เอกสารรายงานโมเดลการเรียนรู้การใช้งานใบมีดเครื่องบดข้าวโพดสำหรับผลิตอาหารสัตว์

10 มีนาคม 2567

Document Control Preparation

Action	Name	Student ID	Date
Prepared by:	จิราพร สอนบุญมา	66130136	มีนาคม 2567
	ปัญญา หลีนวรัตน์	66130283	
	เพชรณพรรณ อภิโชคพลากรณ์	66130404	
	อารักษ์ ลิบน้อย	66130503	
	หทัยรัตน์ เจนวิทยา	66130662	

Version	Release Date	Change Notice	Pages Affected	Remarks
1.0	10 มีนาคม 2567	N/A	All	

โมเดลการเรียนรู้การใช้งานใบมีดเครื่องบดข้าวโพดสำหรับผลิตอาหารสัตว์

1. ที่มาและปัญหา

ในยุคปัจจุบันที่เกิดความต้องการในการผลิตอาหารสัตว์อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น เครื่องบดอาหารสัตว์เป็นอุปกรณ์ที่สำคัญในกระบวนการผลิต เครื่องบดอาหารสัตว์มักถูกใช้ในการบดวัตถุดิบที่หลากหลาย เช่น ข้าว, ข้าวโพด, และอื่น ๆ เพื่อให้ได้ผลิตภัณฑ์อาหารสัตว์ที่มีคุณภาพและปริมาณเพียงพอสำหรับการใช้งานในฟาร์มหรือโรงงานการผลิตอาหารสัตว์ต่าง ๆ

อย่างไรก็ตาม ในการใช้งานเครื่องบดอาหารสัตว์ เราพบว่าใบมีดมักมีปัญหาเกี่ยวกับความเสื่อมสภาพ ซึ่งอาจส่งผลให้ผลผลิตภัณฑ์ที่ได้มีคุณภาพลดลง นอกจากนี้ การเสื่อมสภาพของใบมีดยังอาจเป็นสาเหตุให้เกิดอุบัติเหตุหรือบาดเจ็บขณะใช้งานได้อีกด้วย

โรงงานทำอาหารสำเร็จรูป จะมีการใช้งานเครื่องบดอาหารสัตว์เพื่อที่จะนำผลผลิตไปแปรรูปเป็นอาหารรูปแบบอื่นๆ ซึ่งการใช้งานเครื่องบดอาหารนั้นจะได้ปริมาณ output ผลผลิต ที่สม่ำเสมอ แต่ถ้าหากใบมีดสำหรับเครื่องบดอาหารนั้นเริ่มเสื่อมสภาพ จะทำให้ได้ output ผลผลิตออกมาไม่คงที่ และสิ้นเปลืองพลังงานไฟฟ้า

โดยปกติแล้วทางโรงงานจะแก้ปัญหาโดยการหาก output ผลผลิตลดลง ทางโรงงานก็จะทำการตรวจสอบดังนี้

1. เช็คกระแสไฟหากพบว่ามีการใช้กระแสไฟที่สูง (จาก Dashboard Monitoring) เกินไปกว่าปกติ ซึ่งสวนทางกับ output ผลผลิตที่ได้ จึงจะทำการแก้ไขโดยการเปลี่ยนใบมีดสำหรับเครื่องบดอาหาร
2. มีการตรวจสอบวิธีอื่นๆด้วยเพื่อยืนยันว่าไม่ได้เป็นที่ปัญหาอื่นในการทำให้ output ผลผลิตออกมาน้อยลง
3. เปลี่ยนใบมีด

วิธีการแก้ปัญหาที่ถูกต้องควรจะเป็นการ Predict Grinding Machine เพื่อทำนายคุณภาพใบมีดสำหรับเครื่องบดอาหาร ว่ามีอายุการใช้งานได้ถึงเมื่อไหร่ และควรเปลี่ยนเมื่อไหร่ เพื่อจะทำให้อัตราการใช้พลังงานไฟฟ้าไม่สิ้นเปลือง และ output ผลผลิตออกมาอย่างสม่ำเสมอ

2. วัตถุประสงค์

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาโมเดลทำนายความเสื่อมสภาพของใบมีดในเครื่องบดอาหารสัตว์ โดยใช้ข้อมูลประวัติของการใช้งาน รวมถึงสภาพแวดล้อมและเงื่อนไขการทำงานที่เป็นไปได้ เพื่อช่วยลดความเสี่ยงที่เกิดจากการใช้งานใบมีดที่เสื่อมสภาพ และเพิ่มประสิทธิภาพในกระบวนการผลิตวัตถุดิบสำหรับผลิตอาหารสัตว์ โดยแยกวัตถุประสงค์ดังนี้

1. เพื่อแก้ไขปัญหาการใช้พลังงานไฟฟ้าที่มากเกินไป

2. เพื่อให้ได้ Output ผลผลิตที่สม่ำเสมอและได้คุณภาพ

3. เพื่อลดต้นทุนการเปลี่ยนปริมาณใบมีดที่ไม่จำเป็น

4. เพื่อสร้างโมเดลการเรียนรู้ของ Machine Learning ในการคำนวณระยะเวลาที่ควรเปลี่ยนใบมีดของเครื่องบดอาหารเมื่อถึงรอบที่ควรเปลี่ยน

5. เพื่อป้องกันการสึกหรอของ Machine และคุณภาพของ output ผลผลิต

6. เพื่อเพิ่มประสิทธิภาพการทำงาน การรักษาคุณภาพของผลิตภัณฑ์จะช่วยสร้างความเชื่อถือในตลาดและรักษาลูกค้า

7. เพื่อแก้ไขปัญหาใบมีดที่เสื่อมสภาพเป็นการลดค่าใช้จ่ายที่เกี่ยวข้องกับการผลิตและบำรุงรักษาอุปกรณ์ในอนาคต

8. เพื่อแก้ไขปัญหาใบมีดที่เสื่อมสภาพและเป็นวิธีที่มีประสิทธิภาพในการลดความสูญเสียที่เกิดจากการผลิตไม่คงที่ ซึ่งจะช่วยเพิ่มกำไรของโรงงาน

3. ขอบเขต

วัตถุประสงค์ของการวิจัย

1. ศึกษาวิธีการใช้งานใบมีดในเครื่องบดอาหารสัตว์เพื่อวิเคราะห์ประสิทธิภาพและประโยชน์ที่ได้จากการใช้งานใบมีดในกระบวนการ

2. ศึกษาคุณสมบัติและรูปแบบของใบมีดที่เหมาะสมสำหรับการใช้งานในเครื่องบดอาหารสัตว์ เช่น วัสดุ, ความคม, ความแข็งแรง

3. ประสิทธิภาพของการบดอาหารสัตว์โดยการวัดประสิทธิภาพและประสิทธิผลของการใช้งานใบมีดในกระบวนการบดอาหารสัตว์ เช่น ความสามารถในการลดขนาดของวัตถุดิบ, ความสมบูรณ์ของผลิตภัณฑ์ที่ได้, และประสิทธิภาพในการใช้พลังงาน

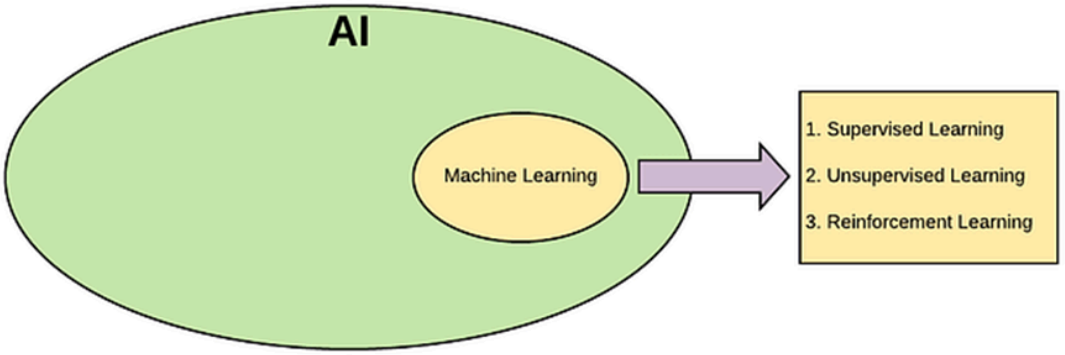
4. การค้นหาและพัฒนาเทคโนโลยีใหม่เพื่อปรับปรุงประสิทธิภาพและความปลอดภัยของการใช้งานใบมีดในเครื่องบดอาหารสัตว์

โดยจากหัวข้อวัตถุประสงค์ดังกล่าวจะนำมาเพื่อทำนายความเสื่อมสภาพของใบมีดในเครื่องบดอาหารโดยใช้เทคนิค Machine Learning โดยใช้ อัลกอริทึม Supervised Learning เช่น Support Vector Machine (SVM) หรือ Random Forest เพื่อสร้างโมเดลทำนาย โดยใช้ชุดข้อมูลที่เก็บได้จากการรวบรวมจากตัวเซนเซอร์ระบบตรวจจับในเครื่องบดอาหาร และปริมาณการใช้พลังงานในแต่ละครั้ง

และจะมีการวัดผลโดยใช้ชุดข้อมูลทดสอบที่แยกออกมาจากชุดข้อมูลเพื่อวัดประสิทธิภาพ โดยใช้เมตริกที่เหมาะสมเช่น Accuracy, Precision, Recall หรือ F1-score โดยโมเดลจะถูกสร้างขึ้นและทดสอบบนเครื่องคอมพิวเตอร์ หลังจากที่ได้ผลลัพธ์ที่มีประสิทธิภาพแล้ว โมเดลจะถูกนำไปใช้งานในสถานะการณ์จริงในโรงงานโดยการติดตั้งบนระบบคอมพิวเตอร์และอุปกรณ์ที่เกี่ยวข้องในโรงงาน

Supervised Learning

Supervised Learning คือการเรียนรู้แบบมีผู้สอน ข้อดี คือ ทำได้ง่าย ต้นทุนต่ำ เพียงใช้คอมพิวเตอร์เครื่องเดียวก็สามารถศึกษาและทำงานจนเห็นผลได้เลย



รูปที่ 1 อธิบาย Supervised Learning

จากภาพ เราจะเห็นได้ว่า Machine Learning หรือการเรียนรู้ของเครื่องจักร (ซึ่งก็คือคอมพิวเตอร์) นั้นแบ่งออกเป็นประเภทใหญ่ๆ ได้ 3 ประเภท นั่นก็คือ

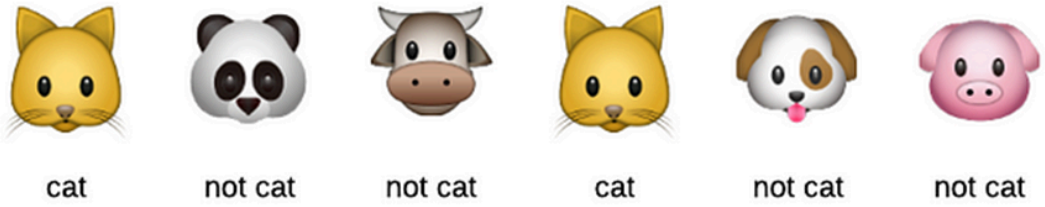
1. Supervised Learning คือ การเรียนรู้แบบมีผู้สอน
2. Unsupervised Learning คือ การเรียนรู้แบบไม่มีผู้สอน
3. Reinforcement Learning คือ การเรียนรู้ผ่านการให้รางวัล

โดยที่ วิธีการ ที่เราเลือกใช้สำหรับโปรเจคนี้ก็คือ Supervised Learning ซึ่งมีข้อดีตามที่ได้กล่าวไปคือ ทำได้ง่าย ต้นทุนต่ำ เพียงใช้คอมพิวเตอร์เครื่องเดียวก็สามารถศึกษาและทำงานจนเห็นผลได้เลย

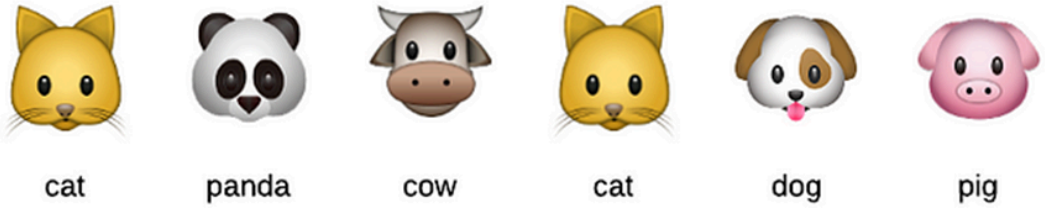
Supervised Learning หรือการเรียนรู้แบบมีผู้สอนนั้น คือการทำให้คอมพิวเตอร์สามารถหาคำตอบของปัญหาได้ด้วยตัวเอง หลังจากเรียนรู้จากชุดข้อมูลตัวอย่างไปแล้วระยะหนึ่ง

