****

**รายงาน**

**เรื่อง**

**Machine Learning และภาษาไพธอนสำหรับวิทยาศาสตร์ข้อมูล**

**จัดทําโดย**

**นายกฤษณพงษ์ เพ็งบุญ 6330300038**

**นายจิรเมธ สุทธาวาณิชย์ 6330300119**

**นายชญานนท์ พูลวาสน์ 6330300151**

**นายชญานิน ตลับเงิน 6330300160**

**เสนอ**

**ผศ.ดร.กุลวดี สมบูรณ์วิวัฒน์**

**รายงานนี้เป็นส่วนหนึ่งของรายวิชา**

**03603351 วิทยาศาสตร์ข้อมูลเบื้องต้นหมู่เรียนบรรยาย 800**

**ภาคต้น ปีการศึกษา 2565**

**มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา**

**คํานํา**

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา 03603351 วิทยาศาสตร์ข้อมูลเบื้องต้นเพื่อใช้ในการศึกษาค้นคว้าเกี่ยวกับ Machine Learning และภาษาไพธอนสำหรับวิทยาศาสตร์ข้อมูล

ทางผู้จัดทําหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อื่นที่สนใจในเรื่อง Machine Learning และภาษาไพธอนสำหรับวิทยาศาสตร์ข้อมูล หากมีข้อผิดพลาดประการใด ทางผู้จัดทําก็ขออภัยมา ณ ที่นี้ด้วย

ผู้จัดทํา

นายกฤษณพงษ์ เพ็งบุญ 6330300038

นายจิรเมธ สุทธาวาณิชย์ 6330300119

นายชญานนท์ พูลวาสน์ 6330300151

นายชญานิน ตลับเงิน 6330300160

**สารบัญ**

คำนำ a

สารบัญ........................................................................................................................................ b

Machine Learning……………………………………………………………………………………………………...1

หลักการของอัลกอริทึม k-Means และ SVM…………………………………………………………………….7

SVM………..………………………………………………………………………………………………………..7

K-mean………..………………………………………………………………………………………………….10

Supervised Learning………..………………………………………………………………………………………..13

Unsupervised Learning…..…………………………………………………………………………………………14

ภาษาไพธอนสำหรับวิทยาศาสตร์ข้อมูล……………………………………………………………………………15

numpy………..…………………………………………………………………………………………………..15

scipy………..……………………………………………………………………………………………….……..21

mathplotlib……………………………………………………………………………………………….…...24

seaborn………………………………………………………………………………………………………..…47

pandas………………………………………………………………………………………………………..….48

scikit-learn (Supervised) ……………………………………………………………………………....57

scikit-learn (Unsupervised) ……………………………………………………………………..……59

บรรณานุกรม............................................................................................................................62

Machine learning บน MNIST dataset

ฐานข้อมูลของตัวเลขที่เขียนด้วยลายมือที่ใช้กันทั่วไปสำหรับการฝึกอบรมต่างๆ การประมวลผลภาพระบบฐานข้อมูลที่ใช้กันอย่างแพร่หลายสำหรับการฝึกอบรมและการทดสอบในด้านการเรียนรู้ของเครื่อง สร้างขึ้นโดย "ผสมซ้ำ" ตัวอย่างจากชุดข้อมูลดั้งเดิมของ NIST ผู้สร้างรู้สึกว่าเนื่องจากชุดข้อมูลการฝึกอบรมของ NIST ถูกนำมาจากสำนักสำรวจสำมะโนประชากรของอเมริกาพนักงานในขณะที่ชุดข้อมูลการทดสอบถูกนำมาจากนักเรียนมัธยมปลายชาวอเมริกัน แต่ก็ไม่เหมาะสำหรับการทดลอง แมชชีนเลิร์นนิงนอกจากนี้ภาพขาวดำจาก NIST ยังถูกทำให้เป็นมาตรฐานเพื่อให้พอดีกับกรอบขอบ 28x28 พิกเซลและการต่อต้านนามแฝงซึ่งนำเสนอระดับสีเทา

ฐานข้อมูล MNIST ประกอบด้วยภาพการฝึกอบรม 60,000 ภาพและภาพทดสอบ 10,000 ภาพ ครึ่งหนึ่งของชุดการฝึกและครึ่งหนึ่งของชุดทดสอบถูกนำมาจากชุดข้อมูลการฝึกของ NIST ในขณะที่อีกครึ่งหนึ่งของชุดฝึกและอีกครึ่งหนึ่งของชุดทดสอบนั้นนำมาจากชุดข้อมูลการทดสอบของ NIST ผู้สร้างเดิมของฐานข้อมูลจะเก็บรายการวิธีการบางอย่างที่ทดสอบไว้ในกระดาษต้นฉบับพวกเขาใช้เครื่องสนับสนุนเวกเตอร์เพื่อให้ได้อัตราความผิดพลาด 0.8% ชุดข้อมูลเพิ่มเติมที่คล้ายกับ MNIST ที่เรียกว่า EMNIST ได้รับการเผยแพร่ในปี 2560 ซึ่งมีภาพการฝึกอบรม 240,000 ภาพและภาพการทดสอบตัวเลขและอักขระที่เขียนด้วยลายมือ 40,000 ภาพ

รูปภาพประกอบด้วย ลูกศร

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 1 อักขระที่เขียนด้วยลายมือ

ดึงข้อมูล MNIST โดยใช้ sklearn

เป็นตัวดึงข้อมูล คำสั่งในการดึงข้อมูลอยู่ในมอดูลย่อยชื่อว่า datasets เป็นการดึงข้อมูลที่มีอยู่แล้วมาใช้ ข้อมูลเหล่านั้นมีขนาดใหญ่เกินกว่าที่จะใส่ติดมาในตัวมอดูล จึงถูกใส่ไว้อยู่ในเว็บ จึงต้องต่อเน็ตอยู่จึงจะใช้ได้ฟังก์ชัน fetch\_openml ซึ่งอยู่ภายในมอดูล datasets มีไว้ดึงข้อมูลที่ถูกเตรียมไว้ในเว็บ [http://openml.org](http://openml.org/)  
เว็บนี้เป็นเว็บที่รวบรวมข้อมูลที่ใช้เป็นแบบฝึกสำหรับการเรียนรู้ของเครื่องมากมายหลายชนิด สามารถเขียนได้ดังนี้

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 2 import datasets

ในที่นี้โหลดมาแล้วข้อมูลจะถูกเก็บอยู่ในตัวแปร mnist ข้อมูลจะถูกเก็บอยู่ในแอตทริบิวต์ที่ชื่อ data โดยเป็นอาเรย์สองมิติ ส่วน ค่าตัวเลขที่เป็นคำตอบ (0~9) จะอยู่ใน target

A picture containing graphical user interface

Description automatically generated

รูปที่ 3 ขนาดของข้อมูล

ข้อมูลทั้งหมด 70000 แถว ก็คือมีภาพตัวเลขทั้งหมด 70000 ภาพ ส่วน 784 นี่คือขนาดของข้อมูล โดยข้อมูลนี้เป็นค่าความเข้มของดินสอในแต่ละช่อง ขนาดภาพ 28×28 จึงมี 784 ค่า  
ลอง print(mnist.data[1]) จะได้ค่าตัวเลขตั้งแต่ 0 (บริเวณว่าง) ไปจนถึง 255

A picture containing chart

Description automatically generated รูป 4 เอามาวาดเป็นภาพดูได้ โดยต้องทำการ reshape เป็น 28×28 ก่อน

