CREDIT CARD FRAUD DETECTION WITH IMBALANCED DATA USING MACHINE LEARNING MODELS

ณัฐนันท์ สะเริญรัมย์

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2567

การตรวจจับการฉ้อโกงบัตรเครดิตที่ใช้ข้อมูลไม่สมดุลด้วยแบบจำลองการเรียนรู้ของเครื่อง

ณัฐนันท์ สะเริญรัมย์

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2567

CREDIT CARD FRAUD DETECTION WITH IMBALANCED DATA USING MACHINE LEARNING MODELS

NUTHANAN SAREANRAM

An Independent Study Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Data Science)

Faculty of Science, Srinakharinwirot University

2024

สารนิพนธ์

เรื่อง

การตรวจจับการฉ้อโกงบัตรเครดิตที่ใช้ข้อมูลไม่สมดุลด้วยแบบจำลองการเรียนรู้ของเครื่อง

ของ

ณัฐนันท์ สะเริญรัมย์

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร.นุวีย์ วิวัฒนวัฒนา)

..... ประธาน

(ผู้ช่วยศาสตราจารย์ ดร.อัครินทร์ ไพบูลย์พานิช)

..... กรรมการ

(อาจารย์ ดร.บรรพตรี คมขำ)

ชื่อเรื่อง	การตรวจจับการฉ้อโกงบัตรเครดิตที่ใช้ข้อมูลไม่สมดุลด้วยแบบจำลองการเรียนรู้ของเครื่อง
ผู้วิจัย	ณัฐนันท์ สะเริญรัมย์
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2567
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. นุวีย์ วิวัฒนวัฒนา

การฉ้อโกงบัตรเครดิตเป็นปัญหาที่ส่งผลกระทบทางการเงินอย่างรุนแรงต่อทั้งผู้บริโภคและสถาบันการเงิน การพัฒนาแบบจำลองที่สามารถตรวจจับธุรกรรมผิดปกติได้อย่างแม่นยำจึงมีความสำคัญอย่างยิ่ง โดยเฉพาะเมื่อข้อมูลธุรกรรมมีลักษณะไม่สมดุล (Imbalanced Data) ซึ่งส่งผลให้การเรียนรู้ของแบบจำลองมีข้อจำกัด งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาแบบจำลองให้สามารถตรวจจับธุรกรรมทุจริตได้อย่างมีประสิทธิภาพ โดยแบ่งการดำเนินการออกเป็นสองส่วน คือ (1) การเปรียบเทียบประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องจำนวน 5 แบบ ได้แก่ Decision Tree, Random Forest, XGBoost, K-Nearest Neighbors และ CatBoost พร้อมการปรับพารามิเตอร์ให้เหมาะสม และ (2) การนำแบบจำลองที่มีประสิทธิภาพสูงสุดมาทดสอบร่วมกับเทคนิคการจัดการข้อมูลไม่สมดุล 6 เทคนิค ได้แก่ Random Oversampling, SMOTE, Tomek Links, Edited Nearest Neighbors (ENN), SMOTEENN และ SMOTETomek โดยข้อมูลที่ใช้ในการศึกษาเป็นข้อมูลธุรกรรมบัตรเครดิตที่ประกอบด้วยข้อมูลเชิงธุรกรรม ข้อมูลพื้นที่ และข้อมูลผู้ถือบัตร ผลการทดลองในส่วนแรกพบว่าแบบจำลอง XGBoost ให้ผลลัพธ์ดีที่สุดในการจำแนกธุรกรรมทุจริต โดยให้ค่า Precision เท่ากับ 0.96, Recall เท่ากับ 0.83 และ F1-Score เท่ากับ 0.89 ในการทดลองส่วนที่สอง พบว่าเมื่อใช้ XGBoost ร่วมกับเทคนิค SMOTE และ SMOTETomek แบบจำลองมีประสิทธิภาพดีมากขึ้น โดยให้ค่า Precision เท่ากับ 0.97,Recall เท่ากับ 0.86 และ F1-Score เท่ากับ 0.91 ทั้งนี้ แบบจำลองยังสามารถจำแนกธุรกรรมปกติ (Class 0) ได้อย่างแม่นยำ โดยมีค่า Accuracy เท่ากับ 1 ในทุกตัวชี้วัด ผลการวิจัยชี้ให้เห็นว่า การใช้เทคนิคการจัดการข้อมูลไม่สมดุลควบคู่กับแบบจำลองที่มีประสิทธิภาพ ช่วยเพิ่มประสิทธิภาพในการตรวจจับธุรกรรมทุจริตเพิ่มมากขึ้น

คำสำคัญ : การตรวจจับการฉ้อโกง, บัตรเครดิต, การเรียนรู้ของเครื่อง, ข้อมูลไม่สมดุล, การจำแนกประเภท

Title	CREDIT CARD FRAUD DETECTION WITH IMBALANCED DATA USING MACHINE LEARNING MODELS
Author	NUTHANAN SAREANRAM
Degree	MASTER OF SCIENCE
Academic Year	2024
Thesis Advisor	Assistant Professor Dr. Nuwee Wiwatwattana

Credit card fraud poses a significant financial threat to both consumers and financial institutions, particularly when transaction data are imbalanced, which limits model performance. This study aims to develop an effective fraud detection model using a two-stage approach. First, five machine learning algorithms—Decision Tree, Random Forest, XGBoost, K-Nearest Neighbors, and CatBoost—were compared and optimized to identify the most suitable model for classifying fraudulent transactions. XGBoost achieved the highest performance, with a precision of 0.96, recall of 0.83, and F1-score of 0.89. In the second stage, the model was further evaluated using six data imbalance handling techniques: Random Oversampling, SMOTE, Tomek Links, Edited Nearest Neighbors (ENN), SMOTEENN, and SMOTETomek. The integration of XGBoost with SMOTE and SMOTETomek improved the model's performance, achieving a precision of 0.97, recall of 0.86, and F1-score of 0.91. Furthermore, the model demonstrated perfect accuracy in identifying non-fraudulent transactions. These results highlight that combining robust classification algorithms with effective data imbalance handling techniques significantly enhances fraud detection capabilities and supports practical implementation in financial systems.

Keyword : Fraud Detection, Credit Card, Machine Learning, Imbalanced Data, Classification

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูงต่อ ผศ.ดร. นุวีย์ วิวัฒนวัฒนา อาจารย์ที่ปรึกษา ซึ่งได้ให้คำแนะนำ คำปรึกษา ตลอดจนข้อเสนอแนะที่มีคุณค่าในทุกขั้นตอนของการดำเนินงานวิจัย ทำให้งานวิจัยฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

นอกจากนี้ ข้าพเจ้าขอขอบคุณ ผศ.ดร. อัครินทร์ ไพบูลย์พานิช และ อ.ดร. บรรพตรี คมขำ ซึ่งเป็นกรรมการสอบที่ได้ให้ข้อเสนอแนะอันเป็นประโยชน์อย่างยิ่งต่อการปรับปรุงและพัฒนางานวิจัยฉบับนี้ให้สมบูรณ์ยิ่งขึ้น

ข้าพเจ้าขอแสดงความขอบคุณต่อคณาจารย์ทุกท่านในสาขาวิชาวิทยาการข้อมูล และเจ้าหน้าที่ของคณะวิทยาศาสตร์ ที่ให้ความช่วยเหลือ อำนวยความสะดวก และส่งเสริมการเรียนรู้ตลอดระยะเวลาที่ข้าพเจ้าได้ศึกษาในหลักสูตรนี้

ท้ายที่สุด ข้าพเจ้าขอขอบคุณครอบครัว เพื่อน และบุคคลใกล้ชิดทุกท่านที่เป็นกำลังใจสำคัญในการเรียนและการทำวิจัยครั้งนี้เป็นอย่างยิ่ง

ณัฐนันท์ สะเริญรัมย์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฎ

สารบัญรูปภาพ ..... ฑ

บทที่ 1 บทนำ 1

1. ความเป็นมาและความสำคัญของปัญหา 1

2. จุดประสงค์ของงานวิจัย 3

3. ขอบเขตงานวิจัย 4

4. กรอบแนวคิดในการวิจัย 5

5. ประโยชน์ที่คาดว่าจะได้รับ 6

บทที่ 2 ทฤษฎี และงานวิจัยที่เกี่ยวข้อง 7

1. ทฤษฎีเกี่ยวกับอัลกอริทึมในการจำแนกประเภท (Classification Algorithms) 7

1.1 Decision Tree (DT) 7

1.2 Random Forest (RF) 9

1.3 Extreme Gradient Boosting (XGBoost) 11

1.4 K-Nearest Neighbors (KNN) 12

1.5 Categorical Boosting (CatBoost) 14

2. ทฤษฎีเกี่ยวกับการวัดประสิทธิภาพแบบจำลองการจำแนกประเภท 15

2.1 เมทริกซ์ความสับสน (Confusion Matrix) 15

2.2 ค่าความถูกต้อง (Accuracy) 16
2.3 ค่าความเที่ยง (Precision) 16
2.4 ค่าเรียกคืน (Recall) 17
2.5 F1 Score 17
3. วิศวกรรมคุณลักษณะ (Feature engineering) 18
3.1 กระบวนการทำ Standardization 18
3.2 การแปลงข้อมูลประเภทหมวดหมู่ (Categorical data transformation) 18
3.2.1 One-Hot (One-hot encoding) 19
3.2.2 Label Encoding 19
4. เทคนิคการจัดการกับข้อมูลที่ไม่สมดุล 20
4.1 Random Oversampling (ROS) 20
4.2 Synthetic Minority Oversampling Technique (SMOTE) 21
4.3 Tomek Links 22
4.4 Edited Nearest Neighbors (ENN) 23
4.5 SMOTEENN 24
4.6 SMOTETomek 24
5. งานวิจัยที่เกี่ยวข้อง 25
บทที่ 3 วิธีดำเนินการวิจัย 30
1. กระบวนการทำงานของแบบจำลองการเรียนรู้ของเครื่อง 31
2. การเก็บรวบรวมข้อมูล (Data Collection) 32
3. การสำรวจข้อมูล (Exploratory Data Analysis: EDA) 34
4. การเตรียมข้อมูล (Data preparation) 41
4.1 การทำความสะอาดข้อมูล 41

4.2 การเปลี่ยนข้อมูลแบบหมวดหมู่และตัวเลข 41
4.3 การจัดการกับข้อมูลที่ไม่สมดุล 42
5. การสร้างแบบจำลอง 46
5.1 การสร้างแบบจำลองด้วยชุดข้อมูลไม่สมดุล 46
5.2 การสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยชุดข้อมูลที่มีการจัดกาข้อมูลไม่สมดุล 50
บทที่ 4 ผลการดำเนินงานวิจัย 57
1. การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้ข้อมูลที่ไม่สมดุล 58
1.1 Decision Tree (DT) 58
1.2 Random Forest (RF) 59
1.3 Extreme Gradient Boosting (XGBoost) 60
1.4 K-Nearest Neighbors (KNN) 61
1.5 Categorical Boosting (CatBoost) 62
2.การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้ข้อมูลที่ผ่านการปรับสมดุล 64
2.1 Random Over-Sampling (ROS) 65
2.2 Synthetic Minority Over-sampling Technique (SMOTE) 66
2.3. Tomek Links (TL) 67
2.4 Edited Nearest Neighbors (ENN) 68
2.5 SMOTEENN 69
2.6 SMOTETomek 70
บทที่ 5 สรุปผลการทดลอง 73
1. สรุปผลการวิจัย 73
2. อภิปรายผลการวิจัย 77

3. ข้อเสนอแนะ.....79

บรรณานุกรม 2

ประวัติผู้เขียน.....6

สารบัญตาราง

หน้า

ตาราง 1 Confusion matrix สำหรับการจำแนกประเภทแบบไบนารี 16
ตาราง 2 สรุปงานวิจัยที่เกี่ยวข้อง 28
ตาราง 3 ตัวแปรของข้อมูลธุรกรรมบัตรเครดิต 33
ตาราง 4 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค Random Oversampling 43
ตาราง 5 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค SMOTE 43
ตาราง 6 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค Tomek Links 44
ตาราง 7 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลที่ไม่สมดุลด้วย ENN 45
ตาราง 8 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค SMOTEENN 45
ตาราง 9 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลที่ไม่สมดุลด้วย SMOTETomek 46
ตาราง 10 ค่าพารามิเตอร์ของแบบจำลอง Decision Tree 47
ตาราง 11ค่าพารามิเตอร์ของแบบจำลอง Random Forest 48
ตาราง 12 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) 48
ตาราง 13 ค่าพารามิเตอร์ของแบบจำลอง K-Nearest Neighbors (KNN) 49
ตาราง 14 ค่าพารามิเตอร์ของแบบจำลอง Categorical Boosting (CatBoost) 50
ตาราง 15 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล ROS 51
ตาราง 16 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล SMOTE 52
ตาราง 17 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล TL 53
ตาราง 18 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล ENN 54

ตาราง 19 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล SMOTEENN 55

ตาราง 20 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล SMOTETomek 56

ตาราง 21 ผลการวัดประสิทธิภาพของแบบจำลอง Decision Tree 58

ตาราง 22 ผลการวัดประสิทธิภาพของแบบจำลอง Random Forest 59

ตาราง 23 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost 60

ตาราง 24 ผลการวัดประสิทธิภาพของแบบจำลอง K-Nearest Neighbors (KNN) 61

ตาราง 25 ผลการวัดประสิทธิภาพของแบบจำลอง CatBoost 62

ตาราง 26 ผลการวัดประสิทธิภาพของแบบจำลอง โดยไม่ได้ใช้เทคนิคจัดการข้อมูลไม่สมดุลของธุรกรรมปกติ (Class 0) 63

ตาราง 27 ผลการวัดประสิทธิภาพของแบบจำลอง โดยไม่ได้ใช้เทคนิคจัดการข้อมูลไม่สมดุลของธุรกรรมทุจริต (Class1) 63

ตาราง 28 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล Random Oversampling 65

ตาราง 29 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล SMOTE 66

ตาราง 30 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล Tomek Links 67

ตาราง 31 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล ENN 68

ตาราง 32 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล SMOTEENN 69

ตาราง 33 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล SMOTETomek 71

ตาราง 34 ผลการทดลองของอัลกอริทึม XGBoost ที่มีการจัดการข้อมูลมีสมดุล (Class 0) 71

ตาราง 35 ผลการทดลองของอัลกอริทึม XGBoost ที่มีการจัดการข้อมูลมีสมดุล (Class 1) 71

ตาราง 36 ผลการทดลองของอัลกอริทึมการจำแนกประเภทต่างๆ โดยไม่ได้ใช้เทคนิคจัดการข้อมูลไม่สมดุล 74

ตาราง 37 ผลการทดลองของอัลกอริทึม XGBoost โดยใช้เทคนิคการจัดการความไม่สมดุลข้อมูล
..... 75

สารบัญรูปภาพ

หน้า

ภาพประกอบ 1 แบบจำลองพื้นฐาน Decision Tree 8
ภาพประกอบ 2 แบบจำลองพื้นฐาน Random Forest 10
ภาพประกอบ 3 แบบจำลองพื้นฐาน XGBoost 12
ภาพประกอบ 4 แบบจำลองพื้นฐาน K-Nearest Neighbor (KNN) 13
ภาพประกอบ 5 แบบจำลองพื้นฐาน CatBoost 14
ภาพประกอบ 6 One-Hot Encoding Technique 19
ภาพประกอบ 7 One-Hot Encoding Technique 20
ภาพประกอบ 8 Random Oversampling Technique 21
ภาพประกอบ 9 Synthetic Minority Oversampling Technique 22
ภาพประกอบ 10 Tomek Links Technique 23
ภาพประกอบ 11 Edited Nearest Neighbors (ENN) Technique 24
ภาพประกอบ 12 กระบวนการทำงานของแบบจำลอง 31
ภาพประกอบ 13 แสดงถึงลักษณะของธุรกรรมบัตรเครดิต 34
ภาพประกอบ 14 แสดงถึงเพศของผู้ถือบัตรเครดิต 35
ภาพประกอบ 15 แสดงการกระจายตัวของอายุ 36
ภาพประกอบ 16 แสดงถึงอาชีพของผู้ถือบัตรเครดิต 37
ภาพประกอบ 17 แสดงถึงกลุ่มธุรกรรม 38
ภาพประกอบ 18 แสดงถึงจำนวนเงินในธุรกรรม 39
ภาพประกอบ 19 แสดงถึงจำนวนธุรกรรมบัตรเครดิตตามช่วงเวลา 40
ภาพประกอบ 20 Confusion Matrix Decision Tree 58
ภาพประกอบ 21 Confusion Matrix Random Forest 59

ภาพประกอบ 22 Confusion Matrix XGBoost 60
ภาพประกอบ 23 Confusion Matrix K-Nearest Neighbors (KNN) 61
ภาพประกอบ 24 Confusion Matrix Categorical Boosting (CatBoost) 62
ภาพประกอบ 25 Confusion Matrix XGBoost กับข้อมูล ROS 65
ภาพประกอบ 26 Confusion Matrix XGBoost กับข้อมูล SMOTE 66
ภาพประกอบ 27 Confusion Matrix XGBoost กับข้อมูล Tomek Links 67
ภาพประกอบ 28 Confusion Matrix XGBoost กับข้อมูล ENN 68
ภาพประกอบ 29 Confusion Matrix XGBoost กับข้อมูล SMOTEENN 69
ภาพประกอบ 30 Confusion Matrix XGBoost กับข้อมูล SMOTETomek 70

บทที่ 1
บทนำ

1. ความเป็นมาและความสำคัญของปัญหา

บัตรเครดิตได้กลายเป็นเครื่องมือทางการเงินที่มีบทบาทสำคัญในชีวิตประจำวันของผู้คน เนื่องจากเป็นวิธีการชำระเงินที่ได้รับความนิยมอย่างแพร่หลายในหลากหลายสถานการณ์ ทั้งการซื้อสินค้าและบริการผ่านช่องทางออนไลน์ และการชำระเงินแบบทั่วไป การพัฒนาของธุรกิจออนไลน์ที่เติบโตอย่างต่อเนื่องควบคู่ไปกับเทคโนโลยีการชำระเงินแบบดิจิทัล ทำให้ผู้คนสามารถหลีกเลี่ยงการพกเงินสดจำนวนมากติดตัว ซึ่งช่วยลดความเสี่ยงและเพิ่มความสะดวกสบายในการใช้จ่ายได้อย่างมีประสิทธิภาพ สิ่งที่ทำให้บัตรเครดิตน่าสนใจเป็นพิเศษคือสิทธิประโยชน์ที่มอบให้กับผู้ถือบัตร ไม่ว่าจะเป็นการซื้อสินค้าโดยไม่ต้องชำระเงินทันที แต่สามารถจ่ายคืนในภายหลัง ซึ่งช่วยให้ผู้ใช้สามารถบริหารจัดการค่าใช้จ่ายได้อย่างยืดหยุ่น นอกจากนี้ ยังมีโปรแกรมผ่อนชำระสินค้าและบริการที่มักมาพร้อมกับอัตราดอกเบี้ย 0% ในระยะเวลาที่กำหนด ซึ่งเหมาะสำหรับการซื้อสินค้าที่มีราคาสูง เช่น เครื่องใช้ไฟฟ้า หรืออุปกรณ์อิเล็กทรอนิกส์ ทำให้ผู้ใช้สามารถซื้อสินค้าที่ต้องการได้แม้จะมีงบประมาณจำกัด นอกเหนือจากความสะดวกในด้านการชำระเงิน บัตรเครดิตยังมีโปรแกรมสะสมคะแนน หรือ Rewards Program ซึ่งช่วยให้ผู้ใช้ได้รับคะแนนสะสมทุกครั้งที่ทำการซื้อสินค้า คะแนนเหล่านี้สามารถนำไปแลกเป็นของรางวัล เช่น บัตรของขวัญจากร้านค้าชั้นนำ ตั๋วเครื่องบิน หรือแม้กระทั่งการแลกเป็นเงินคืน (Cashback) เพื่อนำไปใช้ในโอกาสถัดไป ตัวอย่างเช่น ธนาคารบางแห่งมีโปรโมชั่นคืนเงินสด 2% จากการใช้จ่ายในซูเปอร์มาร์เก็ต หรือ 5% สำหรับการทานอาหารในร้านอาหารที่ร่วมรายการ สิ่งเหล่านี้ช่วยให้ผู้ถือบัตรเครดิตสามารถประหยัดค่าใช้จ่ายได้ในชีวิตประจำวัน นอกจากนี้ บัตรเครดิตยังมอบข้อเสนอพิเศษที่ตอบโจทย์ไลฟ์สไตล์ของผู้ใช้ เช่น ส่วนลดพิเศษจากร้านค้าชั้นนำ การเข้าถึงห้องรับรองผู้โดยสารพิเศษในสนามบิน หรือบริการช่วยเหลือฉุกเฉินบนท้องถนน สิทธิพิเศษเหล่านี้ถูกออกแบบมาให้เหมาะสมกับประเภทบัตร ทั้งหมดนี้แสดงให้เห็นถึงประโยชน์และความคุ้มค่าที่บัตรเครดิตมอบให้กับผู้ถือบัตร

การที่บัตรเครดิตได้รับความนิยมอย่างแพร่หลายนั้น นอกจากจะเพิ่มความสะดวกสบายในการชำระเงินและการใช้จ่ายแล้ว ยังนำมาซึ่งความเสี่ยงด้านความปลอดภัยทางการเงิน โดยเฉพาะอย่างยิ่งแนวโน้มการโจรกรรมบัตรเครดิตที่เพิ่มสูงขึ้นตามการใช้งานที่แพร่หลาย ซึ่งไม่เพียงแต่ส่งผลกระทบต่อผู้บริโภค แต่ยังส่งผลต่อสถาบันการเงินที่ให้บริการบัตรเครดิตอีกด้วย ดังนั้น การตรวจจับและป้องกันการโจรกรรมจึงเป็นสิ่งสำคัญในการลดผลกระทบและความ

เสียหายที่อาจเกิดขึ้น สำหรับผู้บริโภค การถูกโจรกรรมบัตรเครดิตอาจนำมาซึ่งความสูญเสียในหลาย ๆ ด้าน ไม่เพียงแต่ในด้านการเงิน เช่น การสูญเสียเงินจากบัญชีหรือวงเงินบัตรเครดิต แต่ยังรวมถึงเวลาที่ต้องเสียไปในการยื่นคำร้องเพื่อขอคืนเงินจากสถาบันการเงิน ซึ่งกระบวนการเหล่านี้มักสร้างความยุ่งยากและความเครียดให้กับผู้เสียหาย นอกจากนี้ เหตุการณ์ดังกล่าวยังอาจทำให้ผู้บริโภคขาดความมั่นใจในระบบความปลอดภัยของบัตรเครดิตและสถาบันการเงิน ส่งผลให้มีการลดการใช้งานหรือเปลี่ยนไปใช้บริการของสถาบันอื่น ในด้านสถาบันการเงินที่ให้บริการบัตรเครดิต ผลกระทบจากการโจรกรรมข้อมูลอาจมีความรุนแรงยิ่งขึ้น เนื่องจากนอกจากจะต้องรับผิดชอบความเสียหายทางการเงิน เช่น การคืนเงินให้แก่ลูกค้าแล้ว ยังต้องเผชิญกับความเสี่ยงต่อการสูญเสียความเชื่อมั่นจากผู้บริโภค ซึ่งอาจส่งผลกระทบต่อชื่อเสียงและความน่าเชื่อถือขององค์กรในระยะยาว ความเสียหายที่เกิดขึ้นอาจลดจำนวนลูกค้าและสร้างผลกระทบต่อการดำเนินธุรกิจในอนาคต

ด้วยความเสี่ยงที่เพิ่มสูงขึ้นจากการโจรกรรมบัตรเครดิต สถาบันการเงินและผู้ให้บริการบัตรเครดิตจำเป็นต้องดำเนินมาตรการรักษาความปลอดภัยที่มีประสิทธิภาพและทันสมัยมากขึ้น เพื่อลดความเสี่ยงและป้องกันความเสียหายที่อาจเกิดขึ้น ในอดีต สถาบันการเงินมักใช้วิธีการแบบดั้งเดิม เช่น การตั้งกฎเกณฑ์ล่วงหน้า (Rule-Based) เพื่อระบุพฤติกรรมที่น่าสงสัย การตรวจสอบด้วยแรงงานมนุษย์จากฝ่ายที่ดูแลการทุจริต (Fraud Team) หรือการตรวจจับโดยอาศัยประวัติการใช้งานย้อนหลัง ซึ่งวิธีเหล่านี้แม้จะสามารถช่วยคัดกรองธุรกรรมที่ผิดปกติได้ในระดับหนึ่ง แต่ก็มีข้อจำกัดหลายอย่าง เช่น ไม่สามารถตรวจจับพฤติกรรมใหม่ ๆ ได้ทันเวลา มีความล่าช้าในการตอบสนองต่อเหตุการณ์ที่เกิดขึ้นจริง และมีอัตราการแจ้งเตือนผิดพลาดสูง ส่งผลให้ลูกค้าถูกระงับการใช้งานโดยไม่จำเป็น หรือได้รับประสบการณ์การใช้งานที่ไม่ดี อีกทั้งยังต้องใช้ทรัพยากรบุคคลจำนวนมากในการติดตาม ตรวจสอบ และวิเคราะห์ข้อมูลธุรกรรมที่น่าสงสัยอย่างต่อเนื่อง ซึ่งกระบวนการเหล่านี้ไม่เพียงเพิ่มภาระให้กับบุคลากรภายในองค์กร แต่ยังไม่สามารถรองรับปริมาณข้อมูลที่เพิ่มขึ้นอย่างรวดเร็วในยุคดิจิทัลได้อย่างมีประสิทธิภาพ ด้วยข้อจำกัดเหล่านี้ จึงทำให้เกิดการพัฒนาและนำเทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning) มาใช้ในการตรวจจับพฤติกรรมการใช้จ่ายที่ผิดปกติ เทคโนโลยีนี้มีความสามารถในการวิเคราะห์ข้อมูลจำนวนมหาศาลในเวลาอันรวดเร็ว ด้วยการประมวลผลข้อมูลจากการใช้งานในอดีต และสร้างรูปแบบพฤติกรรมของผู้ถือบัตร เทคโนโลยี Machine Learning สามารถระบุธุรกรรมที่มีลักษณะผิดปกติหรือมีความเสี่ยงต่อการทุจริตได้อย่างแม่นยำ ซึ่งช่วยให้สถาบันการเงินสามารถตอบสนองต่อภัยคุกคามได้อย่างทันท่วงที ลดความสูญเสียทางการเงิน และเพิ่มประสิทธิภาพในการป้องกันการโจรกรรม

ข้อมูล และยังเพิ่มความปลอดภัยให้กับผู้ใช้งาน มาตรการเหล่านี้ช่วยลดโอกาสที่บุคคลภายนอกจะสามารถเข้าถึงข้อมูลส่วนตัวหรือทำธุรกรรมโดยไม่ได้รับอนุญาต ทำให้ผู้ถือบัตรสามารถใช้บัตรเครดิตได้อย่างมั่นใจมากยิ่งขึ้น

แม้ว่าสถาบันการเงินและผู้ให้บริการบัตรเครดิต จะนำมาตรการป้องกันที่เข้มงวดและเทคโนโลยีทันสมัยมาช่วยลดความเสี่ยงจากการทุจริต แต่กลับพบปัญหาสำคัญที่เกี่ยวข้องกับความไม่สมดุลของข้อมูลที่ใช้ในการพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งโดยธรรมชาติแล้ว ธุรกรรมที่เป็นการทุจริตมีจำนวนที่น้อยมากเมื่อเทียบกับธุรกรรมปกติ ความไม่สมดุลนี้เป็นความท้าทายที่สำคัญ เนื่องจากข้อมูลส่วนใหญ่ที่ใช้ในการฝึกฝนแบบจำลองจะมาจากธุรกรรมปกติ ซึ่งมีปริมาณมหาศาลเมื่อเทียบกับธุรกรรมที่เป็นการทุจริต เมื่อแบบจำลอง Machine Learning ถูกฝึกด้วยข้อมูลที่ไม่สมดุล แบบจำลองมักจะเรียนรู้และปรับตัวตามข้อมูลที่พบมากกว่า นั่นคือข้อมูลของธุรกรรมปกติ ส่งผลให้แบบจำลองให้ความสำคัญกับการวิเคราะห์และการทำนายธุรกรรมปกติเป็นหลัก ในขณะที่ความสามารถในการตรวจจับธุรกรรมที่เป็นการทุจริตอาจถูกลดความสำคัญลง ปัญหานี้ส่งผลให้แบบจำลองมีแนวโน้มที่จะตรวจไม่พบธุรกรรมทุจริต หรืออาจระบุได้ไม่แม่นยำเพียงพอ ซึ่งถือเป็นอุปสรรคที่สำคัญในการใช้งานจริงของแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) การแก้ไขปัญหานี้จำเป็นต้องใช้เทคนิคการประมวลผลข้อมูลเฉพาะ เช่น การสร้างข้อมูลที่สมดุลโดยใช้เทคนิค Oversampling หรือ SMOTE เพื่อเพิ่มธุรกรรมที่เป็นการทุจริต หรือการใช้วิธีการประเมินผลที่เน้นความแม่นยำในการตรวจจับธุรกรรมทุจริตโดยเฉพาะ การพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ที่สามารถรับมือกับข้อมูลที่ไม่สมดุลอย่างมีประสิทธิภาพจึงเป็นหัวใจสำคัญในการยกระดับการป้องกันการทุจริตทางการเงินให้มีความแม่นยำและครอบคลุมมากยิ่งขึ้น

