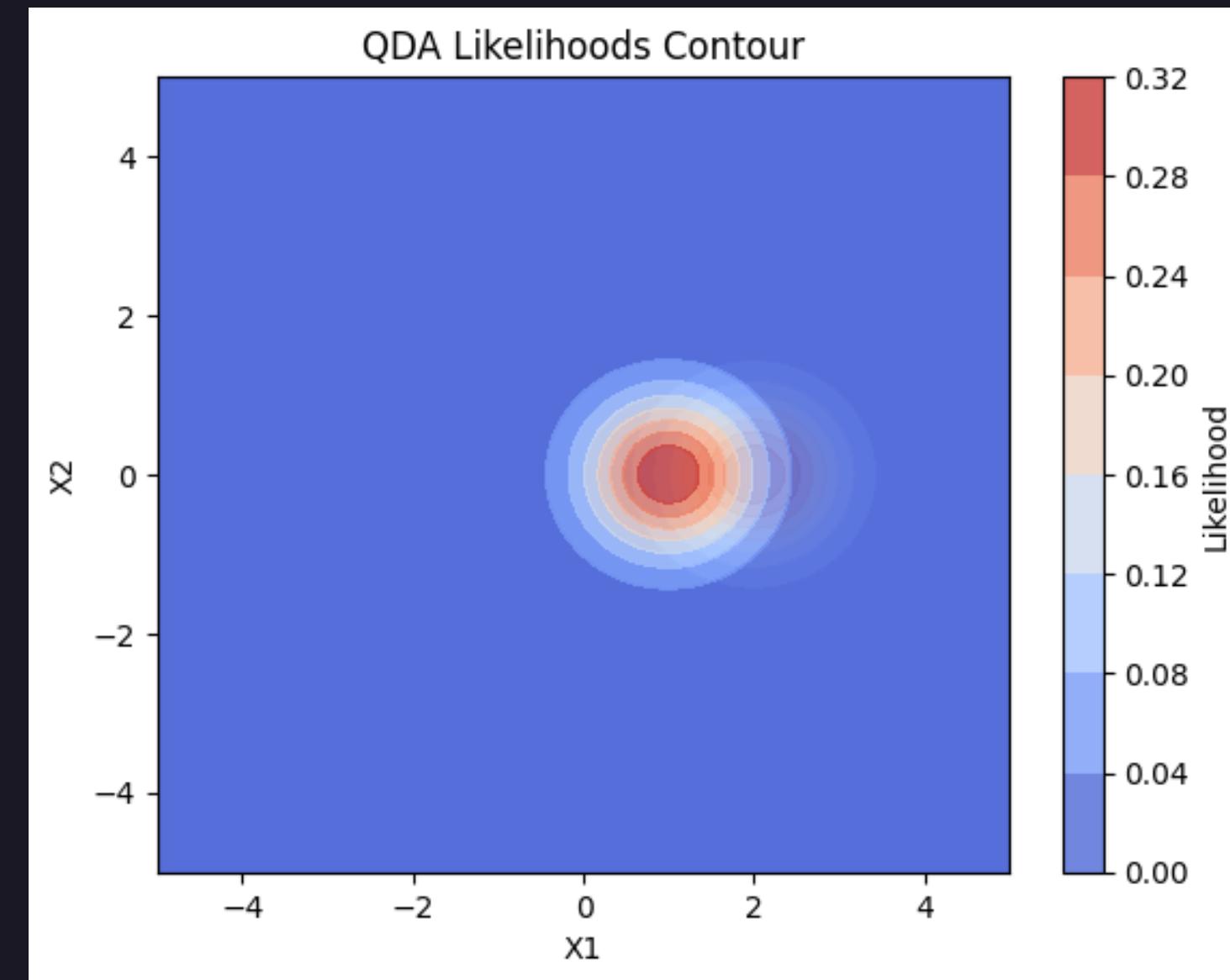
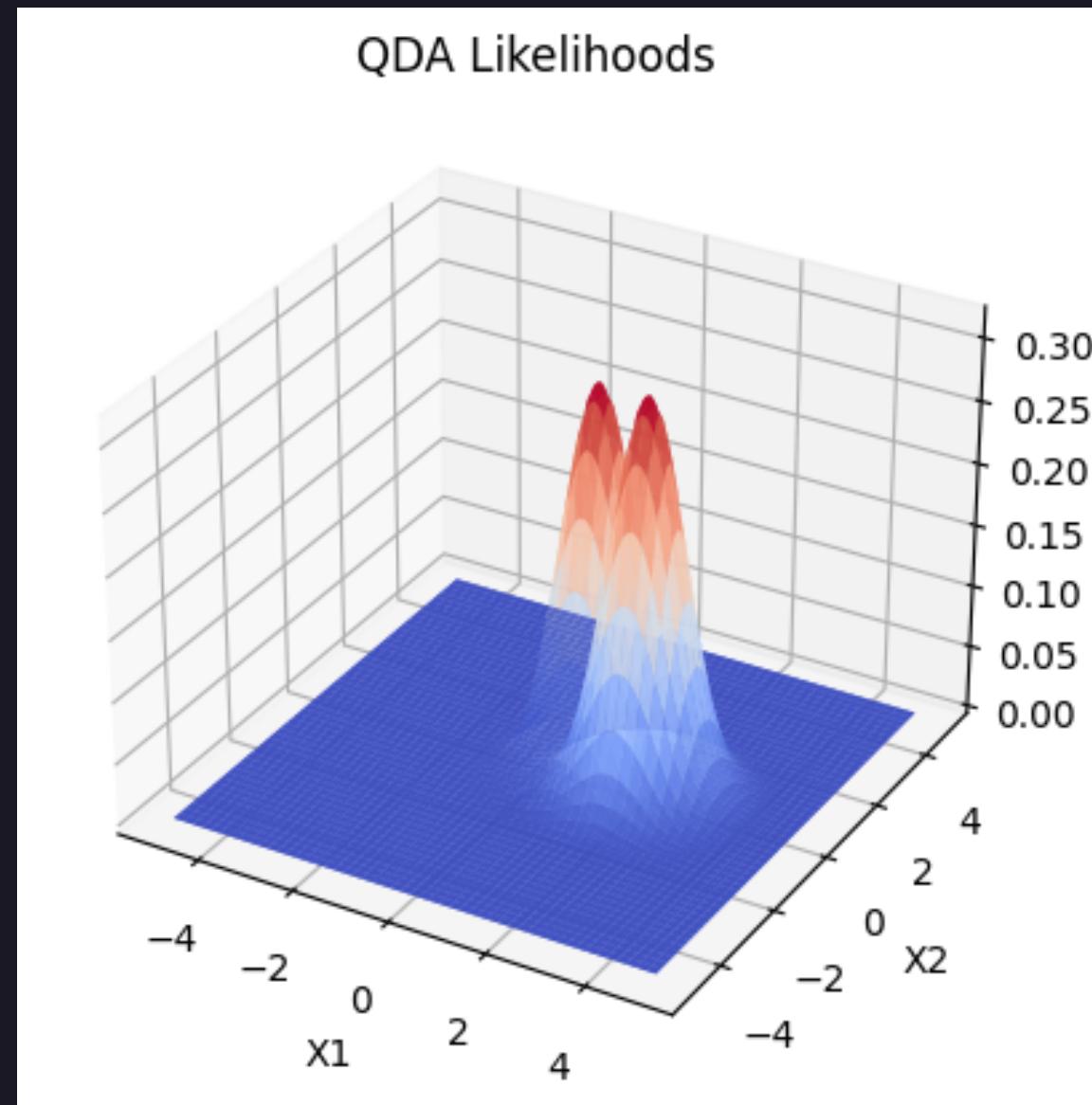
case 2: $\text{var1} \neq \text{var2}$

