รายงาน

การทำนายการอนุมัติสินเชื่อด้วยข้อมูลทางการเงินและประวัติการกู้ยืม

โดย

นายกษิดิ์เดช พลายเผือก 64130500004

เสนอ

ดร.นิวรรณ วัฒนกิจรุ่งโรจน์

รายงานนี้เป็นส่วนหนึ่งของวิชา INT27101 Machine Learning

คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

1. บทนำ

การอนุมัติสินเชื่อเป็นกระบวนการสำคัญในธุรกิจการเงินและสินเชื่อที่มีผลต่อกำไรและความเสี่ยงขององค์กร การใช้ข้อมูลทางการเงินและประวัติการกู้ยืมของลูกค้าในการทำนายการอนุมัติสินเชื่อเป็นเครื่องมือที่มีประสิทธิภาพในการช่วยวิเคราะห์และตัดสินใจอย่างมีเหตุผล

2. ข้อมูลที่ใช้

ข้อมูลทางการเงินและประวัติการกู้ยืมของลูกค้า จำนวน 32,586 คลาส โดยมีรายละเอียด ดังนี้

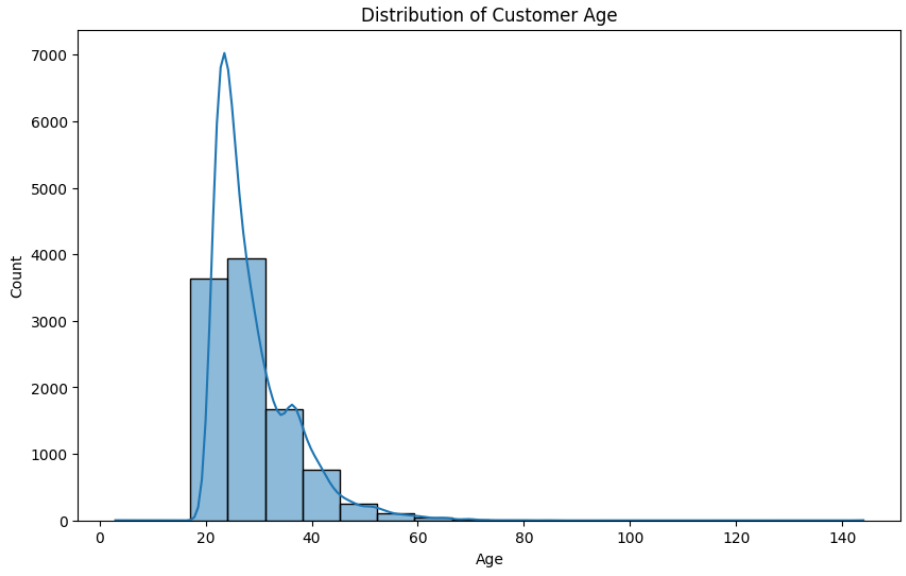
1. จำนวนข้อมูล: 32,586 ชุดข้อมูล
2. นวนตัวแปร: 13 ตัวแปร
3. ตัวแปรที่สำคัญ:
    1. customer_age: อายุของลูกค้า
    2. customer_income: รายได้ของลูกค้า
    3. home_ownership: สถานะเจ้าบ้านของลูกค้า
    4. employment_duration: ระยะเวลาการทำงานปัจจุบันของลูกค้า
    5. loan_intent: จุดประสงค์ในการกู้ยืมเงิน
    6. loan_grade: เกรดของสินเชื่อ
    7. loan_amnt: จำนวนเงินที่กู้ยืม
    8. loan_int_rate: อัตราดอกเบี้ยของสินเชื่อ
    9. term_years: ระยะเวลาการกู้ยืม
    10. historical_default: ประวัติการค่างวดเกินกำหนด
    11. cred_hist_length: ระยะเวลาประวัติเครดิตของลูกค้า
    12. Current_loan_status: สถานะปัจจุบันของสินเชื่อ

โดยจะใช้ข้อมูลข้างต้น เพื่อนำมาทำนาย loan status หรือสถานะการอนุมัติสินเชื่อในอนาคต

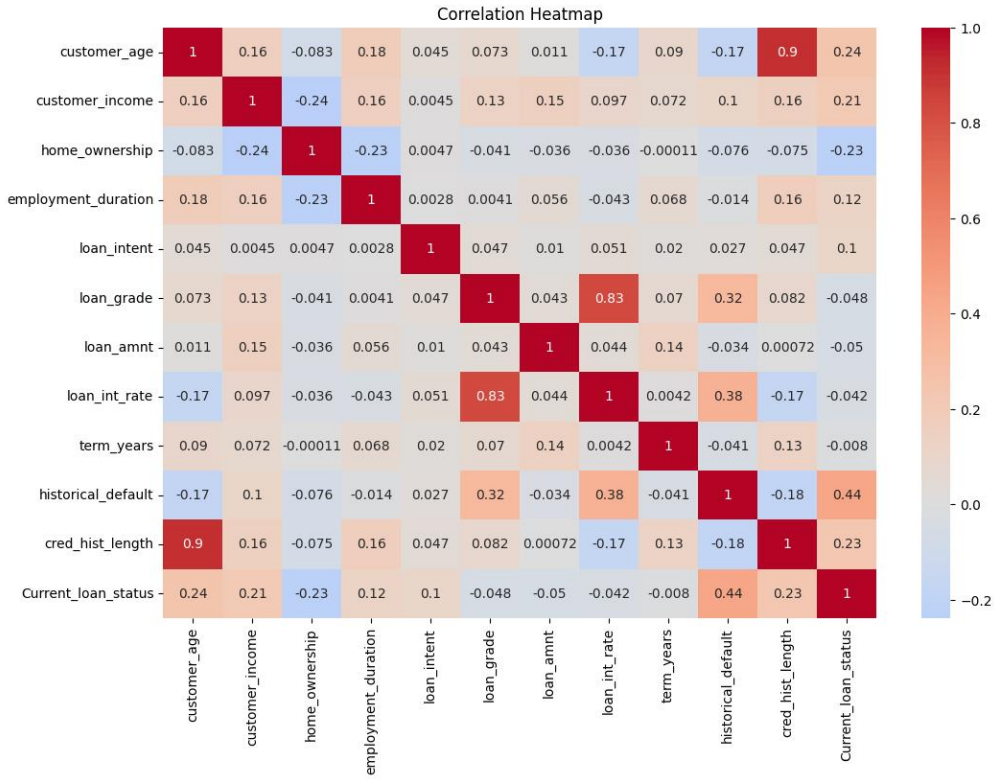
2.1 ธรรมชาติของข้อมูล

ข้อมูลที่ใช้ในการวิเคราะห์และทดลองการทำนายเป็นข้อมูลทางการเงินและประวัติการกู้ยืมของลูกค้าที่เก็บรวบรวมโดยองค์กรทางการเงิน มีลักษณะดังนี้