งานวิจัยนี้เน้นศึกษา วิเคราะห์ และเปรียบเทียบวิธีการจัดการกับปัญหาข้อมูลที่ไม่สมดุล (Imbalanced Data) และเปรียบเทียบประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ในการจำแนกประเภท (Classification) โดยทดลองกับชุดข้อมูลธุรกรรมบัตรเครดิต ซึ่งมีจำนวนข้อมูลทั้งหมด 1,296,675 แถว และ 24 ตัวแปร

2. จุดประสงค์ของงานวิจัย

ในการวิจัยครั้งนี้ผู้วิจัยได้ตั้งความมุ่งหมายไว้ดังนี้

1. เพื่อสร้างแบบจำลองแบ่งกลุ่มธุรกรรมบัตรเครดิต โดยจำแนกธุรกรรมบัตรเครดิตเป็น 2 ประเภท (Class) คือ ธุรกรรมปกติ และธุรกรรมทุจริต โดยแบบจำลองที่นำมาใช้ในการศึกษาประกอบไปด้วย 5 แบบจำลอง ได้แก่

1. Decision Tree (DT)

2. Random Forest (RF)

3. Extreme Gradient Boosting (XGBoost)

4. K-Nearest Neighbors (KNN)

5. Categorical Boosting (CatBoost)

พร้อมทั้งเปรียบเทียบและประเมินประสิทธิภาพของแบบจำลอง หลังจากดำเนินการปรับค่าพารามิเตอร์(Parameter Tuning) เพื่อให้แต่ละแบบจำลองเหมาะสมสำหรับข้อมูล ซึ่งช่วยให้สามารถวิเคราะห์ผลลัพธ์ได้อย่างแม่นยำ

2. ประยุกต์ใช้เทคนิคในการจัดการกับปัญหาข้อมูลที่ไม่สมดุล (Imbalanced Data) ซึ่งเป็นปัญหาสำคัญที่มักพบในงานวิเคราะห์ข้อมูลและการพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) โดยปัญหาดังกล่าวเกิดขึ้นเมื่อจำนวนข้อมูลในแต่ละประเภท (Class) ของกลุ่มเป้าหมาย (Target Classes) มีความแตกต่างกันอย่างมาก โดยใช้เทคนิคการจัดการข้อมูลที่ไม่สมดุลดังนี้

1. Random Over-Sampling (ROS)

2. Synthetic Minority Over-sampling Technique (SMOTE)

3. Tomek Links (TL)

4. Edited Nearest Neighbors (ENN)

5. SMOTEENN

6. SMOTETomek

3. ขอบเขตงานวิจัย

1. ชุดข้อมูลที่ใช้ในการวิจัย

สำหรับข้อมูลที่ใช้ในการทดลองเป็นชุดข้อมูลธุรกรรมบัตรเครดิตขนาดใหญ่จากแหล่งข้อมูลสาธารณะ Kaggle.com ชุดข้อมูลธุรกรรมบัตรเครดิตนี้ประกอบด้วยบันทึกรายละเอียดของธุรกรรมบัตรเครดิตต่าง ๆ รวมถึงข้อมูลเกี่ยวกับเวลาทำรายการ จำนวนเงิน และข้อมูลส่วนบุคคลของผู้ใช้บัตร รวมถึงข้อมูลของร้านค้าที่เกี่ยวข้องกับแต่ละธุรกรรม ประกอบด้วยข้อมูล 24 ตัวแปร และ 1,296,675 แถว (Lee, 2024)

2. ตัวแปรที่ศึกษา

1. Unnamed คือ ตัวระบุลำดับหรือดัชนีของแถวข้อมูล
2. trans_date_trans_time คือ วันที่และเวลาที่ทำธุรกรรม

3. cc_num คือ หมายเลขบัตรเครดิตที่ใช้ในธุรกรรม
4. merchant คือ ชื่อผู้ขายหรือร้านค้าที่ทำธุรกรรม
5. category คือ ประเภทของร้านค้าหรือบริการ
6. amt คือ จำนวนเงินที่ใช้ในธุรกรรม
7. first คือ ชื่อของผู้ถือบัตร
8. last คือ นามสกุลของผู้ถือบัตร
9. gender คือ เพศของผู้ถือบัตร
10. street คือ ชื่อถนนของที่อยู่ผู้ถือบัตร
11. city คือ เมืองของที่อยู่ผู้ถือบัตร
12. state คือ รัฐที่ผู้ถือบัตรอาศัยอยู่
13. zip คือ รหัสไปรษณีย์ของผู้ถือบัตร
14. late คือ ละติจูดของที่อยู่ผู้ถือบัตร
15. long คือ ลองจิจูดของที่อยู่ผู้ถือบัตร
16. city_pop คือ จำนวนประชากรในเมืองที่เกิดธุรกรรม
17. job คือ อาชีพของผู้ถือบัตร
18. dob คือ วันเกิดของผู้ถือบัตร
19. trans_num คือ รหัสประจำธุรกรรม
20. unix_time คือ เวลาของการทำธุรกรรมในรูปแบบธุรกรรม
21. merch_lat คือ ลองจิจูดของสถานที่ทำธุรกรรม
22. merch_long คือ ลองจิจูดของสถานที่ทำธุรกรรม
23. is_fraud คือ ตัวระบุว่าธุรกรรมนั้นเป็นการทุจริตหรือไม่ (Target Class)
24. merch_zipcode คือ รหัสไปรษณีย์ของสถานที่ทำธุรกรรม

4. กรอบแนวคิดในการวิจัย

งานวิจัยนี้มีเป้าหมายในการศึกษาและพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) เพื่อจำแนกประเภทของธุรกรรมบัตรเครดิตออกเป็นสองประเภท ได้แก่ ธุรกรรมปกติและธุรกรรมทุจริต โดยใช้เทคนิคการเรียนรู้แบบมีผู้สอน โดยใช้ชุดข้อมูลธุรกรรมบัตรเครดิตจากแหล่งข้อมูลสาธารณะ Kaggle.com ซึ่งประกอบด้วยข้อมูล 24 ตัวแปร และ 1,296,675 แถว

โดยขั้นตอนในการศึกษาแบบจำลองแบ่งออกเป็นสองขั้นตอนหลัก โดยเริ่มจากการสร้างแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ด้วยอัลกอริทึมต่าง ๆ ได้แก่ Decision

Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Categorical Boosting (CatBoost) และ Extreme Gradient Boosting (XGBoost) และทดลองปรับพารามิเตอร์ของแต่ละแบบจำลองให้เหมาะสมกับชุดข้อมูลและมีประสิทธิภาพมากที่สุด จากนั้นจึงนำอัลกอริทึมที่มีประสิทธิภาพมากที่สุดในขั้นตอนก่อนหน้ามาทดลองกับชุดข้อมูลที่ได้จัดการปัญหาความไม่สมดุลของข้อมูล ซึ่งมักเกิดขึ้นในกรณีของธุรกรรมทุจริตที่มีจำนวนน้อยกว่าธุรกรรมปกติอย่างมาก โดยใช้วิธีการต่าง ๆ ได้แก่ การใช้เทคนิค Random Over-Sampling (ROS), Synthetic Minority Over-sampling Technique (SMOTE), Tomek Links (TL), Edited Nearest Neighbors (ENN), SMOTEENN และ SMOTETomek เพื่อสร้างสมดุลในชุดข้อมูล รวมทั้งการทดลองปรับพารามิเตอร์ของแบบจำลองเพื่อให้ได้ค่าที่เหมาะสมที่สุดที่ทำให้ประสิทธิภาพของแบบจำลองเพิ่มขึ้น

5. ประโยชน์ที่คาดว่าจะได้รับ

1. งานวิจัยนี้จะช่วยเพิ่มประสิทธิภาพในการตรวจจับธุรกรรมทุจริตโดยการประยุกต์ใช้เทคนิคการจัดการข้อมูลที่ไม่สมดุล และการสร้างแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ที่เหมาะสม ทำให้ระบบสามารถระบุธุรกรรมที่มีความผิดปกติได้อย่างแม่นยำและรวดเร็ว ลดโอกาสในการพลาดธุรกรรมที่เป็นการทุจริต

2. ผลการวิจัยสามารถนำไปประยุกต์ใช้ในระบบตรวจจับการทุจริตของสถาบันการเงินเพื่อช่วยปรับปรุงและพัฒนามาตรการป้องกันให้มีประสิทธิภาพมากยิ่งขึ้น โดยสถาบันการเงินสามารถนำแบบจำลองและเทคนิคการจัดการข้อมูลที่ไม่สมดุลจากงานวิจัยนี้ไปปรับใช้กับระบบที่มีอยู่ เพื่อเพิ่มความแม่นยำในการตรวจจับธุรกรรมที่ผิดปกติ และลดข้อผิดพลาดที่อาจเกิดขึ้น เช่น การแจ้งเตือนที่ไม่จำเป็น หรือการพลาดตรวจจับธุรกรรมทุจริต

3. ช่วยลดความสูญเสียทั้งในแง่ของทรัพย์สินของผู้บริโภคและความเสียหายทางการเงินของสถาบันการเงิน การตรวจจับที่แม่นยำช่วยลดเวลาที่ต้องใช้ในการตรวจสอบและตอบสนองต่อธุรกรรมทุจริต ทำให้สามารถป้องกันความเสียหายได้ตั้งแต่ต้น อีกทั้งยังช่วยลดปัญหาการแจ้งเตือนที่ไม่จำเป็น ซึ่งอาจสร้างความไม่สะดวกให้กับลูกค้า ซึ่งจะช่วยเสริมสร้างความน่าเชื่อถือในระบบการเงิน และเพิ่มความมั่นใจให้กับผู้ใช้บริการ

บทที่ 2
ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และได้นำเสนอตามหัวข้อต่อไปนี้

1.ทฤษฎีเกี่ยวกับอัลกอริทึมในการจำแนกประเภท (Classification Algorithms)

2.ทฤษฎีเกี่ยวกับการวัดประสิทธิภาพแบบจำลองการจำแนกประเภท

3.วิศวกรรมคุณลักษณะ (Feature Engineering)

4. เทคนิคการจัดการกับข้อมูลที่ไม่สมดุล

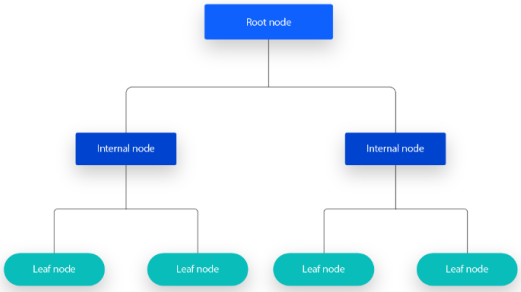
5.งานวิจัยที่เกี่ยวข้อง

1. ทฤษฎีเกี่ยวกับอัลกอริทึมในการจำแนกประเภท (Classification Algorithms)

ทฤษฎีเกี่ยวกับอัลกอริทึมในการจำแนกประเภท (Classification Algorithms) เป็นส่วนสำคัญของการเรียนรู้แบบกำกับดูแล (Supervised Learning) ซึ่งใช้ในการพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ที่สามารถทำนายหรือจำแนกประเภทของข้อมูลได้ อัลกอริทึมที่ใช้ในการสร้างแบบจำลอง ได้แก่

1.1 Decision Tree (DT)

ต้นไม้ตัดสินใจ (Decision Tree) เป็นหนึ่งในอัลกอริทึมการเรียนรู้ของเครื่องที่มีประสิทธิภาพมากที่สุดสำหรับงานจำแนกประเภท โดยมีลักษณะเรียบง่าย ง่ายต่อการสร้าง และเข้าใจได้ง่าย อีกทั้งยังมีความเร็วในการเรียนรู้และจำแนกข้อมูล ซึ่งทำให้สามารถให้ผลลัพธ์ที่แม่นยำพอเหมาะสำหรับงานจำแนกประเภท นอกจากนี้ การสร้างต้นไม้ตัดสินใจยังสามารถเลือกคุณลักษณะที่มีความสามารถในการแยกแยะสูงจากคุณลักษณะที่มีจำนวนมาก และสามารถรับมือกับข้อมูลรบกวน (Noisy Data) ได้อย่างมีประสิทธิภาพ ต้นไม้ตัดสินใจมีโครงสร้างคล้ายแผนผังลำดับงาน โดยโหนดภายใน (Internal Nodes) แทนการทดสอบบนคุณลักษณะหนึ่ง ๆ กิ่งของต้นไม้แทนผลลัพธ์ของการทดสอบ และโหนดใบไม้ (Leaf Nodes) แทนค่าประเภทหรือป้ายกำกับคลาสที่คาดการณ์ไว้ (Taloba. & Ismail., 2019) ดังภาพประกอบที่ 1



ภาพประกอบ 1 แบบจำลองพื้นฐาน Decision Tree

ที่มา: (https://www.ibm.com/think/topics/decision-trees)

การจำแนกประเภทด้วยต้นไม้ตัดสินใจ (Decision Tree) เป็นกระบวนการที่ใช้โครงสร้างข้อมูลลักษณะเป็นต้นไม้ ประกอบด้วยโหนดและเส้นเชื่อมที่จัดเรียงในลักษณะลำดับชั้น โดยเป็นวิธีการแบบโลภ (Greedy Approach) ที่ใช้ในการแก้ปัญหาการจำแนกประเภท โดยการแบ่งปัญหาออกเป็นชุดของการทดสอบย่อย ๆ อย่างต่อเนื่อง ซึ่งจะค่อย ๆ แยกชุดข้อมูลออกเป็นกลุ่มย่อยที่มีขนาดเล็กลงตามระดับความลึกของต้นไม้ ต้นไม้ตัดสินใจจะทำการประมาณคุณสมบัติที่ไม่รู้ของคุณลักษณะ โดยตั้งคำถามเกี่ยวกับคุณสมบัติที่รู้แล้วทีละข้ออย่างต่อเนื่อง กระบวนการจำแนกประเภทของต้นไม้ตัดสินใจแบ่งออกเป็นสองขั้นตอน ได้แก่ การสร้างต้นไม้ (Tree-Induction) และการตัดแต่งต้นไม้ (Tree-Pruning) ในขั้นตอนการสร้างต้นไม้ อัลกอริทึมจะเริ่มต้นด้วยชุดข้อมูลทั้งหมดที่โหนดราก จากนั้นทำการแบ่งข้อมูลตามเกณฑ์การแยกที่กำหนดไว้ให้เป็นกลุ่มย่อย และทำขั้นตอนนี้ซ้ำไปเรื่อย ๆ กับแต่ละกลุ่มย่อย จนกว่าข้อมูลในแต่ละกลุ่มจะเป็นคลาสเดียวกันทั้งหมดหรือมีขนาดเล็กเพียงพอ ส่วนในขั้นตอนการตัดแต่งต้นไม้ จะทำการตัดแต่งต้นไม้ที่เติบโตเต็มที่ให้เล็กลง เพื่อหลีกเลี่ยงปัญหา Overfitting และเพิ่มความแม่นยำของแบบจำลอง

ให้ S_j เป็นชุดข้อมูลฝึกสอนที่มีอยู่ในโหนดภายในหมายเลข j หลังจากที่มีการแบ่งโหนดชุดข้อมูลจะถูกแยกออกเป็นกลุ่มลูกทางซ้าย S_j^l และกลุ่มลูกทางขวา S_j^r ซึ่งจะต้องเป็นไปตามเงื่อนไขดังนี้ (i) $S_j = S_j^l \cup S_j^r$ คือชุดข้อมูลทั้งหมดของโหนด j ต้องเป็นผลรวมของลูกทั้งสองฝั่ง (ii) $s_j^l \cap s_j^r = \emptyset$ คือทั้งสองชุดไม่มีข้อมูลทับซ้อนกัน (iii) $s_j^l = s_{2j} + 1$ และ (iv) $s_j^r = s_{2j} + 1$ หมายถึงการจัดตำแหน่งลูกทางซ้าย และขวาของโหนด j ตามลำดับดัชนีในต้นไม้แบบเลขดัชนีทวิภาคในกระบวนการสร้างต้นไม้ตัดสินใจ การแยกข้อมูลจะดำเนินการโดยใช้เกณฑ์การแบ่ง ต่าง ๆ ในส่วนถัดไปจะกล่าวถึงรายละเอียดสำคัญของเกณฑ์ Gini Index (GI)

ดัชนี Gini (Gini Index) เป็นตัวชี้วัดความไม่บริสุทธิ์ของข้อมูล หรือความไม่เท่าเทียมกันของคลาสที่ปรากฏในชุดข้อมูลที่กำหนด โดยค่าของ Gini Index จะอยู่ในช่วงระหว่าง [0, 1] ซึ่งค่า 0 หมายถึงความบริสุทธิ์สูงสุด และค่า 1 หมายถึงความไม่บริสุทธิ์สูงสุด ให้ D เป็นชุดข้อมูลที่กำลังพิจารณา ดัชนี Gini สำหรับชุดข้อมูล D เมื่อพิจารณาแอตทริบิวต์ x สามารถคำนวณได้ตามสมการที่ 1 ต่อไปนี้ (Vikas Jain, 2018)

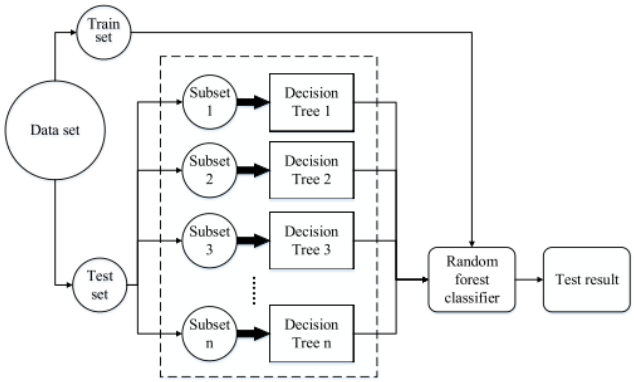
$$GI_x(D) = \sum_{i=1}^{k} p(X = x_j)(1 - \sum_{i=1}^{v} P(Y = y_i)^2) \quad (1)$$

โดยที่ v คือจำนวนคลาสทั้งหมด k คือจำนวนค่าที่เป็นไปได้ของแอตทริบิวต์ x, y คือค่าผลลัพธ์ และ y_i คือค่าของคลาสที่ i วัตถุประสงค์ของ Gini Index คือการเลือกค่าของแอตทริบิวต์ $x = x_j$ สำหรับใช้ในการแบ่งข้อมูล โดยต้องเลือกให้สามารถลดความไม่บริสุทธิ์ของข้อมูลในชุดข้อมูลทั้งหมดให้ได้มากที่สุด ฟังก์ชันสามารถเขียนได้ดังสมการที่ 2

$$GI(x = \chi_i) = \arg\max_{\forall J}\{GI_x(D)\} \quad (2)$$

1.2 Random Forest (RF)

Random Forest model เป็นอัลกอริทึมที่พัฒนาต่อจาก Decision Tree โดยใช้เทคนิคการรวมกลุ่มของต้นไม้การตัดสินใจ (Ensemble of Decision Trees) เพื่อเพิ่มความแม่นยำและความน่าเชื่อถือในผลลัพธ์ อัลกอริทึมนี้สร้างต้นไม้การตัดสินใจหลายต้นโดยใช้กระบวนการสุ่มตัวอย่างข้อมูลย่อย (Sub-sampling) และสุ่มเลือกคุณลักษณะ (Features) สำหรับการสร้างแต่ละต้นไม้ ซึ่งต้นไม้เหล่านี้ทำงานอย่างอิสระจากกัน ผลลัพธ์ที่ได้จากต้นไม้หลายต้นจะถูกรวมเข้าด้วยกันโดยการลงคะแนนเสียง (Voting) เพื่อให้ได้คำตอบสุดท้ายในกรณีของการจำแนกประเภท เทคนิคการสุ่มที่หลากหลายช่วยลดปัญหา Overfitting ซึ่งเป็นข้อจำกัดของ Decision Tree แบบเดี่ยว และเพิ่มความสามารถในการจัดการกับข้อมูลรบกวน (Noisy Data) หรือข้อมูลที่มีความซับซ้อนได้ดี ดังภาพประกอบที่ 2 (Hu et al., 2021)



ภาพประกอบ 2 แบบจำลองพื้นฐาน Random Forest

ที่มา: (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9349646)

อัลกอริทึม Random Forest ใช้วิธีการเลือกคุณลักษณะ (Feature Selection) โดยอิงจากค่าสัมประสิทธิ์ Gini (Gini Coefficient) และใช้วิธีการตัดสินใจที่อิงจากต้นไม้จำแนกและถดถอย (Classification and Regression Trees – CART) ในการสร้างโมเดล โดยค่าความ Gini เป็นตัวแทนของระดับความสับสนของโมเดล ซึ่งยิ่งมีค่าน้อยเท่าใด ก็ยิ่งแสดงว่าโมเดลมีความสับสนน้อยลง หรือสามารถจำแนกข้อมูลได้ชัดเจนมากขึ้น สูตรคำนวณ Gini Coefficient สำหรับการแจกแจงความน่าจะเป็น (Probability Distribution) มีลักษณะดังนี้

$$Gini(p) = 2p(1-p) \tag{3}$$

โดยที่ p คือความน่าจะเป็นของสถานการณ์ที่เป็นไปได้ และ $Gini(p)$ คือค่าสัมประสิทธิ์ Gini สำหรับปัญหาการจำแนกแบบสองกลุ่ม (Dichotomy Problem) หลังจากทำการไล่พิจารณาจุดแบ่ง (Segmentation Points) ทั้งหมดของแต่ละคุณลักษณะแล้ว ข้อมูลตัวอย่างจะถูกแบ่งออกเป็นสองกลุ่ม ตามความสัมพันธ์ระหว่างค่าของคุณลักษณะ (Characteristic Parameters) กับค่าเกณฑ์ที่ใช้แบ่ง (Thresholds)

$$D = \begin{cases} D_1, & C > T_c \\ D_2, & C \leq T_c \end{cases} \tag{4}$$

ค่าสัมประสิทธิ์ Gini ของตัวอย่างภายใต้เงื่อนไขที่ $c > T_c$ ถูกแสดงไว้ในสมการ 4

$$Gini(D, C) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \quad (5)$$

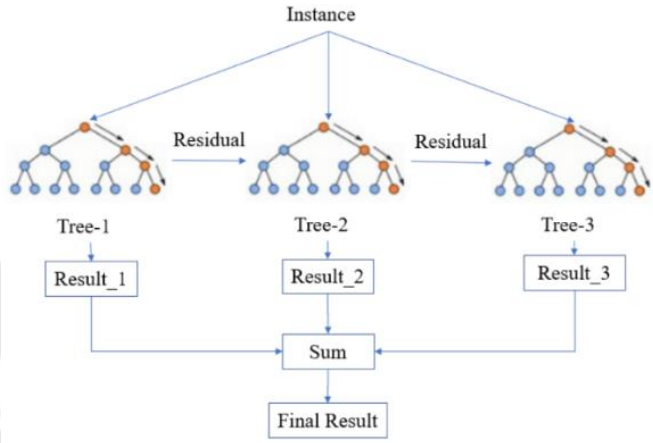
$Gini(D, C)$ หมายถึงความไม่แน่นอนของชุดข้อมูล D หลังจากถูกแบ่งโดยใช้คุณลักษณะ C ส่วน Gini(D1) และ Gini(D2) คือค่าความไม่แน่นอนของกลุ่มข้อมูล D1 และ D2 ที่ได้จากการแบ่งตามลำดับ

การสร้างต้นไม้ตัดสินใจดำเนินการตามหลักของต้นไม้จำแนกและถดถอย โดยเริ่มจากการพิจารณาค่า threshold ที่เป็นไปได้ทั้งหมดของแต่ละคุณลักษณะ แล้วคำนวณค่าความไม่แน่นอนของชุดข้อมูลที่แบ่งได้ จากนั้นเลือกคุณลักษณะที่ให้ค่าความไม่แน่นอนต่ำที่สุด พร้อมค่า threshold ที่เกี่ยวข้องมาเป็นจุดแบ่งของโหนด หากจำนวนตัวอย่างในโหนดหรือความลึกของต้นไม้เป็นไปตามเงื่อนไขที่กำหนด จะถือว่าการสร้างต้นไม้เสร็จสมบูรณ์และส่งคืนต้นไม้ตัดสินใจ แต่หากยังไม่เป็นไปตามเงื่อนไข ระบบจะทำขั้นตอนเดิมซ้ำในโหนดย่อยทั้งสอง

1.3 Extreme Gradient Boosting (XGBoost)

XGBoost เป็นวิธีการเรียนรู้แบบรวม (Ensemble Learning) ที่อิงจากต้นไม้ตัดสินใจ (Decision Tree) โดยอัลกอริทึมของ XGBoost ประกอบด้วยองค์ประกอบหลัก 4 ด้าน ได้แก่ ตัวจำแนกแบบอ่อน (Weak Classifier) ที่อิงจากต้นไม้ตัดสินใจ, ฟังก์ชันการสูญเสีย (Loss Function), กลยุทธ์การเพิ่มประสิทธิภาพ (Promotion Strategy) และการเลือกคุณลักษณะ (Feature Selection) ในด้านของตัวจำแนกแบบอ่อน (Weak Classifier) XGBoost ใช้ตัวจำแนกที่มีพื้นฐานจากต้นไม้ตัดสินใจเป็นโมเดลฐาน (Base Model) โดยปรับปรุงความแม่นยำของโมเดลผ่านการเรียนรู้แบบวนซ้ำอย่างต่อเนื่อง (Iterative Learning) สำหรับฟังก์ชันการสูญเสีย (Loss Function) XGBoost ใช้ฟังก์ชันการสูญเสียแบบมีน้ำหนัก ซึ่งประกอบด้วยสองส่วน ได้แก่ ส่วนแรกคือความแตกต่างระหว่างค่าที่ทำนายได้กับค่าจริงของโมเดล (Residual) และส่วนที่สองคือเทอมการปรับค่าความซับซ้อนของโมเดล (Regularization Term) เพื่อควบคุมไม่ให้โมเดลซับซ้อนเกินไป ซึ่งช่วยลดปัญหา Overfitting ในด้านของกลยุทธ์การเพิ่มประสิทธิภาพ (Promotion Strategy) XGBoost ใช้กลยุทธ์สองแบบ ได้แก่ Gradient Boosting และ Regularized Gradient Boosting โดย Gradient Boosting เป็นกลยุทธ์ที่ใช้การไล่ระดับความชันเชิงลบ (Negative Gradient Descent) เพื่อทำให้ค่าฟังก์ชันการสูญเสียลดลงในการวนซ้ำแต่ละครั้ง ส่วน Regularized Gradient Boosting คือการเพิ่มเทอมการปรับค่าความซับซ้อนเข้าไปในกระบวนการ

Gradient Boosting เพื่อควบคุมโมเดลให้ไม่ซับซ้อนเกินไป ในด้านการเลือกคุณลักษณะ (Feature Selection) XGBoost ใช้วิธีการคำนวณความสำคัญของแต่ละคุณลักษณะ (Feature Importance) เพื่อประเมินว่าคุณลักษณะใดส่งผลต่อผลลัพธ์ของโมเดลมากที่สุด (ShuoWang, 2023) ดังแผนภาพกระบวนการทำงานของอัลกอริทึม XGBoost แสดงไว้ในภาพประกอบที่ 3

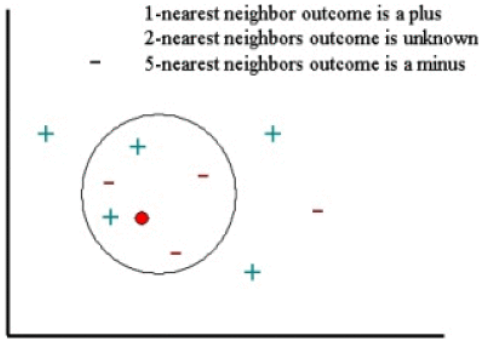


ภาพประกอบ 3 แบบจำลองพื้นฐาน XGBoost

ที่มา: (https://www.researchgate.net/figure/Simplified-structure-of-XGBoost_fig2_348025909)