Chart

Description automatically generated

รูป 5 จะเห็นว่าเป็นรูปเลข 0

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

รูป 6 ลองดูตัวเลขอื่นๆ

Shape

Description automatically generated รูป 7 ตัวเลข 1 ถึง 9

ตัวเลขพวกนี้คือสิ่งที่จะป้อนให้โปรแกรมของเราเรียนรู้ แบบนี้โปรแกรมเราก็จะเหมือนกับเด็กน้อยไร้เดียงสาที่พยายามหัดจดจำ หัดแยกแยะว่าภาพนี้คือตัวเลขนี้ ภาพนั้นคือตัวเลขนั้น

**สรุปหลักการของอัลกอริทึม k-Means และ SVM**

**Support Vector Machine**

เป็นอัลกอริทึมที่สามารถนำมาช่วยแก้ปัญหาการจำแนกข้อมูล ใช้ในการวิเคราะห์ข้อมูลและจำแนกข้อมูล โดยอาศัยหลักการของการหาสัมประสิทธิ์ของสมการเพื่อสร้างเส้นแบ่งแยกกลุ่มข้อมูลที่ถูกป้อนเข้าสู่กระบวนการสอนให้ระบบเรียนรู้ โดยเน้นไปยังเส้นแบ่งแยกแยะกลุ่มข้อมูลได้ดีที่สุด

**แนวความคิดของ Support Vector Machine**

เกิดจากการที่นำค่าของกลุ่มข้อมูลมาวางลงในฟีเจอร์สเปซ (Feature Space) จากนั้นจึงหาเส้นที่ใช้แบ่งข้อมูลทั้งสองออกจากกันโดยจะสร้างเส้นแบ่ง (Hyperplane) ที่เป็นเส้นตรงขึ้นมา และเพื่อให้ทราบว่าเส้นตรงที่แบ่งสองกลุ่มออกจากกันนั้น เส้นตรงใดเป็นเส้นที่ดีที่สุด

Chart, scatter chart

Description automatically generated

Diagram

Description automatically generatedChart, bubble chart

Description automatically generated รูปที่ 1 รูปที่ 2

สมมติว่าเราต้องการคัดแยกข้อมูลออกเป็น 2 กลุ่ม โดยใช้เส้นแบ่งที่เป็นเส้นตรง จะเห็นว่ามีเส้นตรงจำนวนมากที่สามารถคัดแยกได้ แต่เส้นตรงเส้นไหนที่ดีที่สุด เราจะนิยาม Margin เป็นผลรวมระยะห่างของเส้นตรงที่เป็นเส้นแบ่ง ถึงเส้นตรงที่ผ่านข้อมูลที่ใกล้ที่สุดและขนานกับเส้นแบ่งของทั้งสองกลุ่ม จะเห็นว่า H1 แม้จะสามารถแบ่งข้อมูลทั้งสองกลุ่มออกได้เช่นกัน แต่ ระยะในการแบ่งจากเส้นแบ่งไปถึงข้อมูลที่ใกล้ที่สุดนั้นมีขนาดน้อย แต่จากเส้น H2 จะเป็นเส้นที่แบ่งกลุ่มที่กว้างมากที่สุดของทั้งสองกลุ่มคือให้ค่า maximum margin เราเรียกข้อมูลที่อยู่บน margin นี้ว่า Support Vector

จากการกระจายตัวของข้อมูลในรูปที่ 1 จะเห็นว่าสามารถแบ่งแยกออกเป็น 2 กลุ่มได้อย่างชัดเจน ซึ่งโดยปกติแล้วเราจะใช้ linear model (หรือสมการเส้นตรง) เพื่อทำการแบ่งข้อมูลออกเป็น 2 คลาส ทว่า linear model นี้สามารถเป็นไปได้หลากหลายเส้นดังในรูปที่ 2

## **การเลือกโมเดล**

เราควรจะเลือกโมเดลที่ไม่ overfitting หรือโมเดลที่ไม่จำรูปแบบของข้อมูล training มากเกินไป จากตัวอย่างจะเห็นว่าโมเดลที่ 1 และโมเดลที่ 3 จะมีจุดหนึ่งที่ linear model อยู่ใกล้กับข้อมูลแต่ละคลาสมากเกินไป นั่นคือ ถ้ามีข้อมูลใหม่ที่อยู่ห่างออกไปสักเล็กน้อยก็จะทำนายผิดไปพลาดไป ดังนั้นในตัวอย่างนี้จึงควรจะเลือกโมเดลที่ 2  (เส้นสีแดง) และนั่นคือหลักการของ SVM ที่เลือก linear model ที่มีระยะห่างระหว่าง 2 คลาสห่างกันมากที่สุด ดังแสดงในรูป

Chart, scatter chart

Description automatically generated

Overfitting คือ

การที่โมเดลจดจำรูปแบบของข้อมูล training มากเกินไปจนไม่สามารถทำนายข้อมูล unseen data ได้

ประโยชน์

การวิเคราะห์จำแนกกลุ่ม ด้วยการวิเคราะห์จากตัวแปรตาม 1 ตัวและตัวแปรอิสระตั้งแต่ 1 ตัวขึ้น ไป การวิเคราะห์ด้วยวิธีนี้นอกจากจะสามารถจำแนกความ แตกต่างระหว่างกลุ่มได้และยังสามารถบอกได้ว่าตัวแปรใดจำแนกได้ดีมากน้อยกว่ากัน นั่นคือสามารถ บอกประสิทธิภาพหรือน้ำหนักในการจำแนกของการจัดเข้ากลุ่ม นอกจากนี้การวิเคราะห์จำแนกกลุ่มยัง สามารถพยากรณ์การเข้าสู่กลุ่มของข้อมูลใหม่ด้วย ดังนั้นการวิเคราะห์จำแนกกลุ่มจึงเป็นเทคนิคการวิเคราะห์ความสัมพันธ์หรือการหาสาเหตุได้

**K means**

วิธีการหนึ่งใน Data mining อยู่ในกลุ่มของ Unsupervised Learning หรือแปลตรงๆคือการเรียนรู้แบบไม่ต้องสอน (Supervised Learning ต้องสอนก่อนต้องจับ Train และต้อง Test เป็นต้น) โดยหน้าที่หลักของ K-means คือการแบ่งกลุ่ม แบบ Clustering ซึ่งการแบ่งกลุ่มในลักษณะนี้จะใช้พื้นฐานทางสถิติ ซึ่งแน่นอนว่าต้องมีตัวเลขประกอบ อย่างน้อย 2 ตัวแปรขึ้นไป

**วิธีการของ K-means** มี 4 ขั้นตอนดังนี้

1. กำหนดจำนวนกลุ่มขึ้นมาก่อนเช่น 2 กลุ่มหรือหมายความว่าค่า K=2 (กำหนดเป็น C1 และ C2)และสุ่มตำแหน่งแกน x, y ให้กับ C1  และ C2 จะได้  C1(x1,y1) และ C2(x2,y2)

2. ดูตำแหน่งของสมาชิกแต่ละสมาชิกว่าอยู่ใกล้ใครมากกว่ากันก็ให้คนนั้นเป็นสมาชิกของ C นั้น จาก ตรงนี้เราจะรู้แล้วว่าสมาชิกแต่ละคนอยู่ในกลุ่มใดระหว่าง C1 และ C2

