TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

MULTIPLE REGRESSION

DEEP LEARNING 101

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

facebook/tautologyai
www.tautology.live

KRIN CHINPRASATSAK

DATASCIENTIST
INSTRUCTOR
SPEAKER

Experienced data scientist, instructor and speaker,
AI consultant for the public and private sectors,
CEO & Co-founder of MADEBYAI and QuantMetric

DL101 : Multiple Regression

Lecture

เรียนอย่างเห็นภาพ

Workshop

Workshop

All Topics

TAUTOLOGY
INNOVATION
SCHOOL
AI
OVERVIEW
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
INTRODUCTION

TAUTOLOGY
INNOVATION
SCHOOL
MODEL
CREATION
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
BASIC
WORKSHOP
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
MODEL
IMPROVEMENT
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
ADVANCED
WORKSHOP
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
DATA
PREPARATION
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
MODEL
EVALUATION
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
CROSS
VALIDATION
BY TAUTOLOGY

TAUTOLOGY
INNOVATION
SCHOOL
BASIC MATH FOR
DEEP LEARNING
BY TAUTOLOGY

DL101 : Multiple Regression

Course Journey

Facebook Group

TAUTOLOGY Deep Learning

https://www.facebook.com/groups/2723268457980889

Document for DL101

https://github.com/TAUTOLOGY-EDUCATION/DL101

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

AI OVERVIEW

AI OVERVIEW

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

AI Overview

- AI and Machine Learning
- Supervised Learning

AI and Machine Learning

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

History of AI

Enigma Code

History of AI

The Bombe : Enigma code-breaking machine

History of AI

Turing test

What is AI?

"AI is a computer system able to perform tasks that ordinarily require human intelligence... Many of these artificial intelligence systems are **powered by machine learning**, some of them are **powered by deep learning** and some of them are **powered by very boring things like rules**."

Jeremy Achin, DataRobot CEO, speech at the Japan AI Experience in 2017

AI, ML and DL

Type of Machine Learning

1. Supervised Learning Algorithms
2. Unsupervised Learning Algorithms
3. Reinforcement Learning Algorithms

Type of Machine Learning

Supervised Learning

Classification

Regression

Unsupervised Learning

Clustering

Dimensionality Reduction

Association

Reinforcement Learning

Type of Machine Learning

AI Overview

- **AI and Machine Learning**
- Supervised Learning

Supervised Learning

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Concept of Supervised Learning

Concept of Supervised Learning

Data ⇨ Model ⇨ Prediction

Regression and Classification

Classification

น้ำหนัก (kg)	ความดัน (mmHg)	เป็นโรคเบาหวาน
65	130	ไม่เป็น
42	142	ไม่เป็น
56	171	เป็น
71	129	เป็น
59	135	ไม่เป็น
60	**127**	**?**

ตัวอย่างการพยากรณ์โรคเบาหวาน โดยใช้ตัวแปรต้น คือ น้ำหนัก และ ความดัน

Classification

น้ำหนัก (kg)	ความดัน (mmHg)	เป็นโรคเบาหวาน
65	130	ไม่เป็น
42	142	ไม่เป็น
56	171	เป็น
71	129	เป็น
59	135	ไม่เป็น

Data

Model

น้ำหนัก (kg)	ความดัน (mmHg)	เป็นโรคเบาหวาน
60	**127**	**?**

Prediction

Regression

พื้นที่บ้าน (ตร.ม.)	จำนวนชั้น	ราคา (ล้านบาท)
165	1	1.89
211	2	10.03
200	2	30.1
143	1	1.99
187	2	4.5
142	**1**	**?**

ตัวอย่างการพยากรณ์ราคาบ้าน โดยใช้ตัวแปรต้น คือ พื้นที่บ้าน และ จำนวนชั้นของบ้าน

Regression

พื้นที่บ้าน (ตร.ม.)	จำนวนชั้น	ราคา (ล้านบาท)
165	1	1.89
211	2	10.03
200	2	30.1
143	1	1.99
187	2	4.5

Data

Model

พื้นที่บ้าน (ตร.ม.)	จำนวนชั้น	ราคา (ล้านบาท)
142	**1**	**?**

2.99

Prediction

AI Overview

- AI and Machine Learning
- Supervised Learning

DL101 : Multiple Regression

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

INTRODUCTION

INTRODUCTION

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Introduction

- What is Multiple Regression?
- Real World Application

What is Multiple Regression?

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

What is Multiple Regression?

Multiple Regression เป็นหนึ่งใน algorithm ประเภท supervised learning ที่ใช้สำหรับแก้ปัญหา regression โดยมีหลักการทำงานคือการสร้างสมการเชิงเส้นที่ใช้ตัวแปรต้นเพื่อพยากรณ์ตัวแปรตาม

What is Multiple Regression?

What is Multiple Regression?

สมการคณิตศาสตร์ของ Multiple Regression

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

โดย - $\hat{y}$ คือ ตัวแปรตาม (predicted target)

- $x_1, x_2, x_3, \ldots, x_p$ คือ ตัวแปรต้น (feature)
- $w_0, w_1, w_2, \ldots, w_p$ คือ สัมประสิทธิ์ (coefficient)

What is Multiple Regression?

$$\hat{y} = w_0 + w_1 x_1$$
$$(\hat{y} = a + bx)$$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2$$

Introduction

- **What is Multiple Regression?**
- Real World Application

Real World Application

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Real World Application

การประเมิณค่าตัวของนักฟุตบอลในตำแหน่งกองหน้า
โดยพิจารณาจาก อายุ ส่วนสูง การมีส่วนร่วมกับประตู จำนวนเกมที่ลงเล่น เป็นต้น

อ้างอิง : [2018, Yunus et al] Multiple Linear Regression Approach For Estimating the Market Value of Football Players in Forward Position

Real World Application

การคาดการณ์ค่าน้ำค่าไฟของฟาร์มโคนม

โดยพิจารณาจาก จำนวนโคนม จำนวนเครื่องรีดนมวัว จำนวนเครื่องขัดพื้นไฟฟ้า เป็นต้น

อ้างอิง : [2018, Shine et al] Multiple linear regression modelling of on-farm direct water and electricity consumption on pasture based dairy farms

Real World Application

พยากรณ์ปริมาณพืชที่จะปลูกได้

โดยพิจารณาจาก สภาพดิน สภาพอากาศ ปริมาณน้ำฝน อุณหภูมิ เป็นต้น

อ้างอิง : [2017, Aditya Shastry, HA Sanjay and E. Bhanusree]
Prediction of Crop Yield Using Regression Techniques

Introduction

- ✓ What is Multiple Regression?
- ✓ Real World Application

DL101 : Multiple Regression

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

MODEL CREATION

MODEL CREATION

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Multiple Regression

Multiple Regression เป็นหนึ่งใน algorithm ประเภท supervised learning

Concept of Supervised Learning

Data ⇨ Model ⇨ Prediction

Model Creation

- Data
- Model
- Prediction

Data

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Data

- Data Stating
- Data Requirement

Data Stating

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\cdots$	$\mathbf{x}_p$	$\mathbf{y}$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\cdots$	$x_{1,p}$	y_1
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\cdots$	$x_{2,p}$	y_2
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\cdots$	$x_{3,p}$	y_3
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$\cdots$	$x_{n,p}$	y_n

Data Stating

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\cdots$	$\mathbf{x}_p$	$\mathbf{y}$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\cdots$	$x_{1,p}$	y_1
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\cdots$	$x_{2,p}$	y_2
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\cdots$	$x_{3,p}$	y_3
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$\cdots$	$x_{n,p}$	y_n

- n คือ จำนวน sample
- p คือ จำนวน feature

Data Stating

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\cdots$	$\mathbf{x}_p$	$\mathbf{y}$
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\cdots$	$x_{1,p}$	y_1
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\cdots$	$x_{2,p}$	y_2
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\cdots$	$x_{3,p}$	y_3
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$\cdots$	$x_{n,p}$	y_n

- $x_{2,3}$ คือ sample ที่ 2 feature ที่ 3
- $x_{3,p}$ คือ sample ที่ 3 feature ที่ p
- $x_{n,p}$ คือ sample ที่ n feature ที่ p
- y_2 คือ target ของ sample ที่ 2
- y_3 คือ target ของ sample ที่ 3
- y_n คือ target ของ sample ที่ n

Data Stating

Example

- เราต้องการจะพยากรณ์ราคาบ้าน โดยดูองค์ประกอบจากจำนวนห้องน้า, จำนวนห้องนอน, พื้นที่ของบ้าน, ราคาที่ดินต่อตารางวา

Data

จำนวนห้องนอน (ห้อง)	จำนวนห้องน้ำ (ห้อง)	พื้นที่ของบ้าน (ตร.ว.)	ราคาที่ดิน (บาท/ตร.ว.)	ราคาขายบ้าน (ล้านบาท)
2	2	70	25000	3.5
3	2	120	30000	5.2
1	1	50	20000	1.2
2	1	80	35000	4.0

Data Stating

- ข้อมูลตามแนวแถว คือ Sample

จำนวนห้องนอน (ห้อง)	จำนวนห้องน้ำ (ห้อง)	พื้นที่ของบ้าน (ตร.ว.)	ราคาที่ดิน (บาท/ตร.ว.)	ราคาขายบ้าน (ล้านบาท)
2	2	70	25000	3.5
3	2	120	30000	5.2
1	1	50	20000	1.2
2	1	80	35000	4.0

Data Stating

- ข้อมูลตามแนวหลัก คือ Feature and Target
  - Feature (ตัวแปรต้น) คือ ข้อมูลที่ส่งผลให้เกิด target
  - Target (ตัวแปรตาม) คือ ข้อมูลที่เราสนใจจะพยากรณ์

Feature

จำนวนห้องนอน (ห้อง)	จำนวนห้องน้ำ (ห้อง)	พื้นที่ของบ้าน (ตร.ว.)	ราคาที่ดิน (บาท/ตร.ว.)
2	2	70	25000
3	2	120	30000
1	1	50	20000
2	1	80	35000

Target

ราคาขายบ้าน (ล้านบาท)
3.5
5.2
1.2
4.0

Data Stating

- Feature and Target
  - เราสามารถแยก และปรับให้เป็น matrix ได้ดังนี้

$$X = \begin{bmatrix} 2 & 2 & 70 & 25000 \\ 3 & 2 & 120 & 30000 \\ 1 & 1 & 50 & 20000 \\ 2 & 1 & 80 & 35000 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 3.5 \\ 5.2 \\ 1.2 \\ 4.0 \end{bmatrix}$$

Data

- **Data Stating**
- Data Requirement

Data Requirement

- ข้อมูลต้องอยู่ในรูปแบบของตาราง
- ข้อมูลต้องเป็น numerical

จำนวนห้องนอน (ห้อง)	จำนวนห้องน้ำ (ห้อง)	พื้นที่ของบ้าน (ตร.ว.)	ราคาที่ดิน (บาท/ตร.ว.)	ราคาขายบ้าน (ล้านบาท)
2	2	70	25000	3.5
3	2	120	30000	5.2
1	1	50	20000	1.2
2	1	80	35000	4.0

Data Requirement

- ตัวอย่างข้อมูลที่สามารถใช้งานได้เลย และยังไม่สามารถใช้งานได้

พื้นที่ของบ้าน (ตร.ว.)	ราคาที่ดิน (บาท/ตร.ว.)	ราคาขายบ้าน (ล้านบาท)
70	25000	3.5
120	30000	5.2
50	20000	1.2
80	35000	4.0

exp (yr)	position	salary
1	secretary	26000
4	engineer	48000
3	accountant	41500
1	engineer	26500

Data Requirement

- เราสามารถแปลงได้โดยสามารถใช้ความรู้ในส่วนของ Data Preparation

exp (yr)	position	salary
1	secretary	26000
4	engineer	48000
3	accountant	41500
1	engineer	26500

exp (yr)	accountant	engineer	secretary	salary
1	0	0	1	26000
4	0	1	0	48000
3	1	0	0	41500
1	0	1	0	26500

Data Requirement

For more information

Data Preparation

Data

- Data Stating
- Data Requirement

Model Creation

- **Data**
- Model
- Prediction

Model

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Model

- Assumption
- How to create model
- Further Reading

Assumption

1. Linear Relationship
2. Normality of Residuals
3. Homoscedasticity
4. No Missing Features
5. No Multicollinearity

Assumption

For more information

Model Improvement

Model

- **Assumption**
- How to create model
- Further Reading

How to create model

Multiple regression คือ สมการเชิงเส้นที่ใช้ตัวแปรต้นเพื่อพยากรณ์ตัวแปรตาม

How to create model

Multiple regression คือ สมการเชิงเส้นที่ใช้ตัวแปรต้นเพื่อพยากรณ์ตัวแปรตาม

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

โดย

- $\hat{y}$ คือ ตัวแปรตาม (predicted target)
- $x_1, x_2, \ldots, x_p$ คือ ตัวแปรต้น (feature)
- $w_0, w_1, \ldots, w_p$ คือ สัมประสิทธิ์ (coefficient)