1.4 K-Nearest Neighbors (KNN)

K-Nearest Neighbor (KNN) เป็นอัลกอริทึมแบบ Supervised Learning โดยจะพิจารณาการจัดประเภทของธุรกรรมใหม่ตามกลุ่มของ k ตัวอย่างที่ใกล้ที่สุด การทำงานของ KNN ขึ้นอยู่กับสามปัจจัยหลัก ได้แก่ วิธีวัดระยะห่าง (เช่น Euclidean สำหรับข้อมูลเชิงตัวเลข หรือ Matching Coefficient สำหรับข้อมูลประเภท) กฎในการตัดสินคลาส และจำนวนเพื่อนบ้าน k ที่เลือกใช้ โดยค่า k ที่มากจะช่วยลดผลกระทบจาก noise ได้ เทคนิคนี้ต้องมีข้อมูลทั้งกลุ่มที่เป็นธุรกรรมปกติ และฉ้อโกงเพื่อนำมาเทรน ดังภาพประกอบที่ 4 (N.Malini & student, 2017)



ภาพประกอบ 4 แบบจำลองพื้นฐาน K-Nearest Neighbor (KNN)

ที่มา: (https://ieeexplore.ieee.org/document/7972424)

จุดมุ่งหมายหลักของเทคนิคนี้คือการจัดประเภท (Classification) ให้กับจุดข้อมูลที่ไม่เคยเห็นมาก่อน โดยพิจารณาคลาสที่พบมากที่สุด (Dominant Class) จากเพื่อนบ้านที่ใกล้ที่สุดจำนวน k ตัวในชุดข้อมูลฝึก โมเดลจะตัดสินผลการจำแนกจากเสียงโหวตของเพื่อนบ้านทั้งหมด โดย k จะต้องเป็นจำนวนเต็มบวกเสมอ โดยเพื่อนบ้านเหล่านั้นจะต้องมาจากชุดข้อมูลที่รู้คลาสที่ถูกต้องอยู่แล้ว โดยเราจะต้องมีวิธีวัดระยะทาง หรือความแตกต่างระหว่างจุดข้อมูล โดยใช้ตัวแปรต้น (Independent Variables) เป็นพื้นฐาน ซึ่งค่าระยะทาง $d(x, y)$ ที่นิยมมากที่สุดคือระยะทางแบบยูคลิด (Euclidean Distance) โดยระยะทางระหว่างสอง $x = (x_1, x_2, \dots, x_n)$ และ $y = (y_1, y_2, \dots, y_n)$ จะถูกคำนวณตามสมการที่ 6 (Roxana Aldea, 2014)

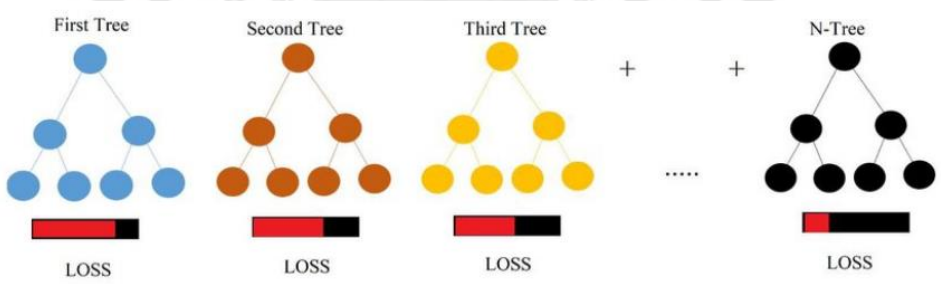
$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (6)$$

ขั้นตอนในการคำนวณ K-Nearest Neighbors มีทั้งหมด 5 ขั้นตอน

1. กำหนดค่า k
2. คำนวณระยะทางระหว่างข้อมูลใหม่กับข้อมูลฝึกทั้งหมด
3. เรียงลำดับระยะทาง และเลือก k จุดข้อมูลที่มีระยะใกล้ที่สุด
4. รวบรวมคลาสของเพื่อนบ้านทั้ง k ตัว
5. ตัดสินคลาสของข้อมูลใหม่โดยใช้วิธีเสียงข้างมาก (Majority Vote)

1.5 Categorical Boosting (CatBoost)

CatBoost เป็นเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ที่ใช้แนวทาง Gradient Boosting บนฐานของต้นไม้ตัดสินใจ (Decision Trees) โดยจะสร้างต้นไม้ตัดสินใจทีละต้นแบบลำดับต่อเนื่อง ซึ่งแต่ละต้นถัดไปจะถูกฝึกให้มีค่าความสูญเสีย (Loss) ที่ลดลงจากต้นก่อนหน้า กระบวนการนี้ช่วยให้แบบจำลองสามารถเรียนรู้จากข้อผิดพลาดของต้นไม้ก่อนหน้าได้อย่างมีประสิทธิภาพ CatBoost จะคำนวณค่าคงเหลือ (Residuals) โดยอิงจากการฝึกแบบจำลองย่อยที่มีจำนวนเท่ากับจำนวนข้อมูล (Data Points) และพิจารณาเฉพาะข้อมูลก่อนหน้าในการคำนวณ ทำให้สามารถทำนายผลได้รวดเร็ว จุดเด่นของ CatBoost คือความสามารถในการจัดการข้อมูลประเภท (Categorical Features) ได้โดยตรงอย่างมีประสิทธิภาพและแม่นยำ (A. Singh et al., 2022) ดังภาพประกอบที่ 5



ภาพประกอบ 5 แบบจำลองพื้นฐาน CatBoost

ที่มา: (https://www.researchgate.net/figure/CatBoost-Regression-model_fig4_373893617)

เมื่อได้รับชุดข้อมูลฝึกซึ่งประกอบด้วยกลุ่มของตัวอย่างข้อมูลและฟีเจอร์ x พร้อมตัวแปรเป้าหมายที่เกี่ยวข้อง CatBoost จะสร้างชุดของต้นไม้ตัดสินใจแบบวนซ้ำ (Iteratively) โดยแต่ละต้นไม้จะพยากรณ์ค่าความชันเชิงลบ (Negative Gradient) ของฟังก์ชันการสูญเสีย (Loss Function) เทียบกับค่าที่โมเดลพยากรณ์ไว้ กระบวนการฝึกของ CatBoost ได้รวมการปรับแต่งต่าง ๆ ที่ช่วยเพิ่มทั้งความแม่นยำและความเร็วของการฝึก ในเชิงคณิตศาสตร์ CatBoost ดำเนินการ Gradient Boosting โดยการสร้างต้นไม้ตัดสินใจชุดหนึ่งแบบวนซ้ำ โดยในการวนซ้ำครั้งที่ t ค่าพยากรณ์ $\hat{y}_i^{(t)}$ สำหรับข้อมูลตัวที่ i คำนวณได้จากสูตร

$$\hat{y}_i^{(t)} = \sum_{k=1}^{k} f_k(x_i) + \sum_{k=1}^{k_t} w_k \cdot I[q(x_i) \epsilon N_k] \quad (7)$$

โดยที่

$\hat{y}_i^{(t)}$ คือ ค่าที่โมเดลพยากรณ์ในการวนซ้ำครั้งที่ t

f_k คือ ต้นไม้การถดถอยต้นที่ k

w_k คือ น้ำหนักของต้นไม้ต้นที่ t

$q(x_i)$ คือ ดัชนีของใบไม้ที่ข้อมูล x_i ตกอยู่

N_k คือ กลุ่มของใบไม้ในต้นไม้ต้นที่ k

$I[\cdot]$ คือ ฟังก์ชันตัวบ่งชี้ (Indicator Function)

2. ทฤษฎีเกี่ยวกับการวัดประสิทธิภาพแบบจำลองการจำแนกประเภท

การวัดประสิทธิภาพของแบบจำลองเป็นขั้นตอนสำคัญในการประเมินผลการทำงานของแบบจำลอง ช่วยให้สามารถวิเคราะห์และเข้าใจประสิทธิภาพของแบบจำลองในการจำแนกประเภทข้อมูลได้อย่างชัดเจน ตัวชี้วัดที่ใช้ในการประเมินแบบจำลองสำหรับงานการจำแนกประเภทมีดังนี้

2.1 เมทริกซ์ความสับสน (Confusion Matrix)

ในการแก้ปัญหาการจำแนกประเภทแบบไบนารี การประเมินประสิทธิภาพของแบบจำลองสามารถกำหนดได้จาก Confusion Matrix ซึ่งเป็นเครื่องมือพื้นฐานที่ใช้ในการแสดงจำนวนตัวอย่างที่ทำนายถูกต้องและผิดพลาด โดยค่าใน Confusion Matrix จะประกอบด้วย True Positive (TP) และ True Negative (TN) ซึ่งแสดงจำนวนตัวอย่างที่แบบจำลองทำนายได้ถูกต้องสำหรับคลาสเป้าหมาย และคลาสไม่ใช่เป้าหมาย ขณะที่ False Positive (FP) และ False Negative (FN) แสดงจำนวนตัวอย่างที่ทำนายผิดพลาด การใช้ข้อมูลเหล่านี้สามารถคำนวณตัวชี้วัดประสิทธิภาพที่หลากหลาย เช่น Accuracy ซึ่งวัดสัดส่วนของการทำนายที่ถูกต้อง Precision ซึ่งวัดความแม่นยำของการระบุคลาสเป้าหมาย Recall ซึ่งวัดความสามารถของแบบจำลองในการระบุคลาสเป้าหมายได้ครบถ้วน และ F1 Score ที่เป็นค่าเฉลี่ยเชิงฮาร์มอนิกของ Precision และ Recall (M & M.N, 2015)

ตาราง 1 Confusion matrix สำหรับการจำแนกประเภทแบบไบนารี

	Actual Positive Class	Actual Negative Class
Predicted Positive Class	True Positive (TP)	False Positive (FP)
Predicted Negative Class	False Negative (FN)	True Negative (TN)

จากตาราง Confusion Matrix แสดงผลตัวแปรของการทำนายของแบบจำลองการจำแนกประเภท โดยมีรายละเอียดดังนี้

True Positive (TP) คือ จำนวนข้อมูลที่เป็น ธุรกรรมทุจริตจริง และถูกทำนายว่าเป็นทุจริต

True Negative (TN) คือ จำนวนข้อมูลที่เป็น ธุรกรรมปกติจริง และถูกทำนายว่าเป็นปกติ

False Positive (FP) คือ จำนวนข้อมูลที่เป็น ธุรกรรมปกติจริง แต่ถูกทำนายว่าเป็นทุจริต

False Negative (FN) คือ จำนวนข้อมูลที่เป็น ธุรกรรมทุจริตจริง แต่ถูกทำนายว่าเป็นปกติ

2.2 ค่าความถูกต้อง (Accuracy)

ค่าความถูกต้อง (Accuracy) เป็นตัวชี้วัดที่ใช้กันอย่างแพร่หลายในการประเมินประสิทธิภาพของแบบจำลองการจำแนกประเภท (Classification Models) โดยแสดงถึง สัดส่วนของจำนวนการทำนายที่ถูกต้องทั้งหมดเมื่อเทียบกับจำนวนตัวอย่างทั้งหมด ตัวชี้วัดนี้ใช้ง่ายและมักเป็นที่นิยม เนื่องจากความสามารถในการวัดที่เข้าใจง่ายและไม่ซับซ้อน อย่างไรก็ตาม ค่าความถูกต้องมีข้อจำกัดสำคัญ เช่น ไม่สามารถประเมินประสิทธิภาพของตัวจำแนกข้อมูลในกรณีที่มีความไม่สมดุลระหว่างกลุ่มข้อมูลได้ดี สูตรในการคำนวณ Accuracy คือ (M & M.N, 2015)

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{FP}+\text{TN}+\text{FN})} \quad (8)$$

2.3 ค่าความเที่ยง (Precision)

ค่าความแม่นยำ (Precision) เป็นตัวชี้วัดที่ใช้เพื่อประเมินความสามารถของแบบจำลองในการคาดการณ์คลาสเป้าหมายได้อย่างถูกต้อง เมื่อเปรียบเทียบกับจำนวนข้อมูลทั้งหมดที่ทำนายว่าอยู่ในคลาสเป้าหมาย โดยคำนวณจากอัตราส่วนระหว่างจำนวนข้อมูลที่อยู่ในคลาสเป้าหมายจริงที่ถูกทำนายอย่างถูกต้อง ตัวชี้วัดนี้มีความสำคัญในกรณีที่ความถูกต้องของการ

ทำนายสำหรับคลาสเป้าหมายเป็นเรื่องสำคัญ เพราะมันสามารถสะท้อนให้เห็นถึงความแม่นยำของแบบจำลองได้โดยตรง (M & M.N, 2015)

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})} \quad (9)$$

2.4 ค่าเรียกคืน (Recall)

ค่าการดึงกลับ (Recall) เป็นตัวชี้วัดที่ใช้ประเมินความสามารถของแบบจำลองในการทำนายข้อมูลคลาสเป้าหมายได้อย่างถูกต้องจากจำนวนข้อมูลคลาสเป้าหมายทั้งหมดในชุดข้อมูล โดยคำนวณจากอัตราส่วนระหว่างจำนวนข้อมูลคลาสเป้าหมายที่ถูกทำนายอย่างถูกต้อง (True Positive) กับจำนวนข้อมูลคลาสเป้าหมายทั้งหมดในชุดข้อมูล (True Positive + False Negative) ตัวชี้วัดนี้มีความสำคัญอย่างยิ่งในสถานการณ์ที่การตรวจจับข้อมูลคลาสเป้าหมายทั้งหมดมีความสำคัญ (M & M.N, 2015)

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (10)$$

2.5 F1 Score

ค่า F1 Score เป็นตัวชี้วัดที่ใช้ประเมินประสิทธิภาพของแบบจำลองในการจำแนกข้อมูล โดยสะท้อนถึงสมดุลระหว่างความแม่นยำในการทำนายข้อมูลคลาสเป้าหมาย (Precision) และความสามารถในการตรวจจับข้อมูลคลาสเป้าหมายทั้งหมด (Recall) ตัวชี้วัดนี้มีความสำคัญในกรณีที่ข้อมูลมีความไม่สมดุลระหว่างคลาส เช่น คลาสหนึ่งมีจำนวนมากกว่าอีกคลาสอย่างมาก F1 Score ช่วยให้สามารถประเมินประสิทธิภาพของแบบจำลองได้ครอบคลุมยิ่งขึ้น เพราะคำนึงถึงทั้งความถูกต้องและการครอบคลุมของการทำนาย (M & M.N, 2015)

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision}+\text{Recall}} \quad (11)$$

3. วิศวกรรมคุณลักษณะ (Feature engineering)

วิศวกรรมคุณลักษณะ (Feature Engineering) เป็นกระบวนการสำคัญในการเตรียมข้อมูล (Data Preparation) ซึ่งมีวัตถุประสงค์หลักเพื่อเพิ่มประสิทธิภาพในการวิเคราะห์ข้อมูลของแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) รวมถึงการสร้างคุณลักษณะใหม่จากคุณลักษณะเดิม โดยอาศัยการประมวลผลหรือคำนวณเพื่อดัดแปลงข้อมูลให้มีความเหมาะสมมากขึ้นต่อการนำไปใช้ โดยมีกระบวนการที่เกี่ยวข้องดังต่อไปนี้

3.1 กระบวนการทำ Standardization

กระบวนการทำ Standardization เป็นขั้นตอนในการปรับข้อมูลให้มีมาตรฐานเดียวกัน โดยการแปลงข้อมูลให้อยู่ในรูปแบบที่มีค่าเฉลี่ยเท่ากับศูนย์ (Mean = 0) และส่วนเบี่ยงเบนมาตรฐานเท่ากับหนึ่ง (Standard Deviation = 1) เพื่อให้ข้อมูลทุกคุณลักษณะมีขนาดหรือสเกลที่เท่าเทียมกัน กระบวนการนี้มีความสำคัญโดยเฉพาะในกรณีที่คุณลักษณะต่าง ๆ ของข้อมูลมีหน่วยหรือช่วงค่าที่แตกต่างกัน ซึ่งอาจส่งผลกระทบต่อการเรียนรู้ของแบบจำลอง ข้อมูลที่ไม่ได้ปรับสเกลอาจทำให้ผลลัพธ์มีความลำเอียงต่อคุณลักษณะที่มีช่วงค่ามากกว่า การทำ Standardization จึงช่วยเพิ่มความแม่นยำและประสิทธิภาพของแบบจำลอง ดังสมการ

$$z = \frac{x - \mu}{\sigma} \tag{12}$$

โดยที่

z = ค่าที่ถูกปรับสเกลแล้ว (Z-score)

x = ค่าจริงในข้อมูล

μ = ค่าเฉลี่ยของข้อมูล

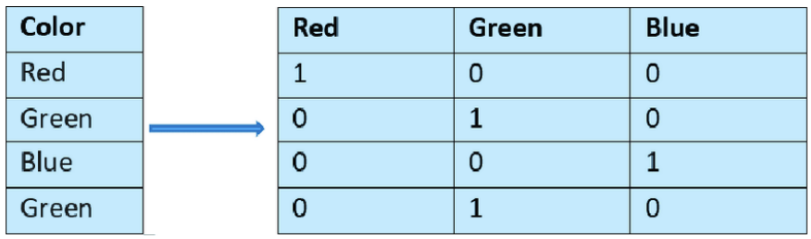
σ = ส่วนเบี่ยงเบนมาตรฐานของข้อมูล

3.2 การแปลงข้อมูลประเภทหมวดหมู่ (Categorical data transformation)

การแปลงข้อมูลประเภทหมวดหมู่ (Categorical Data Transformation) คือกระบวนการปรับเปลี่ยนข้อมูลที่มีลักษณะเป็นหมวดหมู่ หรือข้อมูลที่ไม่เป็นตัวเลข (เช่น ข้อความหรือกลุ่มที่มีการจำแนก) ให้อยู่ในรูปแบบที่เหมาะสมสำหรับการประมวลผลในแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งมักต้องการข้อมูลในรูปแบบตัวเลข (Numerical Data)

3.2.1 One-Hot (One-hot encoding)

เทคนิคการเข้ารหัสแบบ One-Hot (One-Hot Encoding) ซึ่งเป็นวิธีการที่ได้รับความนิยมอย่างมากในการแปลงข้อมูลเชิงหมวดหมู่ให้อยู่ในรูปแบบที่เหมาะสมต่อการใช้งานในงานด้านการเรียนรู้ของเครื่อง (Machine Learning) โดยเฉพาะอย่างยิ่งในกรณีที่แบบจำลองไม่สามารถจัดการข้อมูลในรูปแบบข้อความหรือค่าหมวดหมู่ได้โดยตรง วิธีการนี้จะเปลี่ยนค่าหมวดหมู่ในข้อมูลให้เป็นเวกเตอร์ไบนารีที่มีความยาวเท่ากับจำนวนหมวดหมู่ทั้งหมดในชุดข้อมูล โดยในเวกเตอร์นี้จะมีค่าเป็น 1 เพียงตำแหน่งเดียวซึ่งสอดคล้องกับหมวดหมู่ที่กำลังพิจารณา ขณะที่ตำแหน่งอื่น ๆ ทั้งหมดจะมีค่าเป็น 0 ตัวอย่างเช่น หากข้อมูลมีหมวดหมู่ทั้งหมด d ค่า เวกเตอร์ที่ได้จากการเข้ารหัสแบบ One-Hot จะมีความยาว d โดยในตำแหน่งที่ i ซึ่งแสดงถึงหมวดหมู่ที่ i จะถูกกำหนดค่าเป็น 1 และตำแหน่งอื่นจะถูกกำหนดค่าเป็น 0จ กระบวนการนี้ช่วยลดปัญหาที่อาจเกิดจากการเข้ารหัสด้วยตัวเลข (Label Encoding) ซึ่งอาจทำให้แบบจำลองตีความค่าหมวดหมู่ที่สูงกว่ามีความสำคัญมากกว่า (Johnson & Khoshgoftaar, 2021) ดังภาพประกอบ 6



Color
Red
Green
Blue
Green

Red	Green	Blue
1	0	0
0	1	0
0	0	1
0	1	0

ภาพประกอบ 6 One-Hot Encoding Technique

ที่มา: (https://www.researchgate.net/figure/An-example-of-one-hot-encoding_fig1_344409939)

3.2.2 Label Encoding

Label Encoding เป็นกระบวนการแปลงข้อมูลประเภทหมวดหมู่ (เช่น สี หรือชื่อ) ดังภาพประกอบ 7 ให้เป็นตัวเลขโดยการแมปแต่ละค่าที่ไม่ซ้ำกันให้กับเลขจำนวน เพื่อให้สามารถนำไปใช้ในกระบวนการประมวลผลข้อมูลและการเรียนรู้ของเครื่อง (Machine Learning) ต่อไปได้อย่างมีประสิทธิภาพ แม้ว่าวิธีนี้จะมีความง่ายต่อการใช้งานและช่วย

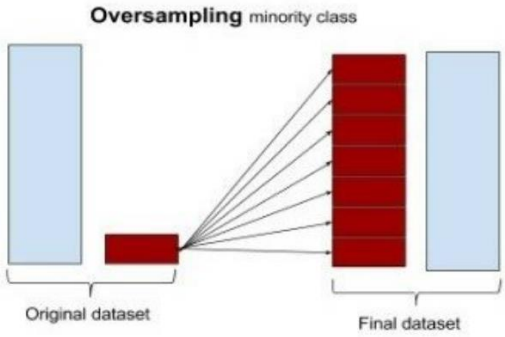
ประหยัดหน่วยความจำ แต่ก็มีข้อจำกัดที่สำคัญคือ แบบจำลองอาจเข้าใจผิดว่ามีความสัมพันธ์เชิงลำดับ (Du et al., 2024)

ภาพประกอบ 7 One-Hot Encoding Technique

4. เทคนิคการจัดการกับข้อมูลที่ไม่สมดุล

4.1 Random Oversampling (ROS)

การเพิ่มตัวอย่างแบบสุ่ม (Random Oversampling - ROS) เป็นวิธีที่ใช้แก้ปัญหาความไม่สมดุลของข้อมูลโดยการเพิ่มขนาดของคลาสที่มีจำนวนน้อย ด้วยการทำซ้ำตัวอย่างในคลาสนั้นแบบสุ่ม วิธีนี้ช่วยเพิ่มความสมดุลระหว่างคลาสส่วนใหญ่และคลาสส่วนน้อย อย่างไรก็ตาม การเพิ่มตัวอย่างในลักษณะนี้ส่งผลให้ขนาดของชุดข้อมูลเพิ่มขึ้น ซึ่งอาจทำให้ต้นทุนด้านการประมวลผลสูงขึ้น และยังมีความเสี่ยงต่อการเกิดปัญหาการจดจำรูปแบบ (Overfitting) ในกรณีที่แบบจำลองเรียนรู้จากข้อมูลที่ซ้ำกันมากเกินไป ทำให้การคาดการณ์ในสถานการณ์ใหม่ขาดความแม่นยำ การเลือกใช้วิธีนี้จึงควรพิจารณาอย่างรอบคอบ โดยเฉพาะในบริบทของชุดข้อมูลขนาดใหญ่ (Bauder et al., 2018) ดังภาพประกอบ 8



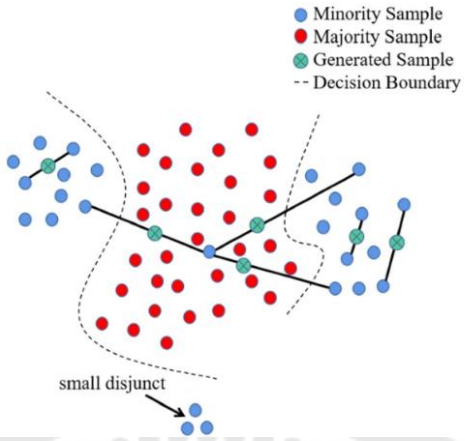
ภาพประกอบ 8 Random Oversampling Technique

ที่มา: (https://www.researchgate.net/figure/Random-Oversampling_fig1_371936273)

4.2 Synthetic Minority Oversampling Technique (SMOTE)

เทคนิค SMOTE เป็นวิธีการแก้ปัญหาข้อมูลไม่สมดุลในงานด้านการเรียนรู้ของเครื่อง (Machine Learning) โดยเฉพาะเมื่อข้อมูลในคลาสหนึ่งมีจำนวนน้อยกว่าคลาสอื่นมาก วิธีการนี้จะสร้างข้อมูลสังเคราะห์ขึ้นมาใหม่เพื่อเพิ่มจำนวนข้อมูลในคลาสที่มีจำนวนน้อย ทำให้แบบจำลองสามารถเรียนรู้ลักษณะของข้อมูลในคลาสนี้ได้ดียิ่งขึ้น

หลักการทำงานของ SMOTE คือการหาข้อมูลใกล้เคียง (K-Nearest Neighbors) ของข้อมูลในคลาสที่มีจำนวนน้อย จากนั้นสร้างข้อมูลใหม่ขึ้นมาโดยการสอดแทรกระหว่างข้อมูลเดิมกับข้อมูลใกล้เคียง โดยมีการเพิ่มความสุ่มเพื่อให้ข้อมูลสังเคราะห์มีความหลากหลายมากขึ้น เทคนิคนี้ช่วยแก้ปัญหาการขาดแคลนข้อมูลในคลาสที่มีจำนวนน้อย และทำให้แบบจำลองมีความสามารถในการทำนายผลลัพธ์ได้แม่นยำยิ่งขึ้น ดังภาพประกอบ 9 (Yi et al., 2022)

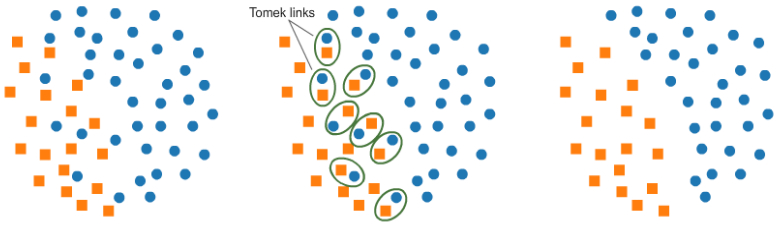


ภาพประกอบ 9 Synthetic Minority Oversampling Technique

ที่มา: (https://link.springer.com/article/10.1007/s40747-021-00638-w)

4.3 Tomek Links

Tomek Links เป็นเทคนิคการทำ Under-Sampling สำหรับจัดการกับปัญหาข้อมูลไม่สมดุล โดยมุ่งเน้นที่การลบตัวอย่างในคลาสที่มีจำนวนมาก (Majority Class) ซึ่งอยู่ใกล้กับตัวอย่างของคลาสที่มีจำนวนน้อย (Minority class) บริเวณจุดแบ่งแยกของข้อมูล (Decision Boundary) ความสัมพันธ์ของ Tomek Link เกิดขึ้นเมื่อมีตัวอย่างข้อมูลสองตัวจากคนละคลาสที่อยู่ใกล้กันมากที่สุดตามระยะทาง Euclidean และทั้งสองตัวอย่างเป็นเพื่อนบ้านที่ใกล้ที่สุดซึ่งกันและกัน การลบตัวอย่างในคลาสที่มีจำนวนมากออกจากคู่ Tomek Link จะช่วยกำจัดจุดข้อมูลที่อาจทำให้เกิดความซ้อนทับระหว่างคลาส และส่งผลให้ขอบเขตการจำแนกของอัลกอริทึมมีความชัดเจนมากขึ้น (Mahsa Bahrami et al., 2023) ดังภาพประกอบ 10

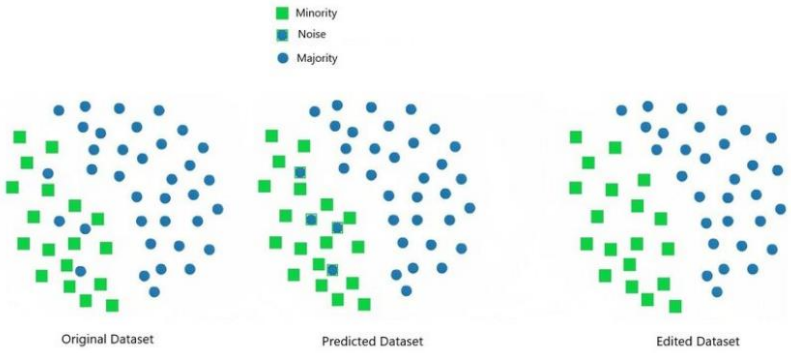


ภาพประกอบ 10 Tomek Links Technique

ที่มา: (https://www.kaggle.com/code/marcinrutecki/smote-and-tomek-links-for-imbalanced-data)

