การวิเคราะห์เชิงเปรียบเทียบของอัลกอริทึมการเรียนรู้ของเครื่องสำหรับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคาร

COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR PREDICTING DEPOSIT ACCOUNT CHURN OF BANK CUSTOMER

กฤตภาส โสมากุล
KITTAPAS SOMAKUN

การศึกษาอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์
ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2568

KMITL-2025-SC-M-017-052

COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR PREDICTING DEPOSIT ACCOUNT CHURN OF BANK CUSTOMER

KITTAPAS SOMAKUN

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN DATA SCIENCE AND ANALYTICS

KMITL DIGITAL ANALYTICS AND INTELLIGENCE CENTER SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2025

KMITL-2025-SC-M-017-052

หัวข้อการค้นคว้าอิสระ การวิเคราะห์เชิงเปรียบเทียบของอัลกอริทึมการเรียนรู้ของเครื่องสำหรับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคาร
นักศึกษา นาย กฤตภาส โสมากุล
รหัสประจำตัว 66056007
ปริญญา วิทยาศาสตรมหาบัณฑิต
ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
สาขาวิชา วิทยาการข้อมูลและการวิเคราะห์
พ.ศ. 2568
อาจารย์ที่ปรึกษาการค้นคว้าอิสระ รองศาสตราจารย์ ดร.ณชญาดา กมลมิธิชม

บทคัดย่อ

การยกเลิกบริการของลูกค้าเป็นปัญหาสำคัญที่ส่งผลกระทบต่อความยั่งยืนของธนาคาร เนื่องจากการสูญเสียลูกค้าที่มีค่าอาจทำให้รายได้ของธนาคารลดลงและมีผลต่อการเติบโตในระยะยาว การทำนายการยกเลิกบริการของลูกค้าธนาคารจึงเป็นเครื่องมือที่มีความสำคัญในการช่วยให้ธนาคารสามารถคาดการณ์ลูกค้าที่มีแนวโน้มจะยกเลิกบริการได้ล่วงหน้า และดำเนินการเพื่อรักษาลูกค้าเหล่านั้น

การศึกษาอิสระนี้มุ่งเน้นการพัฒนาโมเดลการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารโดยประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่องที่หลากหลาย เช่น Logistic Regression, AdaBoost, CatBoost, Gradient Boosting, Random Forest และ XGBoost เพื่อตรวจสอบประสิทธิภาพในการทำนายการยกเลิกบริการ โดยใช้ข้อมูลลูกค้ารวมถึงข้อมูลประชากรศาสตร์และพฤติกรรมการใช้บริการต่าง ๆ ของลูกค้า

จากการทดสอบและประเมินผลโมเดล พบว่า AdaBoost สามารถให้ผลการทำนายที่มีความแม่นยำสูงสุดในด้าน Accuracy, Precision, Recall และ Precision-Recall Curves ตามด้วย Logistic Regression ในขณะที่ Random Forest, Gradient Boosting และ CatBoost สามารถให้ผลการทำนายที่ดีในบางกรณี แต่มีความแม่นยำน้อยกว่า โดยโมเดลที่มีประสิทธิภาพสูงที่สุดสามารถช่วยธนาคารในการระบุลูกค้าที่มีแนวโน้มจะยกเลิกบริการได้ ซึ่งสามารถใช้ในการออกแคมเปญการรักษาลูกค้าหรือโปรโมชันต่าง ๆ

การวิจัยนี้เสนอแนวทางในการนำผลการทำนายการยกเลิกบริการมาใช้ในการปรับกลยุทธ์ทางการตลาดและการบริการลูกค้า ที่จะช่วยให้ธนาคารสามารถรักษาฐานลูกค้ารายเดิมและลดการสูญเสียลูกค้าได้อย่างมีประสิทธิภาพ

คำสำคัญ : การทำนายการยกเลิกบริการของลูกค้า ธนาคาร การเรียนรู้ของเครื่อง การสุ่มป่าไม้ อะแดปทีฟบูสท์ เอ็กซ์จีบูสท์ การรักษาลูกค้า

Independent Study	Comparative Analysis of Machine Learning Algorithms for Predicting Deposit Accout Churn of Bank Customer
Student	Mr. Kittapas Somakun
Student ID.	66056007
Degree	Master of Science
	Kmitl Digital Analytics and Intelligence Center
Program	Data Science and Analytics
Year	2025
Independent Advisor	Assoc. Prof. Dr. Nachayadar Kamolmitisom

ABSTRACT

Customer churn is a significant issue that affects the sustainability of banks, as losing valuable customers can reduce revenue and impact long-term growth. Predicting customer churn in banking is, therefore, an essential tool that enables banks to anticipate customers who are likely to leave and take proactive measures to retain them.

This independent study focuses on developing a predictive model for customer churn in banking using various machine learning techniques, including Logistic Regression, AdaBoost, CatBoost, Gradient Boosting, Random Forest, and XGBoost. The study evaluates the effectiveness of these models in predicting customer churn based on customer data, including demographic information and service usage behavior.

Through testing and evaluation, the results indicate that AdaBoost provides the highest accuracy in terms of Accuracy, Precision, Recall and Precision-Recall Curves, followed by Logistic Regression. While Random Forest, Gradient Boosting, and CatBoost also perform well in certain cases, they exhibit lower overall accuracy. The most effective model can help banks identify customers at risk of churning, enabling them to implement targeted retention campaigns and promotional offers.

This research proposes a strategy for utilizing churn prediction results to refine marketing and customer service strategies, allowing banks to maintain their existing customer base and reduce churn effectively.

Keywords: Churn Prediction, Banking, Machine Learning, Random Forest, AdaBoost, XGBoost, Customer Retention

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณอาจารย์ที่ปรึกษา รศ.ดร.ณชญาดา กมลมิธิชม ที่ให้การสนับสนุนและคำแนะนำอย่างดีเยี่ยมตลอดระยะเวลาการทำวิจัยนี้ อาจารย์ได้ชี้แนะแนวทางการศึกษาและการพัฒนาโมเดลการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารที่เป็นส่วนสำคัญในการทำให้การวิจัยนี้สำเร็จลุล่วงไปได้

ขอขอบพระคุณผู้สนับสนุนทุกท่านที่ให้ข้อมูลและแหล่งข้อมูลที่จำเป็นสำหรับการวิจัยนี้ โดยเฉพาะ นส.รินทร์ลภัส จินานุศิลปสาท ที่ให้คำแนะนำและการปรึกษาในการสร้างโมเดลและการวิเคราะห์ข้อมูล

ขอบคุณเพื่อนร่วมงานและเพื่อนนักศึกษาทุกท่านที่ร่วมให้คำแนะนำและสนับสนุนทางด้านความคิดและการแก้ปัญหาต่าง ๆ ในระหว่างการดำเนินงานวิจัยนี้

นอกจากนี้ ขอบคุณครอบครัวของข้าพเจ้าที่ให้กำลังใจและสนับสนุนในทุก ๆ ด้านตลอดระยะเวลาที่ทำการวิจัยนี้

สุดท้าย ข้าพเจ้าขอขอบคุณทุกท่านที่ได้ช่วยเหลือในด้านต่าง ๆ เพื่อให้งานวิจัยนี้ประสบผลสำเร็จตามวัตถุประสงค์และเป้าหมายที่ตั้งไว้

ขอขอบคุณอย่างยิ่ง

กฤตภาส โสมากุล

สารบัญ

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันธนาคารและสถาบันการเงินต้องเผชิญกับความท้าทายในการรักษาลูกค้า เนื่องจากการแข่งขันที่รุนแรงและตัวเลือกทางการเงินที่หลากหลาย ลูกค้าอาจเลือกเปลี่ยนไปใช้บริการของธนาคารคู่แข่งได้ตลอดเวลา ปัญหาการยกเลิกบริการของลูกค้า (Customer Churn) เป็นหนึ่งในปัจจัยสำคัญที่ส่งผลกระทบต่อรายได้และความมั่นคงขององค์กร การลดอัตราการยกเลิกบริการไม่เพียงแต่ช่วยรักษาฐานลูกค้าเดิม แต่ยังช่วยลดต้นทุนในการหาลูกค้าใหม่ ที่มีค่าใช้จ่ายสูงกว่าการรักษาลูกค้าเก่าหลายเท่า

เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ได้รับความสนใจมากขึ้นในการพัฒนาโมเดลเพื่อทำนายความเป็นไปได้ที่ลูกค้าจะยกเลิกบริการ โดยใช้ข้อมูลพฤติกรรมลูกค้า เช่น ประวัติการทำธุรกรรม การใช้ผลิตภัณฑ์ทางการเงิน และปัจจัยประชากรศาสตร์ การนำอัลกอริทึมการเรียนรู้ของเครื่องมาประยุกต์ใช้สามารถช่วยให้ธนาคารสามารถวิเคราะห์แนวโน้มการยกเลิกบริการของลูกค้าได้อย่างแม่นยำ พร้อมทั้งสามารถดำเนินมาตรการเชิงรุกเพื่อรักษาลูกค้าได้อย่างมีประสิทธิภาพ

อย่างไรก็ตาม มีอัลกอริทึมการเรียนรู้ของเครื่องหลากหลายรูปแบบที่สามารถนำมาทำนายการยกเลิกบริการของลูกค้า เช่น Logistic Regression, Random Forest, Gradient Boosting, CatBoost, AdaBoost และ XGBoost แต่ละอัลกอริทึมมีข้อดีและข้อเสียที่แตกต่างกันในแง่ของความแม่นยำ เวลาในการประมวลผล และความสามารถในการตีความผลลัพธ์ ดังนั้นจึงจำเป็นต้องทำการศึกษาเชิงเปรียบเทียบเพื่อระบุอัลกอริทึมที่เหมาะสมสำหรับการทำนายปัญหานี้

การวิเคราะห์เชิงเปรียบเทียบของอัลกอริทึมการเรียนรู้ของเครื่องสำหรับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารจะช่วยให้เข้าใจถึงประสิทธิภาพของแต่ละวิธี รวมถึงการเลือกใช้คุณลักษณะของข้อมูลที่เหมาะสม ที่จะเป็นแนวทางสำคัญในการพัฒนาระบบแจ้งเตือนล่วงหน้า และออกแบบกลยุทธ์เพื่อรักษาลูกค้าได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของการวิจัย

1) เพื่อศึกษาปัจจัยต่าง ๆ ที่ส่งผลต่อการยกเลิกบริการของลูกค้าธนาคาร

2) เพื่อพัฒนาและทดสอบโมเดลการทำนายการยกเลิกบริการของลูกค้าธนาคารโดยประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่อง เช่น Logistic Regression, Random Forest, CatBoost, AdaBoost, Gradient Boosting และ XGBoost

3) เพื่อประเมินประสิทธิภาพของโมเดลการทำนายการยกเลิกบริการด้วยมาตรวัดต่าง ๆ เช่น Accuracy, Precision, Recall, F1-Score และ Precision-Recall Curves

4) เพื่อเสนอแนวทางในการใช้ผลการทำนายในการดำเนินกลยุทธ์การรักษาลูกค้าในธนาคาร

1.3 ขอบเขตของการวิจัย

การวิจัยนี้มุ่งเน้นไปที่การวิเคราะห์พฤติกรรมของลูกค้าธนาคารเพื่อทำนายแนวโน้มการยกเลิกบริการที่เป็นปัญหาสำคัญของอุตสาหกรรมธนาคาร การศึกษานี้ใช้ข้อมูลลูกค้าและปัจจัยที่

ส่งผลต่อการตัดสินใจยกเลิกบริการ เพื่อนำไปสู่การพัฒนาแนวทางป้องกันและรักษาลูกค้าได้อย่างมีประสิทธิภาพ

1.3.1 การรวบรวมข้อมูล

1) ใช้ชุดข้อมูลจากธนาคารที่มีข้อมูลลูกค้า เช่น วันที่เปิดบัญชี สถานะบัญชี อายุปัจจุบัน เพศ อาชีพ แหล่งที่มาของรายได้ ช่วงของรายได้ ระยะเวลาการเป็นลูกค้า ระบบมือถือที่ใช้ การเข้าร่วมโปรโมชัน จำนวนที่ทำรายการวันที่ฝากเงินครั้งแรก เป็นต้น

2) ข้อมูลที่ได้รับมา ได้มาจากสถาบันทางการเงินในระยะเวลา 4 ปีย้อนหลัง มีจำนวน 627,470 ข้อมูล เก็บข้อมูลวันที่ 21 กุมภาพันธ์ 2568

3) การทำความสะอาดข้อมูล เช่น การจัดการค่าที่ขาดหายไป และการแปลงข้อมูลที่ไม่ใช่ตัวเลขให้อยู่ในรูปแบบที่เหมาะสมด้วยวิธีการ One-Hot encoding

1.3.2 อัลกอริทึมที่นำมาเปรียบเทียบ

1) Logistic Regression
2) Random Forest
3) CatBoost
4) AdaBoost
5) Gradient Boosting
6) XGBoost

1.3.3 เกณฑ์การเปรียบเทียบประสิทธิภาพ

1) ค่า Accuracy
2) ค่า Precision
3) ค่า Recall
4) ค่า F1-score
5) ค่า Precision-Recall Curves

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 เพิ่มความแม่นยำในการคาดการณ์

1) การเปรียบเทียบอัลกอริทึมช่วยให้สามารถเลือกโมเดลที่มีประสิทธิภาพสูงในการคาดการณ์การยกเลิกบริการของลูกค้า

2) ลดข้อผิดพลาดในการทำนาย ทำให้ธนาคารสามารถดำเนินการเชิงรุกเพื่อลดอัตราการสูญเสียลูกค้า

1.4.2 การพัฒนากลยุทธ์การรักษาลูกค้า

1) ช่วยให้ธนาคารสามารถออกแบบมาตรการเชิงป้องกัน เช่น โปรแกรมความภักดี โปรโมชัน หรือบริการเสริมที่ตรงกับความต้องการของลูกค้า

2) เพิ่มความพึงพอใจของลูกค้าและเสริมสร้างความสัมพันธ์ระยะยาว

1.4.3 การวิเคราะห์เชิงลึกเกี่ยวกับปัจจัยที่ส่งผลต่อการยกเลิกบริการ

1) ช่วยให้เข้าใจปัจจัยสำคัญที่ทำให้ลูกค้าตัดสินใจยกเลิกบริการ เช่น ค่าธรรมเนียม อัตราดอกเบี้ย หรือคุณภาพการให้บริการ

2) สนับสนุนการตัดสินใจของธนาคารในการพัฒนากลยุทธ์เพื่อลดอัตราการยกเลิก

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การทำนายการยกเลิกบริการ

การทำนายลูกค้าที่มีแนวโน้มจะยกเลิกบริการที่ใช้ เช่น ยกเลิกการใช้บริการของธนาคาร ในอนาคตการทำนายการยกเลิกบริการนี้สามารถทำได้โดยการใช้ข้อมูลจากลูกค้ารายเดิม เช่น วันที่เปิดบัญชี สถานะบัญชี อายุปัจจุบัน เพศ อาชีพ แหล่งที่มาของรายได้ ช่วงของรายได้ ระยะเวลาการเป็นลูกค้า ระบบมือถือที่ใช้ การเข้าร่วมโปรโมชัน จำนวนที่ทำรายการวันที่ฝากเงินครั้งแรก ที่จะช่วยให้ธนาคารสามารถหาลูกค้าที่มีความเสี่ยงที่จะยกเลิกบริการพร้อมทั้งสามารถพัฒนากลยุทธ์ในการรักษาลูกค้าได้

2.1.2 การเรียนรู้ของเครื่อง และการจำแนกประเภท

การเรียนรู้ของเครื่องเป็นแขนงหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence - AI) ที่ช่วยให้ระบบสามารถเรียนรู้จากข้อมูลและทำการคาดการณ์หรือจำแนกประเภทได้โดยไม่ต้องเขียนกฎเกณฑ์ที่ตายตัว (Mitchell, 1997) โดยอัลกอริทึมที่ใช้ในงานวิจัยนี้สามารถจัดอยู่ในกลุ่มของ Supervised Learning ที่เป็นการเรียนรู้แบบมีผู้สอน โดยใช้ข้อมูลที่มีป้ายกำกับ (Labeled Data) ในการฝึกโมเดล

2.1.3 การวิเคราะห์อัตราการยกเลิกบริการของลูกค้า

อัตราการยกเลิกบริการเป็นตัวชี้วัดซึ่งใช้สำหรับประเมินว่าลูกค้าออกจากบริการขององค์กรในช่วงเวลาที่กำหนดหรือไม่ ที่มีทฤษฎีสำคัญที่เกี่ยวข้อง ได้แก่:

1) ทฤษฎี CLV (Customer Lifetime Value) ซึ่งใช้คำนวณมูลค่าของลูกค้าตลอดช่วงเวลาที่ใช้บริการ

2) ทฤษฎี Switching Cost ที่อธิบายต้นทุนที่ลูกค้าต้องเผชิญเมื่อเปลี่ยนไปใช้บริการของคู่แข่ง ที่มีผลต่อการตัดสินใจยกเลิกบริการ

2.1.4 Pearson Correlation

สหสัมพันธ์เพียร์สัน (Pearson Correlation) เป็นสถิติที่ใช้วัดความสัมพันธ์เชิงเส้นระหว่างตัวแปรสองตัว โดยค่าสหสัมพันธ์ที่คำนวณได้จะอยู่ในช่วง -1 ถึง 1 ที่แสดงถึงระดับความสัมพันธ์ระหว่างตัวแปรดังกล่าว ซึ่ง 0 หมายถึง ตัวแปรไม่มีความสัมพันธ์กัน ค่าใกล้ 1 แสดงว่าตัวแปรมีความสัมพันธ์กันมาก ค่าใกล้ -1 แสดงว่าตัวแปรมีความสัมพันธ์กันมากในทิศทางตรงกันข้าม

สูตรการคำนวณ Pearson Correlation Coefficient (r) คือ

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{[n\sum x_i^2 - (\sum x_i)^2][n\sum y_i^2 - (\sum y_i)^2]}} \quad \textbf{(2.1)}$$

โดยที่

x_i และ y_i คือ ค่าของตัวแปร x และ y ในแต่ละจุดข้อมูล

n คือ จำนวนข้อมูล

การใช้ Pearson Correlation สามารถช่วยให้เราทราบได้ว่าความสัมพันธ์ระหว่างสองตัวแปรมีความแข็งแรงแค่ไหน และความสัมพันธ์นั้นเป็นบวกหรือลบ

2.1.5 Spearman's Rank Correlation

สหสัมพันธ์สเปียร์แมน (Spearman's Rank Correlation) เป็นสถิติที่ใช้วัดความสัมพันธ์ระหว่างตัวแปรสองตัว โดยพิจารณาอันดับของข้อมูลแทนค่าจริง เหมาะสำหรับข้อมูลที่ไม่ได้มีการกระจายแบบปกติ หรือมีความสัมพันธ์ที่ไม่เป็นเชิงเส้น (Non-Linear Relationship)

