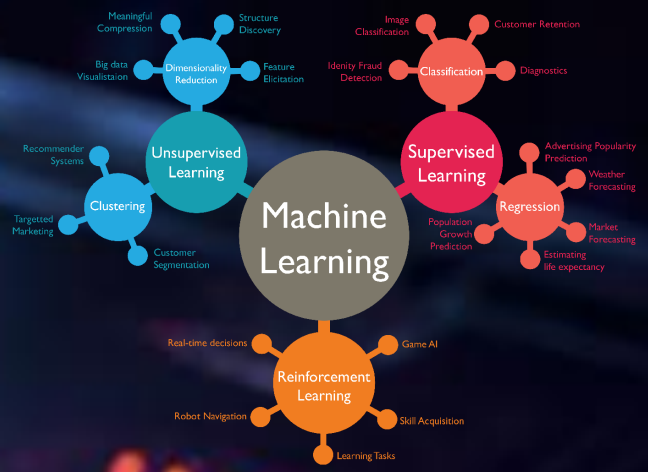


บทที่ 7

การจำแนกข้อมูลด้วยวิธี Random Forest (Data Classification with Random Forest)

วิชา การวิเคราะห์ข้อมูลขั้นสูง (6608202) (Advanced Data Analytics)

Asst. Prof. Dr. Nattapong Songneam

xnattapong@hotmail.com

http://www.siam2dev.com

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Agenda

0: ทบทวน
7.1 ความหมายของ Random Forest
7.2 คุณสมบัติหลักของ Random Forest
7.3 กระบวนการทำงานของ Random Forest
7.4 การจำแนกข้อมูล (Data Classification) ด้วยวิธี Random Forest
7.5 ขั้นตอนในการจำแนกข้อมูลด้วย Random Forest
7.6 ข้อดีของการใช้ Random Forest สำหรับการจำแนกข้อมูล
7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest
7.8 สรุป

7

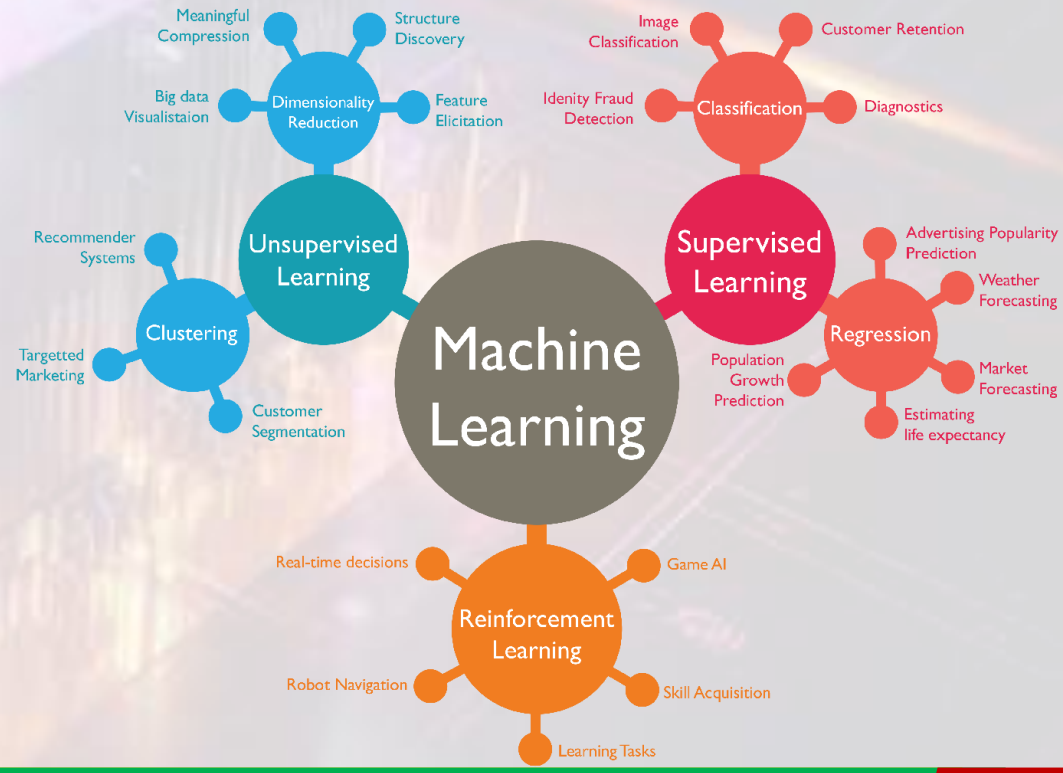
บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

MLM (Machine Learning Model) คืออะไร

MLM (Machine Learning Model) คืออะไร

MLM (Machine Learning Model) หรือแบบจำลองการเรียนรู้ของเครื่อง คือการสร้างระบบที่สามารถเรียนรู้และทำนายข้อมูลได้จากข้อมูลที่ได้รับการฝึกฝนมาก่อนหน้านี้ ด้วยการใช้เทคนิคต่าง ๆ ทางด้านวิทยาศาสตร์ข้อมูลและสถิติ แบบจำลองการเรียนรู้ของเครื่องสามารถนำไปประยุกต์ใช้ในหลายๆ ด้าน เช่น การทำนายผลลัพธ์ การจัดหมวดหมู่ การวิเคราะห์ข้อมูลเชิงลึก และอื่นๆ



7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

ประเภทของแบบจำลองการเรียนรู้ของเครื่อง

1. การเรียนรู้แบบมีผู้สอน (Supervised Learning): แบบจำลองที่ฝึกฝนจากข้อมูลที่มีป้ายกำกับ (labeled data) เช่น การทำนายราคาบ้านจากข้อมูลราคาที่มีอยู่
2. การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) : แบบจำลองที่ฝึกฝนจากข้อมูลที่ไม่มีป้ายกำกับ เช่น การจัดกลุ่มลูกค้าตามพฤติกรรมการซื้อ
3. การเรียนรู้แบบกึ่งผู้สอน (Semi-Supervised Learning): ผสมผสานระหว่างข้อมูลที่มีป้ายกำกับและไม่มีป้ายกำกับ
4. การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) : แบบจำลองที่เรียนรู้ผ่านการทดลองและการตอบสนองจากสภาพแวดล้อม

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

การแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (test set)

การแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (test set) เป็นกระบวนการสำคัญในขั้นตอนการสร้างและประเมินโมเดลการวิเคราะห์ข้อมูลหรือการเรียนรู้ของเครื่อง (Machine Learning) โดยมีรายละเอียดดังนี้:

1. **ชุดข้อมูลฝึกสอน (Training Set)**
2. **ชุดข้อมูลทดสอบ (Test Set)**

การแบ่งข้อมูลเป็นขั้นตอนสำคัญที่ช่วยให้โมเดลการเรียนรู้ของเครื่องมีความน่าเชื่อถือและมีประสิทธิภาพในการนำไปใช้งานจริง

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

การแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (test set)

1. ชุดข้อมูลฝึกสอน (Training Set)

ความหมาย: ชุดข้อมูลฝึกสอนเป็นชุดข้อมูลที่ใช้ในการฝึกสอนโมเดล โดยโมเดลจะทำการเรียนรู้จากข้อมูลชุดนี้ ซึ่งหมายถึงการทำความเข้าใจรูปแบบ (patterns) และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ เช่น ข้อมูลอินพุต (input features) และผลลัพธ์ (output labels) ที่ถูกต้อง