- **จำนวนข้อมูลและคุณภาพข้อมูล**:
    - มีข้อมูลทั้งหมด 32,586 แถว (rows)
    - ประกอบไปด้วยข้อมูลที่มีความสมบูรณ์และข้อมูลที่ขาดหายไปบ้าง
    - ข้อมูลเชิงตัวเลข (numerical data) ประกอบด้วยตัวแปรอายุของลูกค้า (customer age), ระยะเวลาการทำงาน (employment duration), อัตราดอกเบี้ยของสินเชื่อ (loan rate), ระยะเวลากู้ยืม (term years), และอื่น ๆ
    - ข้อมูลเชิงกฎ (categorical data) ประกอบด้วยสถานะการค้างชำระหนี้ (historical default), วัตถุประสงค์ของสินเชื่อ (loan intent), ระดับความเสี่ยงของสินเชื่อ (loan grade), สถานะปัจจุบันของสินเชื่อ (Current loan status), และอื่น ๆ
- **แนวโน้มและการกระจายของข้อมูล**:
    - ข้อมูลอายุของลูกค้ามีค่าเฉลี่ยอยู่ที่ 27.73 ปี โดยมีค่าสูงสุดที่ 144 ปี
    - ระยะเวลาการทำงานเฉลี่ยของลูกค้าอยู่ที่ 4.79 ปี โดยมีค่าสูงสุดที่ 123 ปี
    - อัตราดอกเบี้ยของสินเชื่อมีค่าเฉลี่ยอยู่ที่ 11.01% และมีค่าสูงสุดที่ 23.22%
    - ระยะเวลากู้ยืมเฉลี่ยอยู่ที่ 4.76 ปี โดยมีค่าสูงสุดที่ 10 ปี
    - ความยาวของประวัติเครดิตเฉลี่ยอยู่ที่ 5.80 ปี และมีค่าสูงสุดที่ 30 ปี



แนวโน้มการกระจายตัวของช่วงอายุในชุดข้อมูล



ความสัมพันธ์ของข้อมูล

จำนวนข้อมูลในแต่ละ Class

คลาส	จำนวน
customer_age	32586
customer_income	32586
home_ownership	32586
employment_duration	31691
loan_intent	32586
loan_grade	32586
loan_amnt	32585
loan_int_rate	29470
term_years	32586
historical_default	11849
cred_hist_length	32586
Current_loan_status	32582

ตัวอย่างข้อมูล

customer_id	customer_age	customer_income	home_ownership	employment_duration	loan_intent	loan_grade	loan_amnt	loan_int_rate	term_years	historical_default	cred_hist_length	Current_loan_status
1.0	22	59000	REN	123	PER	C	£35,000	16.02	10	Y	3	DEF
2.0	21	9600	OWN	5.0	EDU	A	£1,000	11.14	1	NaN	2	NOD
3.0	25	9600	MOR	1.0	MED	B	£5,500	12.87	5	N	3	DEF
4.0	23	65500	REN	4.0	MED	B	£35,000	15.23	10	N	2	DEF
5.0	24	54400	REN	8.0	MED	B	£35,000	14.27	10	Y	4	DEF
. . .												

หมายเหตุ PER = PERSONAL EDU = EDUCATION

MED = MEDICAL REN = RENT

MOR = MORTGAGE DEF = DEFAULT

NOD = NO DEFAULT

2.2 การจัดเตรียมข้อมูล

- ตรวจสอบข้อมูลที่ขาดหายไปและปรับปรุงตามความเหมาะสม เช่น ลบข้อมูลที่ขาดหาย, เติมค่าข้อมูล , หรือลบแถวหรือคอลัมน์ที่ไม่มีข้อมูล
- แปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมสำหรับการวิเคราะห์ โดยการเปลี่ยนข้อมูลให้อยู่ในรูปแบบตัวเลขเมื่อจำเป็น
- เลือกคุณลักษณะที่สำคัญและมีผลต่อการวิเคราะห์ และลบคุณลักษณะที่ไม่จำเป็น
- แบ่งข้อมูลออกเป็นชุดข้อมูลฝึกและชุดข้อมูลทดสอบ เพื่อใช้ในการวิเคราะห์และประเมินประสิทธิภาพของโมเดล
- ปรับข้อมูลให้อยู่ในช่วงที่เหมาะสมสำหรับการวิเคราะห์ และปรับมาตรฐานของข้อมูลให้เหมาะสม

หลังจากที่ได้ทำการจัดเตรียมข้อมูลแล้ว ข้อมูลที่ได้จะมีคุณภาพและเชื่อถือได้มากยิ่งขึ้น เพื่อนำไปใช้ในการวิเคราะห์และการตัดสินใจในธุรกิจอย่างมีเสถียรภาพและมั่นใจยิ่งขึ้น

```
age_threshold_min = 18

df.dropna(inplace=True)

df['loan_amnt'] = df['loan_amnt'].str.replace('£', '').str.replace(',', '').astype(float)
```

ตัวอย่างการจัดการข้อมูล

3. วิธีการสร้างโมเดล

การสร้างโมเดลในขั้นตอนนี้เริ่มต้นด้วยการเตรียมข้อมูลโดยการเข้ารหัสข้อมูลแบบหมวดหมู่ (Categorical Encoding) โดยใช้เทคนิค Label Encoding ซึ่งเป็นกระบวนการแปลงข้อมูลข้อความหรือข้อมูลแบบหมวดหมู่เป็นตัวเลข เพื่อให้โมเดลสามารถทำงานได้ ขั้นตอนการสร้างโมเดลดังกล่าวมีดังนี้:

1. การนำเข้าข้อมูล: เริ่มต้นด้วยการนำเข้าโมดูลที่จำเป็นจาก scikit-learn เพื่อใช้ในการเตรียมข้อมูล
2. การสร้าง LabelEncoder: สร้างอ็อบเจ็กต์ของ LabelEncoder สำหรับแต่ละคุณลักษณะทางสถิติเพื่อใช้ในการแปลงข้อมูลแบบหมวดหมู่ให้กลายเป็นตัวเลข
3. การแปลงข้อมูล: นำ LabelEncoder ที่สร้างขึ้นมาใช้ในการแปลงข้อมูลแบบหมวดหมู่ใน DataFrame โดยที่แต่ละคอลัมน์จะถูกแปลงเป็นตัวเลข