สูตรการคำนวณ Spearman's Rank Correlation คือ

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \quad (2.2)$$

โดยที่

d_i คือ ผลต่างระหว่างอันดับของคู่ข้อมูลแต่ละคู่ (อันดับของตัวแปร x กับอันดับของตัวแปร y)

n คือ จำนวนข้อมูล

2.1.6 SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) เป็นเทคนิคที่ใช้สำหรับการปรับสมดุลข้อมูลที่มีการกระจายของคลาสไม่เท่ากัน โดยเฉพาะเมื่อที่คลาสหนึ่งมีจำนวนข้อมูลน้อยกว่าอีกคลาสหนึ่งอย่างมีนัยสำคัญ (Class Imbalance) หลักการของ SMOTE คือการสร้างตัวอย่างข้อมูลสังเคราะห์ในคลาสที่มีจำนวนน้อยกว่า แทนที่จะทำการสุ่มซ้ำข้อมูลเดิม วิธีนี้ช่วยเพิ่มจำนวนข้อมูลในคลาสที่มีน้อย โดยไม่ทำให้เกิดปัญหาการเรียนรู้แบบซ้ำซ้อน (Overfitting)

ขั้นตอนการทำงานของ SMOTE

1. เลือกตัวอย่างจากกลุ่มที่มีจำนวนน้อยแบบสุ่ม
2. หา K-Nearest Neighbors (KNN) ของตัวอย่างที่เลือก
3. เลือกเพื่อนบ้านหนึ่งตัวอย่างแบบสุ่ม
4. สร้างตัวอย่างใหม่โดยคำนวณ จุดกลางระหว่างตัวอย่างที่เลือกกับเพื่อนบ้าน ตามสูตร

$$X_{new} = X_{old} + \lambda\left(X_{neighbor} - X_{old}\right) \quad (2.3)$$

โดยที่ λ เป็นค่าที่สุ่มระหว่าง 0 และ 1

2.1.7 Principal Component Analysis

PCA (Principal Component Analysis) เป็นเทคนิคซึ่งใช้สำหรับ ลดมิติของข้อมูล (Dimensionality Reduction) โดยการแปลงข้อมูลให้เป็นชุดของตัวแปรใหม่ที่เรียกว่า องค์ประกอบหลัก (Principal Components) ซึ่งสามารถอธิบายความแปรปรวนของข้อมูลเดิมได้มากที่สุด

ขั้นตอนการคำนวณ PCA

1) ปรับสเกลข้อมูล (Standardization)

2) คำนวณ Covariance Matrix

3) หาค่า Eigenvalues และ Eigenvectors

4) เลือก Principal Components ตามค่าความแปรปรวนสะสม

5) แปลงข้อมูลไปยังแกนใหม่

2.1.7.1 การหาค่าเฉลี่ยและการปรับสเกลข้อมูล (Mean Centering & Standardization)

ก่อนใช้ PCA จำเป็นต้องทำให้ข้อมูลมีค่าเฉลี่ยเป็นศูนย์เพื่อให้การคำนวณมีประสิทธิภาพมากขึ้น

$$X_{centered} = X - \bar{X} \tag{2.4}$$

เมื่อ $\bar{X}$ คือค่าเฉลี่ยของข้อมูล

หากข้อมูลมีช่วงค่าที่แตกต่างกัน อาจใช้การปรับสเกล (Standardization) โดยทำให้ข้อมูลมีค่าเฉลี่ยเป็นศูนย์และมีส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1

$$X_{scaled} = \frac{X - \bar{X}}{\sigma} \tag{2.5}$$

เมื่อ σ คือค่าความแปรปรวน

2.1.7.2 Covariance Matrix

Covariance เป็นค่าที่ใช้วัด ความสัมพันธ์เชิงเส้น ระหว่างตัวแปร

$$Cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right) \tag{2.6}$$

เมื่อ n คือจำนวนข้อมูล

เมื่อนำไปใช้กับข้อมูลหลายมิติ จะได้ Covariance Matrix (C) ที่เป็นเมทริกซ์ขนาด d x d ที่แสดงความสัมพันธ์ระหว่างฟีเจอร์ สูตรคือ

$$C = \frac{1}{n} X^T X \quad (2.7)$$

เมื่อ X^T คือเมทริกซ์สลับเปลี่ยนของ X

2.1.7.3 Eigenvalues และ Eigenvectors

PCA คำนวณ Eigenvalues และ Eigenvectors ของ Covariance Matrix เพื่อนำไปใช้สร้าง Principal Components

$$Cv = \lambda v \quad (2.8)$$

โดยที่

C = Covariance Matrix

v = Eigenvector

λ = Eigenvalue

Eigenvectors กำหนด ทิศทางของ Principal Components

Eigenvalues กำหนด ปริมาณของความแปรปรวนที่อธิบายได้โดย Principal Components

2.1.7.4 การเลือก Principal Components (PC)

เลือก Eigenvectors ที่มี Eigenvalues สูงที่สุด เป็น Principal Components (PC) ตัวอย่างเช่น

PC1 มี Eigenvalue สูงสุด (อธิบายความแปรปรวนได้มากที่สุด)

PC2 มี Eigenvalue รองลงมา และตั้งฉากกับ PC1

ให้เลือกเฉพาะ k Principal Components ที่สามารถอธิบายความแปรปรวนสะสมได้ตามที่ต้องการ สูตรคือ

$$Explained\ Variance\ Ratio = \frac{\lambda_i}{\sum \lambda} \quad (2.9)$$

2.1.7.5 Singular Value Decomposition (SVD)

อีกวิธีหนึ่งในการคำนวณ PCA คือใช้ SVD จากสูตร

$$X = U \Sigma V^T \quad (2.10)$$

โดยที่

U เป็นเมทริกซ์ของ Eigenvectors

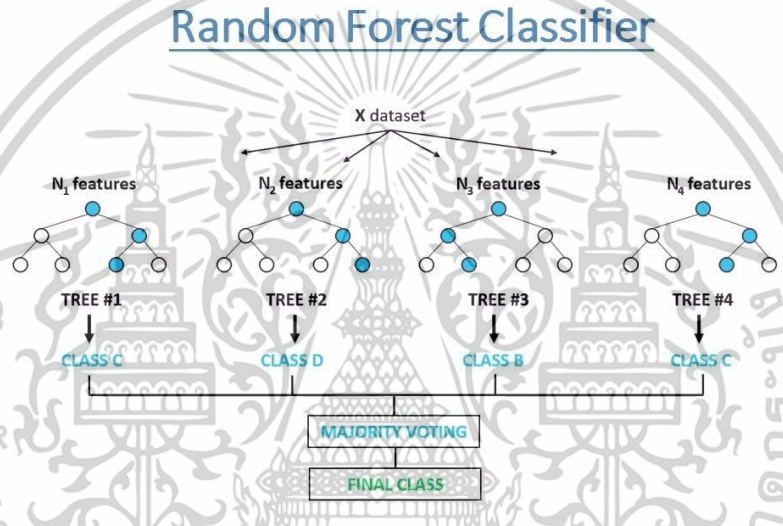
Σ เป็นเมทริกซ์ของ Eigenvalues

V^T เป็นเมทริกซ์ของเวกเตอร์ฐาน

SVD มักถูกใช้แทนการหาค่า Eigenvectors โดยตรงเพราะมีประสิทธิภาพสูงกว่าเมื่อข้อมูลมีขนาดใหญ่

2.1.8 Random Forest

เป็นอัลกอริทึมการเรียนรู้ของเครื่องที่พัฒนาขึ้นจากแนวคิดของ Decision Tree และ Bagging เพื่อเพิ่มความแม่นยำและลดการเกิด Overfitting เริ่มจาก Leo Breiman นักสถิติและนักวิทยาการข้อมูลชื่อดัง ได้พัฒนาแนวคิด Bagging (Bootstrap Aggregating) ในปี 1996 หลังจากนั้นในปี 2001 Breiman ได้พัฒนา Random Forest ขึ้นโดยขยายแนวคิดของ Bagging และเพิ่มการสุ่ม (Random Subspace Method) ดังแสดงในรูปที่ 2.1

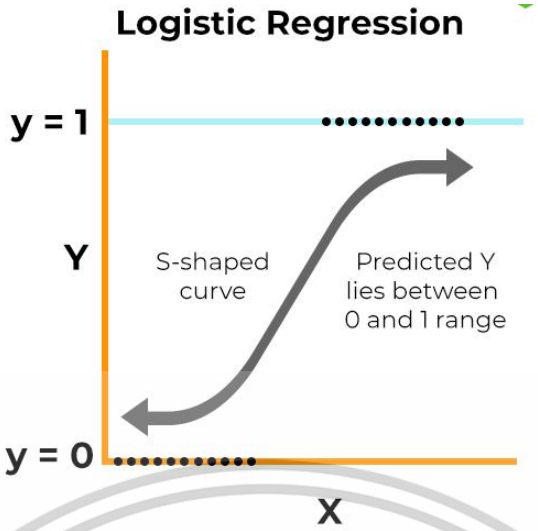


รูปที่ 2.1 รูปการทำงานของ Random Forest

รูปที่ 2.1 แสดงให้เห็นถึงการทำงานของ Random Forest Classifier ซึ่งเป็นอัลกอริทึมการจำแนกประเภทที่ใช้ต้นไม้การตัดสินใจหลายต้น TREE #1, TREE #2, TREE #3 และ TREE #4 ในการจำแนกข้อมูลจากชุดข้อมูล X แต่ละต้นไม้จะใช้คุณลักษณะ N1, N2, N3 และ N4 ตามลำดับ และสร้างคลาส CLASS C, CLASS D, CLASS B, CLASS C การจำแนกขั้นสุดท้ายจะถูกกำหนดโดยการลงคะแนนเสียงส่วนใหญ่จากคลาสที่คาดการณ์โดยแต่ละต้นไม้ ส่งผลให้ได้ FINAL CLASS.

2.1.9 Logistic Regression

ในปี 1838 แนวคิดของ ฟังก์ชันโลจิสติก ถูกพัฒนาขึ้นโดย Pierre François Verhulst นักคณิตศาสตร์ชาวเบลเยียมโดย Verhulst ใช้ฟังก์ชันนี้เพื่อสร้าง Logistic Growth Model ซึ่งใช้ในการศึกษา การเติบโตของประชากร ในปี 1920 - 1930 (Joseph Berkson, 1944) เป็นคนแรกที่นำเสนอ Logit Model และใช้ Logistic Regression ในงานด้านชีวสถิติ

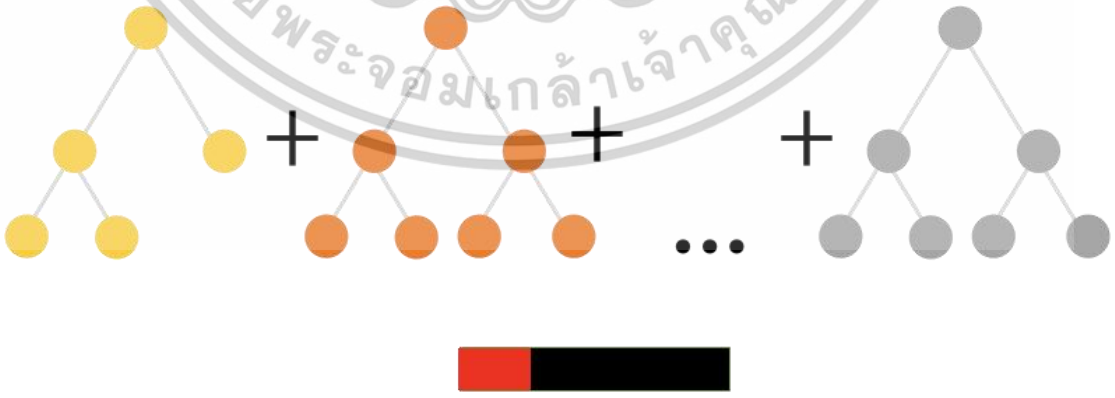


รูปที่ 2.2 รูปการแบ่งกลุ่มของ Logistic Regression

รูปที่ 2.2 กราฟแสดงถึงโมเดล Logistic Regression ซึ่งเป็นเทคนิคทางสถิติที่ใช้ในการจำแนกประเภทข้อมูล โดยเส้นโค้งรูปตัว S บนกราฟแสดงถึงความสัมพันธ์ระหว่างตัวแปร X และความน่าจะเป็นของผลลัพธ์ Y ที่อยู่ในช่วงระหว่าง 0 ถึง 1

2.1.10 CatBoost

Yandex ที่เป็นบริษัทที่เชี่ยวชาญด้าน Search Engine และ AI ได้พัฒนาอัลกอริทึม Gradient Boosting สำหรับใช้ภายใน พบว่าปัญหาสำคัญของอัลกอริทึมทั่วไปคือการจัดการข้อมูลประเภท Categorical (เช่น ข้อมูลข้อความ, หมวดหมู่) อัลกอริทึมที่มีอยู่ เช่น XGBoost และ LightGBM ต้องใช้การ One-hot Encoding หรือ Label Encoding ซึ่งอาจทำให้สูญเสียข้อมูลสำคัญ ในปี 2017 Yandex เปิดตัว CatBoost อย่างเป็นทางการ เป็นอัลกอริทึม Gradient Boosting ที่มีการปรับปรุงการจัดการข้อมูล Categorical โดยใช้ Ordered Target Statistics และ Model Oblivious Trees



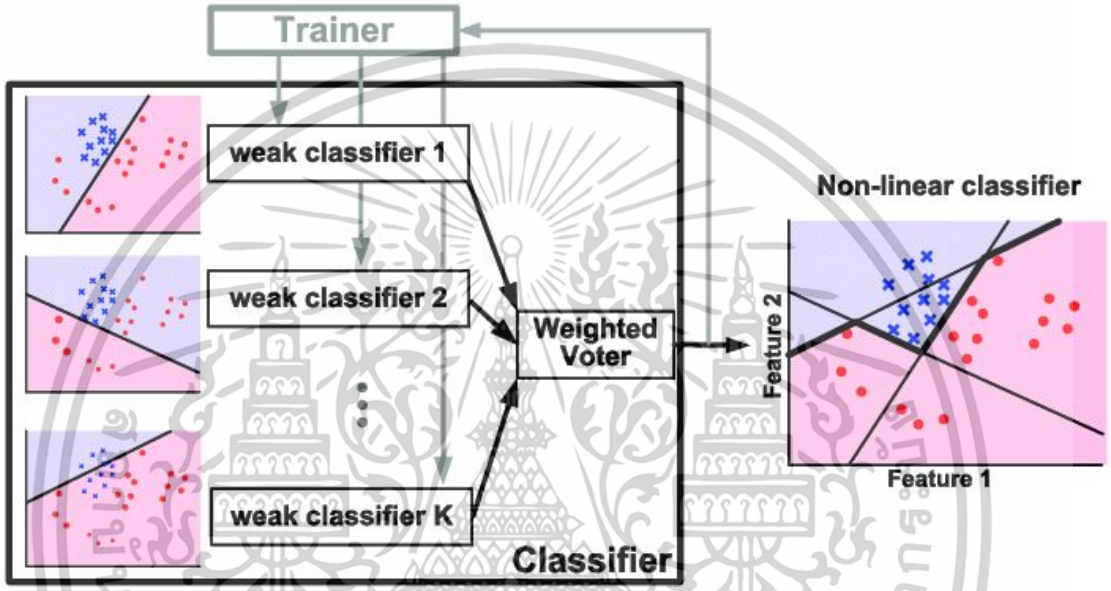
รูปที่ 2.3 รูปการแบ่งกลุ่มของ CatBoost

รูปที่ 2.3 แสดงการเพิ่มต้นไม้หลายต้น ทีละต้นจนถึงต้นไม้ที่ N โดยแถบสีดำและสีแดงที่อยู่ใต้ต้นไม้แต่ละต้นแสดงถึงข้อผิดพลาดที่เกิดขึ้นในแต่ละขั้นตอนของการสร้างต้นไม้

กระบวนการนี้ช่วยให้เห็นว่าข้อผิดพลาดลดลงอย่างไรเมื่อมีการเพิ่มต้นไม้ใหม่ในแต่ละขั้นตอนของการเรียนรู้

2.1.11 AdaBoost

ในปี 1995 Yoav Freund และ Robert Schapire สองนักวิทยาการคอมพิวเตอร์ ได้นำเสนอแนวคิดของ AdaBoost ในงานวิจัยชื่อ "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting" ซึ่ง Schapire เคยแสดงให้เห็นว่า Boosting สามารถเปลี่ยนตัวจำแนกแบบอ่อนแอให้กลายเป็นตัวจำแนกที่ทรงพลังได้



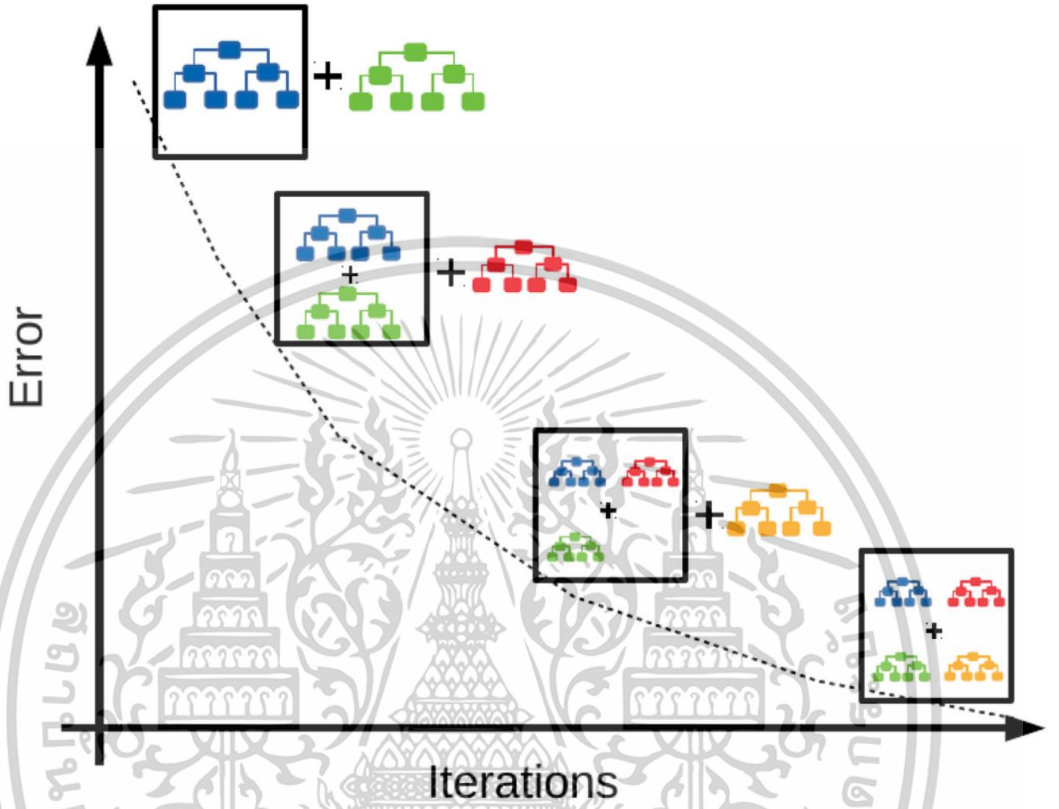
รูปที่ 2.4 รูปการทำงานของ AdaBoost

รูปที่ 2.4 แสดงการทำงานของ AdaBoost ซึ่งเป็นอัลกอริทึมการเรียนรู้ของเครื่องที่ใช้ในการรวมตัวจำแนกที่อ่อนแอหลายตัว (weak classifiers) ให้เป็นตัวจำแนกที่แข็งแกร่ง (strong classifier) หนึ่งตัว โดยมีตัวฝึก (trainer) ที่ป้อนข้อมูลเข้าไปยังตัวจำแนกที่อ่อนแอหลายตัว (weak classifier 1, weak classifier 2 และ weak classifier K) ตัวจำแนกที่อ่อนแอเหล่านี้จะถูกนำมารวมกันโดยใช้การลงคะแนนแบบถ่วงน้ำหนัก (weighted voter) และจะมีการปรับน้ำหนักของข้อมูลที่ถูกจำแนกผิดพลาดในแต่ละรอบการฝึกฝน เพื่อให้โมเดลในรอบถัดไปให้ความสำคัญกับข้อมูลเหล่านี้มากขึ้น เพื่อสร้างตัวจำแนกสุดท้าย (final classifier) ในด้านขวาของภาพแสดงกราฟตัวจำแนกแบบไม่เชิงเส้น (non-linear classifier) โดยมี Feature 1 และ Feature 2 ซึ่งเครื่องหมายกากบาทสีน้ำเงินและวงกลมสีแดงแทนคลาสที่แตกต่างกัน

