การวิเคราะห์ความเสี่ยงด้านเครดิตด้วยวิธีการเรียนรู้ของเครื่อง
Credit Risk Analysis by Machine Learning

ภควัฒน์ โกมลมาลย์ (Pakkawat Komonman)
ภาควิชาวิศวกรรมข้อมูลขนาดใหญ่ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์
645162020007@dpu.ac.th

บทคัดย่อ

บทความนี้มีวัตถุประสงค์เพื่อทำการวิเคราะห์ข้อมูลและสร้างการเรียนรู้ของเครื่องกับชุดข้อมูล Credit Card Fraud Detection จาก Kaggle.com มีจำนวนแถว 307,511 และ 122 คอลัมน์ ชุดข้อมูลเป็นข้อมูลในอดีตของผู้สมัครขอสินเชื่อบัตรเครดิต และมีคลาสคำตอบเป็นสองกลุ่ม คือ 1 ผู้ถือบัตรมีการชำระหนี้ล่าช้าเกินกว่าวันที่กำหนด และ0 แทนผู้ถือบัตรมีการชำระหนี้ไม่เกินวันที่กำหนด ซึ่งสองคลาสนี้มีจำนวนข้อมูลไม่สมดุลกัน

ธุรกิจการให้อนุมัติสินเชื่อบัตรเครดิต มีเป้าหมายคือลดการผิดนัดชำระหนี้โดยการไม่อนุมัติให้สินเชื่อกับผู้ขอสินเชื่อที่มีความเสี่ยงด้านเครดิต ซึ่งต้องทำการวิเคราะห์ความเสี่ยงด้านเครดิตได้อย่างถูกต้อง รวดเร็ว ดังนั้นการใช้การเรียนรู้ของเครื่องเป็นเครื่องมือที่ดีที่จะสามารถช่วยหาสาเหตุที่เกี่ยวข้องอย่างมีนัยะสำคัญกับความเสี่ยงทางเครดิตได้อย่างเหมาะสม กำจัดความผิดพลาดจากการตัดสินใจอนุมัติเครดิตโดยมนุษย์ และสามารถอธิบายแปรผลของของผลลัพธ์ได้ง่าย

การทำการเรียนรู้ของเครื่อง เริ่มจากการเตรียมข้อมูล โดยทำความเข้าใจความหมายของข้อมูล ทำการเติมค่าที่สูญหาย ทำการสำรวจการกระจายตัวของฟีเจอร์ เพื่อลดความเอนเอียงของโมเดลที่จะทำนายไปในทางข้อมูลที่มีจำนวนมากกว่าหรือข้อมูลที่มีค่าที่มาก ด้วยวิธีการแบ่งช่วงของข้อมูล และทำข้อมูลที่เป็นตัวเลขของแต่ละฟีเจอร์ให้อยู่ในช่วงเดียวกัน แล้วจึงทำข้อมูลในส่วนของคลาสให้สมดุลกัน โดยใช้วิธีการสุ่มลดจำนวนข้อมูลของคลาสหลักที่มีจำนวนมากกว่า ให้เท่ากับจำนวนข้อมูลของคลาสรองที่น้อยกว่า หลังจากนั้นทำการคัดเลือกฟีเจอร์ด้วยวิธีการ FCBF แล้วสร้างโมเดลด้วยวิธีการ Ensemble แบบ Bagging และAdaBoost โดยการเลือกใช้การเรียนรู้ของเครื่องแบบ Gradient Boosted Trees และ Decision Tree เพราะต้องการแปรผลลัพธ์ให้เข้าใจได้ง่าย แล้วทำการปรับค่าพารามิเตอร์ของโมเดล จะเลือกปรับค่า Recall ของคลาส true ที่ต้องการให้ตรวจพบ และค่า f-measure ต้องมีค่ามากตามด้วย แล้ววัดประสิทธิภาพของโมเดลด้วย 10 Fold Cross-Validation ซึ่งได้ผลลัพธ์ได้ Recall คลาส true 71.16% f-measure 22.98% ดังนั้นโมเดลนี้จำเป็นต้องมีการปรับปรุงก่อนการนำไปใช้งานจริง