ตัวอย่างที่ 1 หากจะให้เปรียบเทียบก็เหมือนกับการสอนเด็ก ลองนึกภาพว่าผู้สอนชี้ภาพสัตว์ให้เด็กที่ไม่เคยเห็นดู แล้วบอกว่าสัตว์ตัวไหนคือแมว ตัวไหนไม่ใช่แมว ชี้ไป 2–3 วัน ให้เด็กได้เจอสัตว์หลายๆ ประเภท จนเด็กเริ่มเข้าใจ วันที่ 4–5 อาจจะลองเอาแมวตัวที่เด็กไม่เคยเห็นมาให้ดูสัก 10 ตัว รวมกับสัตว์อื่นๆ อีกจำนวนหนึ่ง โดยคราวนี้ไม่บอกว่าสัตว์ตัวไหนคือแมว ตัวไหนไม่ใช่แมว ถ้าเด็กตอบถูกก็แปลว่าการสอนมีประสิทธิภาพ

ในทำนองเดียวกัน หากสอนเด็กไปเลยว่า สัตว์ที่เด็กเห็นนั้นเป็น แมว หมา หรือหมู เด็กก็อาจจะตอบได้มากกว่าแค่ แมว หรือไม่ใช่แมว วิธีนี้อาจจะต้องใช้กระบวนการสอนที่มีความซับซ้อนมากขึ้นไปอีก เรียกวิธีการสอนเด็กทั้ง 2 แบบนี้ว่า Classification ซึ่งจะได้ผลลัพธ์ตามภาพด้านล่าง

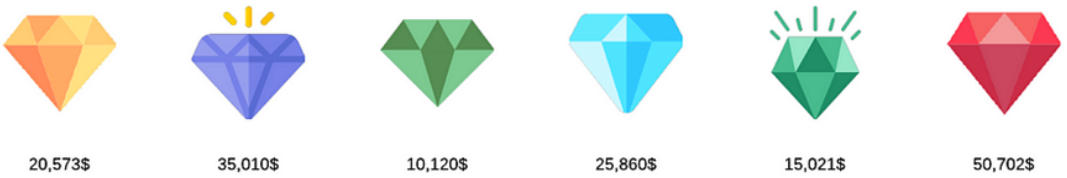


รูปที่ 2 อธิบายผลลัพธ์ที่ได้จากการสอนเด็กแบบ Classification ที่ไม่ซับซ้อน



รูปที่ 3 อธิบายผลลัพธ์ที่ได้จากการสอนเด็กแบบ Classification ที่ซับซ้อน

ตัวอย่างที่ 2 เรียกเด็กอีกคนมาสอนเรื่องราคาเพชร (diamond) ผู้สอนหยิบเพชรอันนึง ขนาด 2 กะรัต สีเหลือง ระดับความสะอาด VS2 แล้วบอกเด็กว่า อันนี้ราคา 2 ล้านบาท หยิบอีกเม็ดขนาด 3 กะรัต สีฟ้า ระดับความสะอาด VS1 แล้วบอกเด็ก 3 ล้านบาท ทำแบบนี้ไปหลายๆ เม็ดจนเด็กเกิด model หรือ logic ในการคาดเดาราคาของเพชรขึ้นในหัว จนวันนึงสุ่มหยิบเพชรเม็ดใหม่ขึ้นมา ก็อาจให้เด็กคาดเดาราคาได้เลย เราเรียกกระบวนการสอนเด็กแบบนี้ว่า Regression ดังภาพด้านล่าง

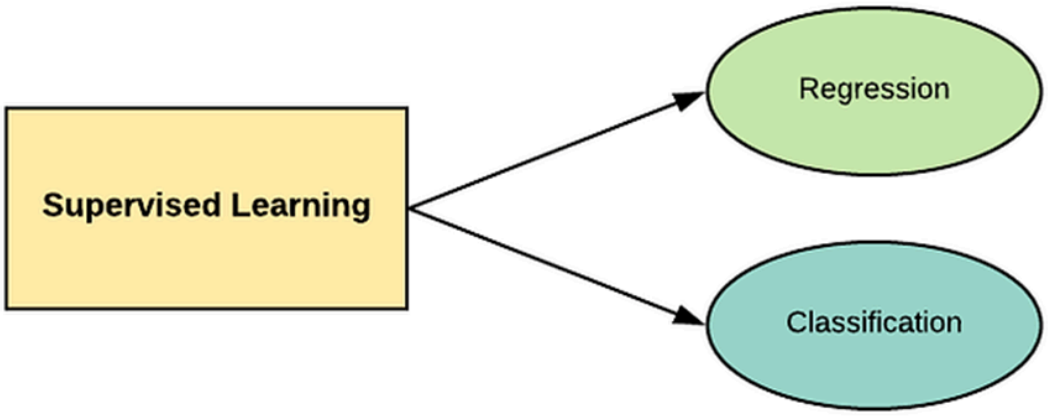


รูปที่ 4 อธิบายหลักการสอนเรื่องเพชร แบบ Regression

ซึ่งจากทั้งสองตัวอย่าง การที่จะทำให้เด็กเรียนรู้ได้ว่าอะไรคือเพชรแบบไหน หรืออะไรคือแมว ก็จำเป็นที่จะต้องมีตัวอย่างของเพชร และตัวอย่างของแมว เพื่อให้เด็กได้เรียนรู้ ซึ่งในทาง machine learning ก็เปรียบเสมือนเครื่องคอมพิวพ์เตอร์คือเด็ก ส่วนตัวอย่างก็คือการที่สร้างโมเดล ส่งเข้าไปเพื่อให้เครื่องได้เรียนรู้นั่นเอง ดังนั้น จึงสามารถกล่าวได้พอสังเขปว่า การให้เครื่องเรียนรู้แบบ Supervised Learning มีเพื่อจุดประสงค์สำหรับ

1.เพื่อการแบ่งแยกประเภท เรียกว่า Classification

2.เพื่อการคาดเดา เรียกว่า Regression



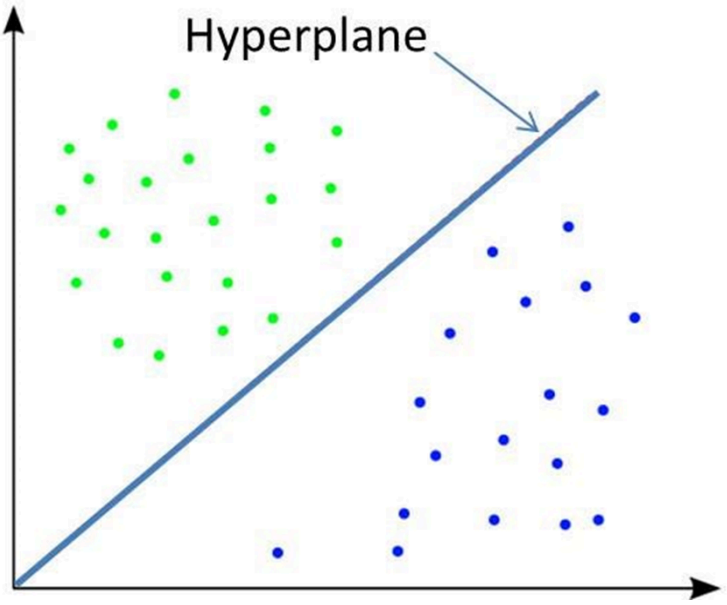
รูปที่ 5 อธิบายสรุปจุดประสงค์สำหรับ Supervised Learning

ซึ่งจุดประสงค์ของโปรเจคนี้ คือการที่ต้องการคาดเดาใบมีดของเครื่องจักร ว่ามีการเสื่อมสภาพ หรือถึงรอบระยะเวลาที่ต้องเปลี่ยนใบมีดหรือไม่ ดังนั้น จึง ให้เครื่องคอมพิวเตอร์ เรียนรู้โมเดลเพื่อการคาดเดา โดยใช้วิธีการ regression

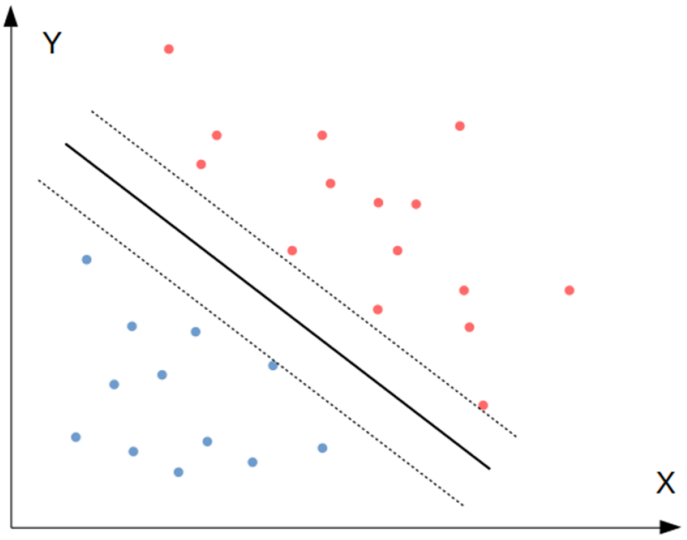
โดย algorithms สำหรับ regression ใน machine learning มีให้เลือกใช้งานในหลายรูปแบบ โดยในโปรเจคนี้จะเลือกนำเสนอ 2 ตัว ที่คิดว่าเหมาะสมสำหรับงานนั่น คือ Support Vector Machine (SVM) และ Random Forest

1.Support Vector Machines (SVM)

เป็นหนึ่งในโมเดล Machine Learning ที่ใช้ในการจำแนกข้อมูล หรือแบ่งกลุ่มข้อมูลโดยจะสร้างเส้นตรงที่ใช้แบ่งกลุ่มข้อมูล (Hyperplane) และหาเส้นที่ดีที่สุด ดังภาพ



รูปที่ 6 อธิบายตัวอย่าง SVM kindsonthegenius.com



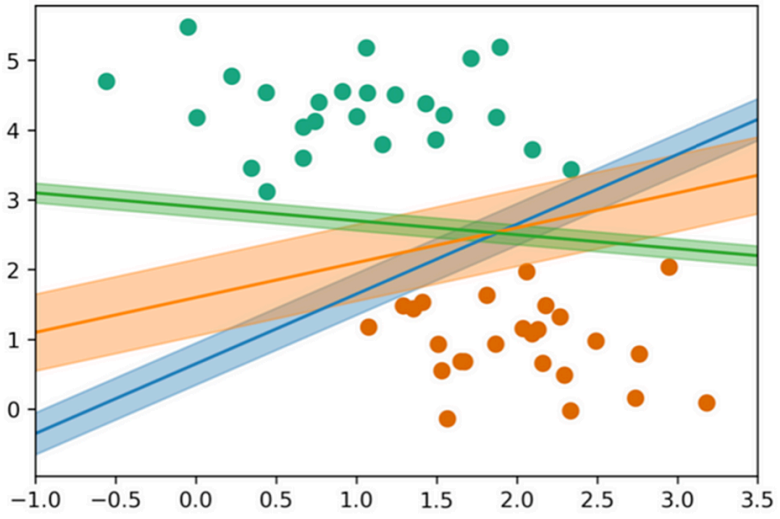
รูปที่ 7 อธิบายตัวอย่าง SVM kindsonthegenius.com

จากภาพเป็นปัญหา Binary classification ต้องการจำแนกข้อมูลออกเป็นสองจำพวก คือสีน้ำเงินและสีแดง สิ่งที่ SVM ทำ คือการหาเส้นแบ่งการตัดสินใจที่เป็นเส้นทึบ ซึ่งเส้นนี้จะเกิดขึ้นระหว่างกลางของเส้นประด้านซ้ายและขวา โดยมีเงื่อนไขว่าจะต้องหาคู่ของเส้นประที่กว้างที่สุดเท่าที่จะเป็นไปได้

โดยคู่ของเส้นประที่กว้างที่สุดเท่าที่จะเป็นไปได้นี้ จะมีสองแบบ คือ

1) Hard margin classification คือคู่เส้นประที่ห้ามไม่ให้มีจุดข้อมูลอยู่ในพื้นที่ระหว่างเส้นประ