Posterior probability



3. เขียนโปรแกรมสำหรับสร้างตัวจำแนกกำลังสอง

Spherical Covariance

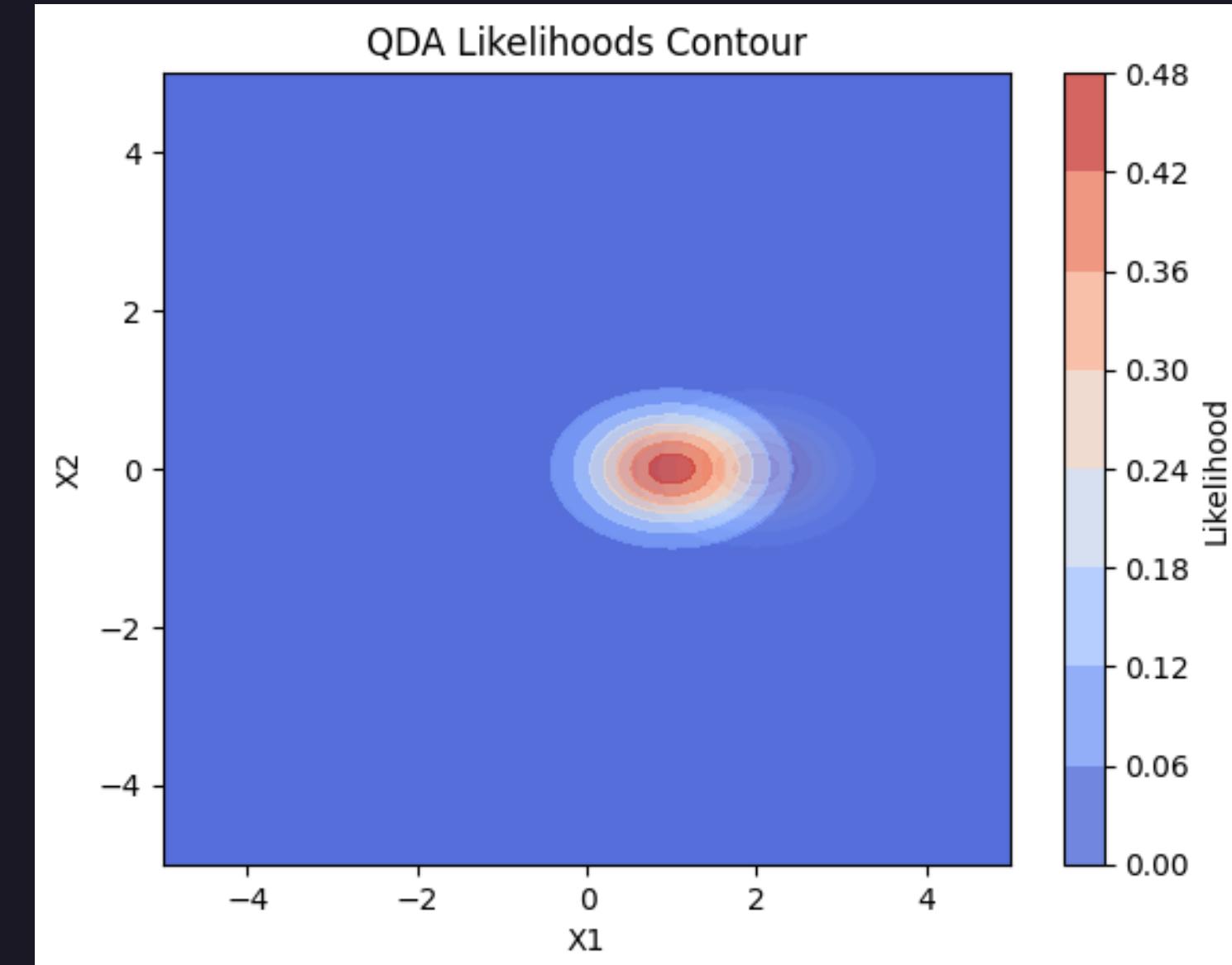
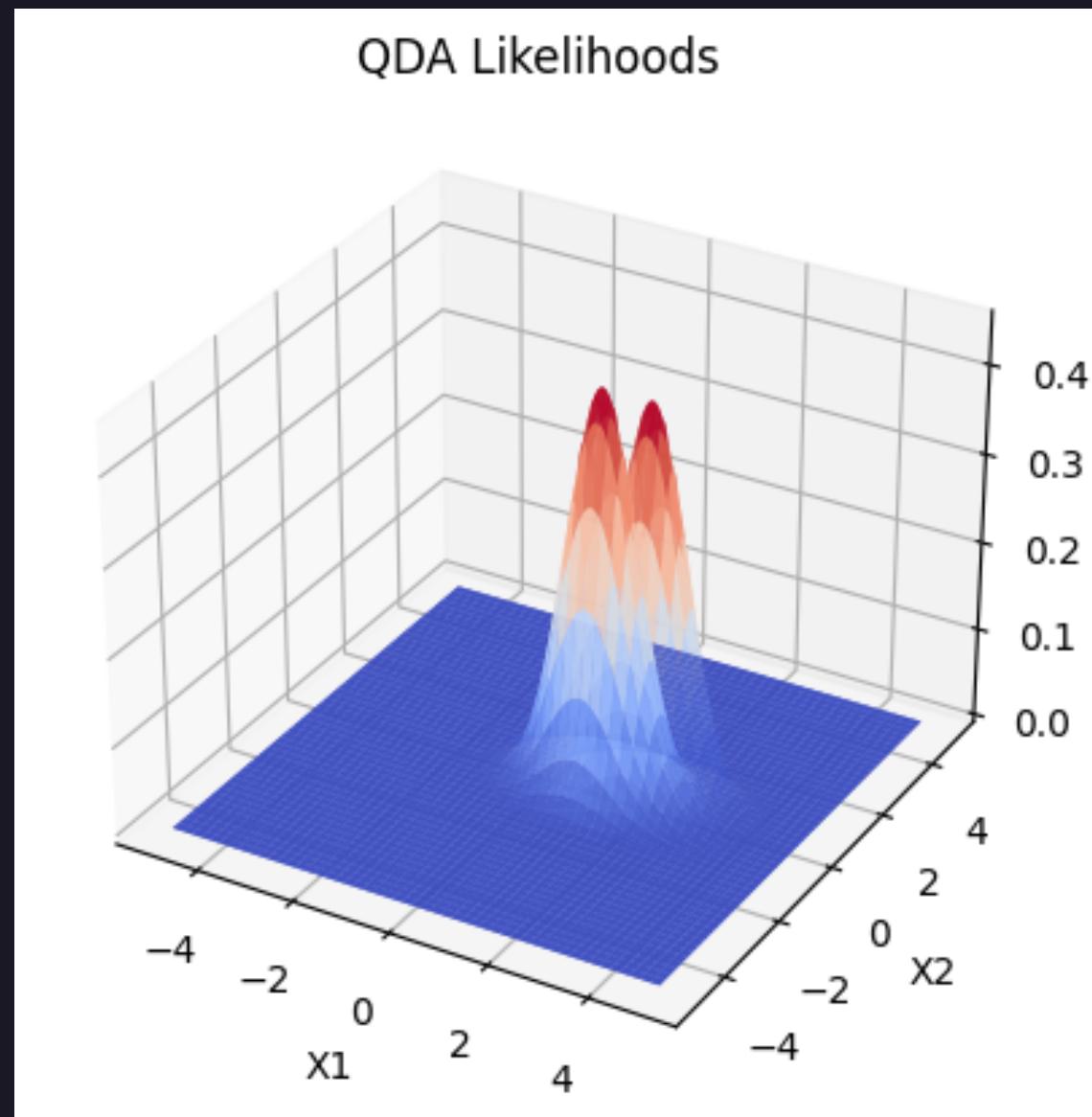


$$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

$p(c_1) = p(c_2)$
covariance เท่ากัน

3.เขียนโปรแกรมสำหรับสร้างตัวจำแนกกำลังสอง

Ellipsoid Covariance



$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

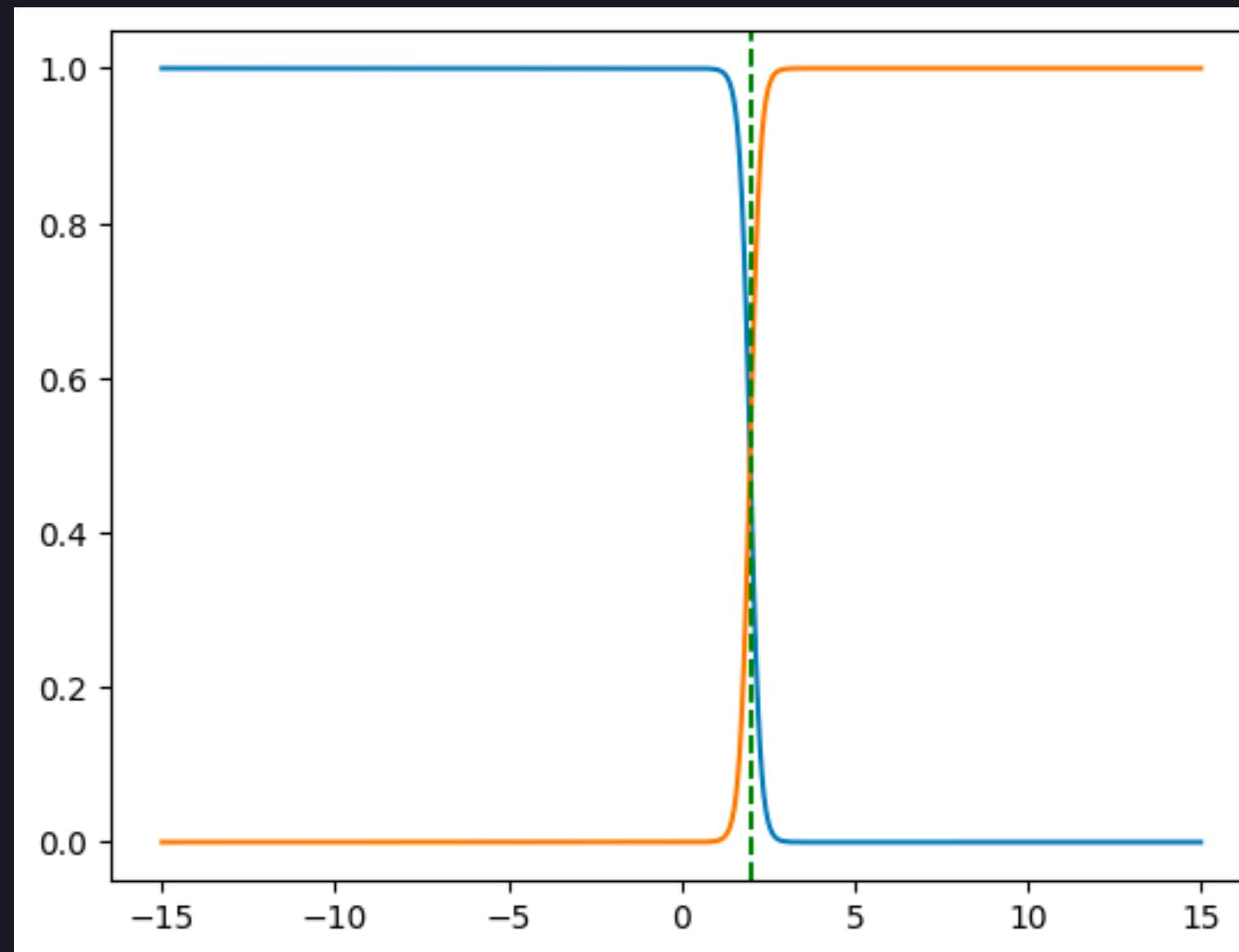
$p(c_1) \neq p(c_2)$
covariance ไม่เท่ากัน
ความแปรปรวน $x_1 > x_2$