ขั้นตอนต่อไปคือการเตรียมข้อมูลสำหรับการสร้างโมเดล เริ่มต้นด้วยการแบ่งข้อมูลเป็นชุดฟีเจอร์ (Features) และตัวแปรเป้าหมาย (Target Variable) โดยใช้คำสั่ง train_test_split จาก scikit-learn เพื่อแบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบตามอัตราส่วนที่กำหนด

1. Import Libraries: นำเข้าโมดูลที่จำเป็นเพื่อใช้ในการสร้างโมเดล เช่น train_test_split สำหรับการแบ่งข้อมูล
2. Split Data: แบ่งข้อมูลเป็น Features (X) และ Target Variable (y) โดยใช้ drop เพื่อลบคอลัมน์ที่ไม่ได้ใช้ในการสร้างโมเดลออกจาก DataFrame
3. แบ่งข้อมูล: ใช้ train_test_split เพื่อแบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ โดยกำหนดอัตราส่วนของชุดทดสอบด้วย test_size=0.2 ซึ่งหมายถึงการแบ่งข้อมูลให้ 20% เป็นชุดทดสอบและ 80% เป็นชุดฝึก โดย random_state=42 จะทำให้การแบ่งข้อมูลมีความสุ่มที่สถานะคงที่ เพื่อให้ผลลัพธ์สามารถทำซ้ำได้แม้ว่าจะมีการรันโค้ดในครั้งต่าง ๆ ก็ตาม

จากนั้นจะทำการแบ่งข้อมูลเพื่อเตรียมสำหรับการสร้างโมเดล โดย:

1. Features (X): เป็นชุดข้อมูลที่ใช้ในการสร้างโมเดล โดยลบคอลัมน์ที่เป็นตัวแปรตาม (target variable) และคอลัมน์ที่ไม่จำเป็นออกจาก DataFrame โดยใช้ .drop(['Current_loan_status','customer_id'], axis=1) และเก็บไว้ในตัวแปร X
2. Target Variable (y): เป็นตัวแปรตามที่ต้องการทำนาย ในที่นี้คือ 'Current_loan_status' ซึ่งถูกเก็บไว้ในตัวแปร y

ดังนั้นหลังจากขั้นตอนนี้ เราจะได้ชุดข้อมูล X ที่ประกอบด้วย features ทั้งหมดที่ใช้ในการสร้างโมเดล และตัวแปร y ที่เป็นค่าตามที่เราต้องการทำนาย พร้อมสำหรับการสร้างโมเดลในขั้นตอนต่อไป

ในการสร้างโมเดล ได้กำหนดรายการของโมเดลที่ต้องการสร้างและทดสอบ ซึ่งประกอบไปด้วย Logistic Regression, Random Forest, AdaBoost, XG Boost, และ Support Vector Machine (SVM) โดยใช้คลาสที่มีให้จากไลบรารี scikit-learn สำหรับแต่ละโมเดล โดยใช้การแบ่งข้อมูลเป็นชุด train/test และวนซ้ำ 5 รอบ เพื่อให้แน่ใจว่าผลลัพธ์ไม่ได้รับผลกระทบจากการแบ่งข้อมูลแบบสุ่ม ในแต่ละรอบที่ทำการ train แต่ละโมเดลบนชุด train และทำนายบนชุด test เพื่อประเมินประสิทธิภาพของโมเดลด้วยค่าความแม่นยำ (accuracy)

โดยโมเดลที่เลือกมาข้างต้นนั้น มีรายละเอียด คุณสมบัติพื้นฐาน และกระบวนการทำงาน ดังต่อไปนี้

1. **Logistic Regression**:

    - **คุณสมบัติ**: เป็นโมเดลที่ใช้สำหรับการจำแนกประเภทข้อมูลที่มีความน่าจะเป็นอยู่ในกลุ่มที่กำหนด โดยใช้ฟังก์ชัน logistic function เพื่อคำนวณความน่าจะเป็นว่าข้อมูลจะอยู่ในกลุ่มไหน ซึ่งจะมีผลลัพธ์เป็นค่าความน่าจะเป็น (probability) ที่ข้อมูลจะอยู่ในกลุ่มที่กำหนด
    - **กระบวนการทำงาน**: Logistic Regression จะคำนวณหาเส้นแบ่งระหว่างกลุ่มข้อมูลได้ด้วยการปรับพารามิเตอร์ของสมการ logistic function ให้เหมาะสมกับข้อมูลที่มีอยู่ โดยการใช้กระบวนการ gradient descent หรือ solver ที่เหมาะสม เพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำสูงที่สุดต่อข้อมูลที่มีการแบ่งแยกเชิงเส้นหรือเป็นไปได้ในพื้นที่ของข้อมูล

2. **Random Forest**:

    - **คุณสมบัติ**: เป็นโมเดลที่ใช้เทคนิคของการสร้างต้นไม้หลายต้นแล้วรวมผลลัพธ์จากต้นไม้แต่ละต้น เพื่อลดความผิดพลาดและป้องกันการเกิด overfitting ในข้อมูลที่มีจำนวนคุณลักษณะ (features) มาก
    - **กระบวนการทำงาน**: Random Forest จะสร้างต้นไม้แต่ละต้นด้วยการสุ่มตัวอย่างข้อมูลและคุณลักษณะที่ใช้ในการสร้างต้นไม้ จากนั้นรวมผลลัพธ์ที่ได้จากต้นไม้ทั้งหมดเพื่อให้ได้คำตอบที่ถูกต้องมากยิ่งขึ้น โดยสามารถป้องกันการเกิด overfitting ได้ดี เนื่องจากการใช้หลายต้นไม้ที่สร้างขึ้นเองและมีการสุ่มข้อมูล

3. **AdaBoost**:

    - **คุณสมบัติ**: เป็นโมเดลที่ทำงานโดยการปรับปรุงความแม่นยำของการจำแนกโดยรวมผลลัพธ์จากโมเดลอื่น ๆ ที่ไม่มีความแม่นยำดีเท่าเช่นกัน เพื่อเพิ่มประสิทธิภาพในการจำแนก
    - **กระบวนการทำงาน**: AdaBoost จะเริ่มต้นด้วยการสร้างโมเดลหนึ่ง แล้วพยายามแก้ไขความผิดพลาดที่เกิดขึ้นด้วยการให้ความสำคัญมากขึ้นกับตัวอย่างที่ทำนายผิดพลาด จากนั้นจะสร้างโมเดลต่อไปที่จะแก้ไขความผิดพลาดนั้น ๆ ให้ได้ดีขึ้น โดยมีการเลือกโมเดลที่ได้ผลลัพธ์ที่ดีที่สุดจากการทดลองก่อนหน้า การทำงานของ AdaBoost จะใช้การเรียนรู้ชนิด Ensemble เพื่อเพิ่มความแม่นยำของโมเดล