2) Soft margin classification คืออนุญาตให้มีข้อมูลอยู่ในพื้นที่ระหว่างเส้นประได้บ้าง

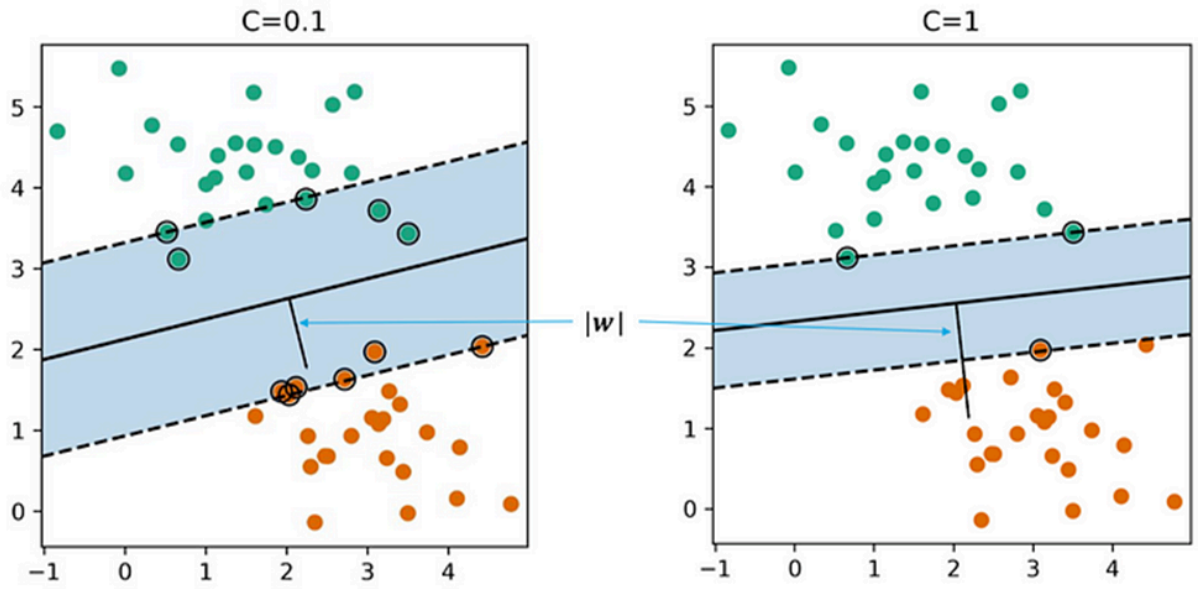


รูปที่ 8 อธิบายตัวอย่าง Max Margin and Support Vectors

การแบ่งข้อมูลสามารถแบ่งได้หลายเส้นแต่จะต้องเลือกเส้นที่มี Margin มากที่สุด คือ เส้นที่มีระยะแบ่งกว้างที่สุด เช่น เส้นสีส้มมีระยะมากที่สุด หาก Margin แคบไปขยับข้อมูลเดียวอาจจะทำให้ข้ามไปอีกฝั่งหนึ่งได้เลยทำให้มีโอกาส Overfit สูง ดังนั้น จึงจะเลือก Margin เยอะทำให้ Overfit น้อย หรือเรียกกว่า Soft Margin ดังภาพด้านบน

<u>Parameter C</u>

คือการเลือกระดับของการอนุญาตให้มีการละเมิดขอบเขตเส้นประ โดยถ้า C มาค่าน้อย หมายความว่ายอมให้มีขอบเขตที่กว้างขึ้น นั่นแปลว่ามี Regularization(การทำให้เป็นมาตรฐาน) มากขึ้น ดังภาพด้านล่าง

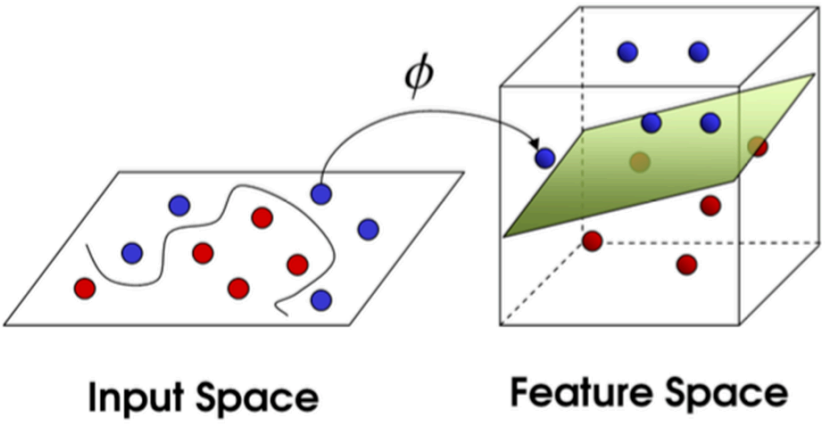


รูปที่ 9 แสดง Regularization

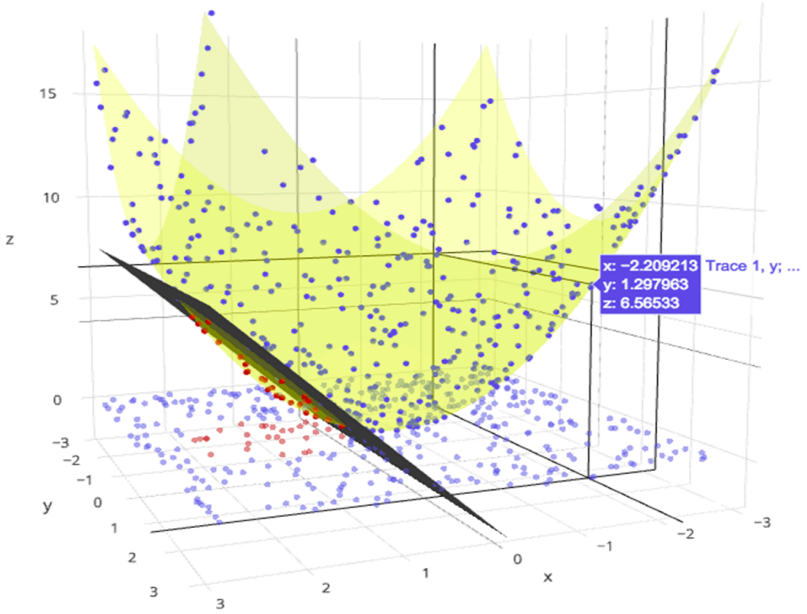
Parameter C การปรับ parameter C จะทำให้ขนาดของเส้นแบ่งเปลี่ยนแปลงได้ โดยที่ C มากจะทำให้พื้นที่แคบลง C น้อยจะทำให้พื้นที่กว้างขึ้น

Kernels

หากข้อมูลไม่สามารถแบ่งกลุ่มได้ด้วยเส้นตรง (linear) จึงได้มีวิธีการ Kernels ที่เป็น non-linear เข้ามาแก้ไขปัญหาที่เกิดขึ้น โดยวิธีการคือ สร้างมิติขึ้นมาจากเดิม 2D เป็น 3D แล้วลากเส้นตัดผ่านตรงกลางจะทำให้สามารถแบ่งข้อมูลออกไปกลุ่มได้ ดังภาพตัวอย่างด้านล่าง



Example Kernels :



รูปที่ 10 แสดงการสร้างรูปแบบจาก 2D เป็น 3D เพื่อทำการแบ่งกลุ่มข้อมูล

Support Vector Machines แม้จะถูกออกแบบมาสำหรับ Binary classification แต่สามารถนำไปประยุกต์ใช้กับ Multiclass classification และ Linear regression ได้โดยง่าย โดยใช้หลักการเดิมแต่เปลี่ยนรายละเอียดเล็กน้อย ซึ่งจะไม่กล่าวถึงในที่นี้ ขอให้รู้ว่าในระดับการใช้งาน

ข้อดีของ Support Vector Machines

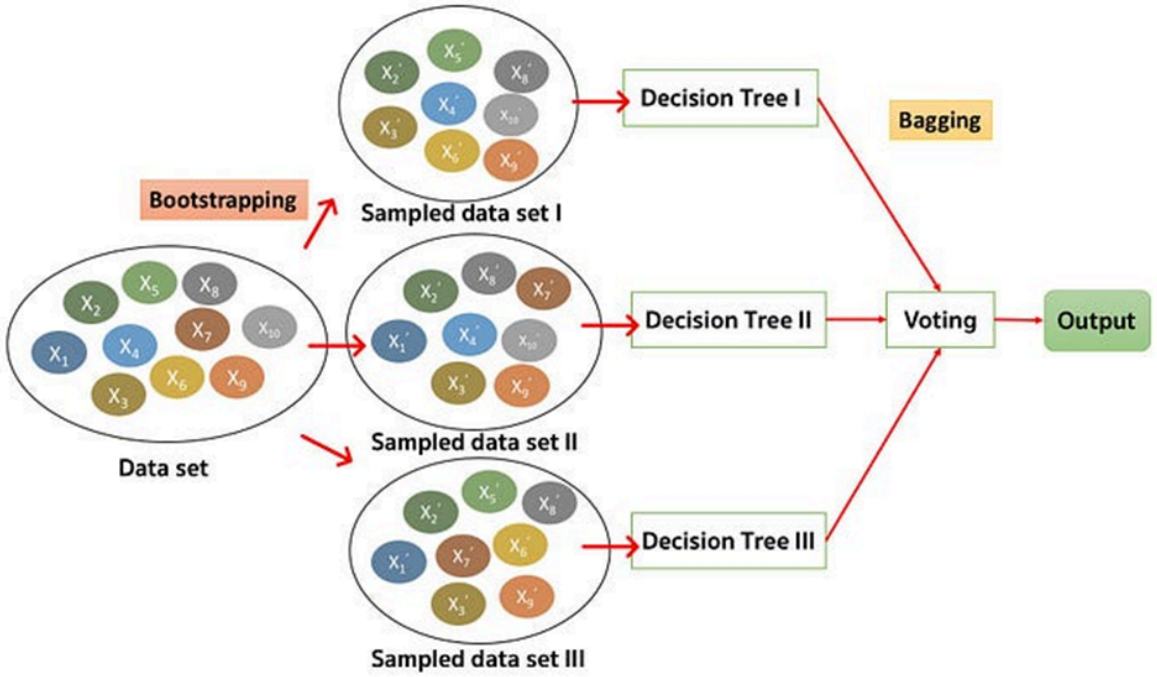
มีความยืดหยุ่นและทำงานได้ดี โดยเฉพาะอย่างยิ่งเมื่อข้อมูลมีความซับซ้อน (หลาย Feature) แต่จำนวนตัวอย่างไม่มาก (ต่ำกว่าแสนรายการ)

2. Random Forest

หลักการของ Random Forest คือ สร้าง model จาก Decision Tree หลายๆ model ย่อยๆ (ตั้งแต่ 10 model ถึง มากกว่า 1000 model) โดยแต่ละ model จะได้รับ data set ไม่เหมือนกัน ซึ่งเป็น subset ของ data set ทั้งหมด ตอนทำ prediction ก็ให้แต่ละ Decision Tree ทำ prediction ของใครของมัน และคำนวณผล prediction ด้วยการ vote output ที่ ถูกเลือกโดย Decision Tree มากที่สุด (กรณี classification) หรือ หาค่า mean จาก output ของแต่ละ Decision Tree (กรณี regression)

Decision Tree แต่ละ model ใน Random Forest ถือว่าเป็น weak learner — ประมาณว่าเป็น model ที่ไม่เก่งเท่าไหร่ แต่พอนำเอาแต่ละ Decision Tree มาทำ prediction ร่วมกัน ก็จะได้ model รวมที่มีความเก่ง และแม่นยำมากกว่า Decision Tree ที่ทำ prediction แบบเดี่ยวๆ

ภาพหลักการทำ Random Forest



รูปที่ 11 แสดงตัวอย่างหลักการทำ Random Forest

จากภาพ หลักการทำ Random Forest คือ

1. sample ข้อมูล (bootstrapping) จาก data set ทั้งหมด ให้ได้ข้อมูลออกมา n ชุด ที่ไม่เหมือนกัน ตามจำนวน Decision Tree ใน Random Forest เช่น data set ตั้งต้นมีอยู่ 10 feature (X1,X2,...,X10) แต่ละ Decision Tree จะได้ feature ไปไม่เหมือนกัน และจะได้ข้อมูลไม่ครบทุก row ด้วย จาก data set ทั้งหมดด้วย (X1 -> X1',X2->X2',...)
2. สร้าง model Decision Tree สำหรับแต่ละชุดข้อมูล
3. ทำ aggregation(รวบรวม) ผลลัพธ์ จากแต่ละ model (bagging) เช่น voting ในกรณี classification หรือ หาค่า mean ในกรณี regression