2.1.12 Gradient Boosting

Jerome H. Friedman นักสถิติชื่อดังจากมหาวิทยาลัยสแตนฟอร์ด ได้พัฒนา Gradient Boosting และตีพิมพ์งานวิจัยชื่อ "Greedy Function Approximation: A

Gradient Boosting Machine" แทนที่จะใช้ Weight Adjustments แบบ AdaBoost, Friedman ใช้ Gradient Descent เพื่อลดค่าความผิดพลาดของโมเดล

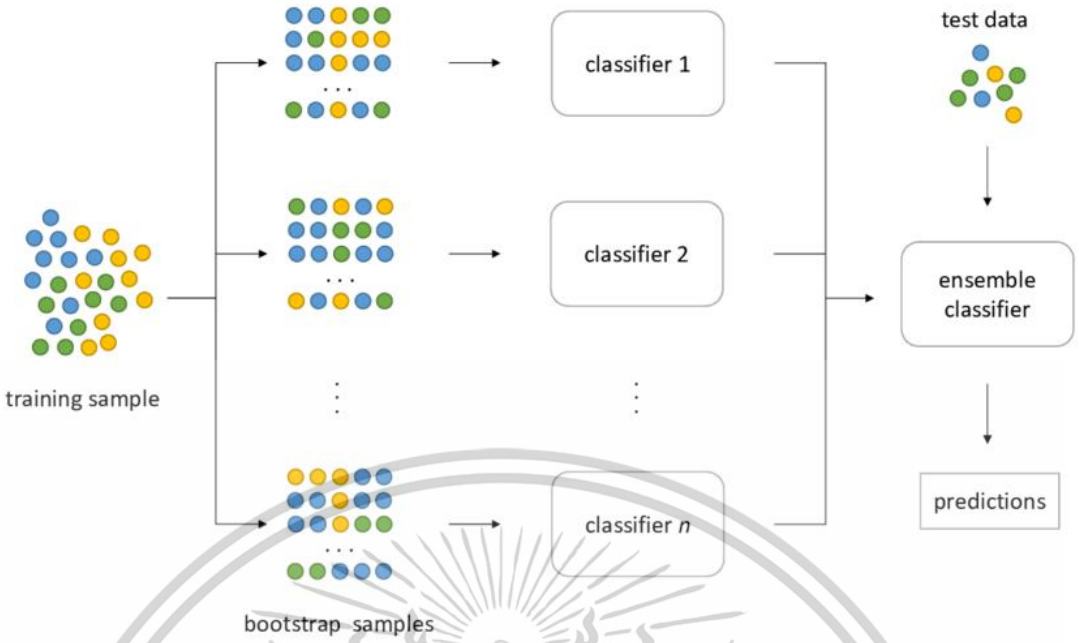


รูปที่ 2.5 รูปการทำงานของ Gradient Boosting

รูปที่ 2.5 แสดงการทำงานของ Gradient Boosting โดยที่แกน x คือค่า Iterations แสดงถึงจำนวนครั้งที่ทำการปรับปรุงโมเดล และแกน y คือค่า Error หรือค่าความผิดพลาด จากกราฟแสดงให้เห็นว่าค่าความผิดพลาดลดลงเมื่อจำนวน Iterations เพิ่มขึ้น โดยมีการเพิ่มต้นไม้ (tree) ที่มีสีต่าง ๆ (น้ำเงิน, แดง, เขียว, เหลือง) เข้าด้วยกันในแต่ละขั้นตอนเพื่อช่วยลดค่าความผิดพลาด

2.1.13 XGBoost

ปัญหาหลักของ Gradient Boosting แบบดั้งเดิมคือ ช้าและกินทรัพยากรสูง โดยเฉพาะกับข้อมูลขนาดใหญ่ Tianqi Chen นักวิจัยจากมหาวิทยาลัยวอชิงตันจึงได้พัฒนา XGBoost ที่เป็นส่วนหนึ่งของงานวิจัยระดับปริญญาเอกของเขา เป้าหมายหลักของ XGBoost คือทำให้ Gradient Boosting เร็วขึ้น, ใช้หน่วยความจำน้อยลง และรองรับการประมวลผลแบบขนาน ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 รูปการทำงานของ XGBoost

รูปที่ 2.6 แสดงการทำงานของ XGBoost เริ่มต้นด้วยชุดข้อมูลการฝึกอบรมที่ประกอบด้วยจุดสีต่าง ๆ ซึ่งเป็นข้อมูลที่ใช้ในการสร้างแบบจำลอง จะถูกแบ่งออกเป็น Bootstrap Samples โดยแต่ละ Bootstrap Samples จะมีตัวจำแนก (Classifier) เป็นของตัวเอง ซึ่งเป็นแบบจำลองการทำนายที่แตกต่างกัน และผลลัพธ์จากตัวจำแนกหลายตัวจะถูกนำมารวมกันเพื่อสร้างตัวจำแนกรวม (Ensemble Classifier) ซึ่งเป็นการรวมผลลัพธ์จากหลายแบบจำลองเพื่อเพิ่มความแม่นยำในการทำนาย

2.1.14 Accuracy

Accuracy คือค่าที่ใช้วัดความถูกต้องของโมเดลในการจำแนกข้อมูลทั้งหมด คำนวณจากสัดส่วนของจำนวนการทำนายที่ถูกต้องต่อจำนวนตัวอย่างทั้งหมด

สูตรของ Accuracy คือ

$$Accuracy = \frac{(TP + TN)}{(TP + FN + FP + TN)} \quad (2.11)$$

โดยที่

TP (True Positive) จำนวนตัวอย่างที่เป็นบวกจริง และโมเดลทำนายว่าบวก

TN (True Negative) จำนวนตัวอย่างที่เป็นลบจริง และโมเดลทำนายว่าลบ

FP (False Positive) จำนวนตัวอย่างที่เป็นลบจริง แต่โมเดลทำนายว่าบวก

FN (False Negative) จำนวนตัวอย่างที่เป็นบวกจริง แต่โมเดลทำนายว่าลบ

โดยสรุปแล้ว Accuracy คืออัตราส่วนของจำนวนกรณีที่ทำนายถูกต้อง (ทั้งบวกและลบ) ต่อจำนวนการทำนายทั้งหมด ค่า TP, TN, FP และ FN ได้มาจาก Confusion Matrix ตัวอย่างของ Confusion Matrix แสดงอยู่ในตาราง 2.1

ตารางที่ 2.1 ตาราง Confusion Matrix

		ข้อมูลจริง	
		Positive	Negative
ข้อมูลที่ทำนาย	Positive	TP	FP
	Negative	FN	TN

2.1.15 Precision

Precision วัดความแม่นยำของการทำนายที่เป็นบวก โดยคำนวณจากสัดส่วนของค่าทรูพอซิทีฟ (True Positive) ต่อจำนวนทั้งหมดของกรณีที่ถูกทำนายว่าเป็นบวก สามารถคำนวณได้ตามสูตร

$$Precision = \frac{TP}{TP + FP} \quad (2.12)$$

Precision บ่งบอกว่ามีจำนวนกรณีที่ถูกทำนายว่าเป็นบวกว่ามีกี่กรณีที่ถูกจัดประเภทอย่างถูกต้อง

2.1.16 Recall

Recall (หรือที่เรียกว่า Sensitivity หรือ True Positive Rate) เป็นค่าที่ใช้วัดความสามารถของโมเดลในการระบุข้อมูลที่เป็นบวก (Positive) ได้อย่างถูกต้อง

สูตรของ Recall คือ

$$Recall = \frac{TP}{TP + FN} \quad (2.13)$$

ค่า recall นี้ช่วยในการวัดโมเดลว่าสามารถระบุกรณีที่เป็นบวกได้ดีเพียงใด และมีประโยชน์อย่างยิ่งในกรณีที่การพลาดข้อมูลที่เป็นบวก (False Negative) มีผลกระทบมาก เช่น การตรวจโรคหรือการค้นหาข้อมูลที่สำคัญ

2.1.17 F1-Score

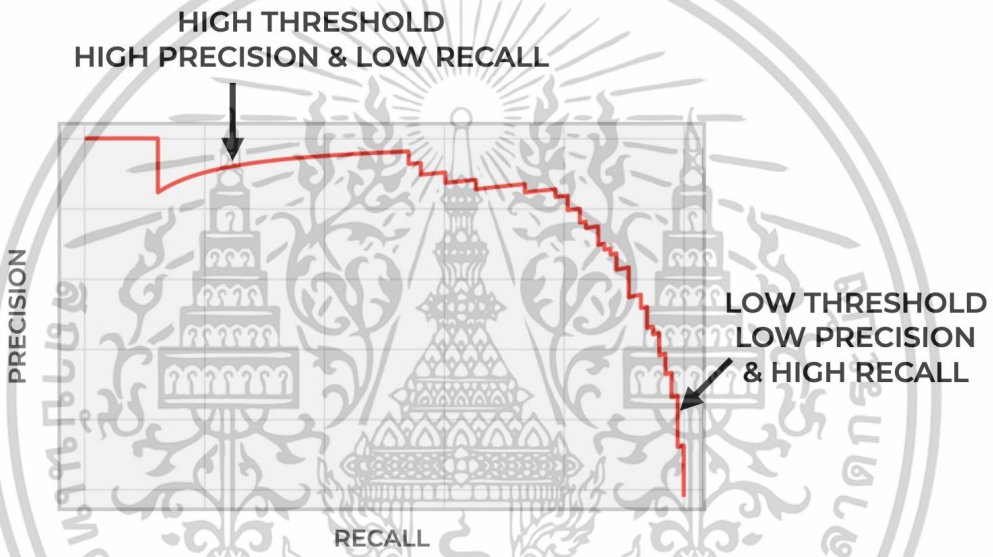
ค่า F1-score คือค่าเฉลี่ยฮาร์มอนิกระหว่าง Precision และ Recall ที่ช่วยให้เกิดความสมดุลระหว่างทั้งสองค่าดังกล่าว โดยเฉพาะอย่างยิ่งเมื่อชุดข้อมูลมีความไม่สมดุล สูตรของ F1-score คือ

$$F1 = 2 \times (\frac{Precision \times Recall}{Precision \times Recall}) \quad (2.14)$$

F1-score ช่วยในการประเมินโมเดลเมื่อทั้งค่า Precision และ Recall มีความสำคัญเท่าเทียมกัน โดยทั่วไปแล้ว ค่า F1-score ที่สูงกว่า 0.7 ถือว่าเป็นค่าที่ยอมรับได้ อย่างไรก็ตาม ค่าที่เหมาะสมขึ้นอยู่กับบริบทและความต้องการของงานที่ทำ เช่น งานที่ต้องการความแม่นยำสูงหรืองานด้านการแพทย์อาจต้องการค่า F1-score ที่สูงกว่า 0.8

2.1.18 Precision-Recall Curves

Precision-Recall Curves เป็นกราฟที่ใช้แสดงความสัมพันธ์ระหว่าง Precision (ความแม่นยำ) และ Recall (ความครอบคลุม) ของโมเดล Classification โดยเฉพาะในปัญหาที่มีความไม่สมดุลของข้อมูล (Imbalanced Data) ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 รูปการตัวอย่างของ Precision-Recall Curves

โดยที่

แกน X คือค่า Recall

แกน Y คือค่า Precision

แต่ละจุดในกราฟแสดงให้เห็นว่าค่า Precision และ Recall ว่าเปลี่ยนไปอย่างไรซึ่งความสัมพันธ์ของ 2 ค่านี้เรียกว่า Threshold

โดยทั่วไป Threshold ต่ำ หมายถึง Recall สูง แต่ถ้า Precision ต่ำ หมายถึงโมเดลรับรู้ Positive ได้เยอะขึ้นแต่ผิดพลาดบ่อย ในขณะที่ Threshold สูง จะทำให้ Precision สูง แต่ Recall ต่ำ โมเดลแม่นยำขึ้นแต่จับ Positive ได้ไม่ครบ

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยในด้านการทำนายการยกเลิกบริการของลูกค้าธนาคาร

หลายงานวิจัยได้ทำการศึกษาเกี่ยวกับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารตัวอย่างเช่น:

1) Li & Lee (2016) ได้ทำการศึกษาการใช้เทคนิค Random Forest ในการทำนายการยกเลิกบริการของลูกค้าในธุรกิจธนาคาร โดยพบว่า Random Forest สามารถให้ผลการทำนายที่มีความแม่นยำสูงเมื่อเทียบกับโมเดลอื่น ๆ

2) Verbeke et al. (2012) ได้ศึกษาการใช้ Decision Trees และ Logistic Regression ในการทำนายการยกเลิกบริการของลูกค้าในสถาบันการเงิน โดยเน้นที่การใช้ข้อมูลทางประชากรศาสตร์ของลูกค้าและพฤติกรรมการใช้บริการ

3) Nguyen & Simkin (2013) ได้สำรวจปัจจัยที่มีผลต่อการยกเลิกบริการของลูกค้า โดยเน้นที่การบริการลูกค้าที่มีคุณภาพต่ำ และการนำเสนอโปรแกรมรางวัลและสิทธิพิเศษเพื่อลดการยกเลิกบริการ

2.2.2 งานวิจัยที่ใช้การเรียนรู้ของเครื่องในการทำนายการเลิกใช้บริการ

การประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่องได้รับความสนใจอย่างมากในการทำนายการยกเลิกบริการ ตัวอย่างของงานวิจัยในด้านนี้ได้แก่:

1) Zhou et al. (2017) ใช้เทคนิค Support Vector Machines (SVM) และ Neural Networks ในการทำนายการยกเลิกบริการของลูกค้าผ่านข้อมูลทางการเงินและพฤติกรรมการใช้บริการ โดยสามารถทำนายการยกเลิกบริการได้อย่างมีประสิทธิภาพ

2) Chaurasia & Pal (2014) ใช้ Naive Bayes และ Random Forest ในการทำนาย Churn โดยการใช้ข้อมูลจากฐานข้อมูลลูกค้าเพื่อศึกษาพฤติกรรมการยกเลิกบริการ และพบว่า Random Forest มีประสิทธิภาพสูงในการทำนาย

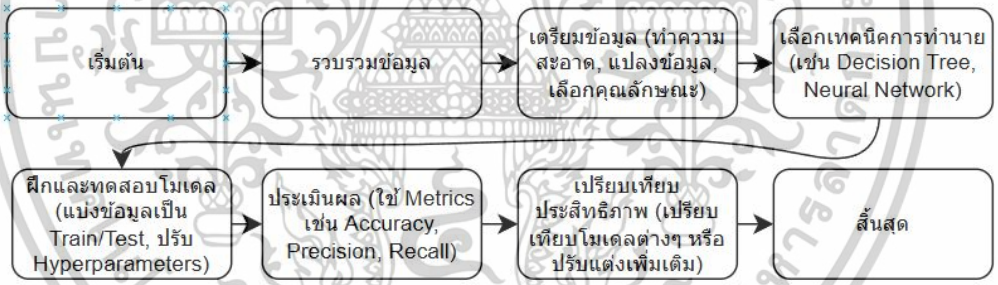
3) Rafael et al. (2018) เปรียบเทียบหลายโมเดล Machine Learning ในการทำนายการยกเลิกบริการของลูกค้า โดยใช้ทั้ง Logistic Regression, Decision Trees และ XGBoost และพบว่า XGBoost ให้ผลลัพธ์ที่ดีที่สุดในด้านความแม่นยำ

บทที่ 3
วิธีการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงวิธีการซึ่งใช้สำหรับเปรียบเทียบประสิทธิภาพของอัลกอริทึมการเรียนรู้ของเครื่องในการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคาร ในปัจจุบันธนาคารมีข้อมูลบัญชีของลูกค้าอยู่ทั้งหมด 627,470 บัญชี มีบัญชีที่ปิดไปแล้วทั้งหมด 10,764 บัญชี มีเงื่อนไขในการสมัคร คือ ต้องเป็นบุคคลธรรมดา สัญชาติไทย อายุ 15-70 ปีบริบูรณ์ โดยมีขั้นตอนการดำเนินการวิจัยดังต่อไปนี้

3.1 การออกแบบการวิจัย

การวิจัยในหัวข้อ “การวิเคราะห์เชิงเปรียบเทียบของอัลกอริทึมการเรียนรู้ของเครื่องสำหรับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคาร” มุ่งเน้นไปที่การทำนายลูกค้าที่จะยกเลิกบริการของธนาคารโดยการใช้ข้อมูลที่ระบบของธนาคารจัดเก็บไว้ เช่น วันที่เปิดบัญชี สถานะบัญชี อายุปัจจุบัน เพศ อาชีพ แหล่งที่มาของรายได้ ช่วงของรายได้ ระยะเวลาการเป็นลูกค้า ระบบมือถือที่ใช้ การเข้าร่วมโปรโมชัน จำนวนที่ทำรายการ วันที่ฝากเงินครั้งแรก โดยการออกแบบการวิจัยในที่นี้จะใช้เทคนิคการวิเคราะห์ข้อมูลเชิงพาณิชย์และการทำนายโดยใช้การเรียนรู้ของเครื่องเพื่อช่วยให้สามารถทำนายการยกเลิกบริการได้แม่นยำ ซึ่งขั้นตอนการดำเนินงานได้แสดงในรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินงาน

รูปที่ 3.1 แสดงขั้นตอนการดำเนินงานในกระบวนการวิเคราะห์ข้อมูล

โดยที่

เริ่มต้น คือ เริ่มต้นกระบวนการ
รวบรวมข้อมูล คือ รวบรวมข้อมูลที่จำเป็น
เตรียมข้อมูล คือ ทำความสะอาดข้อมูล, แปลงข้อมูล, เลือกคุณลักษณะ
เลือกเทคนิคการนำมาใช้ คือ เช่น Decision Tree, Neural Network
ศึกษาและทดสอบโมเดล คือ แบ่งชุดข้อมูล Train/Test, ปรับ Hyperparameters
ประเมินผล คือ ใช้ Metrics เช่น Accuracy, Precision, Recall
เปรียบเทียบประสิทธิภาพ คือ เปรียบเทียบโมเดลต่าง ๆ หรือปรับแต่งเพิ่มเติม

สิ้นสุด คือ จบกระบวนการ

3.2 การรวบรวมข้อมูล

การรวบรวมข้อมูลถือเป็นขั้นตอนแรกที่สำคัญในกระบวนการวิจัย โดยข้อมูลซึ่งใช้สำหรับทำนายการยกเลิกบริการของลูกค้าธนาคารในระยะเวลา 4 ปีประกอบไปด้วย

3.2.1 ข้อมูลส่วนบุคคล

1) อายุ
2) เพศ
3) อาชีพ
4) แหล่งที่มารายได้
5) ช่วงรายได้
6) สถานภาพสมรส
7) ยี่ห้อมือถือ
8) การยืนยันอีเมล

3.2.2 ข้อมูลทางการเงิน

1) จำนวนครั้งการโอนในปี 2024
2) จำนวนเงินโอนในปี 2024
3) ยอดเงินฝากคงเหลือ
4) ยอดเงินฝากครั้งแรก
5) วันที่เอาเงินฝากครั้งแรก

3.2.3 ข้อมูลการใช้บริการ

1) วันที่เปิดบัญชี
2) การยืนยันอีเมล
3) ช่องทางยืนยันตัวตน
4) การใช้งานผลิตภัณฑ์อื่น
5) วันที่เข้าใช้งานล่าสุด
6) การร่วมโปรโมชันจับฉลาก
7) การร่วมโปรโมชัน 9.9
8) การร่วมโปรโมชัน งานออนไลน์

3.2.4 ข้อมูลประวัติการยกเลิกบริการ

1) สถานะปิดบัญชี

ข้อมูลทั้งหมดมีจำนวน 627,470 ตัวอย่าง และประกอบด้วย 22 คุณลักษณะ (Features)

3.3 การเตรียมข้อมูล

ก่อนที่จะนำข้อมูลไปเพื่อทำนาย ต้องมีการทำความสะอาดข้อมูลและการเตรียมข้อมูลดังนี้

1) การจัดการค่าผิดปกติ (Outliers) ตรวจสอบและจัดการข้อมูลที่ผิดปกติหรือไม่สมเหตุสมผล

2) การจัดการข้อมูลที่ขาดหายไป เช่น การใช้ค่าเฉลี่ยหรือการคาดการณ์ค่าที่ขาดหายไปด้วยเทคนิคต่าง ๆ

3) การแปลงข้อมูลเชิงตัวเลขหรือข้อมูลที่ไม่สามารถใช้งานได้ในโมเดล เช่น การใช้ One-Hot Encoding สำหรับข้อมูลเชิงหมวดหมู่

4) การสร้างตัวอย่างข้อมูลปลอมในกลุ่มที่มีจำนวนน้อย เพื่อแก้ปัญหาความไม่สมดุลของข้อมูล เช่น SMOTE

5) การเปลี่ยนข้อมูลที่มีหลายตัวแปรให้เป็นชุดของตัวแปรใหม่ที่เรียกว่า Principal Components ซึ่งอธิบายความแปรปรวนของข้อมูลให้ได้มากที่สุด

6) การเลือกคุณลักษณะโดยใช้ Pearson Correlation และ Spearman Correlation วิเคราะห์ความสัมพันธ์ของข้อมูล

7) การแบ่งข้อมูลออกเป็น Training Set และ Test Set เพื่อใช้ในการฝึกและทดสอบโมเดล

3.4 การเลือกเทคนิคการทำนาย

หลังจากเตรียมข้อมูลแล้ว จะต้องเลือกเทคนิคการทำนายที่เหมาะสมเพื่อทำนายการยกเลิกบริการของลูกค้า โดยเทคนิคที่ใช้ได้แก่

1) Logistic Regression โมเดลที่ใช้สำหรับการทำนายผลลัพธ์ที่เป็นสองสถานะ เช่น การยกเลิกบริการ (Churn) หรือไม่ยกเลิก

2) Random Forest การใช้หลาย ๆ Decision Tree ในการทำนาย เพื่อเพิ่มความแม่นยำ

3) CatBoost ใช้ในงานที่เกี่ยวกับการจัดการข้อมูลที่เป็นตัวแปรเชิงหมวดหมู่

4) AdaBoost เหมาะสำหรับงานที่ต้องการเพิ่มประสิทธิภาพของโมเดลที่มีความเรียบง่ายและต้องการปรับปรุงความแม่นยำให้สูงขึ้นโดยไม่ซับซ้อนเกินไป

5) Gradient Boosting สามารถปรับตัวให้ทำงานได้ดีทั้งในข้อมูลขนาดเล็กและใหญ่ได้

6) XGBoost เหมาะสำหรับการทำงานกับข้อมูลที่มีลักษณะหลากหลายและมีขนาดใหญ่

3.5 การฝึกและทดสอบโมเดล

หลังจากเลือกเทคนิคที่ใช้แล้ว จะต้องมีการฝึกโมเดลโดยใช้ Training Set และทดสอบด้วย Test Set โดยขั้นตอนนี้จะเป็นการปรับแต่งพารามิเตอร์ของโมเดลให้เหมาะสมและประเมินประสิทธิภาพของโมเดล

1) การฝึกโมเดลจะใช้ Training Set ในการฝึกโมเดลโดยปรับพารามิเตอร์ต่าง ๆ เช่น learning rate เพื่อเพิ่มความแม่นยำของผลลัพธ์ให้มากที่สุด

2) การทดสอบโมเดลโดยใช้ Test Set เพื่อทดสอบความสามารถของโมเดลในการทำนายโดยใช้ Evaluation Metrics เช่น Accuracy, Precision, Recall, F1-Score และ Precision-Recall Curves ในการประเมินผล

3.6 การประเมินผล

หลังจากทดสอบโมเดลแล้ว จะต้องมีการประเมินผลการทำนายด้วยการใช้ Evaluation Metrics เพื่อหาค่าความแม่นยำและประสิทธิภาพของโมเดล

1) Accuracy อัตราความถูกต้องของการทำนาย

2) Precision ค่าความแม่นยำในการทำนายลูกค้าที่จะยกเลิกบริการ

3) Recall ค่าความสามารถของโมเดลในการระบุลูกค้าที่จะยกเลิกบริการได้ถูกต้องจากจำนวนลูกค้าที่จะยกเลิกทั้งหมด

4) F1-Score ค่าความสมดุลระหว่าง Precision และ Recall

5) Precision-Recall Curve การวัดความสามารถของโมเดลในการแยกแยะลูกค้าที่ยกเลิกบริการออกจากลูกค้าที่ยังใช้บริการ

ตารางที่ 3.1 ตาราง Evaluation Matrix

เมตริก	สูตร	จุดประสงค์
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	ความแม่นยำโดยรวมของการจำแนกประเภท
Precision	$\frac{TP}{TP+FP}$	การวัดความแม่นยำในการทำนายบวก
Recall	$\frac{TP}{TP+FN}$	การครอบคลุมของกรณีบวกจริง
F1-score	$\frac{Precision \times Recall}{Precision+Recall}$	การวัดที่คำนึงถึงความสมดุลของ Precision และ Recall

3.7 การเปรียบเทียบประสิทธิภาพ

ผลการทดสอบจากอัลกอริทึมแต่ละตัวจะถูกนำมาเปรียบเทียบกันโดยใช้ค่าความแม่นยำ Accuracy, Precision, Recall, F1-Score และ Precision-Recall Curve ที่จะช่วยให้สามารถเลือกอัลกอริทึมที่เหมาะสมที่สุดในการทำนายการยกเลิกบริการของลูกค้าธนาคาร

3.8 เครื่องมือที่ใช้สำหรับวิจัย

การดำเนินการวิจัยนี้จะใช้เครื่องมือและภาษาโปรแกรมดังนี้

- Python ใช้สำหรับการพัฒนาโปรแกรมและการฝึกโมเดล
- Libraries สำหรับการวิจัยในการสร้างและทดสอบโมเดล Numpy, Pandas, Scikit-learn, Matplotlib, XGBoost, CatBoost
- Jupyter Notebook สำหรับการเขียนโปรแกรมและการแสดงผลการทดสอบ

บทที่ 4
ผลการทดลอง และการอภิปรายผล

ในบทนี้จะนำเสนอผลการทดลองจากการทดสอบโมเดลต่าง ๆ ซึ่งใช้สำหรับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารโดยการใช้เครื่องมือและอัลกอริทึมการเรียนรู้ของเครื่องจำนวน 6 ตัว ได้แก่ Logistic Regression, Random Forest, CatBoost, AdaBoost, Gradient Boosting และ XGBoost การเปรียบเทียบผลการทดสอบจะพิจารณาจากเมตริกที่สำคัญ ได้แก่ Accuracy, Precision, Recall, F1-Score และ Precision-Recall Curve ที่เป็นตัวชี้วัดประสิทธิภาพของโมเดลในงานทำนายการยกเลิกบริการ

4.1 ผลการทดลอง

ในการศึกษานี้ได้ทำการทดสอบโมเดลการเรียนรู้ของเครื่องหลายรูปแบบเพื่อนำมาการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารจากเทคนิคที่ได้กล่าวไปแล้วในบทที่ 3 ได้แก่

4.1.1 การรวบรวมข้อมูล

ตัวอย่างข้อมูลที่ถูกรวบรวมโดยธนาคารอยู่ในรูปที่ 4.1 และ 4.2 จำนวน 10 คอลัมน์

	cust_no	open_acct_date	email_validate_ind	verify_sub_channel	current_age	gender	occupation	income_source	income_range	marital_status	...	close_kept	luckydra
80512	000000YY4XZ3X1	8/14/2024	Y	SCB	32	Male	Freelancer/Self-Employed	Savings	15,000 - 29,999 Baht	Single	...	NaN	
175294	000000180031Y5	11/2/2020	Y	Branch	35	Female	Company Employee	Savings	100,000 - 399,999 Baht	Single	...	NaN	
321642	00000000YX7ZZY	6/16/2020	Y	Branch	56	Male	Company Employee	Salary/Wages	30,000 - 49,999 Baht	Married (Registered)	...	NaN	
559012	000000YZ587547	8/26/2024	Y	KBANK	27	Female	Company Employee	Salary/Wages	15,000 - 29,999 Baht	Single	...	NaN	
68674	000000YZ4Z855Z	2/17/2022	Y	7-11	38	Male	Company Employee	Salary/Wages	15,000 - 29,999 Baht	Divorced	...	NaN	
382783	000000YZ001443	8/9/2021	Y	7-11	57	Female	Freelancer/Self-Employed	Private Business	15,000 - 29,999 Baht	Married (Registered)	...	NaN	
62648	0000001Y873Z48	9/22/2020	Y	Branch	42	Female	Company Employee	Salary/Wages	50,000 - 99,999 Baht	Single	...	NaN	
580649	000000Y488Z3XY	12/31/2024	Y	KBANK	39	Female	Business Owner (with Commercial/Juristic Perso...	Private Business	15,000 - 29,999 Baht	Single	...	NaN	
478500	00000015XZ1XYX	5/11/2024	Y	TMB	32	Female	Company Employee	Salary/Wages	15,000 - 29,999 Baht	Single	...	NaN	
402978	00000033Y40Y4X	2/18/2024	N	SCB	26	Female	Civil Servant	Salary/Wages	15,000 - 29,999 Baht	Single	...	NaN	

10 rows × 24 columns

รูปที่ 4.1 ตัวอย่างข้อมูลส่วนที่ 1

รูปที่ 4.1 แสดงตารางที่มีคอลัมน์และแถวที่ประกอบด้วยข้อมูลต่าง ๆ โดยคอลัมน์มีชื่อดังนี้ cust_no, email_validate_ind, verify_sub_channel, current_age, gender, occupation, income_source, income_range, marital_status และ acct_status

luckydrawy2024	s99_24	sonline_kept_event2024	no_of_txn2024_c	trxn_amt2024_c	keptapp_bal_31dec24	first_deposit_date	first_deposit_amt	fisrtdate_join
NaN	NaN	NaN	NaN	NaN	0.00	NaN	0.0	8/14/2024
NaN	NaN	NaN	10.0	31.95	4428.20	11/2/2020	10000.0	11/2/2020
NaN	NaN	NaN	5.0	2086.08	2087.03	6/16/2020	100.0	6/16/2020
NaN	NaN	NaN	8.0	4111.17	4111.39	8/25/2024	99.0	8/25/2024
NaN	NaN	NaN	NaN	NaN	0.09	2/18/2022	1000.0	2/17/2022
NaN	NaN	NaN	45.0	2581499.41	486934.01	8/9/2021	5000.0	8/9/2021
0.0	0.0	0.0	37.0	1833978.40	27589.73	9/22/2020	10000.0	9/22/2020
NaN	NaN	NaN	1.0	100.00	100.00	12/31/2024	100.0	12/31/2024
NaN	NaN	NaN	28.0	20955.17	55.17	5/11/2024	1000.0	5/11/2024
NaN	NaN	NaN	107.0	28089.23	1000.08	2/18/2024	6.0	2/18/2024

รูปที่ 4.2 ตัวอย่างข้อมูลส่วนที่ 2

รูปที่ 4.2 แสดงตารางที่มีคอลัมน์และแถวที่ประกอบด้วยข้อมูลตัวเลขต่าง ๆ โดยคอลัมน์มีชื่อดังนี้ c99_2024, c_online_event2024, no_of_txn2024_c, trxn_amt2024_c, bal_31dec24, first_deposit_amt, difference_lld, difference_oad, oad_mld และ nb_months_fdd

4.1.2 การเตรียมข้อมูล

4.1.2.1 ตรวจสอบข้อมูลที่ได้รับจากธนาคาร

นำข้อมูลธนาคารมาตรวจสอบว่ามีทั้งหมดกี่คอลัมน์, มีประเภทข้อมูลอะไรบ้าง และจำนวนข้อมูลในแต่ละคอลัมน์

```
RangeIndex: 627470 entries, 0 to 627469
Data columns (total 24 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   cust_no                 627470 non-null  object
 1   open_acct_date          627470 non-null  object
 2   email_validate_ind      627470 non-null  object
 3   verify_sub_channel      611067 non-null  object
 4   current_age             627470 non-null  int64
 5   gender                  627470 non-null  object
 6   occupation              627469 non-null  object
 7   income_source           627470 non-null  object
 8   income_range            627469 non-null  object
 9   marital_status          627470 non-null  object
 10  acct_status             627470 non-null  int64
 11  is_open_tgt             627470 non-null  int64
 12  phone_os                588884 non-null  object
 13  max_login_date          626489 non-null  object
 14  close_kept              10764 non-null   object
 15  luckydrawy2024          336668 non-null  float64
 16  s99_24                  336668 non-null  float64
 17  sonline_kept_event2024  336668 non-null  float64
 18  no_of_txn2024_c         407673 non-null  float64
 19  trxn_amt2024_c          407673 non-null  float64
 20  keptapp_bal_31dec24     605141 non-null  float64
 21  first_deposit_date      515943 non-null  object
 22  first_deposit_amt       616731 non-null  float64
 23  fisrtdate_join          627465 non-null  object
dtypes: float64(7), int64(3), object(14)
```

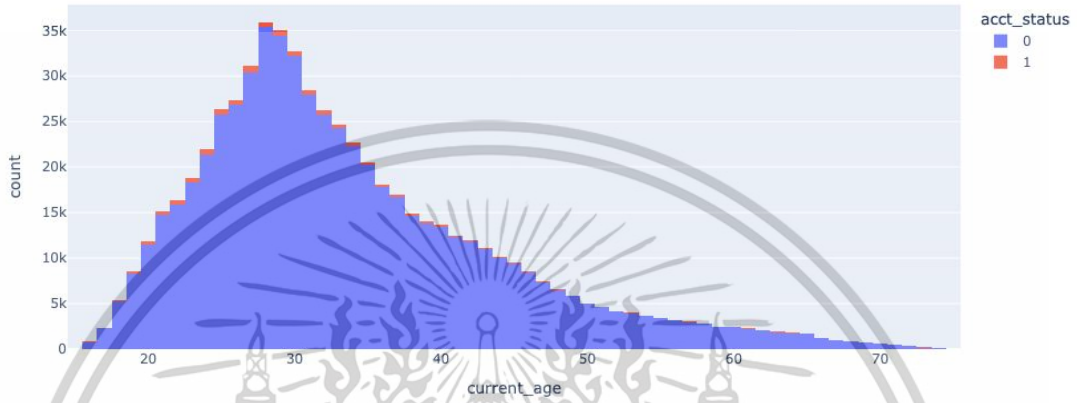
รูปที่ 4.3 ภาพรวมของข้อมูลที่ได้จากธนาคาร

รูปที่ 4.3 แสดงตารางที่มี 24 คอลัมน์ พร้อมข้อมูลต่าง ๆ เช่น ประเภทของข้อมูล, จำนวนข้อมูลที่ไม่เป็นค่าว่าง และชื่อคอลัมน์