4. เขียนโปรแกรมสำหรับสร้างตัวจำแนกเชิงเส้น

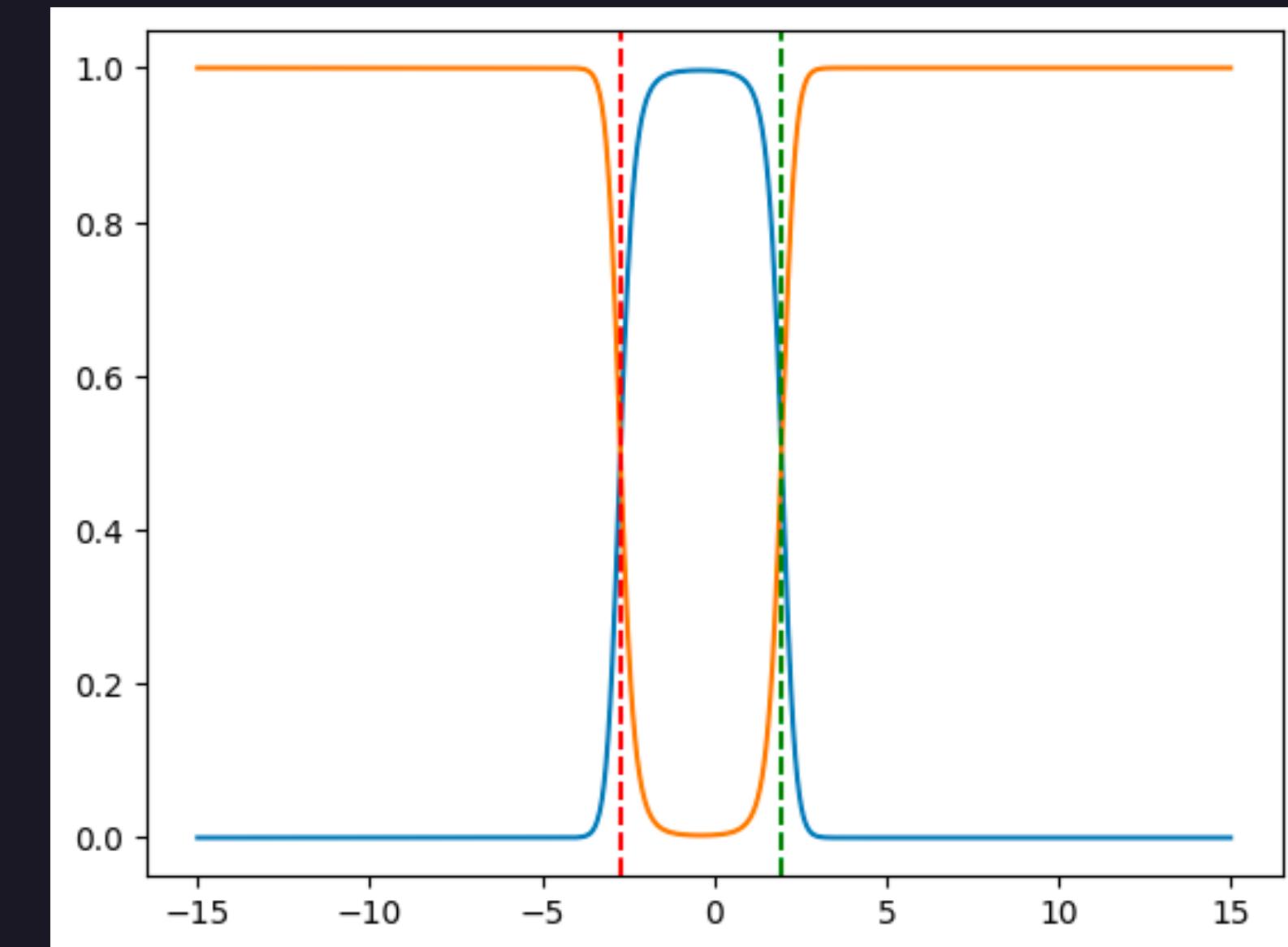
ทั้ง 4 ข้อให้ วาดกราฟ likelihood, posterior และขอบตัดสินใจ โดยกำสองรูปแบบ คือ

- กำหนดค่าพารามิเตอร์ของการแจกแจง
- สุ่มตัวอย่างเพื่อนำมาคำนวณค่าพารามิเตอร์ของการแจกแจง