4.4 Edited Nearest Neighbors (ENN)

Edited Nearest Neighbors (ENN) เป็นเทคนิคการทำ Under-Sampling ที่ถูกพัฒนาขึ้นเพื่อลดจำนวนตัวอย่างของคลาสที่มีจำนวนมาก (Majority Class) โดยมุ่งเน้นการลบตัวอย่างที่อยู่ใกล้บริเวณขอบเขตการจำแนกของคลาสต่าง ๆ หากตัวอย่างในคลาสที่มีจำนวนมากมีป้ายกำกับ (Label) ที่แตกต่างจากเพื่อนบ้านที่ใกล้ที่สุด ตัวอย่างนั้นรวมถึงเพื่อนบ้านที่ไม่สอดคล้องกันจะถูกลบออกจากชุดข้อมูล การลบนี้ช่วยขจัดจุดข้อมูลที่มีแนวโน้มก่อให้เกิดความคลาดเคลื่อนในการเรียนรู้ ส่งผลให้แบบจำลองสามารถระบุขอบเขตของแต่ละคลาสได้แม่นยำ (Mahsa Bahrami et al., 2023) ดังภาพประกอบ 11



ภาพประกอบ 11 Edited Nearest Neighbors (ENN) Technique

ที่มา: (https://www.researchgate.net/figure/Edited-Nearest-Neighbors_fig6_371936273)

4.5 SMOTEENN

แนวคิดเบื้องหลังเทคนิค SMOTE + ENN มีความคล้ายคลึงกับการใช้ SMOTE ร่วมกับ Tomek Links โดยมีเป้าหมายเพื่อปรับสมดุลข้อมูลและปรับปรุงคุณภาพของชุดข้อมูลฝึก (Training Set) ให้ชัดเจนมากขึ้น อย่างไรก็ตาม ENN (Edited Nearest Neighbors) มีแนวโน้มที่จะลบตัวอย่างข้อมูลออกมากกว่า Tomek Links จึงให้ผลลัพธ์ในการทำความสะอาดข้อมูล (Data Cleaning) ที่ลึกและครอบคลุมกว่า โดย ENN จะลบตัวอย่างจากทั้งคลาสที่มีจำนวนมากและคลาสที่มีจำนวนน้อย เทคนิคนี้จึงช่วยลด Noise และความคลาดเคลื่อนบริเวณขอบเขตการจำแนกได้ดียิ่งขึ้น (Gustavo E. A. P. A. Batista & Monard, 2004)

4.6 SMOTETomek

แม้การทำ Over-Sampling โดยใช้เทคนิค SMOTE (Synthetic Minority Over-Sampling Technique) จะสามารถช่วยปรับสมดุลระหว่างคลาสได้โดยการสร้างตัวอย่างจำลองในคลาสที่มีจำนวนน้อย แต่ยังคงมีปัญหาหลายประการในชุดข้อมูลที่มีการกระจายคลาสไม่สมดุล เช่น กลุ่มของข้อมูลแต่ละคลาส (Class Clusters) อาจยังไม่ชัดเจนเพียงพอ โดยเฉพาะในกรณีที่ตัวอย่างของคลาสที่มีจำนวนมากแทรกตัวเข้ามาในพื้นที่ของคลาสที่มีจำนวนน้อย อีกทั้ง SMOTE อาจสร้างตัวอย่างจำลองที่ลึกเกินไปในพื้นที่ของคลาสส่วนใหญ่ ส่งผลให้เกิดการขยายขอบเขตของคลาสที่มีจำนวนน้อยมากเกินไป และนำไปสู่การเรียนรู้แบบ Overfitting เพื่อแก้ปัญหาดังกล่าว เทคนิค Tomek Links จึงถูกนำมาใช้ร่วมกับ SMOTE โดยจะลบจุดข้อมูลที่อยู่ในขอบเขตคลาสที่ไม่ชัดเจน เพื่อทำให้กลุ่มของแต่ละคลาสมีความชัดเจนและแยกจากกันมากขึ้น (Gustavo E. A. P. A. Batista & Monard, 2004)

5. งานวิจัยที่เกี่ยวข้อง

5.1 บทความวิจัยเรื่อง Credit card fraud detection method based on KRSMOTE+ENN and XGBoost algorithm (Ma, 2024)

บทความวิจัยนี้ได้นำเสนอบทความเกี่ยวกับการตรวจจับการฉ้อโกงบัตรเครดิตโดยใช้การเรียนรู้ของเครื่อง (Machine Learning) และเน้นไปที่การแก้ไขปัญหาการตรวจจับการฉ้อโกงในชุดข้อมูลที่ไม่สมดุลสูง ซึ่งประกอบด้วยธุรกรรมจำนวน 284,807 รายการ โดยมีธุรกรรมที่เป็นการฉ้อโกงเพียง 492 รายการ คิดเป็นสัดส่วนประมาณ 0.172% ของข้อมูลทั้งหมด

วิธีการที่นำเสนอในบทความนี้คือการสุ่มตัวอย่างแบบไฮบริดโดยใช้เทคนิค KRSMOTE+ENN ร่วมกับอัลกอริทึม XGBoost ซึ่งช่วยจัดการกับข้อมูลที่ไม่สมดุลและเพิ่มความแม่นยำในการตรวจจับธุรกรรมฉ้อโกง นอกจากนี้ การประเมินผลได้ดำเนินการผ่านการทดสอบแบบ 10 fold Cross-Validation โดยพิจารณาค่าความแม่นยำ (Accuracy), ความแม่นยำในการเรียกคืน (Recall) , คะแนน F1, และ AUC (พื้นที่ใต้เส้นโค้ง)

ผลลัพธ์ที่ได้จากการใช้วิธีนี้แสดงให้เห็นว่าการรวมกันของ KRSMOTE+ENN และ XGBoost ให้ผลลัพธ์ที่เกือบสมบูรณ์แบบ โดยมีค่าตัวชี้วัดทั้งหมดประมาณ 99.99% ซึ่งสูงกว่าอัลกอริทึมอื่นๆ เช่น KNN, Decision Tree และ Random Forest

5.2 บทความวิจัยเรื่อง Credit card fraud detection based on ensemble machine learning classifiers (J & Senthilselvi, 2022)

งานวิจัยนี้มุ่งเน้นการเปรียบเทียบวิธีการจัดการข้อมูลที่ไม่สมดุลผ่านเทคนิคการสุ่มตัวอย่างที่หลากหลาย เช่น Random Oversampling, ADASYN และ SMOTE โดยการสุ่มตัวอย่างเหล่านี้มีบทบาทในการเพิ่มจำนวนข้อมูลกลุ่มทุจริตในชุดข้อมูลเพื่อสร้างสมดุลระหว่างธุรกรรมปกติและธุรกรรมทุจริต วิธีการดังกล่าวช่วยให้แบบจำลองสามารถเรียนรู้จากข้อมูลธุรกรรมทุจริตได้ดีขึ้น และลดความลำเอียงในการทำนาย

นอกจากนี้ งานวิจัยนี้ยังเปรียบเทียบประสิทธิภาพของแบบจำลอง Machine Learning ที่ใช้ในการจำแนกประเภท (Classification) โดยใช้เทคนิคแบบจำลองการจำแนกประเภท เช่น XGBoost ซึ่งเป็นหนึ่งในแบบจำลองที่ได้รับความนิยมและมีประสิทธิภาพสูงในงานที่เกี่ยวข้องกับข้อมูลที่ไม่สมดุล การทดสอบนี้มีวัตถุประสงค์เพื่อวัดประสิทธิภาพของแต่ละเทคนิคในการจำแนกธุรกรรมปกติและธุรกรรมทุจริต โดยใช้เกณฑ์วัดผลที่ครอบคลุม เช่น Accuracy, Recall, และ F1 Score เพื่อดูว่าเทคนิคใดให้ผลลัพธ์ที่แม่นยำที่สุดในการตรวจจับธุรกรรมทุจริต

ผลการวิจัยจะช่วยให้เข้าใจว่าเทคนิคการจัดการข้อมูลที่ไม่สมดุลและการใช้แบบจำลองการจำแนกประเภทใดที่มีประสิทธิภาพสูงสุดในการตรวจจับการฉ้อโกงบัตรเครดิต ซึ่งสามารถนำไปปรับปรุงการตรวจจับธุรกรรมทุจริตในสถาบันการเงินและช่วยลดความเสี่ยงด้านความปลอดภัยได้ในระยะยาว

5.3 บทความวิจัยเรื่อง Fraud detection techniques for credit card transactions (Y. Singh et al., 2022)

งานวิจัยนี้มีจุดประสงค์เพื่อพัฒนาวิธีการตรวจจับธุรกรรมฉ้อโกงในบัตรเครดิตโดยใช้เทคนิค Machine learning และการวิเคราะห์ข้อมูล เพื่อเพิ่มความแม่นยำในการจำแนกธุรกรรมที่ผิดปกติ (Outlier) และลดข้อผิดพลาดในการระบุธุรกรรมที่ถูกต้องว่าเป็นการฉ้อโกง งานวิจัยนี้มุ่งเน้นการแก้ปัญหาความไม่สมดุลของข้อมูล โดยเฉพาะในกรณีที่จำนวนธุรกรรมฉ้อโกงมีน้อยกว่าธุรกรรมปกติอย่างมาก ซึ่งเป็นปัจจัยสำคัญที่ส่งผลต่อประสิทธิภาพของแบบจำลองการเรียนรู้

วิธีการศึกษาเริ่มจากการใช้ชุดข้อมูลธุรกรรมบัตรเครดิตจาก Kaggle.com ซึ่งประกอบด้วยข้อมูล 31 คอลัมน์ โดยใช้กระบวนการเตรียมข้อมูล เช่น การตรวจสอบค่าที่ขาดหาย การปรับมาตรฐานข้อมูล และการสร้าง Heatmap เพื่อตรวจสอบความสัมพันธ์ของข้อมูล จากนั้นได้นำอัลกอริทึม Local Outlier Factor (LOF) ซึ่งวิเคราะห์ค่าความผิดปกติจากข้อมูลใกล้เคียง และ Isolation Forest Algorithm ที่ใช้การแบ่งส่วนข้อมูลแบบสุ่มเพื่อตรวจจับค่าผิดปกติ โดยอัลกอริทึมทั้งสองได้รับการปรับปรุงเพื่อรองรับข้อมูลที่มีความไม่สมดุล และช่วยระบุธุรกรรมที่อาจเป็นการฉ้อโกง

ผลการวิจัยพบว่าแบบจำลองที่พัฒนาสามารถบรรลุความแม่นยำสูงถึง 99.6% ในการตรวจจับธุรกรรมฉ้อโกง อย่างไรก็ตาม ความแม่นยำในการจำแนกธุรกรรมฉ้อโกง (Recall) ยังคงต้องพัฒนาเพิ่มเติม เพื่อเพิ่มความครอบคลุมในการตรวจจับ งานวิจัยนี้ยังแนะนำให้ใช้เทคนิค Oversampling เช่น SMOTE เพื่อเพิ่มจำนวนธุรกรรมฉ้อโกงในข้อมูล และเสนอให้เพิ่มชุดข้อมูลและพัฒนาอัลกอริทึมเพิ่มเติมในอนาคต เพื่อรองรับความซับซ้อนของการฉ้อโกงที่เปลี่ยนแปลงอยู่เสมอ ซึ่งจะช่วยเพิ่มประสิทธิภาพในการตรวจจับและลดความเสี่ยงด้านความปลอดภัยในระบบการเงิน

5.4 บทความวิจัยเรื่อง Analysis of credit card fraud transaction detection using machine learning algorithms (Sahu & Sahu, 2023)

งานวิจัยนี้มีจุดประสงค์เพื่อวิเคราะห์และพัฒนาวิธีการตรวจจับธุรกรรมฉ้อโกงในบัตรเครดิต โดยมุ่งเน้นการเปรียบเทียบประสิทธิภาพของอัลกอริทึม Machine Learning หลากหลายรูปแบบ เพื่อหาแนวทางที่เหมาะสมที่สุดในการจำแนกธุรกรรมที่ถูกต้องและธุรกรรมฉ้อโกง ทั้งนี้ความท้าทายของปัญหาเกิดจากความไม่สมดุลของข้อมูลระหว่างธุรกรรมปกติและธุรกรรมฉ้อโกง ซึ่งส่งผลต่อความแม่นยำของการตรวจจับ

โดยงานวิจัยใช้ชุดข้อมูลจาก Kaggle ซึ่งมีข้อมูลธุรกรรมกว่า 284,807 รายการ โดยมีเพียง 492 รายการที่เป็นธุรกรรมฉ้อโกง จากนั้นจึงนำข้อมูลเข้าสู่กระบวนการวิเคราะห์ด้วยอัลกอริทึม Machine Learning เช่น K-Nearest Neighbors(KNN), Support Vector Machine(SVM), Logistic Regression และ Random Forest โดยอัลกอริทึมเหล่านี้ถูกวิเคราะห์และเปรียบเทียบประสิทธิภาพผ่านตัวชี้วัด เช่น Accuracy และ Recall เพื่อประเมินความสามารถในการตรวจจับธุรกรรมฉ้อโกงได้อย่างมีประสิทธิภาพ

ผลการทดลองพบว่าอัลกอริทึม K-Nearest Neighbors(KNN) มีประสิทธิภาพสูงสุดด้วยความแม่นยำ 99.9% รองลงมาคือ Random Forest, Logistic Regression และ SVM ซึ่งทั้งหมดนี้มีความแม่นยำเกิน 99% ในขณะที่ Naïve Bayes มีประสิทธิภาพต่ำสุดที่ 98.6% งานวิจัยนี้เน้นถึงความสำคัญของการเลือกใช้อัลกอริทึมที่เหมาะสมในบริบทของข้อมูลที่ไม่สมดุล และชี้ให้เห็นว่า Machine Learning มีศักยภาพอย่างมากในการพัฒนาระบบตรวจจับธุรกรรมฉ้อโกงที่มีความรวดเร็วและแม่นยำสูง ซึ่งช่วยลดความเสี่ยงด้านการเงินและเพิ่มความเชื่อมั่นในระบบการเงินได้อย่างมีประสิทธิภาพ

5.5 บทความวิจัยเรื่อง Credit card fraud detection using XGBoost for imbalanced data (Purwar, 2023)

งานวิจัยนี้มีจุดประสงค์เพื่อพัฒนาวิธีการตรวจจับการฉ้อโกงในธุรกรรมบัตรเครดิตที่มีปัญหาความไม่สมดุลของข้อมูล โดยใช้ XGBoost เป็นตัวจำแนกและปรับปรุงด้วยเทคนิคการสุ่มตัวอย่าง เช่น ADASYN, SMOTE และ Random Oversampling เพื่อเพิ่มความแม่นยำในการคาดการณ์ว่าธุรกรรมใดเป็นการฉ้อโกง พร้อมทั้งเปรียบเทียบประสิทธิภาพของแต่ละวิธีโดยใช้ตัวชี้วัดที่ครอบคลุม เช่น F1-score, Recall และ Accuracy

ซึ่งงานวิจัยนี้ใช้ชุดข้อมูลจาก Kaggle ซึ่งประกอบด้วยธุรกรรม 284,807 รายการ โดยมีเพียง 492 รายการที่เป็นการฉ้อโกง (Class 1) และที่เหลือเป็นธุรกรรมปกติ (Class 0) ขั้นแรก ทำการเตรียมข้อมูลผ่านกระบวนการจัดการค่าผิดปกติ การสุ่มตัวอย่างข้อมูลเพื่อแก้ปัญหาความไม่

สมดุล จากนั้นใช้ XGBoost เป็นแบบจำลองพื้นฐาน พร้อมปรับปรุงประสิทธิภาพด้วยการสุ่มตัวอย่างข้อมูลเชิงบวกในอัตราที่เหมาะสม แบบจำลองได้รับการประเมินโดยใช้เกณฑ์ Accuracy, Recall และ F1-score

ผลการทดลองพบว่า XGBoost ที่ใช้ ADASYN ให้ค่า Recall สูงสุดที่ 81.5% ในขณะที่ XGBoost ที่ใช้ Random Oversampling มีค่า F1-score สูงสุดที่ 82.78% และค่า Accuracy สูงสุดที่ 99.93% เมื่อเปรียบเทียบกับอัลกอริทึมอื่น ๆ เช่น Logistic Regression, Decision Tree และ Support Vector Machine งานวิจัยนี้แสดงให้เห็นว่า XGBoost ผสานกับการสุ่มตัวอย่างที่เหมาะสมสามารถเพิ่มความแม่นยำและประสิทธิภาพในการตรวจจับการฉ้อโกงได้อย่างมีนัยสำคัญ และเสนอให้พัฒนาแบบจำลองเพิ่มเติมในอนาคตเพื่อปรับปรุงการคาดการณ์และการใช้งานในสถานการณ์จริง

ตาราง 2 สรุปงานวิจัยที่เกี่ยวข้อง

ลำดับ	ชื่อบทความ	วิธีการวิจัย	ผลลัพธ์
1	Credit card fraud detection method based on KRSMOTE+ENN and XGBoost algorithm.	Imbalance data technique: KRSMOTE+ENN Model: XGBoost	Accuracy: 99.5% Recall: 82.3% F1-score: 88.4% AUC: 98.6%
2	Credit card fraud detection based on ensemble machine learning classifiers	Imbalance data technique: SMOTE Model: Ensemble Classifiers	Accuracy: 96% F1-score: 57.95% Precision: 90.5%

ตาราง 2 (ต่อ)

ลำดับ	ชื่อบทความ	วิธีการวิจัย	ผลลัพธ์
3	Fraud detection techniques for credit card transactions	LOF Isolation Forest	Accuracy: 97.2% Precision: 76.8% Recall: 81.4%
4	Analysis of credit card fraud transaction detection using machine learning algorithms	Model: KNN, Decision Tree, SVM, Logistic Regression, Naïve Bayes	KNN: Accuracy: 99.9% Precision: 95.4% Recall: 98.2% Naïve Bayes: Accuracy: 98.6%
5	Credit card fraud detection using XGBoost for imbalanced data	Model: KNN, Decision Tree, SVM, Logistic Regression, Naïve Bayes	KNN: Accuracy: 99.9% Precision: 95.4% Recall: 98.2%

บทที่ 3
วิธีดำเนินการวิจัย

ในการดำเนินการวิจัยครั้งนี้ ผู้วิจัยได้มุ่งเน้นศึกษาการจำแนกประเภทธุรกรรมบัตรเครดิต โดยใช้แบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) และหาอัลกอริทึมที่มีประสิทธิภาพในการจำแนกธุรกรรมทุจริตของรายการธุรกรรมบัตรเครดิต เพื่อช่วยเพิ่มประสิทธิภาพในการตรวจจับธุรกรรมทุจริต โดยศึกษาแบบจำลองการเรียนรู้ของเครื่องที่ประกอบไปด้วย Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Categorical Boosting (CatBoost) และ Extreme Gradient Boosting (XGBoost) นอกจากนี้ ผู้วิจัยยังได้ศึกษาปัญหาข้อมูลที่ไม่สมดุลของข้อมูลที่ใช้ในการพัฒนาแบบจำลองการเรียนรู้ของเครื่อง ซึ่งโดยธรรมชาติแล้วธุรกรรมทุจริตจะมีจำนวนน้อยมากเมื่อเทียบกับธุรกรรมปกติ โดยได้ศึกษาการใช้ Random Over-Sampling (ROS), Synthetic Minority Over-sampling Technique (SMOTE), Tomek Links (TL), Edited Nearest Neighbors (ENN), SMOTEENN และ SMOTETomek เพื่อสร้างสมดุลในชุดข้อมูลในการวิจัยครั้งนี้ ผู้วิจัยได้ดำเนินการขั้นตอนการวิจัย ดังนี้

1.กระบวนการทำงานของแบบจำลองการเรียนรู้ของเครื่อง

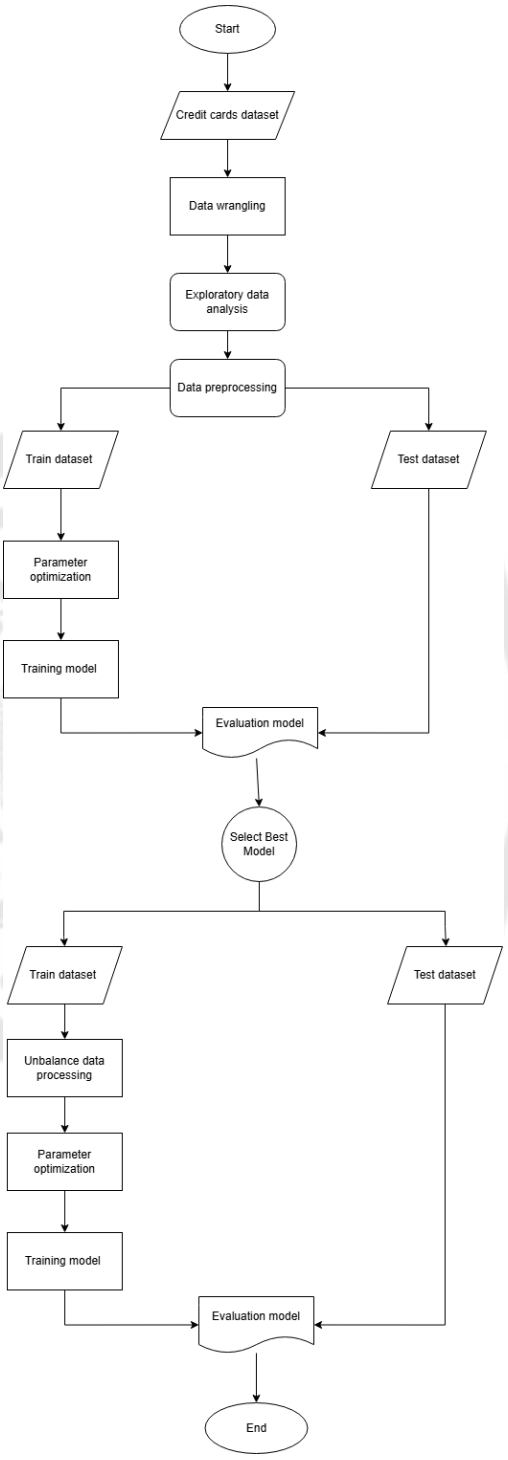
2.การเก็บรวบรวมข้อมูล (Data Collection)

3. การสำรวจข้อมูล (Exploratory Data Analysis: EDA)

4. การเตรียมข้อมูล (Data Preparation)

5. การสร้างแบบจำลองการเรียนรู้ของเครื่อง

1. **กระบวนการทำงานของแบบจำลองการเรียนรู้ของเครื่อง**



ภาพประกอบ 12 กระบวนการทำงานของแบบจำลอง

จากภาพประกอบ 12 แสดงขั้นตอนการสร้างแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) สำหรับจำแนกธุรกรรมจากข้อมูลบัตรเครดิต สามารถอธิบายลำดับกระบวนการได้อย่างเป็นระบบ โดยกระบวนการเริ่มต้นจาก การนำเข้าชุดข้อมูลบัตรเครดิต (Credit Cards Dataset) ซึ่งเป็นข้อมูลดิบที่ยังไม่ได้ผ่านการจัดการ จากนั้นเข้าสู่ขั้นตอน การเตรียมข้อมูล (Data Wrangling) เพื่อทำความสะอาดข้อมูล เช่น การจัดการกับค่าที่ขาดหาย การแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม รวมถึงการจัดโครงสร้างให้พร้อมสำหรับการวิเคราะห์ เมื่อข้อมูลถูกจัดเตรียมเรียบร้อยแล้ว จะเข้าสู่ขั้นตอนของ การวิเคราะห์ข้อมูลเบื้องต้น (Exploratory Data Analysis) เพื่อตรวจสอบลักษณะการกระจายของข้อมูล ความสัมพันธ์ระหว่างตัวแปร และการระบุปัญหา เช่น ความไม่สมดุลของข้อมูล ที่อาจมีผลต่อประสิทธิภาพของแบบจำลอง หลังจากนั้นจะเป็นขั้นตอนของ การเตรียมข้อมูลก่อนการเรียนรู้ (Data Preprocessing) ซึ่งอาจประกอบด้วยการเข้ารหัสข้อมูล และการปรับสเกลข้อมูล (Scaling)

โดยในการทดลองนี้จะแบ่งเป็น 2 ขั้นตอน ขั้นตอนแรกจะแบ่งข้อมูลออกเป็น 2 ชุดคือ ชุดข้อมูลสำหรับการฝึก (Train Dataset) และชุดข้อมูลสำหรับการทดสอบ (Test Dataset) หลังจากนั้น ชุดข้อมูล Train Dataset จะถูกนำไปใช้ในการสร้างแบบจำลองด้วยแบบจำลองต่าง ๆ รวมไปถึงการปรับค่าพารามิเตอร์ (Parameter Optimization) เพื่อให้เหมาะสมกับแบบจำลอง เมื่อทำการฝึกแบบจำลองเรียบร้อยแล้ว จะทำการประเมินประสิทธิภาพของแบบจำลอง (Evaluation Model) ด้วย Test Dataset และเลือกแบบจำลองที่ดีที่สุดจากแบบจำลองที่ได้จากขั้นตอนก่อนหน้า เพื่อนำไปทดลองกับการปรับสมดุลข้อมูล

ในขั้นตอนที่ 2 ชุดข้อมูลฝึกจะถูกนำมาผ่านกระบวนการจัดการปัญหาความไม่สมดุลของข้อมูล เพื่อเพิ่มความสมดุลของข้อมูล จากนั้นทำการฝึกแบบจำลองและปรับพารามิเตอร์อีกครั้ง สุดท้ายประเมินผลแบบจำลองโดยใช้ชุดข้อมูลทดสอบ (Test Dataset) และใช้ตัวชี้วัด Confusion Matrix, Recall, Precision และ F1-Score เพื่อวัดประสิทธิภาพของแบบจำลอง

2. การเก็บรวบรวมข้อมูล (Data Collection)

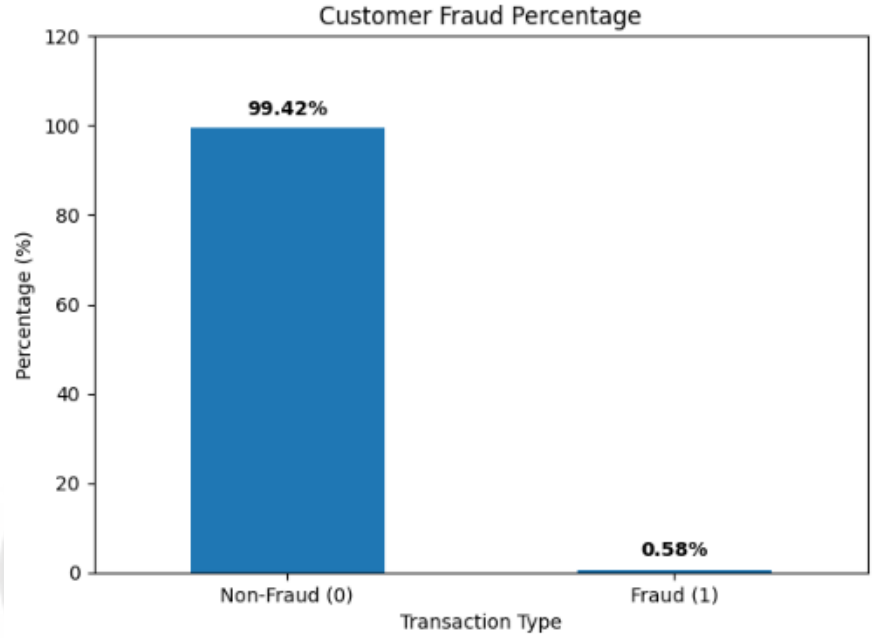
การดำเนินการวิจัยนี้ใช้ชุดข้อมูล Credit Card Transactions Dataset ซึ่งเป็นชุดข้อมูลที่รวบรวมจากเว็บสาธารณะ Kaggle.com (Lee, 2024) โดยมีข้อมูลธุรกรรมบัตรเครดิตทั้งที่เป็นธุรกรรมปกติและธุรกรรมที่อาจเป็นการทุจริต ชุดข้อมูลนี้ประกอบไปด้วยข้อมูลทั้งหมด 1,296,675 แถว และ 24 คอลัมน์ที่ให้รายละเอียดเกี่ยวกับลักษณะต่าง ๆ ของธุรกรรมและข้อมูลพื้นฐานที่เกี่ยวข้องกับผู้ถือบัตรเครดิต ดังตาราง 3