4.1.2.2 สร้างภาพข้อมูล

1) สร้างภาพการกระจายตัวอายุของลูกค้าธนาคาร

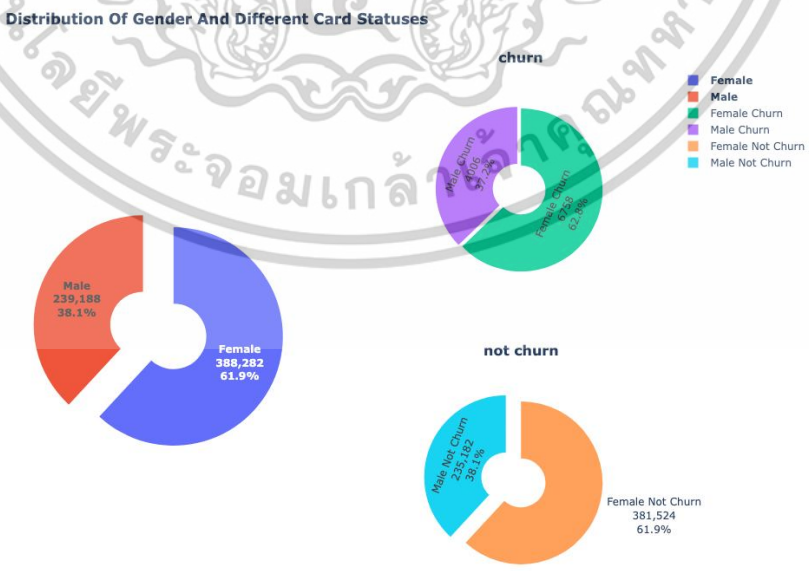
2)



รูปที่ 4.4 ภาพการกระจายตัวอายุของลูกค้าธนาคาร

รูปที่ 4.4 แสดงการกระจายตัวอายุของลูกค้าธนาคารเทียบกับการยกเลิกบริการโดยที่ 1 คือการยกเลิกบริการมีสีส้ม และ 0 คือยังใช้งานอยู่มีสีน้ำเงิน

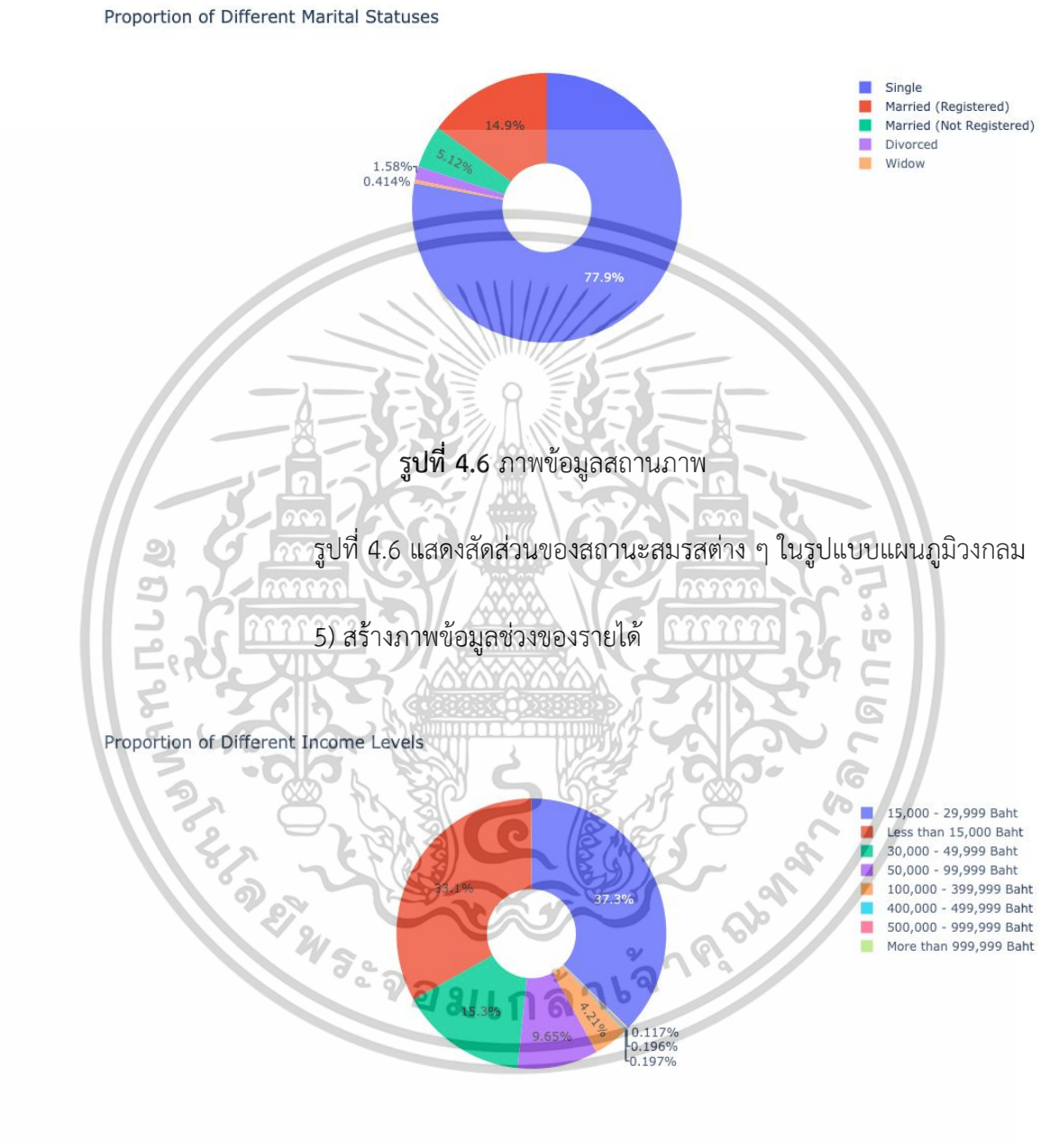
3) สร้างภาพข้อมูลเพศ



รูปที่ 4.5 ภาพการกระจายตามเพศ

รูปที่ 4.5 แสดงการกระจายตามเพศเปรียบเทียบกับการยกเลิกบริการ

4) สร้างภาพข้อมูลสถานภาพ



รูปที่ 4.6 ภาพข้อมูลสถานภาพ

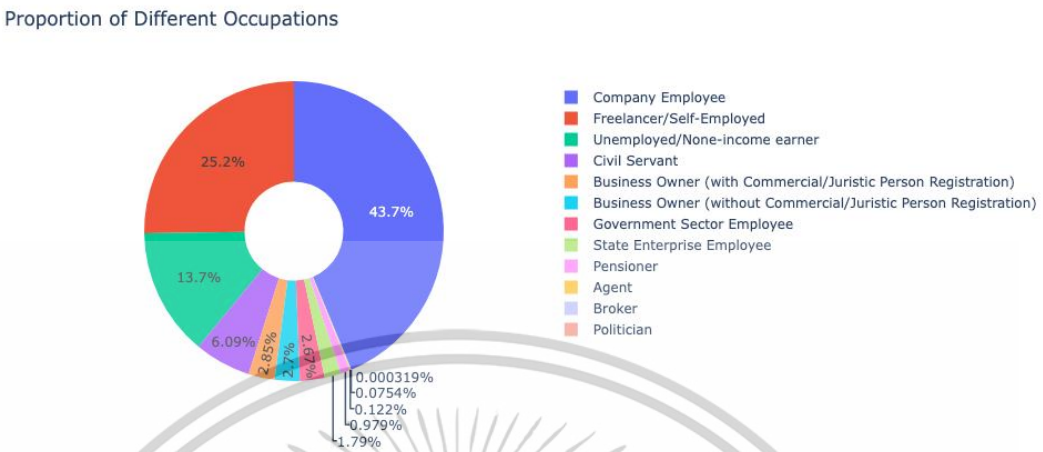
รูปที่ 4.6 แสดงสัดส่วนของสถานะสมรสต่าง ๆ ในรูปแบบแผนภูมิวงกลม

5) สร้างภาพข้อมูลช่วงของรายได้

รูปที่ 4.7 ภาพข้อมูลช่วงของรายได้

รูปที่ 4.7 แสดงสัดส่วนของช่วงของรายได้ในรูปแบบแผนภูมิวงกลม

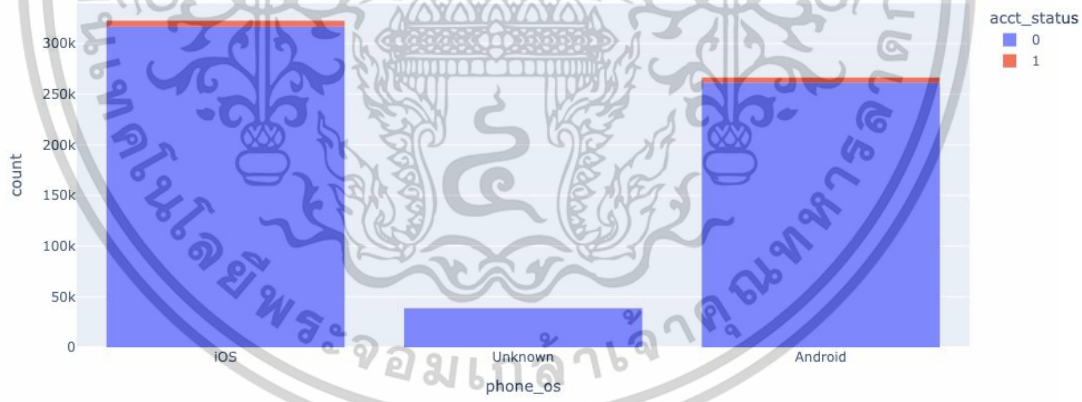
6) สร้างภาพข้อมูลอาชีพ



รูปที่ 4.8 ภาพข้อมูลอาชีพ

รูปที่ 4.8 แสดงสัดส่วนของอาชีพต่าง ๆในรูปแบบแผนภูมิวงกลม

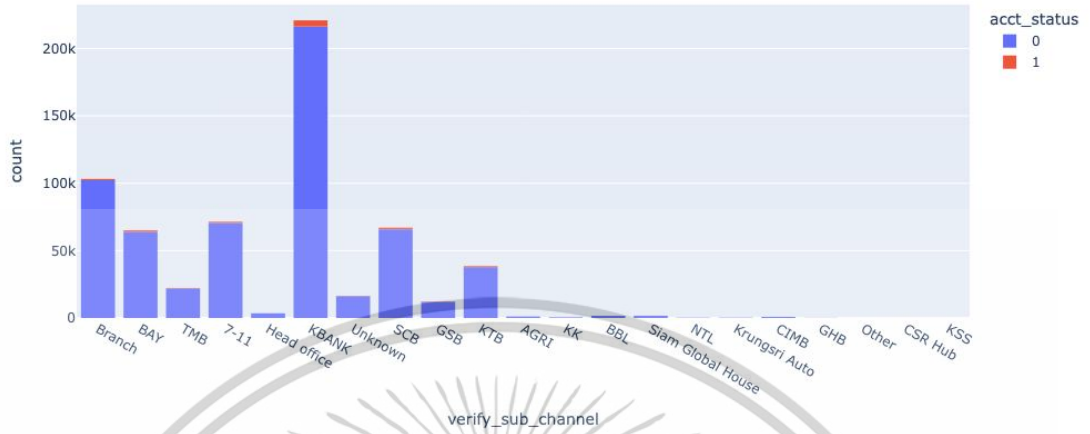
7) สร้างภาพข้อมูลระบบปฏิบัติการมือถือ



รูปที่ 4.9 ภาพข้อมูลระบบปฏิบัติการมือถือ

รูปที่ 4.9 แสดงการกระจายตัวระบบปฏิบัตการมือถือของลูกค้าธนาคาร เทียบกับการยกเลิกบริการโดยที่ 1 คือการยกเลิกบริการมีสีส้ม และ 0 คือยังใช้งานอยู่มีสีน้ำเงิน

8) สร้างภาพช่องทางการยืนยันตัวตน



รูปที่ 4.10 ภาพข้อมูลช่องทางการยืนยันตัวตน

รูปที่ 4.10 แสดงการกระจายตัวช่องทางการยืนยันตัวตนของลูกค้าธนาคาร เทียบกับการยกเลิกบริการโดยที่ 1 คือการยกเลิกบริการมีสีส้ม และ 0 คือยังใช้งานอยู่มีสีน้ำเงิน

4.1.2.3 แก้ไขข้อมูลที่ขาดหายไปและเปลี่ยนประเภทข้อมูล

จากรูปที่ 4.3 จะเห็นได้ว่าข้อมูลบางส่วนเป็นค่าว่างประกอบไปด้วยคอลัมน์ดังต่อไปนี้ verify_sub_channel, phone_os, luckydrawy2024, s99_24 conline_event2024, no_of_txn2024, trxn_amt2024, bal_31dec24 และ first_deposit_amt ได้มีการแก้ไขดังนี้

1) ข้อมูลประเภท object ได้แก่ ช่องทางการยืนยันตัวตนมีการยืนยันผ่านระบบอื่นๆจึงใส่ค่า Unknown และระบบปฏิบัติการมือถือที่ไม่ใช่ android หรือ ios มีข้อมูลที่ขาดหายไปจึงแก้ไขโดยการใส่ค่า Unknown แทน

2) ข้อมูลประเภทตัวเลข การเข้าร่วมโปรโมชั่น luckydraw2024, การเข้าร่วมโปรโมชั่น s99_24 และ การเข้าร่วมโปรโมชั่น conline_event2024 ใส่ 0.0 แทนการไม่ได้เข้าร่วม ส่วนจำนวนครั้งที่ทำรายการ, ยอดเงินที่ทำรายการ, ยอดเงินคงเหลือและยอดเงินฝากครั้งแรกใส่ 0.0 แทนจำนวนที่ขาดหายไป

3) ข้อมูลประเภทวันที่ ทำการแปลงให้เป็นจำนวนเดือนหรือวันแทนเพื่อให้โมเดลสามารถทำงานได้

```
df['verify_sub_channel'] = df['verify_sub_channel'].fillna("Unknown")
df['phone_os'] = df['phone_os'].fillna("Unknown")
```

```
df['luckydrawy2024'] = df['luckydrawy2024'].fillna(0.0)
df['s99_24'] = df['s99_24'].fillna(0.0)
df['sonline_kept_event2024'] = df['sonline_kept_event2024'].fillna(0.0)
df['no_of_txn2024_c'] = df['no_of_txn2024_c'].fillna(0.0)
df['trxn_amt2024_c'] = df['trxn_amt2024_c'].fillna(0.0)
df['keptapp_bal_31dec24'] = df['keptapp_bal_31dec24'].fillna(0.0)

df['first_deposit_amt'] = df['first_deposit_amt'].fillna(0.0)
```

✓ 0.0s

```
df['fdd'] = pd.to_datetime(df['first_deposit_date'], infer_datetime_format=True , format='%m/%d/%Y')
df['difference_fdd'] = (pd.Timestamp('2025-02-21') - df['fdd']) / np.timedelta64(1, 'D')
df['difference_fdd'] = df['difference_fdd'].fillna(0.0)
df['nb_months_fdd'] = (df['difference_fdd'] / 30.44).astype(int)
df = df.drop(['difference_fdd', 'fdd', 'first_deposit_date'], axis=1)
df.head()
```

✓ 0.7s

รูปที่ 4.11 วิธีแก้ไขข้อมูลที่ขาดหายไป

รูปที่ 4.11 แสดงวิธีการแก้ไขข้อมูลที่ขาดหายไปของแต่ละคอลัมน์

4.1.2.4 เปลี่ยนข้อมูลให้โมเดลสามารถใช้งานได้

เนื่องจากโมเดลต้องการรับค่าข้อมูลประเภทตัวเลขเท่านั้น จึงได้ทำการแปลงข้อมูล มีตัวอย่างดังนี้

1) ย้ายคอลัมน์การยกเลิกบริการไปไว้ข้างหน้าสุด

2) แปลงข้อมูลเพศให้อยู่ในรูปแบบ 0 และ 1 โดยที่เพศหญิงมีค่าเป็น 1 และเพศชายมีค่าเป็น 0

3) แปลงข้อมูลระบบปฏิบัติการมือถือกำหนดให้ ios เป็น 1, Android เป็น 0 และให้ Unknown เป็น -1

4) แปลงข้อมูลการยืนยันอีเมล กำหนดให้ Y คือ 1 และ N คือ 0

5) แปลงข้อมูลเป็นข้อมูล Dummy ด้วยวิธีการ One-Hot Encoding ประกอบไปด้วย ข้อมูลอาชีพ, ช่วงของรายได้, แหล่งที่มารายได้, สถานภาพ และช่องทางการยืนยันตัวตน

6) ลบคอลัมน์ที่ถูกแปลงเป็น Dummy ออก

```
df.insert(0, 'acct_status', df.pop('acct_status'))
df_copy = df
df_copy['gender'] = df_copy['gender'].replace({'Female':1,'Male':0})
df_copy['phone_os'] = df_copy['phone_os'].replace({'iOS':1,'Android':0,'Unknown':-1})
df_copy['email_validate_ind'] = df_copy['email_validate_ind'].replace({'Y':1,'N':0})
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['occupation'])],axis=1)
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['income_range'])],axis=1)
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['income_source'])],axis=1)
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['marital_status'])],axis=1)
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['new_verify_sub_channel'])],axis=1)
df_copy.drop(columns =
    [
        'verify_sub_channel',
        'new_verify_sub_channel',
        'occupation',
        'income_source',
        'income_range',
        'marital_status',
    'cust_no'],
    inplace=True)
✓ 1.4s
```

รูปที่ 4.12 การแปลงข้อมูลให้เป็น Dummy

รูปที่ 4.12 แสดงวิธีการแปลงข้อมูลสำหรับไปใช้ในโมเดลและลบคอลัมน์ที่ไม่ได้ใช้ออก

เมื่อทำการแปลงข้อมูลเป็น Dummy จะได้คุณลักษณะใหม่ทั้งหมด 43 คุณลักษณะ ได้ผลลัพธ์ดังรูปที่ 4.13

```
Data columns (total 37 columns):
 #   Column
---  ------
 0   Agent
 1   Broker
 2   Business Owner (with Commercial/Juristic Person Registration)
 3   Business Owner (without Commercial/Juristic Person Registration)
 4   Civil Servant
 5   Company Employee
 6   Freelancer/Self-Employed
 7   Government Sector Employee
 8   Pensioner
 9   Politician
 10  State Enterprise Employee
 11  Unemployed/None-income earner
 12  100,000 - 399,999 Baht
 13  15,000 - 29,999 Baht
 14  30,000 - 49,999 Baht
 15  400,000 - 499,999 Baht
 16  50,000 - 99,999 Baht
 17  500,000 - 999,999 Baht
 18  Less than 15,000 Baht
 19  More than 999,999 Baht
 20  Dividend from instrument / funds
 21  Earnings from service provision
 22  Earnings from trading business / Commodities price
 23  Heritage/Gifts
 24  Instrument/Funds
 25  Investment Capital
 26  Loans
 27  Other
 28  Parent
 29  Private Business
 30  Salary/Wages
 31  Savings
 32  Divorced
 33  Married (Not Registered)
 34  Married (Registered)
 35  Single
 36  Widow
```

รูปที่ 4.13 ข้อมูลที่ถูกทำ One-Hot Encoding

รูปที่ 4.13 แสดงข้อมูลในส่วนที่ถูก One-Hot Encoding หรือทำตัวแปร Dummy เพื่อให้สามารถนำค่าเข้าไปใส่ในโมเดลการเรียนรู้ของเครื่องได้