ข้อดีของRandom Forest

อ้างอิงมาจาก course Machine Learning ของ Fastai โดย Jeremy Howard

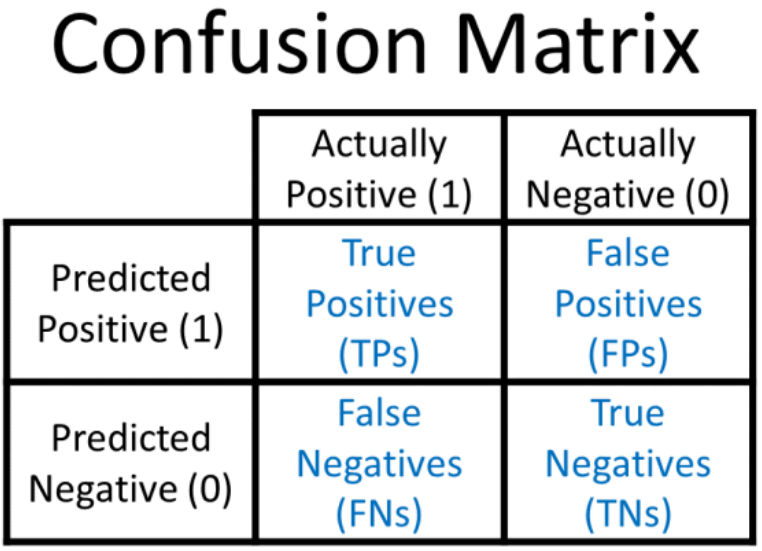
1. Random Forest ใช้ได้ทั้งกับปัญหา classification และ regression
2. Random Forest ใช้ได้ทั้งกับข้อมูล structured (ข้อมูลลักษณะเป็น column/ table) และ unstructured (เช่น รูปภาพ, text)
3. ทำ hyper-parameter tuning ให้ Random Forest ไม่ overfit ไม่ยาก

4. Random Forest ไม่ตั้ง assumption กับ feature ว่าจะต้องกระจายข้อมูลแบบ normal distribution, หรือสัมพันธ์กับ target แบบ linear, และไม่ต้องสร้างความสัมพันธ์ระหว่าง feature เพิ่มเติม (เรียกว่า interaction — เช่น สร้าง feature X_1*X_2 จาก X_1 และ X_2)

5. จากข้อ 4 ประหยัดแรงทำ Feature engineering เช่น ไม่จำเป็นต้องทำ log transform, หรือสร้าง interaction จาก feature

การวัดผล Accuracy, Precision, Recall หรือ F1-score

Confusion Matrix ถือเป็นเครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย หรือ Prediction ที่ทำนายจาก Model ที่สร้างขึ้นใน Machine learning โดยมีไอเดียจากการวัดว่า สิ่งที่เราคิด (Model ทำนาย) กับ สิ่งที่เกิดขึ้นจริง มีสัดส่วนเป็นอย่างไร



รูปที่ 12 ตาราง Confusion Matrix

- True Positive (TP)= สิ่งที่ทำนาย ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณี ทำนายว่าจริง และสิ่งที่เกิดขึ้น ก็คือ จริง
- True Negative (TN)= สิ่งที่ทำนายตรงกับสิ่งที่เกิดขึ้น ในกรณี ทำนายว่า ไม่จริง และสิ่งที่เกิดขึ้น ก็คือ ไม่จริง
- False Positive (FP)= สิ่งที่ทำนายไม่ตรงกับสิ่งที่เกิดขึ้น คือทำนายว่า จริง แต่สิ่งที่เกิดขึ้น คือ ไม่จริง
- False Negative (FN)= สิ่งที่ทำนายไม่ตรงกับที่เกิดขึ้นจริง คือทำนายว่าไม่จริง แต่สิ่งที่เกิดขึ้น คือ จริง

โดย TP,TN,FP,FN ในตารางจะแทนด้วยค่าความถี่

สามารถใช้ Confusion Matrix มาคำนวณ การประเมินประสิทธิภาพของการทำนายด้วย Model ของในรูปแบบค่าต่างๆได้หลายค่า ได้แก่

- Accuracy (ความถูกต้องที่ทายได้ตรงกับสิ่งที่เกิดขึ้นจริง)

Accuracy (ความถูกต้อง) = (TPs + TNs) / (TPs+TNs+FPs + FNs)

หรือกล่าวได้ว่า Accuracy = ผลรวมของตัวเลขบนเส้นทแยงมุมในตาราง Confusion Matrix / จำนวน observations ทั้งหมด

โดย ความเป็นจริงแล้ว Confusion matrix ไม่จำเป็นต้องเป็นแบบ 2x2 หรือมีผลลัพธ์แค่ 2 แบบ เสมอไป โดยอาจเป็น 3x3,4x4, nxn ก็ได้ โดยวิธีการหา Accuracy ก็ใช้แบบเดิม คือ ผลรวมของตัวเลขบนเส้นทแยงมุมในตารางConfusion Matrix / จำนวน observations ทั้งหมด

- Precision (ค่าความแม่นยำ) เป็นการเปรียบเทียบ การทำนายที่ถูกต้องว่า จริง และก็เกิดขึ้นจริง (TP) กับ การทำนายว่า จริง แต่สิ่งที่เกิดขึ้น คือ ไม่จริง (FP)

Precision = TPs / (TPs + FPs)

- Recall (ความถูกต้องของการทำนายว่าจะเป็น “จริง” เทียบกับ จำนวนครั้งของเหตุการณ์ทั้งทำนาย และ เกิดขึ้น ว่า “เป็นจริง”)

Recall = TPs/(TPs+FNs)

- F1-Score เป็นค่าเฉลี่ยแบบ harmonic mean ระหว่าง precision และ recall จุดประสงค์ของการสร้าง F1 ขึ้นมา คือ เพื่อเป็น single metric ที่วัดความสามารถของโมเดล

F1 = 2 x (Precision x Recall)/(Precision + Recall)

ตัวอย่างเสริมการอธิบาย เพื่อให้เข้าใจมากขึ้น

		Actual	
		Spam	Ham
Prediction	Spam	20	12
	Ham	18	50

อ้างอิงภาพจาก https://datarockie.com/2019/03/30/top-ten-machine-learning-metrics/

ในกรณีที่เราอยากทราบว่า โมเดลของเราทายแม่นขนาดไหน คือ ทายถูกต้องว่าเป็น Spam จากการพยายามทำนายทั้งหมด เราต้องใช้ Precision ก็คือ

Precision ของทำนายSpam = 20 / 32 = 0.625

แต่ถ้าเราต้องการทราบว่า โมเดลที่เราสร้างขึ้น สามารถตรวจจับ Spamได้ถูกต้องขนาดไหน จากSpam Email ทั้งหมด เราต้องใช้ Recall

Recall ของการตรวจจับSpam = 20 /38 = 0.526

แต่ในกรณีที่เราต้องการหาประสิทธิภาพของโมเดลการทำนายนี้ ที่ต้องมีทั้ง การทายถูกต้องว่าสิ่งที่เจออันนั้นคือ Spamจริงๆ และในขณะเดียวกันก็ต้องตรวจจับ Spamได้ด้วย เราก็ต้องเลือกใช้ F1 score ก็คือ เป็นค่าเฉลี่ยของทั้ง Precision และ Recall

F1ของModelนี้ = 2 * ((0.625 * 0.526) / (0.625 + 0.526)) = 0.571

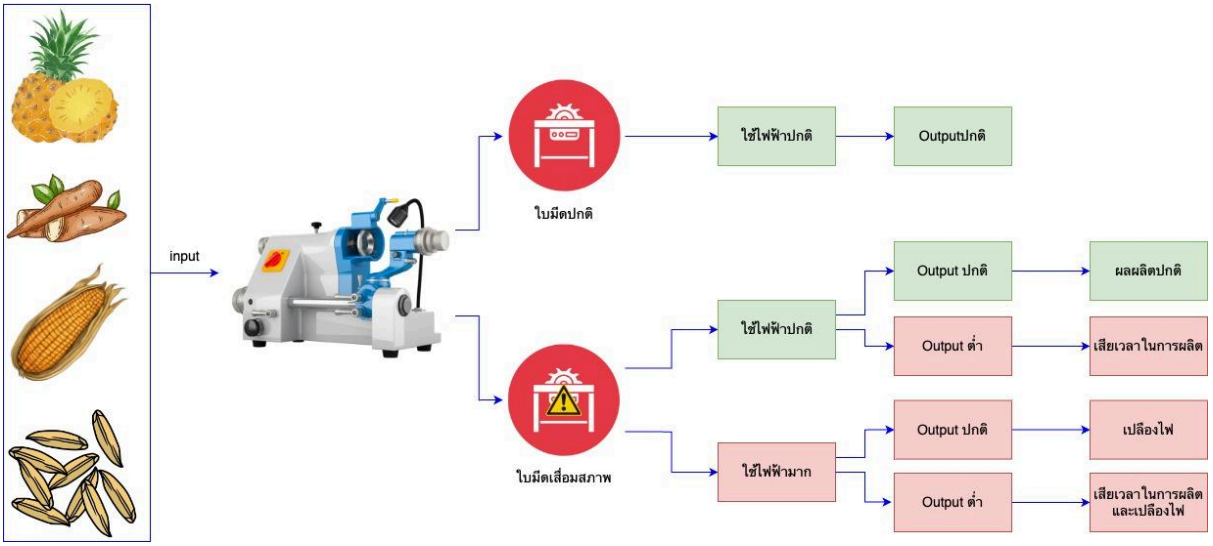
4. กระบวนการสร้างโมเดล

4.1 Business Understanding

การผลิตอาหารสัตว์เป็นกิจกรรมที่มีความสำคัญในการพัฒนาอุตสาหกรรมเกษตรและอุตสาหกรรมอาหารการใช้งานเครื่องบดอาหารสัตว์เป็นหนึ่งในขั้นตอนสำคัญในกระบวนการผลิต เนื่องจากช่วยในการบดวัตถุดิบอาหารให้เป็นขนาดที่เหมาะสมสำหรับการบริโภคของสัตว์ เพื่อให้ได้ผลิตภัณฑ์ที่มีคุณภาพและเสถียรภาพ

โดยในกระบวนการใช้งานเครื่องบดอาหารสัตว์ เราพบว่าใบมีดมักมีปัญหาเกี่ยวกับความเสื่อมสภาพ ซึ่งอาจส่งผลให้ผลิตภัณฑ์ที่ได้มีคุณภาพลดลง นอกจากนี้ การเสื่อมสภาพของใบมีดยังอาจเป็นสาเหตุให้เกิดอุบัติเหตุหรือบาดเจ็บขณะใช้งาน เช่น การหักหรือการขาดแคลนในกระบวนการการทำงาน

วัตถุประสงค์ของการวิเคราะห์นี้คือการพัฒนาโมเดลทำนายความเสื่อมสภาพของใบมีดในเครื่องบดอาหารสัตว์ โดยใช้ข้อมูลประวัติของการใช้งานและเงื่อนไขสภาพแวดล้อม เพื่อช่วยลดความเสี่ยงที่เกิดจากการใช้งานใบมีดที่เสื่อมสภาพ และเพิ่มประสิทธิภาพในกระบวนการผลิต ซึ่งจะมีกระบวนการทำงานในส่วนของเครื่องบดอาหารสัตว์หากมีปัญหา สามารถอธิบายได้ตามแผนภาพดังรูป



รูปที่ 15 Business Understanding การตรวจสอบการทำงานของใบมีดเครื่องบดข้าวโพดผลิตอาหารสัตว์

เตรียมข้อมูล: รวบรวมข้อมูลที่เกี่ยวข้องกับการใช้งานใบมีดและความเสื่อมสภาพที่เกิดขึ้น ข้อมูลมาจากตัวเซนเซอร์ระบบตรวจจับในเครื่องบดอาหารจากนั้นจะทำการตรวจสอบและทำความสะอาดข้อมูลเพื่อให้พร้อมสำหรับการประมวลผล

แบ่งชุดข้อมูล: แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ เพื่อใช้ฝึกโมเดลและทดสอบประสิทธิภาพของโมเดล

สร้างโมเดล: เลือกอัลกอริทึมและโมเดลที่เหมาะสมกับปัญหา สร้างโมเดลทำนาย Support Vector Machine (SVM), Random Forest