Decision Boundary



case 1: $\text{var1} = \text{var2}$



case 2: $\text{var1} \neq \text{var2}$

From Example

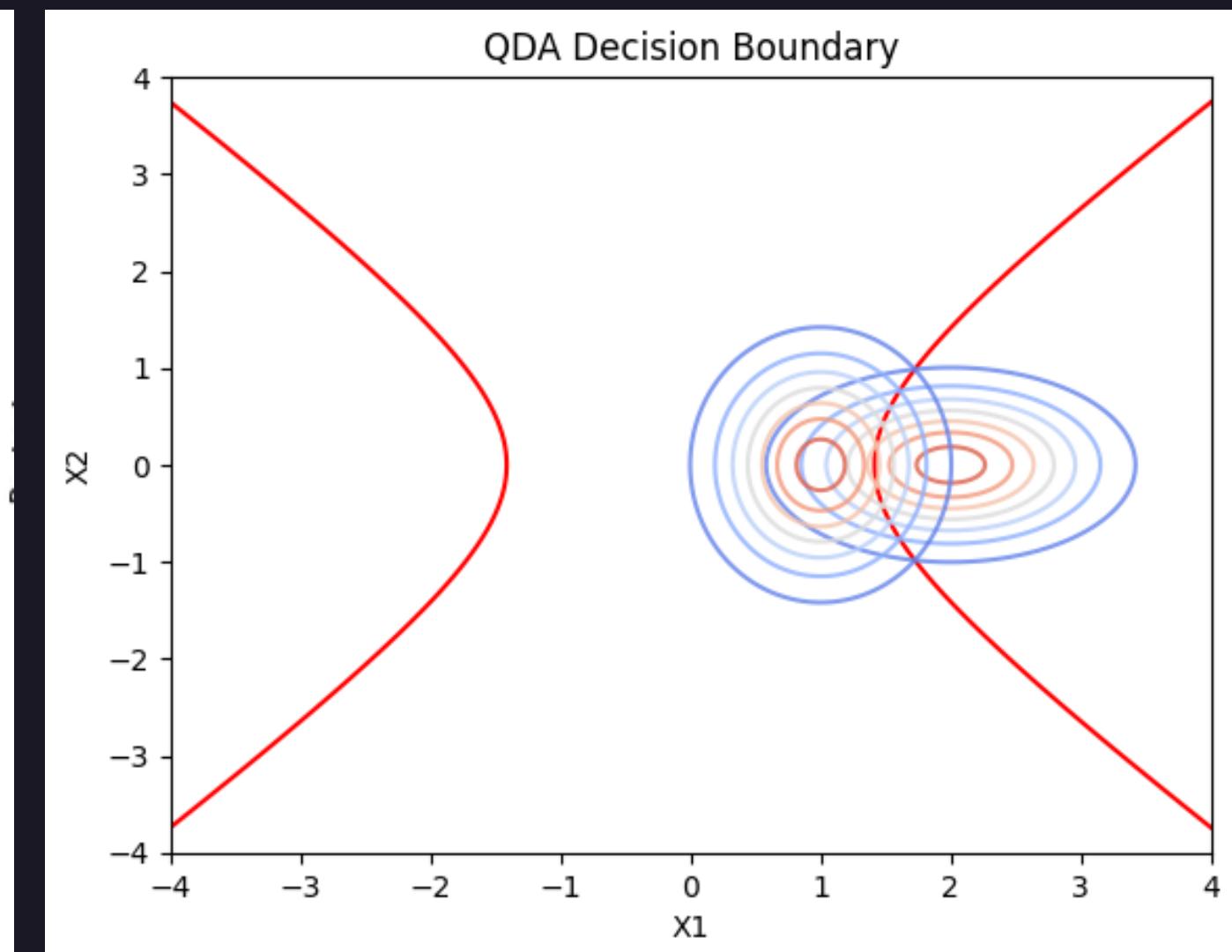
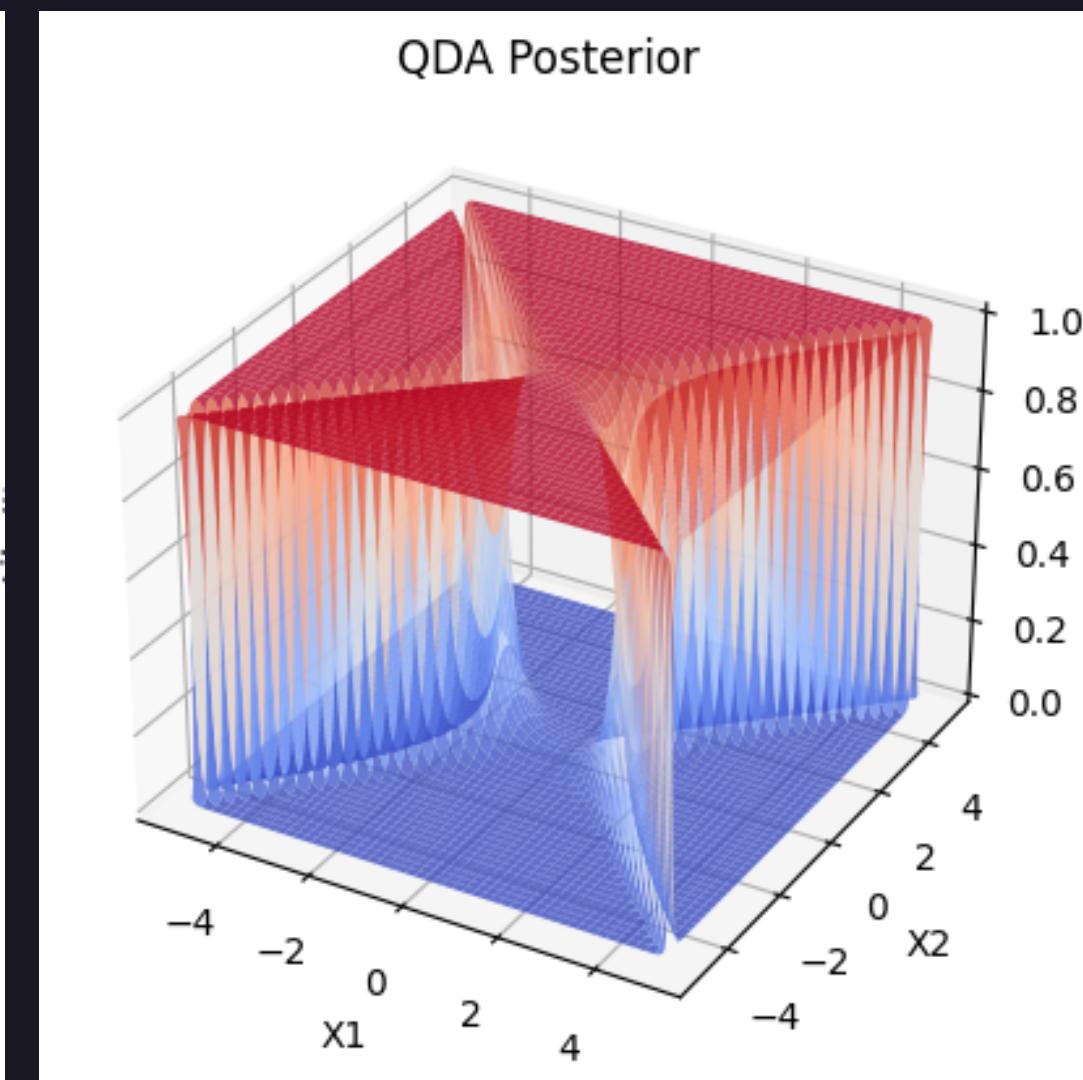
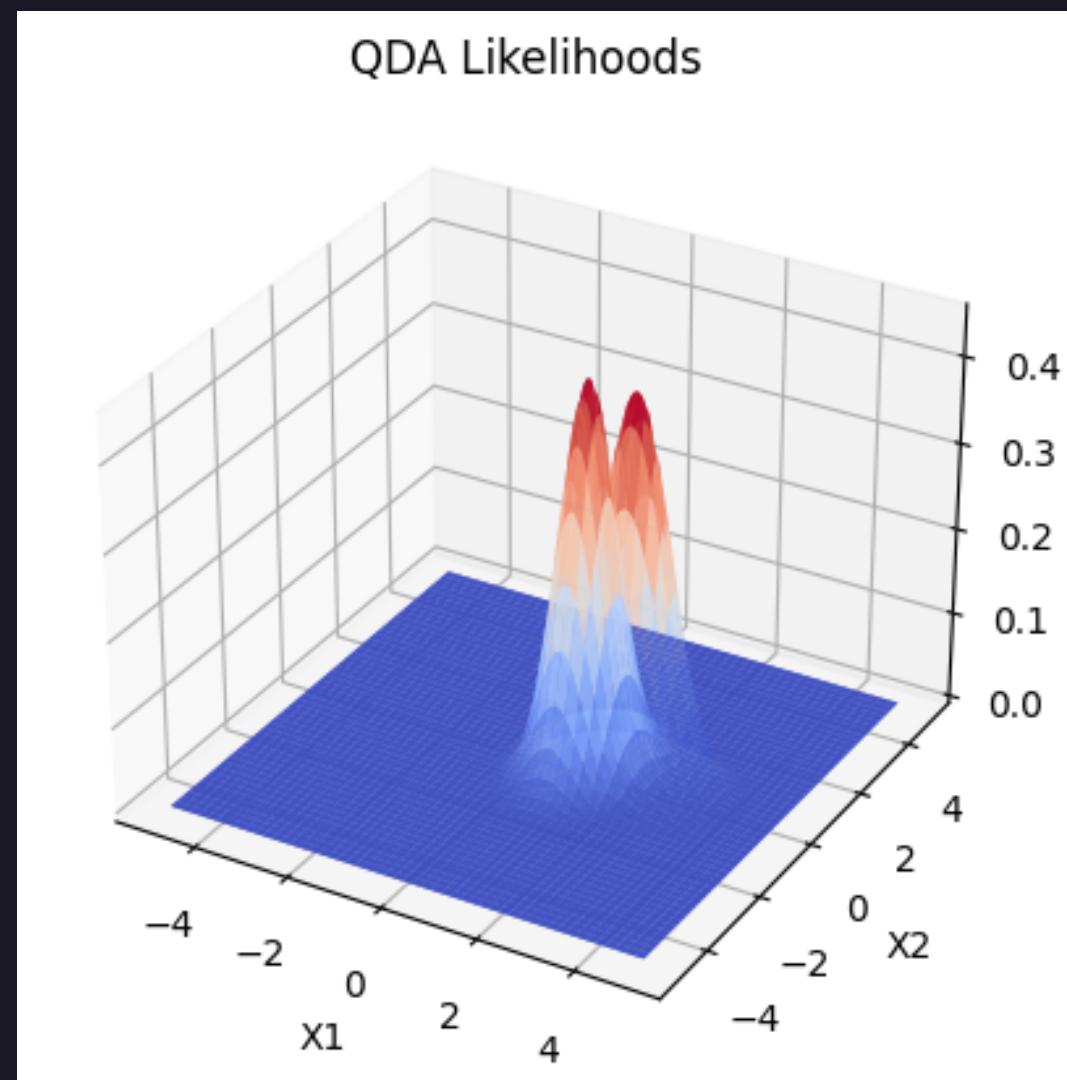
4. เขียนโปรแกรมสำหรับสร้างตัวจำแนกเชิงเส้น