4.1.2.5 แก้ปัญหาข้อมูลที่ไม่สมดุล (Imbalanced Data)

เนื่องจากข้อมูลที่ได้รับมามีค่าการยกเลิกบริการ (Churn) อยู่เพียง 2% จากการทดลองได้ทำการแก้ไขด้วยวิธีการ เทคนิคการสุ่มตัวอย่างเกินขนาด Synthetic Minority Oversampling Technique (SMOTE) โดยได้ทำการเพิ่มข้อมูลการยกเลิกบริการจากเดิม 2% เป็น 10% เพื่อให้โมเดลมีแนวโน้มที่จะทำนายคลาส 0 ที่มีจำนวนน้อยกว่าได้ดีขึ้น

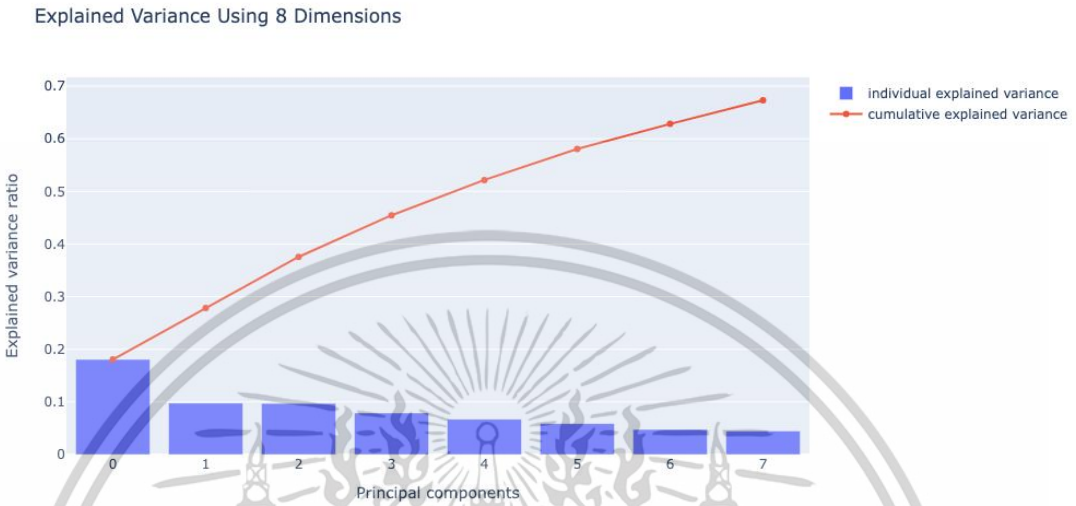
รูปที่ 4.14 การแก้ปัญหาข้อมูลที่ไม่สมดุล (SMOTE)

รูปที่ 4.14 แสดงวิธีการใช้คำสั่ง SMOTE จะเห็นได้ว่าข้อมูลต้นฉบับมีค่าการยกเลิกบริการอยู่ 10,764 คน ทำให้มีค่าเพิ่มขึ้นเป็น 61,670 คน

4.1.2.6 ทำการแปลงค่า Dummy เพื่อลดมิติของข้อมูลด้วยวิธีการ PCA

เนื่องจากตัวแปร Dummy ของเรามีจำนวนมากจึงได้ใช้วิธีการ Principal Component Analysis (PCA) เพื่อลดจำนวนของตัวแปร Dummy ลง โดยจะใช้ตัวแปร Dummy ทั้งหมดที่ได้จากการทำ One-hot encoder ให้อยู่ในรูปแบบของ PC0-PC7 ข้อดีของการใช้วิธีนี้คือ 1.ลดจำนวนมิติของตัวแปร Dummy จาก 43 ตัว ให้เหลือ 8 ตัว ดังรูปที่ X.X 2.ลด overfitting เนื่องจากโมเดลที่มีฟีเจอร์เยอะเกินไปอาจ overfit ได้ง่าย 3.ทำให้การเรียนรู้ของโมเดลเร็วขึ้นเมื่อจำนวนฟีเจอร์ลดลง ข้อเสียของการใช้วิธีนี้คือ 1.หลังจากทำ PCA แล้วจะไม่สามารถอธิบายฟีเจอร์ที่ผ่าน

การทำ PCA แล้วได้ 2.การใช้ PCA เหมาะกับข้อมูล สมมติฐานข้อมูลต่อเนื่อง (continuous) ในที่นี้สามารถใช้วิธีอื่นได้เช่น Multiple Correspondence Analysis (MCA) หรือ Feature Selection



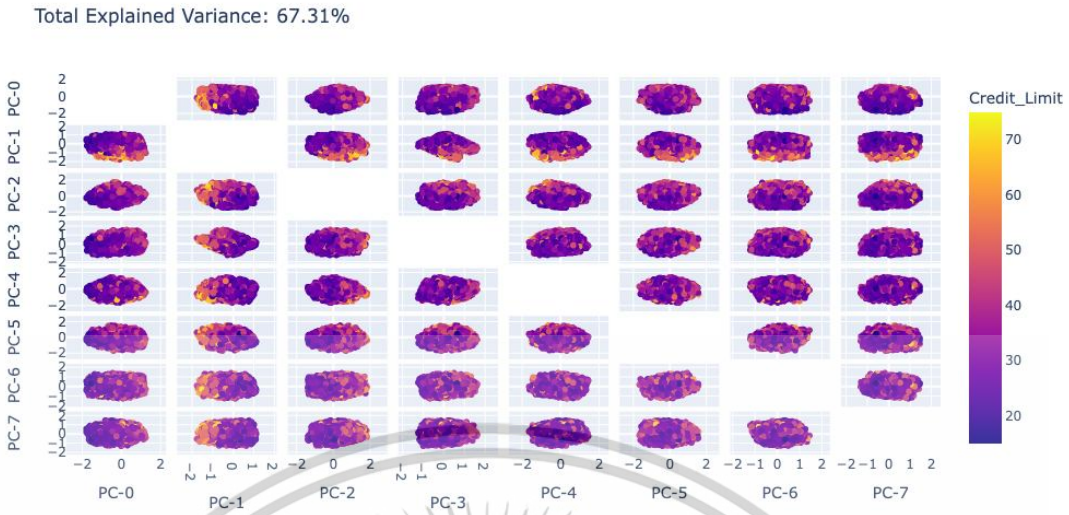
รูปที่ 4.15 อัตราส่วนความแปรปรวนของ PC0-PC7

รูปที่ 4.15 แสดงกราฟอัตราส่วนความแปรปรวนทั้งหมด กราฟนี้มีชุดข้อมูลสองชุดได้แก่

1) ความแปรปรวนที่อธิบายได้แต่ละตัว (individual explained variance) แสดงด้วยแท่งสีน้ำเงิน

2) ความแปรปรวนที่อธิบายได้สะสม (cumulative explained variance) แสดงด้วยเส้นสีแดงที่มีเครื่องหมาย

แท่งสีน้ำเงินแสดงความแปรปรวนที่อธิบายได้แต่ละตัวสำหรับแต่ละองค์ประกอบหลัก โดยองค์ประกอบแรกมีค่ามากที่สุดและค่าขององค์ประกอบถัดไปลดลงเรื่อย ๆ

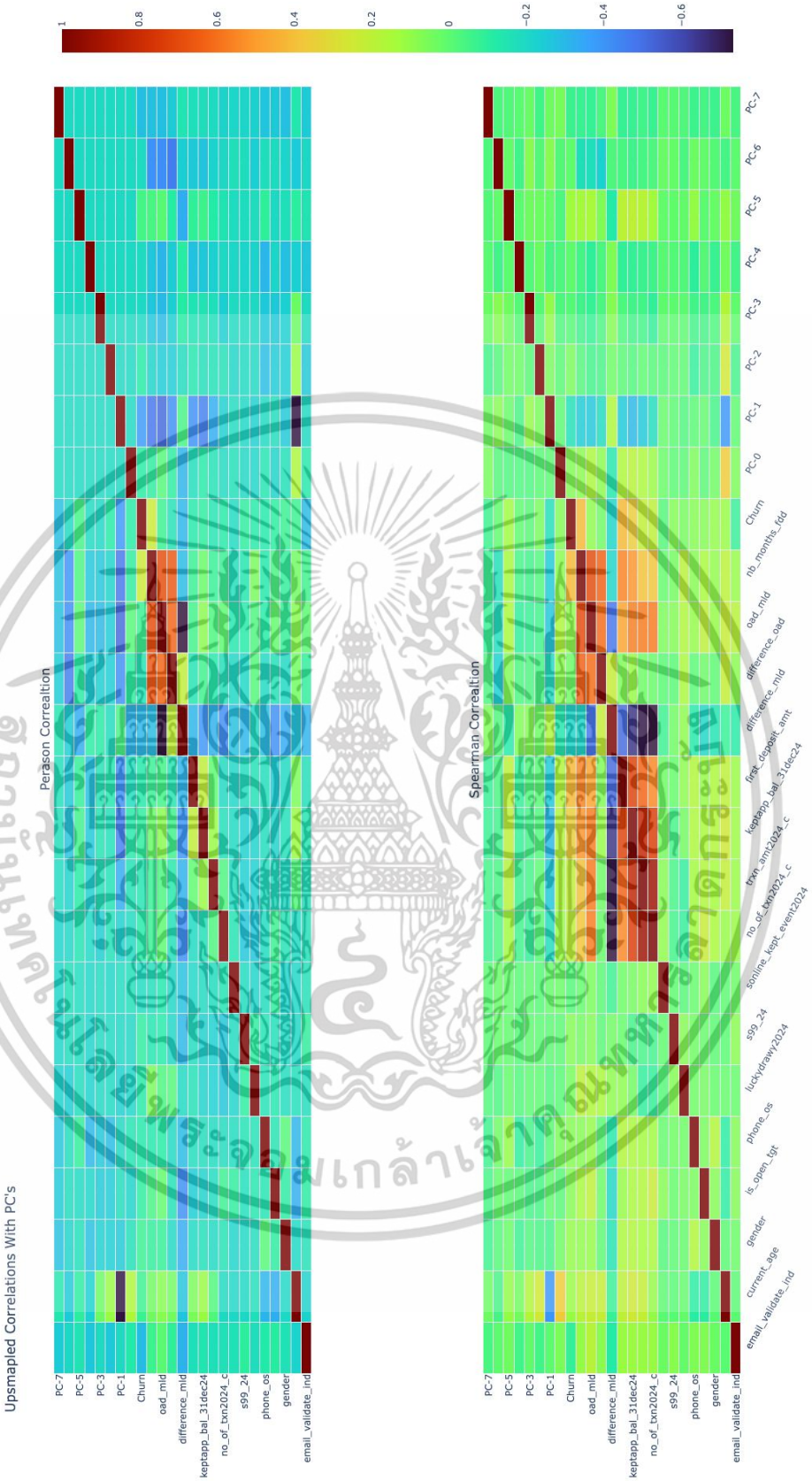


รูปที่ 4.16 ความสัมพันธ์ระหว่างองค์ประกอบหลักต่าง ๆ ในชุดข้อมูล

รูปที่ 4.16 แสดงกราฟกระจาย (scatter plot matrix) ที่แสดงความสัมพันธ์ระหว่างองค์ประกอบหลัก (Principal Components) PC-0 ถึง PC-7 ของชุดข้อมูล โดยมีแกน X และ Y แต่ละแถวและคอลัมน์ในเมทริกซ์แสดง scatter plot ระหว่างคู่ขององค์ประกอบหลัก เช่น PC-0 กับ PC-1, PC-0 กับ PC-2 เป็นต้น ถ้า scatter plot แสดงให้เห็นว่าข้อมูลกระจายดีในองค์ประกอบหลักต้น ๆ (เช่น PC-0, PC-1) แสดงว่า PCA ทำงานได้ดีในการลดมิติ

4.1.2.7 หาความสัมพันธ์ของข้อมูลเพื่อเลือกคุณลักษณะไปใช้ในโมเดล

จากการทดลองได้ใช้วิธีการหาความสัมพันธ์ของ Pearson และ Spearman ในการเลือกคุณลักษณะ ได้ผลลัพธ์ดังรูปที่ 4.17



รูปที่ 4.17 ความสัมพันธ์ของ Pearson และ Spearman ระหว่างคุณลักษณะต่าง ๆ กับองค์ประกอบหลัก

รูปที่ 4.17 แสดงความสัมพันธ์ของ Pearson และ Spearman ระหว่างคุณลักษณะต่าง ๆ กับองค์ประกอบหลัก จากรูปจะเห็นได้ว่าคุณลักษณะดังต่อไปนี้มีความสัมพันธ์กับการยกเลิกบริการได้แก่ 1.luckydrawy2024 มีความสัมพันธ์ไปในทิศทางเดียวกันกับการยกเลิกบริการ, 2.difference_mld มีความสัมพันธ์ในทิศทางตรงกันข้ามกับการยกเลิกบริการ, bal_31dec24, trxn_amt2024_c และ nb_months_fdd จึงได้ทำการเลือกคุณลักษณะเหล่านี้มาทดสอบในโมเดล รวมไปถึงองค์ประกอบหลักทั้งหมดเพื่อให้โมเดลสามารถเรียนรู้จากข้อมูลที่ถูก One-Hot Encoding ด้วย

4.1.3 หาพารามิเตอร์ที่เหมาะสมกับโมเดล

ในการทดสอบโมเดลเราได้ทำการค้นหา Paramter ของแต่ละโมเดลให้เหมาะสมด้วยวิธีการ Random Search ได้ผลลัพธ์ดังนี้

4.1.3.1 การหาค่าพารามิเตอร์ของโมเดล Random Forest

```python
from sklearn.model_selection import RandomizedSearchCV

param_dist = {
    'n_estimators': [100, 200],
    'max_depth': [5, 10, None],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2],
    'class_weight': ['balanced', None]
}

rf = RandomForestClassifier(random_state=42, n_jobs=-1)
rf_search = RandomizedSearchCV(
    rf, param_distributions=param_dist, n_iter=10, cv=3, scoring='recall', n_jobs=-1, random_state=42, verbose=0
)
rf_search.fit(x_small, y_small)

print("Best Params:", rf_search.best_params_)
```

```
Best Params: {'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 2, 'max_depth': 5, 'class_weight': 'balanced'}
```

รูปที่ 4.18 การตั้งค่าหาพารามิเตอร์ของโมเดล Random Forest

รูปที่ 4.18 แสดงการค้นหาพารามิเตอร์โดยใช้ RandomizedSearchCV โดยสุ่มเลือกชุดพารามิเตอร์จำนวน 10 ชุด (n_iter=10) และใช้การแบ่งข้อมูลแบบ cross-validation 3 ชุด (cv=3) เกณฑ์การประเมินใช้ recall เป็นตัวชี้วัดประสิทธิภาพของโมเดล จากผลการทดลองใช้ RandomSearch ได้ผลลัพธ์คือ {'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 2, 'max_depth': 5, 'class_weight': 'balanced'}

4.1.3.2 การหาค่าพารามิเตอร์ของโมเดล AdaBoost

```
param_dist = {
    'n_estimators': [50, 100, 150, 200],
    'learning_rate': [0.01, 0.1, 0.5, 1.0]
}

ada = AdaBoostClassifier(random_state=42)
ada_search = RandomizedSearchCV(ada, param_distributions=param_dist,
                                n_iter=10, cv=3, scoring='recall', n_jobs=-1, random_state=42, verbose=0)

ada_search.fit(x_small, y_small)
print("Best Params:", ada_search.best_params_)
```

```
Best Params: {'n_estimators': 200, 'learning_rate': 1.0}
```

รูปที่ 4.19 การตั้งค่าหาพารามิเตอร์ของโมเดล AdaBoost

รูปที่ 4.19 แสดงการค้นหาพารามิเตอร์โดยใช้ RandomizedSearchCV โดยสุ่มเลือกชุดพารามิเตอร์จำนวน 10 ชุด (n_iter=10) และใช้การแบ่งข้อมูลแบบ cross-validation 3 ชุด (cv=3) เกณฑ์การประเมินใช้ recall เป็นตัวชี้วัดประสิทธิภาพของโมเดล จากผลการทดลองใช้ RandomSearch ได้ผลลัพธ์คือ {'n_estimators': 200, 'learning_rate': 1.0}

4.1.3.3 การหาค่าพารามิเตอร์ของโมเดล XGBoost

```
xgb = XGBClassifier(eval_metric='logloss')

xgb_param = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2],
    'subsample': [0.6, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0]
}

xgb_search = RandomizedSearchCV(xgb, param_distributions=xgb_param,
                                n_iter=10, cv=3, scoring='recall', n_jobs=-1, random_state=42, verbose=0)

xgb_search.fit(x_small, y_small)
print("Best Params:", xgb_search.best_params_)
```

```
Best Params: {'subsample': 0.6, 'n_estimators': 300, 'max_depth': 5, 'learning_rate': 0.2, 'colsample_bytree': 0.8}
```

รูปที่ 4.20 การตั้งค่าหาพารามิเตอร์ของโมเดล XGBoost

รูปที่ 4.20 แสดงการค้นหาพารามิเตอร์โดยใช้ RandomizedSearchCV โดยสุ่มเลือกชุดพารามิเตอร์จำนวน 10 ชุด (n_iter=10) และใช้การแบ่งข้อมูลแบบ cross-validation 3 ชุด (cv=3) เกณฑ์การประเมินใช้ recall เป็นตัวชี้วัดประสิทธิภาพของโมเดล จากผลการทดลองใช้ RandomSearch ได้ผลลัพธ์คือ {'subsample': 0.6, 'n_estimators': 300, 'max_depth': 5, 'learning_rate': 0.2, 'colsample_bytree': 0.8}

4.1.3.4 การหาค่าพารามิเตอร์ของโมเดล GradientBoosting

```
gbc = GradientBoostingClassifier(random_state=42)

gbc_param = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.6, 0.8, 1.0]
}

gbc_search = RandomizedSearchCV(gbc, param_distributions=gbc_param,
                                n_iter=10, cv=3, scoring='recall', n_jobs=-1, random_state=42, verbose=0)

gbc_search.fit(x_small, y_small)
print("Best Params:", gbc_search.best_params_)
```

```
Best Params: {'subsample': 0.8, 'n_estimators': 300, 'max_depth': 5, 'learning_rate': 0.2}
```

รูปที่ 4.21 การตั้งค่าหาพารามิเตอร์ของโมเดล GradientBoosting

รูปที่ 4.21 แสดงการค้นหาพารามิเตอร์โดยใช้ RandomizedSearchCV โดยสุ่มเลือกชุดพารามิเตอร์จำนวน 10 ชุด (n_iter=10) และใช้การแบ่งข้อมูลแบบ cross-validation 3 ชุด (cv=3) เกณฑ์การประเมินใช้ recall เป็นตัวชี้วัดประสิทธิภาพของโมเดล จากผลการทดลองใช้ RandomSearch ได้ผลลัพธ์คือ {'subsample': 0.8, 'n_estimators': 300, 'max_depth': 5, 'learning_rate': 0.2}