How to create model

$\mathbf{x}_1$	$\mathbf{x}_2$	$\cdots$	$\mathbf{x}_p$		$\hat{\mathbf{y}}$
$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,p}$	$\rightarrow$	$\hat{y}_1$
$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,p}$	$\rightarrow$	$\hat{y}_2$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$x_{i,1}$	$x_{i,2}$	$\cdots$	$x_{i,p}$	$\rightarrow$	$\hat{y}_i$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$x_{n,1}$	$x_{n,2}$	$\cdots$	$x_{n,p}$	$\rightarrow$	$\hat{y}_n$

How to create model

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

$$\Downarrow$$

$$\hat{y}_1 = w_0 + w_1 x_{1,1} + w_2 x_{1,2} + \cdots + w_p x_{1,p}$$

$$\hat{y}_2 = w_0 + w_1 x_{2,1} + w_2 x_{2,2} + \cdots + w_p x_{2,p}$$

$$\vdots$$

$$\hat{y}_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \cdots + w_p x_{i,p}$$

$$\vdots$$

$$\hat{y}_n = w_0 + w_1 x_{n,1} + w_2 x_{n,2} + \cdots + w_p x_{n,p}$$

How to create model

"เป้าหมายของเราคือการหา $w_0, w_1, w_2, \ldots, w_p$ เพื่อสร้าง model ของ multiple regression :

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$"

How to create model

"โดยที่ $w_0, w_1, w_2, \ldots, w_p$ เหล่านั้นต้องทำให้ทำให้ผลรวมของ error ระหว่างค่าจริง (y_i) กับค่าพยากรณ์ ($\hat{y}_i$) น้อยที่สุด"

y	$\hat{y}$
y_1	$\hat{y}_1$
y_2	$\hat{y}_2$
⋮	⋮
y_n	$\hat{y}_n$

How to create model

“ เราต้องการหา $w_0, w_1, w_2, \ldots, w_p$ ที่ทำให้ผลรวมของ error ระหว่าง y_i กับ $\hat{y}_i$ น้อยที่สุด ”

How to create model

ผลรวมของ error ระหว่าง y_i กับ $\hat{y}_i$ **เรียกว่า "Cost function"**

How to create model

“ เราต้องการหา $w_0, w_1, w_2, \ldots, w_p$ ที่ทำให้ cost function ต่ำที่สุด ”

How to create model

Cost function ที่เราจะใช้ในการสร้าง model คือ

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

โดยสูตรข้างต้นมีชื่อว่า Sum of Squared Errors หรือ SSE

How to create model

เหตุผลที่เลือกใช้ Sum of Squared Errors (SSE)

1. Error ของแต่ละ sample จะไม่ติดลบ
2. Cost function ที่นิยามแบบ SSE จะสามารถ diff ได้ และมีความต่อเนื่องทุกจุด

How to create model

1. Error ของแต่ละ sample จะไม่ติดลบ

กราฟแสดงข้อมูลระหว่าง**ค่าจริง** y_i (จุด) และ**ค่าพยากรณ์** $\hat{y}_i$ (เส้น)

$$SE = \sum_{i=1}^{n}(y_i - \hat{y}_i)$$

$$= (-0.3) + (0.6) + (0.3) + (0.4) + (-1)$$

$$= 0$$

การมีค่าติดลบทำให้ error หักล้างกัน

How to create model

1. Error ของแต่ละ sample จะไม่ติดลบ

กราฟแสดงข้อมูลระหว่าง**ค่าจริง** y_i (จุด) และ**ค่าพยากรณ์** $\hat{y}_i$ (เส้น)

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

$$= (-0.3)^2 + (0.7)^2 + (0.2)^2 + (0.4)^2 + (-1)^2$$

$$= 0.09 + 0.49 + 0.04 + 0.16 + 1$$

$$= 1.78$$

☑ **การยกกำลังสองทำให้ error ไม่หักล้างกัน**

How to create model

2. Cost function ที่นิยามแบบ SSE จะสามารถ diff ได้ และมีความต่อเนื่องทุกจุด

กราฟแสดงภาพของ error surface โดยที่ cost function เป็น SSE

How to create model

กราฟนี้มาได้อย่างไร?

How to create model

ตัวอย่างการ plot กราฟของ cost function

x_1	x_2	y
0	1	4
2	1	8
1	1	6
2	0	5

ตารางแสดง toy dataset

How to create model

- คำนวณค่า $\hat{y} = w_0 + w_1x_1 + w_2x_2$ จาก w_1 และ w_2 แต่ละชุด (เพื่อความสะดวก เราจะกำหนดให้ $w_0 = 1$ เสมอ)

		$w_0 = 1$ $w_1 = 1.6$ $w_2 = 1.5$	$w_0 = 1$ $w_1 = 1.8$ $w_2 = 2$	$w_0 = 1$ $w_1 = 2$ $w_2 = 3$
$\mathbf{x_1}$	$\mathbf{x_2}$	$\hat{\mathbf{y}}_1$	$\hat{\mathbf{y}}_2$	$\hat{\mathbf{y}}_3$
0	1	2.5	3	4
2	1	5.7	6.6	8
1	1	4.1	4.8	6
2	0	4.2	4.6	5

How to create model

- นำ $\hat{y}$ ของ weight แต่ละชุดไปเปรียบเทียบกับ y เพื่อคำนวณ cost function ($Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$) ออกมา

$w_0 = 1$ $w_1 = 1.6$ $w_2 = 1.5$	$w_0 = 1$ $w_1 = 1.8$ $w_2 = 2$	$w_0 = 1$ $w_1 = 2$ $w_2 = 3$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$
2.5	3	4
5.7	6.6	8
4.1	4.8	6
4.2	4.6	5

y
4
8
6
5

How to create model

$w_0 = 1$
$w_1 = 1.6$
$w_2 = 1.5$

$\hat{y}_1$	y
2.5	4
5.7	8
4.1	6
4.2	5
$Cost = 11.79$	

$w_0 = 1$
$w_1 = 1.8$
$w_2 = 2$

$\hat{y}_2$	y
3	4
6.6	8
4.8	6
4.6	5
$Cost = 4.56$	

$w_0 = 1$
$w_1 = 2$
$w_2 = 3$

$\hat{y}_3$	y
4	4
8	8
6	6
5	5
$Cost = 0$	

How to create model

○ การคำนวณค่า $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ ของ $\hat{\mathbf{y}}_1$ และ $\mathbf{y}$

$w_0 = 1$
$w_1 = 1.6$
$w_2 = 1.5$

$\hat{\mathbf{y}}_1$	$\mathbf{y}$	$(\mathbf{y} - \hat{\mathbf{y}}_1)^2$
2.5	4	2.25
5.7	8	5.29
4.1	6	3.61
4.2	5	0.64
	$Cost =$	11.79

How to create model

○ การคำนวณค่า $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ ของ $\hat{\mathbf{y}}_2$ และ $\mathbf{y}$

$w_0 = 1$
$w_1 = 1.8$
$w_2 = 2$

$\hat{\mathbf{y}}_2$	$\mathbf{y}$	$(\mathbf{y} - \hat{\mathbf{y}}_2)^2$
3	4	1
6.6	8	1.96
4.8	6	1.44
4.6	5	0.16
	$Cost =$	4.56

How to create model

○ การคำนวณค่า $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ ของ $\hat{\mathbf{y}}_3$ และ $\mathbf{y}$

$w_0 = 1$
$w_1 = 2$
$w_2 = 3$

$\hat{\mathbf{y}}_3$	$\mathbf{y}$	$(\mathbf{y} - \hat{\mathbf{y}}_3)^2$
4	4	0
8	8	0
6	6	0
5	5	0
	$Cost =$	0

How to create model

- นำค่า $Cost$ ของ weight แต่ละชุดไป plot ลงกราฟ

	$w_0 = 1$ $w_1 = 1.6$ $w_2 = 1.5$	$w_0 = 1$ $w_1 = 1.8$ $w_2 = 2$	$w_0 = 1$ $w_1 = 2$ $w_2 = 3$
$Cost$	11.79	4.56	0

How to create model

How to create model

การที่ function **diff ได้** และ**ต่อเนื่องทุกจุด**ทำให้เราสามารถ**ใช้ calculus ได้อย่างเต็มที่**

How to create model

“ เราต้องการหา $w_0, w_1, w_2, \ldots, w_p$ ที่ทำให้ cost function น้อยที่สุด ”

How to create model

ดังนั้น จุดที่เราต้องการคือจุดต่ำสุดของ cost function ซึ่งสามารถใช้ calculus ในการหาได้

สามารถใช้ calculus
เพื่อหาจุดต่ำสุดได้เลย

How to create model

โดยที่คุณสมบัติของจุดต่ำสุดคือ

“ gradient ของ cost function เท่ากับ 0 ($\nabla Cost = 0$) ”

ซึ่งหมายความว่า ความชันของ cost function ในมิติใด ๆ เท่ากับ 0

$$\frac{\partial Cost}{\partial w_d} = 0 \; ; d = 0, \dots, p$$

How to create model

การที่ความชันของ cost ในมิติใด ๆ เท่ากับ 0 หรือ $\frac{\partial Cost}{\partial w_d} = 0 \; ; d = 0, \dots, p$ สามารถเขียนเป็นสมการได้ดังต่อไปนี้

$$\frac{\partial Cost}{\partial w_0} = 0$$

$$\frac{\partial Cost}{\partial w_1} = 0$$

$$\vdots$$

$$\frac{\partial Cost}{\partial w_p} = 0$$

How to create model

ดังนั้น $\nabla Cost = 0$ สามารถเขียนใหม่ได้ดังนี้

$$\nabla Cost = \begin{bmatrix} \frac{\partial Cost}{\partial w_0} \\ \frac{\partial Cost}{\partial w_1} \\ \vdots \\ \frac{\partial Cost}{\partial w_p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

ONE
HOUR LATER…

How to create model

Normal Equation คือ สมการที่ใช้ในการหา weight ของ model

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

โดยที่ $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$, $X_b = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

How to create model

Derivation of Normal Equation

Open File
Derive_NormalEq.pdf

How to create model

ตัวอย่างการคำนวณ w ด้วย normal equation

x_1	x_2	y
0	1	4
2	1	8
1	1	6
2	0	5

ตารางแสดง dataset

How to create model

- จากข้อมูลใน dataset เราสามารถเขียน $X, \mathbf{y}$ และ X_b ได้ดังต่อไปนี้

$$X = \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 1 & 1 \\ 2 & 0 \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} 4 \\ 8 \\ 6 \\ 5 \end{bmatrix} \ \text{และ} \ X_b = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix}$$

How to create model

- จากสูตร normal equation $\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$ จะได้ว่า

$$\mathbf{w} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 2 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix}\right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 2 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 8 \\ 6 \\ 5 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 5 & 3 \\ 5 & 9 & 3 \\ 3 & 3 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 23 \\ 32 \\ 18 \end{bmatrix}$$

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

How to create model

ดังนั้น เราจะได้ model ของ multiple regression สำหรับข้อมูลชุดนี้คือ

$$\boxed{\hat{y} = 1 + 2x_1 + 3x_2}$$

How to create model

Exercise of Normal Equation

Open File

Exercise_NormalEq.pdf

How to create model

ตัวอย่าง code สำหรับคำนวณ w

x_1	x_2	y
0	1	4
2	1	8
1	1	6
2	0	5

ตารางแสดง toy dataset

How to create model

- Code สำหรับสร้าง model จากข้อมูลของเราโดยที่

$$X = \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 1 & 1 \\ 2 & 0 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 4 \\ 8 \\ 6 \\ 5 \end{bmatrix}$$

```
In [1]:
reg = LinearRegression()
reg.fit(X, y)

Out[1]: LinearRegression()
```

How to create model

- ค่า w_0 ถูกเก็บไว้ใน attribute ชื่อ `intercept_`

```
In [2]:  reg.intercept_

Out[2]: 1.
```

How to create model

- ค่า $w_1, \ldots, w_p$ ถูกเก็บไว้ใน attribute ชื่อ coef_

```
In [3]:  reg.coef_

Out[3]: array([2., 3.])
```

How to create model

ดังนั้น เราจะได้ $w_0 = 1, w_1 = 2$ และ $w_2 = 3$ ซึ่งสามารถเขียนเป็น model ของ multiple regression สำหรับข้อมูลชุดนี้ได้ดังนี้

$$\boxed{\hat{y} = 1 + 2x_1 + 3x_2}$$

How to create model

Code for this section

Open File

Model Creation.ipynb

How to create model

MATH

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}$$

CODE

$$\texttt{intercept_} = w_0$$

$$\texttt{coef_} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}$$

How to create model

MATH

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

CODE

```
In [1]:  reg = LinearRegression()
         reg.fit(X, y)

Out[1]:  LinearRegression()

In [2]:  reg.intercept_

Out[2]:  1.

In [3]:  reg.coef_

Out[3]:  array([2., 3.])
```

Model

- ✓ **Assumption**
- ✓ **How to create model**
- Further Reading

Further Reading

- Gradient Descent
- Moore-Penrose pseudo inverse
- Feature importance with p-value

Gradient Descent

Moore-Penrose pseudo inverse

“ Normal Equation ”