การใช้งาน: ในกระบวนการฝึกสอน โมเดลจะปรับพารามิเตอร์ภายในเพื่อให้สามารถทำนายผลลัพธ์จากข้อมูลอินพุตได้อย่างถูกต้องมากที่สุด

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

การแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (test set)

2. ชุดข้อมูลทดสอบ (Test Set)

ความหมาย: ชุดข้อมูลทดสอบเป็นชุดข้อมูลที่ใช้ในการประเมินประสิทธิภาพของโมเดลหลังจากที่ได้ทำการฝึกสอนด้วยข้อมูลฝึกสอนแล้ว ชุดข้อมูลทดสอบนี้จะไม่ถูกใช้ในขั้นตอนการฝึกสอน เพื่อให้สามารถตรวจสอบได้ว่าโมเดลสามารถทำงานได้ดีเพียงใดกับข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน

การใช้งาน: ใช้เพื่อวัดความแม่นยำและประสิทธิภาพของโมเดลในการทำนายข้อมูลที่ไม่รู้จัก ทำให้สามารถประเมินได้ว่าโมเดลมีความสามารถในการทั่วไป (generalization) ดีเพียงใด

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

การแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (test set)

การแบ่งสัดส่วน (Split Data)

การแบ่งข้อมูลออกเป็น training set และ test set มักจะใช้สัดส่วนต่าง ๆ ตามปริมาณข้อมูลและลักษณะของปัญหา เช่น:

70/30: 70% ของข้อมูลสำหรับการฝึกสอน และ 30% สำหรับการทดสอบ
80/20: 80% สำหรับการฝึกสอน และ 20% สำหรับการทดสอบ
90/10: 90% สำหรับการฝึกสอน และ 10% สำหรับการทดสอบ

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

วิธี K-Fold Cross-Validation

K-Fold Cross-Validation เป็นเทคนิคที่ใช้เพื่อประเมินประสิทธิภาพของโมเดลการเรียนรู้ของเครื่อง (machine learning) โดยไม่ต้องใช้ข้อมูลทั้งหมดในการฝึกสอนและทดสอบเพียงครั้งเดียว ช่วยให้การประเมินมีความแม่นยำและน่าเชื่อถือมากขึ้น โดยเฉพาะในกรณีที่ข้อมูลมีขนาดไม่มากพอหรือมีความไม่สมดุลในการกระจายของข้อมูล

ขั้นตอนการทำ K-Fold Cross-Validation

1. แบ่งข้อมูลออกเป็น K ส่วน (Folds)
    1. ข้อมูลทั้งหมดจะถูกแบ่งออกเป็น K ส่วน (folds) ที่มีขนาดใกล้เคียงกัน แต่ละ fold จะมีการกระจายของข้อมูลที่หลากหลายและครอบคลุม
    2. ตัวอย่างเช่น หากมี 100 ตัวอย่างข้อมูลและเลือก K = 5 ข้อมูลจะถูกแบ่งออกเป็น 5 fold แต่ละ fold จะมี 20 ตัวอย่าง
2. วนรอบการฝึกสอนและทดสอบ K ครั้ง
    1. ในแต่ละรอบ (iteration) หนึ่งใน K fold จะถูกใช้เป็นชุดทดสอบ (Testing Set) และอีก K-1 fold ที่เหลือจะถูกใช้เป็นชุดฝึกสอน (Training Set)
    2. โมเดลจะถูกฝึกด้วยข้อมูล K-1 fold และทดสอบด้วย fold ที่ถูกเลือกให้เป็นชุดทดสอบ
    3. ทำเช่นนี้ซ้ำไปเรื่อย ๆ จนครบทั้ง K fold โดยในแต่ละครั้ง fold ที่ใช้เป็นชุดทดสอบจะแตกต่างกัน
3. เฉลี่ยผลการทดสอบ
    1. หลังจากการทดสอบ K ครั้งเสร็จสิ้น จะนำผลการทดสอบจากแต่ละรอบมาคำนวณค่าเฉลี่ย เพื่อให้ได้ประเมินประสิทธิภาพของโมเดลโดยรวม
    2. ค่าเฉลี่ยนี้สามารถเป็นค่าความแม่นยำ (accuracy), ค่าเฉลี่ยความคลาดเคลื่อน (mean error), หรือค่าประสิทธิภาพอื่น ๆ ที่เหมาะสมกับงานที่ทำ

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

วิธี K-Fold Cross-Validation

ข้อดีของ K-Fold Cross-Validation

- ลด Bias: โดยการใช้ข้อมูลทั้งหมดในการฝึกสอนและทดสอบ ทำให้ผลการประเมินมีความเป็นกลางมากขึ้น
- ลด Variance: เนื่องจากใช้ข้อมูลหลายชุดในการฝึกสอนและทดสอบ ผลลัพธ์ที่ได้จึงมีความน่าเชื่อถือมากขึ้นและมีความเสถียร
- ใช้ข้อมูลอย่างมีประสิทธิภาพ: K-Fold ใช้ข้อมูลทุกตัวอย่างในการฝึกสอนและทดสอบ ทำให้ไม่สูญเสียข้อมูลในการประเมิน

ข้อเสียของ K-Fold Cross-Validation

- ใช้เวลามากขึ้น: เนื่องจากต้องทำการฝึกสอนและทดสอบ K ครั้ง เวลาในการคำนวณจะเพิ่มขึ้นอย่างมากโดยเฉพาะกับข้อมูลขนาดใหญ่
- การปรับพารามิเตอร์ที่ซับซ้อน: การเลือกค่า K ที่เหมาะสมอาจต้องทดลองหลายครั้ง และในบางกรณี K ที่แตกต่างกันอาจให้ผลลัพธ์ที่แตกต่างกัน

การเลือกค่า K ที่เหมาะสม

- โดยทั่วไป ค่า K ที่ใช้บ่อยคือ 5 หรือ 10 เนื่องจากให้ความสมดุลระหว่างความแม่นยำและเวลาที่ใช้ในการคำนวณ
- สำหรับข้อมูลขนาดใหญ่ที่ใช้เวลามาก อาจเลือก K ที่น้อยลง เช่น K = 3
- สำหรับข้อมูลขนาดเล็ก การใช้ค่า K ที่มากขึ้น (เช่น K = 10) อาจช่วยเพิ่มความแม่นยำในการประเมิน

K-Fold Cross-Validation เป็นวิธีที่มีประสิทธิภาพและเป็นที่นิยมในการประเมินโมเดล สามารถใช้กับปัญหาต่าง ๆ ทั้งการจำแนก (classification) และการทำนายค่า (regression)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

Reviews

การแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (test set)