คำสำคัญ: ความเสี่ยงด้านเครดิต ความไม่สมดุลของคลาส FCBF Ensemble Bagging AdaBoost

Abstract

This paper presents Data Analysis and Machine Learning for Credit Card Fraud Detection dataset from Kaggle.com which it has 307,511 rows and 122 columns. The dataset consists of information from credit card applications and the class consist of imbalanced classes which 1 = Client had late payment more than X days and 0 = Client had not late payment more than X days.

The objectives of credit card business are reducing the default on debt repayment by refusing the applications form whom they have credit risk. Those objectives must be analyzed from the accurate and quick analysis method. So, the machine learning is a decent tool that can analyze the significant reasons for credit risk, eliminate the mistaken credit approval by human and can easily interpret the result.

Machine Learning is started by performing data preparation by data understanding, replacing missing values, exploring data distribution in order to reduce the bias of the model prediction which usually tends to amounts or values of data those can be improved by discretizing, standardizing and balancing uneven class by keeping all of the data in the minority class and decreasing the size of the majority class, then performs features selection by FCBF, then performs Ensemble with Bagging and AdaBoost methods for building machine learning models using Gradient Boosted Trees and Decision Tree because it easily interprets the result, then performs fine tuning parameters based on Recall of true class that needed to be detected and it should has high f-measure score also, then performs the performance validation by 10 Fold-Cross Validation which results Recall of true class = 71.16% , f-measure = 22.98% . So, this model needs an improvement before deployment.

Keyword: Credit Risk, Imbalanced Classes, FCBF, Ensemble, Bagging, AdaBoost.

1. บทนำ

การวิเคราะห์ความเสี่ยงด้านเครดิตด้วยวิธีการเรียนรู้ของเครื่อง เป็นวิธีการที่ให้เครื่องคอมพิวเตอร์ นำข้อมูลในอดีตจากใบสมัครขอสินเชื่อบัตรเครดิต มาหาความสัมพันธ์กับข้อมูลของการผิดนัดชำระหนี้ เพื่อสร้างโมเดลที่ใช้ในการทำนายความเสี่ยงด้านเครดิตของผู้ขอสินเชื่อบัตรเครดิตรายใหม่ เป็นวิธีการที่ธนาคารหลายแห่งเริ่มนำมาใช้ในปัจจุบัน เพราะสามารถนำมาใช้ตัดสินใจการอนุมัติสินเชื่อสำหรับผู้ขอสินเชื่อรายใหม่ ได้อย่างรวดเร็วและถูกต้อง ช่วยลดการถูกผิดนัดชำระหนี้จากผู้ที่มีความเสี่ยงด้านเครดิต และกำจัดความผิดพลาดในการตัดสินใจในขั้นตอนการอนุมัติสินเชื่อของคน ซึ่งบทความนี้จะใช้ชุดข้อมูล Credit Card Fraud Detection จาก Kaggle.com [1] มีจำนวนแถว 307,511 และ 122 คอลัมน์ ประกอบไปด้วยข้อมูลส่วนตัวต่างๆจากการสมัครบัตรเครดิต และมีคลาสคำตอบเป็นสองกลุ่มที่ไม่สมดุลกันคือ 1 แทนการชำระหนี้ล่าช้าเกินกว่าวันที่กำหนด จำนวน 24,825 แถว และ 0 แทนการชำระหนี้ไม่เกินวันที่กำหนด จำนวน 282,686 แถว