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

Feature importance with p-value

Model

- Assumption
- How to create model
- Further Reading

Model Creation

- ✓ **Data**
- ✓ **Model**
- Prediction

Prediction

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Prediction

Multiple regression คือ สมการเชิงเส้นที่ใช้ตัวแปรต้นเพื่อพยากรณ์ตัวแปรตาม

$$\boxed{\hat{y} = w_0 + w_1x_1 + w_2x_2 + \cdots + w_px_p}$$

โดย

- $\hat{y}$ คือ ค่าพยากรณ์ของตัวแปรตาม (predicted target)
- $x_1,\ x_2, x_3,\ \ldots,\ x_p$ คือ ตัวแปรต้น (feature)
- $w_0,\ w_1, w_2,\ \ldots,\ w_p$ คือ สัมประสิทธิ์ (coefficient)

Prediction

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$$

Prediction

$\mathbf{x}_1$	$\mathbf{x}_2$	$\cdots$	$\mathbf{x}_p$		$\hat{\mathbf{y}}$
$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,p}$	$\rightarrow$	$\hat{y}_1$
$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,p}$	$\rightarrow$	$\hat{y}_2$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$x_{i,1}$	$x_{i,2}$	$\cdots$	$x_{i,p}$	$\rightarrow$	$\hat{y}_i$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$x_{n,1}$	$x_{n,2}$	$\cdots$	$x_{n,p}$	$\rightarrow$	$\hat{y}_n$

Prediction

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

$$\hat{y}_1 = w_0 + w_1 x_{1,1} + w_2 x_{1,2} + \cdots + w_p x_{1,p}$$

$$\hat{y}_2 = w_0 + w_1 x_{2,1} + w_2 x_{2,2} + \cdots + w_p x_{2,p}$$

$$\vdots$$

$$\hat{y}_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \cdots + w_p x_{i,p}$$

$$\vdots$$

$$\hat{y}_n = w_0 + w_1 x_{n,1} + w_2 x_{n,2} + \cdots + w_p x_{n,p}$$

Prediction

เพื่อให้สอดคล้องกับ format ของ data เราสามารถเขียนให้อยู่ในรูปของ matrix ได้ดังนี้

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_{11} + w_2 x_{12} + \cdots + w_p x_{1p} \\ w_0 + w_1 x_{21} + w_2 x_{22} + \cdots + w_p x_{2p} \\ \vdots \\ w_0 + w_1 x_{n1} + w_2 x_{n2} + \cdots + w_p x_{np} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}$$

Prediction

$$\boxed{\hat{\mathbf{y}} = X_b \mathbf{w}}$$

โดยที่ $\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}, \quad X_b = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$

Prediction

ตัวอย่างการคำนวณ $\hat{y}$

x_1	x_2
1	1
2	0
3	1
3	0

$\hat{y}$
?
?
?
?

Prediction

- สมมติว่า w ของปัญหานี้ที่เราหามาได้คือ

$$\mathbf{w} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

- และจากข้อมูลใน dataset เราสามารถเขียน X_b ได้ดังต่อไปนี้

$$X_b = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 3 & 1 \\ 1 & 3 & 0 \end{bmatrix}$$

Prediction

จาก model ของ multiple regression $\hat{\mathbf{y}} = X_b\mathbf{w}$ จะได้ว่า

$$\hat{\mathbf{y}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 3 & 1 \\ 1 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} (1 \times 1) + (2 \times 1) + (3 \times 1) \\ (1 \times 1) + (2 \times 2) + (3 \times 0) \\ (1 \times 1) + (2 \times 3) + (3 \times 1) \\ (1 \times 1) + (2 \times 3) + (3 \times 0) \end{bmatrix}$$

$$= \begin{bmatrix} 6 \\ 5 \\ 10 \\ 7 \end{bmatrix}$$

Prediction

ดังนั้น เราจะได้ $\hat{y}$ สำหรับข้อมูลชุดนี้คือ

x_1	x_2
1	1
2	0
3	1
3	0

$\hat{y}$
6
5
10
7

Prediction

ตัวอย่าง code สำหรับหา $\hat{y}$

x_1	x_2
1	1
2	0
3	1
3	0

$\hat{y}$
?
?
?
?

ตารางแสดง toy dataset

Prediction

- Code สำหรับหา $\hat{y}$ จากข้อมูลของเรา (X) โดยที่ $X = \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 3 & 1 \\ 3 & 0 \end{bmatrix}$

```
In [ 4 ]:  1  reg.predict(X)

Out[ 4 ]: array([ 6.,  5., 10.,  7.])
```

Prediction

ดังนั้น เราจะได้ $\hat{y}$ สำหรับข้อมูลชุดนี้คือ

x_1	x_2
1	1
2	0
3	1
3	0

$\hat{y}$
6
5
10
7

Prediction

Code for this section

Open File
Model Creation.ipynb

Prediction

MATH

$$\hat{\mathbf{y}} = X_b \mathbf{w}$$

CODE

```
In [4]:  1  reg.predict(X)

Out[4]: array([4., 5., 6., 7.])
```

Model Creation

- ✓ Data
- ✓ Model
- ✓ Prediction

DL101 : Multiple Regression

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

BASIC WORKSHOP

BASIC WORKSHOP

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Supervised Learning Workflow

Code Pipeline

Import Libraries

1. NumPy
2. pandas
3. scikit learn
4. matplotlib

Code

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import (r2_score,
                             mean_squared_error,
                             mean_absolute_error,
                             mean_absolute_percentage_error)

import warnings
warnings.filterwarnings('ignore')

np.random.seed(12345)
```

Read Data

	age	experience	gpa	degree	position	salary
0	30.0	7.0	3.94	bachelor	engineer	32500.0
1	26.0	2.0	2.86	bachelor	NaN	22500.0
2	27.0	0.0	3.13	doctorate	secretary	37000.0
3	32.0	NaN	3.10	bachelor	engineer	24500.0
4	24.0	1.0	3.81	bachelor	accountant	23500.0
5	35.0	7.0	3.93	doctorate	secretary	43500.0
6	23.0	1.0	3.78	master	accountant	30500.0
7	32.0	8.0	3.04	bachelor	accountant	31500.0
8	27.0	2.0	3.52	bachelor	secretary	18500.0
	⋮	⋮	⋮	⋮	⋮	⋮

Code

```
data = pd.read_csv('salary_dataset.csv')
```

Clean Data

1. Handle Missing Values
2. Handle Outliers

Clean Data

1. **Handle Missing Values**
2. Handle Outliers

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   age         89 non-null     float64
 1   experience  89 non-null     float64
 2   gpa         90 non-null     float64
 3   degree      90 non-null     object
 4   position    89 non-null     object
 5   salary      89 non-null     float64
dtypes: float64(4), object(2)
memory usage: 4.3+ KB
```

Code

- Check Missing Values

```
data.info()
```

- Remove Missing Values

```
data.dropna(axis=0, inplace=True)
```

Clean Data

1. Handle Missing Values
2. **Handle Outliers**

	age	experience	gpa	salary
count	86.000000	86.000000	86.000000	86.000000
mean	28.023256	3.848837	3.278605	31348.837209
std	4.408486	3.702201	0.528937	9255.227384
min	21.000000	0.000000	2.540000	13000.000000
25%	24.000000	1.000000	2.820000	24500.000000
50%	28.000000	3.000000	3.260000	30500.000000
75%	32.000000	7.000000	3.640000	37375.000000
max	35.000000	13.000000	5.880000	54000.000000

Code

- Check Outliers

```
data.describe()
```

- Remove Outliers

```
_filter = data['gpa'] <= 4.00
data = data[_filter]
```

Train/Test

แบ่งข้อมูลออกเป็น 2 ชุด คือ training set และ test set ด้วยอัตราส่วน 80:20 ตามลำดับ

Code

```
target_name = 'salary'
feature_name = list(data.columns.drop(target_name))
```

```
X = data[feature_name]
y = data[target_name]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, shuffle=True)
```

Data Preparation

Data Preparation

Data Preparation

Type of Features

พิจารณาและจำแนก feature ที่มีลักษณะข้อมูลแบบ numerical data และ categorical data ออกจากกัน

Numerical Data ✓

Categorical Data ✓

Code

```
numerical_feature = ['age', 'experience', 'gpa']
categorical_feature = ['degree', 'position']
```

Data Preparation

Type of Categorical Features

พิจารณาและจำแนก feature ที่มีลักษณะข้อมูลแบบ ordinal data และ nominal data ออกจากกัน

```
degree : ['bachelor' 'doctorate' 'master']
position : ['accountant' 'engineer' 'secretary']
```

Code

- Consider each Feature of Categorical Features

```
for feature in categorical_feature:
    print(feature, ':', np.unique(X_train[feature]))
```

- Classify into Ordinal Feature and Nominal Feature

```
ordinal_feature = ['degree']
nominal_feature = ['position']
```

Data Preparation

Ordinal Encoding

การทำ ordinal encoding จะต้องทำแบบเดียวกัน ทั้งใน training set และ test set

	degree
0	bachelor
1	master
2	bachelor
3	bachelor
4	bachelor

	degree
0	0.0
1	1.0
2	0.0
3	0.0
4	0.0

Code

- Ordinal Data

```
ordinal_feature = ['degree']
```

```
for feature in categorical_feature:
    print(feature, ':', np.unique(X_train[feature]))
```

Code

- Ordinal Encoding for **training set**

```
ordinal_encoder = OrdinalEncoder(categories=categories)
X_train[ordinal_feature] = ordinal_encoder.fit_transform(X_train[ordinal_feature])
```

- Ordinal Encoding for **test set**

```
X_test[ordinal_feature] = ordinal_encoder.transform(X_test[ordinal_feature])
```

Data Preparation

One Hot Encoding

การทำ one hot encoding จะต้องทำแบบเดียวกันทั้งใน training set และ test set

	position
0	secretary
1	secretary
2	engineer
3	engineer
4	secretary

	position_accountant	position_engineer	position_secretary
0	0.0	0.0	1.0
1	0.0	0.0	1.0
2	0.0	1.0	0.0
3	0.0	1.0	0.0
4	0.0	0.0	1.0

Code

- Nominal Data

```
nominal_feature = ['position']
```

- One Hot Encoding

```
one_hot_encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
one_hot_encoder.fit(X_train[nominal_feature])
```

```
one_hot_feature = []
for i, feature in enumerate(nominal_feature):
    for cate in one_hot_encoder.categories_[i]:
        one_hot_feature_name = str(feature) + '_' + str(cate)
        one_hot_feature.append(one_hot_feature_name)
```

Code

- One Hot Encoding for **training set**

```
X_train[one_hot_feature] = one_hot_encoder.transform(X_train[nominal_feature])
X_train.drop(nominal_feature, axis=1, inplace=True)
```

- One Hot Encoding for **test set**

```
X_test[one_hot_feature] = one_hot_encoder.transform(X_test[nominal_feature])
X_test.drop(nominal_feature, axis=1, inplace=True)
```

Data Preparation

Feature Scaling

สำหรับ Normal Equation

✕ ไม่จำเป็นต้องทำ ✕

Create Model

Code

- Fit model

```
reg = LinearRegression()
reg.fit(X_train, y_train)
```

- w_0

```
reg.intercept_
```

- $w_1, \dots, w_p$

```
reg.coef_
```

Import Libraries
Read Data
Clean Data
Train
Test
Data Preparation
Data Preparation
Create Model
Prediction
Prediction
Model Evaluation
Model Evaluation
Model Deployment

Prediction

Code

- Prediction for **training set**

```
y_pred_train = reg.predict(X_train)
```

- Prediction for **test set**

```
y_pred_test = reg.predict(X_test)
```

Model Evaluation

1. Scoring ($R^2, MSE, MAE, MAPE$)
2. Scatter Plot between Predicted & Actual Values

Model Evaluation

1. **Scoring ($R^2, MSE, MAE, MAPE$)**
2. Scatter Plot between Predicted & Actual Values

- Scoring for **training set**

```
r2_score =                          0.8834907214681481
mean_squared_error =                9954664.630857592
mean_absolute_error =               2661.546753531732
mean_absolute_percentage_error = 0.09401655935291867
```

- Scoring for **test set**

```
r2_score =                          0.8921423690778735
mean_squared_error =                9282847.245418865
mean_absolute_error =               2569.1394359308074
mean_absolute_percentage_error = 0.09654987694303836
```

Code

- Scoring for **training set**

```
print('r2_score =\t\t\t', r2_score(y_train, y_pred_train))
print('mean_squared_error =\t\t', mean_squared_error(y_train, y_pred_train))
print('mean_absolute_error =\t\t', mean_absolute_error(y_train, y_pred_train))
print('mean_absolute_percentage_error =', mean_absolute_percentage_error(y_train, y_pred_train))
```

- Scoring for **test set**

```
print('r2_score =\t\t\t', r2_score(y_test, y_pred_test))
print('mean_squared_error =\t\t', mean_squared_error(y_test, y_pred_test))
print('mean_absolute_error =\t\t', mean_absolute_error(y_test, y_pred_test))
print('mean_absolute_percentage_error =', mean_absolute_percentage_error(y_test, y_pred_test))
```

Model Evaluation

1. Scoring ($R^2, MSE, MAE, MAPE$)
2. **Scatter Plot between Predicted & Actual Values**

- Scatter Plot for **training set**

- Scatter Plot for **test set**

Code

- Scatter Plot for **training set**

```
plt.scatter(y_pred_train, y_train)

plt.plot(y_pred_train, y_pred_train, color='red')

plt.title('Scatter Plot between Predicted & Actual Values')
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

- Scatter Plot for **test set**

```
plt.scatter(y_pred_test, y_test)

plt.plot(y_pred_test, y_pred_test, color='red')

plt.title('Scatter Plot between Predicted & Actual Values')
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

Model Deployment

Code

```python
import pickle

pickle.dump((reg,
             ordinal_encoder,
             one_hot_encoder,
             feature_name,
             numerical_feature,
             ordinal_feature,
             nominal_feature),
            open('salary_model.pickle', 'wb'))
```

Import Libraries
Read Data
Clean Data
Train
Test
Data Preparation
Data Preparation
Create Model
Prediction
Prediction
Model Evaluation
Model Evaluation
Model Deployment

TAUTOLOGY
INNOVATION
SCHOOL

AI in Marketing

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Abstract

สร้าง model เพื่อพยากรณ์รายได้ที่ได้รับจากการยิง ads โดย feature ที่นำมาใช้ คือ ข้อมูลต่าง ๆ ที่เก็บได้จากการยิง ads เช่น

- จำนวนเงินที่ใช้ในการยิง ads
- จำนวนครั้งที่มีคนคลิก ads

Why this project important?