3. ปรับ x, y ของ C1 และ C2 ใหม่ให้อยู่ตรงกลางของกลุ่ม

4. ทำ ตามข้อ 2 และข้อ 3 อีกครั้งจนกว่า C1 และ C2 ตำแหน่งไม่เปลี่ยน

Chart, scatter chart

Description automatically generated

รูปที่ 1 หาสมาชิกที่อยู่ใกล้กลุ่มที่สุด

Chart, scatter chart

Description automatically generated รูปที่ 2 หาสมาชิกที่อยู่ใกล้กลุ่มที่สุด

Chart, scatter chart

Description automatically generated

รูปที่ 3 C1 และ C2 ย้ายไปตรงกลาง

Chart, scatter chart

Description automatically generated รูปที่ 4 หาสมาชิกที่อยู่ใกล้กลุ่มที่สุด

Chart, scatter chart

Description automatically generated

รูปที่ 5 C1 และ C2 ย้ายไปตรงกลาง

**Logistic Regression Analysis**

การวิเคราะห์การทดถอยทางโลจีสติก เป็นเทคนิคการวิเคราะห์ตัวแปรเชิงพหุที่มีวัตถุประสงงค์ เพื่อประมาณค่าหรือทำนายเหตุการณ์ที่สนใจว่าจะเกิดขึ้นหรือไม่เหตุการณ์นั้นภายใต้อิทธิตัวปัจจัย แบบจำลองโลจีสติกประกอบด้วยตัวแปรตามหรือตัวแปรเกณฑ์ ที่ต้องเป็นตัวแปรแบบทวินาม กล่าวคือ มีได้ 2 ค่า เช่น “เกิด” หรือ “ไม่เกิด” หรือ “เสี่ยง” หรือ “ไม่เสี่ยง” เป็นต้น และตัวแปรอิสระ ที่อาจจะมีตัวเดียวหรือหลายตัวที่เป็นได้ทั้งตัวแปรเชิงกลุ่ม หรือ ตัวแปรแบบต่อเนื่อง การวิเคราะห์แบบถดถอยโลจีสติก เกี่ยวข้องกับทฤษฏีความน่าจะเป็นทวินาม ถูกเรียกว่า Binomial Logistic Regression ถ้าตัวแปรตามเป็นพหุนามจะเรียกว่า Multinomial Logistic Regression การถดถอยโลจีสติก จัดเป็นเครื่องมือวิเคราะห์ข้อมูลในการศึกษาวิจัยที่มีวัตถุประสงค์เพื่อทำนายเหตุการ์ณ หรือ ประเมินความเสี่ยง

**การจำลองการถดถอยโลจีสติก**

การวิเคราะห์การทดถอยทางโลจีสติกการประเมินค่าความน่าจะเป็นของการเกิดเหตุการณ์ มีตัวแบบมาจากฟังก์ชั่นโลจีสติก หากมีตัวแปรอิสระเพียงตัวเดียวฟังก์ชั่นโลจีสติกจะแสดงความน่าจะเป็นของเหตุการณ์ดังรูปต่อไปนี้

Graphical user interface, text

Description automatically generated

**Principal Components Analysis (PCA)**

คือ การวิเคราะห์องค์ประกอบหลัก การที่จะวิเคราะห์ประกอบหลักได้นั้น องค์ประกอบต้องมีหลายองค์ประกอบรวมกัน อาจจะมีมากกว่าหนึ่ง หรือ มากกว่าหนึ่งร้อย แต่สิ่งที่ต้องการคือองค์ประกอบหลักเท่านั้น ถ้ามองในเชิงสถิติ เพื่อให้เห็นภาพชัดขึ้น

สมมุติมี Feature อยู่ 50 ตัว การสร้างโมเดลจากข้อมูล Feature ทั้งหมดนั้นทำได้ยาก เพราะโอกาสที่ Target variable จะมีความสัมพันธ์กับ 50 Feature นั้นเป็นไปแทบไม่ได้เลย ดังนั้นการทำ PCA จึงเป็นความสัมพันธ์ของข้อมูลภายใน Feature ด้วยกันเอง และให้ความสำคัญกับ Feature ที่มีความสัมพันธ์มากกว่าก่อน PCA จึงเป็นเทคนิคในการลดมิติข้อมูลที่มีอยู่มาก ในเหลือน้อยลงในการคำนวณ

Chart, scatter chart

Description automatically generated

การลดมิติของข้อมูล

การใช้งาน PCA

- เหตุผลหลักของ PCA คือการลดมิติของข้อมูล (High dimensional) ให้มีมิติลดลง เห็นได้บ่อยครั้งในการใช้ PCA สำหรับงาน Image

- เพื่อลดจำนวนตัวแปร (Features) ทั้งนี้ PCA ไม่ได้ทำการลบข้อมูลแต่อย่างใด แต่เป็นการ Transform ข้อมูลในอยู่ในรูปแบบอื่น และการอธิบายแบบอื่น ซึ่งมีประโยชน์มากในงาน Regression เป็นต้น

- การทำ PCA ถือเป็นการแก้ไขปัญหา Multicollinearity ไปโดยอัตโนมัติ เพราะ Process ในการสร้าง Components ของ PCA จะไม่มีความสัมพันธ์กันเกิดขึ้น

**ภาษาไพธอนสำหรับวิทยาศาสตร์ข้อมูล**

**Numpy**

Numpy คือ Library ที่ใช้สำหรับการจัดการและสร้างโตนงสร้างข้อมูลประเภท array เพื่อจัดเก็บและเตรียมไปใช้ต่อมีความสามารถในการคำนวณทางคณิตศาสตร์พื้นฐานบางอย่างที่ List ไม่สามารถทำได้ และกิน Memory น้อยกว่า ซึ่งเป็น Library พื้นฐาน สำหรับการพัฒนาโปรเจค Data Sci

Diagram

Description automatically generated

**ความแตกต่างระหว่าง Numpy กับ Python Lists**

* NumPy Arrays สามารถคำนวณและดำเนินการทางตรรกะใน Matrix , Array หลายมิติ และ Array ได้อย่างรวดเร็ว มากกว่า Python Lists
* ในการใช้งาน NumPy Arrays จะประหยัด Memory ได้มากกว่าใช้ Python Lists
* NumPy Arrays มีขนาดคงที่เมื่อสร้าง ซึ่งแตกต่างจาก Python Lists (ซึ่งสามารถขยายได้แบบไดนามิก) การเปลี่ยนขนาดของ ndarray จะสร้างอาร์เรย์ใหม่และลบต้นฉบับ
* การเปลี่ยนแปลงข้อมูลใน NumPy Arrays ก็สามารถทำได้เร็วกว่า Python Lists
* NumPy Arrays สามารถเข้าถึงข้อมูลภายในได้เร็วกว่า Python Lists

**ความแตกต่างระหว่าง Numpy กับ Pandas**

* การเข้าถึงข้อมูลในแต่ละ index ของ Numpy เร็วกว่า Pandas อย่างมาก
* การทำงานของ Numpy มีเวลาการทำงานอยู่ในหน่วย **nanosecond**
* การทำงานของ Pandas มีเวลาการทำงานอยู่ในหน่วย **millisecond**

**ชนิดข้อมูลของนัมไพ**

Array numpy ต้องเก็บข้อมูลชนิดใดชนิดหนึ่ง ตัวอย่างชนิดข้อมูลดังนี้

Graphical user interface, text, application, email

Description automatically generated

**แอททริบิวต์ของ Numpy**

ใช้บอกโครงสร้างของ array คือ

Graphical user interface, text, application, email

Description automatically generated

**สร้างนัมไพอาร์เรย์แบบใช้ข้อมูลค่าเดียวกัน**

การสร้างอาร์เรย์ที่มีสมาชิกมี ค่าเหมือนกันทั้งหมด โดยมีฟังก์ชันสำหรับการสร้างอาร์เรย์ ดังนี้Table