4. **XGBoost**:

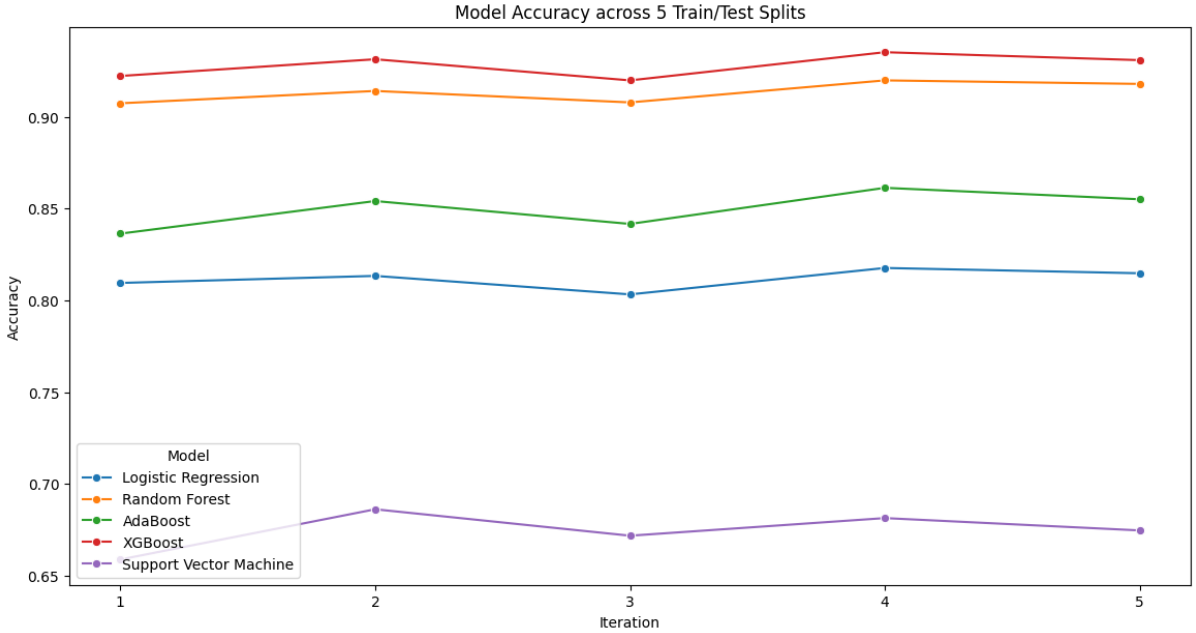
    - **คุณสมบัติ**: เป็นโมเดลที่พัฒนามาจาก Gradient Boosting Machine (GBM) โดยมีประสิทธิภาพในการจัดการกับข้อมูลที่มีจำนวนคุณลักษณะมากและความแม่นยำในการทำนายที่สูง
    - **กระบวนการทำงาน**: XGBoost จะทำงานโดยการสร้างต้นไม้การเรียนรู้ (decision tree) หลายต้นที่ถูกปรับค่าไปเรื่อย ๆ เพื่อลดความผิดพลาดในการทำนาย โดยมีการใช้ Gradient Descent Algorithm ในการปรับค่าแบบ Gradient Boosting และมีการป้องกันการเกิด overfitting ด้วย regularization

5. **Support Vector Machine (SVM)**:

    - **คุณสมบัติ**: เป็นโมเดลที่ใช้ในการจัดกลุ่มข้อมูลที่มีความซับซ้อนและแยกแยะข้อมูลได้อย่างชัดเจน โดยการใช้เส้นแบ่งที่สร้างขึ้นจาก margin ที่มีความเชื่อมั่นสูงที่สุด
    - **กระบวนการทำงาน**: SVM จะทำงานโดยการหาเส้นแบ่งหรือ margin ที่มีระยะที่สูงที่สุดระหว่างกลุ่มข้อมูล โดยการแบ่งแยกระหว่างกลุ่มข้อมูลที่ต้องการจัดแยกโดยใช้ kernel trick เพื่อแปลงข้อมูลให้อยู่ในมิติที่สูงขึ้น และจะสามารถทำนายข้อมูลที่มีการแบ่งแยกที่ไม่ชัดเจนได้อย่างมีประสิทธิภาพ

4. ผลการทดลอง

จากการทดลอง พบว่าโมเดล XGBoost ได้ให้ผลลัพธ์ที่ดีที่สุดเมื่อเทียบกับโมเดลอื่น ๆ ที่ทดสอบ โดยมีความแม่นยำเฉลี่ยประมาณ 93.52% ในการแบ่งชุดข้อมูลเป็นชุด train/test โดยสูงสุดที่การทดลองในรอบที่ 4



ผลลัพธ์จากการทดลอง

ผลลัพธ์จากการใช้ cross-validation บนโมเดลที่เราเลือก (XGBoost) แสดงให้เห็นว่าคะแนน cross-validation ในแต่ละ fold อยู่ในช่วงประมาณ [0.92441512, 0.9160168, 0.92436975, 0.92376951, 0.92617047] ซึ่งเป็นคะแนนที่สูงและมีความแปรปรวนไม่มาก ค่าเฉลี่ยของคะแนน cross-validation คือประมาณ 0.9229483275013666

หลังจากนั้น เราทดสอบโมเดลที่เลือกบนชุดข้อมูลทดสอบแยกออกมา โดยได้ผลลัพธ์ดังนี้:

ความแม่นยำบนชุดทดสอบ: 0.9217850287907869

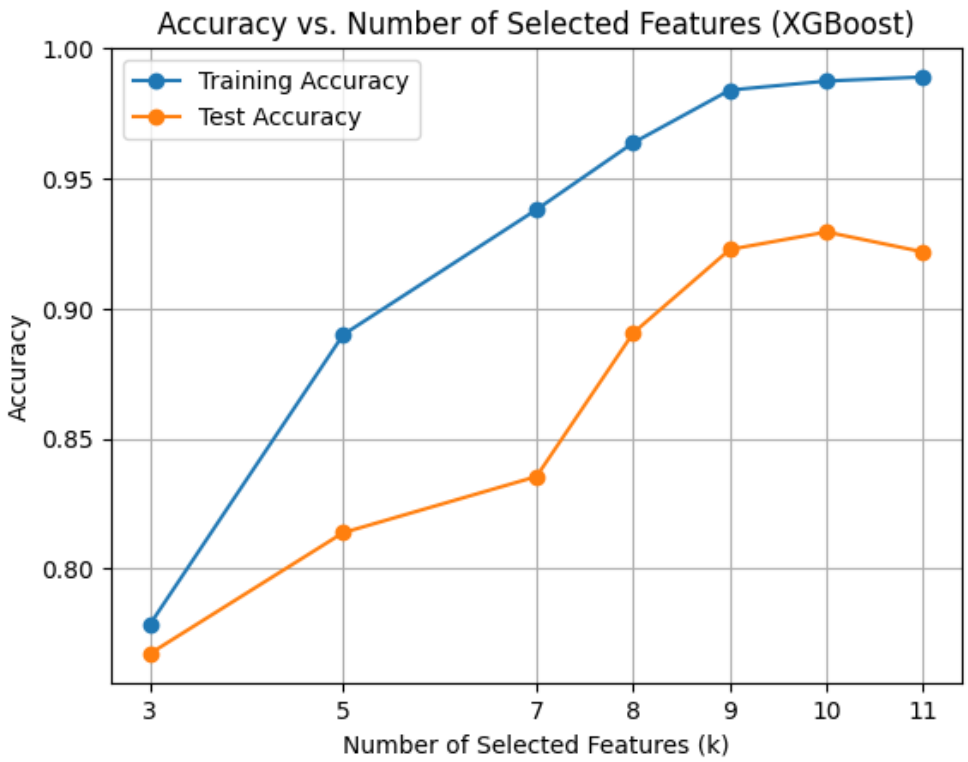
```
Cross-validation scores: [0.92441512 0.9160168  0.92436975 0.92376951 0.92617047]
Mean cross-validation score: 0.9229483275013666
Accuracy on test set: 0.9217850287907869
Confusion Matrix:
[[1064  111]
 [  52  857]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.91      0.93      1175
           1       0.89      0.94      0.91       909

    accuracy                           0.92      2084
   macro avg       0.92      0.92      0.92      2084
weighted avg       0.92      0.92      0.92      2084
```

ผลลัพธ์จากการทดลอง

5. การปรับปรุงคุณภาพโมเดล

ในการปรับปรุงคุณภาพของโมเดล ขั้นตอนแรกคือการเลือกคุณลักษณะที่สำคัญที่สุดจากชุดข้อมูล เพื่อให้โมเดลมีประสิทธิภาพในการทำนายมากที่สุด ในการทดลองนี้ ได้ใช้เทคนิคการเลือกคุณลักษณะที่ดีที่สุด (SelectKBest) โดยใช้การทดสอบ F-test สำหรับการเลือกคุณลักษณะที่มีผลกระทบต่อตัวแปรตาม (target variable) ในกรณีนี้คือ Current loan status

โดยในทดลองใช้ค่า k ที่เป็น [3, 5, 7, 8, 9, 10, 11] เพื่อดูผลกระทบของจำนวนคุณลักษณะที่ถูกเลือกต่อความแม่นยำของโมเดล โดยในแต่ละค่า k เราทำการเลือกคุณลักษณะจากชุดข้อมูลการฝึกและการทดสอบ จากนั้นนำไปใช้กับโมเดล XGBoost เพื่อทำการฝึกและทดสอบ หลังจากนั้นเราทดสอบการทำนายบนชุดข้อมูลการฝึกและการทดสอบเพื่อวัดความแม่นยำของโมเดล และบันทึกค่าความแม่นยำเพื่อนำมาวิเคราะห์ในขั้นตอนถัดไป ผลลัพธ์จะช่วยให้เราเลือกค่า k ที่เหมาะสมที่สุดสำหรับโมเดลที่มีประสิทธิภาพสูงสุด



จากภาพจะเห็นได้ว่า K=10 เป็นช่วงที่ดีที่สุดในช่วงการทดลองนี้ จากนั้นจะนำพารามิเตอร์ที่ดีที่สุด 10 ตัวแรกไปทำการปรับปรุงค่าพารามิเตอร์ของโมเดล XGBoost ต่อ เพื่อปรับปรุงประสิทธิภาพของโมเดล โดยใช้กระบวนการทำแบบจำลองทางการปรับปรุงเฉพาะของ XGBoost ที่ชื่อว่า Grid Search Cross Validation (GridSearchCV) ซึ่งจะทดสอบค่าพารามิเตอร์ต่าง ๆ และเลือกค่าที่ดีที่สุดที่ทำให้โมเดลมีประสิทธิภาพสูงสุด

ซึ่งในการทดลองนี้ กำหนดค่าพารามิเตอร์ที่เป็นไปได้สำหรับ XGBoost แต่ละอันได้แก่ learning_rate, max_depth, min_child_weight, subsample, colsample_bytree, และ n_estimators โดยในแต่ละค่าเราจะทำการทดลองที่ค่าที่กำหนดไว้เพื่อหาค่าที่ดีที่สุด ซึ่ง แต่ละค่าคือ learning_rates: อัตราการเรียนรู้ของโมเดล ค่าที่ใช้ส่วนใหญ่อยู่ระหว่าง 0 ถึง 1 โดยใช้เลขทศนิยม เพื่อระบุความสำคัญของการปรับข้อมูลในแต่ละรอบการฝึก

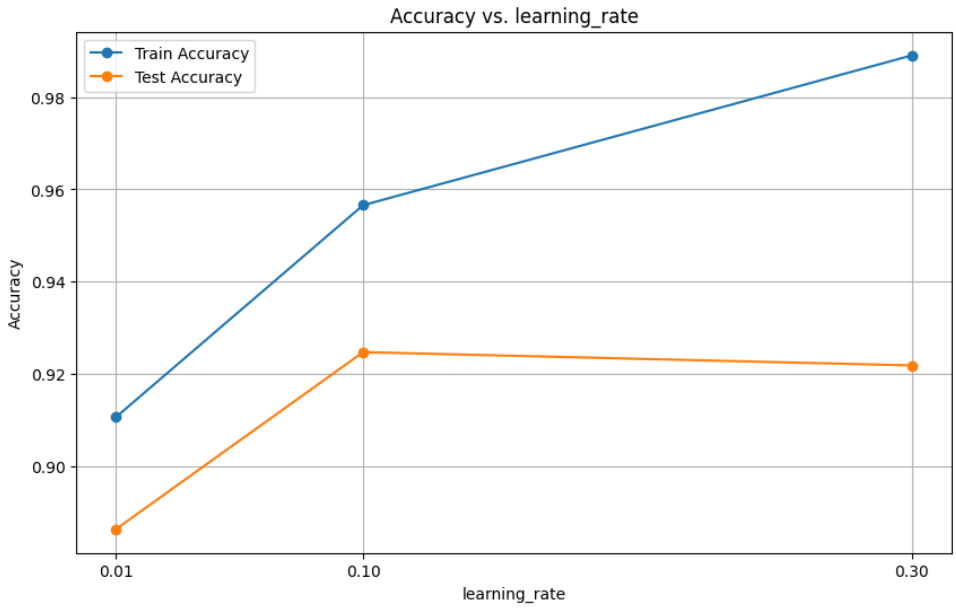
- max_depths: ความลึกสูงสุดของต้นไม้การตัดสินใจ กำหนดระดับของฟีเจอร์ที่ถูกใช้ในการแยกสาขาของต้นไม้ ค่าที่เลือกจะส่งผลต่อความซับซ้อนและความสัมพันธ์ในข้อมูล
- min_child_weights: น้ำหนักขั้นต่ำที่จำเป็นในการแยกสาขาของต้นไม้ ส่งผลต่อการควบคุมความยุ่งเหยิงของข้อมูลและการแยกสาขาของต้นไม้
- subsamples: สัดส่วนของตัวอย่างที่ถูกสุ่มเลือกเพื่อฝึกโมเดลในแต่ละรอบ การเลือกค่าที่เหมาะสมจะช่วยลดการเกิด Overfitting โดยการลดความเชื่อถือในข้อมูล