- สามารถวางแผนรายได้ของบริษัท จาก performance ของ ads ใน case ต่าง ๆ
- สามารถต่อยอดกับการทำ marketing บน platform อื่น ๆ

Who this project is for?

- ผู้ขายสินค้าออนไลน์
- ผู้ดูแลเพจ
- Marketing consultant
- นักวิเคราะห์ข้อมูล

Ads Dataset

chrisBow First commit

Latest commit a7ffe73 on Oct 11, 2018 History

1 contributor

41 lines (41 sloc) 1.48 KB

Raw Blame

Search this file...

spend	clicks	impressions	display	transactions	revenue	ctr	con_rate
22.61	165	8672	0	2	58.88	1.9	1.21
37.28	228	11875	0	2	44.92	1.92	0.88
55.57	291	14631	0	3	141.56	1.99	1.03
45.42	247	11709	0	2	209.76	2.11	0.81
50.22	290	14768	0	3	197.68	1.96	1.03
33.05	172	8698	0	2	204.36	1.98	1.16
12.88	68	2924	0	1	117.32	2.33	1.47
23.87	112	5919	0	1	72.04	1.89	0.89
58.38	306	14789	0	3	290.4	2.07	0.98
48.16	300	14818	0	3	245.52	2.02	1
63.18	355	17488	0	4	264.6	2.03	1.13
81.28	533	26411	0	5	469.4	2.02	0.94
28.33	177	9011	0	2	66.72	1.96	1.13

https://github.com/chrisBow/marketing-regression-part-one

Ads Dataset

Feature

- spend : จำนวนเงินที่ใช้ในการยิง ads
- clicks : จำนวนครั้งที่มีการคลิก ads
- impressions : จำนวนครั้งที่ ads ปรากฏบน website
- display : มี web address ปรากฏบน ads หรือไม่
- transactions : จำนวนครั้งที่มีการทำธุรกรรม
- ctr : จำนวนครั้งที่คลิกโฆษณาต่อจำนวนครั้งที่โฆษณาปรากฏ (clickthrough rate)
- con_rate : จำนวนสินค้าที่ถูกซื้อต่อจำนวนคนที่เข้ามาดูสินค้า (conversion rate)

Target

- revenue : รายได้ที่ได้รับจากการยิง ads

What we learn from this project?

Data Preparation

01. MARKETING

- ads_dataset.csv
- marketing_mc.ipynb
- marketing_md.ipynb
- marketing_model.pickle

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

AI in Investment

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Abstract

สร้าง model เพื่อพยากรณ์ราคาสูงสุดรายวันของ SET50 โดย feature ที่นำมาใช้ คือ technical indicator ต่าง ๆ เช่น

- standard deviation
- RSI

Why this project important?

- สามารถสร้างระบบการซื้อขายที่ปราศจากอารมณ์ของมนุษย์
- สามารถเป็นพื้นฐานสำหรับสร้างระบบการซื้อขายที่ robust มากขึ้น
- สามารถต่อยอดกับการเก็งกำไรบนสินทรัพย์ชนิดอื่น ๆ

Who is this project for?

- นักลงทุน
- ผู้ดูแลกองทุน
- นักวิจัยเชิงปริมาณ
- นักวิเคราะห์ข้อมูล

SET50 Dataset

ข้อมูลของ SET50 สามารถเก็บได้ที่เว็บไซต์ https://www.investing.com

SET 50 Historical Data

Time Frame:
Daily

Download Data

09/06/2011 - 10/07/2021

Date	Price	Open	High	Low	Vol.	Change %
Sep 06, 2011	734.49	724.53	734.49	722.61	836.64K	0.66%
Sep 07, 2011	744.18	736.88	744.25	735.60	915.18K	1.32%
Sep 08, 2011	747.06	740.70	747.06	738.38	844.77K	0.39%
Sep 09, 2011	739.73	747.53	748.28	738.83	1.09M	-0.98%
Sep 12, 2011	723.30	728.89	730.86	720.25	925.92K	-2.22%
Sep 13, 2011	716.87	726.64	728.53	713.70	685.74K	-0.89%
Sep 14, 2011	711.34	720.18	720.69	701.79	845.61K	-0.77%
Sep 15, 2011	721.34	714.74	721.62	708.28	1.19M	1.41%
Sep 16, 2011	718.64	726.70	728.32	716.88	663.14K	-0.37%
Sep 19, 2011	706.87	712.53	714.67	706.87	621.48K	-1.64%
Sep 20, 2011	714.70	706.12	714.70	700.93	643.67K	1.11%
Sep 21, 2011	717.08	711.76	718.85	710.24	650.54K	0.33%

SET50 Dataset

Feature

- Open_0 : ราคาเปิดของวันนั้น ๆ
- STD5_Open_0 : ส่วนเบี่ยงเบนมาตรฐานของราคาเปิด 5 วันล่าสุด
- RSI14_Open_0 : ค่าที่บอกการแกว่งของราคาเปิด 14 วันล่าสุด
- ATR14_0 : ค่าที่บอกความผันผวนของราคา 14 วันล่าสุด

Target

- High_0 : ราคาสูงสุดของวันนั้น ๆ

What we learn from this project?

Data Preparation

02. INVESTMENT/set50

- set50_dataset.csv
- set50_mc.ipynb
- set50_md.ipynb
- set50_model.pickle

02. INVESTMENT/btcusd

- btcusd_dataset.csv
- btcusd_mc.ipynb
- btcusd_md.ipynb
- btcusd_model.pickle

02. INVESTMENT/eurusd

- eurusd_dataset.csv
- eurusd_mc.ipynb
- eurusd_md.ipynb
- eurusd_model.pickle

02. INVESTMENT/xauusd

- xauusd_dataset.csv
- xauusd_mc.ipynb
- xauusd_md.ipynb
- xauusd_model.pickle

**ข้อมูลสามารถเก็บได้ที่เว็บไซต์ https://www.investing.com

TAUTOLOGY
INNOVATION
SCHOOL

Smart Farm

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Abstract

สร้าง model เพื่อพยากรณ์ปริมาณน้ำที่ต้นข้าวต้องการ โดย feature ที่นำมาใช้คือข้อมูลของปัจจัยต่าง ๆ ที่ส่งผลต่อการใช้น้ำของข้าว เช่น

- สภาพดิน
- สภาพอากาศ

Why this project important?

- สามารถควบคุม วางแผนปริมาณน้ำที่จะใช้ได้ และประหยัดต้นทุน
- สามารถนำความรู้ไปต่อยอดเพื่อสร้าง smart farm
- สามารถต่อยอดกับการพยากรณ์ปริมาณน้ำที่พืชชนิดอื่น ๆ ต้องการ

Who this project is for?

- เกษตรกรที่สนใจ AI กับการเกษตร
- ผู้ควบคุม/วางแผนการผลิต
- นักวิเคราะห์ข้อมูล

Rice Dataset

https://www.kaggle.com/prateekkkumar/crop-water-requirement

Rice Dataset

Feature

- SOIL TYPE
- REGION
- TEMPERATURE
- WEATHER CONDITION

Target

- WATER REQUIREMENT

Rice Dataset

ปริมาณน้ำที่พืชต้องการ ถูกเก็บจากเครื่องมือที่มีชื่อว่า “Lysimeter Tank”

What we learn from this project?

Data Preparation

03. SMART FARM/rice

- rice_dataset.xlsx
- rice_mc.ipynb
- rice_md.ipynb
- rice_model.pickle

02. INVESTMENT/banana

- banana_dataset.xlsx
- banana_mc.ipynb
- banana_md.ipynb
- banana_model.pickle

02. INVESTMENT/melon

- melon_dataset.xlsx
- melon_mc.ipynb
- melon_md.ipynb
- melon_model.pickle

TAUTOLOGY
INNOVATION
SCHOOL

AI in Business

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Abstract

สร้าง model เพื่อพยากรณ์จำนวนลูกค้าที่จะมาใช้บริการเช่ายืมจักรยาน โดย feature ที่นำมาใช้ คือ ข้อมูลเหตุการณ์ที่เกิดขึ้นในแต่ละวัน เช่น

- ฤดูกาล
- สภาพอากาศ
- เป็นวันทำงานหรือวันหยุด

Why this project important?

- สามารถนำความรู้ที่ได้จากการสร้าง model ไปประยุกต์ใช้กับธุรกิจประเภทอื่น ๆ ที่มีลักษณะคล้ายกัน
- สามารถ stock สินค้าที่แต่ละสาขาได้อย่างเหมาะสมที่สุด

Who this project is for?

- เจ้าของธุรกิจที่มีลักษณะใกล้เคียงกับ bike sharing เช่น ธุรกิจซักผ้า หยอดเหรียญ ธุรกิจเช่ารถ
- Business consultant
- ผู้จัดการ stock สินค้า
- นักวิเคราะห์ข้อมูล

Bike Sharing Dataset

https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset

Bike Sharing Dataset

Feature

- season : ฤดูกาล
- yr : ปี
- mnth : เดือน
- holiday : เป็น holiday หรือไม่ใช่ holiday
- weekday : วันของแต่ละสัปดาห์
- workingday : เป็น working day หรือไม่ใช่ working day
- weathersit : เป็นวันที่สภาพอากาศเป็นอย่างไร เช่น สภาพอากาศสดใส, ฝนตกหนัก, มีหมอก

Bike Sharing Dataset

Feature

- temp : อุณหภูมิ
- hum : ความชื้น
- windspeed : เป็นวันที่มีความเร็วลมเท่าไร

Target

- count : จำนวนการเช่าจักรยานในวันนั้น ๆ

What we learn from this project?

Data Preparation

04. BIKE RENTAL

- bike_sharing_dataset.csv
- bike_sharing_mc.ipynb
- bike_sharing_md.ipynb
- bike_sharing_model.pickle

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

AI in Insurance

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Abstract

สร้าง model เพื่อประเมิณค่าประกันภัยส่วนบุคคล โดย feature ที่นำมาใช้ คือ ประวัติทั่วไป และประวัติด้านสุขภาพ ของผู้ทำประกัน เช่น

- เพศ
- อายุ
- BMI

Why this project important?

- สามารถวางแผนตั้งราคาเบี้ยประกันภัยที่เหมาะสมที่สุด
- สามารถเพิ่มขีดความสามารถในการแข่งขันกับบริษัทคู่แข่ง

Who this project is for?

- นักคณิตศาสตร์ประกันภัย
- ผู้วางแผนทำประกันภัย
- นักวิเคราะห์ข้อมูล

Insurance Dataset

https://www.kaggle.com/sonujha090/insurance-prediction

Insurance Dataset

Feature

- age : อายุ
- sex : เพศ
- BMI : ค่า BMI
- children : จำนวนบุตร
- region : พื้นที่ที่พักอาศัย

Target

- charges : ค่าประกันที่จ่ายให้กับบริษัทประกันภัย

What we learn from this project?

What we learn from this project?

Feature Engineering

สร้าง feature ใหม่ เพื่อเก็บข้อมูลระดับภาวะน้ำหนักตัวโดยใช้ค่า BMI ในการแบ่ง

	age	sex	bmi	children	region	charges	bmiclass
0	19	0	27.900	0	3	16884.92400	3
1	62	0	26.290	0	2	27808.72510	3
2	27	1	42.130	0	2	39611.75770	4
3	30	1	35.300	0	3	36837.46700	4
4	34	0	31.920	1	0	37701.87680	4

Code

```
data['bmiclass'] = (
    (data['bmi'] < 18.5) * 1
    + ((data['bmi'] >= 18.5) & (data['bmi'] < 23)) * 2
    + ((data['bmi'] >= 23) & (data['bmi'] < 30)) * 3
    + (data['bmi'] >= 30) * 4
)
```

Data Preparation

05. INSURANCE/SMOKER

- Insurance_smoker_dataset.csv
- Insurance_smoker_mc.ipynb
- Insurance_smoker_md.ipynb
- Insurance_smoker_model.pickle

05. INSURANCE/NON SMOKER

- Insurance_non_smoker_dataset.csv
- Insurance_non_smoker_mc.ipynb
- Insurance_non_smoker_md.ipynb
- Insurance_non_smoker_model.pickle

DL101 : Multiple Regression

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

MODEL IMPROVEMENT

MODEL IMPROVEMENT

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

facebook/tautologyai
www.tautology.live

Model Improvement

- Assumption
- Problem with Linearly Dependent
- Solution
- Regularization

Assumption

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Assumption

1. Linear Relationship
2. Normality of Residuals
3. Homoscedasticity
4. No Missing Features
5. No Multicollinearity

Linear Relationship

Linear Relationship : ความสัมพันธ์ระหว่างตัวแปรต้น และตัวแปรตาม ต้องเป็นความสัมพันธ์แบบเชิงเส้นต่อกัน

Linear Relationship

เหตุผลที่เป็นเช่นนี้เพราะ model อยู่ในรูป

$$\boxed{\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p}$$

โดย

- $\hat{y}$ คือ ค่าพยากรณ์ของตัวแปรตาม (predicted target)
- $x_1, x_2, \ldots, x_p$ คือ ตัวแปรต้น (feature)
- $w_0, w_1, w_2, \ldots, w_p$คือ สัมประสิทธิ์ (coefficient)

Linear Relationship

Q : เราจะรู้ได้อย่างไรว่าตัวแปรต้น และตัวแปรตาม มีความสัมพันธ์เชิงเส้นต่อกันหรือไม่?