Description automatically generated

**สร้างนัมไพอาร์เรย์แบบกำหนดช่วงข้อมูล**

การสร้างอาร์เรย์ที่มีสมาชิกอยู่ในช่วงตัวเลขที่กำหนด โดยมีฟังก์ชันสำหรับการสร้างอาร์เรย์ ดังนี้ A picture containing graphical user interface

Description automatically generated

**สร้างนัมไพอาร์เรย์แบบสุ่ม**

การสร้างอาร์เรย์ที่มีสุ่มค่าตัวเลข ด้วยเมธอด random โดยมีฟังก์ชันสำหรับการสร้างอาร์เรย์ ดังนี

Graphical user interface, text, application

Description automatically generated

**ฟังก์ชันทางสถิติ**

ไลบรารี Numpy มีฟังก์ชันสำหรับดำเนินการต่างๆ เป็นจำนวนมาก โดยตัวอย่างฟังก์ชันทางสถิติมีดังนี้

Table

Description automatically generated

**การคัดลอกสมาชิก**

Table

Description automatically generated

**การเพิ่มสมาชิก**

Table

Description automatically generated

**การลบสมาชิก**

Table

Description automatically generated

**การเปลี่ยนขนาดอาร์เรย์**

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated with medium confidence

Table

Description automatically generated

**การดำเนินการอื่นๆ**

Table

Description automatically generated

Graphical user interface, application

Description automatically generated**ตัวอย่างโปรแกรมที่ใช้ numpy**

**Scipy**

SciPy เป็นชุดของอัลกอริทึมทางคณิตศาสตร์และฟังก์ชันความสะดวกสบายที่พัฒนาขึ้นจาก NumPy ของ Python เพิ่มประสิทธิภาพการทำงานให้กับ Python แบบโต้ตอบโดยให้ผู้ใช้มีคําสั่งและคลาสระดับสูงสําหรับการจัดการและแสดงภาพข้อมูล ด้วย SciPy เซสชัน Python แบบโต้ตอบจะกลายเป็นระบบการประมวลผลข้อมูลและการสร้างต้นแบบระบบเช่น MATLAB, IDL, Octave, R-Lab และ SciLab

**ความแตกต่างระหว่าง Numpy กับ Scipy**

* NumPy Arrays สามารถคำนวณและดำเนินการทางตรรกะใน Matrix , Array หลายมิติ และ Array ได้อย่างรวดเร็ว มากกว่า Python Lists
* Numpy มีการดำเนินการทางคณิตศาสตร์ขั้นพื้นฐานเท่านั้นและการดำเนินการขั้นสูงแค่บางอย่าง
* Scipy มีการดำเนินการทางสถิติที่ซับซ้อนทั้งหมดที่จำเป็นสำหรับการวิเคราะห์ข้อมูลที่เหมาะสม

ฟังชันก์ต่างในการทำงานของ Scipy มีดังนี้

Shape

Description automatically generated

**special** ฟังก์ชันพิเศษใน scipy **ใช้เพื่อดําเนินการทางคณิตศาสตร์กับข้อมูลที่กําหนด** ฟังก์ชันพิเศษใน scipy เป็นโมดูลที่มีอยู่ในแพ็คเกจ scipy ภายในฟังก์ชันพิเศษนี้วิธีการที่มีอยู่คือ cbrt - ซึ่งให้รากลูกบาศก์ของจํานวนที่กําหนด

**stats** ประกอบด้วยการแจกแจงความน่าจะเป็น, สถิติสรุปและความถี่, ฟังก์ชันสหสัมพันธ์และการทดสอบทางสถิติ, สถิติที่สวมหน้ากาก, การประมาณค่าความหนาแน่นของเคอร์เนล, ฟังก์ชันกึ่งมอนติคาร์โล และอื่น ๆ

สถิติเป็นพื้นที่ที่มีขนาดใหญ่มากและมีหัวข้อที่อยู่นอกขอบเขตสําหรับ SciPy และครอบคลุมโดยแพ็คเกจอื่น ๆ คือ

* [statsmodels](https://www.statsmodels.org/stable/index.html): การถดถอย, แบบจําลองเชิงเส้น, การวิเคราะห์อนุกรมเวลา, ส่วนขยายไปยังหัวข้อที่ครอบคลุมโดย .scipy.stats
* [Pandas](https://pandas.pydata.org/): ข้อมูลตาราง, ฟังก์ชันอนุกรมเวลา, อินเทอร์เฟซไปยังภาษาสถิติอื่น ๆ
* [PyMC](https://docs.pymc.io/): การสร้างแบบจําลองทางสถิติแบบ Bayesian, การเรียนรู้ของเครื่องความน่าจะเป็น
* [scikit-learn](https://scikit-learn.org/): การจําแนกประเภทการถดถอยการเลือกแบบจําลอง
* [Seaborn](https://seaborn.pydata.org/): การสร้างภาพข้อมูลทางสถิติ
* [rpy2](https://rpy2.github.io/): งูหลามถึงสะพาน R

**spatial** สามารถคํานวณสามเหลี่ยมไดอะแกรม Voronoi โดยใช้ประโยชน์จากไลบรารี [Qhull](http://qhull.org/) นอกจากนี้ยังมีการใช้งาน [**KDTree**](https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html#scipy.spatial.KDTree) สําหรับการสืบค้นจุดเพื่อนบ้านที่ใกล้ที่สุดและยูทิลิตี้สําหรับการคํานวณระยะทางในเมตริกต่างๆ

**sparse** เป็นเครื่องมือสําหรับการสร้างเมทริกซ์ที่ละเอียดโดยใช้โครงสร้างข้อมูลหลายแบบ รวมถึงเครื่องมือสําหรับการแปลงเมทริกซ์ที่หนาแน่นเป็นเมทริกซ์ที่ละเอียด ฟังก์ชันพีชคณิตเชิงเส้น NumPy และ SciPy จํานวนมากที่ทํางานบนอาร์เรย์ NumPy สามารถทํางานบนอาร์เรย์เบาบางของ SciPy ได้อย่างโปร่งใส

**signal** คํานวณอาร์เรย์ดัชนีความล่าช้า / การกระจัดสําหรับสหสัมพันธ์ข้าม 1 มิติ

**optimize** ฟังก์ชันสําหรับการลด (หรือเพิ่ม) ฟังก์ชันวัตถุประสงค์ให้เหลือน้อยที่สุด ซึ่งอาจอยู่ภายใต้ข้อจํากัด มันรวมถึงตัวแก้ปัญหาสําหรับปัญหาที่ไม่เชิงเส้น การเขียนโปรแกรมเชิงเส้นสี่เหลี่ยมจัตุรัสที่ จํากัด และไม่เชิงเส้นการค้นหารากและการปรับเส้นโค้งให้เหมาะสม

**odr** การถดถอยระยะทางมุมฉากซึ่งใช้ในการศึกษาการถดถอย การถดถอยเชิงเส้นพื้นฐานมักใช้เพื่อประเมินความสัมพันธ์ระหว่างตัวแปรสองตัว y และ x โดยการวาดเส้นที่เหมาะสมที่สุดบนกราฟ

**ndimage** หมายถึงภาพ n มิติ งานที่พบบ่อยที่สุดในการประมวลผลภาพมีดังนี้ อินพุต/ เอาต์พุตแสดงภาพ การปรับแต่งพื้นฐาน - การครอบตัดการพลิกการหมุน ฯลฯ