2. หลักการพื้นฐาน

2.1 Under Sampling

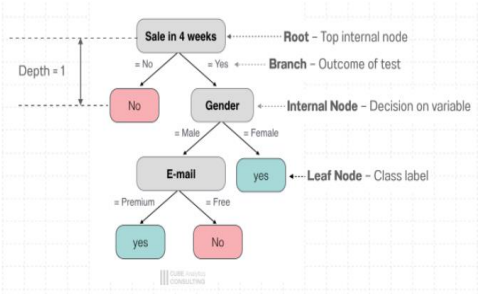
คือการสุ่มลดจำนวนข้อมูลกลุ่มมาก ให้มีจำนวนพอๆ กับข้อมูลกลุ่มน้อย ใช้ในกรณีที่จำนวนข้อมูลของกลุ่มในคลาสไม่สมดุลกัน (Imbalance) เนื่องจาก Classification Algorithms ส่วนใหญ่จะมีแนวโน้มทำนายคำตอบเอนเอียงไปทางกลุ่มของคลาสที่มีจำนวนมากกว่า [2]

2.2 Fast Correlation-Based Filter (FCBF)

เป็นการคัดเลือกฟีเจอร์โดยใช้การคำนวณหาค่าน้ำหนักซึ่งเป็นค่าความสัมพันธ์ระหว่างแต่ละฟีเจอร์และคลาสต่างๆ โดยวิธี FCBF คือการหาความสัมพันธ์แบบ Symmetrical Uncertainty (SU) หรือกล่าวโดยสรุปว่าคือการหาความสัมพันธ์ระหว่างฟีเจอร์ด้วยกันเองและฟีเจอร์กับคลาส โดยที่ฟีเจอร์ใดๆที่มีค่าความสัมพันธ์กับฟีเจอร์ตัวอื่นมากกว่าค่าความสัมพันธ์ระหว่างฟีเจอร์ตัวนั้นนั้นกับคลาสจะถูกตัดอก [3]

2.3 Decision Tree

เป็นโมเดลที่ต้องการจะแยกข้อมูลออกจากกันให้ได้มากที่สุด โดยใช้วิธีการอย่างเช่น Gini Index หรือ Information Gain ทำการเลือกตัวแปรที่แบ่งแยกคลาสคำตอบได้ดีที่สุดมาวางไว้เป็น Node แรก หลังจากนั้นจะหาตัวแปรอื่นๆมาแบ่งข้อมูลในลำดับขั้นต่อไป ซึ่งมีองค์ประกอบดังรูปด้านล่าง [4]



ภาพที่ 1: องค์ประกอบของ Decision Tree

2.4 Gradient Boosted Trees

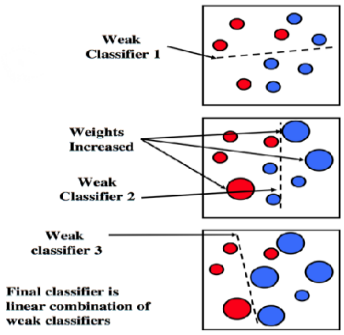
เป็นโมเดลที่มีจุดประสงค์เพื่อลดความ Error ให้ได้มากที่สุด โดยการสร้าง Decision Tree จำนวนหลายๆโมเดลขึ้นมา ซึ่งจะใช้วิธีการเพิ่มโอกาสในการเลือกข้อมูลที่ทำนายผิดในโมเดลในรอบก่อนหน้า มาเป็น training data สำหรับการสร้างโมเดลในรอบถัดไป และทำการ Vote ผลจากทุกโมเดลเป็นผลสรุปของการทำนาย [5]

2.5 Bagging

เป็นการทำ Machine Learning แบบ Ensemble คือการรวมโมเดลที่มีอิสระต่อกันหลายๆโมเดลเข้ามาทำนายผลด้วยกัน โดยวิธี Bagging จะทำการสุ่มข้อมูล subset จากชุดข้อมูล Training Set เดิมออกมาเป็นชุดที่ไม่ซ้ำกัน n ชุด แล้วนำไปสร้างเป็นโมเดลจำนวน n ชุด และใช้การ Vote ทำนายผลจากทุกโมเดลรวมกัน มักใช้ในกรณีที่ต้องการลด overfitting [6]