ฝึกโมเดล: ใช้ชุดข้อมูลการฝึกเพื่อฝึกโมเดลให้เรียนรู้ความสัมพันธ์ระหว่างข้อมูลเข้าและผลลัพธ์ที่ต้องการทำนาย

ประเมินโมเดล: ใช้ชุดข้อมูลทดสอบเพื่อประเมินประสิทธิภาพของโมเดลที่สร้างขึ้น โดยวัดค่าเมตริกที่เหมาะสม Accuracy, Precision, Recall, F1-score

ปรับปรุงและประสิทธิภาพ: ผลการประเมินโมเดล มีการปรับปรุงโมเดลและการทำซ้ำขั้นตอนการฝึกและประเมินเพื่อปรับปรุงประสิทธิภาพของโมเดล

การใช้งาน: เมื่อโมเดลมีประสิทธิภาพและเสถียรพร้อมใช้งาน จะนำโมเดลไปใช้งานในสถานะการณ์จริง โดยติดตั้งใช้งานให้กับโรงงาน

การดูแลและบำรุงรักษา: ตั้งระบบการดูแลและบำรุงรักษาโมเดลเพื่อให้มีประสิทธิภาพและคงทนในการใช้งานในระยะยาว

ประโยชน์ที่คาดหวัง

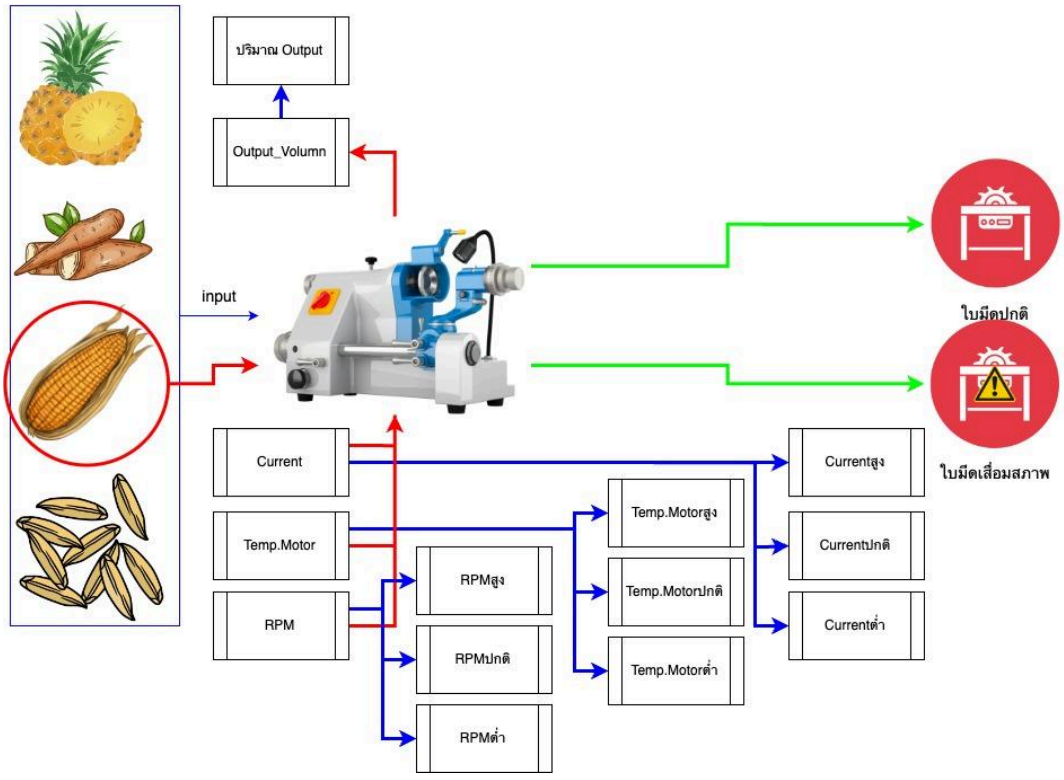
1. ลดความเสี่ยงที่เกิดจากการใช้งานใบมีดที่เสื่อมสภาพ และลดความเสี่ยงของอุบัติเหตุหรือการบาดเจ็บ
2. เพิ่มประสิทธิภาพในกระบวนการผลิตอาหารสัตว์ โดยลดการสูญเสียที่เกิดจากผลิตภัณฑ์ที่มีคุณภาพต่ำ
3. ลดค่าใช้จ่ายในการบำรุงรักษาและซ่อมแซมใบมีด
4. ลดค่าใช้จ่ายในเรื่องของค่าไฟจากกระแสไฟที่สูงผิดปกติ

4.2 Data Understanding

ชื่อชุดข้อมูล : Volume and Current Data for Grinding Machine

คำอธิบายชุดข้อมูล :

ข้อมูลที่ได้มาเป็นข้อมูล Output Volume การผลิตในแต่ละวัน ทำการเปรียบเทียบกับค่าพลังงานกระแสไฟ (Current) ที่ใช้ และมีการนำค่า Temperature Motor มาใช้เพื่อ monitor อุณหภูมิของ motor อีกด้วย กล่าวคือ หากอุณหภูมิของ motor สูงขึ้นกว่าปกติ อาจเป็นผลให้มีการใช้ค่ากระแสไฟที่มากขึ้นตามด้วยซึ่งกรณีนี้ อาจมีสาเหตุจาก root cause อื่นๆ ที่ไม่ได้เกี่ยวกับใบมีดของเครื่องจักรเสื่อมสภาพแต่อย่างใด รวมถึงค่า RPM รอบการหมุนของมอเตอร์ที่อาจส่งผลต่อค่ากระแสไฟ (Current) แต่ในบางกรณีเราอาจพบว่า Temperature Motor มีค่าปกติ แต่มีการใช้กระแสไฟ (Current) ที่มากขึ้น รวมถึงค่า Output Volumn ที่ได้มีค่าน้อยลงไม่ได้ตาม Target กรณีนี้ก็อาจจะเป็นผลของการที่ใบมีดเสื่อมสภาพลง และประวัติการเปลี่ยนใบมีดก็ถูกนำมาใช้เพื่อทำการวิเคราะห์เปรียบเทียบกับ Output Volume การผลิตในแต่ละวัน ซึ่งข้อมูลทั้งหมดนี้จะถูกเก็บอยู่ใน Database ของ Machine แต่ละเครื่อง และเพื่อความแม่นยำของข้อมูลในการสร้าง model จึงมีการใช้ข้อมูลจาก Machine ทั้งหมด 2 เครื่อง เพื่อทำการเปรียบเทียบ result ที่ได้



รูปที่ 16 Data Understanding การตรวจสอบการทำงานของ

ใบมีดเครื่องบดข้าวโพดผลิตอาหารสัตว์

จากการวิเคราะห์ข้อมูลข้างต้นทำให้สามารถสรุปตัวแปรออกมาได้ 7 ตัวแปร และเพิ่มตัวแปรที่ผู้ศึกษาสนใจ โดยมีตัวแปรดังนี้

1. Date ข้อมูลการทำงานในแต่ละวัน โดยจะเป็นข้อมูลการทำงานของเครื่องบดอาหารสัตว์ในแต่ละวัน
2. Time ข้อมูลช่วงเวลาของการทำงานในแต่ละวัน โดยในชุดข้อมูลนี้จะดึง Report ออกมาทุกๆ 2 ชั่วโมง

3. Output_Volumn ผลผลิตที่ได้ในแต่ละช่วงเวลาโดยผลผลิตที่แสดงในข้อมูลจะมีการรวมผลผลิตทุกๆ ก่อนเที่ยงคืนของทุกวัน ซึ่งเป็นปัจจัยสำคัญต่อการคาดเดาความเสื่อมสภาพของใบมีดได้ โดยคำนวณจากปริมาณหรือคุณภาพของผลผลิตที่ได้
4. Current ค่ากระแสไฟที่ใช้ โดยค่ากระแสไฟสูงเกินไปหรือต่ำเกินไป ส่งผลต่อผลผลิตและการทำงานของใบมีด
5. Temp.Motor ค่าอุณหูมิของ Motor เป็นส่วนที่ส่งผลต่อการทำงานของใบมีดและกระแสไฟ
6. RPM จำนวนรอบการหมุนของมอเตอร์ ส่งผลต่อการทำงานของ ค่าอุณหูมิของ Motor, ค่ากระแสไฟ และใบมีด เช่น หากรอบในการหมุนสูงส่งผลให้ใช้กระแสไฟมากขึ้นและการทำงานของใบมีดเพิ่มขึ้น ทำให้เกิดการเสื่อมสภาพเร็วกว่าปกติ
7. Date_Change_Grinding วันที่มีการเปลี่ยนใบมีดในแต่ละครั้ง ซึ่งมีผลต่อการตัดสินใจเปลี่ยนใบมีดในรอบถัดไปได้ เช่น หากวันที่ 10-Feb-67 01:46:04 มีการเปลี่ยนใบมีดไปแล้ว และภายในไม่ถึงเดือนจะทำการเปลี่ยนใบมีดอีกได้หรือไม่

จำนวนข้อมูล : **2 ชุดข้อมูล**

- ข้อมูลเครื่องบดอาหาร A : 4801 แถว / 7 คอลัมน์
- ข้อมูลเครื่องบดอาหาร B : 4801 แถว / 7 คอลัมน์

ตารางรายละเอียดข้อมูล :

ลำดับ	ชื่อตัวแปร	คำอธิบาย	ประเภทข้อมูล	ตัวอย่างข้อมูล
1	Date	Date Stamp	Ordinal	18-02-2566
2	Time	Time Stamp	Ordinal	9:20:12 PM
3	Output_Volumn	ผลผลิตที่ได้ในแต่ละช่วงเวลา	Num	19.138
4	Current	ค่ากระแสไฟที่ใช้	Num	457
5	Temp.Motor	ค่าอุณหูมิของ Motor	Num	63
6	RPM	จำนวนรอบการหมุนของมอเตอร์	Num	210.0672
7	Date_Change_Grinding	ประวัติวันที่เปลี่ยนใบมีดล่าสุด	Ordinal	27-01-23

4.2 การสำรวจข้อมูล (Data Exploration)

ทำการ Exploration เพื่อดูข้อมูลเบื้องต้นทั้ง 2 ชุดข้อมูล โดยทำการหา Shape data, missing values, Null & Nan values, Duplicates data, Describing Data (min,max, mean, sd, count, percentiles (25%,50%,75%,90%,95% and 99%))

- **Data Exploration ข้อมูลเครื่องบดอาหาร A : 4801 แถว / 7 คอลัมน์**
  - หา Shape data

```
import pandas as pd


df = pd.read_excel('SampleData-GD1-1.xlsx')
shape = df.shape


print("Shape Data:", shape)
```

Result :

Shape Data: (4801, 7)

ข้อมูลมี 4801 แถว 7 คอลัมน์

  - หา missing values

```
missing_values = df.isna()
print("จำนวน missing values ในแต่ละคอลัมน์:")


print(missing_values)
```

Result :

จำนวน missing values ในแต่ละคอลัมน์:

	Date	Time	Output_Volumn	RPM	Current	Temp.Motor	date_change_grinding
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
4796	False	False	False	False	False	False	False
4797	False	False	False	False	False	False	False
4798	False	False	False	False	False	False	False
4799	False	False	False	False	False	False	False

4800	False	False	False	False	False	False	False

4801 rows × 7 columns

ผลลัพธ์ที่แสดง ค่า False ทั้งหมด คือในข้อมูลทั้งหมดที่นำมาใช้ไม่มีข้อมูลหายในทุกๆคอลัมน์

- หา Null & Nan values

```
import numpy as np

null_nan_values = df.isnull().sum()

print("จำนวนของ null values และ NaN values ในแต่ละคอลัมน์:")


print(null_nan_values)
```

Result :

จำนวนของ null values และ NaN values ในแต่ละคอลัมน์:

Date	0
Time	0
Output_Volumn	0
RPM	0
Current	0
Temp.Motor	0
Date_Change_Grinding	0

dtype: int64

ผลลัพธ์ที่แสดง คือ ตัวแปรหรือข้อมูลในคอลัมน์ทั้งหมดไม่มีค่า Null หรือค่า NaN

- หา Duplicates data

```
columns = df.columns.tolist()

duplicates_count = {}

for col in columns:
    values = df[col].tolist()
    unique_values = set(values)
    count = 0
    for val in unique_values:
        if values.count(val) > 1:
            count += 1
    duplicates_count[col] = count

print("จำนวนค่าที่ซ้ำกัน:")
```

```
for column, count in duplicates_count.items():
    print(f"{column}: {count}")
```