**linalg** คํานวณ eigenvalues จากปัญหา eigenvalue ทั่วไปหรือทั่วไป ฟังก์ชันนี้ส่งกลับค่า Eigen และเวกเตอร์ Eigen

**io ฟังก์ชั่นยูทิลิตี้เพิ่มเติมสําหรับการเพิ่มประสิทธิภาพสถิติและการประมวลผลสัญญาณ**

**interpolate** สร้างฟังก์ชันตามจุดข้อมูลคงที่ซึ่งสามารถประเมินได้ทุกที่ภายในโดเมนที่กําหนดโดยข้อมูลที่กําหนดโดยใช้การแก้ไขเชิงเส้น ด้วยการใช้ข้อมูลข้างต้นให้เราสร้างฟังก์ชันการแก้ไขและวาดกราฟที่แก้ไขใหม่

**integrate** การรวมเชิงตัวเลขคือการคํานวณโดยประมาณของอินทิกรัลโดยใช้เทคนิคเชิงตัวเลข

**fftpack** ช่วยให้การคํานวณการแปลงฟูริเยร์เป็นไปอย่างรวดเร็ว เป็นภาพประกอบสัญญาณอินพุต (มีเสียงดัง) อาจมีลักษณะดังนี้ - นําเข้า numpy เป็น np time\_step = 0.02 ช่วงเวลา = 5

**constants** **เป็นโมดูลย่อยภายในไลบรารี Scipy** ที่ทําสิ่งนี้ให้เรา มันมีรายการที่สมบูรณ์ของค่าคงที่ทางคณิตศาสตร์สากลค่าคงที่ทางกายภาพและหน่วย

**cluster** โมดูลลําดับชั้นมีฟังก์ชันสําหรับการจัดกลุ่มแบบลําดับชั้นและการสลับกัน คุณสมบัติของมันรวมถึงการสร้างคลัสเตอร์ลําดับชั้นจากระยะไกล

**ตัวอย่างโปรแกรมที่ใช้ scipy**

Chart

Description automatically generated

**Matplotlib**

เป็น library ที่มีคำสั่งใช้สร้างกราฟต่างๆ หลายรูปแบบที่สามารถปรับเปลี่ยนการแสดงผล สี เนื้อหา รายละเอียดต่างๆ ได้ครบถ้วน และเป็นเครื่องมือพื้นฐานที่ใช้กันในการแสดงผลข้อมูล

**การสร้างกราฟเส้น**

ใช้คำสั่ง plt.plot() ที่รับค่า x และค่า y โดย ในตัวอย่างใส่เป็น array matplotlib จะนำข้อมูลที่ตำแหน่งตรงกันมาประกอบเป็นจุดพิกัดและนำไป plot เช่นจุดแรกจะเกิดจากข้อมูลตัวแรกจาก x และข้อมูลตัวแรกจาก y สร้างเป็นพิกัด x,y

**ข้อมูลที่ใช้**

กราฟเส้นเหมาะกับการใช้แสดงข้อมูลที่มีความต่อเนื่อง เช่น ความสัมพันธ์ของ x และ y = x\*\*2

In [7]:

x = np.linspace(1,10,10)

y = x\*\*2

**การใช้งาน**

เพื่อสร้างกราฟจะใช้การสั่งคำสั่งเพื่อทำสิ่งต่างๆ และตามด้วยคำสั่ง plt.show() เพื่อประมวลผลคำสั่งต่างๆ ที่เกี่ยวข้องกับการแสดงผลกราฟ และแสดงออกมาเป็นกราฟตามคำสั่ง

การ plot กราฟ y = x

In [8]:

plt.plot(x,x)

plt.show()

Chart, line chart

Description automatically generated

**การ plot หลายข้อมูล**

ใช้การสั่งคำสั่ง plt.plot() แทนการสร้างกราฟสำหรับ 1 ข้อมูล สามารถสร้างกราฟของกี่ข้อมูลก็ได้ ให้สั่งคำสั่งตามแต่ละลำดับไป เช่น การสร้างกราฟจากข้อมูล y = x และข้อมูล y = x\*\*2 บนกราฟเดียวกันสามารถเขียนได้ลักษณะดังนี้

In [9]:

plt.plot(x,x)

plt.plot(x,x\*\*2)

plt.show()

Chart, line chart

Description automatically generated

**สร้างความแตกต่างแต่ละข้อมูล**

ในการอ่านทำความเข้าใจกราฟที่ประกอบไปด้วยหลายข้อมูลอาจทำให้เกิดความสับสนได้ เราสามารถสร้างความแตกต่างของการแสดงผลแต่ละข้อมูลได้ผ่านการเปลี่ยน สี ลักษณะเส้นกราฟ และสัญลักษณ์ ของแต่ละข้อมูล

ผ่านการระบุ string ที่ใส่สัญลักษณ์ของแต่ละค่าเอาไว้ เช่น กราฟเส้นสีดำใช้ 'k' กราฟเส้นทึบใช้ '-' กราฟที่มีสัญลักษณ์ข้าวหลามตัดใช้ 'D' เขียนรวมได้เป็น 'k-D' และระบุเป็น parameter เอาไว้ใน plt.plot() ใช้งานในลักษณะดังนี้

In [10]:

plt.plot(x,x,'k-D')

plt.plot(x,x\*\*2)

plt.show()

Chart, line chart

Description automatically generated

**การใส่แกน x,y**

เพื่อเพิ่มความเข้าใจค่าที่ระบุในแกน เราสามารถแปะ label สำหรับแกน x,y ได้ด้วยคำสั่ง .xlabel() .ylabel() ที่รับค่า string ที่เก็บชื่อที่ต้องการเอาไว้

In [11]:

plt.plot(x,x,'k-D')

plt.plot(x,x\*\*2)

plt.xlabel('x')

plt.ylabel('y')

plt.show()

Chart, line chart

Description automatically generated

**ชื่อแต่ละข้อมูล**

เราสามารถแสดงผลหลายข้อมูลบนกราฟได้พร้อมกัน ซึ่งอาจทำให้เกิดความสับสนเกี่ยวกับข้อมูลที่แสดงอยู่ได้ ซึ่งการระบุชื่อของตัวข้อมูลหรือ label เพิ่มเติมจากสี ลักษณะเส้นกราฟ และสัญลักษณ์ พิเศษจะช่วยให้กราฟเข้าใจได้ง่ายขึ้น

**label**

ในการระบุชื่อข้อมูลที่ต้องการ เราจะใช้การระบุ label เป็น parameter สำหรับ plt.plot() เพื่อตั้งชื่อให้กับข้อมูล ซึ่ง label จะแสดงผลผ่าน legend ซึ่งต้องเขียนคำสั่งเพื่อแสดงผล legend

**legend**

คือ กล่องข้อความที่แสดงชื่อของแต่ละข้อมูล โดยการระบุผ่านสี ลักษณะเส้นกราฟ และสัญลักษณ์ ซึ่งจะแสดงต่อเมื่อใช้คำสั่ง plt.legend()

• ตำแหน่งของ legend

สามารถปรับแต่งได้โดยใช้ parameter loc= ตามด้วย string ที่ระบุค่าการกำหนดตำแหน่งที่มีให้เลือก ดังนี้ best, upper right, upper left, lower left, lower right, right, center left, center right, lower center, upper center, center

ซึ่งหากไม่ระบุตำแหน่งจะใช้ default เป็น best คือให้ matplotlib เลือกให้ ว่าควรอยู่ที่ใด