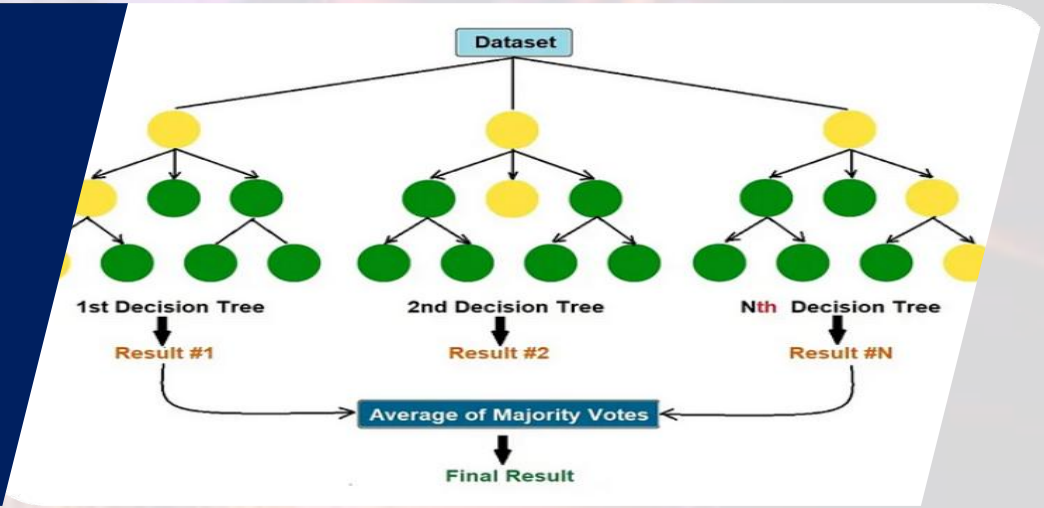
เหตุผลในการแบ่งข้อมูล

- ลดการเกิด Overfitting หากไม่มีการแบ่งข้อมูล โมเดลอาจเรียนรู้เฉพาะรูปแบบจากชุดข้อมูลฝึกสอนเพียงอย่างเดียว ซึ่งอาจทำให้เกิด Overfitting หรือการทำงานไม่ดีเมื่อเจอกับข้อมูลใหม่
- เพิ่มความเชื่อมั่นในผลลัพธ์ การมีชุดข้อมูลทดสอบช่วยให้สามารถตรวจสอบได้ว่าโมเดลมีความสามารถในการทั่วไปที่ดี ไม่ได้จำแค่ข้อมูลฝึกสอนเพียงอย่างเดียว

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

การจำแนกข้อมูลด้วยวิธี Random Forest

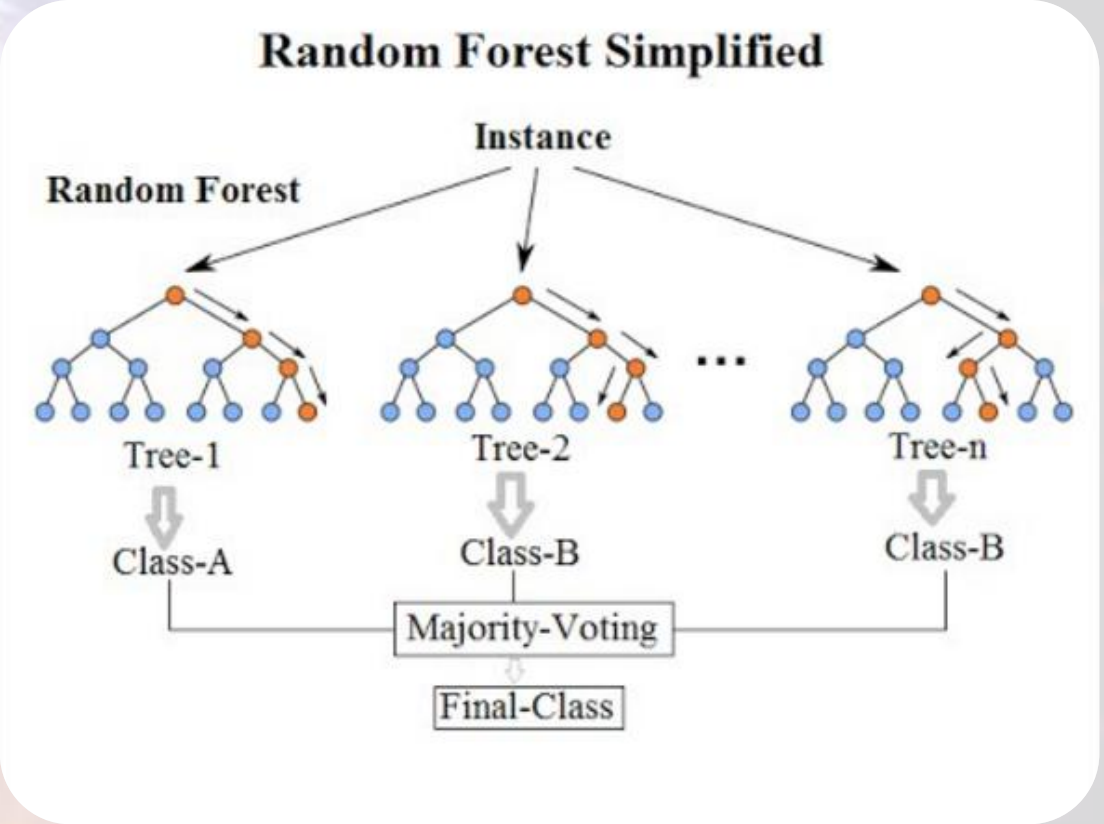


7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.1 ความหมายของ Random Forest

Random Forest คือ อัลกอริธึมการเรียนรู้ของเครื่อง (Machine Learning) แบบ Supervised Learning ที่ใช้สำหรับการจำแนกข้อมูล (classification) และการทำนาย (regression) โดยเป็นการรวมผลลัพธ์จากหลายๆ ต้นไม้ตัดสินใจ (Decision Trees) ที่ทำงานร่วมกัน เพื่อให้ได้ผลลัพธ์ที่แม่นยำและเชื่อถือได้มากขึ้น



7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.1 ความหมายของ Random Forest