Linear Relationship

A : ตรวจสอบความสัมพันธ์ระหว่างตัวแปรต้นและตัวแปรตามได้ด้วย

i) Scatter plot ระหว่าง $\hat{y}$ กับ y
ii) Assumption ข้อที่ 2
iii) Assumption ข้อที่ 3

Linear Relationship

scatter plot ระหว่าง $\hat{y}$ กับ y

☒ Nonlinear Relationship

☑ Linear Relationship

Assumption

1. **Linear Relationship**
2. Normality of Residuals
3. Homoscedasticity
4. No Missing Features
5. No Multicollinearity

Normality of Residuals

Normality of residuals : residuals ต้องมีการกระจายตัวแบบ normal distribution ที่มี mean เท่ากับ 0 และ variance เท่ากับ σ^2 ($\varepsilon \sim N(0, \sigma^2)$)

Normality of Residuals

Residuals คือ ผลต่างระหว่างค่าจริง (y) กับค่าพยากรณ์ ($\hat{y}$)

$$\boldsymbol{\varepsilon} = \mathbf{y} - \hat{\mathbf{y}}$$

$\boldsymbol{\varepsilon}$		$\mathbf{y}$		$\hat{\mathbf{y}}$
ε_1		y_1		$\hat{y}_1$
ε_2	$=$	y_2	$-$	$\hat{y}_2$
$\vdots$		$\vdots$		$\vdots$
ε_3		y_n		$\hat{y}_n$

Normality of Residuals

เหตุผลที่ residuals มี mean เท่ากับ 0 เพราะ $\hat{y}$ ที่ได้จาก model ต้องเป็นค่ากลางของ y

Normality of Residuals

เหตุผลที่ residuals มีการกระจายตัวแบบ normal distribution เพราะ เป็นการยืนยันว่า w ที่เราได้จากการ minimize SSE เป็น w ที่ดีที่สุด

Assumption

1. ✓ **Linear Relationship**
2. ✓ **Normality of Residuals**
3. Homoscedasticity
4. No Missing Features
5. No Multicollinearity

Homoscedasticity

Homoscedasticity : variance ของ residuals เป็นค่าคงที่ตลอดค่า $\hat{y}$

Homoscedasticity

เหตุผลที่เป็นเช่นนี้เพราะ residuals ต้องกระจายตัวแบบเดิมถึงแม้ $\hat{y}$ จะมีค่าเพิ่มขึ้น

Homoscedasticity

เหตุผลที่เป็นเช่นนี้เพราะ residuals ต้องกระจายตัวแบบเดิมถึงแม้ $\hat{y}$ จะมีค่าเพิ่มขึ้น

Assumption

1. Linear Relationship
2. Normality of Residuals
3. Homoscedasticity
4. No Missing Features
5. No Multicollinearity

Assumption

CHECK ASSUMPTION

- 01. MARKETING
- 02. INVESTMENT
- 03. SMART FARM
- 04. BIKE RENTAL
- 05. INSURANCE

Assumption

		Linear Relationship (Scatter plot)	Normality of Residuals	Homoscedasticity
MARKETING		✓	✗	✓
INVESTMENT	SET50	✓	○	✓
	EURUSD	✓	○	✓
	XAUUSD	✓	○	○
	BTCUSD	✓	○	✗

✓ = ถูกต้องตาม assumption

○ = พอจะยอมรับได้ว่าเป็นไปตาม assumption

✗ = ไม่ถูกต้องตาม assumption

Assumption

		Linear Relationship (Scatter plot)	Normality of Residuals	Homoscedasticity
SMART FARM	RICE	○	○	✕
	BANANA	✕	○	✕
	MELON	✕	○	✕
BIKE RENTAL		✓	○	○
INSURANCE	SMOKER	✓	✓	✓
	NON SMOKER	✕	✕	✕

✓ = ถูกต้องตาม assumption

○ = พอจะยอมรับได้ว่าเป็นไปตาม assumption

✕ = ไม่ถูกต้องตาม assumption

Assumption

Note

Q : ถ้าทั้ง 3 ข้อนี้ไม่เป็นจริง ต้องทำอย่างไร?

- ☒ Scatter plot ระหว่าง $\hat{y}$ กับ y
- ☒ Assumption ข้อที่ 2
- ☒ Assumption ข้อที่ 3

Assumption

Note

Assumption

Note

Assumption

Note

Nonlinear Transformation

Consider Other Models such as Neural Network

Further reading :
https://people.revoledu.com/kardi/tutorial/Regression/nonlinear/NonLinearTransformation.htm

No Missing Features

No missing features : มีตัวแปรต้นที่ใช้ในการพยากรณ์ $\hat{y}$ อย่างครบถ้วน

No Missing Features

เหตุที่เป็นเช่นนี้ อธิบายได้ด้วยสมการดังต่อไปนี้

$$\boxed{y = \hat{y} + \varepsilon}$$

และ

$$\boxed{\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p}$$

No Missing Features

สมมติให้ ตัวแปรต้นที่เกี่ยวข้องกับการพยากรณ์ $\hat{y}$ มี 4 ตัว ได้แก่ x_1, x_2, x_3, x_4
และตัวแปรต้นที่เราใช้สร้าง model มี 4 ตัว ได้แก่ x_1, x_2, x_3, x_4

$$y = \hat{y} + \varepsilon$$

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

No Missing Features

สมมติให้ ตัวแปรต้นที่เกี่ยวข้องกับการพยากรณ์ $\hat{y}$ มี 4 ตัว ได้แก่ x_1, x_2, x_3, x_4
และตัวแปรต้นที่เราใช้สร้าง model มี 3 ตัว ได้แก่ x_1, x_2, x_4

$$y = \hat{y} + \varepsilon$$

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

No Missing Features

สมมติให้ ตัวแปรต้นที่เกี่ยวข้องกับการพยากรณ์ $\hat{y}$ มี 4 ตัว ได้แก่ x_1, x_2, x_3, x_4
และตัวแปรต้นที่เราใช้สร้าง model มี 2 ตัว ได้แก่ x_1, x_2

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

y_hat

$$y = \hat{y} + \varepsilon$$

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

Assumption

- ✓ 1. Linear Relationship
- ✓ 2. Normality of Residuals
- ✓ 3. Homoscedasticity
- ✓ 4. No Missing Features
- 5. No Multicollinearity

No Multicollinearity

No Multicollinearity : ตัวแปรต้นต้องไม่มีความสัมพันธ์เชิงเส้นต่อกัน (ไม่สามารถใช้ตัวแปรต้นเพื่อพยากรณ์ตัวแปรต้นตัวอื่นได้)

<u>ตัวอย่าง (1)</u>

x_1	x_2	x_3	x_4
-5	-9	-11	-12
-6	19	5	-4
-1	1	-3	-6
8	2	1	15

No Multicollinearity

ตัวอย่าง (1)

x_1	x_2	x_3	x_4
-5	-9	-11	-12
-6	19	5	-4
-1	1	-3	-6
8	2	1	15

No Multicollinearity

ตัวอย่าง (1)

x_4
-12
-4
-6
15

$=$

$\hat{x}_4$
-12
-4
-6
15

$= 2$

x_1
-5
-6
-1
8

$-$

x_2
-9
19
1
2

$+$

x_3
-11
5
-3
1

No Multicollinearity

<u>ตัวอย่าง (1)</u>

$\mathbf{x_4}$
-12
-4
-6
15

$=$

$\hat{\mathbf{x}}_4$
-12
-4
-6
15

"Perfect Multicollinearity"
or
"Linearly Dependent"

No Multicollinearity

ตัวอย่าง (1)

x_1	x_2	x_3	x_4
-5	-9	-11	-12
-6	19	5	-4
-1	1	-3	-6
8	2	1	15

No Multicollinearity

<u>ตัวอย่าง (1)</u>

x_1	x_2	x_3
-5	-9	-11
-6	19	5
-1	1	-3
8	2	1

No Multicollinearity

<u>ตัวอย่าง (2)</u>

x_1	x_2	x_3	x_4
-2	3	6	4
1	2	5	1
3	-1	-2	2
-1	-3	1	5

No Multicollinearity

<u>ตัวอย่าง (2)</u>

$\mathbf{x_3}$
6
5
−2
1

$\approx$

$\hat{\mathbf{x}}_3$
7.04
2.75
−0.84
0.19

$= -0.34$

$\mathbf{x_1}$
−2
1
3
−1

$+1.20$

$\mathbf{x_2}$
3
2
−1
−3

$+0.69$

$\mathbf{x_4}$
4
1
2
5

No Multicollinearity

<u>ตัวอย่าง (2)</u>

$\mathbf{x_3}$		$\hat{\mathbf{x}}_3$
6		7.04
5	$\approx$	2.75
-2		-0.84
1		0.19

$$R^2_{\mathbf{x}_3} = 0.80$$

"Multicollinearity"

No Multicollinearity

เกณฑ์การวัดค่า R^2

$R^2 > 0.6$
$R^2 > 0.8$
$R^2 > 0.9$

No Multicollinearity

<u>ตัวอย่าง (2)</u>

x_1	x_2	x_3	x_4
-2	3	6	4
1	2	5	1
3	-1	-2	2
-1	-3	1	5

No Multicollinearity

<u>ตัวอย่าง (2)</u>

x_1	x_2	x_4
-2	3	4
1	2	1
3	-1	2
-1	-3	5

No Multicollinearity

เครื่องมือทางคณิตศาสตร์ที่ใช้ตรวจสอบความสัมพันธ์เชิงเส้น
ของตัวแปรต้นมีชื่อว่า **Variance Inflation Factor (VIF)**

Feature Reading : Variance Inflation Factor (VIF)

No Multicollinearity

Note

“ ในงานวิจัยสมัยใหม่ โดยเฉพาะด้าน neural network หรือ deep learning (ซึ่งมีรากฐานมาจาก multiple regression โดยตรง) การที่เราจะรักษา assumption ข้อนี้ไว้ แทบจะเป็นไปไม่ได้เลย ”

No Multicollinearity

Note

เพราะ feature ในโลกความจริงนั้นมีความขึ้นต่อกันสูง ยกตัวอย่างเช่น

1. ข้อมูลรูปภาพ

Ref : https://en.wikipedia.org/wiki/Digital_image_processing#/media/File:Lotus_free.jpg

No Multicollinearity

Note

2. Feature ที่เกิดจากการทำ one hot encoding (ซึ่งก่อให้เกิด linearly dependent แน่นอน)

x_0	x_1	x_2	x_3
1	0	1	0
1	0	0	1
1	1	0	0
1	1	0	0

No Multicollinearity

Note

3. Feature ที่เกิดจากการ encode ในลักษณะที่คล้ายกับ one hot encoding เช่น count vectorization สำหรับข้อมูลที่เป็นภาษา

	and	document	first	is	one	second	the	third	this
This is the first document	0	1	1	1	0	0	1	0	1
This document is the second document	0	2	0	1	0	1	1	0	1
And this is the third one.	1	0	0	1	1	0	1	1	1
Is this the first document	0	1	1	1	0	0	1	0	1

No Multicollinearity

Q : ถ้า feature ที่ใช้มี multicollinearity จะส่งผลอย่างไร?