Result :

จำนวนค่าที่ซ้ำกัน :

```
Date: 401
Time: 12
Output_Volumn: 36
RPM: 59
Current: 15
Temp.Motor: 40
Date_Change_Grinding: 13
```

ผลลัพธ์ที่แสดง คือ ทุกตัวแปรมีค่าที่ซ้ำกันอยู่ดังผลลัพธ์ที่แสดง

- หา Describing Data (min,max, mean, sd, count, percentiles)

```
description = df.describe(percentiles=[.25, .5, .75, .90, .95, .99])


print("ข้อมูลสถิติพื้นฐานของ DataFrame ของ ตัวอย่างชุดข้อมูลที่ 1 'SampleData-GD1-1' : ")
print(description)
```

Result :

ข้อมูลสถิติพื้นฐานของ DataFrame ของ ตัวอย่างชุดข้อมูลที่ 1 'SampleData-GD1-1' :

	Output_Volumn	RPM	Current	Temp.Motor	
count	4801.000000	4801.000000	4801.000000	4801.000000	
mean	201.469203	207.645701	286.115192	75.407623	
sd	122.593647	14.017385	157.471520	9.003486	
min	0.904936	0.637418	-10.600000	53.000000	
25%	101.113045	206.862091	188.801605	68.000000	
50%	184.578934	209.451324	294.461334	76.000000	
75%	295.148499	211.448212	427.502869	82.000000	
90%	379.334930	212.420303	465.779633	87.000000	
95%	417.593597	213.261368	478.847076	91.000000	
99%	477.348114	215.669495	508.696930	95.000000	
max	581.840393	219.191406	894.630737	97.000000	

ผลลัพธ์ที่แสดง คือ ค่า min,max, mean, sd, count, percentiles ของข้อมูลสามารถคำนวณค่าทางสถิติได้ เช่น ผลผลิตที่ได้ในแต่ละช่วงเวลา (Output_Volumn), จำนวนรอบการหมุนของมอเตอร์ (RPM), ค่ากระแสไฟที่ใช้ (Current), และค่าอุณหูมิของ Motor (Temp.Motor)

- หา Correlation of variables

```
correlation = df.corr()

print("ค่าสหสัมพันธ์ของตัวแปร:")

print(correlation)
```

Result :

ค่าสหสัมพันธ์ของตัวแปร:

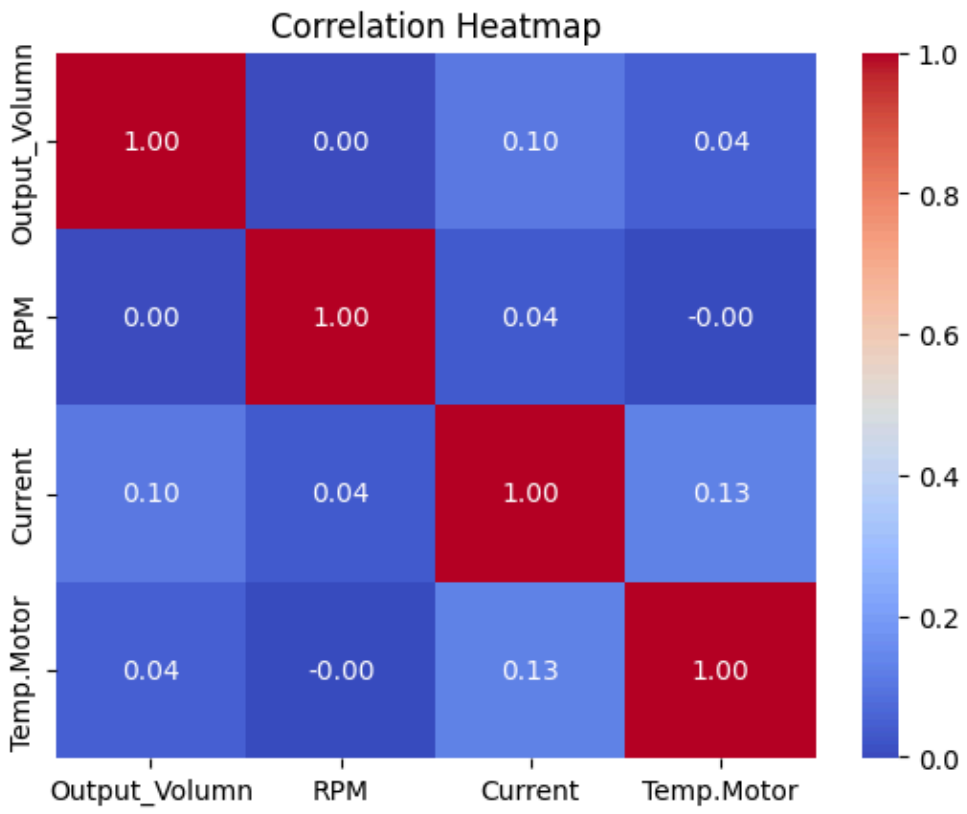
	Output_Volumn		RPM		Current	Temp.Motor
Output_Volumn	1.000000	0.003321	0.100971	0.044810		
RPM		0.003321	1.000000	0.036502	-0.002412	
Current		0.100971	0.036502	1.000000	0.125469	
Temp.Motor	0.044810	-0.002412	0.125469	1.000000		

```
correlation = df.corr()


sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap')

plt.show()
```



ผลลัพธ์ที่แสดง คือ

- Output_Volumn: มีความสัมพันธ์เชิงบวกกับ Current และ Temp.Motor
- RPM: มีความสัมพันธ์เชิงบวกกับ Current

- Current: มีความสัมพันธ์เชิงบวกกับ Temp.Motor

- ตัวแปรที่มีความสัมพันธ์ กับ Output_Volumn มากที่สุด คือ Current และ Temp.Motor

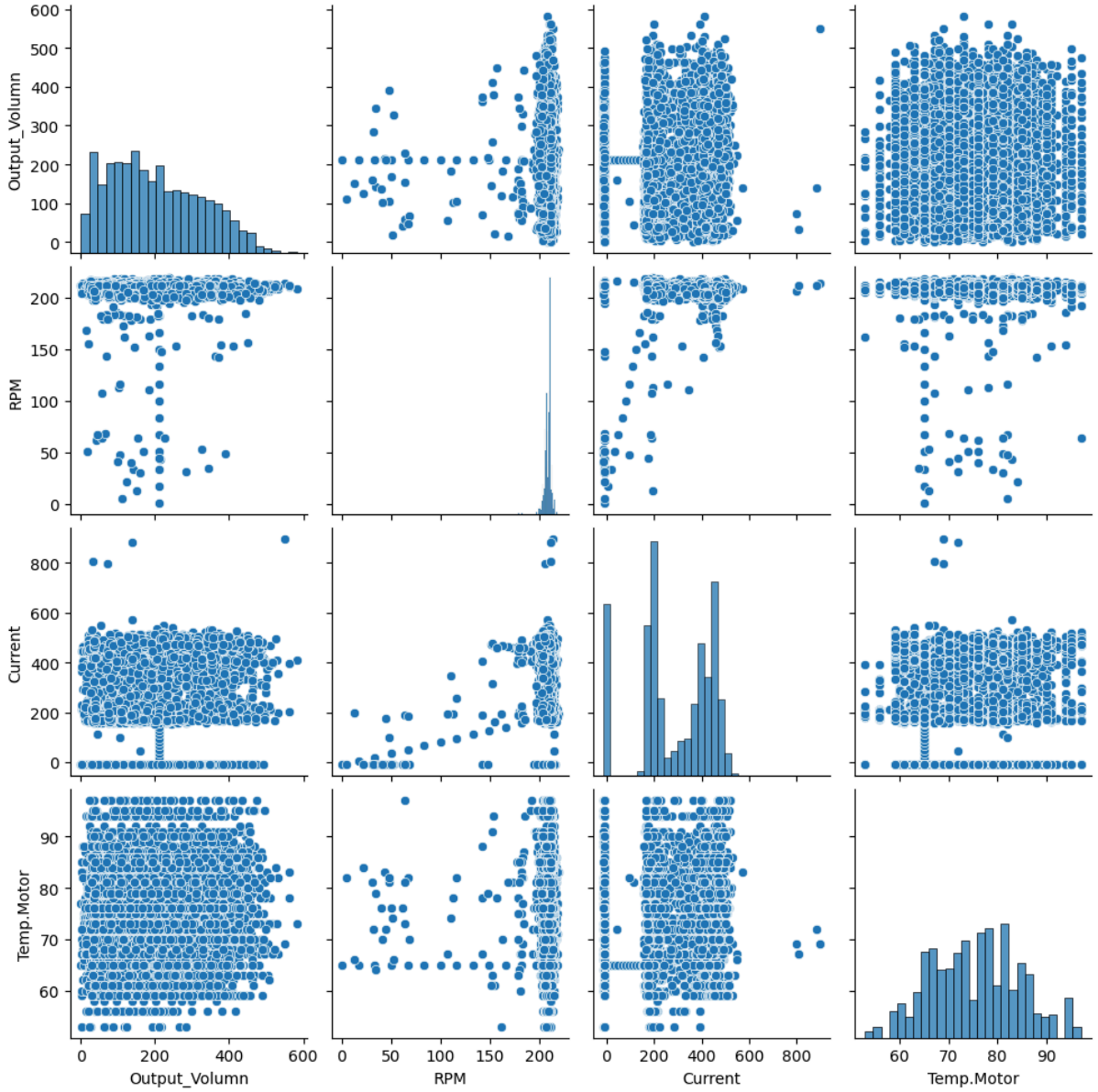
    - Bivariate Analysis

```
import seaborn as sns
import matplotlib.pyplot as plt

correlation = df.corr()

plt.figure(figsize=(10, 6))
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()

sns.pairplot(df)
plt.show()
```



ผลลัพธ์ที่แสดง คือ

กราฟแบบกระจาย (Scatter plot) แสดงความสัมพันธ์ระหว่างตัวแปร Output_Volumn (ผลผลิตที่ได้ในแต่ละช่วงเวลา) กับตัวแปรอื่นๆ 3 ตัวแปร ได้แก่

- RPM: รอบต่อนาที
- Current: กระแสไฟฟ้า
- Temp.Motor: อุณหภูมิมอเตอร์

การวิเคราะห์ความสัมพันธ์:

- Output_Volumn กับ RPM:
    - มีความสัมพันธ์เชิงบวก อ่อนแอ หมายความว่า RPM ที่สูงขึ้น มีแนวโน้มส่งผลต่อ Output_Volumn ที่สูงขึ้น เล็กน้อย

    - กราฟแสดงจุดกระจายอยู่ เหนือเส้นแนวโน้ม หมายความว่า Output_Volumn จริง มักจะ มากกว่า Output_Volumn ที่คาดการณ์ จาก RPM
- Output_Volumn กับ Current:
    - มีความสัมพันธ์เชิงบวก ปานกลาง หมายความว่า Current ที่สูงขึ้น มีแนวโน้มส่งผลต่อ Output_Volumn ที่สูงขึ้น
    - กราฟแสดงจุดกระจายอยู่ ใกล้เคียงกับเส้นแนวโน้ม หมายความว่า Output_Volumn จริง มักจะ ใกล้เคียงกับ Output_Volumn ที่คาดการณ์ จาก Current
- Output_Volumn กับ Temp.Motor:
    - มีความสัมพันธ์เชิงบวก อ่อนแอ หมายความว่า Temp.Motor ที่สูงขึ้น มีแนวโน้มส่งผลต่อ Output_Volumn ที่สูงขึ้น เล็กน้อย
    - กราฟแสดงจุดกระจายอยู่ เหนือเส้นแนวโน้ม หมายความว่า Output_Volumn จริง มักจะ มากกว่า Output_Volumn ที่คาดการณ์ จาก Temp.Motor

