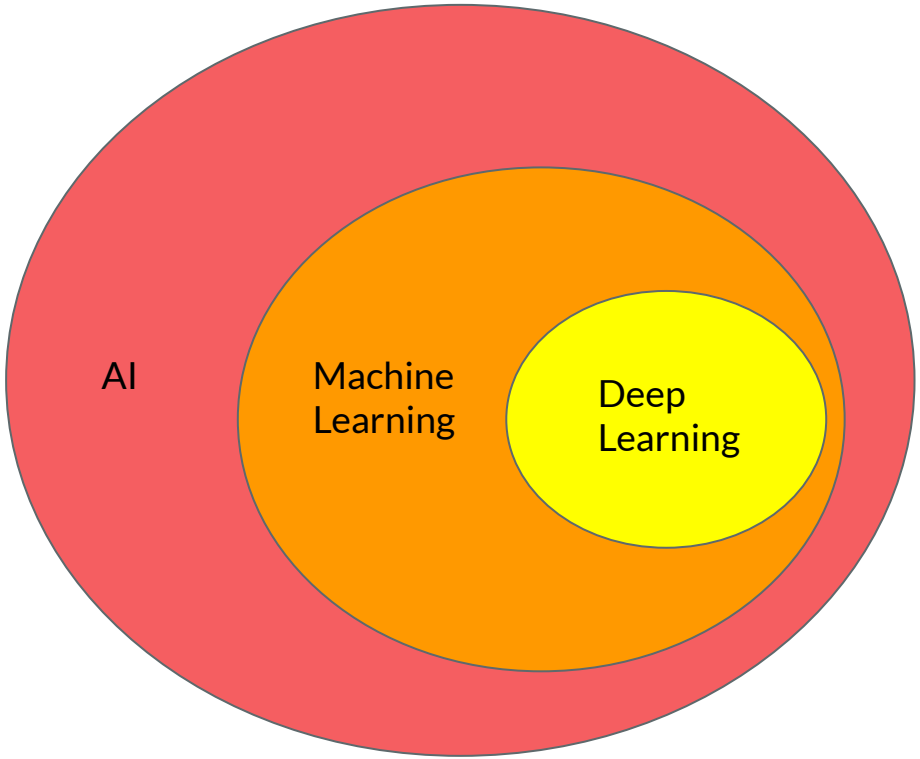


Generative Model
- Chat GPT
- Deepdreamgenerator

Machine Learning
- Decision tree
- Naive Bayes
- K nearest neighbor
- Support Vector Machine
- Genetic Algorithm

Deep Learning
- Convolutional Neural Network
- Recurrent Neural Network

Machine Learning

- Supervised Learning (Dataset => สร้างโมเดล => ทำนาย)
  - Regression
  - Classification
- Unsupervised Learning
  - Clustering
  - Dimensionality Reduction
  - Association
- Reinforcement Learning
  - No need for dataset
  - Agent + Environment => AlphaGo

Supervised Learning - Regression

Imagine you want to build a model to predict the selling prices of houses based on various features of those houses. You have a dataset that includes information about different houses and their selling prices. Your task is to create a regression model that can predict the selling price of a house based on its features.

Square footage: The size of the house in square feet.
Number of bedrooms: The count of bedrooms in the house.
Number of bathrooms: The count of bathrooms in the house.

Target : Price

SQ	Bedroom	Bathroom	Price
200	4	2	2M
300	5	2	3M
200	3	3	2.5M
400	5	4	?

Supervised Learning - Classification

Imagine you want to build a model to automatically classify emails as either "spam" or "not spam" (also known as "ham"). You have a dataset of emails, each labeled as spam or not spam, along with the content and metadata of those emails. Your task is to create a classification model that can automatically categorize new emails as spam or not spam.

Attachments: Information about any attachments.
Number of links: The number of hyperlinks in the email.
Word count: The total number of words in the email.

Target : Spam or not spam

Attach	links	words	spam/ not spam
0	2	200	not spam
1	5	202	spam
2	3	305	spam
2	3	400	?

