

Crop yield Prediction

การทำนายผลผลิต

สมาชิก

ภูริภัทร สุ่มสุข

พชรพล แดงมณี

กตัญญู เมืองมุงคุณ

ภูติณัฐ สระทองบุตร

โครงการนี้เป็นส่วนหนึ่งของวิชาการวิเคราะห์ข้อมูลขนาดใหญ่
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการข้อมูลประยุกต์

มีนาคม 2567

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

คำนำ

การทำนายผลผลิตของพืชสามารถทำได้โดยใช้ข้อมูลทางด้านพื้นที่เพาะปลูก สภาพอากาศ ประเภทของพืช การให้น้ำ การใช้ปุ๋ย และข้อมูลเกี่ยวกับโรคและแมลงที่อาจมีผลกระทบต่อผลผลิต ด้วยก็ได้ เพื่อให้คำแนะนำเช่นการปรับปรุงการจัดการแปลงเพาะปลูกหรือระบบการเก็บรักษาน้ำ

ภูริภัทร สุนสุข

พชรพล แดงมณี

กตัญญ เมืองมุงคุณ

ภูตินันท์ สระทองบุตร

สารบัญ

หลักการและเหตุผล

ผลผลิต คือ ผลลัพธ์ของการปลูกไม่ว่าจะเป็น ผัก ผลไม้ หรือข้าวผลผลิตจะได้มากหรือน้อยอาจมีปัจจัยจาก

ปุ๋ย : แสดงถึงปริมาณหรือความเข้มข้นของปุ๋ยที่ใช้ในบางหน่วย, อุณหภูมิ: สภาพอุณหภูมิในช่วงระยะเวลาการทำฟาร์ม, ไนโตรเจน (N): ปริมาณไนโตรเจนที่มีอยู่ในปุ๋ยหรือดิน โดยทั่วไปจะวัดเป็นปอนด์ต่อเอเคอร์, ฟอสฟอรัส (P): ปริมาณฟอสฟอรัสที่มีอยู่ในปุ๋ยหรือดิน โดยทั่วไปจะวัดเป็นปอนด์ต่อเอเคอร์, โพแทสเซียม (K): ปริมาณโพแทสเซียมที่มีอยู่ในปุ๋ยหรือดิน โดยทั่วไปจะวัดเป็นปอนด์ต่อเอเคอร์, ผลผลิต (คิว/เอเคอร์): ปริมาณพืชผลที่เก็บเกี่ยวได้ต่อเอเคอร์ โดยทั่วไปจะวัดเป็นบุชเชล ตัน หรือหน่วยอื่นๆ ที่เหมาะสม