ตาราง 3 ตัวแปรของข้อมูลธุรกรรมบัตรเครดิต

ลำดับ	ชื่อคอลัมน์	คำอธิบาย
1	Unnamed	ตัวระบุลำดับหรือดัชนีของแถวข้อมูล
2	trans_date_trans_time	วันที่และเวลาที่ทำธุรกรรม
3	cc_num	หมายเลขบัตรเครดิตที่ใช้ในธุรกรรม
4	merchant	ชื่อผู้ขายหรือร้านค้าที่ทำธุรกรรม
5	category	ประเภทของร้านค้าหรือบริการ
6	amt	จำนวนเงินที่ใช้ในธุรกรรม
7	first	ชื่อของผู้ถือบัตร
8	last	นามสกุลของผู้ถือบัตร
9	gender	เพศของผู้ถือบัตร
10	street	ชื่อถนนของที่อยู่ผู้ถือบัตร
11	city	เมืองของที่อยู่ผู้ถือบัตร
12	state	รัฐที่ผู้ถือบัตรอาศัยอยู่
13	zip	รหัสไปรษณีย์ของผู้ถือบัตร
14	late	ละติจูดของที่อยู่ผู้ถือบัตร
15	long	ลองจิจูดของที่อยู่ผู้ถือบัตร
16	city_pop	จำนวนประชากรในเมืองที่เกิดธุรกรรม
17	job	อาชีพของผู้ถือบัตร
18	dob	วันเกิดของผู้ถือบัตร
19	trans_num	รหัสประจำธุรกรรม
20	unix_time	เวลาของการทำธุรกรรมในรูปแบบธุรกรรม
21	merch_lat	ลองจิจูดของสถานที่ทำธุรกรรม
22	merch_long	ลองจิจูดของสถานที่ทำธุรกรรม
23	is_fraud	ตัวระบุว่าธุรกรรมนั้นเป็นการทุจริตหรือไม่ (Target Class)
24	merch_zipcode	รหัสไปรษณีย์ของสถานที่ทำธุรกรรม

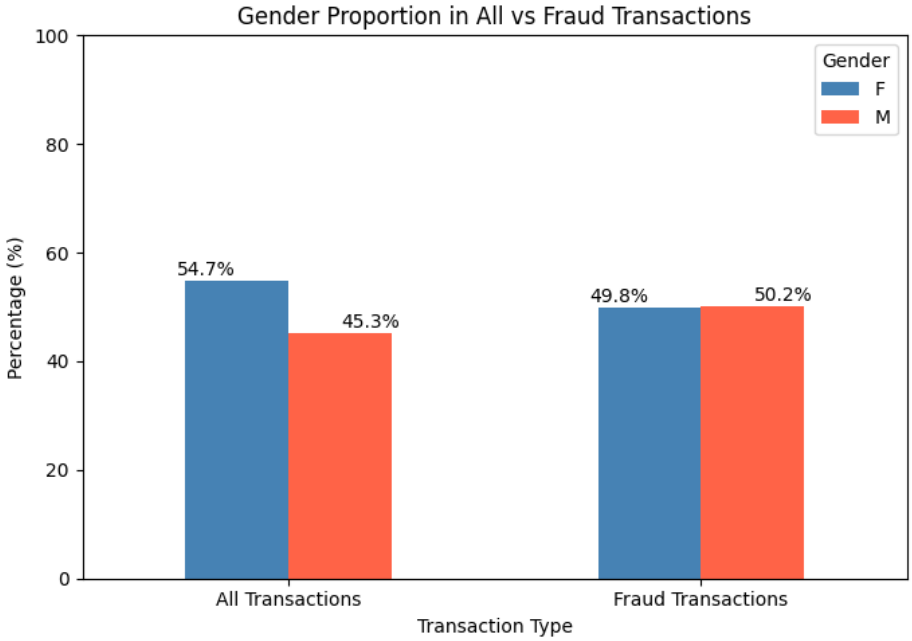
3. **การสำรวจข้อมูล** (Exploratory Data Analysis: EDA)

กระบวนการสำรวจข้อมูลมุ่งเน้นไปที่การวิเคราะห์ความสัมพันธ์ระหว่างตัวแปร ตลอดจนการทำความเข้าใจลักษณะของข้อมูลธุรกรรมบัตรเครดิต โดยกระบวนการนี้ช่วยให้สามารถระบุรูปแบบการใช้งานและพฤติกรรมที่เกี่ยวข้องกับการทำธุรกรรมของลูกค้าได้อย่างชัดเจน



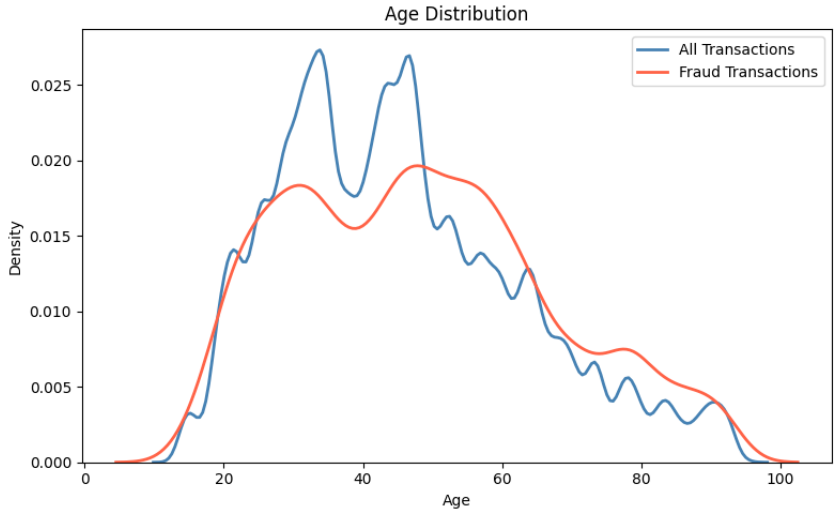
ภาพประกอบ 13 แสดงถึงลักษณะของธุรกรรมบัตรเครดิต

จากการวิเคราะห์ข้อมูลเบื้องต้น (Exploratory Data Analysis) พบว่า ธุรกรรมบัตรเครดิตที่มีลักษณะเป็นธุรกรรมทุจริตมีสัดส่วนน้อยมากเมื่อเทียบกับธุรกรรมปกติ โดยธุรกรรมปกติมีสัดส่วนถึง 99.42% ขณะที่ธุรกรรมทุจริตมีสัดส่วนเพียง 0.58% ซึ่งบ่งชี้ว่าข้อมูลชุดนี้มีความไม่สมดุล (Imbalanced Data) อย่างมาก ดังแสดงในภาพที่ 13



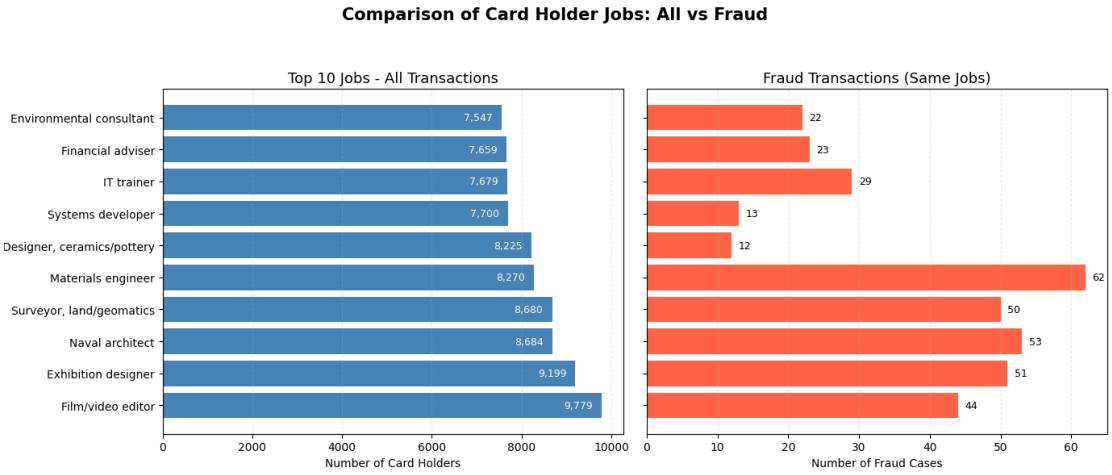
ภาพประกอบ 14 แสดงถึงเพศของผู้ถือบัตรเครดิต

จากภาพประกอบที่ 14 แสดงสัดส่วนเพศของผู้ถือบัตรเครดิตในกลุ่มข้อมูลทั้งหมดและกลุ่มธุรกรรมทุจริต พบว่าเพศหญิง (F) มีสัดส่วน 54.7% ของผู้ถือบัตรทั้งหมด ขณะที่เพศชาย (M) คิดเป็น 45.3% สะท้อนให้เห็นว่าเพศหญิงเป็นกลุ่มผู้ใช้บัตรหลักในภาพรวม อย่างไรก็ตาม เมื่อพิจารณาเฉพาะกลุ่มที่เกี่ยวข้องกับธุรกรรมทุจริต สัดส่วนของเพศหญิงลดลงเหลือ 49.8% ขณะที่เพศชายเพิ่มขึ้นเป็น 50.2% ซึ่งสะท้อนว่าเพศชายมีแนวโน้มที่จะถูกโจรกรรมข้อมูลธุรกรรมสูงกว่าเพศหญิง



ภาพประกอบ 15 แสดงการกระจายตัวของอายุ

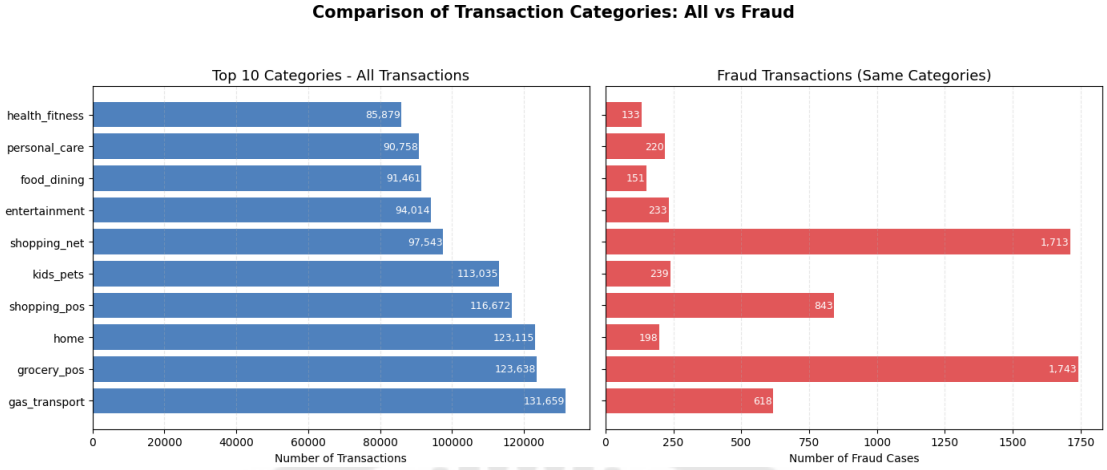
จากภาพประกอบที่ 15 แสดงการกระจายตัวของอายุ (Age Distribution) เปรียบเทียบระหว่างธุรกรรมทั้งหมด และธุรกรรมที่เป็นการทุจริต พบว่าอายุของผู้ทำธุรกรรมส่วนใหญ่อยู่ในช่วง 25–45 ปี โดยเฉพาะช่วงอายุประมาณ 30–40 ปี มีความหนาแน่นสูงสุดในการทำธุรกรรมโดยรวม อย่างไรก็ตาม เมื่อพิจารณาเฉพาะธุรกรรมทุจริต จะพบว่ามีแนวโน้มการกระจายตัวที่กว้างขึ้น และมีความหนาแน่นค่อนข้างสูงในช่วงอายุ 20–60 ปี โดยเฉพาะช่วงอายุ 30–55 ปี ซึ่งบ่งชี้ว่ากลุ่มอายุดังกล่าวมีความเสี่ยงในการเกิดธุรกรรมทุจริตมากกว่าช่วงอายุอื่น ทั้งนี้ ในกลุ่มอายุที่มากกว่า 60 ปีขึ้นไป ความหนาแน่นของธุรกรรมโดยรวมจะลดลงอย่างชัดเจน แต่ยังคงพบการกระจายของธุรกรรมทุจริตอยู่บ้าง



ภาพประกอบ 16 แสดงถึงอาชีพของผู้ถือบัตรเครดิต

จากภาพประกอบที่ 16 แสดงการเปรียบเทียบจำนวนผู้ถือบัตรตามอาชีพ 10 อันดับแรก กับจำนวนธุรกรรมทุจริตในกลุ่มอาชีพเดียวกัน พบว่า แม้อาชีพ Film/video editor, Exhibition designer และ Naval architect จะมีจำนวนผู้ถือบัตรสูง แต่ไม่ได้มีจำนวนธุรกรรมทุจริตมากที่สุด ขณะที่ Materials engineer, Surveyor, land/geomatics และ Naval architect กลับมีจำนวนการทุจริตสูงสุด ทั้งที่ไม่ได้อยู่ในอันดับต้นของผู้ถือบัตรโดยรวม

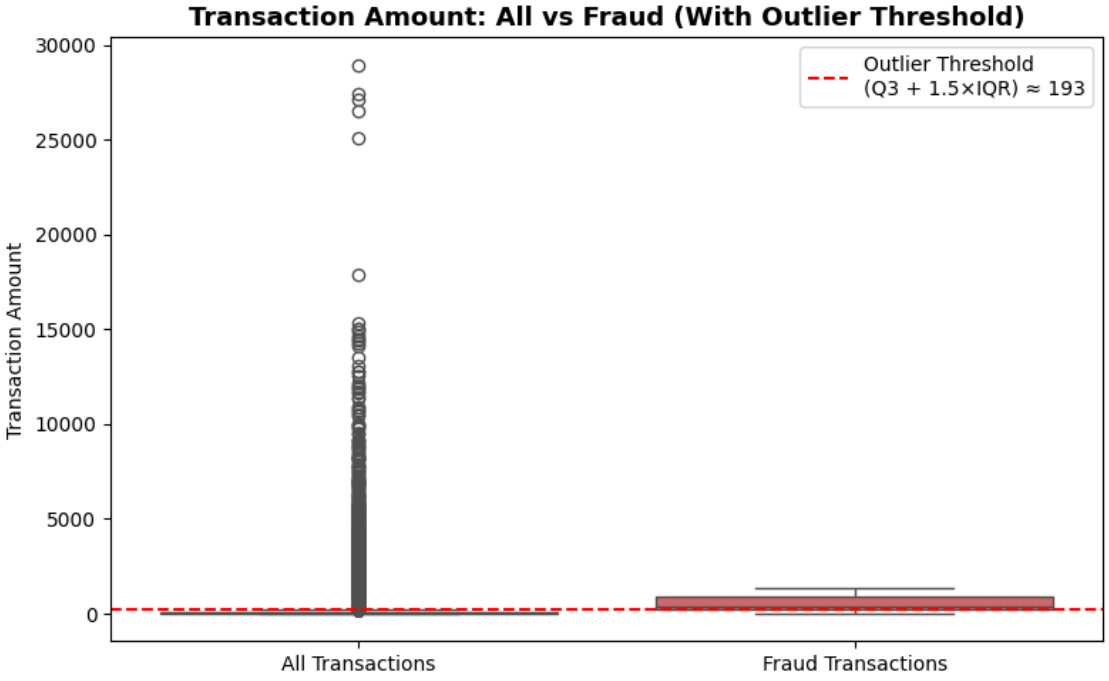
สิ่งนี้สะท้อนให้เห็นว่า ความเสี่ยงของการเกิดธุรกรรมทุจริตไม่ได้ขึ้นอยู่กับจำนวนผู้ถือบัตรเพียงอย่างเดียว แต่อาจเกี่ยวข้องกับลักษณะการใช้งานบัตรหรือพฤติกรรมในแต่ละอาชีพ เช่น อาชีพบางประเภทอาจมีการทำธุรกรรมที่มีความเสี่ยงสูง หรือมีลักษณะเฉพาะที่เอื้อต่อการเกิดทุจริตมากกว่า



ภาพประกอบ 17 แสดงถึงกลุ่มธุรกรรม

จากภาพประกอบที่ 17 แสดงการเปรียบเทียบหมวดหมู่ของธุรกรรมระหว่างธุรกรรมทั้งหมดและธุรกรรมทุจริต พบว่าหมวดหมู่ที่มีจำนวนธุรกรรมโดยรวมสูงสุด ได้แก่ Gas_transport, Grocery_pos และ Home แต่จำนวนธุรกรรมทุจริตกลับพบมากที่สุดในหมวด Grocery_pos และ Shopping_net ซึ่งแต่ละหมวดมีจำนวนทุจริตสูงถึง 1,743 และ 1,713 รายการตามลำดับ

สิ่งนี้สะท้อนว่า หมวด Shopping_net ซึ่งไม่ได้อยู่ในกลุ่มที่มีจำนวนธุรกรรมสูงสุด กลับมีความเสี่ยงในการเกิดการทุจริตสูงมาก อาจเกิดจากลักษณะของการซื้อสินค้าผ่านระบบออนไลน์ที่อาจมีช่องโหว่มากกว่า ขณะที่หมวด Grocery_pos แม้จะมีจำนวนธุรกรรมสูงอยู่แล้ว ก็ยังคงพบการทุจริตในสัดส่วนที่สูงด้วยเช่นกัน ในทางตรงกันข้าม หมวดที่มีจำนวนธุรกรรมสูงอย่าง Gas_transport, Kids_pets หรือ Entertainment กลับมีจำนวนธุรกรรมทุจริตค่อนข้างต่ำ

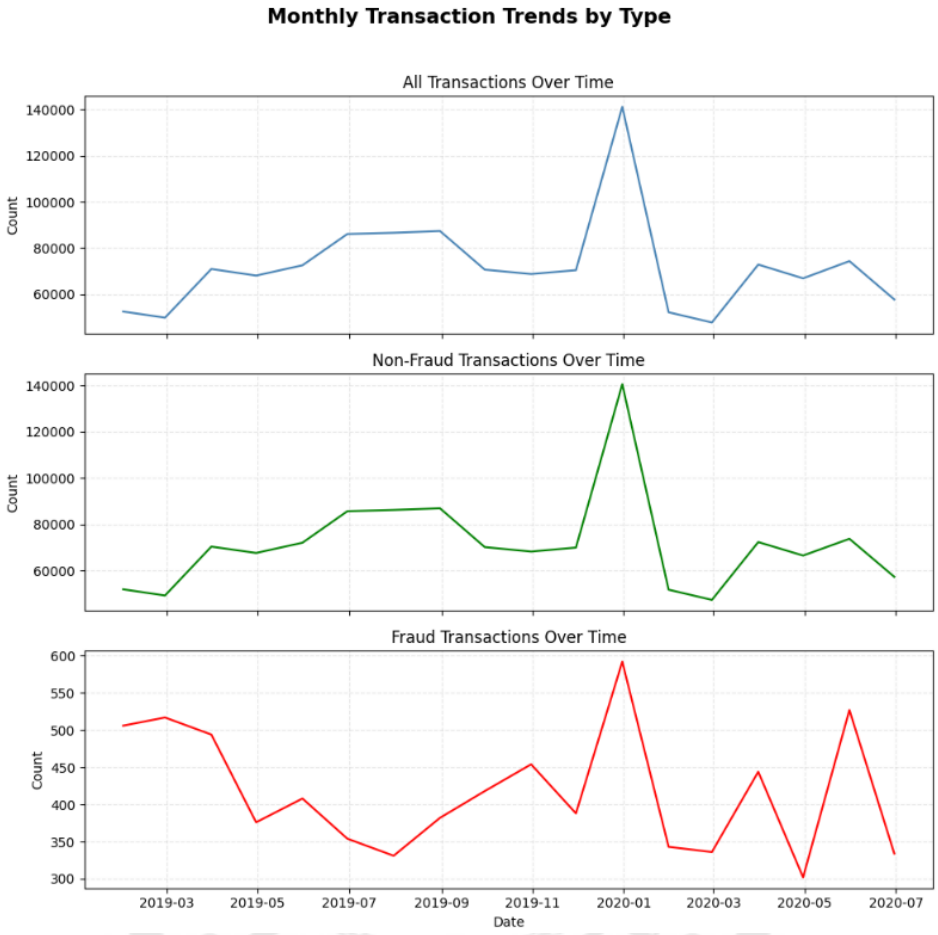


ภาพประกอบ 18 แสดงถึงจำนวนเงินในธุรกรรม

จากภาพประกอบที่ 18 แสดงการเปรียบเทียบจำนวนเงินของธุรกรรมทั้งหมดและธุรกรรมที่เป็นการทุจริต พบว่า ธุรกรรมทั้งหมดมีการกระจายของจำนวนเงินที่กว้าง และมีค่าผิดปกติ (Outliers) จำนวนมาก โดยเฉพาะในช่วงยอดธุรกรรมที่สูงกว่า 10,000 ขึ้นไป ขณะที่ธุรกรรมทุจริตกลับมีช่วงของจำนวนเงินที่แคบกว่า โดยส่วนใหญ่อยู่ในช่วงประมาณ 500–1,500 และมีค่ากลาง (Median) สูงกว่าธุรกรรมทั่วไปอย่างชัดเจน สะท้อนให้เห็นว่าผู้โจรกรรมธุรกรรมทุจริตมักเลือกทำธุรกรรมในช่วงยอดเงินระดับกลาง ซึ่งอาจช่วยหลีกเลี่ยงการถูกตรวจจับโดยระบบ หรือไม่ให้ดูผิดปกติจนเกินไป

แม้ในธุรกรรมปกติจะมียอดสูงมากในบางรายการ แต่ธุรกรรมทุจริตมีลักษณะการกระจุกตัวของจำนวนเงินที่ชัดเจนและสม่ำเสมอกว่า

เมื่อพิจารณาร่วมกับเกณฑ์การตรวจจับ Outlier ตามเส้นประสีแดงในภาพ จะพบว่าธุรกรรมทุจริตเกือบทั้งหมดมีจำนวนเงินที่เกินเกณฑ์นี้ ซึ่งหมายความว่าในเชิงสถิติ ธุรกรรมทุจริตส่วนใหญ่อยู่ในกลุ่มที่ถือเป็นค่าผิดปกติของข้อมูลโดยรวม



ภาพประกอบ 19 แสดงถึงจำนวนธุรกรรมบัตรเครดิตตามช่วงเวลา

จากภาพประกอบที่ 19 แสดงแนวโน้มธุรกรรมรายเดือน พบว่าแนวโน้มของธุรกรรมทั้งหมด และธุรกรรมปกติ (Non-fraud) มีลักษณะใกล้เคียงกัน โดยเฉพาะในช่วงต้นปี 2020 ที่เกิดจำนวนธุรกรรมสูงสุด อย่างไรก็ตาม หลังจากนั้นจำนวนธุรกรรมลดลงอย่างรวดเร็ว ก่อนจะกลับมาอยู่ในระดับคงที่อีกครั้งในช่วงกลางปี ซึ่งอาจสะท้อนถึงปัจจัยเชิงฤดูกาล (Seasonality) หรือเหตุการณ์เฉพาะช่วงเวลา เช่น โปรโมชั่นส่งท้ายปี

ในส่วนของธุรกรรมทุจริต แม้จำนวนจะมีความผันผวนไม่สม่ำเสมอ แต่ก็พบแนวโน้มจุดสูงสุดในช่วงเวลาเดียวกัน คือประมาณต้นปี 2020 เช่นเดียวกับธุรกรรมประเภทอื่น โดยจำนวนธุรกรรมทุจริตยังคงมีการสลับขึ้นลงเป็นระยะตลอดช่วงเวลา และไม่ได้ลดลงอย่างชัดเจนหลังจากจุดสูงสุด

4. การเตรียมข้อมูล (Data preparation)

ในขั้นตอนนี้เป็นขั้นตอนการเตรียมข้อมูลก่อนที่จะสร้างแบบจำลอง โดยผู้วิจัยได้แบ่งข้อมูลออกเป็น 2 ชุด คือ ชุดข้อมูลสำหรับการเรียนรู้ และชุดข้อมูลสำหรับทดสอบ โดยที่สัดส่วนชุดข้อมูลสำหรับการเรียนรู้เป็น 70% ได้ข้อมูลทั้งหมด 907,672 แถว และชุดข้อมูลสำหรับทดสอบเป็น 30% ได้ข้อมูลทั้งหมด 389,003 แถว

4.1 การทำความสะอาดข้อมูล

ในขั้นตอนการสำรวจข้อมูลเบื้องต้น (Exploratory Data Analysis) ผู้วิจัยได้ตรวจสอบตัวแปรต่าง ๆ ภายในชุดข้อมูล พบว่า ตัวแปร merch_zipcode มีค่าที่หายไป (Missing Values) คิดเป็นสัดส่วนประมาณ 15% ของข้อมูลทั้งหมด ซึ่งอาจส่งผลต่อความแม่นยำของแบบจำลอง อย่างไรก็ตาม ในชุดข้อมูลยังมีตัวแปรอื่นที่สามารถใช้แทนตัวแปร merch_zipcode ได้อย่างมีประสิทธิภาพ คือ merch_lat และ merch_long ซึ่งแสดงพิกัดละติจูดและลองจิจูดของร้านค้าโดยตรง ทำให้สามารถระบุตำแหน่งของร้านค้าได้แม่นยำกว่าการใช้รหัสไปรษณีย์เพียงอย่างเดียว

นอกจากนี้ ผู้วิจัยยังได้ดำเนินการตรวจสอบความซ้ำซ้อนของตัวแปร และพบว่ามีหลายตัวแปรที่มีลักษณะข้อมูลซ้ำซ้อนกัน หรือไม่มีความจำเป็นต่อการสร้างแบบจำลอง จึงได้ทำการคัดกรองและลบตัวแปรที่ไม่เกี่ยวข้องหรือไม่จำเป็นในการวิเคราะห์ออกจากชุดข้อมูล เพื่อให้แบบจำลองมีประสิทธิภาพมากยิ่งขึ้น และลดความซับซ้อนของข้อมูล โดยตัวแปรที่ถูกลบออกได้แก่ Unnamed 0, Cc_num, First, Last, Street, City, State, Zip, Trans_num, Unix_time, Txn_dt, Txn_time และ Merch_zipcode

4.2 การเปลี่ยนข้อมูลแบบหมวดหมู่และตัวเลข

การเปลี่ยนข้อมูลแบบหมวดหมู่ (Categorical data) และข้อมูลตัวเลข (Numerical data) ให้เหมาะสมกับการนำไปใช้งานในแบบจำลองทางการเรียนรู้ของเครื่อง (Machine learning) เป็นหนึ่งในขั้นตอนที่สำคัญในการเพิ่มประสิทธิภาพของแบบจำลอง เพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำมากขึ้น

การแปลงข้อมูลหมวดหมู่ เพื่อให้แบบจำลองซึ่งไม่สามารถนำข้อมูลเหล่านี้มาใช้งานได้โดยตรง จึงต้องมีการแปลงให้อยู่ในรูปแบบที่แบบจำลองสามารถประมวลผลได้ โดยพิจารณาเปลี่ยนข้อมูลที่เป็นหมวดหมู่ ได้แก่ Merchant, Category, Gender, Job ด้วยเทคนิค One-hot encoding เพื่อใช้กับแบบจำลอง K-Nearest Neighbors (KNN) และเทคนิค Label encoding เพื่อใช้กับแบบจำลอง Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost), K-Nearest Neighbors (KNN), และ Categorical Boosting (CatBoost)

หลังจากนั้นจึงพิจารณาข้อมูลตัวเลขอื่น ๆ ได้แก่ Amt, Lat, Long, City_pop, Merch_lat, Merch_long, Trans_year, Trans_month, Trans_day, Trans_weekday, Trans_hour, Trans_minute, Trans_second, Age, และ Distance พบว่าข้อมูลมีหน่วยวัดที่แตกต่างกันมาก จึงได้ทำ Standardization หรือ Z-score Scaling ซึ่งเป็นเทคนิคการปรับขนาดข้อมูล (Feature Scaling) เพื่อทำให้ข้อมูลมีการกระจายตัวที่เหมาะสมสำหรับการวิเคราะห์ โดยการทำ Standardization จะช่วยให้ข้อมูลอยู่ในช่วงที่สมดุล และลดปัญหาที่ข้อมูลบางตัวมีค่าใหญ่เกินไป

4.3 การจัดการกับข้อมูลที่ไม่สมดุล

จากการสำรวจข้อมูล พบว่าจำนวนตัวอย่างธุรกรรมที่ระบุว่าเป็นการทุจริตและไม่ทุจริตมีความไม่สมดุลอย่างชัดเจน ซึ่งส่งผลต่อประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ที่พัฒนาขึ้น โดยแบบจำลองอาจมีแนวโน้มให้ความสำคัญกับกลุ่มข้อมูลที่มีจำนวนมากกว่า ทำให้ไม่สามารถตรวจจับธุรกรรมที่เป็นการทุจริตได้อย่างแม่นยำ ซึ่งอาจนำไปสู่ข้อผิดพลาดที่สำคัญ เช่น การระบุธุรกรรมทุจริตว่าเป็นธุรกรรมปกติ

ดังนั้นจึงต้องแก้ไขปัญหาความไม่สมดุลของข้อมูล โดยการทำให้ข้อมูลทั้งสองกลุ่มมีจำนวนใกล้เคียงกัน โดยเลือกใช้เทคนิคการแก้ไขข้อมูลไม่สมดุล ได้แก่ เทคนิค Random Over-Sampling (ROS), Synthetic Minority Over-Sampling Technique (SMOTE), Tomek Links (TL), Edited Nearest Neighbors (ENN), SMOTEENN และ SMOTETomek เพื่อเปรียบเทียบประสิทธิภาพ