2.6 AdaBoost (Adaptive Boosting)

เป็นการทำ Machine Learning แบบ Ensemble คือการสร้างโมเดลแบ่งคลาส ซึ่งจะให้ค่าน้ำหนักความสำคัญมากกว่าในจุดที่แบ่งคลาสผิดในแต่ระรอบ เพื่อเพิ่มโอกาสในการนำจุดที่มีความผิดพลาดไปเรียนรู้ในการสร้างโมเดลแบ่งคลาสในรอบถัดไป มักใช้กับการแก้ปัญหาความแม่นยำต่ำ [7]



ภาพที่ 2: การทำงานของ AdaBoost

2.7 Confusion Matrix

คือตารางแสดงความถี่ผลลัพธ์ของการทำนายจากModel ที่เราสร้างขึ้นกับ สิ่งที่เกิดขึ้นจริง

True Positive (TP) คือสิ่งที่ทำนาย ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณีทำนายว่าจริง และสิ่งที่เกิดขึ้น ก็คือจริง

True Negative (TN) คือสิ่งที่ทำนายตรงกับสิ่งที่เกิดขึ้น ในกรณีทำนายว่า ไม่จริง และสิ่งที่เกิดขึ้นก็คือไม่จริง

False Positive (FP) คือสิ่งที่ทำนายไม่ตรงกับสิ่งที่เกิดขึ้น คือทำนายว่าจริง แต่สิ่งที่เกิดขึ้นคือ ไม่จริง

False Negative (FN) คือสิ่งที่ทำนายไม่ตรงกับที่เกิดขึ้นจริง คือทำนายว่าไม่จริง แต่สิ่งที่เกิดขึ้นคือ จริง

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

ภาพที่ 3: Confusion Matrix

2.8 ตัววัดประสิทธิภาพของของโมเดล

Accuracy (ความถูกต้องของโมเดล) คิดจากจำนวนครั้งที่ทำนายถูกหารด้วยจำนวนครั้งที่ทำนายทั้งหมด จาก Confusion Matrix

$$Accuracy = \frac{TPs + TNs}{TPs + TNs + FPs + FNs}$$

Precision (ความแม่นยำในการทำนายของโมเดล) คือสัดส่วนของ การทำนายที่ถูกต้องกับการทำนายทั้งหมดรวมทั้งที่ถูกและผิด

$$Precision\ (positive) = \frac{TPs}{TPs + FPs}$$

$$Precision\ (negative) = \frac{TNs}{TNs + FNs}$$

Recall คือสัดส่วนความถูกต้องของการทำนายว่าจะเป็นจริง กับสิ่งที่เกิดขึ้นจริงในคลาส ใช้ตรวจสอบว่าโมเดลสามารถตรวจจับความจริงของ Training dataset ได้เท่าไรในคลาสนั้นๆ

$$Recall(positive) = \frac{TPs}{TPs + FNs}$$

$$Recal(negative) = \frac{TNs}{TNs + FPs}$$

f-measure คือการวัดประสิทธิภาพของโมเดลโดยใช้ค่าเฉลี่ยของ Precision และ Recall [8]

$$f - measure = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall}$$

ROC (Receiver Operating Characteristics) ใช้แสดงกราฟความสัมพันธ์ระหว่างข้อมูลที่ทำนายถูก (แกน Y) และทำนายผิด (แกน X) โดยที่ ROC Curve ที่มีค่าเข้าใกล้ 1 จะแสดงว่าโมเดลมีประสิทธิภาพดี เนื่องจากมีค่า True Positive เยอะ โดยมี AUC (Area under curve) คือพื้นที่ใต้กราฟ โดยที่ยิ่งมีค่าสูงแสดงว่าโมเดลมีประสิทธิภาพดี [9]

ภาพที่ 4: ROC Graph

3. วิธีการดำเนินงาน

3.1 การเตรียมข้อมูล

ใช้ซอฟต์แวร์ RapidMiner ในการทำในทุกขั้นตอน ขั้นแรกนำเข้าข้อมูลด้วย Operator Retrieve application data แล้วทำการตั้งค่าคอลัมน์ ID