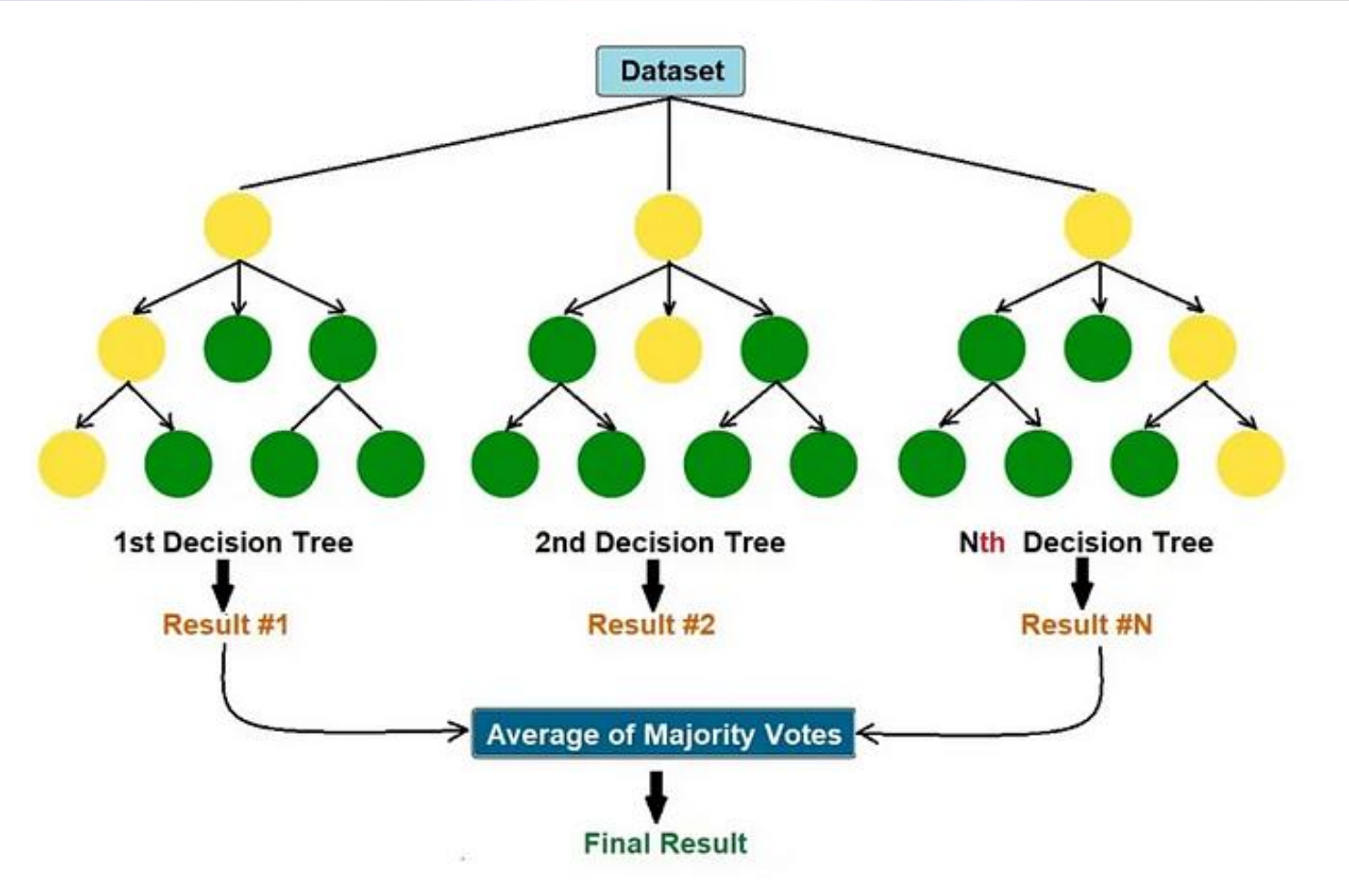
Random Forest หรือ "ป่าการสุ่ม" หรือ "ป่าแบบสุ่ม" เป็นวิธีการทางสถิติและการเรียนรู้ของเครื่อง (machine learning) ที่ใช้สำหรับการจำแนกประเภท (classification) และการทำนายค่า (regression) โดยการสร้างโมเดลการตัดสินใจหลายต้น (decision trees) และรวมผลการทำนายจากต้นไม้หลาย ๆ ต้นเข้าด้วยกันเพื่อลดความผิดพลาดของการทำนายและเพิ่มความแม่นยำ

วิธีการ Random Forest ทำงานโดยการสร้างต้นไม้การตัดสินใจจำนวนมากจากชุดข้อมูลที่แตกต่างกันโดยการสุ่มตัวอย่างข้อมูลและคุณลักษณะ (features) จากข้อมูลดั้งเดิม เมื่อได้รับผลจากต้นไม้แต่ละต้น ระบบจะใช้วิธีการลงคะแนนเสียง (voting) หรือการเฉลี่ย (ในกรณีของ regression) เพื่อให้ได้ผลลัพธ์สุดท้ายที่เป็นการทำนายที่แม่นยำและมีความเสถียรมากขึ้น

Random Forest ถูกใช้อย่างแพร่หลายในการแก้ปัญหาต่าง ๆ เช่น การจำแนกรูปภาพ, การพยากรณ์โรค, การวิเคราะห์ตลาด และอื่น ๆ เนื่องจากมีความทนทานต่อ overfitting และสามารถทำงานได้ดีแม้ในกรณีที่มีคุณลักษณะที่ไม่สำคัญอยู่มากในชุดข้อมูล

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest



7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.2 คุณสมบัติหลักของ Random Forest

1. Ensemble Method Random Forest ใช้เทคนิคที่เรียกว่า "Bagging" (Bootstrap Aggregating) โดยการสร้างต้นไม้ตัดสินใจหลาย ๆ ต้นจากการสุ่มตัวอย่างข้อมูลและคุณลักษณะ (features) ที่แตกต่างกันในแต่ละต้น จากนั้นใช้การโหวตส่วนใหญ่ (majority vote) ในการเลือกผลลัพธ์ที่ดีที่สุด

2. ลดการ Overfitting เนื่องจาก Random Forest เป็นการรวมผลลัพธ์จากหลาย ๆ ต้นไม้ การคาดการณ์จะมีความเสถียรมากขึ้นและมีโอกาสเกิดการ overfitting น้อยกว่าการใช้ต้นไม้ตัดสินใจเพียงต้นเดียว

3. ความสามารถในการจัดการข้อมูลจำนวนมาก Random Forest สามารถจัดการกับข้อมูลที่มีจำนวนคุณลักษณะ (features) มาก ๆ ได้ดี โดยเฉพาะในกรณีที่ข้อมูลมีความซับซ้อนหรือมี noise

4. การจัดอันดับคุณลักษณะ Random Forest สามารถใช้ในการประเมินความสำคัญของคุณลักษณะ (feature importance) ซึ่งเป็นประโยชน์ในการเลือกคุณลักษณะที่สำคัญที่สุดสำหรับการสร้างโมเดล

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.3 กระบวนการทำงานของ Random Forest

7.3 กระบวนการทำงานของ Random Forest

1. การสุ่มตัวอย่างข้อมูล (Bootstrap Sampling) สุ่มตัวอย่างข้อมูลจากชุดข้อมูลฝึกสอน โดยบางข้อมูลอาจถูกเลือกซ้ำหลายครั้ง

2. การสร้างต้นไม้ตัดสินใจ สำหรับแต่ละตัวอย่างที่สุ่มมา จะสร้างต้นไม้ตัดสินใจโดยใช้ชุดคุณลักษณะที่สุ่มเลือกบางส่วนเท่านั้น

3. การรวมผลลัพธ์ เมื่อสร้างต้นไม้ตัดสินใจครบทุกต้นแล้ว ผลลัพธ์จากแต่ละต้นไม้จะถูกนำมารวมกัน โดยใช้การโหวตส่วนใหญ่ (สำหรับการจำแนกข้อมูล) หรือค่าเฉลี่ย (สำหรับการทำนาย) เพื่อให้ได้ผลลัพธ์สุดท้าย

Random Forest เป็นอัลกอริธึมที่มีประสิทธิภาพและใช้งานได้หลากหลาย ทำให้เป็นที่นิยมอย่างมากในงานด้านข้อมูลวิทยาศาสตร์ (data science) และการวิเคราะห์ข้อมูล

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.4 การจำแนกข้อมูล (Data Classification) ด้วยวิธี Random Forest

7.4 การจำแนกข้อมูล (Data Classification) ด้วยวิธี Random Forest

การจำแนกข้อมูล (Data Classification) ด้วยวิธี Random Forest เป็นกระบวนการที่ใช้ในการทำนายหรือจัดกลุ่มข้อมูลโดยอาศัยการรวมผลลัพธ์จากต้นไม้ตัดสินใจหลายๆ ต้น เพื่อเพิ่มความแม่นยำในการทำนายและลดปัญหา overfitting ที่มักเกิดขึ้นเมื่อใช้ต้นไม้ตัดสินใจเพียงต้นเดียว

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.5 ขั้นตอนในการจำแนกข้อมูลด้วย Random Forest

1. เตรียมชุดข้อมูล

- แบ่งชุดข้อมูลเป็นสองส่วน: ชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (test set) เพื่อประเมินประสิทธิภาพของโมเดล
- ข้อมูลที่ใช้ประกอบด้วยคุณลักษณะ (features) และคลาสหรือป้ายกำกับ (labels) ที่ต้องการจำแนก