4.3.1 การสุ่มเพิ่มข้อมูลแบบสุ่ม (Random Oversampling)

เทคนิค Random Oversampling เป็นวิธีการเพิ่มจำนวนตัวอย่างในกลุ่มที่มีจำนวนน้อย (Minority Class) โดยการทำซ้ำตัวอย่างที่มีอยู่แล้ว เพื่อให้มีความสมดุลมากขึ้น วิธีนี้ช่วยให้แบบจำลองสามารถเรียนรู้ลักษณะของกลุ่มที่มีจำนวนน้อย (Minority Class) ได้ดีขึ้น

ในกรณีนี้ ชุดข้อมูลถูกแบ่งสำหรับการฝึกแบบจำลองในสัดส่วนร้อยละ 70 จากข้อมูลทั้งหมดจำนวน 907,672 แถว โดยก่อนการใช้เทคนิค Random Oversampling ข้อมูลธุรกรรมปกติ (Class 0) มีจำนวนเท่ากับ 902,451 แถว และข้อมูลธุรกรรมทุจริต (Class 1) เท่ากับ 5,221 แถว

หลังจากใช้เทคนิค Random Oversampling แล้ว ข้อมูลธุรกรรมทุจริต (Class 1) เพิ่มขึ้นเท่ากับ 902,451 แถว เท่ากับกลุ่มปกติ ส่งผลให้ได้ชุดข้อมูลที่มีความสมดุลทั้งสองคลาส

ตาราง 4 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค Random Oversampling

Class	Before	After
0	902,451	902,451
1	5,221	902,451

4.3.2 Synthetic Minority Oversampling Technique (SMOTE)

เทคนิค Smote (Synthetic Minority Oversampling Technique) เป็นวิธีการสร้างตัวอย่างใหม่ในกลุ่มที่มีจำนวนน้อย (Minority Class) โดยการสร้างข้อมูลสังเคราะห์ขึ้นมาจากข้อมูลเดิม เทคนิคนี้เริ่มต้นด้วยการสุ่มเลือกตัวอย่างหนึ่งจากกลุ่ม Minority แล้วทำการค้นหาเพื่อนบ้านที่ใกล้ที่สุดโดยใช้วิธี K-nearest Neighbors (KNN) ซึ่งในกรณีนี้กำหนดค่า k เท่ากับ 5 จากนั้นสุ่มเลือกหนึ่งในเพื่อนบ้านเหล่านั้นและใช้จุดข้อมูลทั้งสองในการสร้างตัวอย่างใหม่ ซึ่งจะอยู่ในระหว่างจุดข้อมูลต้นฉบับและจุดของเพื่อนบ้าน โดยตัวอย่างที่ได้จะมีลักษณะใกล้เคียงกับข้อมูลในกลุ่ม Minority เดิม ช่วยให้แบบจำลองสามารถเรียนรู้โครงสร้างข้อมูลของกลุ่มที่เกิดน้อยได้ดียิ่งขึ้น

สำหรับชุดข้อมูลนี้ ก่อนการปรับสมดุล ข้อมูลธุรกรรมปกติ (Class 0) มีจำนวน 902,451 แถว และข้อมูลธุรกรรมทุจริต (Class 1) มีจำนวนเพียง 5,221 แถว หลังจากใช้เทคนิค SMOTE แล้ว จำนวนของกลุ่มข้อมูลธุรกรรมทุจริต (Class 1) เพิ่มขึ้นเท่ากับกลุ่มปกติ ส่งผลให้ชุดข้อมูลธุรกรรมทุจริต (Class 1) หลังปรับสมดุลมีจำนวน 902,451 แถว

ตาราง 5 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค SMOTE

Class	Before	After
0	902,451	902,451
1	5,221	902,451

4.3.3 Tomek Links

เทคนิค Tomek Links เป็นวิธีการปรับสมดุลข้อมูลที่เน้นการลบตัวอย่างที่อยู่ใกล้ขอบเขตระหว่างคลาส โดยพิจารณาคู่ของตัวอย่างสองจุดที่อยู่คนละคลาสและอยู่ใกล้กันมากที่สุด หากพบว่าคู่นั้นเป็นเพื่อนบ้านที่ใกล้ที่สุดของกันและกัน ตัวอย่างที่อยู่ในคลาสที่

มีจำนวนมากกว่า (Majority Class) จะถูกลบออก จุดประสงค์หลักคือเพื่อลดการทับซ้อนระหว่างคลาส (Class Overlap) และทำให้เส้นแบ่งเขตข้อมูล (Decision Boundary) ชัดเจนยิ่งขึ้น

จากตัวอย่างข้อมูลมีจำนวนทั้งหมด 907,672 แถว โดยประกอบด้วยข้อมูลธุรกรรมปกติ (Class 0) จำนวน 902,451 ตัวอย่าง และข้อมูลธุรกรรมทุจริต (Class 1) จำนวน 5,221 แถว หลังจากใช้เทคนิค Tomek Links (TI) ทำการลบตัวอย่างในคลาสที่ทับซ้อนกันออก จำนวนข้อมูลในกลุ่มปกติลดลงเหลือ 901,737 แถว ขณะที่จำนวนของกลุ่มธุรกรรมทุจริตยังคงเท่าเดิม

ตาราง 6 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค Tomek Links

Class	Before	After
0	902,451	901,737
1	5,221	5,221

4.3.4 Edited Nearest Neighbors (ENN)

เทคนิค ENN (Edited Nearest Neighbors) เป็นวิธีการลบข้อมูลแบบเฉพาะจุด โดยพิจารณาจากความไม่สอดคล้องกันระหว่างตัวอย่างและเพื่อนบ้านใกล้เคียง หากตัวอย่างใดมี Label ที่แตกต่างจากเสียงข้างมากของเพื่อนบ้านจำนวน 3 ตัวอย่างแรก (K = 3) จะถือว่าเป็นข้อมูลที่มีแนวโน้มผิดปกติ หรือเป็น Noise และจะถูกลบออกจากชุดข้อมูล เทคนิคนี้ช่วยลดความไม่แน่นอนของข้อมูล (Noise Reduction) และช่วยให้เส้นแบ่งระหว่างคลาส (Decision Boundary) มีความชัดเจนยิ่งขึ้น

สำหรับชุดข้อมูลนี้ ก่อนใช้เทคนิค ENN มีจำนวนข้อมูลธุรกรรมปกติ (Class 0) เท่ากับ 902,451 แถว และข้อมูลธุรกรรมทุจริต (Class 1) เท่ากับ 5,221 แถว หลังจากลบข้อมูลที่ไม่สอดคล้องกันด้วยเทคนิค ENN แล้ว จำนวนข้อมูลธุรกรรมปกติ (Class 0) ลดลงเหลือ 897,621 แถว ขณะที่ข้อมูลธุรกรรมทุจริต (Class 1) ยังคงอยู่ที่ 5,221 แถว

ตาราง 7 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลที่ไม่สมดุลด้วย ENN

Class	Before	After
0	902,451	897,621
1	5,221	5,221

4.3.5 SMOTEENN

เทคนิค SMOTEENN เป็นการผสมผสานระหว่างการเพิ่มข้อมูล (SMOTE) และการลบข้อมูลที่ไม่สอดคล้อง (ENN) โดยเริ่มจากการใช้ SMOTE สร้างตัวอย่างใหม่ในกลุ่มที่มีจำนวนข้อมูลน้อย (Minority Class) เพื่อเพิ่มความสมดุลของข้อมูล จากนั้นจึงนำข้อมูลที่ได้ไปใช้กับเทคนิค ENN เพื่อลบจุดข้อมูลที่มีความขัดแย้งกับเพื่อนบ้านใกล้เคียง เทคนิคนี้มีจุดเด่นคือสามารถเพิ่มความสมดุลของข้อมูล พร้อมทั้งลดความซ้อนทับของคลาส (Class Overlap) และทำให้ขอบเขตของคลาส (Decision Boundary) ชัดเจนยิ่งขึ้น

สำหรับชุดข้อมูลนี้ ก่อนใช้เทคนิค SMOTEENN มีจำนวนข้อมูลธุรกรรมปกติ (Class 0) เท่ากับ 902,451 แถว และข้อมูลธุรกรรมทุจริต (Class 1) เท่ากับ 5,221 แถว หลังจากใช้เทคนิค SMOTEENN แล้ว จำนวนข้อมูลธุรกรรมทุจริต (Class 1) เพิ่มขึ้นเป็น 902,451 แถว ขณะที่จำนวนข้อมูลธุรกรรมปกติ (Class 0) ถูกลบออกบางส่วน เหลือเพียง 884,615 แถว

ตาราง 8 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลด้วยเทคนิค SMOTEENN

Class	Before	After
0	902,451	884,615
1	5,221	902,451

4.3.6 SMOTETomek

เทคนิค SMOTETomek เป็นการผสมผสานระหว่างสองเทคนิค คือ SMOTE (Synthetic Minority Oversampling Technique) และ Tomek Links โดยเริ่มจากการใช้ SMOTE เพื่อสร้างตัวอย่างใหม่ในกลุ่มที่มีจำนวนน้อย (Minority Class) ด้วยการสร้าง

ข้อมูลสังเคราะห์ที่อยู่ระหว่างตัวอย่างจริงและเพื่อนบ้านที่ใกล้เคียง จากนั้นจึงใช้ Tomek Links ลบจุดข้อมูลที่อยู่ใกล้ขอบเขตระหว่างคลาส โดยจะลบตัวอย่างที่อยู่ในกลุ่มที่มีจำนวนมากกว่า (Majority Class) ในคู่ของ Tomek Link ออก ทำให้เส้นแบ่งระหว่างคลาส (Decision Boundary) มีความชัดเจนมากขึ้น

ข้อมูลก่อนการใช้เทคนิค SMOTETomek มีข้อมูลธุรกรรมปกติ (Class 0) เท่ากับ 902,451 แถว และข้อมูลธุรกรรมทุจริต (Class 1) เพียง 5,221 แถว หลังจากใช้เทคนิค SMOTETomek แล้ว ข้อมูลในทั้งสองคลาสมีจำนวนเท่ากัน คือ 902,451 แถว

ตาราง 9 แสดงจำนวนข้อมูลก่อน และหลังปรับข้อมูลที่ไม่สมดุลด้วย SMOTETomek

Class	Before	After
0	902,451	902,451
1	5,221	902,451

5. การสร้างแบบจำลอง

5.1 การสร้างแบบจำลองด้วยชุดข้อมูลไม่สมดุล

เมื่อเตรียมข้อมูลเรียบร้อยแล้ว ขั้นตอนนี้จะเป็นการสร้างแบบจำลองโดยใช้ข้อมูลที่ไม่สมดุล โดยใช้ข้อมูลฝึกทั้งหมด 907,672 แถว แบ่งเป็นข้อมูลธุรกรรมปกติ (Class 0) จำนวน 902,451 แถว และข้อมูลธุรกรรมทุจริต (Class 1) จำนวน 5,221 แถว ส่วนข้อมูลสำหรับทดสอบมีจำนวน 389,003 แถว

ผู้วิจัยได้ใช้ไลบรารี Scikit-learn ในการดำเนินการ และมีการกำหนด Random state = 42 เพื่อให้สามารถทำซ้ำผลลัพธ์ได้อย่างสม่ำเสมอ

สำหรับการประเมินประสิทธิภาพของแบบจำลอง ใช้วิธีการแบ่งข้อมูลแบบ Stratified K-Fold Cross-validation จำนวน 5 ชุด และกำหนดค่าพารามิเตอร์ด้วยตนเอง (Manual Setting) เพื่อควบคุมความซับซ้อนของโมเดล และลดความเสี่ยงจากการ Overfitting

5.1.1 แบบจำลอง Decision Tree (DT)

ในการสร้างแบบจำลอง Decision Tree (DT) ผู้วิจัยได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ให้เหมาะสม ดังนี้

1. max_depth คือ ความลึกสูงสุดของต้นไม้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 5, 10 และ 100

2. min_samples_split คือ จำนวนตัวอย่างขั้นต่ำเพื่อแบ่งโหนด ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 2, 5, 10 และ 100

3. min_samples_leaf คือ จำนวนตัวอย่างขั้นต่ำในแต่ละใบไม้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 2, 5, 10 และ 100

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Decision Tree (DT) ที่เหมาะสมกับลักษณะของข้อมูล แสดงดังตาราง 10

ตาราง 10 ค่าพารามิเตอร์ของแบบจำลอง Decision Tree

พารามิเตอร์	ค่า
criterion	gini
max_depth	100
min_samples_split	10
min_samples_leaf	10
class_weight	None

5.1.2 Random Forest (RF)

ในการสร้างแบบจำลอง Random Forest (RF) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 100, 200 ,500 และ 1000

2. max_depth คือ ความลึกของต้นไม้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 10, 50 และ 100

3.min_samples_split คือ จำนวนข้อมูลขั้นต่ำที่ต้องมีเพื่อให้โหนด แยกต่อได้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 2, 5 และ 10

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Random Forest (RF) ที่เหมาะสมกับลักษณะของข้อมูล แสดงดังตาราง 11

ตาราง 11ค่าพารามิเตอร์ของแบบจำลอง Random Forest

พารามิเตอร์	ค่า
n_estimators	1000
max_depth	100
min_samples_split	10
class_weight	None

5.1.3 Extreme Gradient Boosting (XGBoost)

ในการสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 100, 1000 และ 2000

2. max_depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7, 10 และ 100

3. learning_rate คือ ความแรงของการเรียนแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.1, 0.2 และ 0.3

4. subsample คือ สัดส่วนข้อมูลที่สุ่มมาในแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.7, 0.8 และ 0.9

5. min_child_weight คือ จำนวนรวมของ Weight ขั้นต่ำที่จะยอม Split Node ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7, 10 และ 11

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) ที่เหมาะสมกับลักษณะของข้อมูล แสดงดังตาราง 12

ตาราง 12 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost)

พารามิเตอร์	ค่า
n_estimators	1000
max_depth	7

ตาราง 12 (ต่อ)

พารามิเตอร์	ค่า
learning_rate	0.2
subsample	0.8
min_child_weight	10

5.1.4 K-Nearest Neighbors (KNN)

ในการสร้างแบบจำลอง K-Nearest Neighbors (KNN) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. n_neighbors คือ จำนวนเพื่อนบ้านที่ใกล้ที่สุด ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 3, 7 และ 9

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง K-Nearest Neighbors (KNN) ที่เหมาะสมกับลักษณะของข้อมูล แสดงดังตาราง 13

ตาราง 13 ค่าพารามิเตอร์ของแบบจำลอง K-Nearest Neighbors (KNN)

พารามิเตอร์	ค่า
n_neighbors	7

5.1.5 Categorical Boosting (CatBoost)

ในการสร้างแบบจำลอง Categorical Boosting (CatBoost) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. Iterations คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 100, 1000 และ 2000

2. Depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7,10 และ 16

3. learning_rate คือ ความแรงของการเรียนรู้แต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.1, 0.2 และ 0.3

4. min_child_samples คือ จำนวนข้อมูลขั้นต่ำที่ต้องมีในแต่ละ Leaf ก่อนที่โหนดจะสามารถแตกออกได้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 10 และ 100

5. subsample คือ สัดส่วนข้อมูลที่ใช้ต่อรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.7, 0.8 และ 0.9

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Categorical Boosting (CatBoost) ที่เหมาะสมกับลักษณะของข้อมูล ดังตาราง 14

ตาราง 14 ค่าพารามิเตอร์ของแบบจำลอง Categorical Boosting (CatBoost)

พารามิเตอร์	ค่า
Iterations	100
Depth	10
learning_rate	0.2
min_child_samples	10
subsample	0.8

5.2 การสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยชุดข้อมูลที่มีการจัดกาข้อมูลไม่สมดุล

เมื่อได้ผลลัพธ์ของประสิทธิภาพแบบจำลองที่ดีที่สุดในขั้นตอนก่อนหน้าแล้ว ผู้วิจัยได้นำแบบจำลองมาทดลองกับชุดข้อมูลที่ถูกปรับด้วยเทคนิคการแก้ปัญหาข้อมูลไม่สมดุลจำนวน 6 เทคนิค โดยผู้วิจัยได้ใช้ไลบรารี Scikit-learn ในการดำเนินการสร้างแบบจำลอง และกำหนด Random_state = 42 เพื่อให้สามารถทำซ้ำผลลัพธ์ได้อย่างสม่ำเสมอ สำหรับการประเมินประสิทธิภาพของแบบจำลอง ใช้วิธีการแบ่งข้อมูลแบบ Stratified K-Fold Cross Validation จำนวน 5 ชุด และกำหนดค่าพารามิเตอร์ด้วยตนเอง (Manual Setting) เพื่อควบคุมความซับซ้อนของโมเดล

5.2.1 Random Over-Sampling (ROS)

ในการสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยข้อมูลที่มีการจัดการข้อมูลที่ไม่สมดุลด้วยเทคนิค Random Over-Sampling (ROS) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 100, 1000, 2000 และ 3000

2. max_depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 4, 7, 10, 100 และ 1000

3. learning_rate คือ ความแรงของการเรียนแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.01, 0.1, 0.2, 0.3 และ 0.5

4. subsample คือ สัดส่วนข้อมูลที่สุ่มมาในแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.7, 0.9 และ 0.9

5. min_child_weight คือ จำนวนรวมของ Weight ขั้นต่ำที่จะยอม Split Node ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 4, 7, 10 และ 100

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) ที่ใช้ข้อมูลที่มีการจัดการข้อมูลไม่สมดุลด้วยเทคนิค Random Over-Sampling (ROS) ที่เหมาะสมกับลักษณะของข้อมูล ดังตาราง 15

ตาราง 15 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล ROS

พารามิเตอร์	ค่า
n_estimators	3000
max_depth	1000
learning_rate	0.2
subsample	0.7
min_child_weight	1

5.2.2 Synthetic Minority Over-sampling Technique (SMOTE)

ในการสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยข้อมูลที่มีการจัดการข้อมูลที่ไม่สมดุลด้วยเทคนิค Synthetic Minority Over-sampling Technique (SMOTE) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น1000, 2000 และ 3000

2. max_depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น7, 100 และ 1000

3. learning_rate คือ ความแรงของการเรียนแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.01, 0.1 และ 0.2

4. subsample คือ สัดส่วนข้อมูลที่สุ่มมาในแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.8 และ 1

5. min_child_weight คือ จำนวนรวมของ Weight ขั้นต่ำที่จะยอม Split Node ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 4, 7 และ 10

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) ที่ใช้ข้อมูลที่มีการจัดการข้อมูลไม่สมดุลด้วยเทคนิค Synthetic Minority Over-sampling Technique (SMOTE) ที่เหมาะสมกับลักษณะของข้อมูล ดังตาราง 16

ตาราง 16 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล SMOTE

พารามิเตอร์	ค่า
n_estimators	2000
max_depth	7
learning_rate	0.1
subsample	0.8
min_child_weight	7

5.2.3 Tomek Links (TL)

ในการสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยข้อมูลที่มีการจัดการข้อมูลที่ไม่สมดุลด้วยเทคนิค Tomek Links (TL) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 1000, 2000, 3000 และ 5000

2. max_depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7, 10, 100 และ 1000

3. learning_rate คือ ความแรงของการเรียนแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.1, 0.2 และ 0.3

4. subsample คือ สัดส่วนข้อมูลที่สุ่มมาในแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.8 และ 1

5. min_child_weight คือ จำนวนรวมของ Weight ขั้นต่ำที่จะยอม Split node ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 5, 7 และ 10

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) ที่ใช้ข้อมูลที่มีการจัดการข้อมูลไม่สมดุลด้วยเทคนิค Tomek Links (TL) ที่เหมาะสมกับลักษณะของข้อมูลแสดงดังตาราง 17

ตาราง 17 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล TL

พารามิเตอร์	ค่า
n_estimators	1000
max_depth	7
learning_rate	0.1
subsample	0.8
min_child_weight	7

5.2.4 Edited Nearest Neighbors (ENN)

ในการสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยข้อมูลที่มีการจัดการข้อมูลที่ไม่สมดุลด้วยเทคนิค Edited Nearest Neighbors (ENN) ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 2000, 3000 และ 4000

2. max_depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7, 10 และ 100

3. learning_rate คือ ความแรงของการเรียนแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.01, 0.1 และ 0.2

4. subsample คือ สัดส่วนข้อมูลที่สุ่มมาในแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.8 และ 1

5. min_child_weight คือ จำนวนรวมของ Weight ขั้นต่ำที่จะยอม Split Node ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7, 10 และ 100

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) ที่ใช้ข้อมูลที่มีการจัดการข้อมูลไม่สมดุลด้วยเทคนิค Edited Nearest Neighbors (ENN) ที่เหมาะสมกับลักษณะของข้อมูล แสดงดังตาราง 18

ตาราง 18 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล ENN

พารามิเตอร์	ค่า
n_estimators	4000
max_depth	7
learning_rate	0.1
subsample	0.8
min_child_weight	100

5.2.5 SMOTEENN

ในการสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยข้อมูลที่มีการจัดการข้อมูลที่ไม่สมดุลด้วยเทคนิค SMOTEENN ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 2000 และ 4000

2. max_depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7 และ 10

3. learning_rate คือ ความแรงของการเรียนแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.1 และ 0.3

4. subsample คือ สัดส่วนข้อมูลที่สุ่มมาในแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.8 และ 1

5. min_child_weight คือ จำนวนรวมของ Weight ขั้นต่ำที่จะยอม Split Node ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 1, 7 และ 10

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) ที่ใช้ข้อมูลที่มีการจัดการข้อมูลไม่สมดุลด้วยเทคนิค SMOTEENN ที่เหมาะสมกับลักษณะของข้อมูล แสดงดังตาราง 19

ตาราง 19 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล SMOTEENN

พารามิเตอร์	ค่า
n_estimators	2000
max_depth	7
learning_rate	0.1
subsample	0.8
min_child_weight	7

5.2.6 SMOTETomek

ในการสร้างแบบจำลอง Extreme Gradient Boosting (XGBoost) ด้วยข้อมูลที่มีการจัดการข้อมูลที่ไม่สมดุลด้วยเทคนิค SMOTETomek ได้ทำการหา Hyperparameter โดยการปรับค่าพารามิเตอร์ ให้เหมาะสม ดังนี้

1. n_estimators คือ จำนวนรอบการเรียนรู้ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 1000, 2000, 3000 และ 5000

2. depth คือ ความลึกของต้นไม้แต่ละต้น ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 7, 10, 100 และ 500

3.learning_rate คือ ความแรงของการเรียนแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.1 และ 0.2

4. subsample คือ สัดส่วนข้อมูลที่สุ่มมาในแต่ละรอบ ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 0.8 และ 1

5. min_child_weight คือ จำนวนรวมของ Weight ขั้นต่ำที่จะยอม Split Node ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์เป็น 1, 7 และ 10

หลังจากทดลองปรับพารามิเตอร์แล้ว พบว่าค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) ที่ใช้ข้อมูลที่มีการจัดการข้อมูลไม่สมดุลด้วยเทคนิค SMOTETomek ที่เหมาะสมกับลักษณะของข้อมูลแสดงดังตาราง 20

ตาราง 20 ค่าพารามิเตอร์ของแบบจำลอง Extreme Gradient Boosting (XGBoost) กับข้อมูล SMOTETomek

พารามิเตอร์	ค่า
n_estimators	2000
max_depth	7
learning_rate	0.1
subsample	0.8
min_child_weight	7

บทที่ 4
ผลการดำเนินงานวิจัย

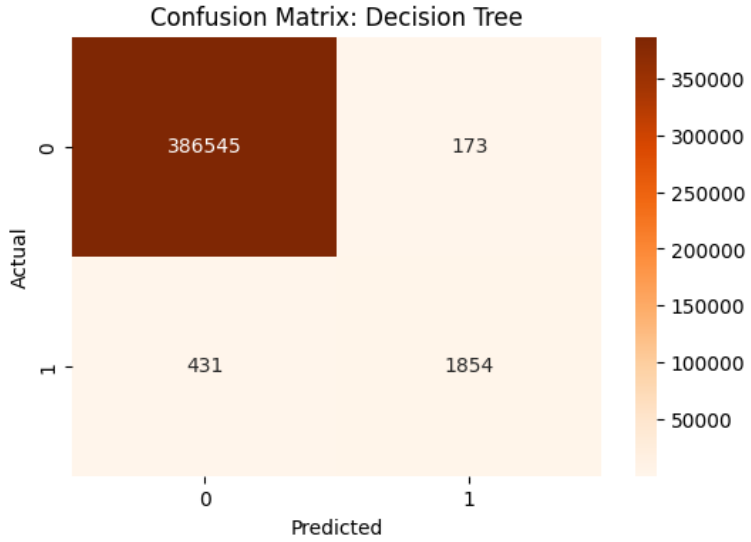
ในการวิจัยการตรวจจับการฉ้อโกงบัตรเครดิตที่ใช้ข้อมูลไม่สมดุลด้วยแบบจำลองการเรียนรู้ของเครื่อง โดยอาศัยข้อมูลจากแหล่งข้อมูลสาธารณะ Kaggle.com ผู้วิจัยได้นำเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) มาประยุกต์ใช้ร่วมกับเทคนิคการจัดการข้อมูลไม่สมดุล (Imbalance Data) เพื่อทำการจำแนกประเภทของธุรกรรมบัตรเครดิตว่าเป็นธุรกรรมที่ทุจริตหรือไม่ โดยดำเนินการศึกษาครอบคลุมตั้งแต่ขั้นตอนการเตรียมข้อมูล การสำรวจและวิเคราะห์คุณลักษณะของข้อมูล การเลือกแบบจำลองที่เหมาะสม ตลอดจนการประเมินผลการทำงานของแบบจำลอง

การวิจัยนี้มีจุดมุ่งหมายเพื่อศึกษาผลกระทบของการจัดการข้อมูลที่ไม่สมดุลต่อความสามารถในการจำแนกของแบบจำลอง โดยแบ่งการทดลองออกเป็น 2 ส่วน ได้แก่

1. การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้ข้อมูลที่ไม่สมดุลด้วยอัลกอริทึม
    - 1.1 Decision Tree (DT)
    - 1.2 Random Forest (RF)
    - 1.3 Extreme Gradient Boosting (XGBoost)
    - 1.4 K-Nearest Neighbors (KNN)
    - 1.5 Categorical Boosting (CatBoost)
2. การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้ข้อมูลที่ผ่านการปรับสมดุลด้วยเทคนิค
    - 2.1 Random Over-Sampling (ROS)
    - 2.2 Synthetic Minority Over-sampling Technique (SMOTE)
    - 2.3 Tomek Links (TL)
    - 2.4 Edited Nearest Neighbors (ENN)
    - 2.5 SMOTEENN
    - 2.6 SMOTETomek

1. การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้ข้อมูลที่ไม่สมดุล

1.1 Decision Tree (DT)



ภาพประกอบ 20 Confusion Matrix Decision Tree

การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม Decision Tree จากการทดลอง พบว่า แบบจำลองมีความสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,545 ตัวอย่าง (True Negative) และเกิดการทำนายผิดเป็นธุรกรรมทุจริตจำนวน 173 ตัวอย่าง (False Positive) สำหรับธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,854 ตัวอย่าง (True Positive) และเกิดการจำแนกผิดพลาดจำนวน 431 ตัวอย่าง (False Negative) ดังภาพประกอบ 20

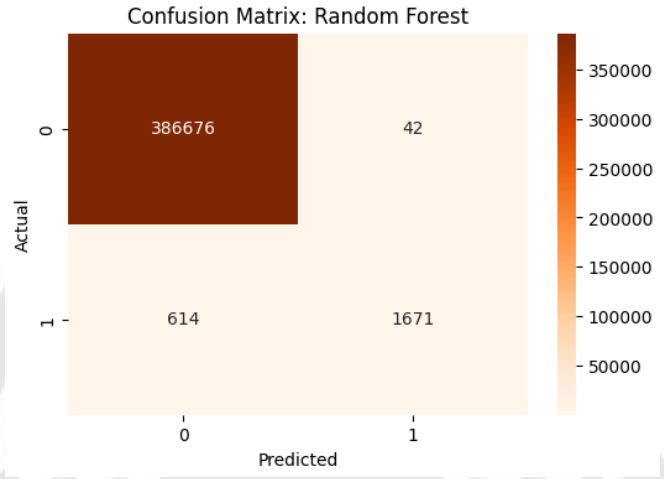
ตาราง 21 ผลการวัดประสิทธิภาพของแบบจำลอง Decision Tree