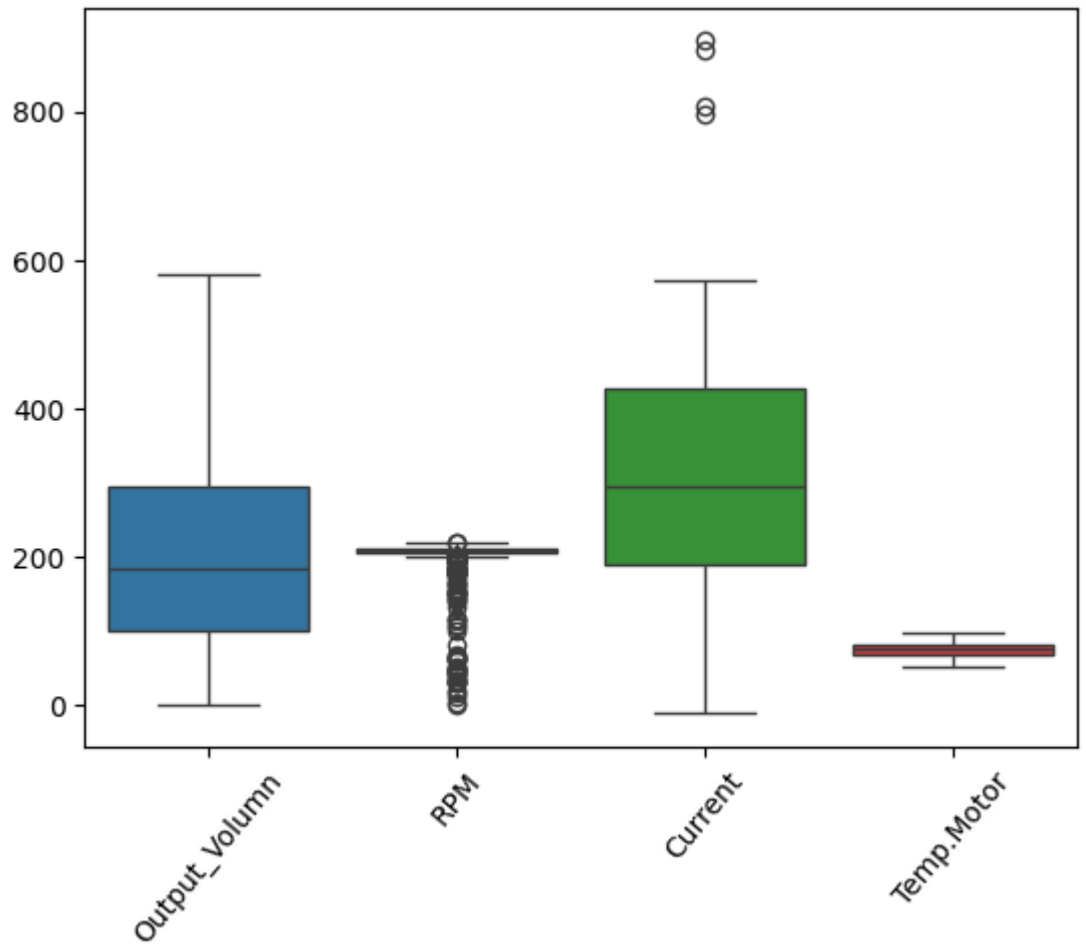
จะเห็นได้ว่า:

- ตัวแปรที่มีความสัมพันธ์ กับ Output_Volumn มากที่สุด คือ Current
- ตัวแปรที่มีความสัมพันธ์ กับ Output_Volumn อ่อนแอ คือ RPM และ Temp.Motor

- หา Outlier

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.boxplot(data=df)
plt.xticks(rotation=50)
plt.show()
```



จากภาพ Heatmap ของ df ทั้งหมดสำหรับการทำนายการเสื่อมสภาพของใบมีดเครื่องบดอาหารสัตว์ มี outlier อยู่ 2 จุด ดังนี้

1. Outlier ที่ค่า RPM

ควรตัด Outlier ออก เพราะมีจำนวนมากเกินไป อาจจะเป็นสาเหตุที่ส่งผลต่อ โมเดล ที่ใช้ทำนายการเสื่อมสภาพใบมีด อาจจะทำให้ โมเดล คาดการณ์ผิดพลาด

2. Outlier ที่ค่า Current

ควรเก็บไว้ เพราะเป็น Outlier ที่ไม่ชัดเจน อาจจะไม่ส่งผลต่อ โมเดล ที่ใช้ทำนายการเสื่อมสภาพ

- **Data Exploration ข้อมูลเครื่องบดอาหาร B : 4801 แถว / 7 คอลัมน์**

    - หา Shape data ของ df2

```
import pandas as pd


df2 = pd.read_excel('SampleData-GD2-2.xlsx')
shape = df2.shape


print("Shape Data:", shape)
```

Result :

Shape Data: (4801, 7)

ข้อมูลมี 4801 แถว 7 คอลัมน์

- หา missing values ของ df2

```
missing_values = df2.isna()
print("จำนวน missing values ในแต่ละคอลัมน์:")


print(missing_values)
```

Result :

จำนวน `missing values` ในแต่ละคอลัมน์ :

	Date	Time	Current	Temp.Motor	Output_Volumn	RPM	date_change_grinding
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
4796	False	False	False	False	False	False	False
4797	False	False	False	False	False	False	False
4798	False	False	False	False	False	False	False
4799	False	False	False	False	False	False	False
4800	False	False	False	False	False	False	False

4801 rows × 7 columns

ผลลัพธ์ที่แสดง ค่า False ทั้งหมด คือในข้อมูลทั้งหมดที่นำมาใช้ไม่มีข้อมูลหายในทุกๆคอลัมน์

- หา Null & Nan values ของ df2

```
import numpy as np

null_nan_values = df2.isnull().sum()

print("จำนวนของ null values และ NaN values ในแต่ละคอลัมน์:")


print(null_nan_values)
```

Result :

จำนวนของ null values และ NaN values ในแต่ละคอลัมน์:

Date	0
Time	0
Current	0
Temp.Motor	0
Output_Volumn	0
RPM	0
Date_Change_Grinding	0

dtype: int64

ผลลัพธ์ที่แสดง คือ ตัวแปรหรือข้อมูลในคอลัมน์ทั้งหมดไม่มีค่า Null หรือค่า NaN

- หา Duplicates data

```
columns = df.columns.tolist()

duplicates_count = {}

for col in columns:
    values = df[col].tolist()
    unique_values = set(values)
    count = 0
    for val in unique_values:
        if values.count(val) > 1:
            count += 1
    duplicates_count[col] = count

print("จำนวนค่าที่ซ้ำกัน:")
for column, count in duplicates_count.items():
    print(f"{column}: {count}")
```

Result :

จำนวนค่าที่ซ้ำกัน :

```
Date: 401
Time: 12
Current: 15
Temp.Motor: 40
Output_Volumn: 20
RPM: 60
Date_Change_Grinding: 13
```

ผลลัพธ์ที่แสดง คือ ทุกตัวแปรมีค่าที่ซ้ำกันอยู่ดังผลลัพธ์ที่แสดง

- หา Describing Data (min,max, mean, sd, count, percentiles)

```
description = df2.describe(percentiles=[.25, .5, .75, .90, .95, .99])


print("ข้อมูลสถิติพื้นฐานของ DataFrame ของ ตัวอย่างชุดข้อมูลที่ 2 'SampleData-GD2-2' : ")
print(description)
```

Result :

ข้อมูลสถิติพื้นฐานของ DataFrame ของ ตัวอย่างชุดข้อมูลที่ 1 'SampleData-GD1-2' :

	Current	Temp.Motor	Output_Volumn	RPM
count	4801.000000	4801.000000	4801.000000	4801.000000
mean	285.401525	75.401166	178.169365	209.190221
sd	157.546941	9.002562	110.895758	15.147666
min	-10.367414	53.000000	0.019208	0.637418
25%	189.842499	68.000000	87.164856	206.862091
50%	302.100006	76.000000	165.488358	208.889496
75%	426.130615	82.000000	257.749359	211.419418
90%	465.850342	87.000000	337.924011	225.622574
95%	479.024811	91.000000	375.243347	225.622574
99%	506.205688	95.000000	435.988281	225.622574
max	585.277405	97.000000	575.294739	225.622574

ผลลัพธ์ที่แสดง คือ ค่า min,max, mean, sd, count, percentiles ของข้อมูลสามารถคำนวณค่าทางสถิติได้ เช่น ผลผลิตที่ได้ในแต่ละช่วงเวลา (Output_Volumn), จำนวนรอบการหมุนของมอเตอร์ (RPM), ค่ากระแสไฟที่ใช้ (Current), และค่าอุณหูมิของ Motor (Temp.Motor)

- หา Correlation of variables

```
correlation = df2.corr()


print("ค่าสหสัมพันธ์ของตัวแปร:")
print(correlation)
```

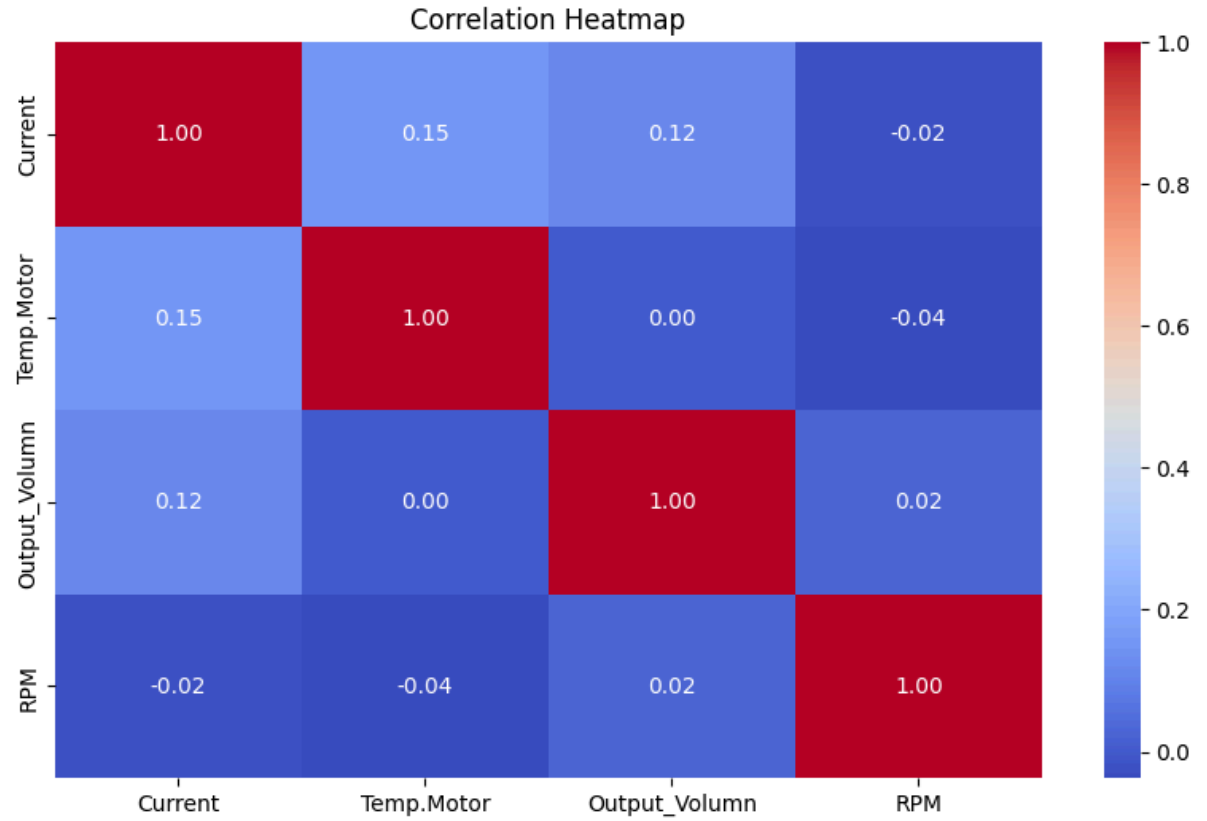
Result :

ค่าสหสัมพันธ์ของตัวแปร:

	Current	Temp.Motor	Output_Volumn	RPM
Current	1.000000	0.152039	0.116762	-0.022497
Temp.Motor	0.152039	1.000000	0.002744	-0.037297
Output_Volumn	0.116762	0.002744	1.000000	0.024776
RPM	-0.022497	-0.037297	0.024776	1.000000