วัตถุประสงค์

- เพื่อหาอัลกอริทึมที่เหมาะสมในการทำนายผลผลิต
- เพื่อสร้างโมเดลทำนายผลผลิต

Dataset







ข้อมูลนี้ นำมาจาก Kaggle

<https://www.kaggle.com/yaminh/crop-yield-prediction/data>

มี Rows ทั้งหมด 109

มี Columns ทั้งหมด 7

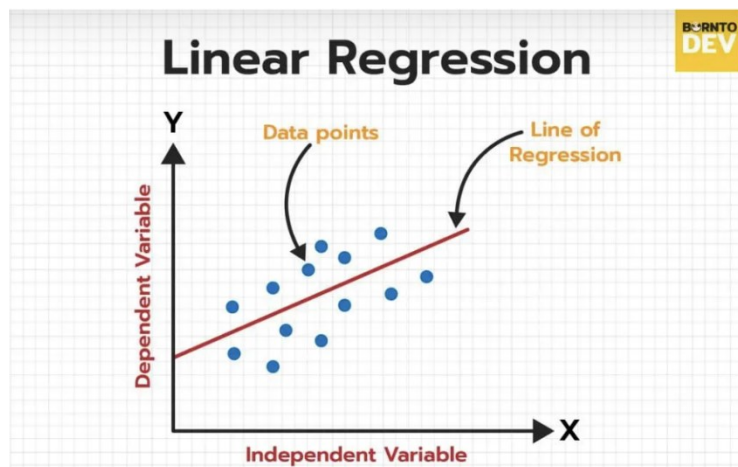
Example

# Rain Fall (mm)	# Fertilizer	▲ Temperatue	# Nitrogen (N)	# Phosphorus (P)	# Potassium (K)	# Yeild (Q/acre)
 4001.3k	 5080	28 39 Other (75) 18% 13% 69%	 5980	 1825	 1522	 5.512
1230	80	28	80	24	20	12
480	60	36	70	20	18	8
1250	75	29	78	22	19	11
450	65	35	70	19	18	9
1200	80	27	79	22	19	11
500	70	34	74	22	16	10
1275	71	28	77	21	20	11
425	65	37	67	18	15	7
1200	77	27	78	23	20	12
400	50	39	60	18	15	6
1280	80	26	80	24	20	12

อัลกอริทึมที่ใช้ในการทดลอง

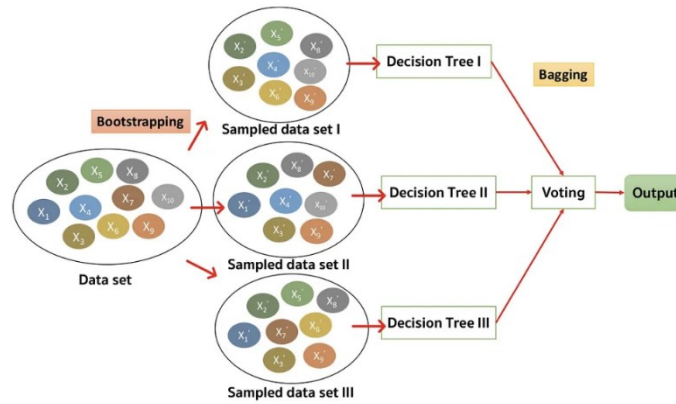
Liner Regression

“Linear Regression” หรือ “การวิเคราะห์การถดถอยเชิงเส้น” เกิดจากการรวมกันของคำว่า “Linear” ที่แปลว่าเชิงเส้น และ “Regression” ที่แปลว่าการถดถอย โดย Linear Regression จะเป็นความสัมพันธ์ของตัวแปรหรือสิ่งที่เรากำลังสนใจ ซึ่งจะถูกใช้กับการคำนวณค่าที่เป็นตัวเลข เพื่อหาความสัมพันธ์หรือทำนายข้อมูลต่าง ๆ



Random Forest Regression

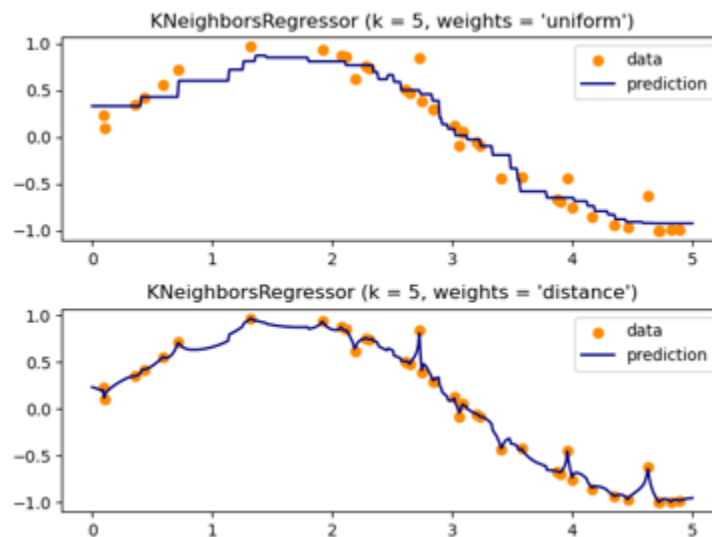
หลักการของ Random Forest คือ สร้าง model จาก Decision Tree หลายๆ model ย่อยๆ (ตั้งแต่ 10 model ถึง มากกว่า 1000 model) โดยแต่ละ model จะได้รับ data set ไม่เหมือนกัน ซึ่งเป็น subset ของ data set ทั้งหมด ตอนทำ prediction ก็ให้แต่ละ Decision Tree ทำ prediction ของใครของมัน และคำนวณผล prediction ด้วยการ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