Unsupervised Learning

Reinforcement Learning

$19^{*}19 = 361$

$3^{361} = 1.740897e+172$

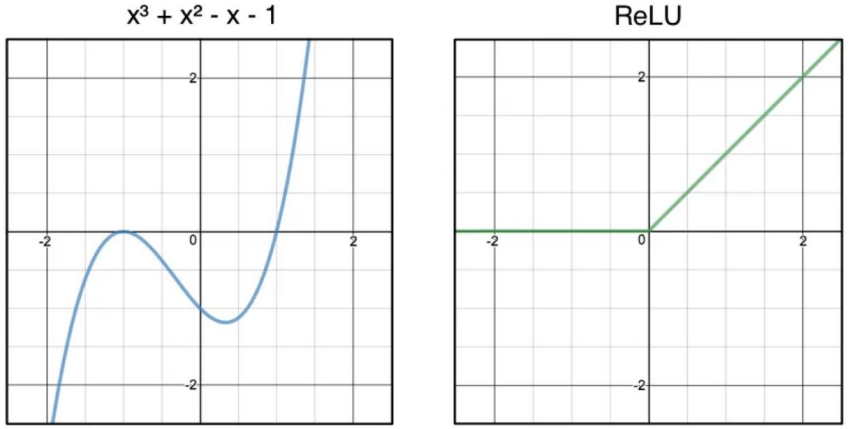
status = ดำ ขาว ว่าง

- Agent = ผู้เล่น
- Environment = บอร์ดของเกม Go และสถานการณ์ที่เกิดขึ้นในเกม
- Action = การเดินหมาก
- Reward = ผลลัพธ์จาก action
    - มากที่สุดตอนนี้ไม่ใช่ดีที่สุด ต้องวางแผนล่วงหน้ายาวๆ google ใช้ DL ในการคำนวน reward

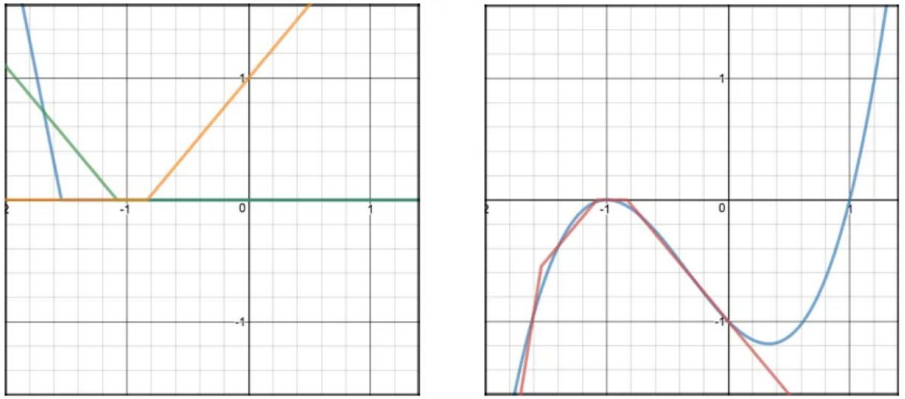
Deep learning

```
def relu(x):
    if x > 0:
        return x
    else:
        return 0
```

- ใช้ได้ดีกับข้อมูลที่ไม่สามารถตีความได้โดยง่าย เช่น ภาพ เสียง ภาษา
- ตรวจจับวัตถุ, self driving car, ChatGPT



Left: target function **Right:** raw materials



Left: 3 ReLU functions **Right**: Weighted sum of the 3 ReLU functions

https://towardsdatascience.com/can-neural-networks-really-learn-any-function-65e106617fc6

Linear Regression (Supervised Learning)

- ลักษณะจะเป็นการนำตัวแปรต้น 1 ตัว (x) มาพยากรณ์ตัวแปรตาม 1 ตัว (y)