ทั้ง 4 ข้อให้ วาดกราฟ likelihood, posterior และขอบตัดสินใจ โดยทำสองรูปแบบ คือ

- กำหนดค่าพารามิเตอร์ของการแยกแยะ
- สุ่มตัวอย่างเพื่อนำมาคำนวณค่าพารามิเตอร์ของการแยกแยะ

$$p(c_1) = p(c_2) \quad \mu_1 = [2 \ 0]^T \quad \mu_2 = [1 \ 0]^T \quad \Sigma_1 = [[0.5 \ 0], [0 \ 0.25]] \quad \Sigma_2 = [[0.25 \ 0], [0 \ 0.5]]$$

covariance ไม่เท่ากัน



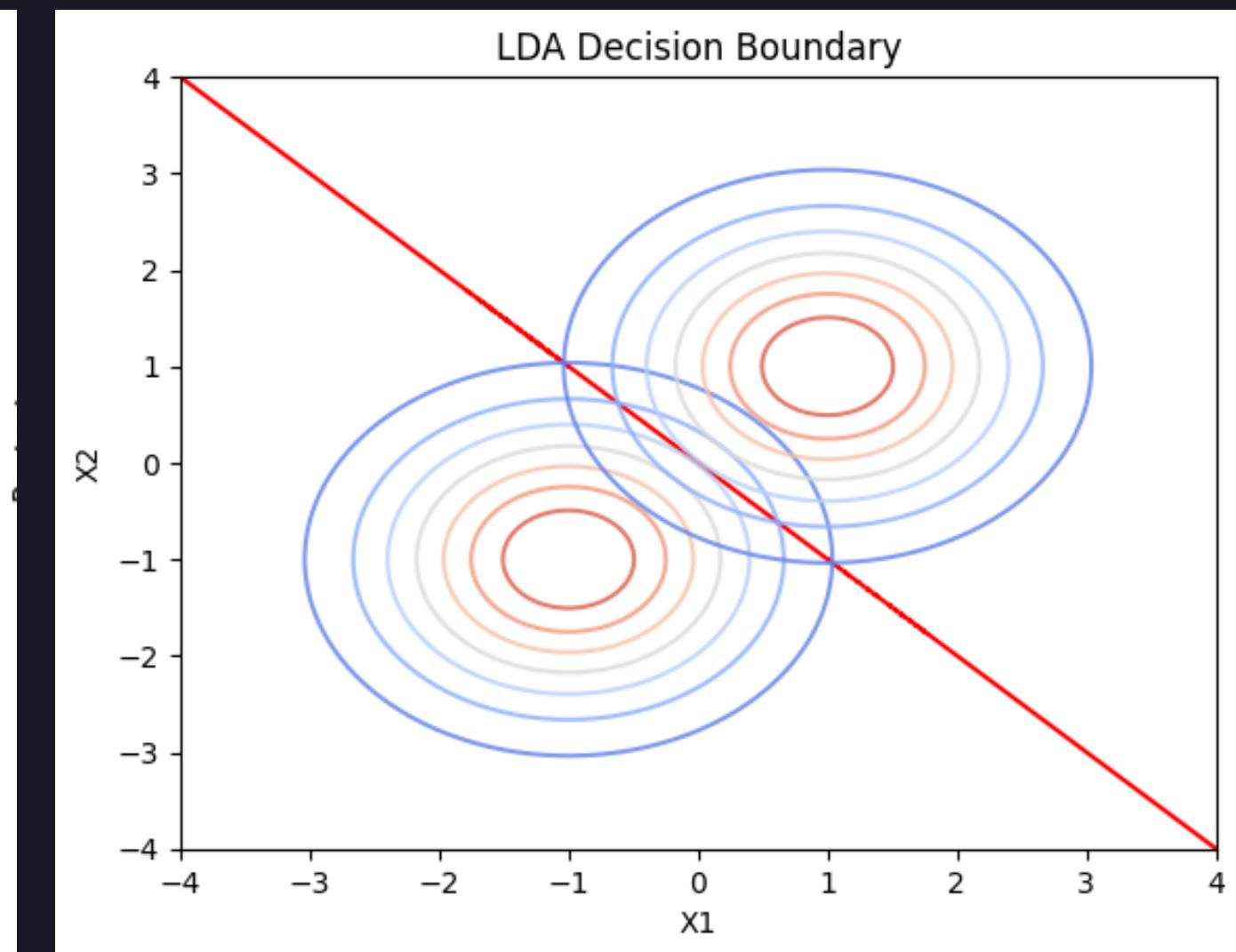
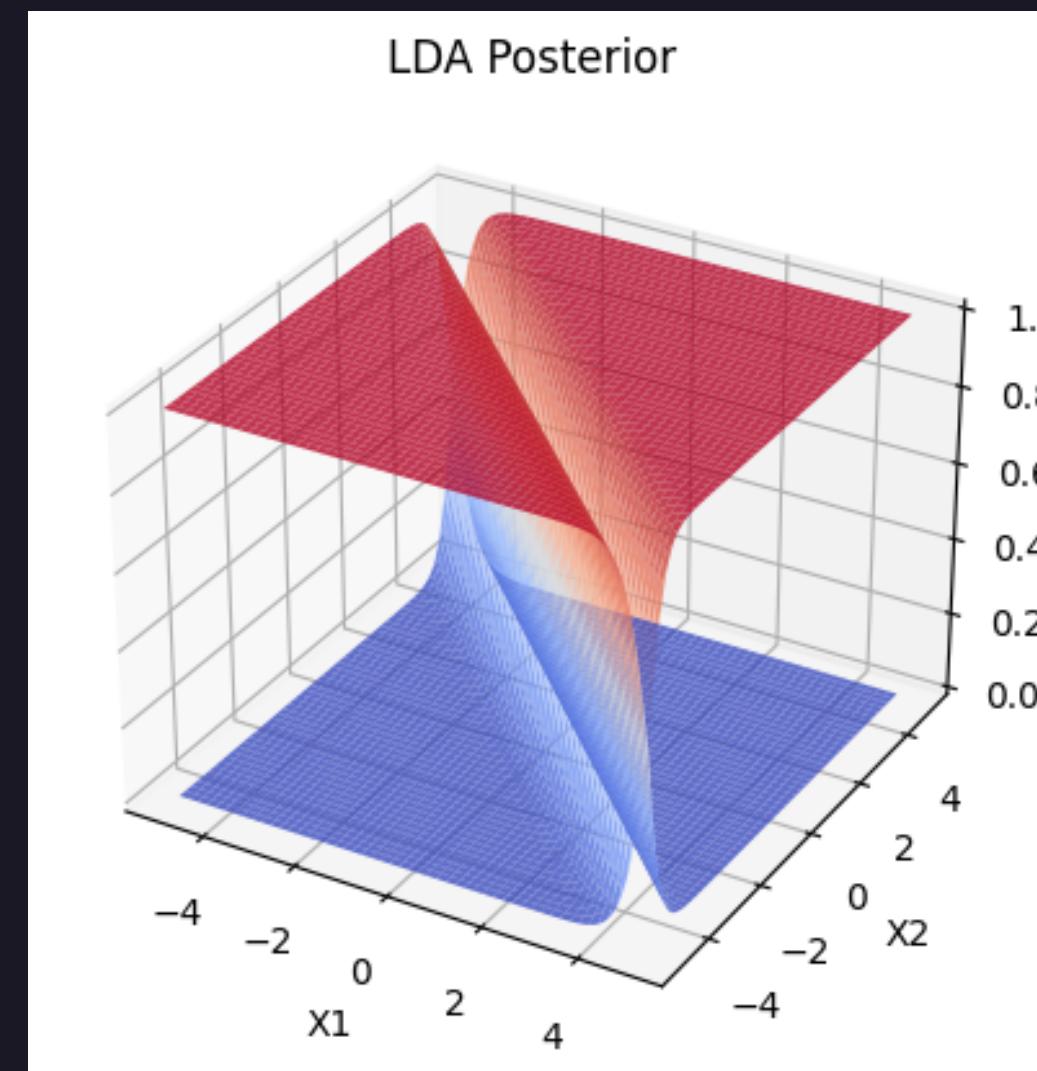
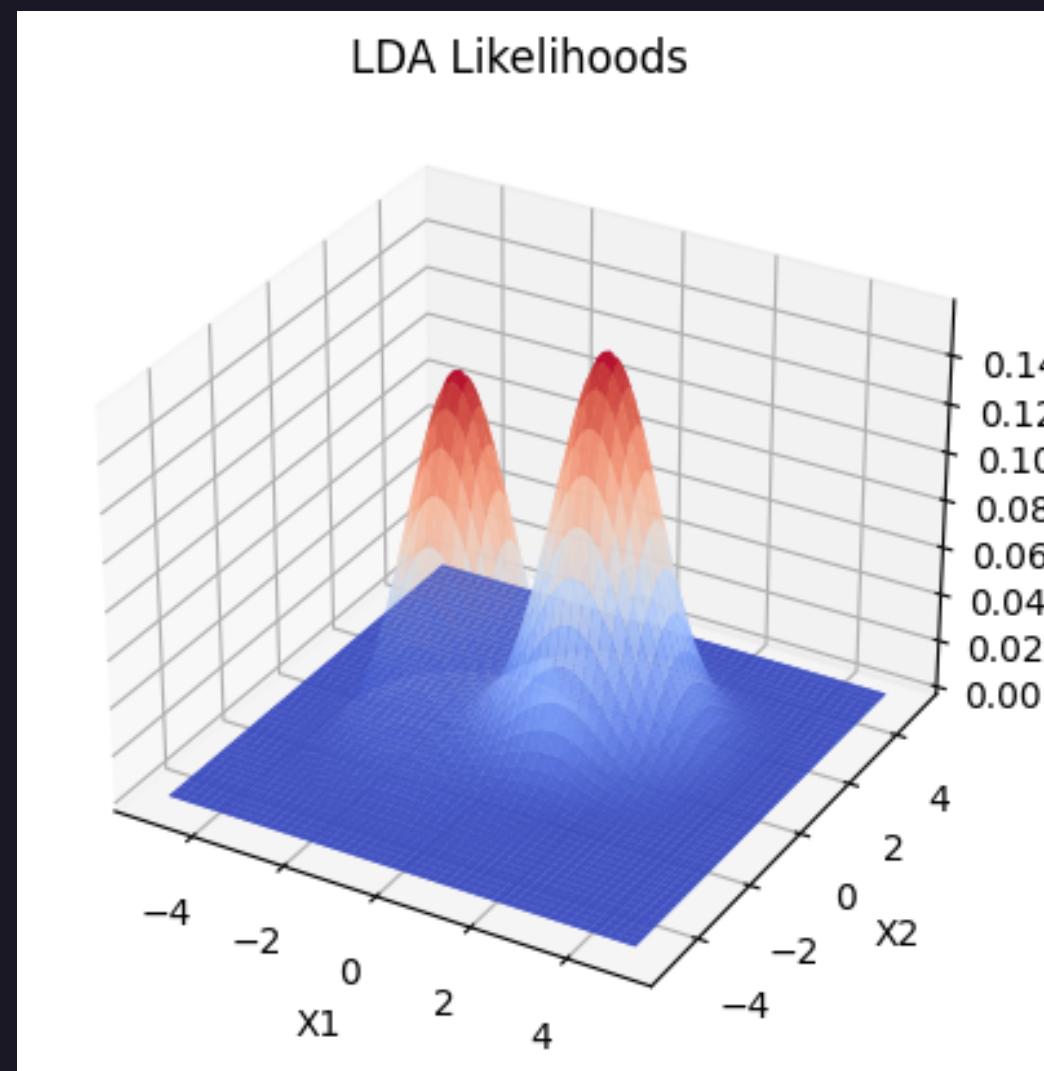
4. เขียนโปรแกรมสำหรับสร้างตัวจำแนกเชิงเส้น