ภาพ 1-หลักการทำ Random Forest

KNeighborsRegressor

KNeighborsRegressor เป็นอัลกอริทึมใน machine learning ที่ใช้ในงาน regression ซึ่งหน้าที่หลักของมันคือการทำนายค่าตัวแปรตามที่กำหนดโดยใช้ข้อมูลจำนวน k จำนวน (k neighbors) ที่ใกล้เคียงที่สุดในชุดข้อมูลเพื่อจำลองค่าหรือคุณสมบัติของตัวแปรตามที่ต้องการทำนาย



ขั้นตอนการเตรียมข้อมูลทำความสะอาดข้อมูล

ในการสร้างโมเดลเพื่อพยากรณ์ จะทำการรวบรวมข้อมูลและนำข้อมูลเหล่านั้นมาแบ่งออกเป็น 2 ส่วน คือ

- 1) ข้อมูลที่ใช้สำหรับการสอน (Train) หมายถึง การนำข้อมูลไปสร้างสมการเพื่ออธิบายรูปแบบข้อมูลนั้น ๆ เรียกสมการที่สร้างขึ้นว่า โมเดลสำหรับแทนข้อมูลกลุ่มนี้
- 2) ข้อมูลที่ใช้สำหรับการทดสอบ (Test) หมายถึง การนำข้อมูลเพื่อป้อนให้กับสมการหรือโมเดลทางคณิตศาสตร์ เพื่อคำนวณหาประสิทธิภาพของการพยากรณ์

โดยในที่นี้จะแบ่งเป็น Train 80% และ Test 20%

วิธีการวัดประสิทธิภาพ

การวัดประสิทธิภาพของโมเดลทำนายในที่นี้จะใช้ค่าเฉลี่ยของความคลาดเคลื่อน (error) ระหว่างค่าที่ทำนายได้กับค่าจริง Root Mean Squared Error (RMSE) และ R-squared (หรือ coefficient of determination) ซึ่งเป็นสถิติที่บ่งบอกถึงความเหมาะสมของโมเดลทำนายกับข้อมูลจริง ๆ

Root Mean Squared Error (RMSE): คือค่าเฉลี่ยของความคลาดเคลื่อนระหว่างค่าทำนายและค่าจริง ๆ โดยมี การนำมายกกำลังสองก่อนและหลังการหาร เพื่อให้ค่าคลาดเคลื่อนทั้งหมดเป็นบวก โดยค่า RMSE ที่น้อยกว่ายิ่งดี เนื่องจากหมายถึงค่าทำนายที่ใกล้เคียงกับค่าจริงมากขึ้น

R-squared (R²): เป็นค่าที่บ่งบอกถึงความสามารถในการอธิบายความแปรปรวนของข้อมูลโดยโมเดล ถ้าค่า R-squared เข้าใกล้ 1 แสดงว่าโมเดลสามารถอธิบายข้อมูลได้ดี เมื่อเทียบกับค่าเฉลี่ยของค่าตามตัวแปรตาม (dependent variable) ในชุดข้อมูล

กระบวนการใช้การวิเคราะห์

เตรียมข้อมูล:

นำเข้าข้อมูล: นำเข้าข้อมูลจาก Kaggle โดยในชุดข้อมูลจะมีหลายละเอียดเกี่ยวกับ อุณหภูมิ, ไนโตรเจน, ฟอสฟอรัส, โพแทสเซียม และ ผลผลิต

จัดรูปแบบข้อมูล: แปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมสำหรับการวิเคราะห์ ซึ่งอาจรวมถึงการเลือกคุณลักษณะ (features) ที่มีผลต่อผลผลิตและการแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมสำหรับการประมวลผล

สร้างโมเดลทำนาย:

เลือกโมเดลทำนาย: โดยในที่นี้จะใช้ LinearRegression, RandomForestRegressor และ KNeighborsRegressor

การฝึกโมเดล: ใช้ข้อมูลที่มีอยู่ในการฝึกโมเดล โดยใช้ขั้นตอนการฝึกที่เหมาะสมกับโมเดลที่เลือก เช่น การปรับพารามิเตอร์, การใช้ Cross Validation เพื่อประเมินประสิทธิภาพของโมเดล

การทดสอบ (Model Evaluation):