No Multicollinearity

- **Effect of Multicollinearity**
- **Effect of Linearly Dependent**

No Multicollinearity

- **Effect of Multicollinearity**
  - ✓ ไม่สามารถวิเคราะห์ feature importance ด้วย p-value ได้
  - ✓ ไม่ส่งผลกระทบต่อ performance ของ model

No Multicollinearity

- **Effect of Linearly Dependent**
  - ✓ ไม่สามารถวิเคราะห์ feature importance ด้วย p-value ได้
  - ✗ มีโอกาสสูงที่จะส่งผลกระทบต่อ performance ของ model

Assumption

- ✓ 1. Linear Relationship
- ✓ 2. Normality of Residuals
- ✓ 3. Homoscedasticity
- ✓ 4. No Missing Features
- ✓ 5. No Multicollinearity

Model Improvement

- **Assumption**
- Problem with Linearly Dependent
- Solution
- Regularization

Problem with Linearly Dependent

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Problem with Linearly Dependent

- ถ้า X_b เป็น linearly dependent

$$X_b = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$x_0 = x_1 + x_2 + x_3$$

Problem with Linearly Dependent

แล้ว $X_b^T X_b$ เป็น linearly dependent

$$X_b^T X_b = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Problem with Linearly Dependent

- ถ้า $X_b^T X_b$ เป็น linearly dependent

$$X_b^T X_b = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$x_0 = x_1 + x_2 + x_3$$

Problem with Linearly Dependent

แล้ว $X_b^T X_b$ ไม่สามารถหา inverse ได้

Invertible Matrix Theorem. *Let A be an $n \times n$ matrix, and let $T : \mathbf{R}^n \to \mathbf{R}^n$ be the matrix transformation $T(x) = Ax$. The following statements are equivalent:*

1. *A is invertible.*
2. *A has n pivots.*
3. $\mathrm{Nul}(A) = \{0\}$.
4. *The columns of A are linearly independent.*
5. *The columns of A span $\mathbf{R}^n$.*
6. *$Ax = b$ has a unique solution for each b in $\mathbf{R}^n$.*
7. *T is invertible.*
8. *T is one-to-one.*
9. *T is onto.*

Ref : https://textbooks.math.gatech.edu/ila/invertible-matrix-thm.html

Problem with Linearly Dependent

ดังนั้น เราจะไม่สามารถคำนวณ normal equation ได้

Normal Equation

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

Problem with Linearly Dependent

Q : ถ้า $X_b^T X_b$ inverse ไม่ได้ แปลว่าไม่มีคำตอบรึเปล่า?

Normal Equation

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

Problem with Linearly Dependent

A : มีคำตอบ และคำตอบมีมากมายมหาศาล

Problem with Linearly Dependent

กราฟนี้มาอย่างไร?

Problem with Linearly Dependent

ตัวอย่างการหาสมการร่องคำตอบ

x_1	x_2	x_3	y
0	1	0	3.8
0	0	1	5.1
1	0	0	3
0	1	0	4.2
0	0	1	4.9

ตารางแสดง dataset ที่ feature linearly dependent

Problem with Linearly Dependent

Normal Equation

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

$$\left(X_b^T X_b\right)\mathbf{w} = X_b^T \mathbf{y}$$

Problem with Linearly Dependent

$$\boxed{(X_b^T X_b)\mathbf{w} = X_b^T \mathbf{y}}$$

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Problem with Linearly Dependent

$$\boxed{(X_b^T X_b)\mathbf{w} = X_b^T \mathbf{y}}$$

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{y}$
0	1	0	3.8
0	0	1	5.1
1	0	0	3
0	1	0	4.2
0	0	1	4.9

$$X_b = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 3.8 \\ 5.1 \\ 3 \\ 4.2 \\ 4.9 \end{bmatrix}$$

Problem with Linearly Dependent

เราสามารถคำนวณ $X_b^T X_b$ และ $X_b^T \mathbf{y}$ ได้ดังนี้

- $X_b^T X_b = \begin{bmatrix} 5 & 1 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}$

- $X_b^T \mathbf{y} = \begin{bmatrix} 21 \\ 3 \\ 8 \\ 10 \end{bmatrix}$

Problem with Linearly Dependent

จาก normal equation เราจะได้ว่า

$$\left(X_b^T X_b\right)\mathbf{w} = X_b^T \mathbf{y}$$

$$\begin{bmatrix} 5 & 1 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 21 \\ 3 \\ 8 \\ 10 \end{bmatrix}$$

$$\begin{bmatrix} 5w_0 + w_1 + 2w_2 + 2w_3 \\ w_0 + w_1 \\ 2w_0 + 2w_2 \\ 2w_0 + 2w_3 \end{bmatrix} = \begin{bmatrix} 21 \\ 3 \\ 8 \\ 10 \end{bmatrix}$$

Problem with Linearly Dependent

จากนั้นทำการแก้สมการหลายตัวแปร

$$5w_0 + w_1 + 2w_2 + 2w_3 = 21$$
$$w_0 + w_1 = 3$$
$$2w_0 + 2w_2 = 8$$
$$2w_0 + 2w_3 = 10$$

เราจะได้สมการของคำตอบเป็น

$$w_0 = 5 - w_3$$
$$w_1 = -2 + w_3$$
$$w_2 = -1 + w_3$$
$$w_3 \in \mathbb{R}$$

Problem with Linearly Dependent

ซึ่งจะสามารถ plot เป็นกราฟได้ดังนี้

สมการร่องคำตอบ

$$w_0 = 5 - w_3$$
$$w_1 = -2 + w_3$$
$$w_2 = -1 + w_3$$
$$w_3 \in \mathbb{R}$$

Problem with Linearly Dependent

Q : แล้วเราจะหาคำตอบได้อย่างไร?
ในเมื่อ $X_b^T X_b$ ไม่สามารถหา inverse ได้

Normal Equation

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

Problem with Linearly Dependent

A : เราสามารถใช้ pseudo inverse ในการหา inverse ของ $X_b^T X_b$ แทนได้

Normal Equation

$$\mathbf{w} = \left(X_b^T X_b\right)^{-1} X_b^T \mathbf{y}$$

Problem with Linearly Dependent

ซึ่ง pseudo inverse ที่เราจะใช้มีชื่อเรียกว่า

“ Moore-Penrose ”

Note : sklearn ใช้ Moore-Penrose เช่นกัน

Problem with Linearly Dependent

“ จากประสบการณ์ทำงานจริง เมื่อ X_b มี feature ที่เป็น linearly dependent จำนวนมาก จะทำให้การหาคำตอบด้วย Moore-Penrose มีปัญหา และนำมาซึ่ง**คำตอบที่มีขนาดมหึมา** ”

Problem with Linearly Dependent

บางครั้งการฝืนธรรมชาติของ Moore-Penrose นำมาซึ่ง**คำตอบที่มีขนาดมหึมา**

```python
for coef in reg.coef_:
    print(coef)
```

```
-0.4229530799092025
-3727805854.664848
-3727880650.457581
-7401406299.58381
-12440201347.051556
-12439757389.52999
-12439097475.793343
-12438372051.419
-12439847778.274954
⋮
```

รูปแสดงค่า weight ของ dataset car price

Problem with Linearly Dependent

การที่**คำตอบมีขนาดมหึมา** จะทำให้

“performance ของ model แย่ เมื่อเจอกับ unseen data”

กราฟแสดงข้อมูลระหว่างค่าจริง และค่าพยากรณ์
บน training set ของ car price

กราฟแสดงข้อมูลระหว่างค่าจริง และค่าพยากรณ์
บน test set ของ car price

Problem with Linearly Dependent

<u>ตัวอย่าง</u>

x_1	x_2	x_3	y
0	1	0	3.8
0	0	1	5.1
1	0	0	3
0	1	0	4.2
0	0	1	4.9

ตารางแสดง dataset ที่ feature linearly dependent

Problem with Linearly Dependent

<u>ตัวอย่าง</u>

$$w_0 = 5 - w_3$$
$$w_1 = -2 + w_3$$
$$w_2 = -1 + w_3$$
$$w_3 \in \mathbb{R}$$

Problem with Linearly Dependent

<u>ตัวอย่าง</u>

ให้ $w_3 = 1{,}000{,}000$

จะได้ $w_0 = 5 - 1{,}000{,}000 = -999{,}995$

$w_1 = -2 + 1{,}000{,}000 = 999{,}998$

$w_2 = -1 + 1{,}000{,}000 = 999{,}999$

Problem with Linearly Dependent

<u>ตัวอย่าง</u>

$$\hat{y} = -999{,}995 + 999{,}998\, x_1 + 999{,}999\, x_2 + 1{,}000{,}000\, x_3$$

- Training set

x_1	x_2	x_3	y	$\hat{y}$
0	1	0	3.8	4
0	0	1	5.1	5
1	0	0	3	3
0	1	0	4.2	4
0	0	1	4.9	5

Problem with Linearly Dependent

<u>ตัวอย่าง</u>

$$\hat{y} = -999{,}995 + 999{,}998\, x_1 + 999{,}999\, x_2 + 1{,}000{,}000\, x_3$$

- Test set

x_1	x_2	x_3	y	$\hat{y}$
0	0	0	1	$-999{,}995$

"performance ของ model แย่
เมื่อเจอกับ unseen data"

Problem with Linearly Dependent

Conclusion

X_b is linearly dependent

↓

$X_b^T X_b$ is linearly dependent

↓

$X_b^T X_b$ is non-invertible

↓

Moore-Penrose pseudo inverse

lot of linearly dependent features

↘ ↙

tremendous solution

↓

terrible performance on unseen data

Model Improvement

- ✓ **Assumption**
- ✓ **Problem with Linearly Dependent**
- Solution
- Regularization

Solution

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Solution

เพื่อแก้ปัญหานี้ นักคณิตศาสตร์จึงเกิดแนวคิดว่า

" อยากหาคำตอบ (w) บนร่องนี้ที่มีขนาดเล็กที่สุด "

Solution

$$\sum_{j=0}^{p} w_j$$

$$\sum_{j=0}^{p} w_j^2$$

$$\sum_{j=0}^{p} |w_j|$$

Solution

$$\sum_{j=0}^{p} w_j$$

$$\sum_{j=0}^{p} w_j^2$$

$$\sum_{j=0}^{p} |w_j|$$

Solution

“ คำตอบ (w) บนร่องที่มีขนาดเล็กที่สุด ”

Solution

“ คำตอบ (w) บนร่องที่ $\sum_{j=0}^{p} w_j^2$ มีค่าน้อยที่สุด ”

Solution

คณิตศาสตร์ที่เราใช้แก้ปัญหานี้คือ **Lagrange Multipliers**

Solution

minimize $\sum_{j=0}^{p} w_j^2$

subject to สมการร่องคำตอบ

Solution

<u>ตัวอย่าง</u>

x_1	x_2	x_3	y
0	1	0	4
0	0	1	5
1	0	0	3

ตารางแสดง dataset ที่ feature linearly dependent

Solution

<u>ตัวอย่าง</u>

$$w_0 = 5 - w_3$$
$$w_1 = -2 + w_3$$
$$w_2 = -1 + w_3$$
$$w_3 \in \mathbb{R}$$

Solution

<u>ตัวอย่าง</u>

$$\textbf{minimize} \quad \sum_{j=0}^{p} w_j^2$$

$$\textbf{subject to} \quad w_0 = 5 - w_3$$

$$w_1 = -2 + w_3$$

$$w_2 = -1 + w_3$$

Solution

<u>ตัวอย่าง</u>

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

Solution

<u>ตัวอย่าง</u>

$$\hat{y} = 3 + 0x_1 + x_2 + 2x_3$$

- Training set

x_1	x_2	x_3	y	$\hat{y}$
0	1	0	4	4
0	0	1	5	5
1	0	0	3	3

Solution

<u>ตัวอย่าง</u>

$$\hat{y} = 3 + 0x_1 + x_2 + 2x_3$$

- Test set

x_1	x_2	x_3	y	$\hat{y}$
0	0	0	1	3

Solution

วิธีนี้แก้ปัญหาได้อย่างที่เราต้องการ แต่ก็แลกมาด้วยความซับซ้อน

$(X_b^T X_b)\mathbf{w} = X_b^T \mathbf{y}$ → Reduce echelon → สมการของร่อง → Lagrange multipliers → คำตอบ ($\mathbf{w}$) ที่มีขนาดเล็กที่สุด

Solution

นักคณิตศาสตร์จึงเกิดแนวคิดต่อยอดว่า

“ อยากได้วิธีการที่ซับซ้อนน้อยกว่านี้ เพื่อให้ได้คำตอบ (w) ที่ไม่มหึมาก็พอ ”

Solution

“คำตอบ (w) ที่ไม่มหึมา”

Solution

$$\left(X_b^T X_b\right)\mathbf{w} = X_b^T \mathbf{y}$$

+

reduce echelon

+

สมการของร่อง

+

Lagrange multipliers

minimize $\sum_{j=0}^{p} w_j^2$

subject to $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = c$

Solution

minimize $\sum_{j=0}^{p} w_j^2$

subject to $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = c$

“คำตอบ (w) ที่ไม่มหึมา”

Solution

minimize $\sum_{j=0}^{p} w_j^2$

subject to $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = c$

$\equiv$

minimize $Cost\ (\nabla Cost = 0)$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$

Solution

Derivation of Equivalent

Open File

Derive_Equivalent.pdf

Solution

minimize $Cost\ (\nabla Cost = 0)$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$

“คำตอบ (w) ที่ไม่มหึมา”

Solution

minimize $Cost\ (\nabla Cost = 0)$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$

Solution

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

$$\lambda = 0.001$$

=

Solution

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

$\lambda = 1$

=

Solution

Q : ไปเปลี่ยนคำตอบแบบนี้
แล้ว performance จะดีหรอ**?**

Solution

ในปี 1970 นักคณิตศาสตร์ ได้ค้นพบสมบัติอันยิ่งใหญ่ของการ minimize

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

Ref : Ridge Regression : Biased Estimation for Nonorthogonal Problems

Solution

" นักคณิตศาสตร์ค้นพบว่า การใส่ $\lambda > 0$ จะทำให้ error ของ model ลดลงกว่าการไม่ใส่ λ ($\lambda = 0$) สำหรับ data ทุกประเภท "