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.92	0.81	0.86	

จากการประเมินประสิทธิภาพของแบบจำลอง Decision Tree พบว่า การจำแนกธุรกรรมปกติ (Class 0) มีค่าตัวชี้วัด Precision, Recall และ F1-Score เท่ากับ 1 ทั้งหมด ซึ่งหมายความ

ว่าแบบจำลอง Decision Tree สามารถจำแนกธุรกรรมปกติได้อย่างมีประสิทธิภาพ สำหรับการจำแนกธุรกรรมทุจริต (Class 1) ค่าตัวชี้วัด Precision เท่ากับ 0.92 ค่า Recall เท่ากับ 0.81 และค่า F1-Score เท่ากับ 0.86

1.2 Random Forest (RF)



ภาพประกอบ 21 Confusion Matrix Random Forest

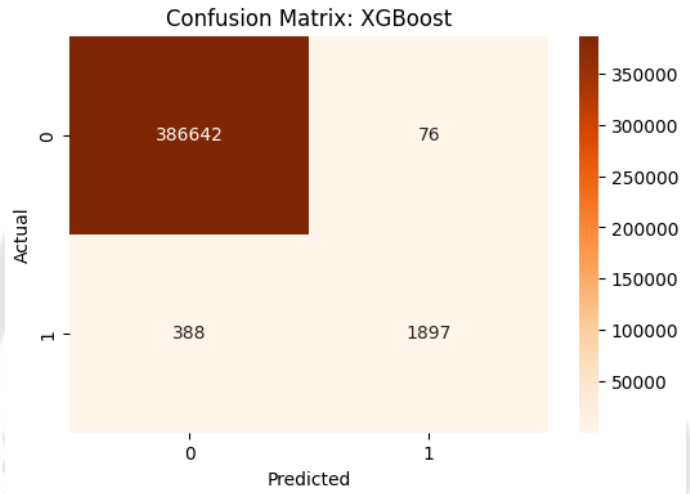
การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม Random Forest จากการทดลองพบว่า แบบจำลองมีความสามารถในการจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,676 ตัวอย่าง (True Negative) และเกิดการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 42 ตัวอย่าง (False Positive) สำหรับธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,671 ตัวอย่าง (True Positive) และเกิดการจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 614 ตัวอย่าง (False Negative) ดังภาพประกอบ 21

ตาราง 22 ผลการวัดประสิทธิภาพของแบบจำลอง Random Forest

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.98	0.73	0.85	

จากการประเมินประสิทธิภาพของแบบจำลอง Random Forest พบว่า ในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด Precision, Recall และ F1-Score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองสามารถมีค่า Precision เท่ากับ 0.98 ค่า Recall เท่ากับ 0.73 และค่า F1-Score เท่ากับ 0.85

1.3 Extreme Gradient Boosting (XGBoost)



ภาพประกอบ 22 Confusion Matrix XGBoost

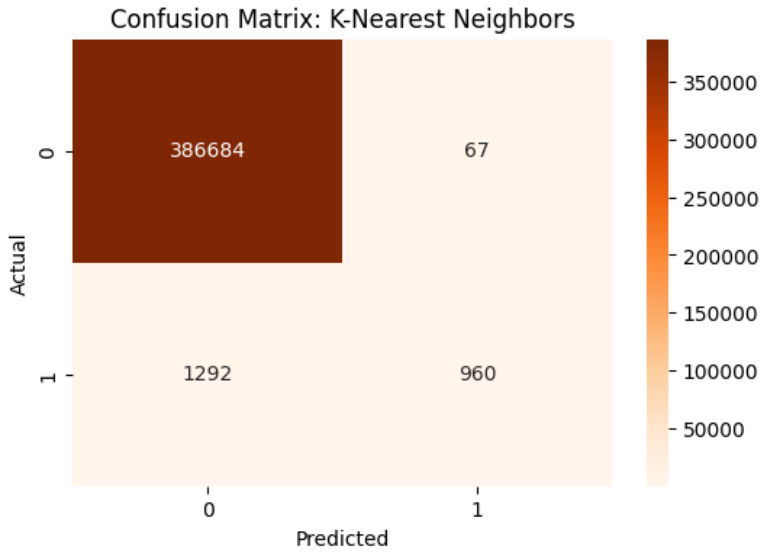
การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม XGBoost จากการทดลองพบว่าแบบจำลองมีความสามารถในการจำแนกธุรกรรมปกติ (Class 0) ได้อย่างแม่นยำ โดยสามารถจำแนกได้ถูกต้องจำนวน 386,642 ตัวอย่าง (True Negative) และเกิดการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 76 ตัวอย่าง (False Positive) สำหรับธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,897 ตัวอย่าง (True Positive) และเกิดการจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 388 ตัวอย่าง (False Negative) ดังภาพประกอบ 22

ตาราง 23 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.96	0.83	0.89	

จากการประเมินประสิทธิภาพของแบบจำลอง XGBoost พบว่า ในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด Precision, Recall และ F1-Score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลอง XGBoost มีค่า Precision เท่ากับ 0.98 ค่า Recall เท่ากับ 0.73 และค่า F1-Score เท่ากับ 0.85

1.4 K-Nearest Neighbors (KNN)



ภาพประกอบ 23 Confusion Matrix K-Nearest Neighbors (KNN)

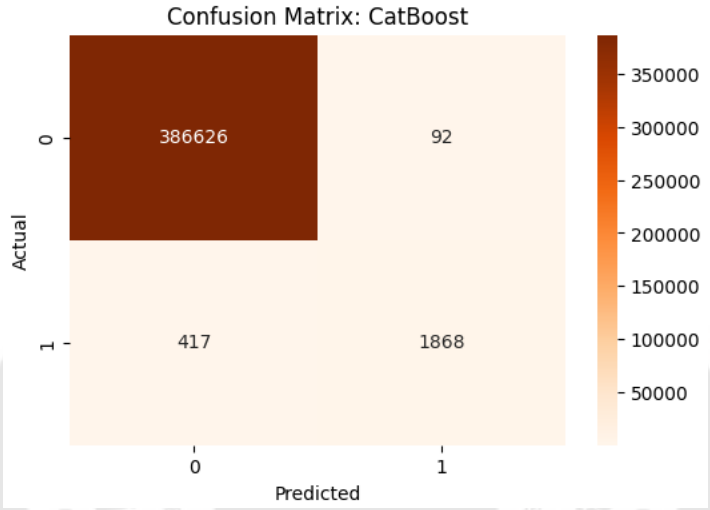
การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม K-Nearest Neighbors (KNN) พบว่าแบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,684 ตัวอย่าง (True Negative) และมีการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 67 ตัวอย่าง (False Positive) สำหรับธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 960 ตัวอย่าง (True Positive) และมีการจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 1,292 ตัวอย่าง (False Negative) ดังภาพประกอบ 23

ตาราง 24 ผลการวัดประสิทธิภาพของแบบจำลอง K-Nearest Neighbors (KNN)

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.91	0.41	0.57	

จากการประเมินประสิทธิภาพของแบบจำลอง K-Nearest Neighbors (KNN) พบว่าในการจำแนกธุรกรรมปกติ (Class 0) มีค่าตัวชี้วัด Precision, Recall และ F1-score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบมีค่า Precision เท่ากับ 0.91 ค่า Recall เท่ากับ 0.41 และค่า F1-score เท่ากับ 0.57

1.5 Categorical Boosting (CatBoost)



ภาพประกอบ 24 Confusion Matrix Categorical Boosting (CatBoost)

การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม CatBoost พบว่าแบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,626 ตัวอย่าง (True Negative) และเกิดการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 92 ตัวอย่าง (False Positive) ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,868 ตัวอย่าง (True Positive) และเกิดการจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 417 ตัวอย่าง (False Negative) ดังภาพประกอบ 24

ตาราง 25 ผลการวัดประสิทธิภาพของแบบจำลอง CatBoost

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.95	0.82	0.88	

จากการประเมินประสิทธิภาพของแบบจำลอง CatBoost พบว่าในการจำแนกธุรกรรมปกติ (Class 0) แบบจำมีค่าตัวชี้วัด Precision, Recall และ F1-score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองมีค่า Precision เท่ากับ 0.95 ค่า Recall เท่ากับ 0.82 และค่า F1-score เท่ากับ 0.88

ตาราง 26 ผลการวัดประสิทธิภาพของแบบจำลอง โดยไม่ได้ใช้เทคนิคจัดการข้อมูลไม่สมดุลของธุรกรรมปกติ (Class 0)

Training Model	Accuracy	Precision	Recall	F1-Score
KNN	1	1	1	1
Random Forest	1	1	1	1
Decision tree	1	1	1	1
XGBoost	1	1	1	1
CatBoost	1	1	1	1

ตาราง 27 ผลการวัดประสิทธิภาพของแบบจำลอง โดยไม่ได้ใช้เทคนิคจัดการข้อมูลไม่สมดุลของธุรกรรมทุจริต (Class1)

Training Model	Accuracy	Precision	Recall	F1-Score
KNN	1	0.91	0.41	0.57
Random Forest	1	0.98	0.73	0.85
Decision tree	1	0.92	0.81	0.86
XGBoost	1	0.96	0.83	0.89
CatBoost	1	0.95	0.82	0.88

จากผลการทดลองที่แสดงในตารางที่ 26 และตารางที่ 27 พบว่าแบบจำลอง ได้แก่ K-Nearest Neighbors (KNN), Random Forest, Decision Tree, XGBoost และ CatBoost สามารถจำแนกธุรกรรมประเภทปกติ (Class 0) ได้อย่างแม่นยำ โดยให้ค่า Accuracy, Precision, Recall และ F1-score เท่ากับ 1 ทั้งหมด เนื่องจากข้อมูลมีจำนวนตัวอย่างจำนวนมาก ส่งผลให้

แบบจำลองสามารถเรียนรู้และจำแนกข้อมูลได้อย่างมีประสิทธิภาพ เมื่อพิจารณาประสิทธิภาพของแบบจำลองในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองทั้ง 5 แบบจำลอง มีค่า Accuracy เท่ากับ 1 ทั้งหมด แต่เนื่องจากข้อมูลมีความไม่สมดุลสูง การใช้ค่า Accuracy เพียงอย่างเดียวไม่สามารถสะท้อนประสิทธิภาพของแบบจำลองได้ จึงต้องพิจารณาค่าชี้วัดอื่นโดยเฉพาะ Recall ของธุรกรรมทุจริต (Class 1) เพิ่มเติม

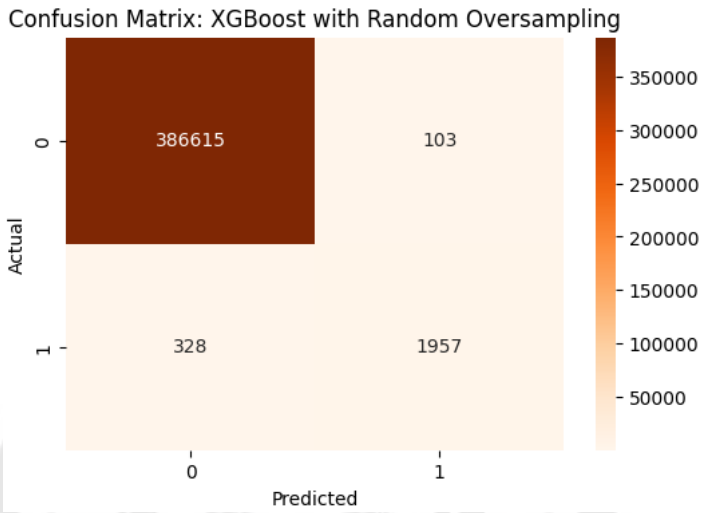
สำหรับแบบจำลอง K-Nearest Neighbors (KNN) แม้จะให้ค่า Precision สูงถึง 0.91 แต่มีค่า Recall เพียง 0.41 ซึ่งต่ำที่สุดในทุกแบบจำลอง แสดงให้เห็นว่าแบบจำลองมีข้อจำกัดในการตรวจจับธุรกรรมทุจริต ส่งผลให้ F1-score อยู่ในระดับต่ำเพียง 0.57 ขณะที่ Random Forest ให้ค่า Precision สูงถึง 0.98 และ Recall เท่ากับ 0.73 ซึ่งอยู่ในระดับที่ดี แต่ยังน้อยกว่าแบบจำลองอื่นบางตัว สำหรับ Decision Tree, XGBoost และ CatBoost แสดงประสิทธิภาพโดยรวมได้ดี โดยเฉพาะ XGBoost ที่ให้ค่า Precision เท่ากับ 0.96, Recall เท่ากับ 0.83 และ F1-score เท่ากับ 0.89 ซึ่งสูงที่สุดในแบบจำลองทั้งหมด แสดงให้เห็นถึงศักยภาพของ XGBoost ในการตรวจจับธุรกรรมทุจริตได้อย่างแม่นยำและครอบคลุมมากที่สุด โดยมี CatBoost และ Decision tree รองลงมาตามลำดับ

2.การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้ข้อมูลที่ผ่านการปรับสมดุล

จากการทดสอบการจำแนกประเภทธุรกรรมการทุจริตบัตรเครดิตโดยใช้ข้อมูลไม่สมดุล พบว่าแบบจำลองที่มีประสิทธิภาพในการจำแนกธุรกรรมบัตรเครดิตมากที่สุดคือ XGBoost โดยมีค่า Recall อยู่ที่ 0.83 ค่า Precision เท่ากับ 0.96 และค่า F1-score เท่ากับ 0.89

ผู้วิจัยจึงเลือกแบบจำลอง XGBoost มาใช้เป็นแบบจำลองหลักในการทดสอบเปรียบเทียบวิธีการจัดการกับข้อมูลที่ไม่สมดุล (Imbalanced Data)

2.1 Random Over-Sampling (ROS)



ภาพประกอบ 25 Confusion Matrix XGBoost กับข้อมูล ROS

การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม XGBoost ร่วมกับเทคนิค Random Oversampling พบว่าแบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,615 ตัวอย่าง (True Negative) และมีการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 103 ตัวอย่าง (False Positive) สำหรับธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,957 ตัวอย่าง (True Positive) และจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 328 ตัวอย่าง (False Negative) ดังภาพประกอบ 25

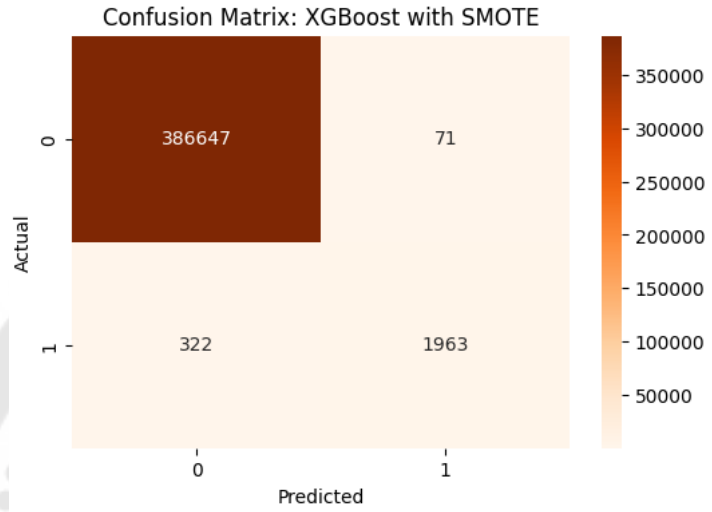
ตาราง 28 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล Random Oversampling

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.94	0.86	0.90	

จากการประเมินประสิทธิภาพของแบบจำลอง XGBoost ที่ปรับสมดุลข้อมูลด้วยเทคนิค Random Oversampling พบว่าในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด

Precision, Recall และ F1-score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองมีค่า Precision เท่ากับ 0.94, Recall เท่ากับ 0.86 และค่า F1-score เท่ากับ 0.90

2.2 Synthetic Minority Over-sampling Technique (SMOTE)



ภาพประกอบ 26 Confusion Matrix XGBoost กับข้อมูล SMOTE

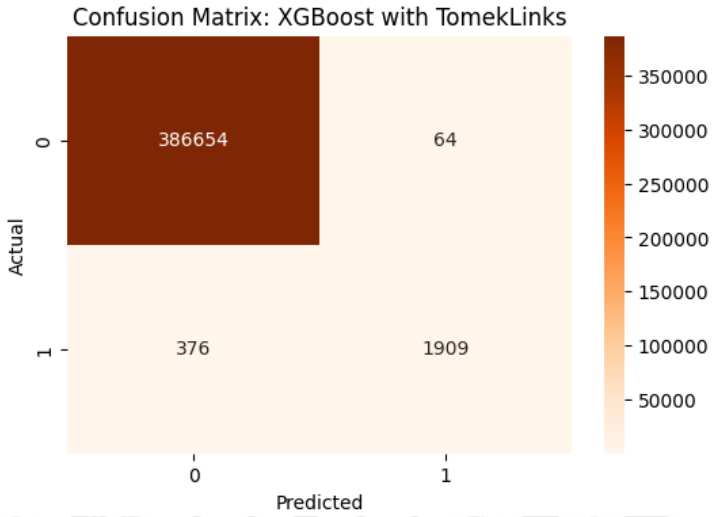
การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม XGBoost ร่วมกับเทคนิค SMOTE พบว่าแบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,647 ตัวอย่าง (True Negative) และมีการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 71 ตัวอย่าง (False Positive) ขณะที่การจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,963 ตัวอย่าง (True Positive) และจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 322 ตัวอย่าง (False Negative) ดังภาพประกอบ 26

ตาราง 29 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล SMOTE

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.97	0.86	0.91	

จากการประเมินประสิทธิภาพของแบบจำลอง XGBoost ที่ใช้เทคนิค SMOTE เพื่อปรับสมดุลข้อมูล พบว่าในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด Precision, Recall และ F1-score เท่ากับ 1 ทั้งหมด ขณะที่การจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองมีค่า Precision เท่ากับ 0.97, Recall เท่ากับ 0.86 และ F1-score เท่ากับ 0.91

2.3. Tomek Links (TL)



ภาพประกอบ 27 Confusion Matrix XGBoost กับข้อมูล Tomek Links

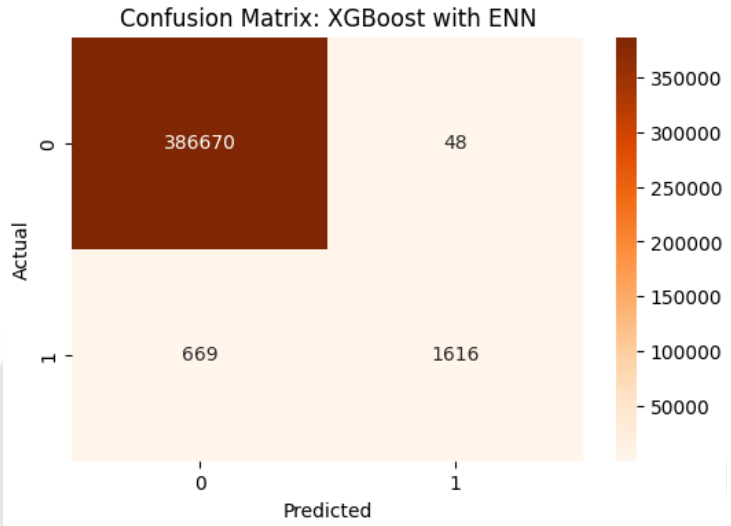
การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม XGBoost ร่วมกับเทคนิค Tomek Links พบว่าแบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,654 ตัวอย่าง (True Negative) และมีการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 64 ตัวอย่าง (False Positive) ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,909 ตัวอย่าง (True Positive) และมีการจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 376 ตัวอย่าง (False Negative) ดังภาพประกอบ 27

ตาราง 30 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล Tomek Links

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.97	0.84	0.90	

จากการประเมินประสิทธิภาพของแบบจำลอง XGBoost ที่ใช้เทคนิค Tomek Links พบว่าในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด Precision, Recall และ F1-score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองมีค่า Precision เท่ากับ 0.97 ค่า Recall เท่ากับ 0.84 และค่า F1-score เท่ากับ 0.90

2.4 Edited Nearest Neighbors (ENN)



ภาพประกอบ 28 Confusion Matrix XGBoost กับข้อมูล ENN

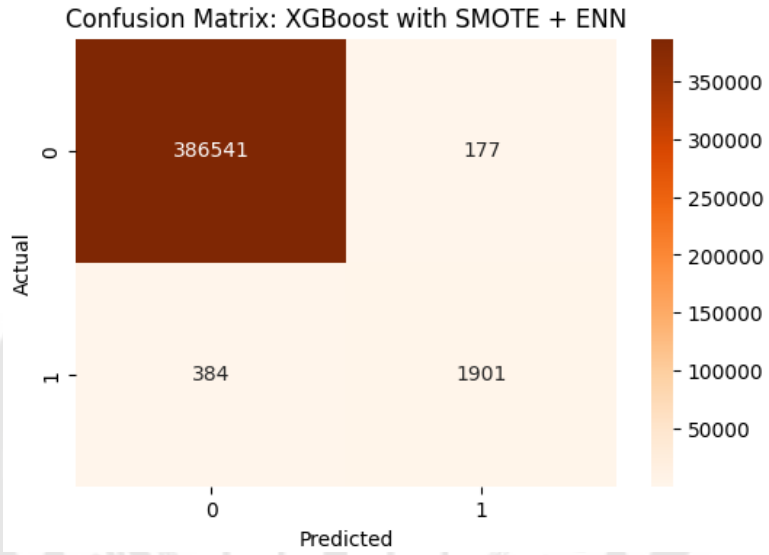
การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม XGBoost ร่วมกับเทคนิค Edited Nearest Neighbors (ENN) พบว่าแบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,670 ตัวอย่าง (True Negative) และมีการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 48 ตัวอย่าง (False Positive) ส่วนในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,616 ตัวอย่าง (True Positive) และเกิดการจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 669 ตัวอย่าง (False Negative) ดังภาพประกอบ 28

ตาราง 31 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล ENN

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.97	0.71	0.82	

จากการประเมินประสิทธิภาพของแบบจำลอง XGBoost ที่ใช้เทคนิค Edited Nearest Neighbors (ENN) พบว่าในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด Precision, Recall และ F1-score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองมีค่า Precision เท่ากับ 0.97 ค่า Recall เท่ากับ 0.71 และ F1-score เท่ากับ 0.82

2.5 SMOTEENN



ภาพประกอบ 29 Confusion Matrix XGBoost กับข้อมูล SMOTEENN

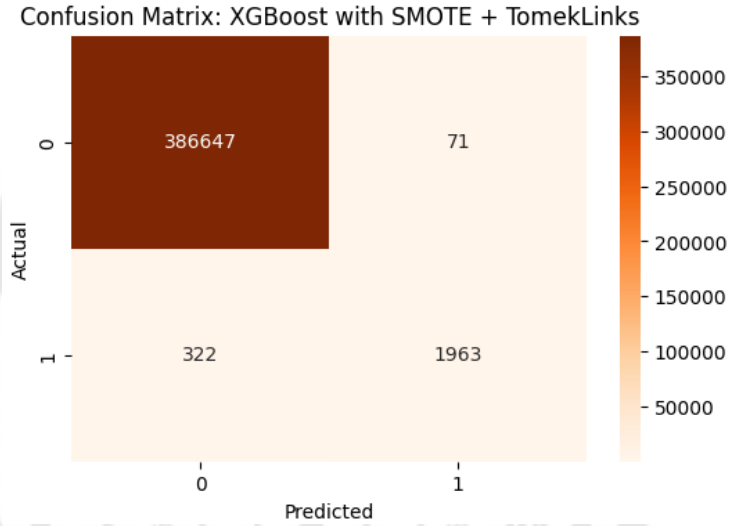
การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม XGBoost ร่วมกับเทคนิค SMOTEENN พบว่าแบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้ถูกต้องจำนวน 386,541 ตัวอย่าง (True Negative) และมีการทำนายผิดว่าเป็นธุรกรรมทุจริตจำนวน 177 ตัวอย่าง (False Positive) ขณะที่ในฝั่งของธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,901 ตัวอย่าง (True Positive) และเกิดการจำแนกผิดพลาดว่าเป็นธุรกรรมปกติจำนวน 384 ตัวอย่าง (False Negative) ดังภาพประกอบ 29

ตาราง 32 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล SMOTEENN

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.91	0.83	0.87	

จากการประเมินประสิทธิภาพของแบบจำลอง XGBoost ที่ใช้เทคนิค SMOTE ร่วมกับ Edited Nearest Neighbors (SMOTE-ENN) พบว่า ในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด Precision, Recall และ F1-Score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองมีค่าตัวชี้วัด Precision เท่ากับ 0.91 ค่า Recall เท่ากับ 0.83 และค่า F1-Score เท่ากับ 0.87

2.6 SMOTETomek



ภาพประกอบ 30 Confusion Matrix XGBoost กับข้อมูล SMOTETomek

การจำแนกประเภทธุรกรรมบัตรเครดิตโดยใช้อัลกอริทึม XGBoost ร่วมกับเทคนิค SMOTETomek พบว่า แบบจำลองสามารถจำแนกธุรกรรมปกติ (Class 0) ได้อย่างแม่นยำ โดยสามารถจำแนกได้ถูกต้องจำนวน 386,647 ตัวอย่าง (True Negative) และมีการทำนายผิดว่าเป็นธุรกรรมทุจริตเพียง 71 ตัวอย่าง (False Positive) ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองสามารถจำแนกได้ถูกต้องจำนวน 1,963 ตัวอย่าง (True Positive) และมีการทำนายผิดว่าเป็นธุรกรรมปกติจำนวน 322 ตัวอย่าง (False Negative) ดังภาพประกอบ 30

ตาราง 33 ผลการวัดประสิทธิภาพของแบบจำลอง XGBoost กับข้อมูล SMOTETomek

Class	Precision	Recall	F1-Score	Accuracy
0	1	1	1	1
1	0.97	0.86	0.91	

จากการประเมินประสิทธิภาพของแบบจำลอง XGBoost ที่ใช้เทคนิค SMOTE ร่วมกับ Tomek Links (SMOTETomek) พบว่า ในการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองมีค่าตัวชี้วัด Precision ค่า Recall และ F1-Score เท่ากับ 1 ทั้งหมด ขณะที่ในการจำแนกธุรกรรมทุจริต (Class 1) แบบจำลองมีค่าตัวชี้วัด Precision เท่ากับ 0.97 ค่า Recall เท่ากับ 0.86 และค่า F1-Score เท่ากับ 0.91

ตาราง 34 ผลการทดลองของอัลกอริทึม XGBoost ที่มีการจัดการข้อมูลมีสมดุล (Class 0)

Imbalance data method with XGBoost	Accuracy	Precision	Recall	F1-Score
Random Oversampling	1	1	1	1
SMOTE	1	1	1	1
Tomek Links	1	1	1	1
ENN	1	1	1	1
SMOTEENN	1	1	1	1
SMOTETomek	1	1	1	1

ตาราง 35 ผลการทดลองของอัลกอริทึม XGBoost ที่มีการจัดการข้อมูลมีสมดุล (Class 1)

Imbalance data method with XGBoost	Accuracy	Precision	Recall	F1-Score
Random Oversampling	1	0.94	0.86	0.90
SMOTE	1	0.97	0.86	0.91
Tomek Links	1	0.97	0.84	0.90
ENN	1	0.97	0.71	0.82

ตาราง 35 (ต่อ)

Imbalance data method with XGBoost	Accuracy	Precision	Recall	F1-Score
SMOTEENN	1	0.91	0.83	0.87
SMOTETomek	1	0.97	0.86	0.91

ผลการทดลองที่แสดงในตารางที่ 34 และตารางที่ 35 เป็นการประเมินประสิทธิภาพของแบบจำลอง XGBoost ใช้ข้อมูลที่มีการจัดการความไม่สมดุลของข้อมูล พบว่าในส่วนของการจำแนกธุรกรรมปกติ (Class 0) แบบจำลองสามารถจำแนกได้อย่างแม่นยำในทุกตัวชี้วัด โดยค่าชี้วัดต่าง ๆ ได้แก่ Accuracy, Precision, Recall และ F1-Score ต่างให้ค่าเท่ากับ 1 ทั้งหมด

อย่างไรก็ตาม เมื่อพิจารณาการจำแนกธุรกรรมทุจริต (Class 1) ซึ่งเป็นกลุ่มข้อมูลส่วนน้อย พบว่าค่า Accuracy เท่ากับ 1 แต่ตัวชี้วัดที่สำคัญอย่าง Recall มีความแตกต่างกันอย่างชัดเจน สำหรับเทคนิคที่ให้ผลลัพธ์ดีที่สุดคือ SMOTE และ SMOTE ร่วมกับ Tomek Links โดยทั้งสองเทคนิคมีค่าตัวชี้วัด Precision เท่ากับ 0.97 ค่า Recall เท่ากับ 0.86 และ F1-Score เท่ากับ 0.91 ซึ่งแสดงให้เห็นถึงความสามารถในการตรวจจับธุรกรรมทุจริตได้อย่างแม่นยำ