การแบ่งชุดข้อมูล (Data Splitting): โดยในที่นี้จะแบ่งเป็น Train 80% และ Test 20%

การประเมิน (Model Evaluation): หลังจากฝึกโมเดลเสร็จสิ้น ใช้ชุดข้อมูลทดสอบเพื่อประเมินประสิทธิภาพของโมเดล โดยใช้เมตริก เช่น Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared

ผลการเรียนรู้ :

	LinearRegression	RandomForestRegressor	KNeighborsRegressor
RMSE	0.094	0.094	-
R2	1.000	0.998	0.746
Mean Squared Error	-	-	0.842

จากผลลัพธ์ที่แสดง พารามิเตอร์ R-squared (R2) เป็นค่าที่ใช้วัดความสามารถในการอธิบายของโมเดล ซึ่งมีค่าสูงสุดที่เป็นไปได้เท่ากับ 1.0

ดังนั้น โมเดล LinearRegression ที่มีค่า R2 เท่ากับ 1.0 ถือเป็นโมเดลที่ดีที่สุดในที่นี้ เนื่องจากสามารถอธิบายข้อมูลทั้งหมดในชุดข้อมูลได้อย่างแม่นยำที่สุด หลังจากนั้นคือ RandomForestRegressor ที่มีค่า R2 เข้าใกล้ 1.0 อยู่เล็กน้อยและ KNeighborsRegressor ที่มีค่า R2 ต่ำกว่าไปนิดหน่อย ซึ่งแสดงให้เห็นว่า Linear Regression น่าจะเหมาะสมกับชุดข้อมูลนี้มากที่สุดในที่นี้

ทดสอบ Model

Linear Regression:

กำหนดโมเดล Linear Regression โดยใช้ PySpark's MLlib กำหนดตารางพารามิเตอร์สำหรับ cross-validation เพื่อปรับแต่งพารามิเตอร์การ regularize และพารามิเตอร์ของ elastic net ทำการ cross-validation โดยใช้ตารางพารามิเตอร์ที่กำหนดเพื่อหาโมเดลที่ดีที่สุดคำนวณหาค่ามิติการประเมินเช่น RMSE (Root Mean Squared Error) และ R-squared สำหรับโมเดล Linear Regression ที่ดีที่สุด

Random Forest Regression:

กำหนดโมเดล Random Forest Regression โดยใช้ PySpark's MLlib กำหนดตารางพารามิเตอร์สำหรับ cross-validation เพื่อปรับแต่งจำนวนต้นไม้และความลึกสูงสุดของต้นไม้ทำการ cross-validation โดยใช้ตารางพารามิเตอร์ที่กำหนดเพื่อหาโมเดลที่ดีที่สุดคำนวณหาค่ามิติการประเมิน (RMSE และ R-squared) สำหรับโมเดล Random Forest Regression ที่ดีที่สุด

K Nearest Neighbors Regression (โดยใช้ scikit-learn):

แปลง PySpark DataFrame เป็น Pandas DataFrame เนื่องจากโมเดล scikit-learn ทำงานกับโครงสร้างข้อมูลของ Pandas แบ่งข้อมูลเป็นชุดฝึกและทดสอบฝึกโมเดล K Nearest Neighbors Regressor ด้วยพารามิเตอร์เริ่มต้นและประเมินด้วย RMSE และ R-squared ทำการค้นหารายการพารามิเตอร์ที่ดีที่สุดสำหรับ KNN Regressor ด้วย cross-validation

พิมพ์ผลลัพธ์ของพารามิเตอร์ที่ดีที่สุดและโมเดลที่ดีที่สุด

สรุปผล

	LinearRegression	RandomForestRegressor	KNeighborsRegressor
RMSE	0.094	0.094	-
R2	1.000	0.998	0.746
Mean Squared Error	-	-	0.842

จากการสรุปผลเราสามารถเห็นได้ว่า Linear Regression และ Random Forest Regression มีประสิทธิภาพที่ดีกว่า K Nearest Neighbors Regression ในการทำนายผลผลิตข้าว โดย Linear Regression มีความแม่นยำสูงที่สุดด้วยค่า R2 เท่ากับ 1.000 และ Random Forest Regression มีค่า R2 เข้าใกล้กับ 1.000