Data =

X	Y
x1	y1
x2	y2
x3	
xn	yn

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}$$

เมื่อ N คือ จำนวนตัวอย่างทั้งหมด

Linear Regression : Data

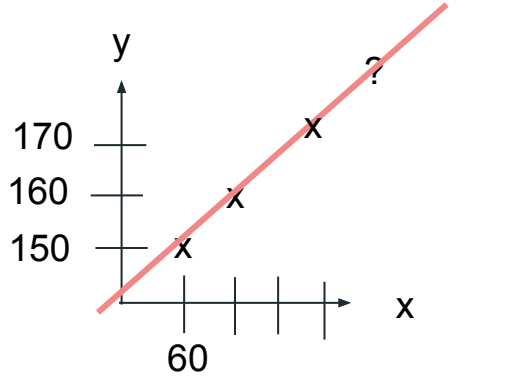
น้ำหนัก (x)	ส่วนสูง (y)
60	150
65	160
70	175

$$X = \begin{bmatrix} 60 \\ 65 \\ 70 \end{bmatrix}$$

$$Y = \begin{bmatrix} 150 \\ 160 \\ 175 \end{bmatrix}$$

สมการเส้นตรง (Linear equation)

$\hat{y} = ax + b$



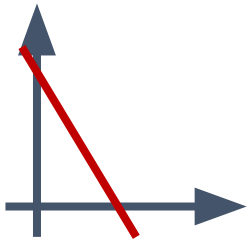
Linear Regression : Model => Slope

ความชัน $\boxed{a = \frac{y_2 - y_1}{x_2 - x_1}}$

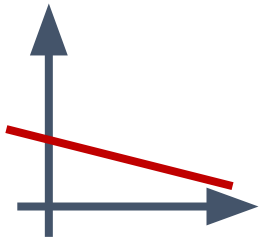
$$\boxed{\hat{y} = ax + b}$$

a คือ ความชัน และ b คือจุดตัดแกน Y

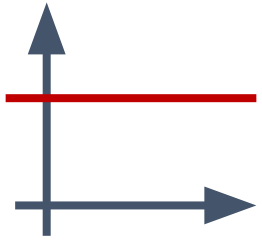
$a < -1$



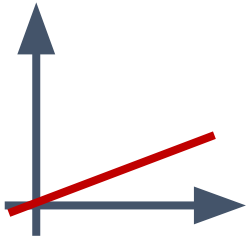
$-1 < a < 0$



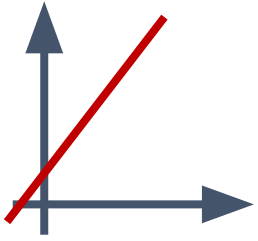
$a = 0$



$0 < a < 1$



$a > 1$

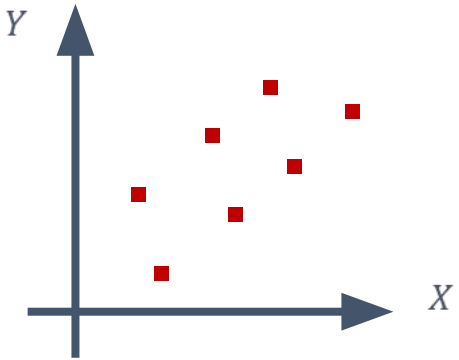


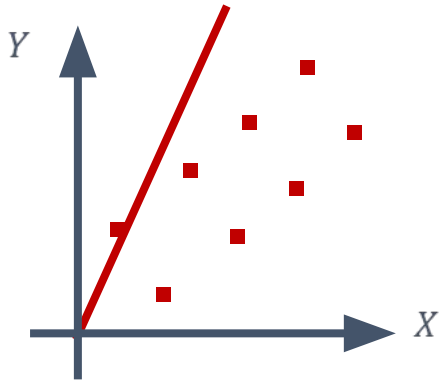
model คือ $\hat{y} = ax + b$

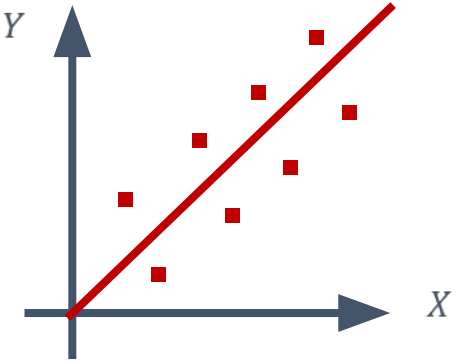
$$\hat{Y} = \begin{bmatrix} \widehat{y_1} \\ \widehat{y_2} \\ \widehat{y_3} \\ \vdots \\ \widehat{y_N} \end{bmatrix} = \begin{bmatrix} ax_1 + b \\ ax_2 + b \\ ax_3 + b \\ \vdots \\ ax_N + b \end{bmatrix} = a \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} + \mathrm{b}$$