2. สร้างต้นไม้ตัดสินใจ (Decision Trees)

- Random Forest จะสร้างต้นไม้ตัดสินใจหลายต้น โดยแต่ละต้นจะถูกสร้างจากการสุ่มเลือกตัวอย่างข้อมูลจากชุดข้อมูลฝึกสอน และสุ่มเลือกคุณลักษณะที่แตกต่างกันในแต่ละต้น
- ในการสร้างแต่ละต้นไม้ ตัดสินใจโดยใช้การแบ่งข้อมูลซ้ำ ๆ กันตามคุณลักษณะต่าง ๆ จนกระทั่งถึงเงื่อนไขการหยุด เช่น ต้นไม้ถึงความลึกสูงสุด หรือไม่มีข้อมูลให้แบ่งอีกต่อไป

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.5 ขั้นตอนในการจำแนกข้อมูลด้วย Random Forest (ต่อ)

3. ทำนายข้อมูลใหม่ (Prediction)

- เมื่อสร้างต้นไม้ตัดสินใจครบแล้ว โมเดลจะใช้ต้นไม้เหล่านี้ในการทำนายข้อมูลใหม่ โดยการให้แต่ละต้นไม้ทำนายว่าตัวอย่างข้อมูลที่เข้ามาอยู่ในคลาสใด
- การตัดสินใจสุดท้ายจะทำโดยการใช้วิธีการโหวตส่วนใหญ่ (majority vote) เช่น หากต้นไม้ส่วนใหญ่ทำนายว่าตัวอย่างข้อมูลนั้นอยู่ในคลาส A โมเดลก็จะทำนายว่าข้อมูลนั้นอยู่ในคลาส A

4. ประเมินประสิทธิภาพของโมเดล

- ใช้ชุดข้อมูลทดสอบในการทดสอบโมเดลที่สร้างขึ้น
- ประเมินผลลัพธ์โดยใช้ตัวชี้วัดต่างๆ เช่น ความแม่นยำ (accuracy), precision, recall, และ F1-score เพื่อดูว่าโมเดลมีประสิทธิภาพเพียงใดในการจำแนกข้อมูล

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.6 ข้อดีของการใช้ Random Forest สำหรับการจำแนกข้อมูล

7.6 ข้อดีของการใช้ Random Forest สำหรับการจำแนกข้อมูล

- ความแม่นยำสูง ด้วยการรวมผลลัพธ์จากต้นไม้หลายๆ ต้น ทำให้โมเดลสามารถต้านทาน noise ในข้อมูลได้ดีขึ้น และมีความแม่นยำสูงขึ้นเมื่อเทียบกับการใช้ต้นไม้ตัดสินใจเพียงต้นเดียว
- ลดปัญหา Overfitting Random Forest ลดความเสี่ยงของการ overfitting ที่มักเกิดขึ้นเมื่อใช้โมเดลที่ซับซ้อนหรือมีความลึกมาก
- จัดการกับข้อมูลที่มีคุณลักษณะจำนวนมาก Random Forest สามารถจัดการกับชุดข้อมูลที่มีคุณลักษณะหลากหลายได้อย่างมีประสิทธิภาพ

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

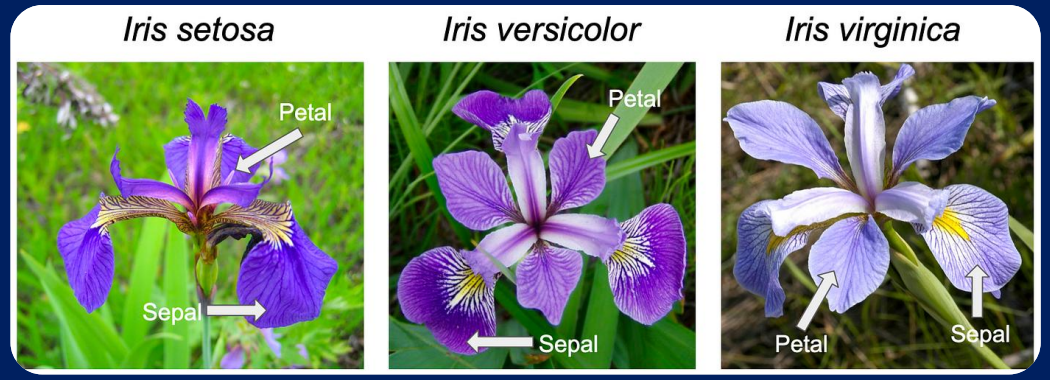
7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

ตัวอย่างข้อมูลสำหรับการจำแนกข้อมูลด้วย Random Forest สามารถใช้ได้หลายประเภท ขึ้นอยู่กับโจทย์ที่ต้องการแก้ไข นี่คือตัวอย่างข้อมูลที่สามารถใช้ได้สำหรับการจำแนกข้อมูล

ตัวอย่างที่ 1: การจำแนกข้อมูลชนิดของดอกไม้ (Iris Dataset)

Iris dataset เป็นหนึ่งในชุดข้อมูลที่เป็นที่รู้จักกันดี ใช้ในการจำแนกชนิดของดอกไม้จากคุณลักษณะต่าง ๆ เช่น ความยาวและความกว้างของกลีบ



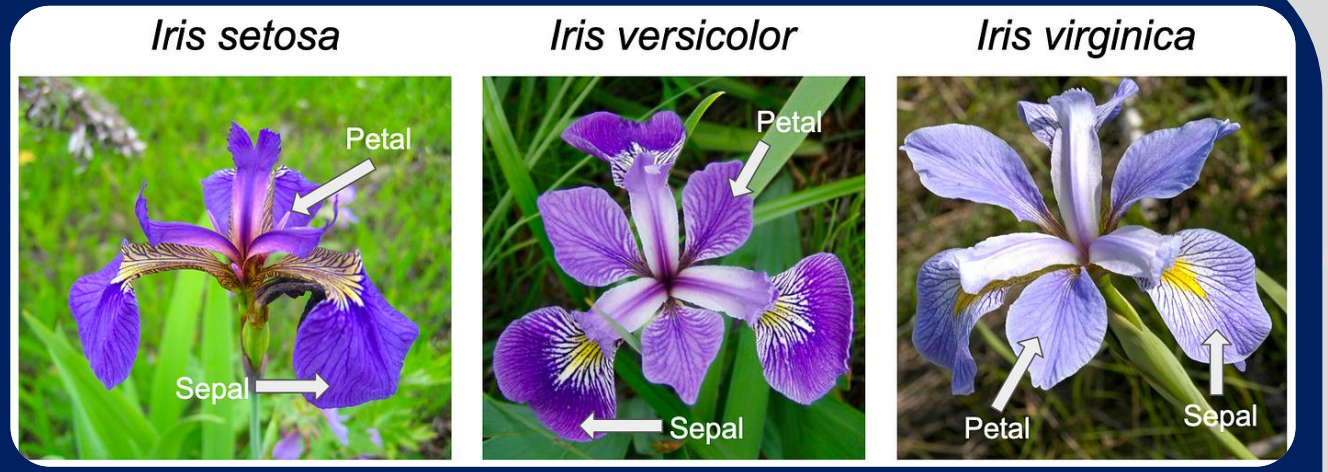
7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

คุณลักษณะ (Features)

- ความยาวของกลีบเลี้ยง (sepal length)
- ความกว้างของกลีบเลี้ยง (sepal width)
- ความยาวของกลีบดอก (petal length)
- ความกว้างของกลีบดอก (petal width)