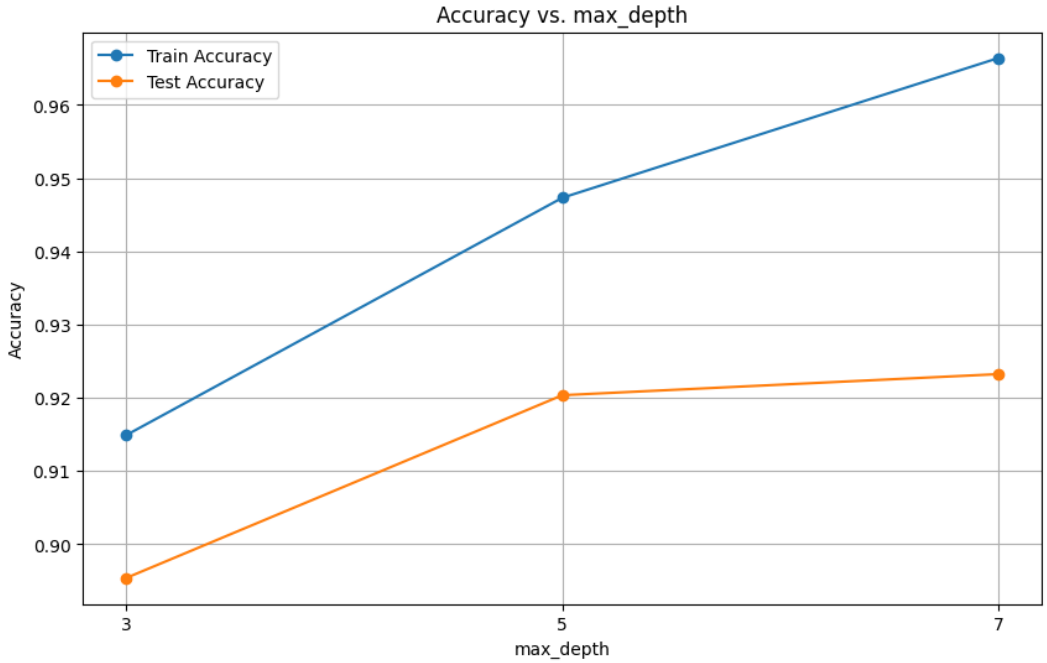
- colsample_bytrees: สัดส่วนของคอลัมน์ (features) ที่ถูกสุ่มเลือกเพื่อฝึกต้นไม้ตัดสินใจในแต่ละรอบ การเลือกค่าที่เหมาะสมจะช่วยลดความซับซ้อนของโมเดลและการเกิด Overfitting
- n_estimatorss: จำนวนของต้นไม้ในโมเดล ค่านี้กำหนดจำนวนของการฝึกซ้ำๆ โดยมีผลต่อความแม่นยำและความเสถียรของโมเดล โดยทั่วไปสมควรเลือกค่าที่มากพอเหมาะกับข้อมูลและขนาดของโมเดล