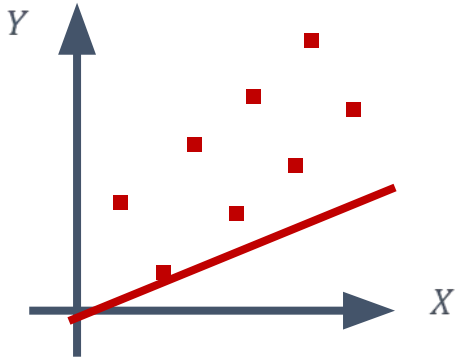
$$\hat{Y} = aX + b$$

น้ำหนัก (x)	ส่วนสูง (y)	
60	150	?
65	160	?
70	175	?

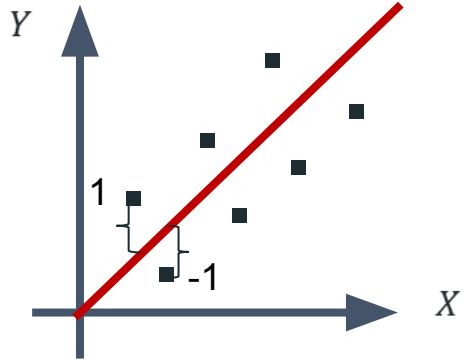

Y
X


Y
X


Y
X


Y
X

Linear Regression : Model หาค่า a กับ b



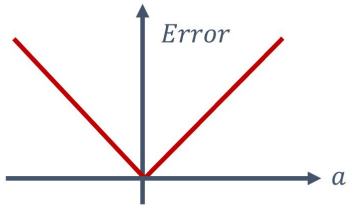
เลือกกราฟนี้เพราะ error น้อยที่สุด

$$\hat{Y} = aX + b$$

ต้องการ a และ b ที่ทำให้เกิด error น้อยที่สุด วัด error ยังไงดี ?

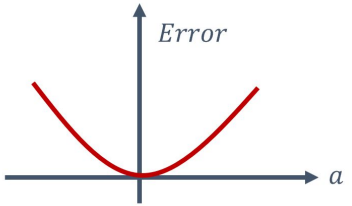
$$\boldsymbol{Error = \sum_{i=1}^{N} (y_i - \hat{y}_i)}$$

$$\boldsymbol{Error = \sum_{i=1}^{N} |y_i - \hat{y}_i|}$$



Sum of squared errors (SSE)

$$\boldsymbol{Error = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$



$$Error = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

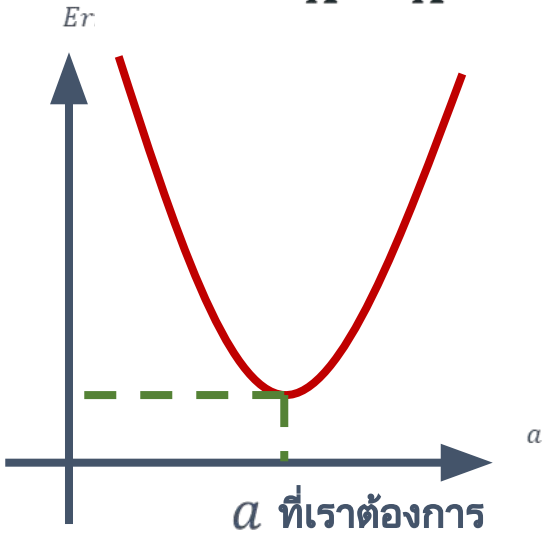
ต้องการหา a กับ b ที่ดีที่สุด คือจุดที่ค่าความชันเป็น 0
ความชันการได้จากการ diff