In [12]:

plt.plot(x,x,'k-D',label='y = x')

plt.plot(x,x\*\*2,label='y = x\*\*2')

plt.legend(loc='best')

plt.xlabel('x')

plt.ylabel('y')

plt.show()

Chart, line chart

Description automatically generated

**การใส่ชื่อกราฟ**

ใช้คำสั่ง plt.title() ที่รับค่า string ที่เก็บข้อมูลชื่อที่ต้องการตั้ง และสามารถใส่ parameter fontsize= เพื่อกำหนดขนาดชื่อกราฟ

In [13]:

plt.title("Graphs showing y = x and y = x\*\*2",fontsize=15)

plt.plot(x,x,'k-D',label='y = x')

plt.plot(x,x\*\*2,label='y = x\*\*2')

plt.legend(loc='best')

plt.xlabel('x')

plt.ylabel('y')

plt.show()

Chart, line chart

Description automatically generated

**กราฟแท่ง**

ใช้คำสั่ง .bar() สำหรับกราฟแท่งแนวตั้ง และ .barh() สำหรับกราฟแท่งแนวนอน ในการสร้างโดยรับค่า x, y เป็น parameter หลัก และสามารถใช้ร่วมกับการตั้งชื่อกราฟ สร้างความแตกต่างแต่ละข้อมูล ตั้งชื่อข้อมูล และอื่นๆ ได้

**ข้อมูลที่ใช้**

กราฟแท่งเหมาะกับการใช้เปรียบเทียบข้อมูลที่ไม่ต่อเนื่องกัน เช่น ยอดขายของสินค้าแต่ละประเภท

In [14]:

x = ['shirts','pants','shorts','shoes']

y = [1000,1200,800,1800]

**กราฟแท่งแนวตั้ง**

In [15]:

plt.bar(x,y)

plt.title('Sales by category')

plt.ylabel('sales revenue')

plt.xlabel('item type')

plt.show()

A picture containing icon

Description automatically generated

**กราฟแท่งแนวนอน**

In [16]:

plt.barh(x,y)

plt.title('Sales by category')

plt.ylabel('item type')

plt.xlabel('sales revenue')

plt.show()

Chart, bar chart

Description automatically generated

**Histogram**

เป็นกราฟที่ใช้เพื่อดูความถี่ของการเกิดขึ้นของข้อมูล จะเหมาะกับข้อมูลที่มีจำนวนมาก ซึ่ง Histogram จะมีการแบ่งข้อมูลออกเป็น bins และนับการเกิดของแต่ละ bins เพื่อนับความถี่ของแต่ละ bins ที่เกิดขึ้นและสร้างเป็นกราฟลักษณะคล้ายกราฟแท่ง

**ข้อมูลที่ใช้**

เราจะสร้างข้อมูล จำนวน 100 ข้อมูล จากการสุ่มจากคำสั่ง np.random.normal() ที่สุ่มข้อมูลมาจาก normal distribution

In [17]:

x = np.random.normal(size=100)

**สร้าง histogram**

ใช้คำสั่ง .hist() รับค่า x และจำนวน bins หรือจำนวนแท่งกราฟที่ต้องการ โดย plt.hist จะทำการแบ่งข้อมูลเป็นกลุ่มต่างๆ จำนวนเท่ากับ bins และทำการนับแต่ละ bins

In [18]:

plt.title('Data Distribution')

plt.hist(x,bins=20)

plt.xlabel('y value')

plt.ylabel('frequency')

plt.show()

Chart, histogram

Description automatically generated

**Scatter Plot**

นิยมใช้แสดงผลข้อมูลเพื่อดูความสัมพันธ์ระหว่างแกน x,y หรือ ระหว่างในชุดข้อมูลเอง การสร้างใช้คำสั่ง .scatter() ที่รับค่า x, y

In [19]:

plt.scatter(np.linspace(1,100,100),np.random.normal(size=100))

plt.show()

Chart, scatter chart

Description automatically generated

**Box Plot**

เป็นกราฟที่ใช้ดูการกระจายตัวของข้อมูล โดยมีพื้นฐานการคำนวนจากการวัด Quartile ซึ่งทำการเรียงข้อมูลทั้งหมดจากน้อยไปมาก และแบ่งข้อมูลออกเป็น 4 ส่วน Q1 Q2 Q3 Q4 โดยมีกล่องตรงกลาง แสดงพื้นที่ Q1 ถึง Q3 มีเส้นสีส้มแสดงค่า median และระความยาวของกล่องตรงกลางเรียกว่าค่า IQR (Interquartile-range, Q3-Q1) และมีเส้นตรงขีดออกไปยังค่า Q1-(1.5 x IQR) ในด้านที่ค่าน้อยกว่า และ Q3+(1.5 x IQR) ในด้านที่ค่ามากกว่า โดยหากข้อมูลใดมีค่ามากกว่า หรือน้อยกว่าจุดสิ้นสุดของเส้นตรงดังกล่าว เราจะเรียกข้อมูลเหล่านั้นว่า Outlier

**ข้อมูลที่ใช้**

ใน boxplot เหมาะกับข้อมูลจำนวนมาก ที่เราต้องการดูการกระจายตัว ในตัวอย่างจะสร้างเลขจำนวน 100 ตัวจากการสุ่มจาก normal distribution

In [20]:

x = np.random.normal(size=100)

**การใช้งาน boxplot**

ใช้คำสั่ง .boxplot() โดยรับค่าที่ต้องการดูการกระจายตัวข้อมูล

In [21]:

plt.boxplot(x)

plt.xlabel('100 data points from Normal Distribution')

plt.ylabel('data values')

plt.show()

Chart, box and whisker chart

Description automatically generated

**การสร้างหลายกราฟ**

ใช้การสร้างพื้นที่ในการสร้างกราฟที่แบ่งแต่ละกราฟออกเป็นส่วนๆ และทำการใช้คำสั่งสร้างกราฟที่เราเรียนไปต่างๆ สร้างกราฟลงไป

**สร้างพื้นที่การทำงาน**

ด้วยคำสั่ง plt.subplots() โดยให้กำหนด nrows= กำหนดจำนวนแถว ncols= กำหนดจำนวนกราฟแต่ละแถว และ figsize= กำหนดขนาดของกราฟทั้งหมดเป็น tuple ที่ประกอบไปด้วย (ความกว้าง,ความสูง)

เช่น การสร้างพื้นที่เพื่อสร้าง 6 กราฟ โดยแบ่งเป็น 3 แถว แถวละ 2 กราฟ โดยมีขนาด 8 x 8 โดยเราจะสร้างตัวแปร fig, axes มา unpack ข้อมูลที่ได้จาก คำสั่ง .subplots()

In [22]:

fig, axes = plt.subplots(nrows=3,ncols=2,figsize=(8,8))

plt.tight\_layout()

A picture containing shoji, building, window, clipart

Description automatically generated

**ใส่กราฟลงพื้นที่**

โดยใช้ข้อมูลในตัวแปร axes ตามด้วยการระบุตำแหน่งตามหลังด้วยสัญลักษณ์ [แถว,คอลัมน์] เพื่อระบุตำแหน่งของกราฟที่ต้องการใส่ ตำแหน่งแรกเริ่มจาก 0

เช่น การใส่กราฟเส้นลงไปที่ตำแหน่ง [0,0] หรือ ซ้ายบน และตามด้วยคำสั่ง .plot() (เพื่อสร้างกราฟเส้น) และเราจะใช้คำสั่ง plt.tight\_layout() เพื่อปรับขนาดให้ตัวอักษรไม่ทับกันโดยอัตโนมัติ