4.1.3.5 การหาค่าพารามิเตอร์ของโมเดล CatBoost

```
cat = CatBoostClassifier(verbose=0)

cat_param = {
    'iterations': [100, 200, 300],
    'depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2],
    'l2_leaf_reg': [1, 3, 5]
}

cat_search = RandomizedSearchCV(cat, param_distributions=cat_param,
                                n_iter=10, cv=3, scoring='recall', n_jobs=-1, random_state=42, verbose=0)

cat_search.fit(x_small, y_small)
print("Best Params:", cat_search.best_params_)
```

```
Best Params: {'learning_rate': 0.1, 'l2_leaf_reg': 5, 'iterations': 200, 'depth': 7}
```

รูปที่ 4.22 การตั้งค่าหาพารามิเตอร์ของโมเดล CatBoost

รูปที่ 4.22 แสดงการค้นหาพารามิเตอร์โดยใช้ RandomizedSearchCV โดยสุ่มเลือกชุดพารามิเตอร์จำนวน 10 ชุด (n_iter=10) และใช้การแบ่งข้อมูลแบบ cross-validation 3 ชุด (cv=3) เกณฑ์การประเมินใช้ recall เป็นตัวชี้วัดประสิทธิภาพของโมเดล จากผลการทดลองใช้ RandomSearch ได้ผลลัพธ์คือ {'learning_rate': 0.1, 'l2_leaf_reg': 5, 'iterations': 200, 'depth': 7}

4.1.3.6 การหาค่าพารามิเตอร์ของโมเดล LogisticRegression

```
lr = LogisticRegression(solver='saga', max_iter=1000)

lr_param = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100],
    'penalty': ['l1', 'l2'],
    'class_weight': ['balanced', None],
}

lr_search = RandomizedSearchCV(lr, param_distributions=lr_param,
                               n_iter=10, cv=3, scoring='recall', n_jobs=-1, random_state=42, verbose=0)

lr_search.fit(x_small, y_small)
print("Best Params:", lr_search.best_params_)


Best Params: {'penalty': 'l1', 'class_weight': 'balanced', 'C': 0.1}
```

รูปที่ 4.23 การตั้งค่าหาพารามิเตอร์ของโมเดล LogisticRegression

รูปที่ 4.23 แสดงการค้นหาพารามิเตอร์โดยใช้ RandomizedSearchCV โดยสุ่มเลือกชุดพารามิเตอร์จำนวน 10 ชุด (n_iter=10) และใช้การแบ่งข้อมูลแบบ cross-validation 3 ชุด (cv=3) เกณฑ์การประเมินใช้ recall เป็นตัวชี้วัดประสิทธิภาพของโมเดล จากผลการทดลองใช้ RandomSearch ได้ผลลัพธ์คือ {'penalty': 'l1', 'class_weight': 'balanced', 'C': 0.1}

4.1.4 เลือกเทคนิคการทำนาย

จากการค้นหาข้อมูลได้ทำการเลือกเทคนิคการทำนายมาทั้ง 6 ชนิดได้แก่

1) Random Forest
2) AdaBoost
3) XGBoost
4) Gradient Boosting
5) CatBoost
6) Logistic Regression

```
rf_pipe = Pipeline(steps =[ ('scale',StandardScaler()), ("RF",RandomForestClassifier(random_state=42)) ])
ada_pipe = Pipeline(steps =[ ('scale',StandardScaler()), ("RF",AdaBoostClassifier(random_state=42,learning_rate=0.4)) ])
xg_pipe = Pipeline(steps =[ ('scale',StandardScaler()), ("RF",XGBClassifier(random_state=42,learning_rate=0.4)) ])
gb_pipe = Pipeline(steps =[ ('scale',StandardScaler()), ("RF",GradientBoostingClassifier(random_state=42)) ])
cb_pipe = Pipeline(steps =[ ('scale',StandardScaler()), ("RF",CatBoostClassifier(verbose=0)) ])
lr_pipe = Pipeline(steps =[ ('scale',StandardScaler()), ("RF",LogisticRegression(random_state=42)) ])
✓ 0.0s
```

รูปที่ 4.24 การตั้งค่าโมเดลการเรียนรู้ของเครื่อง

รูปที่ 4.24 แสดงวิธีการตั้งค่าของโมเดลการเรียนรู้ของเครื่อง

4.1.5 ประเมินผล

หลังจากทดสอบโมเดลแล้ว มีการประเมินผลการทำนายด้วยการใช้ Evaluation Metrics เพื่อหาค่าความแม่นยำและประสิทธิภาพของโมเดล ได้ผลการทดลองดังนี้

1) ผลลัพธ์ของโมเดลต่าง ๆ บนข้อมูลทดสอบ

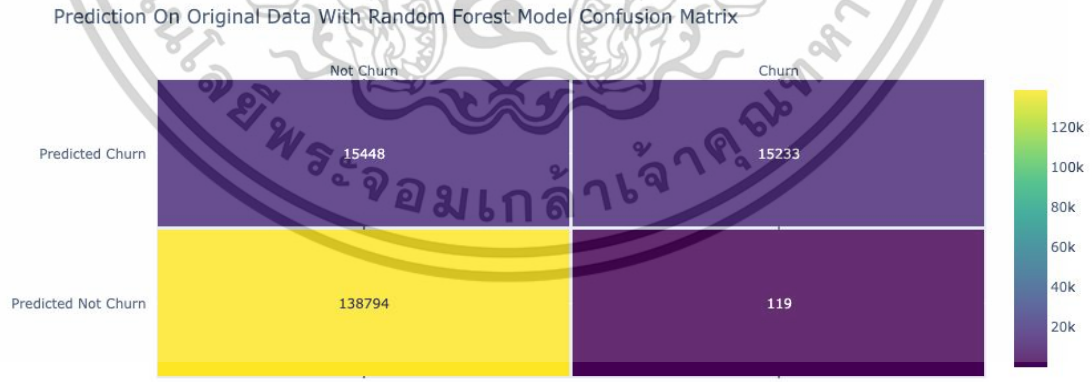
Model Results On Test Data

Model	Accuracy On Test Data	Precision On Test Data	Recall On Test Data	F1 Score On Test Data
Random Forest	0.91	0.99	0.5	0.66
AdaBoost	0.97	0.73	0.91	0.81
XG	0.99	0.88	0.98	0.93
GradientBoosting	0.99	0.87	0.97	0.92
CatBoost	0.92	0.99	0.54	0.69
Logistic	0.86	0.99	0.38	0.55

รูปที่ 4.25 ตารางผลลัพธ์ของข้อมูลทดสอบ

รูปที่ 4.25 แสดงผลลัพธ์ของโมเดลทั้งหมดเมื่อนำมาใช้ทดสอบกับข้อมูลทดสอบที่ถูกแบ่งไว้

2) Confusion Matrix ของโมเดล Random Forest



รูปที่ 4.26 Confusion Matrix สำหรับการทำนายบนข้อมูลทดสอบโดยใช้โมเดล Random Forest

รูปที่ 4.26 แสดง Confusion Matrix สำหรับโมเดล Random Forest ที่ใช้ทำนายการยกเลิกบริการ (churn)

ตารางนี้แบ่งออกเป็นสี่ส่วนได้แก่

ด้านซ้ายบนโมเดลทำนายว่ายกเลิก แต่มีค่าจริงไม่ยกเลิก = 15,448

ด้านขวาบนโมเดลทำนายว่ายกเลิก และมีค่าจริงยกเลิก = 15,233

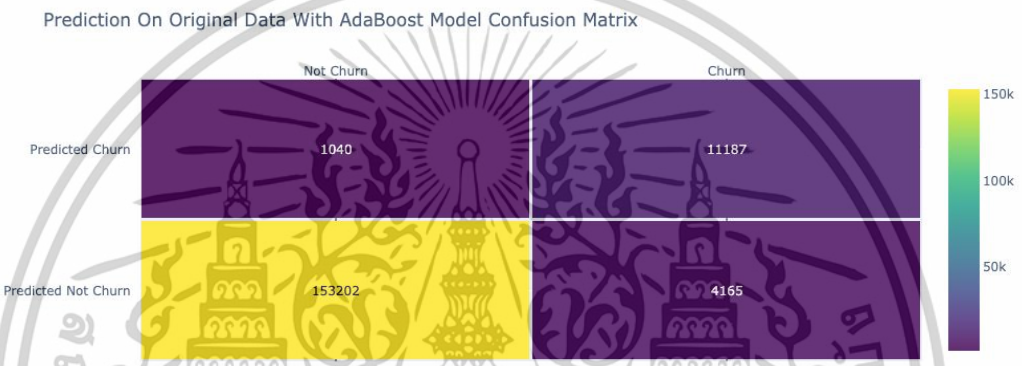
ด้านซ้ายล่างโมเดลทำนายว่าไม่ยกเลิก และมีค่าจริงไม่ยกเลิก = 138,794

ด้านขวาล่างโมเดลทำนายว่าไม่ยกเลิก แต่มีค่าจริงยกเลิก = 119

สีในตารางมีการไล่ระดับจากสีม่วงเข้มไปจนถึงสีเหลือง แสดงถึงจำนวนของตัวอย่างในแต่ละหมวดหมู่

3) Confusion Matrix ของโมเดล AdaBoost

4)



รูปที่ 4.27 Confusion Matrix สำหรับการทำนายบนข้อมูลทดสอบโดยใช้โมเดล AdaBoost

รูปที่ 4.27 แสดง Confusion Matrix สำหรับโมเดล AdaBoost ที่ใช้ทำนายการยกเลิกบริการ (churn)

ตารางนี้แบ่งออกเป็นสี่ส่วนได้แก่

ด้านซ้ายบนโมเดลทำนายว่ายกเลิก แต่มีค่าจริงไม่ยกเลิก = 1,040

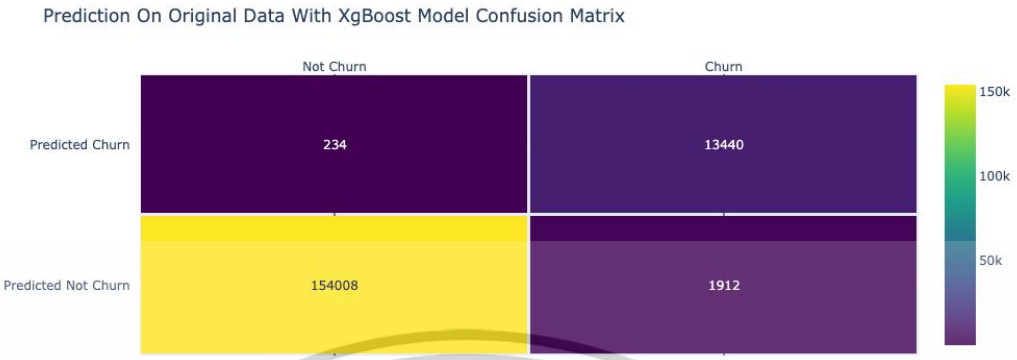
ด้านขวาบนโมเดลทำนายว่ายกเลิก และมีค่าจริงยกเลิก = 11,187

ด้านซ้ายล่างโมเดลทำนายว่าไม่ยกเลิก และมีค่าจริงไม่ยกเลิก = 153,202

ด้านขวาล่างโมเดลทำนายว่าไม่ยกเลิก แต่มีค่าจริงยกเลิก = 4,165

สีในตารางมีการไล่ระดับจากสีม่วงเข้มไปจนถึงสีเหลือง แสดงถึงจำนวนของตัวอย่างในแต่ละหมวดหมู่

5) Confusion Matrix ของโมเดล XGBoost



รูปที่ 4.28 Confusion Matrix สำหรับการทำนายบนข้อมูลทดสอบโดยใช้โมเดล XGBoost

รูปที่ 4.28 แสดง Confusion Matrix สำหรับโมเดล XGBoost ที่ใช้ทำนายการยกเลิกบริการ (churn)

ตารางนี้แบ่งออกเป็นสี่ส่วนได้แก่

ด้านซ้ายบนโมเดลทำนายว่ายกเลิก แต่มีค่าจริงไม่ยกเลิก = 234

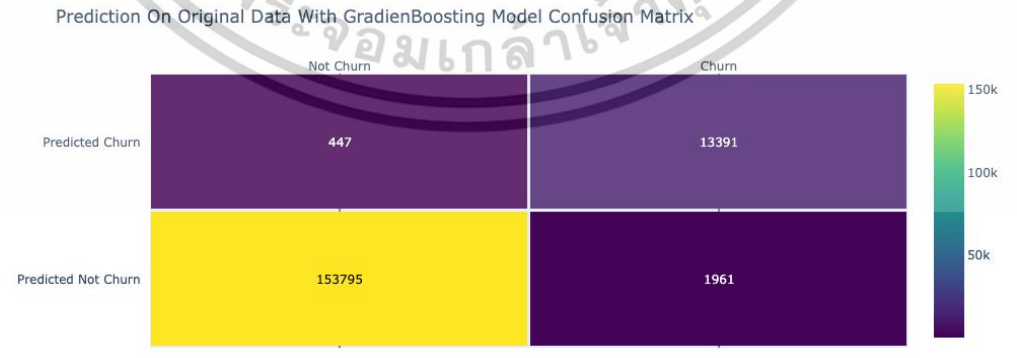
ด้านขวาบนโมเดลทำนายว่ายกเลิก และมีค่าจริงยกเลิก = 13,440

ด้านซ้ายล่างโมเดลทำนายว่าไม่ยกเลิก และมีค่าจริงไม่ยกเลิก = 154,008

ด้านขวาล่างโมเดลทำนายว่าไม่ยกเลิก แต่มีค่าจริงยกเลิก = 1,912

สีในตารางมีการไล่ระดับจากสีม่วงเข้มไปจนถึงสีเหลือง แสดงถึงจำนวนของตัวอย่างในแต่ละหมวดหมู่

6) Confusion Matrix ของโมเดล Gradient Boosting



รูปที่ 4.29 Confusion Matrix สำหรับการทำนายบนข้อมูลทดสอบโดยใช้โมเดล Gradient Boosting

รูปที่ 4.29 แสดง Confusion Matrix สำหรับโมเดล Gradient Boosting ที่ใช้ทำนายการยกเลิกบริการ (churn)

ตารางนี้แบ่งออกเป็นสี่ส่วนได้แก่

ด้านซ้ายบนโมเดลทำนายว่ายกเลิก แต่มีค่าจริงไม่ยกเลิก = 447

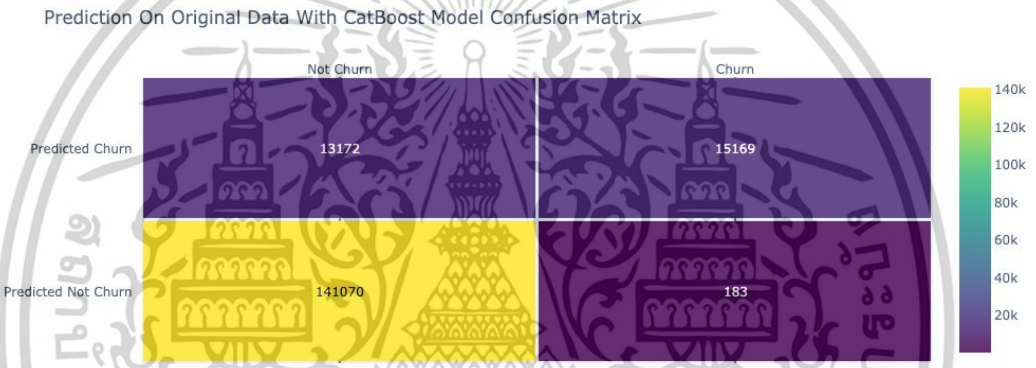
ด้านขวาบนโมเดลทำนายว่ายกเลิก และมีค่าจริงยกเลิก = 13,391

ด้านซ้ายล่างโมเดลทำนายว่าไม่ยกเลิก และมีค่าจริงไม่ยกเลิก = 153,795

ด้านขวาล่างโมเดลทำนายว่าไม่ยกเลิก แต่มีค่าจริงยกเลิก = 1,961

สีในตารางมีการไล่ระดับจากสีม่วงเข้มไปจนถึงสีเหลือง แสดงถึงจำนวนของตัวอย่างในแต่ละหมวดหมู่

7) Confusion Matrix ของโมเดล CatBoost



รูปที่ 4.30 Confusion Matrix สำหรับการทำนายบนข้อมูลทดสอบโดยใช้โมเดล CatBoost

รูปที่ 4.30 แสดง Confusion Matrix สำหรับโมเดล Gradient Boosting ที่ใช้ทำนายการยกเลิกบริการ (churn)

ตารางนี้แบ่งออกเป็นสี่ส่วนได้แก่

ด้านซ้ายบนโมเดลทำนายว่ายกเลิก แต่มีค่าจริงไม่ยกเลิก = 13,172

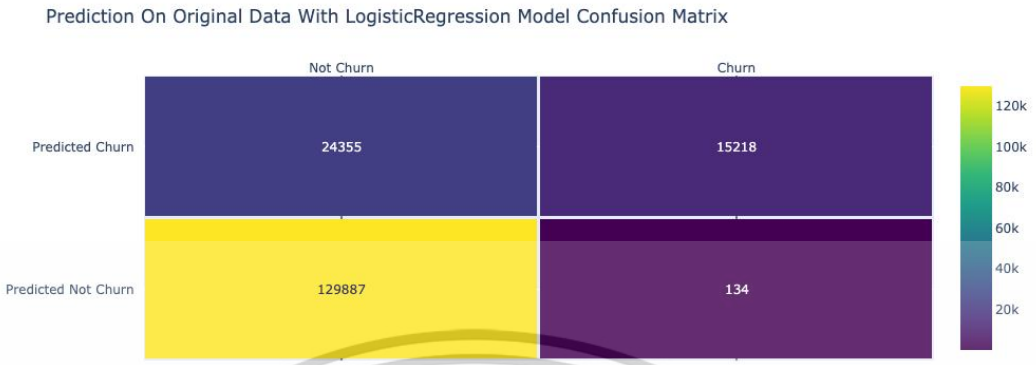
ด้านขวาบนโมเดลทำนายว่ายกเลิก และมีค่าจริงยกเลิก = 15,169

ด้านซ้ายล่างโมเดลทำนายว่าไม่ยกเลิก และมีค่าจริงไม่ยกเลิก = 141,070

ด้านขวาล่างโมเดลทำนายว่าไม่ยกเลิก แต่มีค่าจริงยกเลิก = 183

สีในตารางมีการไล่ระดับจากสีม่วงเข้มไปจนถึงสีเหลือง แสดงถึงจำนวนของตัวอย่างในแต่ละหมวดหมู่