ทั้ง 4 ข้อให้ วาดกราฟ likelihood, posterior และขอบตัดสินใจ โดยทำสองรูปแบบ คือ

- กำหนดค่าพารามิเตอร์ของการแยกแยะ
- สุ่มตัวอย่างเพื่อนำมาคำนวณค่าพารามิเตอร์ของการแยกแยะ

$$p(c_1) = p(c_2) \quad \mu_1 = [-1 \ -1]^T \quad \mu_2 = [1 \ 1]^T \quad \Sigma_1 = [[1 \ 0], [0 \ 1]] \quad \Sigma_2 = [[1 \ 0], [0 \ 1]]$$

covariance เท่ากัน



Generate Data

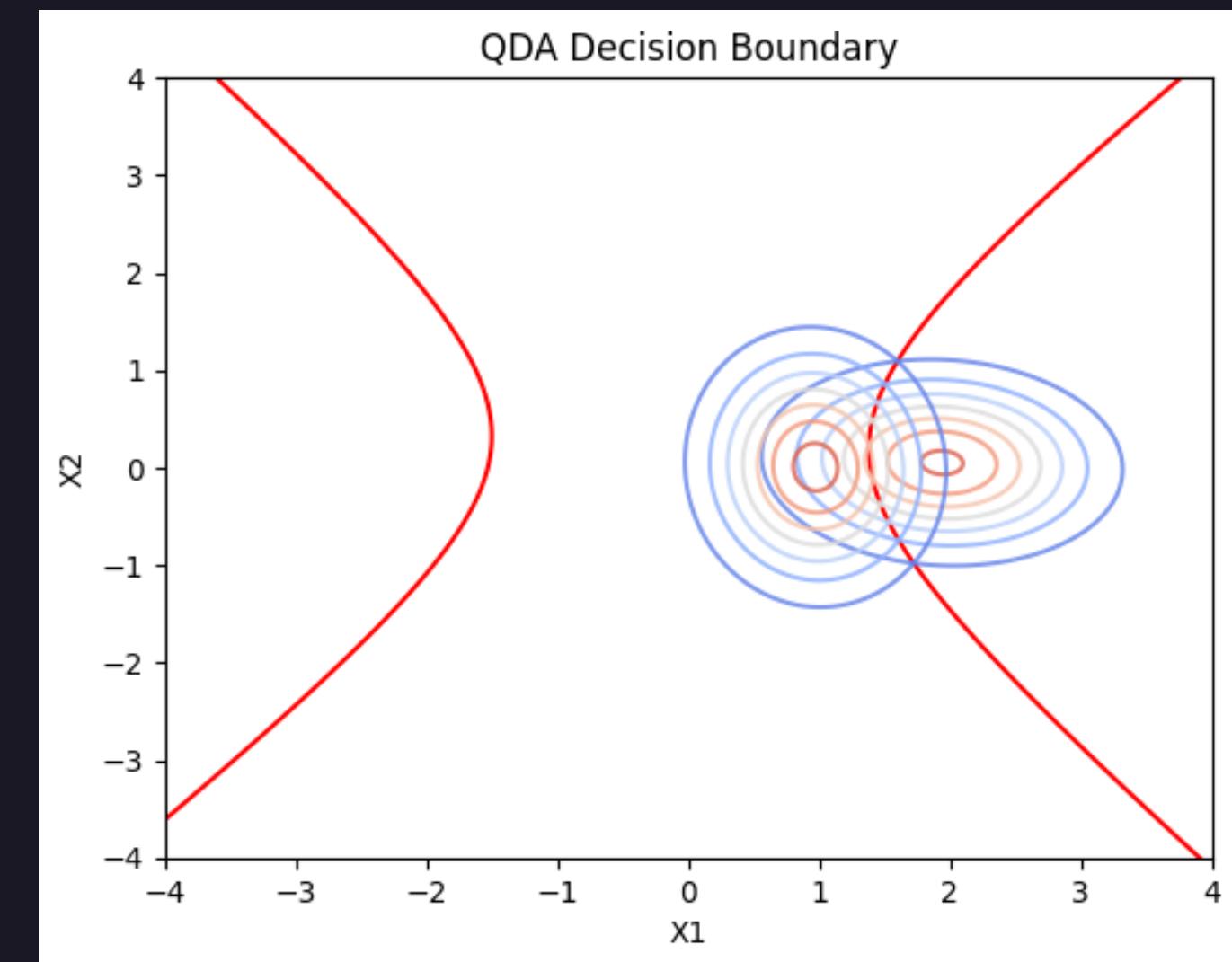
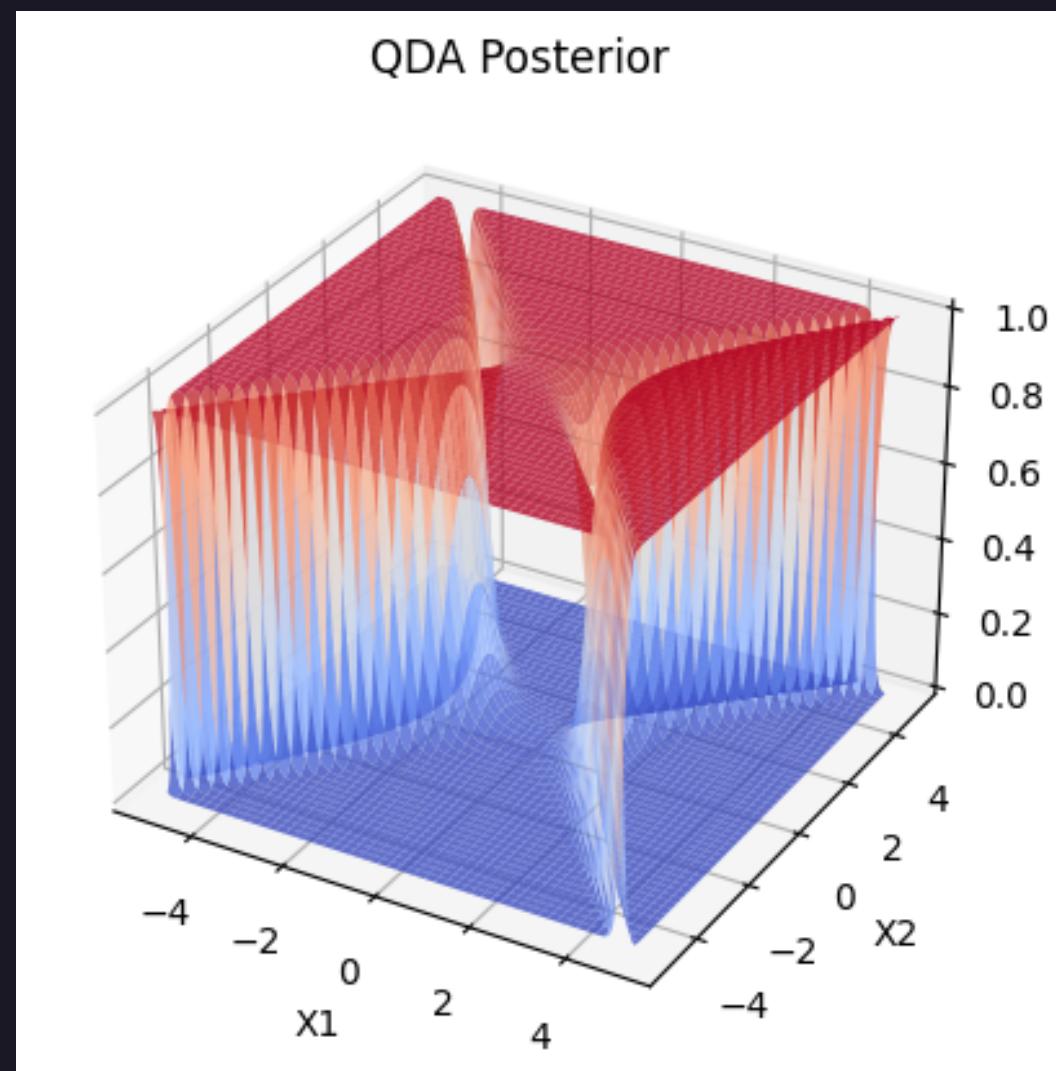
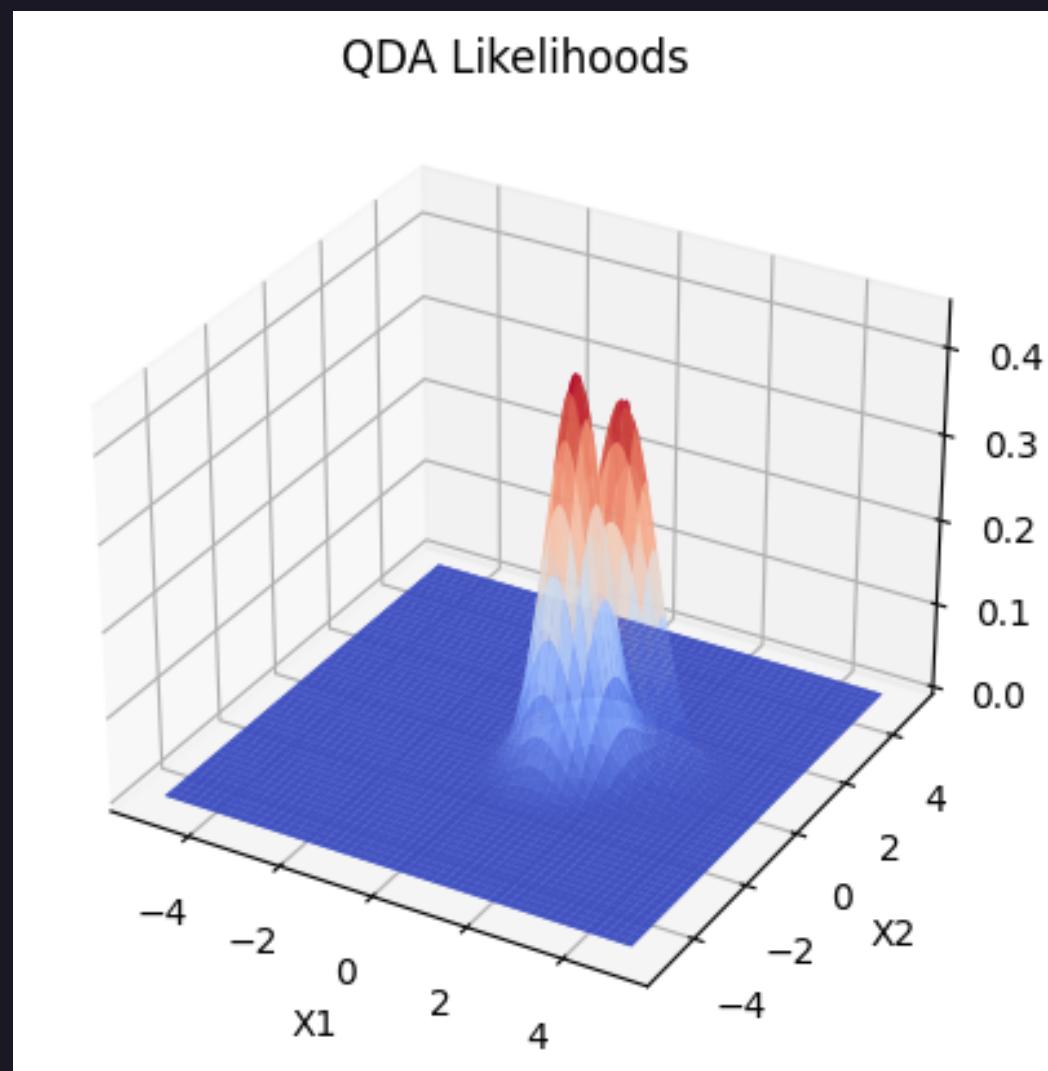
4. เขียนโปรแกรมสำหรับสร้างตัวจำแนกเชิงเส้น