*ยกเว้น $y = \hat{y}$ (model ไม่มี error)

Solution

<u>ตัวอย่าง</u>

x_1	x_2	x_3	y
0	1	0	3.8
0	0	1	5.1
1	0	0	3
0	1	0	4.2
0	0	1	4.9

ตารางแสดง dataset ที่ feature linearly dependent

Solution

<u>ตัวอย่าง</u>

minimize $Cost\ (\nabla Cost = 0)$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$

Solution

<u>ตัวอย่าง</u>

กำหนดให้ $\lambda = 10^{-16}$

ดังนั้น เราจะได้

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 0.5 \\ 1.5 \\ 2.5 \end{bmatrix}$$

Solution

<u>ตัวอย่าง</u>

$$\hat{y} = 2.5 + 0.5x_1 + 1.5x_2 + 2.5x_3$$

- Training set

x_1	x_2	x_3	y	$\hat{y}$
0	1	0	3.8	4
0	0	1	5.1	5
1	0	0	3	3
0	1	0	4.2	4
0	0	1	4.9	5

Solution

<u>ตัวอย่าง</u>

$$\hat{y} = 2.5 + 0.5x_1 + 1.5x_2 + 2.5x_3$$

- Test set

x_1	x_2	x_3	y	$\hat{y}$
0	0	0	1	2.5

Solution

“ Ridge Regression ”

Solution

minimize $\sum_{j=0}^{p} w_j^2$

subject to สมการของร่อง

minimize $\sum_{j=0}^{p} w_j^2$

subject to $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = c$

minimize $Cost\ (\nabla Cost = 0)$

โดยที่ $Cost = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$

Ridge Regression

Model Improvement

- ✓ **Assumption**
- ✓ **Problem with Linearly Dependent**
- ✓ **Solution**
- Regularization

Regularization

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Regularization

- What is Regularization?
- Ridge Regression
- Lasso Regression
- Elastic Net
- Conclusion

What is Regularization?

Regularization คือ วิธีการสร้าง model ให้มีความ general มากขึ้น (ลดปัญหา overfitting)

Ref: https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning

What is Regularization?

“ ช่วยแก้ปัญหา performance แย่ เมื่อเจอกับ unseen data ทำให้ model มี performance ดีขึ้นสำหรับ data ทุกประเภท (linearly dependent, multicollinearity, no multicollinearity) ”

What is Regularization?

สรุปแบบรวบรัดสำหรับผู้ที่ข้ามเนื้อหา Problem & Solution

Linearly dependent → มีโอกาสสูงที่ performance ของ model จะไม่ดี

↓

Ridge Regression
(Regularization)

What is Regularization?

Ridge Regression

Lasso Regression

Elastic Net

Regularization

- **What is Regularization?**
- Ridge Regression
- Lasso Regression
- Elastic Net
- Conclusion

Ridge Regression

- What is Ridge Regression?
- Geometric View
- Properties
- Model Creation
- How to find Lambda
- Code

What is Ridge Regression?

Ridge regression คือ การทำ regularization โดยมี cost function เป็น

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

Ridge Regression

- **What is Ridge Regression?**
- Geometric View
- Properties
- Model Creation
- How to find Lambda
- Code

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

facebook/tautologyai
www.tautology.live

Geometric View

Linearly Dependent

No Multicollinearity

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

https://www.geogebra.org/3d/mzbky4wt

+

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

$\lambda = 0.01$

=

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

$\lambda = 0.1$

=

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

=

$\lambda = 1$

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

=

$\lambda = 10$

Geometric View

Linearly Dependent

- จุดต่ำสุดใหม่ จะอยู่ระหว่างสมการร่องคำตอบ และจุดกำเนิด
- คำตอบจะเปลี่ยนจากสมการร่องคำตอบ เป็นจุดต่ำสุดจุดเดียว
- ยิ่ง λ มาก จุดต่ำสุดใหม่ยิ่งใกล้จุดกำเนิด
- ยิ่ง λ น้อย จุดต่ำสุดใหม่ยิ่งใกล้สมการร่องคำตอบ

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

https://www.geogebra.org/3d/mzbky4wt

+

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

$\lambda = 0.01$

=

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

=

$\lambda = 0.1$

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

=

$\lambda = 1$

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

=

$\lambda = 10$

Geometric View

No Multicollinearity

- จุดต่ำสุดใหม่ จะอยู่ระหว่างจุดต่ำสุดเดิม และจุดกำเนิด
- ยิ่ง λ มาก จุดต่ำสุดใหม่ยิ่งใกล้จุดกำเนิด
- ยิ่ง λ น้อย จุดต่ำสุดใหม่ยิ่งใกล้จุดต่ำสุดเดิม

Ridge Regression

- ✓ **What is Ridge Regression?**
- ✓ **Geometric View**
- Properties
- Model Creation
- How to find Lambda
- Code

Properties

" การใส่ $\lambda > 0$ จะทำให้ error ของ model ลดลงกว่าการไม่ใส่ λ ($\lambda = 0$) สำหรับ data ทุกประเภท "

*ยกเว้น $y = \hat{y}$ (model ไม่มี error)

Ridge Regression

- ✓ **What is Ridge Regression?**
- ✓ **Geometric View**
- ✓ **Properties**
- Model Creation
- How to find Lambda
- Code

Model Creation

$$\nabla Cost = 0$$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$

Model Creation

$$\mathbf{w} = \left(X_b^T X_b + \lambda \mathbf{I}\right)^{-1} X_b^T \mathbf{y}$$

Model Creation

Derivation of Ridge Regression

Open File
Derive_Ridge.pdf

Ridge Regression

- ✓ **What is Ridge Regression?**
- ✓ **Geometric View**
- ✓ **Properties**
- ✓ **Model Creation**
  - How to find Lambda
  - Code

How to find Lambda

Q : แล้วเราจะหา λ ที่ดีที่สุดได้อย่างไร**?**

How to find Lambda

A : ใช้ Cross Validation

How to find Lambda

λ	Fold					Mean	Rank
	1	2	3	4	5		
$\lambda = 0.01$	0.78	0.77	0.72	0.80	0.78	0.77	2
$\lambda = 0.1$	0.80	0.88	0.78	0.72	0.80	0.78	1
$\lambda = 1$	0.74	0.72	0.76	0.75	0.78	0.75	4
$\lambda = 10$	0.82	0.74	0.74	0.74	0.70	0.76	3

How to find Lambda

For more information

Cross Validation

Ridge Regression

- ✓ What is Ridge Regression?
- ✓ Geometric View
- ✓ Properties
- ✓ Model Creation
- ✓ How to find Lambda
- Code

Code

ตัวอย่าง code สำหรับ ridge regression

x_1	x_2	y
0	1	3.9
2	1	7.7
1	1	6.2
2	0	5.2
3	1	9.8

Code

- เก็บค่า λ ที่ต้องการพิจารณาไว้ในตัวแปรชื่อ alphas

```
alphas = np.logspace(-3, 3, num=100)
```

Code

- Code สำหรับสร้าง model จากข้อมูลของเราโดยที่

$$X = \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 1 & 1 \\ 2 & 0 \\ 3 & 1 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 3.9 \\ 7.7 \\ 6.2 \\ 5.2 \\ 4.8 \end{bmatrix}$$

```
reg = RidgeCV(alphas=alphas, cv=5)
reg.fit(X, y)
```

Code

- ค่า λ ที่ดีที่สุด ถูกเก็บไว้ใน attribute ชื่อ `alpha_`

```
reg.alpha_
```

Code

- ค่า w_0 ถูกเก็บไว้ใน attribute ชื่อ `intercept_`

```
reg.intercept_
```

Code

- ค่า $w_1, \dots, w_p$ ถูกเก็บไว้ใน attribute ชื่อ coef_

```
reg.coef_
```

Code

Code for ridge regression

Open File
Regularization.ipynb

Ridge Regression

- ✓ What is Ridge Regression?
- ✓ Geometric View
- ✓ Properties
- ✓ Model Creation
- ✓ How to find Lambda
- ✓ Code

Regularization

- ✓ **What is Regularization?**
- ✓ **Ridge Regression**
- Lasso Regression
- Elastic Net
- Conclusion

Lasso Regression

- What is Lasso Regression?
- Geometric View
- Properties
- Model Creation
- How to find Lambda
- Code

What is Lasso Regression?

Lasso regression คือ การทำ regularization โดยมี cost function เป็น

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

Lasso Regression

- **What is Lasso Regression?**
- Geometric View
- Properties
- Model Creation
- How to find Lambda
- Code

Geometric View

Linearly Dependent

No Multicollinearity

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

+

https://www.geogebra.org/m/c87cckmk

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

$\lambda = 0.01$

=

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

$\lambda = 0.1$

=

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

=

$\lambda = 1$

Geometric View

Linearly Dependent

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

=

$\lambda = 10$

Geometric View

Linearly Dependent

- จุดต่ำสุดใหม่ จะอยู่ระหว่างสมการร่องคำตอบ และจุดกำเนิด
- คำตอบจะเปลี่ยนจากสมการร่องคำตอบ เป็นจุดต่ำสุดจุดเดียว
- ยิ่ง λ มาก จุดต่ำสุดใหม่ยิ่งใกล้จุดกำเนิด
- ยิ่ง λ น้อย จุดต่ำสุดใหม่ยิ่งใกล้สมการร่องคำตอบ

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

https://www.geogebra.org/m/c87cckmk

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

$\lambda = 0.01$

=

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

$\lambda = 0.1$

=

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

$\lambda = 1$

=

Geometric View

No Multicollinearity

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$$

$\lambda = 10$

=

Geometric View

No Multicollinearity

- จุดต่ำสุดใหม่ จะอยู่ระหว่างจุดต่ำสุดเดิม และจุดกำเนิด
- ยิ่ง λ มาก จุดต่ำสุดใหม่ยิ่งใกล้จุดกำเนิด
- ยิ่ง λ น้อย จุดต่ำสุดใหม่ยิ่งใกล้จุดต่ำสุดเดิม

Lasso Regression

- ✓ **What is Lasso Regression?**
- ✓ **Geometric View**
- Properties
- Model Creation
- How to find Lambda
- Code

Properties

1. Generalization
2. Feature Selection

Properties

1. Generalization

" การใส่ $\lambda > 0$ จะทำให้ error ของ model ลดลงกว่าการไม่ใส่ λ ($\lambda = 0$) สำหรับ data ทุกประเภท "

*ยกเว้น $y = \hat{y}$ (model ไม่มี error)

Properties

2. Feature Selection

https://www.geogebra.org/m/c87cckmk

Properties

2. Feature Selection

Properties

2. Feature Selection

https://www.geogebra.org/3d/mzbky4wt

Properties

2. Feature Selection

Lasso Regression

- ✓ **What is Lasso Regression?**
- ✓ **Geometric View**
- ✓ **Properties**
- Model Creation
- How to find Lambda
- Code

Model Creation

$$\nabla Cost = 0$$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p}|w_j|$

Model Creation

$$X_b^T X_b \mathbf{w} + \lambda \nabla |\mathbf{w}| = X_b^T \mathbf{y}$$

Model Creation

Derivation of Lasso Regression

Open File
Derive_Lasso.pdf

Model Creation

$$X_b^T X_b \mathbf{w} + \lambda \nabla |\mathbf{w}| = X_b^T \mathbf{y}$$

Model Creation

Gradient Descent

Coordinate Descent

Lasso Regression

- ✓ **What is Lasso Regression?**
- ✓ **Geometric View**
- ✓ **Properties**
- ✓ **Model Creation**
  - How to find Lambda
  - Code

How to find Lambda

Q : แล้วเราจะหา λ ที่ดีที่สุดได้อย่างไร?

How to find Lambda

A : ใช้ Cross Validation

How to find Lambda

λ	Fold					Mean	Rank
	1	2	3	4	5		
$\lambda = 0.01$	0.78	0.77	0.72	0.80	0.78	0.77	2
$\lambda = 0.1$	0.80	0.88	0.78	0.72	0.80	0.78	1
$\lambda = 1$	0.74	0.72	0.76	0.75	0.78	0.75	4
$\lambda = 10$	0.82	0.74	0.74	0.74	0.70	0.76	3

How to find Lambda

For more information

Cross Validation

Lasso Regression

- ✓ What is Lasso Regression?
- ✓ Geometric View
- ✓ Properties
- ✓ Model Creation
- ✓ How to find Lambda
- Code

Code

ตัวอย่าง code สำหรับ lasso regression

x_1	x_2	y
0	1	3.9
2	1	7.7
1	1	6.2
2	0	5.2
3	1	9.8

Code

- Code สำหรับสร้าง model จากข้อมูลของเราโดยที่

$$X = \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 1 & 1 \\ 2 & 0 \\ 3 & 1 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 3.9 \\ 7.7 \\ 6.2 \\ 5.2 \\ 4.8 \end{bmatrix}$$

```
reg = RidgeCV(alphas=alphas, cv=5)
reg.fit(X, y)
```

Code

- ค่า λ ที่ดีที่สุด ถูกเก็บไว้ใน attribute ชื่อ `alpha_`

```
reg.alpha_
```

Code

- ค่า w_0 ถูกเก็บไว้ใน attribute ชื่อ `intercept_`

```
reg.intercept_
```

Code

- ค่า $w_1, \dots, w_p$ ถูกเก็บไว้ใน attribute ชื่อ coef_

```
reg.coef_
```

Code

Code for lasso regression

Open File
Regularization.ipynb

Lasso Regression

- ✓ What is Lasso Regression?
- ✓ Geometric View
- ✓ Properties
- ✓ Model Creation
- ✓ How to find Lambda
- ✓ Code

Regularization

- ✓ **What is Regularization?**
- ✓ **Ridge Regression**
- ✓ **Lasso Regression**
- Elastic Net
- Conclusion

Elastic Net

- What is Elastic Net?
- Geometric View
- Properties
- Model Creation
- How to find Lambda & $l1_{ratio}$
- Code

What is Elastic Net?