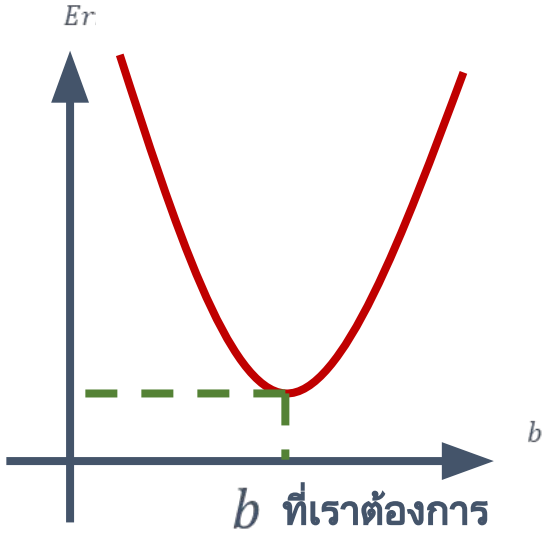
1. $\frac{\partial Error}{\partial a} = 0$

2. $\frac{\partial Error}{\partial b} = 0$

$$a = \frac{\overline{XY} - \bar{X}\bar{Y}}{\overline{X^2} - \bar{X}^2}$$

$$b = \frac{\overline{X^2}\bar{Y} - \overline{XY}\,\bar{X}}{\overline{X^2} - \bar{X}^2}$$





	X	Y
1	2	4
2	6	12
3	10	20
4	8	16
5	7	14

N = 5

$$a = \frac{\overline{XY} - \bar{X}\bar{Y}}{\overline{X^2} - \bar{X}^2}$$

$$\mathbf{b} = \frac{\overline{X^2}\bar{Y} - \overline{XY}\,\bar{X}}{\overline{X^2} - \bar{X}^2}$$

Linear Regression (Supervised Learning)

- Data มีตัวแปรต้น 1 ตัว ตัวแปรตาม 1 ตัว
  - ลอง plot data เพื่อดูว่า x กับ y มีความสัมพันธ์กัน (Correlation) หรือไม่ (> 0.7 หรือ < -0.7 คือสัมพันธ์กัน)

  หรือ
  - ถ้าอยู่ในช่วง -0.7 ถึง 0.7 ให้ลอง ใส่ log แล้ว plot กราฟดู correlation (มันอาจจะมีแนวโน้มเป็น expo, polynomial อยู่ ถ้า ใส่ log จะกลายเป็นเส้นตรง)
- Correlation (สหสัมพันธ์)
  - บอกความเกี่ยวข้องของตัวแปรว่ามีความเกี่ยวข้องกันหรือไม่

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Linear Regression (Correlation)

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

$$\begin{bmatrix} (x_1 - \bar{x})(y_1 - \bar{y}) \\ (x_2 - \bar{x})(y_2 - \bar{y}) \\ (x_3 - \bar{x})(y_3 - \bar{y}) \\ \vdots \\ (x_N - \bar{x})(y_N - \bar{y}) \end{bmatrix} = (X - \bar{X})(Y - \bar{Y})$$

$$\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) = ((X - \bar{X}) * (Y - \bar{Y})).\mathbf{sum}()$$

Linear Regression (Correlation)

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

$$\begin{bmatrix} (x_1 - \bar{x})^2 \\ (x_2 - \bar{x})^2 \\ (x_3 - \bar{x})^2 \\ \vdots \\ (x_N - \bar{x})^2 \end{bmatrix} = \begin{bmatrix} (x_1 - \bar{x})(x_1 - \bar{x}) \\ (x_2 - \bar{x})(x_2 - \bar{x}) \\ (x_3 - \bar{x})(x_3 - \bar{x}) \\ \vdots \\ (x_N - \bar{x})(x_N - \bar{x}) \end{bmatrix} = (X - \bar{X})(X - \bar{X}) = (X - \bar{X})**2$$

$$\sum_{i=1}^{N}(x_i - \bar{x})^2 = ((X - \bar{X}) ** 2).sum()$$

Linear Regression (Correlation)