ขั้นตอนแรกเตรียมคอลัมน์คลาส ด้วย Operator Set Role ระบุคอลัมน์ที่เป็นคลาสคำตอบให้เป็น Label แล้วเปลี่ยนชื่อจาก TARGET เป็น Credit_Risk และใช้ Operator Numerical to Binominal แปลงค่า 0, 1 เป็นค่า false และ true เพื่อให้เหมาะสมกับการนำไปใช้กับตัว Machine Learning ที่จะใช้

ขั้นตอนที่สอง ทำการจัดการกับ Missing Values ในข้อมูลชุดนี้ ที่มีจำนวนมากถึงหลักแสนแถว ดังนั้นจึงต้องพยายามทำการหาค่ามาแทนอย่างเหมาะสมที่สุด โดยการทำความเข้าใจข้อมูลแต่ละคอลัมน์ ซึ่งได้พบว่าฟีเจอร์ OWN_CAR_AGE (อายุของรถยนต์) มีความสัมพันธ์กับ FLAG_OWN _ CAR (มีรถยนต์) โดยถ้าฟีเจอร์ FLAG_OWN _ CAR มีค่าเป็น No แล้ว ฟีเจอร์ OWN_CAR_AGE จะไม่มีการระบุค่า ซึ่งข้อมูลต้นทางไม่ได้ระบุว่าอายุเป็นรายเดือนหรือปี ดังนั้นในที่นี้จะอนุมานว่าเป็นรายเดือน เพราะข้อมูลมีค่าน้อยสุดเท่ากับ 0 และมากสุดเท่ากับ 91 ดังนั้นจึงต้องทำการใช้ Operator Generate Attribute ทำการบวก 1 เข้าไปในทุกค่าของข้อมูล จุดประสงค์หลักคือเพื่อให้ค่า 0 กลายเป็น 1 เพื่อที่จะทำการใช้ Operator Replace Missing Values แทนค่า Missing ให้เป็น 0 เพื่อใช้แทนความหมายว่า ไม่มีรถในครอบครอง หลังจากนั้นทำการแทนค่า Missing Values ต่อในส่วนของกลุ่มฟีเจอร์ที่อยู่อาศัย สิ่งอำนวยความสะดวกและอาชีพ ซึ่งกลุ่มที่เป็นตัวเลขจะถูกทำ Normalize จากแหล่งข้อมูลเดิมมาแล้ว มีค่าตั้งแต่ 0 ถึง 1 ดังนั้นจึงทำการแทนค่า Missing ด้วยค่า 0 ส่วนกลุ่มฟีเจอร์ที่เป็น Binomial จะแทน Missing Values ด้วยค่า NO ต่อมาในส่วนของกลุ่มคอลัมน์ EXT_SOURCE ซึ่งต้นทางแหล่งข้อมูลระบุความหมายไว้ว่า เป็นข้อมูลคะแนนที่ Normalize มาแล้วจากภายนอก มีค่าตั้งแต่ 0 ถึง 1 ดังนั้นจึงแทนค่า Missing ด้วยค่า 0 และสุดท้าย ฟีเจอร์อื่นๆที่มีค่า Missing จำนวนไม่เกินหลักหมื่นซึ่งเป็นจำนวนน้อยมากเมื่อเทียบกับข้อมูลทั้งหมด ฟีเจอร์เหล่านี้จะถูกตัดออกไปด้วย operator Declare Missing Value และ Filter Examples สุดท้ายชุดข้อมูลจะมีจำนวนเหลือทั้งสิ้น 263,947 แถว 120 คอลัมน์

ขั้นที่สอง ทำจัดการสำรวจการกระจายตัวของข้อมูล เพื่อไม่ให้โมเดลที่ได้มีการทำนายผลเอนเอียงไปทางข้อมูลที่มีจำนวนมากหรือมีค่าตัวเลขที่มาก โดยการใช้ Operator Discretize แบ่งกลุ่มข้อมูลเพื่อลดช่วงของข้อมูลในฟีเจอร์กลุ่มที่มีการกระจายตัวกว้างและเบ้ซ้ายหรือขวา แยกข้อมูลเป็นช่วงจำนวน 5 bin แล้วใช้ Operator Normalize แปลงข้อมูลฟีเจอร์กลุ่มที่เป็นตัวเลข ที่มีการกระจายตัวปกติ แบบ Range Transformation ให้อยู่ในช่วง 0 ถึง 1