ขณะที่เทคนิค ENN แม้จะให้ค่า Precision สูงถึง 0.97 แต่มีค่า Recall ต่ำเพียง 0.71 ส่งผลให้ค่า F1-Score อยู่ในระดับต่ำสุดที่ 0.82 ซึ่งบ่งบอกถึงข้อจำกัดในการตรวจจับธุรกรรมทุจริตทั้งหมด ด้านเทคนิค Random Oversampling และ Tomek Links ให้ผลลัพธ์อยู่ในระดับดี โดยมี F1-Score เท่ากับ 0.90 ส่วน SMOTE ร่วมกับ ENN ให้ค่า F1-Score เท่ากับ 0.87 แม้จะมี Recall ที่ดี แต่ Precision ต่ำกว่าวิธีอื่น

จากผลการทดลองนี้สามารถสรุปได้ว่า การใช้เทคนิค SMOTE หรือ SMOTE ร่วมกับ Tomek Links เป็นทางเลือกที่มีประสิทธิภาพสูงในการเพิ่มศักยภาพของแบบจำลอง XGBoost

บทที่ 5
สรุปผลการทดลอง

บัตรเครดิตเป็นเครื่องมือทางการเงินที่สะดวกและได้รับความนิยมสูง แต่การใช้งานที่แพร่หลายก่อให้เกิดความเสี่ยงด้านการฉ้อโกงเพิ่มมากขึ้น งานวิจัยนี้จึงมุ่งศึกษาการประยุกต์ใช้เทคโนโลยี Machine Learning ในการตรวจจับธุรกรรมทุจริต โดยเน้นปัญหาข้อมูลไม่สมดุล (Imbalanced Data) ซึ่งเป็นอุปสรรคสำคัญในการพัฒนาแบบจำลอง ในการวิจัยการตรวจจับการฉ้อโกงธุรกรรมบัตรเครดิตด้วยเทคนิคการเรียนรู้ของเครื่อง ผู้วิจัยได้ประเมินประสิทธิภาพของแบบจำลอง และศึกษาการใช้เทคนิคการจัดการข้อมูลไม่สมดุล เพื่อเปรียบเทียบและสรุปผล โดยแบ่งการนำเสนอออกเป็นหัวข้อดังนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะ

1. สรุปผลการวิจัย

ในการวิจัยการตรวจจับการฉ้อโกงธุรกรรมบัตรเครดิต ซึ่งใช้ข้อมูลที่ไม่สมดุล (Imbalanced Data) ผู้วิจัยได้แบ่งการทดลองออกเป็นสองส่วน โดยส่วนแรกเป็นการเปรียบเทียบประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องจำนวน 5 แบบ ได้แก่ K-Nearest Neighbors (KNN), Random Forest (RF), Decision Tree (DT), Extreme Gradient Boosting (XGBoost) และ Categorical Boosting (CatBoost) เพื่อประเมินความสามารถในการจำแนกประเภทธุรกรรม จากนั้น เมื่อทราบว่าแบบจำลองใดมีประสิทธิภาพสูงสุด ผู้วิจัยจึงนำแบบจำลองนั้นไปทดสอบร่วมกับเทคนิคการจัดการข้อมูลไม่สมดุลจำนวน 6 เทคนิค ได้แก่ Random Oversampling, SMOTE (Synthetic Minority Over-sampling Technique), Tomek Links, Edited Nearest Neighbors (ENN), SMOTEENN และ SMOTETomek เพื่อประเมินว่าเทคนิคใดสามารถเพิ่มประสิทธิภาพของแบบจำลองได้มากที่สุด

จากผลการทดลองในส่วนแรก การประเมินประสิทธิภาพของแบบจำลองในการจำแนกธุรกรรมทุจริต (Class 1) พบว่าแบบจำลอง XGBoost มีประสิทธิภาพโดยรวมสูงที่สุด โดยมีค่าตัวชี้วัด Precision เท่ากับ 0.96 ค่า Recall เท่ากับ 0.83 และ F1-Score เท่ากับ 0.89 ในขณะที่แบบจำลอง CatBoost และ Decision Tree มีผลลัพธ์ที่ใกล้เคียงกัน โดย CatBoost ให้ค่า F1-

Score เท่ากับ 0.88 และ Decision Tree เท่ากับ 0.86 แสดงถึงประสิทธิภาพที่ดีในการจำแนกข้อมูลเช่นกัน

สำหรับแบบจำลอง Random Forest แม้ว่าจะให้ค่า Precision สูงสุดถึง 0.98 แต่มีค่า Recall เพียง 0.73 ส่งผลให้ค่า F1-Score ลดลงมาอยู่ที่ 0.85 ในทางตรงกันข้าม แบบจำลอง K-Nearest Neighbors (KNN) แม้จะให้ค่า Precision สูงถึง 0.91 แต่มีค่า Recall ต่ำสุดเพียง 0.41 ส่งผลให้ F1-Score อยู่ที่ระดับต่ำสุดคือ 0.57 ซึ่งสะท้อนถึงข้อจำกัดของ KNN ในการตรวจจับธุรกรรมทุจริต

นอกจากนี้ เมื่อพิจารณาผลการจำแนกธุรกรรมปกติ (Class 0) ซึ่งเป็นกลุ่มข้อมูลส่วนใหญ่ในชุดข้อมูล พบว่าแบบจำลองทั้งหมด ได้แก่ KNN, Random Forest, Decision Tree, XGBoost และ CatBoost ต่างสามารถจำแนกข้อมูลกลุ่มนี้ได้อย่างแม่นยำ โดยให้ค่า Accuracy, Precision, Recall และ F1-Score เท่ากับ 1 ทั้งหมด ซึ่งแสดงให้เห็นถึงประสิทธิภาพในการเรียนรู้ของแบบจำลองจากกลุ่มข้อมูลที่มีจำนวนมาก ทำให้สามารถแยกแยะได้อย่างแม่นยำ ดังแสดงในตารางที่ 36

ตาราง 36 ผลการทดลองของอัลกอริทึมการจำแนกประเภทต่างๆ โดยไม่ได้ใช้เทคนิคจัดการข้อมูลไม่สมดุล

Training Model	class	Precision	Recall	F1-Score	Accuracy
KNN	1	1	1	1	1
	0	0.91	0.41	0.57	
Random Forest	1	1	1	1	1
	0	0.98	0.73	0.85	
Decision tree	1	1	1	1	1
	0	0.92	0.81	0.86	
XGBoost	1	1	1	1	1
	0	0.96	0.83	0.89	
CatBoost	1	1	1	1	1
	0	0.95	0.82	0.88	

จากผลการทดลองในส่วนที่ 2 ซึ่งเป็นการประเมินประสิทธิภาพของแบบจำลอง XGBoost เมื่อใช้ร่วมกับเทคนิคการจัดการข้อมูลไม่สมดุล สำหรับภารกิจในการจำแนกธุรกรรมทุจริต (Class 1) พบว่าเทคนิคที่ให้ผลลัพธ์ดีที่สุด ได้แก่ SMOTE และ SMOTE ร่วมกับ Tomek Links โดยทั้งสองเทคนิคให้ค่า Precision เท่ากับ 0.97, Recall เท่ากับ 0.86 และ F1-Score เท่ากับ 0.91 ในทางตรงกันข้าม เทคนิค ENN แม้จะมีค่า Precision สูงถึง 0.97 แต่กลับมีค่า Recall ต่ำที่สุดเพียง 0.71 ส่งผลให้ค่า F1-Score ลดลงเหลือ 0.82 ซึ่งเป็นค่าที่ต่ำที่สุดในทุกเทคนิคที่นำมาทดสอบ สำหรับเทคนิค Random Oversampling และ Tomek Links ให้ค่า F1-Score อยู่ในระดับดีที่ 0.90 ขณะที่ SMOTE ร่วมกับ ENN ให้ค่า F1-Score เท่ากับ 0.87 แม้จะมีค่า Recall อยู่ในระดับที่น่าพอใจ แต่ Precision ต่ำกว่าวิธีอื่นเล็กน้อย

นอกจากการประเมินประสิทธิภาพของแบบจำลอง XGBoost ในการจำแนกธุรกรรมทุจริต (Class 1) แล้ว ยังได้พิจารณาถึงความสามารถในการจำแนกธุรกรรมปกติ (Class 0) ซึ่งเป็นกลุ่มข้อมูลที่มีจำนวนมากที่สุดในชุดข้อมูลด้วย ผลการทดลองพบว่า ไม่ว่าจะใช้เทคนิคการจัดการข้อมูลแบบใด แบบจำลอง XGBoost ยังคงสามารถจำแนกข้อมูลธุรกรรมปกติได้อย่างแม่นยำ โดยให้ค่า Accuracy, Precision, Recall และ F1-Score เท่ากับ 1 ในทุกตัวชี้วัด สะท้อนให้เห็นว่าเทคนิคการจัดการข้อมูลไม่สมดุลไม่มีผลกระทบเชิงลบต่อความสามารถของแบบจำลองในการจัดการกับกลุ่มข้อมูลขนาดใหญ่ (Class 0) และในขณะเดียวกันยังช่วยยกระดับประสิทธิภาพในการตรวจจับกลุ่มข้อมูลส่วนน้อย (Class 1) ได้อย่างมีประสิทธิภาพ ดังแสดงในตารางที่ 37

ตาราง 37 ผลการทดลองของอัลกอริทึม XGBoost โดยใช้เทคนิคการจัดการความไม่สมดุลข้อมูล

Imbalanced data method with XGBoost	Class	Precision	Recall	F1-Score	Accuracy
Random Oversampling	1	1	1	1	1
	0	0.94	0.86	0.9	
SMOTE	1	1	1	1	1
	0	0.97	0.86	0.91	
Tomek Links	1	1	1	1	1
	0	0.97	0.84	0.90	

ตาราง 37 (ต่อ)

Imbalanced data method with XGBoost	Class	Precision	Recall	F1-Score	Accuracy
ENN	1	1	1	1	1
	0	0.97	0.71	0.82	
SMOTEENN	1	1	1	1	1
	0	0.91	0.83	0.87	
SMOTETomek	1	1	1	1	1
	0	0.97	0.86	0.91	

จากผลการทดลองทั้งสองส่วน พบว่าแบบจำลอง XGBoost เป็นแบบจำลองที่มีประสิทธิภาพสูงที่สุดในการจำแนกธุรกรรมทุจริต (Class 1) โดยเฉพาะเมื่อพิจารณาตัวชี้วัดที่สำคัญอย่าง Recall, Precision และ F1-Score ซึ่งสะท้อนถึงความสามารถของแบบจำลองในการตรวจจับธุรกรรมทุจริตได้อย่างถูกต้องและครอบคลุม อีกทั้งเมื่อใช้เทคนิคการจัดการข้อมูลไม่สมดุลร่วมกับ XGBoost ยังช่วยเพิ่มประสิทธิภาพของแบบจำลองให้ดียิ่งขึ้น โดยเฉพาะเทคนิค SMOTE และ SMOTE ร่วมกับ Tomek Links ซึ่งให้ค่า Precision และ Recall ที่สมดุลกัน และส่งผลให้ค่า F1-Score สูงที่สุดในบรรดาทุกวิธีที่นำมาทดสอบ แสดงให้เห็นว่าเทคนิคเหล่านี้สามารถลดปัญหาคลาสไม่สมดุลได้อย่างมีประสิทธิภาพ

ในทางตรงกันข้าม เทคนิค ENN แม้จะช่วยเพิ่มค่า Precision ได้ดี แต่กลับมีค่า Recall ต่ำ ส่งผลให้แบบจำลองพลาดการตรวจจับธุรกรรมทุจริตจำนวนมาก และทำให้ประสิทธิภาพโดยรวมลดลง ขณะเดียวกัน แบบจำลอง XGBoost ยังสามารถจำแนกธุรกรรมปกติ (Class 0) ซึ่งเป็นกลุ่มข้อมูลส่วนใหญ่ในชุดข้อมูลได้อย่างแม่นยำ โดยไม่ว่าใช้เทคนิคการจัดการข้อมูลไม่สมดุลด้วยวิธีใดก็ตาม แบบจำลองยังคงให้ค่า Accuracy, Precision, Recall และ F1-Score เท่ากับ 1 อย่างสม่ำเสมอ สะท้อนให้เห็นว่าเทคนิคการจัดการข้อมูลไม่สมดุลไม่มีผลกระทบเชิงลบต่อความสามารถของแบบจำลองในการจำแนกธุรกรรมปกติซึ่งเป็นกลุ่มข้อมูลขนาดใหญ่ และยังช่วยยกระดับประสิทธิภาพของแบบจำลองในการตรวจจับกลุ่มข้อมูลส่วนน้อยได้อย่างมีประสิทธิผลอีกด้วย

2. อภิปรายผลการวิจัย

งานวิจัยนี้ศึกษาการจำแนกประเภท (Classification) ซึ่งเป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยเปรียบเทียบแบบจำลองทั้งหมด 5 แบบ ได้แก่ K-Nearest Neighbors (KNN), Random Forest (RF), Decision Tree (DT), Extreme Gradient Boosting (XGBoost) และ Categorical Boosting (CatBoost) รวมถึงการเปรียบเทียบประสิทธิภาพของเทคนิคการจัดการข้อมูลไม่สมดุลจำนวน 6 เทคนิค ได้แก่ Random Over-Sampling (ROS), Synthetic Minority Over-sampling Technique (SMOTE), Tomek Links (TL), Edited Nearest Neighbors (ENN), SMOTEENN และ SMOTETomek

โดยเริ่มจากการศึกษาแบบจำลองการจำแนกประเภทเพื่อใช้ในการจำแนกประเภทธุรกรรมทุจริต ผู้วิจัยได้ทดลองปรับค่าพารามิเตอร์ของแบบจำลองแต่ละตัวด้วยค่าที่หลากหลายเพื่อค้นหาค่าที่ให้ผลลัพธ์ดีที่สุด โดยประเมินความสามารถในการจำแนกประเภทด้วยค่าตัวชี้วัด Recall, Precision และ F1-score เมื่อได้แบบจำลองที่มีประสิทธิภาพสูงสุดแล้ว จึงนำไปศึกษาร่วมกับเทคนิคการจัดการข้อมูลที่ไม่สมดุล เพื่อเพิ่มประสิทธิภาพของแบบจำลอง

จากผลการศึกษา พบว่าแบบจำลอง XGBoost ให้ผลลัพธ์ดีที่สุด รองลงมาคือ CatBoost, Random Forest (RF) และ Decision Tree (DT) โดยพิจารณาจากค่าตัวชี้วัด Recall, Precision และ F1-score ตามลำดับ ขณะที่แบบจำลอง K-Nearest Neighbors (KNN) มีค่าตัวชี้วัด Recall ต่ำที่สุด

แบบจำลอง XGBoost เป็นแบบจำลองที่มีประสิทธิภาพสูงสุดในการจำแนกประเภทธุรกรรมทุจริต เนื่องจากแบบจำลอง XGBoost สามารถจัดการกับข้อมูลที่ไม่สมดุลได้ดีในตัวอยู่แล้ว โดยการเรียนรู้แบบ Boosting ช่วยเน้นย้ำตัวอย่างที่ถูกจำแนกผิด ซึ่งมักเป็นข้อมูลธุรกรรมทุจริต (Class 1) ทำให้แบบจำลองค่อย ๆ ปรับปรุงความแม่นยำของกลุ่มข้อมูลธุรกรรมทุจริต (Class 1) ไปในแต่ละรอบ

ในขณะที่แบบจำลอง K-Nearest Neighbors (KNN) ซึ่งมีค่า Recall ต่ำที่สุด (0.41) เนื่องจากเป็นแบบจำลองที่ใช้ระยะห่างของข้อมูลในการจำแนกประเภท (Class) จากเพื่อนบ้านที่ใกล้ที่สุด ซึ่งส่งผลเสียในกรณีที่ข้อมูลไม่สมดุล เนื่องจากข้อมูลธุรกรรมทุจริต (Class 1) มีจำนวนน้อย จึงถูกล้อมรอบด้วยข้อมูลธุรกรรมปกติ (Class 0) ทำให้การจำแนกมีความเอนเอียงและผิดพลาดสูง สำหรับแบบจำลอง Decision Tree (DT) มีแนวโน้มที่จะเกิด Overfitting โดยเฉพาะกับข้อมูลธุรกรรมปกติ (Class 0) ส่งผลให้ต้นไม้เรียนรู้ Pattern จากกลุ่มธุรกรรมปกติ (Class 0) อย่างละเอียดเกินไป ขณะที่กลุ่มข้อมูลทุจริต (Class 1) ซึ่งมีจำนวนน้อย มักไม่ได้รับการพิจารณา

เป็นเงื่อนไขหลักในการแบ่งข้อมูลภายในต้นไม้ เนื่องจากเกณฑ์การตัดสินใจมักถูกชี้นำโดยกลุ่มข้อมูลที่มีจำนวนมากกว่า ส่งผลให้แบบจำลองไม่สามารถเรียนรู้ลักษณะสำคัญของกลุ่มนี้ได้อย่างเพียงพอ ขณะที่แบบจำลอง Random Forest (RF) ซึ่งเป็นการรวมต้นไม้หลายต้นเข้าด้วยกัน สามารถลด Overfitting ได้บางส่วน และช่วยเพิ่มความเสถียรของแบบจำลอง อย่างไรก็ตาม เนื่องจากยังใช้วิธีการสุ่มข้อมูลและสุ่มฟีเจอร์ร่วมกันโดยไม่คำนึงถึงอัตราส่วนของประเภทธุรรรม (Class) ทำให้แบบจำลองยังไม่สามารถแนกประเภทธุรกรรมทุจริตได้ดีมากพอ ส่วนแบบจำลอง CatBoost เป็นแบบจำลอง Boosting ที่พัฒนาขึ้นเพื่อรองรับข้อมูลเชิงหมวดหมู่ (Categorical Features) ได้โดยตรง โดยไม่จำเป็นต้องแปลงข้อมูลล่วงหน้า จุดแข็งของ CatBoost จึงอาจไม่ได้ถูกนำมาใช้เต็มประสิทธิภาพ

จากนั้นจึงนำแบบจำลอง XGBoost ซึ่งมีประสิทธิภาพสูงสุดมาใช้ร่วมกับเทคนิคการจัดการข้อมูลไม่สมดุล พบว่าเทคนิคที่ให้ผลดีที่สุดคือ SMOTETomek และ SMOTE รองลงมาคือ Tomek Links ขณะที่เทคนิค Random Oversampling มีค่าตัวชี้วัด Precision ลดลง เทคนิค SMOTEENN มีค่าตัวชี้วัด Precision และ F1-score ลดลง และเทคนิค ENN มีค่าตัวชี้วัด Recall และ Precision ลดลง

โดยเทคนิค SMOTETomek และ SMOTE มีค่าตัวชี้วัด Recall และ F1-score เพิ่มมากขึ้น โดย Recall สูงถึง 0.91 ซึ่งสะท้อนว่าแบบจำลอง XGBoost สามารถตรวจจับธุรกรรมทุจริตได้อย่างครอบคลุม เนื่องจากเทคนิค SMOTE มีจุดเด่นในการสร้างตัวอย่างสังเคราะห์ของข้อมูลกลุ่มทุจริต (Minority) โดยอิงจากข้อมูลจริงที่อยู่ใกล้กัน ทำให้เพิ่มความหลากหลายของข้อมูลโดยไม่ทำให้เกิดความซ้ำซ้อน ขณะที่เทคนิค SMOTETomek รวมข้อดีของ SMOTE เข้ากับ Tomek Links โดยเทคนิค SMOTE ช่วยในการสังเคราะห์ข้อมูลกลุ่มทุจริตเพิ่ม และเทคนิค Tomek Links ช่วยลบตัวอย่างที่อยู่ใกล้ขอบเขตการจำแนกระหว่างประเภทธุรกรรม (Class) และอาจทำให้โมเดลสับสน จึงช่วยให้โมเดลสามารถเรียนรู้ขอบเขตของกลุ่มธุรกรรมทุจริตได้อย่างชัดเจนและลดการทำนายผิดพลาด

ขณะที่เทคนิค Tomek Links ช่วยเพิ่มค่า Precision และ Recall เนื่องจากการลบข้อมูลของกลุ่มธุรกรรมปกติ (Class 0) ที่อยู่ใกล้กับกลุ่มธุรกรรมทุจริต (Class 1) ช่วยลดการทับซ้อนระหว่างคลาส สำหรับเทคนิค Random Oversampling แม้จะช่วยเพิ่มค่า Recall แต่ค่าตัวชี้วัด Precision ลดลง เนื่องจากแม้จะมีการเพิ่มจำนวนตัวอย่างในกลุ่มธุรกรรมทุจริต แต่ข้อมูลที่ถูกสุ่มเพิ่มขึ้นนั้นเป็นเพียงสำเนาของข้อมูลเดิม จึงทำให้ข้อมูลขาดความหลากหลายในการเรียนรู้ ในขณะที่เทคนิค SMOTEENN ซึ่งเป็นการรวมเทคนิค SMOTE กับเทคนิค Edited Nearest

Neighbors (ENN) แม้จะเพิ่มค่าตัวชี้วัด Recall แต่พบว่าค่า Precision และ F1-score ลดลง อาจเกิดจากการที่เทคนิค ENN ลบตัวอย่างข้อมูลที่แตกต่างจากเพื่อนบ้านออกไป ทำให้โมเดลเรียนรู้ข้อมูลได้ไม่ครบถ้วน และสุดท้ายเทคนิค ENN เมื่อใช้เดี่ยว ๆ พบว่าประสิทธิภาพของแบบจำลองลดลงอย่างชัดเจน ทั้งในค่าตัวชี้วัด Recall และ Precision เนื่องจากการลบตัวอย่างข้อมูลบางตัวออกไป ส่งผลให้แบบจำลองขาดข้อมูลสำคัญที่จำเป็นต่อการเรียนรู้ลักษณะของธุรกรรมทุจริตอย่างหลากหลาย

ดังนั้น เทคนิค SMOTETomek และ SMOTE จึงเป็นเทคนิคที่มีประสิทธิภาพสูงสุดในการเพิ่มความสามารถของแบบจำลอง XGBoost ในการตรวจจับธุรกรรมทุจริตบนชุดข้อมูลที่ไม่สมดุลได้อย่างแม่นยำและครอบคลุมมากขึ้น

3. ข้อเสนอแนะ

ผู้ที่สนใจศึกษาต่อยอดจากการวิจัยนี้ อาจพิจารณาปรับเปลี่ยนแนวทางการทดลองโดยเลือกใช้เทคนิคการถ่วงน้ำหนักของคลาส (Class Weight Adjustment) แทนการใช้เทคนิคการสร้างข้อมูลใหม่ (Resampling) เช่น SMOTE หรือ Oversampling เพื่อให้แบบจำลองสามารถเรียนรู้จากข้อมูลที่ไม่สมดุลได้โดยตรง โดยไม่จำเป็นต้องเพิ่มหรือลบข้อมูลจากชุดฝึก ซึ่งอาจช่วยลดปัญหา Overfitting ที่เกิดจากการสร้างข้อมูลซ้ำหรือข้อมูลสังเคราะห์ อีกทั้งยังช่วยให้กระบวนการฝึกแบบจำลองมีความรวดเร็วและใช้ทรัพยากรน้อยลง

นอกจากนี้ ยังสามารถขยายขอบเขตของการศึกษาไปยังแบบจำลองประเภท Deep Learning เนื่องจากธุรกรรมทางการเงินมักมีลักษณะความต่อเนื่องของพฤติกรรมการใช้จ่าย ซึ่งช่วยให้สามารถตรวจจับรูปแบบของธุรกรรมที่ผิดปกติซึ่งอาจไม่สามารถตรวจพบได้ด้วยแบบจำลอง Machine Learning ทั่วไป อีกทั้งแบบจำลอง Deep Learning ยังมีศักยภาพในการเรียนรู้ลักษณะการเปลี่ยนแปลงของข้อมูลในระยะยาว ซึ่งเหมาะกับสถานการณ์จริงที่ผู้ไม่หวังดีอาจมีพฤติกรรมการทุจริตแบบแฝง หรือมีการปรับเปลี่ยนกลยุทธ์อยู่ตลอดเวลา

บรรณานุกรม

Bauder, R. A., Khoshgoftaar, T. M., & Hasanin, T. (2018). Data Sampling Approaches with Severely Imbalanced Big Data for Medicare Fraud Detection | IEEE Conference Publication | IEEE Xplore. *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. https://doi.org/10.1109/ICTAI.2018.00030

Du, W., Wang, H., Shen, J., Meng, G., Li, H., & Zhou, W. (2024). Insurance Anti-fraud based on FL-WOE Encoding for Vertical Federated Learning. *2024 IEEE International Conference on Big Data (BigData)*. https://doi.org/10.1109/BigData62323.2024.10825724

Gustavo E. A. P. A. Batista, R. C. P., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, *6*(1). https://doi.org/10.1145/1007730.1007735

Hu, Y., Zhang, Y., Huang, T., Hu, Z., Fan, Z., & Li, C. (2021). A Detection Method for Electricity Theft Based on Random Forest Algorithm | IEEE Conference Publication | IEEE Xplore. https://doi.org/10.1109/ICPES51309.2020.9349646

J, K., & Senthilselvi, A. (2022). *Credit Card Fraud Detection based on Ensemble Machine Learning Classifiers* 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC),

Johnson, J. M., & Khoshgoftaar, T. M. (2021). Encoding Techniques for High-Cardinality Features and Ensemble Learners | IEEE Conference Publication | IEEE Xplore. *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*. https://doi.org/10.1109/IRI51335.2021.00055

Lee, R. (2024). Credit Card Transactions Dataset. https://www.kaggle.com/datasets/priyamchoksi/credit-card-transactions-dataset/data

M, H., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01-11. https://doi.org/10.5121/ijdkp.2015.5201

Ma, L. (2024). *Credit Card Fraud Detection Method Based on KRSMOTE+ENN and XGBoost Algorithm* Proceedings of the 5th International Conference on Computer Information and Big Data Applications,

Mahsa Bahrami, Mansour Vali, & Kia, H. (2023). Breast Cancer Detection from Imbalanced Clinical Data: A Comparative Study of Sampling Methods. *2023 30th National and 8th International Iranian Conference on Biomedical Engineering (ICBME)*. https://doi.org/10.1109/ICBME61513.2023.10488624

N.Malini, & student, M. P. (2017). Analysis on credit card fraud identification techniques based on KNN and outlier detection. *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. https://doi.org/10.1109/AEEICB.2017.7972424

Purwar, A. M. (2023). Credit card fraud detection using XGBoost for imbalanced data set. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3607947

Roxana Aldea, M. F., Anca Lazăr (2014). Classifications of motor imagery tasks using k-nearest neighbors. *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*. https://doi.org/10.1109/NEUREL.2014.7011475

Sahu, S., & Sahu, N. (2023). *Analysis of Credit Card Fraud Transaction Detection using Machine Learning Algorithms* 2023 6th International Conference on Contemporary Computing and Informatics (IC3I),

ShuoWang. (2023). Comparison and Analysis of the Effect of XGBoost Classification, BP Neural Network Classification and CatBoost Classification on Malware Attack Prediction. *2023 International Conference on Intelligent Communication and Computer Engineering (ICICCE)*. https://doi.org/10.1109/ICICCE61720.2023.00018

Singh, A., Singh, A., Aggarwal, A., & Chauhan, A. (2022). Design and Implementation of Different Machine Learning Algorithms for Credit Card Fraud Detection. *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. https://doi.org/10.1109/ICECCME55909.2022.9988588

Singh, Y., Singh, K., & Singh Chauhan, V. (2022). *Fraud Detection Techniques for Credit Card Transactions* 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM),

Taloba., A. I., & Ismail., S. S. I. (2019). An Intelligent Hybrid Technique of Decision Tree and Genetic Algorithm for E-Mail Spam Detection | IEEE Conference Publication | IEEE Xplore. https://doi.org/10.1109/ICICIS46948.2019.9014756

Vikas Jain, A. P. a. J. S. B. (2018). Investigation of a Joint Splitting Criteria for Decision Tree Classifier Use of Information Gain and Gini Index. *TENCON 2018 - 2018 IEEE Region 10 Conference*. https://doi.org/10.1109/TENCON.2018.8650485

Yi, X., Xu, Y., Hu, Q., Krishnamoorthy, S., Li, W., Tang, Z., Yi, X., Xu, Y., Hu, Q., Krishnamoorthy, S., Li, W., & Tang, Z. (2022). ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection. *Complex & Intelligent Systems 2022 8:3*, *8*(3). https://doi.org/10.1007/s40747-021-00638-w

ประวัติผู้เขียน