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

$$\begin{bmatrix} (y_1 - \bar{y})^2 \\ (y_2 - \bar{y})^2 \\ (y_3 - \bar{y})^2 \\ \vdots \\ (y_N - \bar{y})^2 \end{bmatrix} = \begin{bmatrix} (y_1 - \bar{y})(y_1 - \bar{y}) \\ (y_2 - \bar{y})(y_2 - \bar{y}) \\ (y_3 - \bar{y})(y_3 - \bar{y}) \\ \vdots \\ (y_N - \bar{y})(y_N - \bar{y}) \end{bmatrix} = (Y - \bar{Y})(Y - \bar{Y}) = (Y - \bar{Y})**2$$

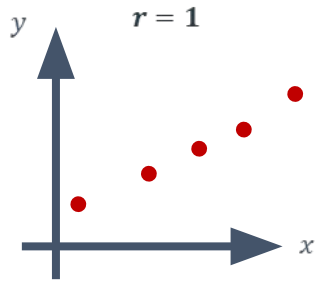
$$\sum_{i=1}^{N}(y - \bar{y})^2 = ((Y - \bar{Y}) ** 2).sum()$$

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

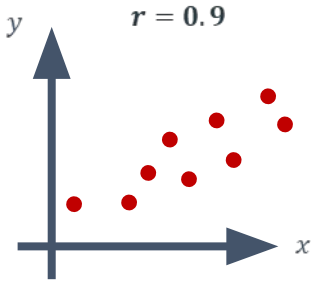
$$fraction = \big((X - \bar{X}) * (Y - \bar{Y})\big).sum()$$

$$denominator = np.sqrt(\big((X - \bar{X}) ** 2\big).sum() * ((Y - \bar{Y}) ** 2).sum())$$

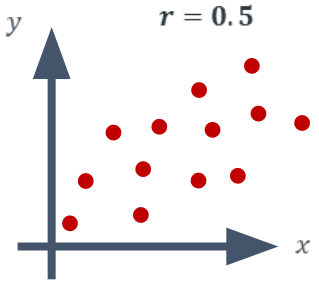
$$r = \frac{fraction}{denominator}$$



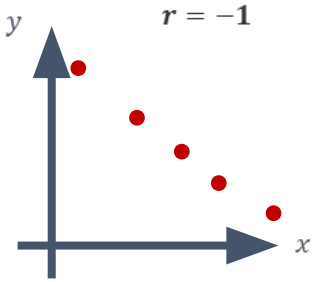
Perfect Positive correlation



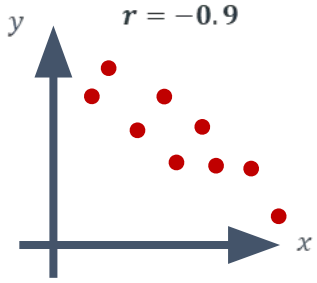
High Positive correlation



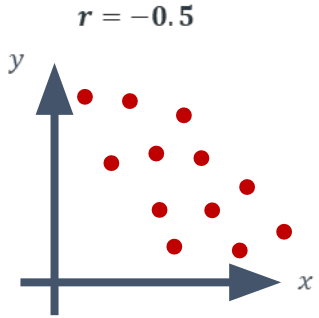
Low Positive correlation



Perfect Negative correlation



High Negative correlation



Low Negative correlation

	X	Y
1	2	4
2	6	12
3	10	20
4	8	16
5	7	14

N = 5

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Code

- Import : numpy, pandas, matplotlib
- Function
  - หา Correlation : input ที่ต้องการคือ X,Y
  - หาค่า a, b : input ที่ต้องการคือ X,Y
  - หา $\hat{Y}$: input ที่ต้องการคือ X, a, b
  - หา error : input ที่ต้องการคือ Y, $\hat{Y}$, Type of error
- Read data
- Create Model
- Predictions

Correlation : r

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

$$fraction = \big((X - \bar{X}) * (Y - \bar{Y})\big).sum()$$

$$denominator = np.sqrt(\big((X - \bar{X}) ** 2\big).sum() * ((Y - \bar{Y}) ** 2).sum())$$

```
def LR_find_r(X, Y):
    Xmean = X.mean()
    Ymean = Y.mean()
    fraction = ((X - Xmean)*(Y - Ymean)).sum()
    denominator = np.sqrt(((X - Xmean)**2).sum()*((Y - Ymean)**2).sum())
    r = fraction/denorminator
    return r
```