Elastic Net คือ การทำ regularization โดยมี cost function เป็น

$$Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda(l1_{ratio})\sum_{j=0}^{p}|w_j| + \lambda(1 - l1_{ratio})\sum_{j=0}^{p} w_j^2$$

Elastic Net

- ✓ **What is Elastic Net?**
- Geometric View
- Properties
- Model Creation
- How to find Lambda & $l1_{ratio}$
- Code

Geometric View

$l1_{ratio} = 1$

$l1_{ratio} = 0.75$

$l1_{ratio} = 0.50$

Geometric View

$l1_{ratio} = 0.25$

$l1_{ratio} = 0.10$

$l1_{ratio} = 0.01$

Geometric View

Linearly Dependent

- จุดต่ำสุดใหม่ จะอยู่ระหว่างสมการร่องคำตอบ และจุดกำเนิด
- คำตอบจะเปลี่ยนจากสมการของร่อง เป็นจุดต่ำสุดจุดเดียว
- ยิ่ง λ มาก จุดต่ำสุดใหม่ยิ่งใกล้จุดกำเนิด
- ยิ่ง λ น้อย จุดต่ำสุดใหม่ยิ่งใกล้สมการร่องคำตอบ

Geometric View

No Multicollinearity

- จุดต่ำสุดใหม่ จะอยู่ระหว่างจุดต่ำสุดเดิม และจุดกำเนิด
- ยิ่ง λ มาก จุดต่ำสุดใหม่ยิ่งใกล้จุดกำเนิด
- ยิ่ง λ น้อย จุดต่ำสุดใหม่ยิ่งใกล้จุดต่ำสุดเดิม

Elastic Net

- ✓ **What is Elastic Net?**
- ✓ **Geometric View**
- Properties
- Model Creation
- How to find Lambda & $l1_{ratio}$
- Code

Properties

1. Generalization
2. Soft Feature Selection

Properties

1. Generalization

" การใส่ $\lambda > 0$ จะทำให้ error ของ model ลดลงกว่าการไม่ใส่ λ ($\lambda = 0$) สำหรับ data ทุกประเภท "

*ยกเว้น $y = \hat{y}$ (model ไม่มี error)

Properties

2. Soft Feature Selection

- $l1_{ratio} = 1$

https://www.geogebra.org/3d/pwcw4ay5

Properties

2. Soft Feature Selection

- $l1_{ratio} = 0.75$

Properties

2. Soft Feature Selection

- $l1_{ratio} = 0.50$

Properties

2. Soft Feature Selection
   - $l1_{ratio} = 0.25$

Properties

2. Soft Feature Selection

- $l1_{ratio} = 0.10$

Properties

2. Soft Feature Selection
   - $l1_{ratio} = 0.01$

Elastic Net

- ✓ **What is Elastic Net?**
- ✓ **Geometric View**
- ✓ **Properties**
- Model Creation
- How to find Lambda & $l1_{ratio}$
- Code

Model Creation

$$\nabla Cost = 0$$

โดยที่ $Cost = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda(l1_{ratio})\sum_{j=0}^{p}|w_j| + 0.5\lambda(1 - l1_{ratio})\sum_{j=0}^{p} w_j^2$

Model Creation

$$\left(X_b^T X_b + \lambda(1 - l1_{ratio})\right)\mathbf{w} + \lambda(l1_{ratio})\nabla|\mathbf{w}| = X_b^T \mathbf{y}$$

Model Creation

Derivation of Elastic net

Open File

Derive_Elastic Net.pdf

Model Creation

$$\left(X_b^T X_b + \lambda(1 - l1_{ratio})\right)\mathbf{w} + \lambda(l1_{ratio})\nabla|\mathbf{w}| = X_b^T \mathbf{y}$$

Model Creation

Gradient Descent

Coordinate Descent

Elastic Net

- ✓ **What is Elastic Net?**
- ✓ **Geometric View**
- ✓ **Properties**
- ✓ **Model Creation**
  - How to find Lambda & $l1_{ratio}$
  - Code

How to find Lambda & $l1_{ratio}$

Q : แล้วเราจะหา λ และ $l1_{ratio}$ที่ดีที่สุดได้อย่างไร**?**

How to find Lambda & $l1_{ratio}$

A : ใช้ Cross Validation

How to find Lambda & $l1_{ratio}$

$l1_{ratio}$	λ	Fold					Mean	Rank
		1	2	3	4	5		
0.1	0.1	0.78	0.77	0.72	0.80	0.78	0.77	2
	1	0.80	0.88	0.78	0.72	0.80	0.78	1
0.5	0.1	0.73	0.70	0.75	0.76	0.77	0.74	5
	1	0.80	0.76	0.74	0.73	0.71	0.748	4
0.9	0.1	0.75	0.72	0.78	0.76	0.79	0.76	3
	1	0.78	0.82	0.77	0.70	0.78	0.77	2

How to find Lambda & $l1_{ratio}$

For more information

Cross Validation

Elastic Net

- ✓ What is Elastic Net?
- ✓ Geometric View
- ✓ Properties
- ✓ Model Creation
- ✓ How to find Lambda & $l1_{ratio}$
- Code

Code

ตัวอย่าง code สำหรับ elastic net

x_1	x_2	y
0	1	3.9
2	1	7.7
1	1	6.2
2	0	5.2
3	1	9.8

Code

- Code สำหรับสร้าง model จากข้อมูลของเราโดยที่

$$X = \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 1 & 1 \\ 2 & 0 \\ 3 & 1 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 3.9 \\ 7.7 \\ 6.2 \\ 5.2 \\ 4.8 \end{bmatrix}$$

```
l1_ratio = [0.01, 0.05, 0.1, 0.5, 0.7, 0.9, 0.95, 0.99, 1]
```

```
reg = ElasticNetCV(l1_ratio=l1_ratio, n_alphas=100, cv=5)
reg.fit(X, y)
```

Code

- ค่า $l1_{ratio}$ ที่ดีที่สุด ถูกเก็บไว้ใน attribute ชื่อ `l1_ratio_`

```
reg.l1_ratio_
```

Code

- ค่า λ ที่ดีที่สุด ถูกเก็บไว้ใน attribute ชื่อ `alpha_`

```
reg.alpha_
```

Code

- ค่า w_0 ถูกเก็บไว้ใน attribute ชื่อ `intercept_`

```
reg.intercept_
```

Code

- ค่า $w_1, \dots, w_p$ ถูกเก็บไว้ใน attribute ชื่อ coef_

```
reg.coef_
```

Code

Code for lasso regression

Open File
Regularization.ipynb

Elastic Net

- ✓ What is Elastic Net?
- ✓ Geometric View
- ✓ Properties
- ✓ Model Creation
- ✓ How to find Lambda & $l1_{ratio}$
- ✓ Code

Regularization

- ✓ **What is Regularization?**
- ✓ **Ridge Regression**
- ✓ **Lasso Regression**
- ✓ **Elastic Net**
- Conclusion

Conclusion

	Normal Equation	Ridge Regression	Lasso Regression	Elastic Net
cost function	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} w_j^2$	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^{p} \lvert w_j \rvert$	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda(l1_{ratio}) \sum_{j=0}^{p} \lvert w_j \rvert + \lambda(1 - l1_{ratio}) \sum_{j=0}^{p} w_j^2$
generalization	✗	✓	✓	✓
λ	✗	✓	✓	✓
$l1_{ratio}$	✗	✗	✗	✓
hyperparameter tuning	✗	CV	CV	CV
Feature selection	✗	✗	✓	✓

Regularization

- What is Regularization?
- Ridge Regression
- Lasso Regression
- Elastic Net
- Conclusion

Model Improvement

- ✓ Assumption
- ✓ Problem with Linearly Dependent
- ✓ Solution
- ✓ Regularization

DL101 : Multiple Regression

TAUTOLOGY
INNOVATION
SCHOOL

TAUTOLOGY

ADVANCED WORKSHOP

ADVANCED WORKSHOP

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Multiple Regression Workflow

Code Pipeline

Data Preparation

Data Preparation

Feature Scaling

Code

- Feature Scaling for **training set**

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

- Feature Scaling for **test set**

```
X_test_scaled = scaler.transform(X_test)
```

Car Price

MADE BY TAUTOLOGY THAILAND
DO NOT PUBLISH WITHOUT PERMISSION

Abstract

สร้าง model เพื่อประเมินราคาของรถยนต์มือสอง โดย feature ที่นำมาใช้ คือ ข้อมูลเกี่ยวกับรถยนต์คันนั้น ๆ และรูปแบบการขาย เช่น

- ยี่ห้อรถยนต์
- เลขกิโลของรถยนต์
- ขายด้วยตัวเอง หรือ ขายผ่านนายหน้า

Why this project Important?

- สามารถประเมินราคารถยนต์ได้อย่างสมเหตุสมผลที่สุด
- สามารถนำความรู้ที่ได้จากการสร้าง model ไปประยุกต์ ใช้กับธุรกิจประเภทอื่น ๆ ที่มีลักษณะคล้ายกัน

Who this project is for?

- เจ้าของธุรกิจเต็นท์รถ
- นักประเมินราคารถยนต์
- ผู้เกี่ยวข้องกับธุรกิจที่ต้องประเมินราคา เช่น ธุรกิจโรงรับจำนำ ร้านซื้อ/ขายเครื่องดนตรี ธนาคาร
- นักวิเคราะห์ข้อมูล

Car Price Dataset

https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho

Car Price Dataset

Feature

- name : ยี่ห้อ และรุ่นของรถยนต์
- year : ปีที่รถยนต์ถูกซื้อ
- km_driven : เลขกิโลของรถยนต์
- fuel : ประเภทของน้ำมันที่รถยนต์ใช้
- seller_type : ขายด้วยตัวเอง หรือ ขายผ่านนายหน้า
- transmission : ระบบเกียร์
- owner : เป็นรถยนต์มือหนึ่ง มือสอง หรือ อื่น ๆ

Target

- selling_price : ราคารถยนต์ที่ขายได้

What we learn from this project?

Data Preparation

What we learn from this project?

Count vectorization

สร้าง feature ใหม่ โดยการหา unique word จากข้อความทั้งหมดใน dataset จากนั้นให้พิจารณาว่าแต่ละข้อความประกอบด้วย unique word อะไรบ้าง และจำนวนกี่ครั้ง

	'apple'	'green'	'is'	'kiwi'	'orange'	'red'
'Apple is red'	1	0	1	0	0	1
'Kiwi is green'	0	1	1	1	0	0
'Orange is orange'	0	0	1	0	2	0

What we learn from this project?

	con_05	con_0i	con_10	con_100	...	con_xza	con_yaris	...
Mahindra Xylo D4	0	0	0	0	...	0	0	...
Maruti Swift Dzire VDI	0	0	0	0	...	0	0	...
Mahindra KUV 100 mFALCON G80 K8 5str	0	0	0	1	...	0	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Code

- Count vectorization for **training set**

```
corpus_train = X_train['name'].tolist()
vectorizer = CountVectorizer()
vectorizer.fit(corpus_train)
cnt_vec_train = vectorizer.transform(corpus_train).toarray()
```

```
cnt_vec_feature_name = ['con_' + feature for feature in vectorizer.get_feature_names()]
```

```
X_train[cnt_vec_feature_name] = cnt_vec_train
X_train.drop('name', axis=1, inplace=True)
```

Code

- Count vectorization for **test set**

```
corpus_test = X_test['name'].tolist()
cnt_vec_test = vectorizer.transform(corpus_test).toarray()
```

```
X_test[cnt_vec_feature_name] = cnt_vec_test
X_test.drop('name', axis=1, inplace=True)
```

01. CAR PRICE

- Normal Equation
- Ridge Regression
- Lasso Regression
- Elastic net

01. CAR PRICE/Normal Equation

- carprice_dataset.csv
- carprice _mc.ipynb
- carprice _md.ipynb
- carprice _model.pickle

01. CAR PRICE/Ridge Regression

- carprice_dataset.csv
- carprice _mc.ipynb
- carprice _md.ipynb
- carprice _model.pickle

01. CAR PRICE/Lasso Regression

- carprice_dataset.csv
- carprice _mc.ipynb
- carprice _md.ipynb
- carprice _model.pickle

01. CAR PRICE/Elastic net

- carprice_dataset.csv
- carprice _mc.ipynb
- carprice _md.ipynb
- carprice _model.pickle

DL101 : Multiple Regression

THANK YOU !

We hope you enjoy our course