ทั้ง 4 ข้อให้ วาดกราฟ likelihood, posterior และขอบตัดสินใจ โดยทำสองรูปแบบ คือ

- กำหนดค่าพารามิเตอร์ของการแยกแยะ
- สุ่มตัวอย่างเพื่อนำมาคำนวณค่าพารามิเตอร์ของการแยกแยะ

$$p(c_1) = p(c_2) \quad \mu_1 = [1.939 \ 0.052]^T \quad \mu_2 = [0.972 \ 0.006]^T$$

$$\Sigma_1 = [[0.482 \ -0.024], [-0.024 \ 0.284]] \quad \Sigma_2 = [[0.249 \ -0.013], [-0.013 \ 0.515]]$$



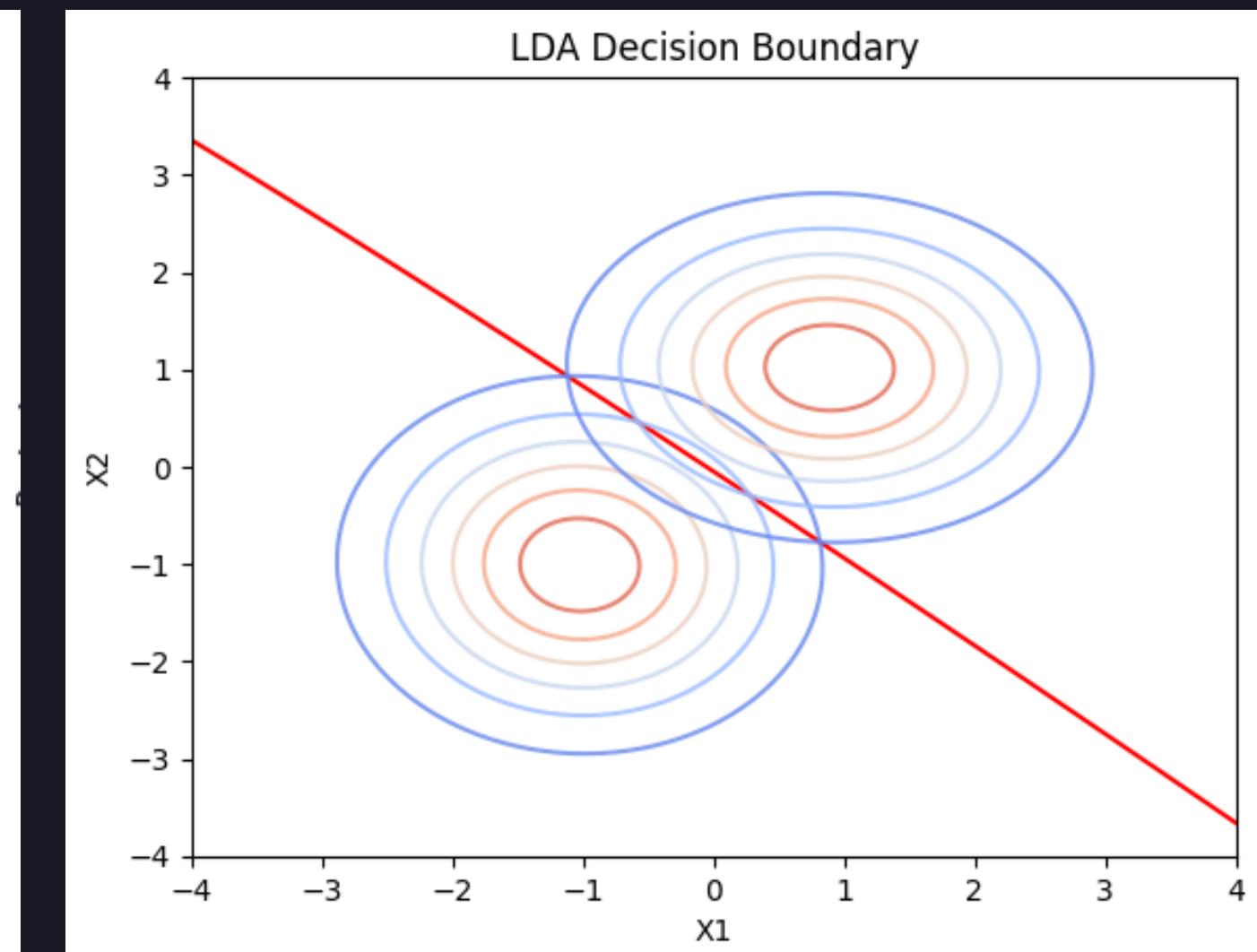
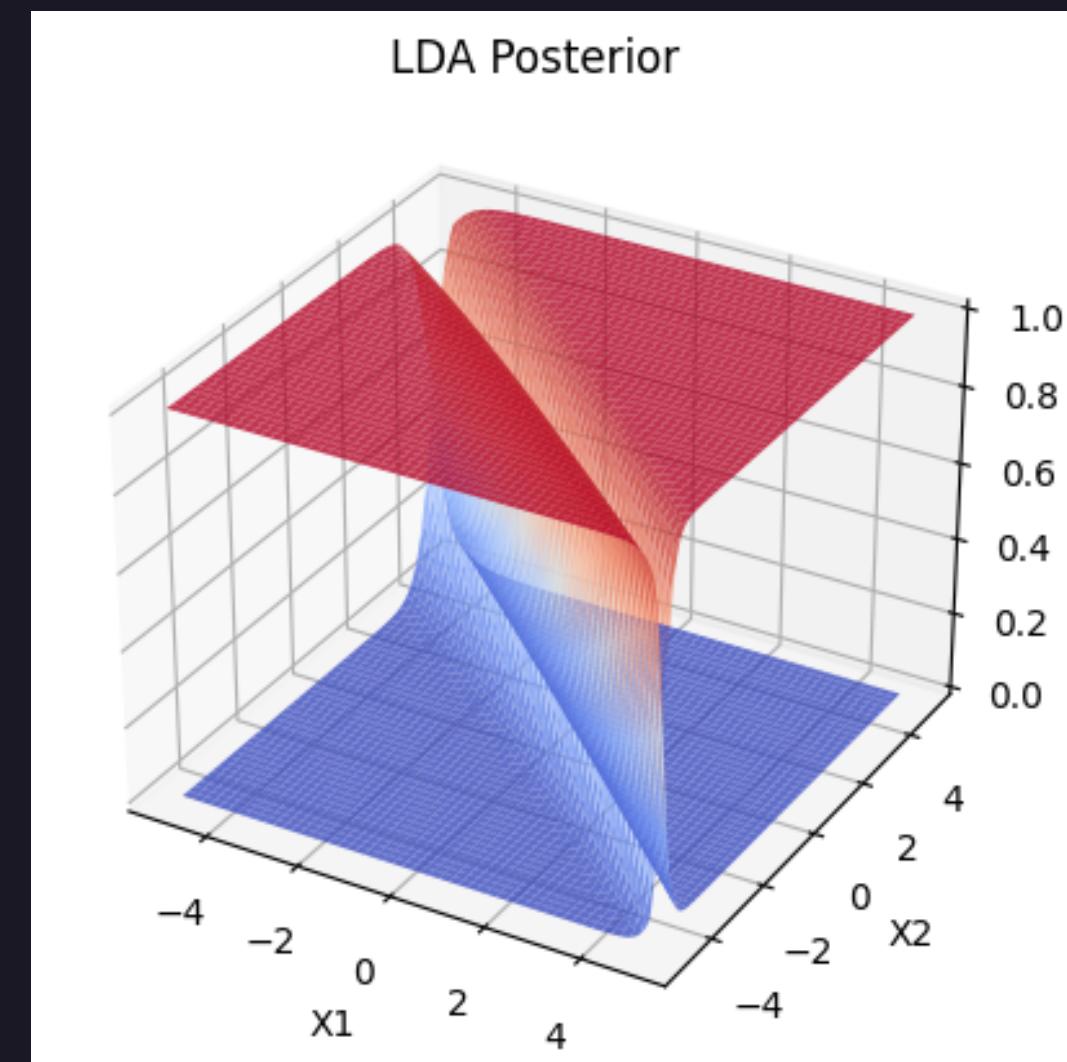
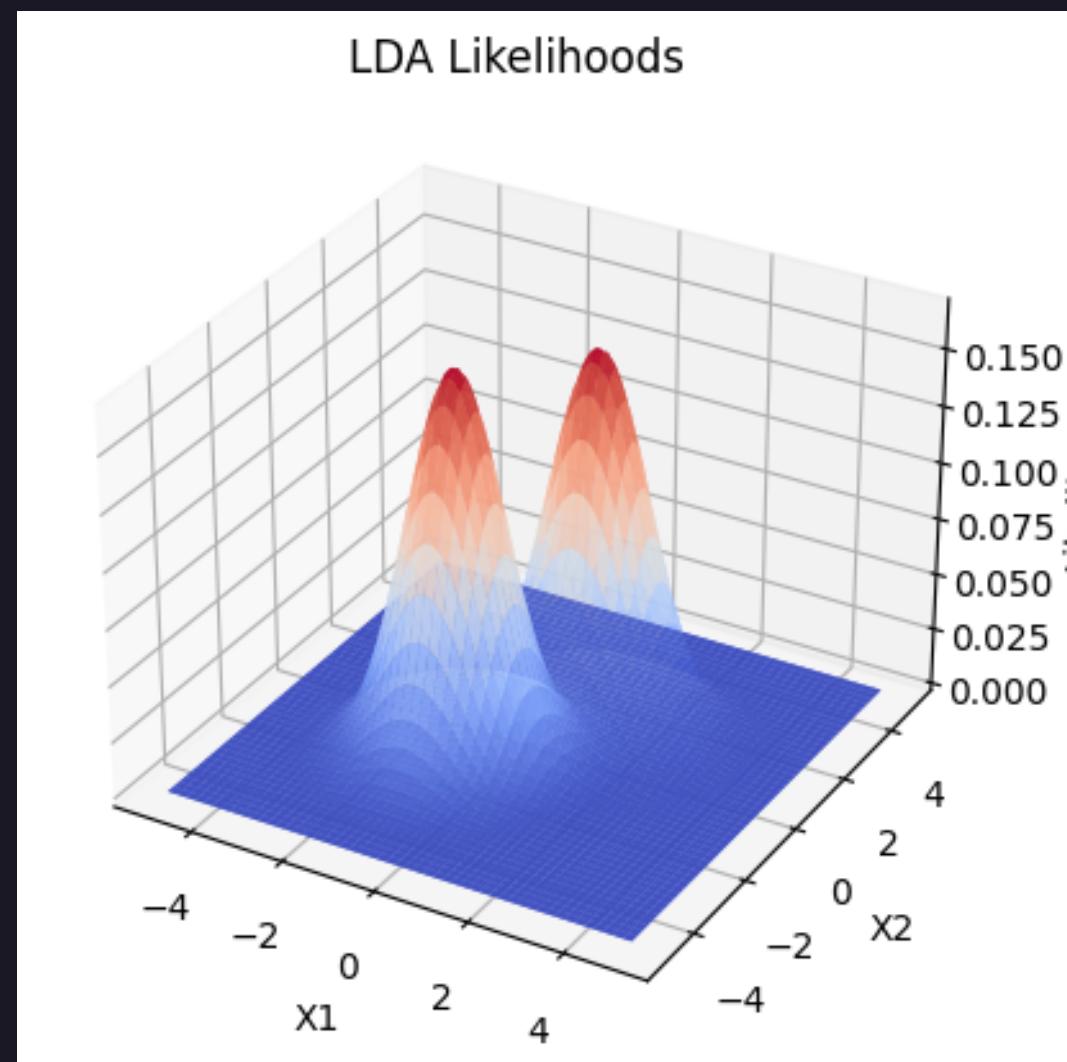
4. เขียนโปรแกรมสำหรับสร้างตัวจำแนกเชิงเส้น

ทั้ง 4 ข้อให้ วาดกราฟ likelihood, posterior และขอบตัดสินใจ โดยทำสองรูปแบบ คือ

- กำหนดค่าพารามิเตอร์ของการแยกแยะ
- สุ่มตัวอย่างเพื่อนำมาคำนวณค่าพารามิเตอร์ของการแยกแยะ

$$p(c_1) = p(c_2) \quad \mu_1 = [-1.026 \ -1.008]^T \quad \mu_2 = [0.889 \ 1.017]^T \quad T$$

$$\Sigma_1 = [[0.906 \ -0.019], [-0.019 \ 0.987]] \quad \Sigma_2 = [[1.061 \ -0.019], [-0.019 \ 0.844]]$$



5. เขียนโปรแกรมสำหรับ plot decision boundary เพื่อ 비교เทียบระหว่าง LDA, QDA และ Logistic regression โดยอาจจะมีการเพิ่มพจน์ second order polynomial