Find : a,b

$$a = \frac{\overline{XY} - \bar{X}\bar{Y}}{\overline{X^2} - \bar{X}^2}$$

$$\mathbf{b} = \frac{\overline{X^2}\bar{Y} - \overline{XY}\,\bar{X}}{\overline{X^2} - \bar{X}^2}$$

```
def LR_find_ab(X, Y):
    XY = X*Y
    X2 = X**2
    Xmean = X.mean()
    Ymean = Y.mean()
    XYmean = XY.mean()
    X2mean = X2.mean()
    denominator = X2mean - Xmean**2
    a = (XYmean - Xmean*Ymean)/denominator
    b = (X2mean*Ymean - Xmean*XYmean)/denominator
    return a, b
```

Find : $\hat{Y}$

$$\hat{y} = ax + b$$

```
def LR_find_Yhat(X, a, b):
    Yhat = a*X + b
    return Yhat
```

Find error

- SSE (Sum Squared Error) : ใช้หาค่า error ของ Y กับ $\hat{Y}$
- MSE (Mean Squared Error) :
- MAE (Mean Absolute Error)
- MAPE (Mean Absolute Percentage Error)

```
def find_error(Y, Yhat, TypeOfError):
    if TypeOfError == 'SSE':
        error = find_SSE(Y, Yhat)
    elif TypeOfError == 'MSE':
        error = find_MSE(Y, Yhat)
    elif TypeOfError == 'MAE':
        error = find_MAE(Y, Yhat)
    elif TypeOfError == 'MAPE':
        error = find_MAPE(Y, Yhat)
    return error
```

Find error : SSE

$$SSE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

```
def find_SSE(Y, Yhat):
    SSE = ((Y - Yhat)**2).sum()
    return SSE
```

	Y	Ŷ
1	4	4.1
2	10	10.2
3	12	11.8
4	8	8.1
5	6	5.8

SSE = ?

Find error : MSE

$$SSE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$MSE = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}$$

$$MSE = \frac{SSE}{N}$$

	Y	Ŷ
1	4	4.1
2	10	10.2
3	12	11.8
4	8	8.1
5	6	5.8

MSE = ?

```
def find_MSE(Y, Yhat):
    N = Y.shape[0]
    SSE = ((Y - Yhat)**2).sum()
    MSE = SSE/N
    return MSE
```

Find : MAE

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \widehat{y_i}|}{N}$$

```
def find_MAE(Y, Yhat):
    N = Y.shape[0]
    MAE = (np.abs(Y - Yhat)).sum()/N
    return MAE
```

	Y	Ŷ
1	4	4.1
2	10	10.2
3	12	11.8
4	8	8.1
5	6	5.8

MAE = ?

Find : MAPE

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y_i}}{y_i}\right|$$

	Y	Ŷ
1	4	4.1
2	10	10.2
3	12	11.8
4	8	8.1
5	6	5.8

MAPE = ?

```
def find_MAPE(Y, Yhat):
    N = Y.shape[0]
    MAPE = np.abs((Y - Yhat)/Y).sum()*100/N
    return MAPE
```

Read Data

Data =

X	Y
x_1	y_1
x_2	y_2
x_3	y_3
$\vdots$	$\vdots$
x_N	y_N

```
Data = pd.read_excel('Data.xlsx')
Or
Data = pd.read_csv('Data.csv')

DataMatrix = Data.as_matrix() #- 3.5
Or
DataMatrix = Data.values #python 3.6 up
```

$$DataMatrix = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ & \vdots \end{bmatrix}$$

Read Data

X = DataMatrix[:, 0]

Y = DataMatrix[:, 1]

$$DataMatrix = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ & \vdots \\ x_N & y_N \end{bmatrix} \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}$$