ป้ายกำกับ (Labels)

- ชนิดของดอกไม้ เช่น Setosa, Versicolor, Virginica

7 บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

ตัวอย่างข้อมูล 10 รายการแรกใน Iris dataset:

Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Target
5.1	3.5	1.4	0.2	0
4.9	3.0	1.4	0.2	0
4.7	3.2	1.3	0.2	0
4.6	3.1	1.5	0.2	0
5.0	3.6	1.4	0.2	0
5.4	3.9	1.7	0.4	0
4.6	3.4	1.4	0.3	0
5.0	3.4	1.5	0.2	0
4.4	2.9	1.4	0.2	0
4.9	3.1	1.5	0.1	0

คำอธิบาย:

1. **Feature Names:**
   - `Sepal Length (cm)` (ความยาวกลีบดอก)
   - `Sepal Width (cm)` (ความกว้างกลีบดอก)
   - `Petal Length (cm)` (ความยาวกลีบเลี้ยง)
   - `Petal Width (cm)` (ความกว้างกลีบเลี้ยง)
2. **Target (0, 1, 2):**
   - 0: Setosa
   - 1: Versicolor
   - 2: Virginica

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

File: Lec07_01_Random_Forest_Iris_dataset.py

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris  # ตัวอย่างการใช้ข้อมูล
# สมมุติว่ามีชุดข้อมูล X และ y
# ในกรณีนี้ ใช้ข้อมูลตัวอย่างจาก sklearn
data = load_iris()
X = data.data
y = data.target
# แบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# สร้างโมเดล Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
# ฝึกสอนโมเดล
model.fit(X_train, y_train)
# ทำนายชุดข้อมูลทดสอบ
y_pred = model.predict(X_test)
# ประเมินผลลัพธ์
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PC

PS D:\DataAnalytics_Project> & C:/Users/User/Appl
Accuracy: 100.00%
PS D:\DataAnalytics_Project>
```

ตัวอย่างที่ 1: จำแนกกลุ่มของดอกไม้ (Iris)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

File: Lec07_02_Random_Forest_Iris_dataset_Predict.py

ตัวอย่างที่ 2: ทำนายชนิดของดอกไม้ (Iris)

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris

# Load Iris dataset
data = load_iris()
X = data.data
y = data.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Create and train the Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100))p=obf

# Add unknown data points
unknown_data = [
    [5.1, 3.5, 1.4, 0.2],  # Example 1
    [6.2, 3.4, 5.4, 2.3],  # Example 2
    [5.9, 3.0, 4.2, 1.5]   # Example 3
]

# Predict and display results for each unknown data point
for i, sample in enumerate(unknown_data, start=1):
    prediction = model.predict([sample])[0]  # Predict class for the sample
    print(f"Unknown data point {i}: {sample} -> Predicted class: {prediction}")
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

File: Lec07_02_Random_Forest_Iris_dataset_Predict.py

ผลลัพธ์

```
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> & C:/User
ec07_01_Random_Forest_Iris_dataset.py
Accuracy: 100.00%
Unknown data point 1: [5.1, 3.5, 1.4, 0.2] -> Predicted class: 0
Unknown data point 2: [6.2, 3.4, 5.4, 2.3] -> Predicted class: 2
Unknown data point 3: [5.9, 3.0, 4.2, 1.5] -> Predicted class: 1
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects>
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

การพยากรณ์สภาพอากาศ (Weather Prediction)

การพยากรณ์สภาพอากาศ (Weather Prediction) เป็นการนำข้อมูลเกี่ยวกับสภาพอากาศ เช่น อุณหภูมิ, ความชื้น, ลม, และอื่นๆ มาวิเคราะห์และจำแนกผลลัพธ์ให้เป็นประเภทต่าง ๆ เช่น ท้องฟ้าปลอดโปร่ง (Clear), มีเมฆบางส่วน (Partly Cloudy), ฝนตก (Rain), หรือหิมะตก (Snow) เป็นต้น สมมติว่ามีชุดข้อมูลเกี่ยวกับสภาพอากาศ ซึ่งสามารถใช้ในการจำแนกว่าวันนั้น ๆ จะมีฝนตกหรือไม่

- คุณลักษณะ (Features)
  - อุณหภูมิ (Temperature)
  - ความชื้น (Humidity)
  - ความเร็วลม (Wind Speed)
  - ความกดอากาศ (Pressure)
- ป้ายกำกับ (Labels)
  - จะมีฝนตก (Yes) หรือไม่ (No)

ตัวอย่างที่ 3: การพยากรณ์สภาพอากาศ (Weather Prediction)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part01 Lec07_03_Random_Forest_ Weather_Dataset.py

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# สมมติว่าเรามีข้อมูลสภาพอากาศในรูปแบบ DataFrame
data = {
    'Temperature': [30, 25, 20, 28, 22, 19, 35],
    'Humidity': [70, 80, 90, 60, 85, 75, 55],
    'Wind Speed': [5, 3, 8, 6, 7, 2, 9],
    'Pressure': [1010, 1020, 1005, 1015, 1008, 1012, 1013],
    'Rain': ['No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'No']
}
df = pd.DataFrame(data)
```

ตัวอย่างที่ 3: การพยากรณ์สภาพอากาศ (Weather Prediction)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part02 Lec07_03_Random_Forest_ Weather_Dataset.py

ตัวอย่างที่ 3: การจำแนกสภาพอากาศ (Weather Dataset)

```
df = pd.DataFrame(data)
# แปลงป้ายกำกับเป็นตัวเลข
df['Rain'] = df['Rain'].map({'No': 0, 'Yes': 1})
# แบ่งข้อมูลเป็นคุณลักษณะและป้ายกำกับ
X = df.drop('Rain', axis=1)
y = df['Rain']
# แบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# สร้างโมเดล Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
# ฝึกสอนโมเดล
model.fit(X_train, y_train)
# ทำนายชุดข้อมูลทดสอบ
y_pred = model.predict(X_test)
# ประเมินผลลัพธ์
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\DataAnalytics_Project> & C:/Users/User/AppData/
Accuracy: 66.67%
PS D:\DataAnalytics_Project>
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part01 Lec07_04_Random_Forest_ Weather_Dataset_Predict.py

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# สมมติว่าเรามีข้อมูลสภาพอากาศในรูปแบบ DataFrame
data = {
    'Temperature': [30, 25, 20, 28, 22, 19, 35],
    'Humidity': [70, 80, 90, 60, 85, 75, 55],
    'Wind Speed': [5, 3, 8, 6, 7, 2, 9],
    'Pressure': [1010, 1020, 1005, 1015, 1008, 1012, 1013],
    'Rain': ['No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'No']
}
```

ตัวอย่างที่ 4: การจำแนกสภาพอากาศ (Weather Dataset)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part02 Lec07_04_Random_Forest_ Weather_Dataset_Predict.py

ตัวอย่างที่ 4: การจำแนกสภาพอากาศ (Weather Dataset)

```
df = pd.DataFrame(data)
df = pd.DataFrame(data)
# แปลงป้ายกำกับเป็นตัวเลข
df['Rain'] = df['Rain'].map({'No': 0, 'Yes': 1})
# แบ่งข้อมูลเป็นคุณลักษณะและป้ายกำกับ
X = df.drop('Rain', axis=1)
y = df['Rain']
# แบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# สร้างโมเดล Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part03 Lec07_04_Random_Forest_ Weather_Dataset_Predict.py

```
# ฝึกสอนโมเดล
model.fit(X_train, y_train)
# ทำนายชุดข้อมูลทดสอบ
y_pred = model.predict(X_test)
# ประเมินผลลัพธ์
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100))
# ข้อมูล unknown 3 รายการ
unknown_data = pd.DataFrame({
    'Temperature': [27, 18, 32],
    'Humidity': [65, 95, 60],
    'Wind Speed': [4, 10, 7],
    'Pressure': [1011, 1006, 1014]
})
```