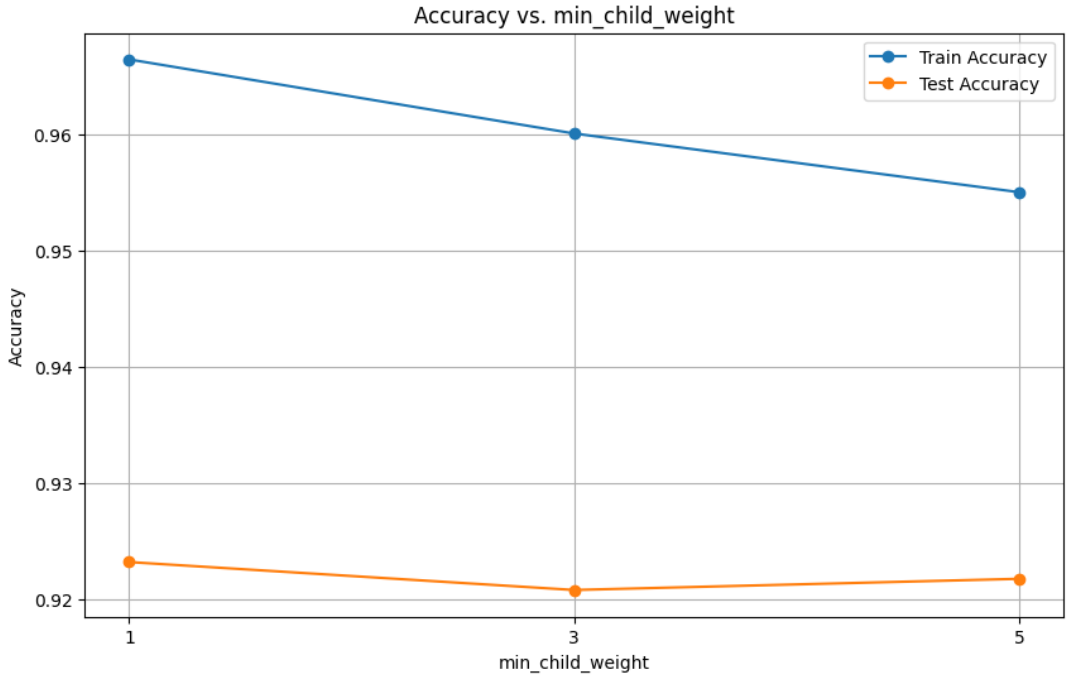
โดยค่าที่ใช้ในการทดลองคือ

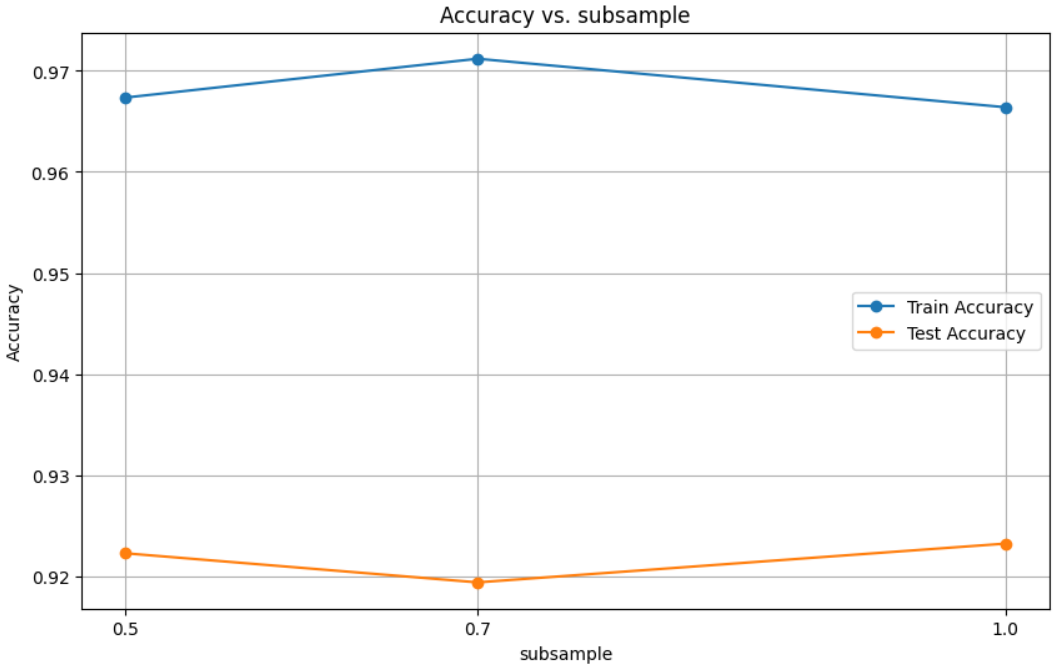
- learning_rates: [0.01, 0.1, 0.3]
- max_depths: [3, 5, 7]
- min_child_weights: [1, 3, 5]
- subsamples: [0.5, 0.7, 1.0]
- colsample_bytrees: [0.5, 0.7, 1.0]
- n_estimatorss: [50, 100, 200]

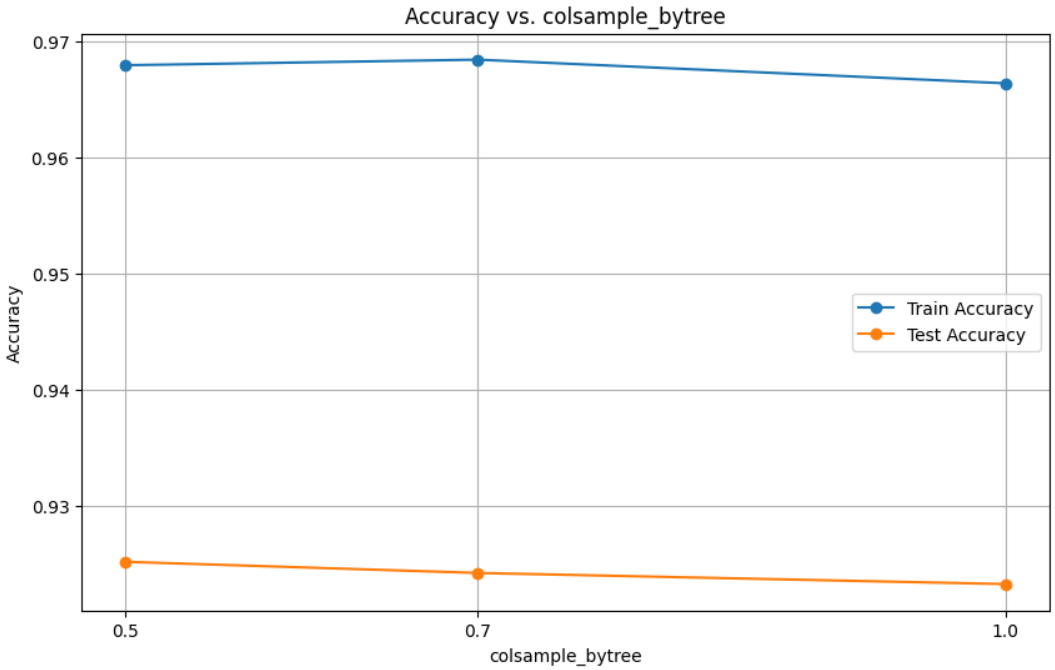
ในการทดลองได้ทำการเลือกค่าพารามิเตอร์ที่ดีที่สุดโดยใช้กระบวนการ GridSearchCV และแสดงผลการปรับปรุงค่าความแม่นยำของโมเดลทั้งในชุดข้อมูลการฝึกและการทดสอบผ่านกราฟ เมื่อเราได้ค่าพารามิเตอร์ที่ดีที่สุดแล้ว เราก็ทำการปรับโมเดล XGBoost ด้วยค่าพารามิเตอร์ที่ดีที่สุดนั้น และทดสอบการทำนายบนชุดข้อมูลการทดสอบ เพื่อวัดประสิทธิภาพของโมเดลที่ปรับปรุงแล้ว สุดท้ายเราจะได้โมเดลที่มีประสิทธิภาพสูงสุดและพร้อมใช้งานกับชุดข้อมูลที่มีอยู่ในปัจจุบัน