Read Data

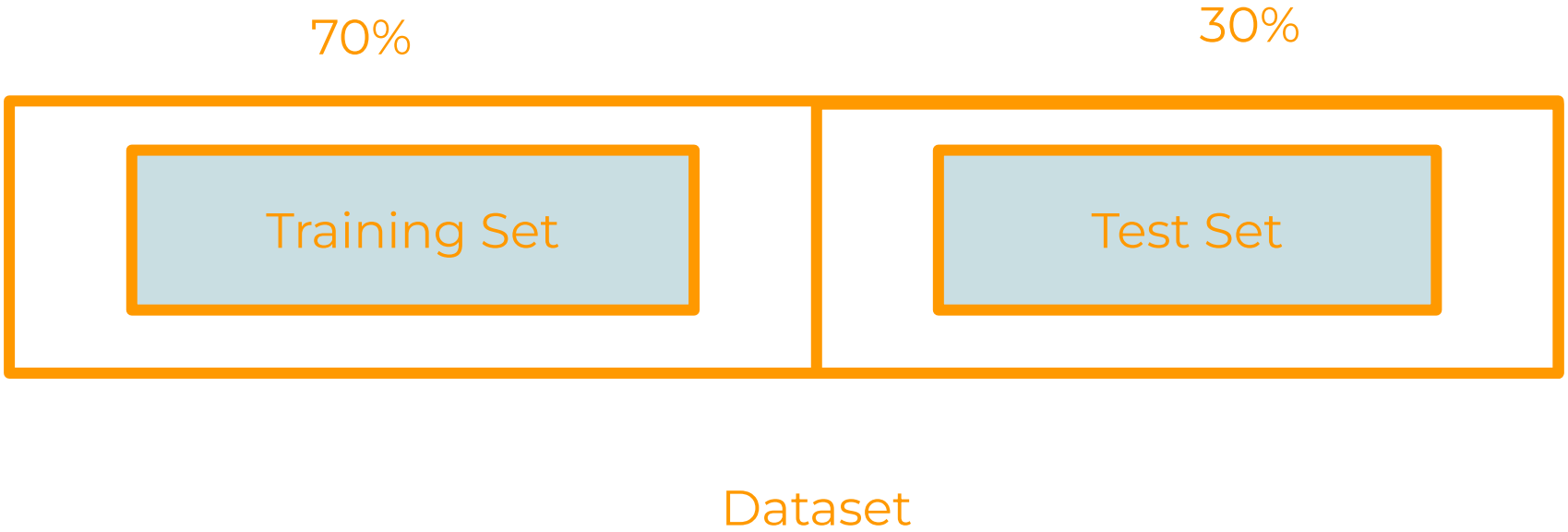
```
Data = pd.read_csv('GrowthOfBacteria.csv’)
DataMatrix = Data.values
N = DataMatrix.shape[0]  #N = 46
X = DataMatrix[:, :1]
Y = DataMatrix[:, 1:]
plt.scatter(X, Y)
```

Day	#Bacteria
1	2.791933131
1.2	2.534075242
1.4	3.80259718
1.6	2.096585234

```
X_Train = X[:35]
Y_Train = Y[:35]
```

```
X_Test = X[35:]
Y_Test = Y[35:]
```

Create Model



Create Model

```
a, b = LR_find_ab(X_Train, Y_Train)

Yhat_Train = LR_find_Yhat(X_Train, a, b)
error_Train = find_error(Y_Train, Yhat_Train, 'MAPE')

plt.plot(Y_Train, label = 'Real Train')
plt.plot(Yhat_Train, label = 'Predicted Train')
plt.legend()

error_Train

plt.scatter(X_Train, Y_Train)
plt.plot(X_Train, Yhat_Train, 'r')
```

Make Prediction

```
Yhat_Test = LR_find_Yhat(X_Test, a, b)
error_Test = find_error(Y_Test, Yhat_Test, 'SSE')
plt.plot(Y_Test, label = 'Real Test')
plt.plot(Yhat_Test, label = 'Predicted Test')
plt.legend()

error_Test
Y_Test = np.exp(Y_Test)
Yhat_Test = np.exp(Yhat_Test)

plt.plot(Y_Test, label = 'Real Test')
plt.plot(Yhat_Test, label = 'Predicted Test')

plt.legend()
plt.scatter(X_Test, Y_Test)
plt.plot(X_Test, Yhat_Test, 'r')
```