In [23]:

fig, axes = plt.subplots(nrows=3,ncols=2,figsize=(8,8))

​

axes[0,0].set\_title("Graphs showing y = x and y = x\*\*2")

axes[0,0].plot(np.linspace(1,10,10),np.linspace(1,10,10),'k-D',label='y = x')

axes[0,0].plot(np.linspace(1,10,10),np.linspace(1,10,10)\*\*2,label='y = x\*\*2')

axes[0,0].legend(loc='best')

axes[0,0].set\_xlabel('x')

axes[0,0].set\_ylabel('y')

​

plt.tight\_layout()

A picture containing calendar

Description automatically generated

ทำซ้ำจนครบทุกกราฟ

In [24]:

fig, axes = plt.subplots(nrows=3,ncols=2,figsize=(8,8))

​

axes[0,0].set\_title("Graphs showing y = x and y = x\*\*2")

axes[0,0].plot(np.linspace(1,10,10),np.linspace(1,10,10),'k-D',label='y = x')

axes[0,0].plot(np.linspace(1,10,10),np.linspace(1,10,10)\*\*2,label='y = x\*\*2')

axes[0,0].legend(loc='best')

axes[0,0].set\_xlabel('x')

axes[0,0].set\_ylabel('y')

​

axes[0,1].set\_title('Sales by category')

axes[0,1].bar(['shirts','pants','shorts','shoes'],[1000,1200,800,1800])

axes[0,1].set\_ylabel('sales revenue')

axes[0,1].set\_xlabel('item type')

​

axes[1,0].set\_title('Sales by category')

axes[1,0].barh(['shirts','pants','shorts','shoes'],[1000,1200,800,1800])

axes[1,0].set\_ylabel('item type')

axes[1,0].set\_xlabel('sales revenue')

​

axes[1,1].set\_title('Data Distribution')

axes[1,1].hist(np.random.normal(size=100),bins=20)

axes[1,1].set\_xlabel('value')

axes[1,1].set\_ylabel('frequency')

​

axes[2,0].set\_title('Scatter Plot')

axes[2,0].scatter(np.linspace(1,100,100),np.random.normal(size=100))

axes[2,0].set\_xlabel('x')

axes[2,0].set\_ylabel('Values')

​

axes[2,1].set\_title('Boxplot')

axes[2,1].boxplot(np.random.normal(size=100))

axes[2,1].set\_xlabel('100 Randomly Generated')

axes[2,1].set\_ylabel('Values')

​

plt.tight\_layout()

Graphical user interface, application, Teams

Description automatically generated

**เปลี่ยน style**

สามารถทำได้โดยการใช้ style ที่ถูกสร้างไว้สำเร็จรูป ซึ่ง seaborn เป็นหนึ่งใน library ที่ใช้ตกแต่ง รวมไปถึง plot กราฟในรูปแบบต่างๆ ที่เราจะใช้เพื่อเปลี่ยนรูปแบบการแสดงผล

เราจะใช้คำสั่ง sns.set\_style() และกำหนด parameter style= ซึ่งมีให้เลือกเบื้องต้น white, dark, whitegrid, darkgrid, ticks สามารถใช้งานได้ดังนี้

In [25]:

import seaborn as sns

sns.set\_style(style='whitegrid')

fig, axes = plt.subplots(nrows=3,ncols=2,figsize=(8,8))

axes[0,0].set\_title("Graphs showing y = x and y = x\*\*2")

axes[0,0].plot(np.linspace(1,10,10),np.linspace(1,10,10),'k-D',label='y = x')

axes[0,0].plot(np.linspace(1,10,10),np.linspace(1,10,10)\*\*2,label='y = x\*\*2')

axes[0,0].legend(loc='best')

axes[0,0].set\_xlabel('x')

axes[0,0].set\_ylabel('y')

axes[0,1].set\_title('Sales by category')

axes[0,1].bar(['shirts','pants','shorts','shoes'],[1000,1200,800,1800])

axes[0,1].set\_ylabel('sales revenue')

axes[0,1].set\_xlabel('item type')

axes[1,0].set\_title('Sales by category')

axes[1,0].barh(['shirts','pants','shorts','shoes'],[1000,1200,800,1800])

axes[1,0].set\_ylabel('item type')

axes[1,0].set\_xlabel('sales revenue')

axes[1,1].set\_title('Data Distribution')

axes[1,1].hist(np.random.normal(size=100),bins=20)

axes[1,1].set\_xlabel('value')

axes[1,1].set\_ylabel('frequency')

axes[2,0].set\_title('Scatter Plot')

axes[2,0].scatter(np.linspace(1,100,100),np.random.normal(size=100))

axes[2,0].set\_xlabel('x')

axes[2,0].set\_ylabel('Values')

axes[2,1].set\_title('Boxplot')

axes[2,1].boxplot(np.random.normal(size=100))

axes[2,1].set\_xlabel('100 Randomly Generated')

axes[2,1].set\_ylabel('Values')

plt.tight\_layout()

Chart

Description automatically generated

**Seaborn**

Seaborn เป็นไลบรารีสำหรับสร้างกราฟิกทางสถิติใน Python สร้างขึ้นจาก matplotlib และทำงานร่วมกันอย่างใกล้ชิดกับ โครงสร้างข้อมูล Pandas ด้วย Seaborn Design สามารถสำรวจและทำความเข้าใจข้อมูลได้อย่างรวดเร็ว Seaborn รวบรวมเฟรมข้อมูลหรืออาร์เรย์ทั้งหมดที่มีข้อมูลทั้งหมดและทำหน้าที่ภายในทั้งหมดที่จำเป็นสำหรับการแมปความหมายและการรวมสถิติเพื่อเปลี่ยนข้อมูลให้เป็นแผนภูมิข้อมูลเป็นนามธรรมความซับซ้อนและช่วยให้สามารถออกแบบแปลงตามความต้องการได้

ตัวอย่างการใช้งาน Seaborn

Chart, box and whisker chart

Description automatically generated

**Pandas**

Pandas เป็น software library ที่ถูกเขียนด้วยภาษา Python ใช้ในการจัดการและวิเคราะห์ข้อมูล หรือ data โดยมี data structures และ operations ต่างๆ ที่ใช้ในการจัดการกับตารางตัวเลขและ time series

Feature ของ Library Pandas

* DataFrame Object สำหรับจัดการข้อมูลด้วยการทำดัชนีแบบรวม
* เครื่องมือสำหรับอ่านและเขียนข้อมูลระหว่างโครงสร้างข้อมูลที่อยู่ในหน่วยความจำและไฟล์ใน Formats อื่นๆ
* การจัดตำแหน่งข้อมูลและรวบรวมข้อมูลสำหรับ Missing Data หรือ ข้อมูลที่ขาดหายไป
* การ Reshape และการ pivot สำหรับชุดหรือข้อมูลหรือ Data set
* การทำ Label สำหรับการ Slicing,การทำ Indexing หรือ การทำดัชนี การย่อยชุดข้อมูลที่มีขนาดใหญ่
* การแทรกและการลบคอลัมน์ ของ Data Structure
* จัดกลุ่มตาม Engine เพื่อให้สามารถใช้ดำเนินการ Split-apply-combine กับ Data set
* การ Merging และ Joining ของ Data set
* การ Indexing แกนแบบตามลำดับชั้น เพื่อในการทำงานกับข้อมูลแบบ High-dimensional ใน Data structure ที่มีมิติต่ำกว่า
* ฟังก์ชั่น Time series: การสร้าง Range ของวันและความถี่ของการเปลี่ยนแปลง,การย้าย Window statistics,การย้าย Window linear regression,การ Shift วัน และ การ Lagging