3.2 การคัดเลือกตัวแปร

ใช้ Operator FCBF และ Operator Select by Weights แล้วหาค่า k หรือจำนวนตัวแปรที่เหมาะสมด้วย Operator Optimize Parameters (Grid) ดังจะกล่าวในขั้นตอนการสร้างโมเดล

3.3 การแก้ปัญหา Imbalanced Data

เนื่องจากจำนวนข้อมูลของคลาสรอง true ซึ่งเป็นสิ่งที่สนใจ มีจำนวนน้อยกว่าคลาสหลัก false แต่ก็มีข้อมูลมากพอที่จะนำไปใช้ เพราะมีอยู่ 20,448 แถว จึงเลือกใช้ Operator Under-Sampling สุ่มลดจำนวนคลาสคำตอบหลักคือกลุ่มคำตอบ false ให้เหลือเท่าคลาสคำตอบรองคือกลุ่มคำตอบ true

3.4 การสร้างโมเดล

การเลือกใช้โมเดลจะพิจารณาจากความเหมาะสมกับข้อมูลแบบที่มีคลาสไม่สมดุลกัน ความรวดเร็ว และความง่ายต่อการแปรความ เพราะการทำนายความเสี่ยงด้านเครดิตอาจจะต้องมีการอธิบายผู้ขอสินเชื่อหรือผู้ที่เกี่ยวข้องว่าทำไมถึงอนุมัติหรือไม่อนุมัติสินเชื่อ และการวัดประสิทธิภาพของโมเดล จะเลือกให้น้ำหนักความสำคัญกับค่า Recall คลาส true คือต้องการหาผู้ที่มีความเสี่ยงด้านเครดิตให้ได้มาก และดู f-score เป็นอันดับรองลงมา ดังนั้นจากเหตุผลดังที่กล่าวมาข้างต้น จึงเลือกใช้การเรียนรู้ของเครื่องวิธี Gradient Boosted Trees และ Decision Tree โดยทำการทดลองเลือกใช้วิธีการสร้างแบบ Ensemble สามเทคนิคดังต่อไปนี้

เทคนิคที่ 1 Ensemble แบบ AdaBoost คัดเลือกตัวแปรแบบ FCBF ได้ผลลัพธ์ในส่วนของ Recall คลาส true ที่ดีมากคือ 83.10% Precision คลาส true 11.07% และ f-measure 19.58% +/- 0.59%

เทคนิคที่ 2 Ensemble แบบ Bagging คัดเลือกตัวแปรแบบ FCBF ได้ผลลัพธ์ในส่วนของ f-measure 24.36% +/- 0.41% Recall คลาส true 66.28% และ Precision คลาส true 14.92%

เทคนิคที่ 3 จากการที่เทคนิคที่ 1 ให้ค่า Recall คลาส true ที่มาก และ เทคนิคที่ 2 ให้ค่า f-measure ที่มากกว่า ดังนั้นจึงใช้วิธีการรวมทั้ง 2 เทคนิค แล้วเพิ่มโมเดล Decision Tree เข้าไปอีก และใช้ Operator Vote สุดท้ายจะได้ผลลัพธ์ของประสิทธิภาพโดยรวมสมดุลกัน แล้วเลือกเทคนิคที่ 3 ไปใช้ปรับหาค่า Parameters ที่เหมาะสมในการสร้างโมเดล โดยใช้วิธี Trial and Error ร่วมกับการใช้ Operator Optimize Parameters (Grid) ซึ่งมีตัวพารามิเตอร์หลักที่ใช้ปรับและได้ผลลัพธ์ที่เหมาะสมตามตารางด้านล่าง