```
correlation = df.corr()


sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



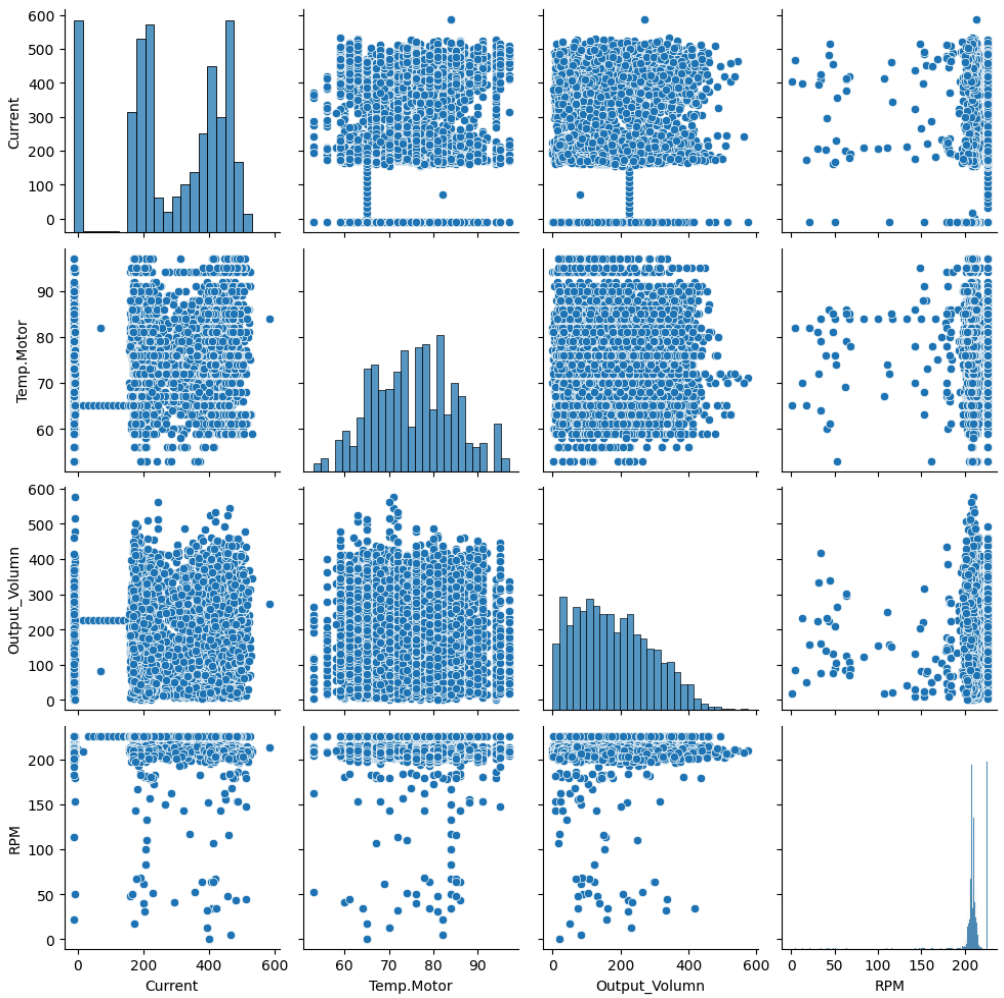
ผลลัพธ์ที่แสดง

ความสัมพันธ์:

- Output_Volumn: มีความสัมพันธ์เชิงบวกกับ Current และTemp.Motor
- RPM: มีความสัมพันธ์เชิงบวกกับ Current
- Current: มีความสัมพันธ์เชิงบวกกับ Temp.Motor
- ตัวแปรที่มีความสัมพันธ์กับ Output_Volumn มากที่สุด: Current และ Temp.Motor

- Bivariate Analysis

```
import seaborn as sns
import matplotlib.pyplot as plt
correlation = df.corr()
plt.figure(figsize=(10, 6))
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
sns.pairplot(df)
plt.show()
```



กราฟแสดงความสัมพันธ์ระหว่าง Output_Volumn (ผลผลิตที่ได้ในแต่ละช่วงเวลา) กับตัวแปรอื่นๆ 3 ตัว ได้แก่ RPM (รอบต่อนาที) Current (กระแสไฟ) และ Temp.Motor (อุณหภูมิมอเตอร์)

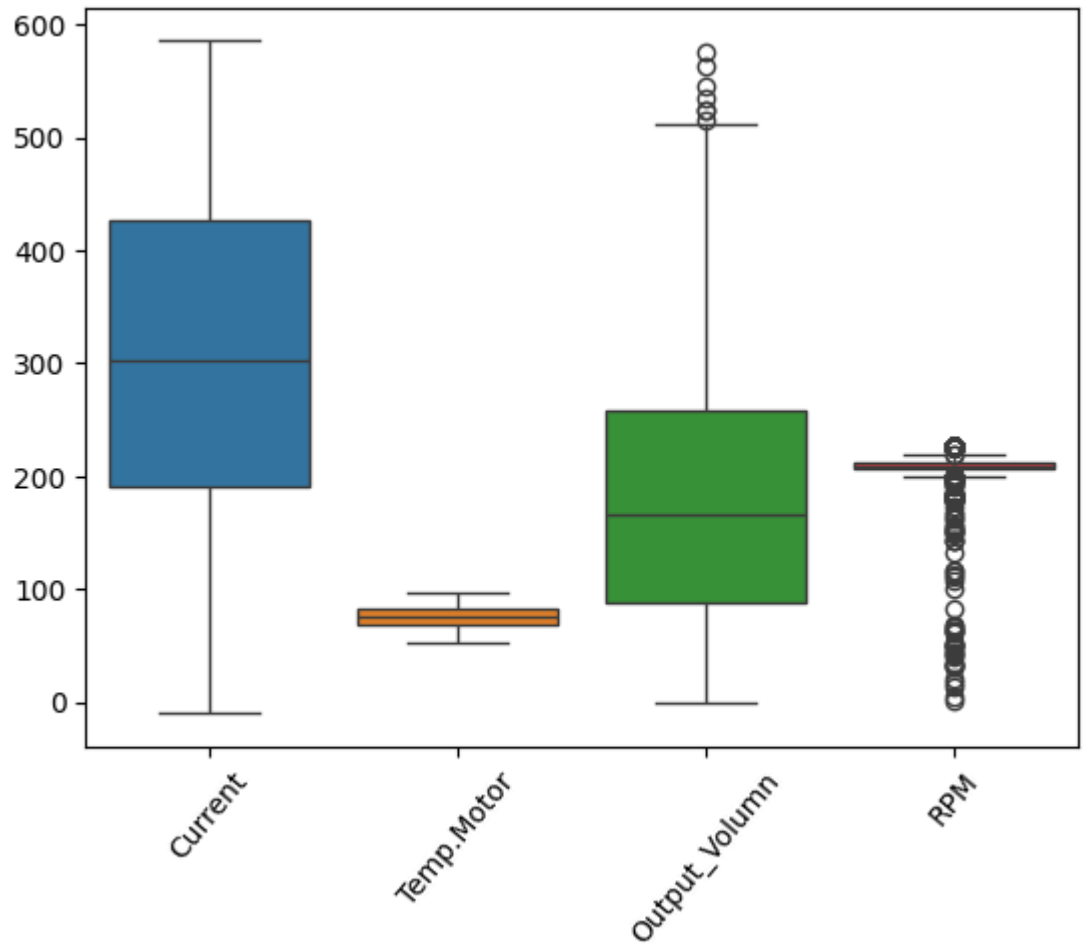
วิเคราะห์ความสัมพันธ์:

- Output_Volumn มีความสัมพันธ์เชิงบวกกับ RPM หมายความว่า เมื่อ RPM สูงขึ้น Output_Volumn ก็จะสูงขึ้นด้วย
- Output_Volumn มีความสัมพันธ์เชิงบวกกับ Current หมายความว่า เมื่อ Current สูงขึ้น Output_Volumn ก็จะสูงขึ้นด้วย
- Output_Volumn มีความสัมพันธ์เชิงบวกกับ Temp.Motor หมายความว่า เมื่อ Temp.Motor สูงขึ้น Output_Volumn ก็จะสูงขึ้นด้วย

- หา Outlier

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.boxplot(data=df)
plt.xticks(rotation=50)
plt.show()
```



จากกราฟ จะมีตัวแปรที่มี Outlier คือ Output_Volumn, RPM มีรายละเอียดดังนี้

1. Outlier ที่ค่า RPM

ควรตัด Outlier ออก เพราะมีจำนวนมากเกินไป อาจจะเป็นสาเหตุที่ส่งผลต่อ โมเดล ที่ใช้ทำนายการเสื่อมสภาพใบมีด อาจจะทำให้ โมเดล คาดการณ์ผิดพลาด

2. Outlier ที่ค่า Output_Volumn

ควรเก็บไว้ เพราะเป็น Outlier ที่ไม่ชัดเจน อาจจะไม่ส่งผลต่อ โมเดล ที่ใช้ทำนายการเสื่อมสภาพ

อ้างอิง

1. IT554 Pattern Recognition and Machine Learning, SWU
2. https://medium.com/@pradyasin/support-vector-machines-svm-943f9a732a69
3. Pagon Gatchalee.(2019).Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ในMachine learning.สืบค้นเมื่อ 28/02/2024.Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ในMachine learning | by Pagon Gatchalee | Medium
4. chemistrytalk.Accuracy vs.Precision.สืบค้นเมื่อ 28/02/2024.https://chemistrytalk.org/accuracy-vs-precision/https://phuri.medium.com/supervised-learning-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%97%E0%B8%B3%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%87-1c0e411a40a2
5. ที่มา https://medium.com/@pradyasin/support-vector-machines-svm-943f9a732a69
6. ที่มา https://guopai.github.io/ml-blog08.html

นิยามคำศัพท์เฉพาะ

1. Bagging ความหมาย วิธีการที่ใช้ในการสร้างแบบจำลองทางสถิติหลายๆ ตัว โดยใช้ข้อมูลที่สุ่มแบบมีการกล่าวถึงข้อมูลแต่ละชุด เราสร้างแบบจำลองหลายๆ ตัว แล้วนำผลลัพธ์ของแต่ละตัวมาเชื่อมกันโดยวิธีการเฉพาะ เช่น การใช้วิธีหาค่าเฉลี่ยหรือหาค่าที่มีความถี่สูงที่สุด เพื่อให้ได้ผลลัพธ์ที่แม่นยำและคาดเดาได้ดีขึ้น แบบจำลองที่ใช้ใน bagging มักจะเป็นแบบไม่เชิงพารามิเตอร์เช่น Decision Trees หรือ Random Forests ซึ่งมักจะมีประสิทธิภาพดีในการทำนายและคาดเดาข้อมูลโดยทั่วไป โดยทั่วไปแล้ว Bagging มักถูกใช้ในการลดความผิดพลาดและเพิ่มความแม่นยำของแบบจำลองทางสถิติ โดยสร้างหลายๆ แบบจำลองแล้วนำผลลัพธ์มาเชื่อมกันโดยวิธีการเฉพาะ ทำให้การทำนายของแบบจำลองดีขึ้นอย่างมีนัยสำคัญ
2. Overfitting ความหมาย ปัญหาที่เกิดขึ้นเมื่อแบบจำลองสถิติหรือโมเดลทางคณิตศาสตร์มีประสิทธิภาพมากเกินไปในการเรียนรู้ข้อมูลส่วนที่มีอยู่ (training data) ซึ่งทำให้การทำนายบนข้อมูลทดสอบหรือข้อมูลที่ไม่เคยเห็นมาก่อน (test data) ไม่แม่นยำเท่าที่ควร การ overfitting มักเกิดขึ้นเมื่อโมเดลมีความซับซ้อนมากเกินไปหรือมีจำนวนพารามิเตอร์มากเกินไปต่อจำนวนข้อมูลที่ใช้ในการเรียนรู้ ทำให้โมเดลจดจำลายข้อมูลส่วนละเอียดที่ไม่จำเป็น และสร้างข้อกำหนดที่ไม่เกี่ยวข้องกับข้อมูลทั่วไป ผลลัพธ์ที่ตามมาคือการทำนายที่ไม่แม่นยำเมื่อใช้กับข้อมูลทดสอบหรือข้อมูลจริง วิธีการป้องกันการ overfitting รวมถึงการใช้ข้อมูลการทดสอบแยกออกจากข้อมูลการฝึก เพื่อทำการตรวจสอบประสิทธิภาพของโมเดล และการใช้เทคนิคต่างๆ เช่น cross-validation เพื่อประเมินประสิทธิภาพของโมเดลในขณะเดียวกันกับการป้องกัน overfitting โดยการใช้เทคนิค regularization เพื่อลดความซับซ้อนของโมเดล หรือการใช้เทคนิค dropout ในโมเดล Deep Learning เป็นต้น การตรวจสอบและป้องกัน overfitting เป็นส่วนสำคัญในการพัฒนาแบบจำลองที่มีประสิทธิภาพและมีความแม่นยำในการทำนายข้อมูลที่ไม่เคยเห็นมาก่อน