ตัวอย่างที่ 4: การจำแนกสภาพอากาศ (Weather Dataset)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part04 Lec07_04_Random_Forest_ Weather_Dataset_Predict.py

ตัวอย่างที่ 4: การจำแนกสภาพอากาศ (Weather Dataset)

```
# ทำนายข้อมูล unknown
predictions = model.predict(unknown_data)

# แปลงผลลัพธ์กลับเป็นข้อความ (0 -> 'No', 1 -> 'Yes')
predicted_classes = ['No' if pred == 0 else 'Yes' for pred in predictions]

# แสดงผลลัพธ์
for i, result in enumerate(predicted_classes, 1):
    print(f"Unknown data {i}: Predicted class -> {result}")
```

```
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> & C:/Use
Lec07_03_Random_Forest_ Weather_Dataset.py"
Accuracy: 66.67%
Unknown data 1: Predicted class -> No
Unknown data 2: Predicted class -> Yes
Unknown data 3: Predicted class -> No
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects>
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

การจำแนกการตรวจจับสแปม (Spam Detection)

ข้อมูลเกี่ยวกับอีเมลสามารถใช้ในการจำแนกว่าสิ่งนั้นเป็นสแปม (spam) หรือไม่

ตัวอย่างที่ 5: การจำแนกการตรวจจับสแปม (Spam Detection)

- คุณลักษณะ (Features)
  - จำนวนคำที่เป็นคำสำคัญ (number of important words)
  - ความยาวของข้อความ (length of the message)
  - การมีคำหรือวลีที่ใช้บ่อยในสแปม (presence of specific words or phrases)

- ป้ายกำกับ (Labels)
  - เป็นสแปม (Spam) หรือไม่เป็นสแปม (Not Spam)

สำหรับการจำแนกการตรวจจับสแปม (Spam Detection) คุณสามารถใช้ชุดข้อมูลที่เป็นที่รู้จักและใช้กันอย่างแพร่หลาย เช่น "SMS Spam Collection Dataset" หรือ "UCI Machine Learning Repository's Spambase dataset" ซึ่งประกอบด้วยข้อมูลอีเมลที่ถูกแท็กว่าเป็นสแปมหรือไม่เป็นสแปม

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

ตัวอย่างที่ 5: การจำแนกการตรวจจับสแปม (Spam Detection)

ID	Number of Important Words	Message Length	Presence of Spam Words	Label
1	5	120	1	Spam
2	2	85	0	Not Spam
3	8	150	1	Spam
4	3	95	0	Not Spam
5	6	200	1	Spam
6	1			
7	7			
8	4	110		
9	5	135		Spam
10	2	75	0	Not Spam
11	9	210	1	Spam
12	3	90	0	Not Spam
13	6	170	1	Spam
14	1	55	0	Not Spam
15	7	190	1	Spam
16	4	115	0	Not Spam
17	8	220	1	Spam
18	2	80	0	Not Spam
19	5	130	1	Spam
20	3	100	0	Not Spam

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part01 Lec07_05_Random_Forest_Spam_SMS.py

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import zipfile
import io
import requests

# โหลดข้อมูลจาก UCI dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00228/smsspamcollection.zip"
response = requests.get(url)
```

ตัวอย่างที่ 5: การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

```
with zipfile.ZipFile(io.BytesIO(response.content)) as z:
    # เปิดไฟล์เฉพาะภายใน ZIP ที่ต้องการ (SMSSpamCollection)
    with z.open("SMSSpamCollection") as f:
        df = pd.read_csv(f, sep='\t', names=["label", "message"], encoding='latin-1')

# เปลี่ยนป้ายกำกับเป็นตัวเลข
df['label'] = df['label'].map({'ham': 0, 'spam': 1})
# แสดงข้อมูลเบื้องต้น
print(df.head())
# แยกคุณลักษณะ (Features) และป้ายกำกับ (Labels)
X = df['message']
y = df['label']
```

Part02 Lec07_05_Random_Forest_Spam_SMS.py

ตัวอย่างที่ 5: การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part03 Lec07_03_Random_Forest_Spam_SMS.py

```
# แปลงข้อความเป็นจำนวนคำ (Bag of Words)
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(X)
# แบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# สร้างโมเดล Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
# ฝึกสอนโมเดล
model.fit(X_train, y_train)
# ทำนายผลชุดข้อมูลทดสอบ
y_pred = model.predict(X_test)
# ประเมินผลลัพธ์
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100))
# รายงานการจำแนกผล
print(classification_report(y_test, y_pred))
```

ตัวอย่างที่ 3: การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

ตัวอย่างการใช้ Random Forest สำหรับจำแนกระดับของโรคเบาหวานโดยใช้ Python สามารถทำได้โดยใช้ scikit-learn และ pandas ในการโหลดและประมวลผลข้อมูล

1. เตรียมข้อมูล

เราจะใช้ชุดข้อมูลตัวอย่าง PIMA Indian Diabetes Dataset ซึ่งมีข้อมูลเกี่ยวกับโรคเบาหวาน หากคุณมีข้อมูลที่มีการแบ่งระดับของโรค (เช่น ปกติ, เบาหวานระดับต้น, เบาหวานระดับรุนแรง) ให้ใช้ข้อมูลดังกล่าว

File Lec07_06_Random_Forest_Spam_SMS.py

ตัวอย่างที่ 6: จำแนกระดับของโรคเบาหวาน (PIMA Indian Diabetes Dataset)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigree Function	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0						
2	197	70						
8	125	96						
4	110	92	0					
10	168	74	0					
10	139	80	0					
1	189	60	23	846	30			
5	166	72	19	175	2			
7	100	0	0	0				
0	118	84	47	230				
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1

File Lec07_06_Random_Forest_Spam_SMS.py

ตัวอย่างที่ 6: จำแนกระดับของโรคเบาหวาน (PIMA Indian Diabetes Dataset)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

File Lec07_06_Random_Forest_Spam_SMS.py

Column Name	Description
Pregnancies	จำนวนครั้งที่ตั้งครรภ์
Glucose	ระดับน้ำตาลในเลือด (mg/dL
BloodPressure	ความดันโลหิต (mm Hg)
SkinThickness	ความหนาของผิวหนัง (triceps skin fold th
Insulin	ระดับอินซูลินในเลือด (mu U/ml)
BMI	ค่าดัชนีมวลกาย (Body Mass Index)
DiabetesPedigreeFunction	ค่าแสดงแนวโน้มของพันธุกรรมเกี่ยวกับโรคเบาหวาน
Age	อายุของผู้ป่วย (ปี)
Outcome	เป็นเบาหวานหรือไม่ (0 = ไม่เป็น, 1 = เป็น)

ตัวอย่างที่ 6: จำแนกระดับของโรคเบาหวาน (PIMA Indian Diabetes Dataset)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part01 Lec07_06_Random_Forest_Diabetes.py

ตัวอย่างที่ 6: การจำแนกการตรวจจับสแปม (Spam Detection)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# โหลดข้อมูล
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
           'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

df = pd.read_csv(url, names=columns)