ตารางที่ 1: ตารางแสดงการปรับหาค่า Parameters

วิธี	Major Parameters
1	FCBF = 23 features AdaBoost 500 iterations Gradient Boosted Trees (Trees = 11, Maximal Depth = 5, Min Rows = 8, Learning Rate = 0.5, Sample Rate = 0.543)
2	FCBF = 23 features Bagging 10 iterations Gradient Boosted Trees (Trees = 11, Maximal Depth = 5, Min Rows = 8, Learning Rate = 0.5, Sample Rate = 0.543)
3	FCBF = 30 features Decision Tree (Criterion = gini_index, Maximal Depth = 10)

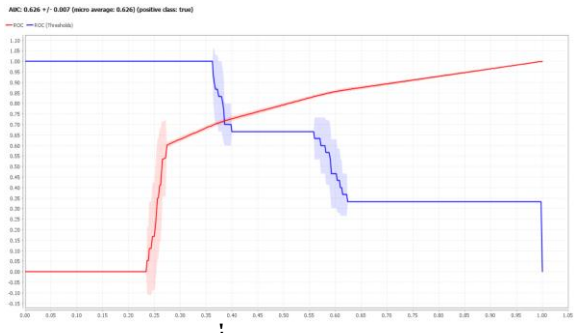
3.5 การทดสอบประสิทธิภาพของโมเดล

ใช้วิธี 10 Fold Cross-Validation แบ่งข้อมูลออกเป็น N = 10 ชุด ข้อมูล N-1 = 9 ชุดใช้สร้างโมเดล ส่วนที่เหลือ 1 ชุด จะใช้เป็นตัวทดสอบประสิทธิภาพของโมเดล แล้วสุ่มสลับชุดข้อมูลสร้างและชุดทดสอบโมเดลเป็นจำนวน N = 10 รอบ โดยใช้การสุ่มแบ่งข้อมูลแบบ Stratified Sampling คือการสุ่มโดยให้แต่ละชุดข้อมูลที่แบ่งมามีสัดส่วนของคลาสเหมือนกันกับข้อมูลชุดเดิม โดยใช้ตัววัดประสิทธิภาพ Accuracy, Precision, Recall, F-Measure, ROC และ AUC

4. ผลการวิเคราะห์ข้อมูล

จากวิธี 10 Fold Cross-Validation ได้ผลลัพธ์ของตัววัดประสิทธิภาพความถูกต้อง 63.04% +/- 1.00% ความแม่นยำการทำนายคลาส false 96.26% การตรวจจับคลาส false 62.36% ความแม่นยำการทำนายคลาส true 13.70%

การตรวจจับคลาส true 71.16% f-measure 22.98% +/- 0.31% AUC = 0.626 +/- 0.007 และ ROC ดังภาพด้านล่าง