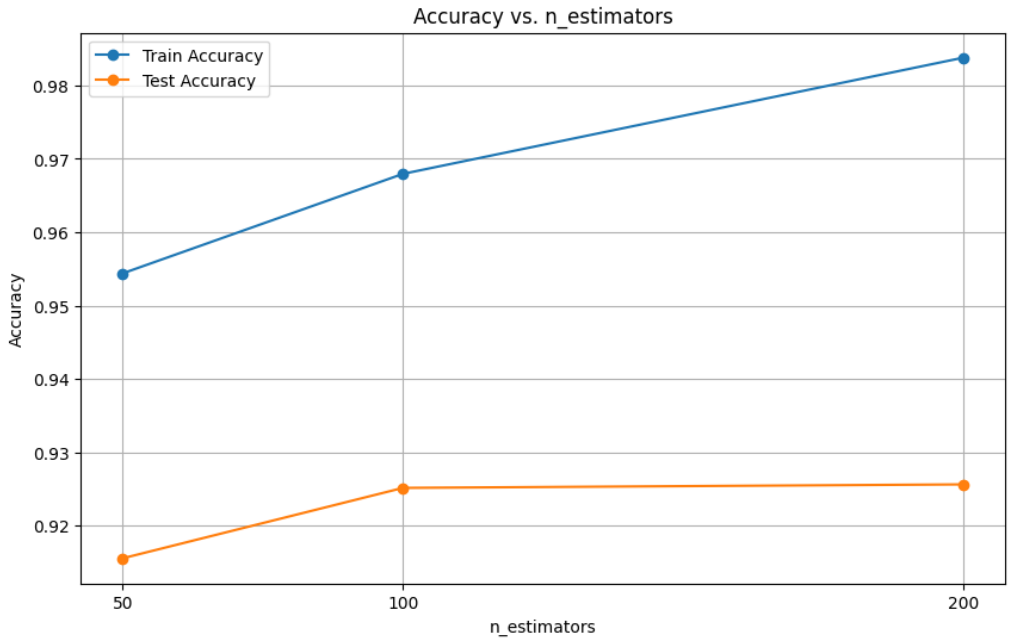



Accuracy vs. max_depth
Train Accuracy
Test Accuracy
0.96
0.95
0.94
0.93
0.92
0.91
0.90
Accuracy
3
5
7
max_depth


Accuracy vs. min_child_weight
Train Accuracy
Test Accuracy
0.96
0.95
0.94
0.93
0.92
Accuracy
1
3
5
min_child_weight


Accuracy vs. subsample
0.97
0.96
0.95
0.94
0.93
0.92
Accuracy
Train Accuracy
Test Accuracy
0.5
0.7
1.0
subsample


Accuracy vs. colsample_bytree
0.97
0.96
0.95
0.94
0.93
Accuracy
Train Accuracy
Test Accuracy
0.5
0.7
1.0
colsample_bytree



หลังจากการทำการปรับปรุงพารามิเตอร์ต่าง ๆ ของโมเดล XGBoost ด้วยกระบวนการปรับค่า (hyperparameter tuning) พบว่าพารามิเตอร์ที่ให้ผลลัพธ์ที่ดีที่สุดมีค่าดังนี้:

- learning_rate: 0.1
- max_depth: 7
- min_child_weight: 1
- subsample: 1.0
- colsample_bytree: 0.5
- n_estimators: 200

```
Cross-validation scores: [0.9096     0.9272     0.9208     0.9216     0.91993595]
Mean cross-validation score: 0.9198271897518013
Accuracy on test set: 0.9539236861051116
Confusion Matrix:
[[2243  137]
 [  55 1732]]
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.94      0.96      2380
           1       0.93      0.97      0.95      1787

    accuracy                           0.95      4167
   macro avg       0.95      0.96      0.95      4167
weighted avg       0.95      0.95      0.95      4167
```

ในการปรับปรุงโมเดล XGBoost ได้ใช้เทคนิค cross-validation เพื่อประเมินประสิทธิภาพของโมเดลโดยใช้ชุดข้อมูลที่มีอยู่ โดยการประเมินนี้ทำให้เราได้ค่าความแม่นยำ (accuracy) จากการทำ cross-validation ซึ่งอยู่ในช่วงระหว่าง 0.9096 ถึง 0.9272 และมีค่าเฉลี่ยของความแม่นยำทั้งหมดอยู่ที่ประมาณ 0.9198

หลังจากนั้นได้ทำการใช้โมเดลที่ปรับปรุงแล้วไปทดสอบกับชุดข้อมูลทดสอบ ซึ่งได้ค่าความแม่นยำบนชุดข้อมูลทดสอบอยู่ที่ประมาณ 0.9539 ซึ่งเป็นผลลัพธ์ที่ดีและแสดงให้เห็นถึงความสามารถในการทำนายของโมเดล

6. สรุปผลการทดลอง

จากจากทดสอบข้างต้น ได้พบว่าโมเดล XGBoost ประสิทธิภาพในการทำนายการอนุมัติสินเชื่อโดยใช้ข้อมูลทางการเงินและประวัติการกู้ยืมของลูกค้า มากที่สุด โดยใช้โดยการปรับพารามิเตอร์หลัก ได้แก่ learning_rate, max_depth, min_child_weight, subsample, colsample_bytree, และ n_estimators โดยผลการปรับปรุงพารามิเตอร์ให้มีค่าต่อไปนี้:

- learning_rate: 0.1

- max_depth: 7

- min_child_weight: 1

- subsample: 1.0

- colsample_bytree: 0.5

- n_estimators: 200

ผลการประเมินโมเดลด้วยการทำ cross-validation บนชุดข้อมูลฝึกและการทดสอบแสดงให้เห็นว่าโมเดลมีความแม่นยำเฉลี่ยอยู่ที่ประมาณ 91.98% ซึ่งเป็นค่าที่ดีต่อการทำนายโดยรวม ในขณะเดียวกัน การประเมินบนชุดข้อมูลทดสอบแสดงให้เห็นถึงความแม่นยำของโมเดลที่สูงอยู่ที่ประมาณ 95.39% โดยมีค่าความแม่นยำสูงสุดในการทำนายของกลุ่มคลาส 0 และค่า recall และ f1-score ที่สูงเป็นอย่างมากสำหรับคลาสทั้งสอง ซึ่งหมายความว่าโมเดลมีความสามารถในการจำแนกข้อมูลที่ดีและมีประสิทธิภาพอย่างสูง

7. อ้างอิง

1. ข้อมูลสำหรับการวิเคราะห์และทดสอบโมเดลถูกนำมาจากชุดข้อมูลที่เรียบเรียงโดย Prakash Raushan บนแพลตฟอร์ม Kaggle

   Prakash Raushan. (n.d.). Loan Dataset. Retrieved from Kaggle: https://www.kaggle.com/datasets/prakashraushan/loan-dataset

2. ในการนำเสนอผลลัพธ์ของโมเดลและการพัฒนาแอปพลิเคชันที่ใช้สำหรับการวิเคราะห์และสร้างรายงานสถิติเกี่ยวกับการอนุมัติสินเชื่อส่วนบุคคล สามารถพบได้จาก K-Loan App ตัวอย่างผลลัพธ์. เข้าถึงได้ที่: https://k-loan.streamlit.app/