# แปลง Outcome (0 = ปกติ, 1 = เบาหวาน) ให้เป็นระดับ เช่น
df['Diabetes_Level'] = df['Outcome'].replace({0: 'Normal', 1: 'Diabetic'})
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part02 Lec07_06_Random_Forest_Diabetes.py

```
# ถ้าคุณมีระดับที่มากขึ้น เช่น "Pre-Diabetes" ให้สร้างเงื่อนไขเพิ่มเอง
# df['Diabetes_Level'] = pd.cut(df['Glucose'], bins=[0, 100, 140, np.inf], labels=['Normal', 'Pre-Diabetes', 'Diabetic'])

# แปลง Label เป็นตัวเลข
df['Diabetes_Level'] = df['Diabetes_Level'].astype('category').cat.codes  # แปลงเป็นตัวเลข 0,1,2

# แยก Features และ Target
X = df.drop(columns=['Outcome', 'Diabetes_Level'])  # ฟีเจอร์
y = df['Diabetes_Level']  # เป้าหมาย

# แบ่งข้อมูลเป็นชุด Train และ Test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# สร้างโมเดล Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
```

ตัวอย่างที่ 6: การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part03 Lec07_06_Random_Forest_Diabetes.py

ตัวอย่างที่ 6: การจำแนกการตรวจจับสแปม (Spam Detection)

```
# ทำนายผล
y_pred = rf.predict(X_test)
# ประเมินผล
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
# ข้อมูลตัวอย่าง (unknown) 5 รายการ
unknown_data = pd.DataFrame([
    [2, 120, 70, 22, 85, 28.1, 0.5, 32],
    [4, 145, 80, 32, 100, 33.2, 0.8, 45],
    [0, 90, 60, 18, 55, 25.6, 0.3, 29],
    [5, 175, 85, 35, 150, 37.1, 1.2, 50],
    [3, 110, 65, 20, 70, 27.5, 0.6, 40]
], columns=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
            'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age'])
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part04 Lec07_06_Random_Forest_Diabetes.py

```
# ทำนายผลลัพธ์
predictions = rf.predict(unknown_data)

# แปลงค่าที่ได้กลับเป็น Label
label_mapping = {0: 'Normal', 1: 'Diabetic'}
predicted_labels = [label_mapping[p] for p in predictions]

# เพิ่มผลลัพธ์ลงในตาราง
unknown_data['Predicted Diabetes Level'] = predicted_labels

# แสดงผลลัพธ์
print(unknown_data)
```

ตัวอย่างที่ 6: การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part03 Lec07_06_Random_Forest_Diabetes.py

3. อธิบายโค้ด

1.โหลดชุดข้อมูล PIMA Indian Diabetes Dataset ซึ่งมีข้อมูลเกี่ยวกั

2.กำหนดระดับของโรคเบาหวาน เช่น Normal, Pre-Diabetes, Diabetic

3.แปลง Label ให้อยู่ในรูปแบบที่โมเดลสามารถเรียนรู้ได้ (ใช้ .cat.codes)

4.ใช้ RandomForestClassifier ในการฝึกโมเดล และแบ่งข้อมูลเป็น 80% Train / 20% Test

5.ประเมินผลด้วย Accuracy และ Classification Report

อธิบายโค้ด การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part03 Lec07_06_Random_Forest_Diabetes.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
Accuracy: 0.7207792207792207
Classification Report:
              precision    recall  f1-score   support

           0       0.61      0.62      0.61        55
           1       0.79      0.78      0.78        99

    accuracy                           0.72       154
   macro avg       0.70      0.70      0.70       154
weighted avg       0.72      0.72      0.72       154

   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  DiabetesPedigreeFunction  Age Predicted Diabetes Level
0            2      120             70             22       85  28.1                       0.5   32                 Diabetic
1            4      145             80             32      100  33.2                       0.8   45                   Normal
2            0       90             60             18       55  25.6                       0.3   29                 Diabetic
3            5      175             85             35      150  37.1                       1.2   50                   Normal
4            3      110             65             20       70  27.5                       0.6   40                 Diabetic
PS C:\AppServ\www\siam2dev.net\E-Learning\DataMining\DM_Projects>
```

ผลลัพธ์ การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

Part03 Lec07_06_Random_Forest_Diabetes.py

ตัวอย่างที่ 6: การจำแนกการตรวจจับสแปม (Spam Detection)

```
# ทำนายผล
y_pr
# ปร
prin
prin
```

```
markdown

Accuracy: 0.81
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.85      0.82       103
           1       0.83      0.76      0.79        87

    accuracy                           0.81       190
   macro avg       0.81      0.81      0.81       190
weighted avg       0.81      0.81      0.81       190
```

```
3. อ
1.โห
2.กำ
3.แป
4.ใช้
5.ปร
```

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.7 ตัวอย่างการประยุกต์การจำแนกข้อมูลด้วย Random Forest

ผลลัพธ์ Lec07_03_Random_Forest_Spam_SMS.py

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONIT

PS D:\DataAnalytics_Project> & C:/Users/User/AppData/Local/Micros
   label                                                message
0      0  Go until jurong point, crazy.. Available only ...
1      0                      Ok lar... Joking wif u oni...
2      1  Free entry in 2 a wkly comp to win FA Cup fina...
3      0  U dun say so early hor... U c already then say...
4      0  Nah I don't think he goes to usf, he lives aro...
Accuracy: 97.55%
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      1448
           1       1.00      0.82      0.90       224

    accuracy                           0.98      1672
   macro avg       0.99      0.91      0.94      1672
weighted avg       0.98      0.98      0.97      1672

PS D:\DataAnalytics_Project>
```

ตัวอย่างที่ 3: การจำแนกการตรวจจับสแปม (Spam Detection)

7

บทที่ 7 การจำแนกข้อมูลด้วยวิธี Random Forest

7.8 สรุป

7.8 สรุป

Random Forest เป็นหนึ่งในเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ที่ได้รับความนิยมอย่างมาก สำหรับการจำแนกข้อมูล (Classification) รวมถึงการถดถอย (Regression) โดยเฉพาะในงานที่มีข้อมูลที่ซับซ้อนและไม่เป็นเชิงเส้น (Non-linear)

สรุปได้ว่า Random Forest เป็นเทคนิคที่มีประสิทธิภาพสูงและยืดหยุ่นในการจำแนกข้อมูล สามารถนำไปประยุกต์ใช้ได้กับงานที่หลากหลาย และเป็นตัวเลือกที่ดีเมื่อข้อมูลมีความซับซ้อนและมีความหลากหลายของคุณลักษณะ

อ้างอิง

1. เว็บไซต์
   - Itsci MJU
   - มหาวิทยาลัยแม่โจ้. (n.d.). *Itsci MJU*. สืบค้นจาก https://www.itsci.mju.ac.th
2. บทความจากเว็บไซต์ Softnix
   - จิราพร, ส. (2018, 6 กันยายน). *ว่าด้วย K-Means และการประยุกต์*. Softnix. สืบค้นจาก https://www.softnix.co.th/2018/09/06/ว่าด้วย-k-means-และการประยุกตร/
3. บทความจากเว็บไซต์ Coraline
   - ปิยะ, ร. (2018, 5 พฤศจิกายน). *Customer segmentation by clustering model*. Coraline. สืบค้นจาก https://www.coraline.co.th/single-post/2018/11/05/Customer-Segmentation-by-clustering-model