Dataframes

หลักๆแล้ว Pandas จะถูกใช้สำหรับการวิเคราะห์ข้อมูลโดยสามารถนำข้อมูลเข้ามาได้หลายรูปแบบ เช่น JSON,SQL,Microsoft Excel เป็นต้น และ มี Operations มากมาที่ใช้ในการเก็บข้อมูล

การสร้าง dataframe โดยกำหนดข้อมูลด้วย Numpy array หรือ Dictionary สามารถกำหนด Index ได้ด้วย Argument ชื่อ Index เช่น

Graphical user interface, text, application, email

Description automatically generated

Read\_csv

สร้าง Dataframe โดยอ่านจากไฟล์ CSV สามารถกำหนด Index ด้วย Index\_col เช่น

Table

Description automatically generated

Columns & dtypes

Column ใช้ดูชื่อ Column ทั้งหมดใน Dataframe dtypes ใช้สำหรับดูชนิดของข้อมูลแต่ละ Column ใน Dataframe

Graphical user interface, application, Word

Description automatically generated

Head & tail

ดูตัวอย่างข้อมูลด้วยคำสั่ง Head หรือ tail โดย head ใช้ดูสำหรับช่วงต้นของ Dataframe และ tail ใช้สำหรับดูช่วงท้ายของ Dataframe สามารถระบุช่วงจำนวนตัวอย่างที่ต้องการได้ โดยถ้าไม่ระบุ คำสั่งจะแสดง 5 ตัวอย่าง

Table

Description automatically generated

to\_numpy

แปลง Dataframe เป็น Numpy Array

Graphical user interface, application

Description automatically generated

Describe

สรุปสถิติเบื้องต้นของข้อมูลแต่ละ column ใน DataFrame เช่น ค่าเฉลี่ย ค่า standard deviation (std) ค่าต่ำสุด ค่าสูงสุด

Graphical user interface, application, table

Description automatically generated

Index

แสดง index ของ DataFrame  
ถ้าเราไม่ได้กำหนด index ตอนสร้าง DataFrame เราจะได้เลขลำดับ 0, 1, 2, 3, … เป็น index

Text, application

Description automatically generated

Sort\_index & sort\_values

sort\_index เรียงลำดับ row ตามค่าของ index  
sort\_values เรียงลำดับ row ตามค่าของ column ที่กำหนด

Table

Description automatically generated

df[ ] & df.

เลือก column ที่ต้องการตามชื่อ column โดยสามารถใส่เป็น string ใน [ ] หรือเป็นชื่อ attribute ก็ได้

Graphical user interface, application

Description automatically generated

df[ : ]

เลือกช่วงของ row ที่ต้องการ โดยใส่เป็นเลขลำดับบรรทัด หรือใส่ตาม index  
ในกรณีนี้ index คือวันที่ และเรามีข้อมูลเรียงจากวันที่ใหม่กว่าไปวันที่เก่ากว่า ดังนั้นเราจึงใส่ index ของวันที่แบบถอยหลัง

Table

Description automatically generated

LOC

เลือก row ที่ต้องการหรือช่วงของ row ที่ต้องการตาม index  
หากไม่ต้องการทุก column สามารถกำหนด column ที่ต้องการได้ด้วยชื่อ column

Graphical user interface, table

Description automatically generated

ILOC

เลือก row ที่ต้องการหรือช่วงของ row ที่ต้องการตามเลขลำดับบรรทัด  
หากไม่ต้องการทุก column สามารถกำหนด column ที่ต้องการได้ด้วยเลขลำดับ column

Table

Description automatically generated

df[ boolean expression ]

เลือก row ที่ต้องการตามเงื่อนไขใน boolean expression

Table

Description automatically generated

## mean, std, median, max, min

คำนวณค่าทางสถิติของแต่ละ column ด้วยคำสั่งต่างๆ เช่น  
mean - ค่าเฉลี่ย  
std - ค่า standard deviation  
median - ค่ามัธยฐาน  
min - ค่าต่ำสุด  
max - ค่าสูงสุด

Graphical user interface, application

Description automatically generated

Groupby

จัดกลุ่มข้อมูลตาม column ที่กำหนด

Table

Description automatically generated

**Diagram

Description automatically generatedSupervised Learning**

แสดงแผนผังการเรียนรู้แบบมีผู้สอน Supervised Learning

Classification- K Nearest Neighbors

K nearest neighbors (KNN)

เป็นหนึ่งในกระบวนการเรียนรู้ที่ง่ายที่สุด : รับค่าใหม่ เป็นข้อมูลที่ไม่รู้จัก โดยค้นหาในฐานข้อมูลที่อ้างอิงสิ่งที่มีคุณลักษณะใกล้เคียง และมอบหมายให้ predominant class

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติรูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

 ดึงข้อมูลจาก Classification ที่บันทึกไว้กลับมาใช้งานต่อ

### Unsupervised learning

เป็นการเรียนรู้ที่ไม่ใช้ labels ใด ๆ ใช้เพียงข้อมูลที่ได้รับ

#### **Dimensionality reduction กับ PCA (Principal Components Analysis)**

สรุปย่อ ๆ PCA คือ การหาแกนที่สำคัญเมื่ออ้างอิงจากข้อมูล

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

Chart, scatter chart

Description automatically generatedสีน้ำเงิน คือ Setosa สีเขียว คือ Versicolour สีแดง คือ Virginica

#### **Clustering**

Chart, scatter chart

Description automatically generatedรูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ  
 การจัดกลุ่ม (Clustering) เป็นวิธีการหนึ่งที่แบ่งข้อมูลออกเป็นกลุ่ม (cluster) โดยพิจารณาให้แต่ละแถวเสมือนเป็นวัตถุ แล้วจับกลุ่มให้กับวัตถุที่มีความคล้ายคลึงกันอยู่ใน cluster เดียวกัน วัตถุที่อยู่ต่าง cluster จะมีความคล้ายคลึงกันน้อยที่สุด

จุดสีน้ำเงินใหญ่คือ cluster ที่ถูกแบ่ง

**บรรณานุกรม**

- https://www.softnix.co.th/2018/09/06/%E0%B8%A7%E0%B9%88%E0%B8%B2%E0 %B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-k-means-%E0%B9%81%E0%B8%A 5%E0%B8%B0%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B8%A2%E0%B8%B8%E0%B8%81%E0%B8%95%E0%B8%A3/

- https://www.glurgeek.com/education/support-vector-machine/?amp=1

https://hackathailand.com/wp-content/uploads/2021/09/B-101-08-Numpy-%E0%B8%A3%E0%B8%B1%E0%B8%8A%E0%B8%81%E0%B8%B8%E0%B8%A5-%E0%B9%80%E0%B8%9B%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3.pdf

- https://docs.scipy.org/doc/scipy/tutorial/index.html#user-guide

- https://www.ultimatepython.co/post/graph-data-visualization-python-matplotlib-seaborn

- https://hitechglitz.com/thailand/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8% A1%E0%B8%B7%E0%B8%AD%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%99%E0%B8%AA%E0%B8%B3%E0%B8%AB%E0%B8%A3%E0%B8%B1%E0%B8%9A%E0%B8%81%E0%B8%B2-3/

- (su-unsu) https://python3.wannaphong.com/2016/01/machine-learning-scikit-learn.html