ภาพที่ 4: ผล ROC Graph

5. สรุปผลการวิเคราะห์ข้อมูล

เนื่องจากทางผู้จัดทำต้องการให้ความสำคัญกับการหาสิ่งที่สนใจคือผู้มีความเสี่ยงทางเครดิตให้มีค่าสูง โมเดลที่ทำออกมาจึงสามารถทำการตรวจหาสิ่งที่สนใจคือผู้มีความเสี่ยงทางเครดิตหรือคลาส true ได้มาก คือมีค่า Recall ประมาณ 71% โดยเลือก ใช้โมเดลรูปแบบ Decision Tree เพื่อสามารถแปรผลให้มนุษย์เข้าใจได้ แต่ Precision หรือความแม่นยำของคลาส true มีค่าต่ำเพียง 13.70% และอีกทั้ง f-measure มีค่าต่ำเพียงแค่ 22.98% โมเดลนี้อาจนำไปตรวจจับผู้มีความเสี่ยงทางเครดิตได้ดี แต่ด้วยความแม่นยำของการทำนายที่ต่ำ จะทำให้เกิดการปฏิเสธผิด หรือปฏิเสธผู้ขอสินเชื่อที่ไม่มีความเสี่ยงด้านเครดิตได้มาก ดังนั้นควรนำชุดข้อมูลมาทำการวิเคราะห์ใหม่ ตั้งแต่ขั้นตอนการนำเข้าข้อมูล ซึ่งชุดข้อมูลอาจมีฟีเจอร์ที่ไม่แข็งแรงพอ ดังนั้นการนำไปใช้จริง ผู้เป็นเจ้าของข้อมูลควรทำการปรับปรุงเพิ่มเติมในส่วนของฟีเจอร์ เช่นการหาและถามคำถามใหม่ในใบสมัครสินเชื่อเป็นต้น นอกจากนี้ยังพบว่าฟีเจอร์ในกลุ่มที่ชื่อว่า EXT_SOURCE ซึ่งแหล่งข้อมูลระบุว่าเป็นข้อมูลจากภายนอก ไม่ได้ระบุความหมายชัดเจน เป็นฟีเจอร์ที่มีความสำคัญสูงซึ่งการสร้างโมเดลแต่ระรอบ ฟีเจอร์นี้มักจะอยู่บนรากหรือระดับบนของ Decision Tree เสมอ ดังนั้นควรมีการทำ Feature Engineering หรือวิเคราะห์กลุ่มฟีเจอร์นี้ออกมาพัฒนาก่อนแล้วทำการวิเคราะห์ข้อมูลสร้างโมเดลใหม่

เอกสารอ้างอิง

[1] Mishra5001, “Credit Card Fraud Detection,” *https://www.kaggle.com/mishra5001/credit-card*, 25 December 2021.

[2] ดร.เอกสิทธิ์ พัชรวงศ์ศักดา, “Imbalanced Data,” *เอกสารการสอนวิชาการวิเคราะห์ข้อมูลขนาดใหญ่ ภาควิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่ มหาวิทยาลัยธุรกิจบัณฑิตย์* ปีที่ 7 หน้า 4-40.

[3] ดร.เอกสิทธิ์ พัชรวงศ์ศักดา, “Attribute (Feature) Selection,” *เอกสารการสอนวิชาการวิเคราะห์ข้อมูลขนาดใหญ่ ภาควิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่ มหาวิทยาลัยธุรกิจบัณฑิตย์* ปีที่ 7 หน้า 157-164.

[4] ดร.เอกสิทธิ์ พัชรวงศ์ศักดา, “Fundamentals of Machine Learning,” *เอกสารการสอนวิชาการวิเคราะห์ข้อมูลขนาดใหญ่ ภาควิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่ มหาวิทยาลัยธุรกิจบัณฑิตย์* ปีที่ 7 หน้า 63-102.

[5] ดร.เอกสิทธิ์ พัชรวงศ์ศักดา, “Advanced Classification Techniques,” *เอกสารการสอนวิชาการวิเคราะห์ข้อมูลขนาดใหญ่ ภาควิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่ มหาวิทยาลัยธุรกิจบัณฑิตย์* ปีที่ 7 หน้า 23-27.

[6] TITIPATA, “[ML]Bagging หรือ Boosting คืออะไร ทำงานอย่างไร?,” *https://tupleblog.github.io/bagging-boosting*, 25 ธันวาคม 2564.

[7] Sirawich Jaichuen, “AdaBoost Algorithm,” *https://sirawichjaichuen.medium.com/adaboost-algorithm-cfe6b58e60fa*, 25 ธันวาคม 2564.

[8] Pagon Gatchalee, “Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ในMachine learning,” *https://medium.com/@pagongatchalee/confusion-matrix -เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย-ในMachine-learning- fba6e3f9508c*, 25 ธันวาคม 2564.

[9] ดร.เอกสิทธิ์ พัชรวงศ์ศักดา, “Fundamentals of Machine Learning,” *เอกสารการสอนวิชาการวิเคราะห์ข้อมูลขนาดใหญ่ ภาควิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่ มหาวิทยาลัยธุรกิจบัณฑิตย์* ปีที่ 7 หน้า 49-60.