8) Confusion Matrix ของโมเดล Logistic Regression



รูปที่ 4.31 Confusion Matrix สำหรับการทำนายบนข้อมูลทดสอบโดยใช้โมเดล Logistic Regression

รูปที่ 4.31 แสดง Confusion Matrix สำหรับโมเดล Logistic Regression ที่ใช้ทำนายการยกเลิกบริการ (churn)

ตารางนี้แบ่งออกเป็นสี่ส่วนได้แก่

ด้านซ้ายบนโมเดลทำนายว่ายกเลิก แต่มีค่าจริงไม่ยกเลิก = 24,355

ด้านขวาบนโมเดลทำนายว่ายกเลิก และมีค่าจริงยกเลิก = 15,218

ด้านซ้ายล่างโมเดลทำนายว่าไม่ยกเลิก และมีค่าจริงไม่ยกเลิก = 129,887

ด้านขวาล่างโมเดลทำนายว่าไม่ยกเลิก แต่มีค่าจริงยกเลิก = 134

สีในตารางมีการไล่ระดับจากสีม่วงเข้มไปจนถึงสีเหลือง แสดงถึงจำนวนของตัวอย่างในแต่ละหมวดหมู่

4.1.6 เปรียบเทียบประสิทธิภาพ

โดยทำการประเมินผลลัพธ์ด้วยค่าตัวชี้วัดต่าง ๆ ได้แก่ Accuracy, Precision, Recall, F1-score และ PR-Curve ซึ่งผลการทดลองสามารถสรุปได้ดังตารางที่ 4.1

ตารางที่ 4.1 ตารางแสดงผลการทดสอบของโมเดลต่าง ๆ ซึ่งใช้สำหรับทำนายการยกเลิกบริการของลูกค้า

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.91	0.99	0.5	0.66
AdaBoost	0.97	0.73	0.91	0.81
XGBoost	0.99	0.88	0.98	0.93
Gradient Boosting	0.99	0.87	0.97	0.92
CatBoost	0.92	0.99	0.54	0.69
Logistic Regression	0.86	0.99	0.38	0.55

จากผลลัพธ์ข้างต้นพบว่า โมเดล XGBoost ให้ผลลัพธ์ที่ดีที่สุด โดยมีค่า Accuracy เท่ากับ 0.99, Precision เท่ากับ 0.88, Recall เท่ากับ 0.98 และ F1-score เท่ากับ 0.93 แสดงให้เห็นว่าโมเดลนี้สามารถจำแนกลูกค้าที่จะยกเลิกบริการได้อย่างแม่นยำมากที่สุดเมื่อเทียบกับโมเดลอื่น

ในขณะที่ Gradient Boosting มีประสิทธิภาพเป็นลำดับรองลงมา โดยมีค่า Accuracy อยู่ที่ 0.99, Precision เท่ากับ 0.87, Recall เท่ากับ 0.97 และ F1-score เท่ากับ 0.92 ซึ่งยังคงเป็นโมเดลที่มีประสิทธิภาพสูงในการพยากรณ์การยกเลิกบริการของลูกค้า

โมเดลที่มีค่าความแม่นยำต่ำที่สุดคือ Logistic Regression ที่มีค่า Accuracy เพียง 0.86 และ Precision อยู่ที่ 0.99 แม้ว่าค่า Precision จะสูงถึง 0.99 แต่ค่า Recall และ F1-score มีค่าเพียง 0.55 แสดงให้เห็นว่าประสิทธิภาพโดยรวมของโมเดลยังไม่ดีพอสำหรับการใช้งานจริง

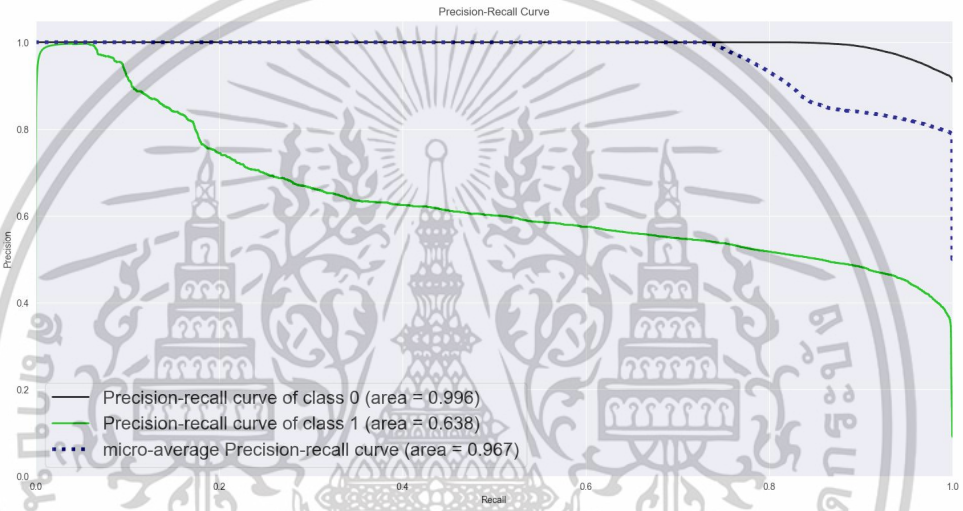
4.2 การอภิปรายผล

จากผลการทดลองสามารถวิเคราะห์ได้ว่าโมเดล XGBoost เป็นตัวเลือกที่ดีที่สุดสำหรับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารเนื่องจากสามารถสร้างสมดุลระหว่าง Precision และ Recall ได้อย่างดีเยี่ยม ทำให้เกิดค่า F1-score ที่สูงมากถึง 0.93 ซึ่งสะท้อนถึงความสามารถของโมเดลในการพยากรณ์กลุ่มลูกค้าที่มีแนวโน้มจะยกเลิกบริการได้อย่างถูกต้อง ในทางตรงกันข้าม แม้ว่าโมเดล AdaBoost และ Gradient Boosting จะมีค่า Recall สูงถึง 0.91 และ 0.97 แต่มีค่าความแม่นยำ (Accuracy) และค่าความเชื่อมั่นในการทำนาย (Precision) ยังค่อนข้างต่ำ จึงอาจนำไปสู่การเกิด False Positive สูง ซึ่งอาจทำให้เกิดความเข้าใจผิดในการนำผลลัพธ์ไปใช้

จากการวิเคราะห์ผลลัพธ์ในครั้งนี้ แนะนำให้ใช้ XGBoost เป็นโมเดลหลักสำหรับการพยากรณ์การยกเลิกบริการของลูกค้าธนาคาร เนื่องจากมีค่าตัวชี้วัดที่ดีที่สุดเมื่อเปรียบเทียบกับโมเดลอื่น ๆ อย่างไรก็ตาม ควรมีการตรวจสอบความสามารถของโมเดลเพิ่มเติมในสถานการณ์ที่มีข้อมูลเปลี่ยนแปลง รวมถึงทดสอบกับชุดข้อมูลใหม่เพื่อให้มั่นใจว่าโมเดลสามารถใช้งานได้จริงในสภาพแวดล้อมของธุรกิจธนาคาร

4.2.1 Logistic Regression

โมเดลพื้นฐานซึ่งใช้สำหรับจำแนกประเภท (Classification) โดยใช้ฟังก์ชันโลจิสติกในการคำนวณความน่าจะเป็นของแต่ละคลาส ผลการทดลองพบว่า Logistic Regression มีค่า Accuracy 0.86 และ Precision 0.99 แต่ค่า Recall และ F1-score ยังคงน้อยที่สุด



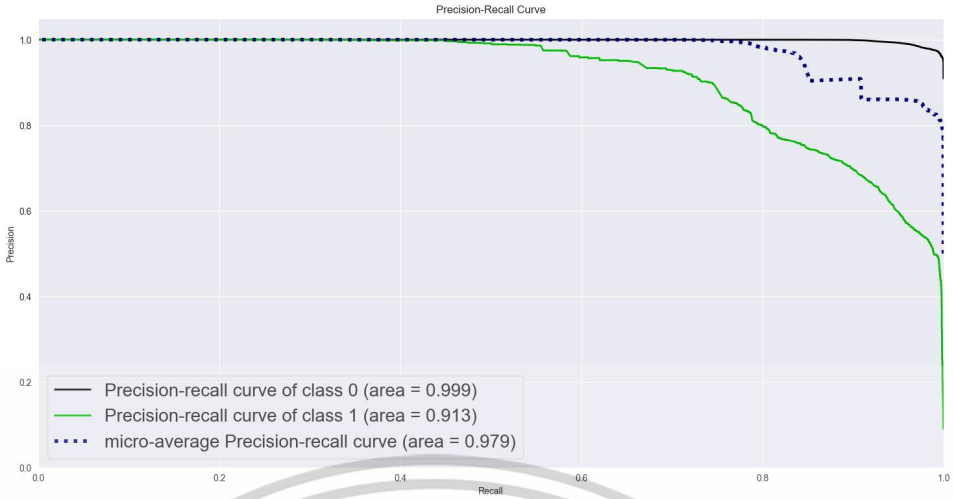
รูปที่ 4.32 กราฟแสดง Precision-Recall ของ Logistic Regression

รูปที่ 4.32 แสดง Precision-Recall (PR) Curve ของโมเดล Logistic Regression ที่แสดงให้เห็นประสิทธิภาพของโมเดลในการจำแนกแต่ละคลาส

- คลาส 0 (เส้นสีดำ, AUC = 0.996) โมเดลทำงานได้ดีมากในการจำแนกคลาสนี้
- คลาส 1 (เส้นสีเขียว, AUC = 0.638) โมเดลยังคงมีปัญหาในการจำแนกคลาสนี้
- ค่า Micro-average ค่อนข้างสูง (AUC = 0.967) แสดงให้เห็นว่าโมเดลโดยรวมยังทำงานได้ดี แม้ว่าจะมีปัญหาในบางคลาส

4.2.2 Random Forest

Random Forest เป็นโมเดลที่ใช้การเรียนรู้แบบการรวมกันของต้นไม้ตัดสินใจ (Decision Trees) หลายต้น ทำให้มีความสามารถในการจัดการข้อมูลที่มีความซับซ้อนได้ดี แต่จากผลการทดลองพบว่าให้ค่า Recall 0.50 และ F1-score 0.66 ซึ่งอาจเป็นเพราะข้อมูลมีความไม่สมดุล (Imbalanced Data) ทำให้โมเดลให้ค่า Precision สูงสุด 0.99 แต่มีค่า Recall ต่ำ



รูปที่ 4.33 กราฟแสดง Precision-Recall ของ Random Forest

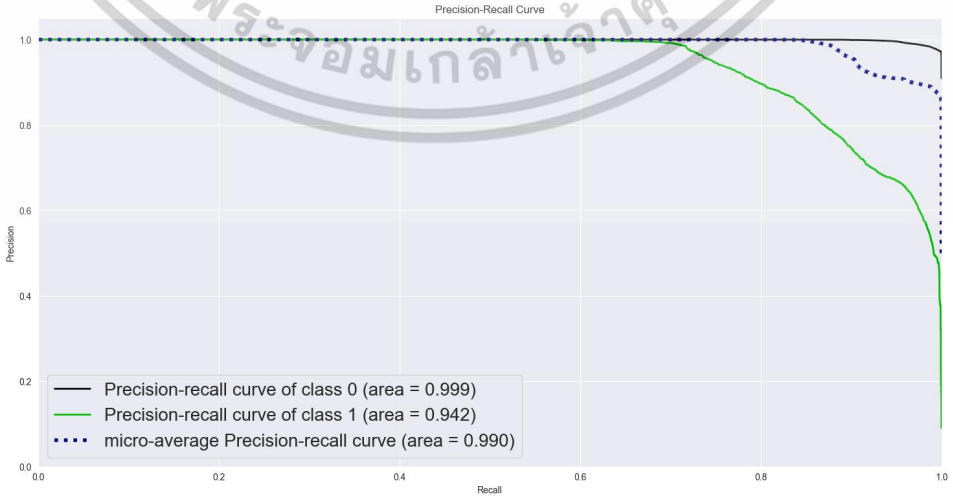
รูปที่ 4.33 แสดงผลลัพธ์ดังนี้

- คลาส 0 (เส้นสีดำ, AUC = 0.999) โมเดลสามารถจำแนกคลาสนี้ได้อย่างแม่นยำเกือบสมบูรณ์แบบ
- คลาส 1 (เส้นสีเขียว, AUC = 0.913) โมเดลสามารถจำแนกคลาสนี้ได้ดี
- ค่า Micro-Average (เส้นประสีน้ำเงิน, AUC = 0.979) เป็นการประเมินผลรวมของความสามารถของโมเดลในทุกคลาส

จากกราฟนี้ แสดงให้เห็นว่าโมเดลจำแนกได้ดีในทุกคลาส

4.2.3 CatBoost

CatBoost เป็นโมเดล Boosting ที่ถูกออกแบบมาให้ทำงานได้ดีบนข้อมูลที่มีลักษณะเป็น Categorical Data โดยเฉพาะ ผลที่ได้จากการทดลองแสดงให้เห็นว่า CatBoost มี Accuracy 0.92 และค่า Precision 0.99 ซึ่งอยู่ในระดับเดียวกับ Random Forest



รูปที่ 4.34 กราฟแสดง Precision-Recall ของ CatBoost

รูปที่ 4.34 แสดงกราฟโมเดล CatBoost ทำนายคลาส 0 ได้ดีมาก เส้นสีดำที่แทบจะอยู่บนสุดของกราฟ แสดงว่า Precision และ Recall ของคลาส 0 สูงมาก (เกือบ 1 ตลอด) พื้นที่ใต้กราฟ (AUC) 0.999 โมเดลแทบไม่มีข้อผิดพลาดเลยในการแยกแยะคลาส 0

4.2.4 AdaBoost

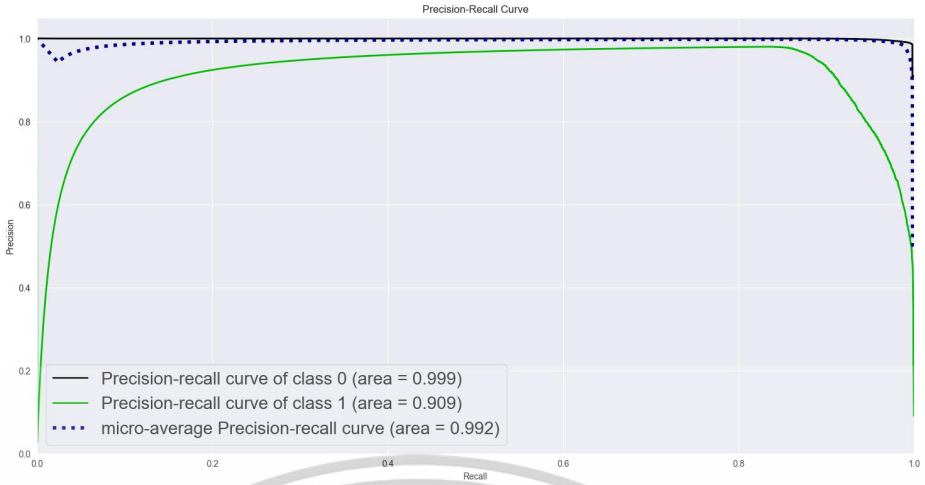
AdaBoost เป็นโมเดลที่ใช้แนวคิดของ Boosting โดยสร้างชุดของ Weak Learners เช่น ต้นไม้ตัดสินใจขนาดเล็ก และทำการปรับปรุงน้ำหนักของตัวอย่างที่คาดการณ์ผิดในแต่ละรอบ ผลลัพธ์จากการทดลองแสดงให้เห็นว่า AdaBoost เป็นโมเดลที่เกือบจะดีในการทำนาย โดยมี Accuracy สูงถึง 0.97 และค่า Precision และ Recall เกือบสมบูรณ์แบบที่ 0.73 และ 0.91

รูปที่ 4.35 กราฟแสดง Precision-Recall ของ AdaBoost

รูปที่ 4.35 แสดงกราฟโมเดล AdaBoost นี้ทำงานได้ดีสำหรับคลาส 0 (AUC = 0.999) ซึ่งหมายความว่าโมเดลสามารถจำแนกตัวอย่างที่เป็นคลาส 0 ได้แม่นยำมาก แต่ประสิทธิภาพรองลงมาสำหรับ Class 1 (AUC = 0.921) แสดงว่าโมเดลอาจมี Bias ไปทางคลาส 0 ค่า Micro-average 0.997 สูงมาก บ่งบอกว่าถ้าดูภาพรวม โมเดลอาจทำงานได้ดี แต่ถ้าชุดข้อมูลไม่สมดุล (imbalanced dataset) โมเดลอาจจะมีแนวโน้มให้ความสำคัญกับคลาส 0 มากกว่า

4.2.5 Gradient Boosting

Gradient Boosting เป็นเทคนิคที่ใช้การเรียนรู้แบบ Boosting เช่นเดียวกับ XGBoost และ AdaBoost แต่ไม่มีการปรับแต่งให้ทำงานเร็วเท่ากับ XGBoost ผลการทดลองพบว่า Gradient Boosting มี Accuracy 0.99 และค่า Precision 0.87 ซึ่งใกล้เคียงกับ XGBoost



รูปที่ 4.36 กราฟแสดง Precision-Recall ของ Gradient Boosting

รูปที่ 4.36 แสดงกราฟโมเดล Gradient Boosting สามารถจำแนกคลาส 0 ได้ดีมาก (AUC = 0.999) แต่ประสิทธิภาพรองลงมาสำหรับ Class 1 (AUC = 0.909) แสดงว่าโมเดลอาจมี Bias ไปทางคลาส 0 ค่า Micro-average 0.992 สูงมาก บ่งบอกว่าถ้าดูภาพรวม โมเดลอาจทำงานได้ดี แต่ถ้าชุดข้อมูลไม่สมดุล (imbalanced dataset) โมเดลอาจจะมีแนวโน้มให้ความสำคัญกับคลาส 0 มากกว่า

4.2.6 XGBoost

XGBoost เป็นอัลกอริทึมที่พัฒนาต่อยอดจาก Gradient Boosting โดยมีการปรับปรุงให้มีประสิทธิภาพสูงขึ้น ซึ่งในการทดลองนี้ XGBoost ให้ค่า Recall 0.98 และ F1-score 0.93 ซึ่งมีประสิทธิภาพสูงที่สุดเมื่อเปรียบเทียบกับทุกๆ โมเดล

รูปที่ 4.37 กราฟแสดง Precision-Recall ของ XGBoost

รูปที่ 4.37 แสดงกราฟโมเดล XGBoost สามารถทำนายคลาส 1 และ คลาส 0 ได้แม่นยำมาก ประสิทธิภาพโดยรวมของโมเดลดีมาก และค่า Micro-average AUC = 0.999 มีค่าใกล้เคียง 1.00 แสดงว่าโมเดลโดยรวมดีมาก

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

ในงานวิจัยนี้ การวิเคราะห์เชิงเปรียบเทียบของอัลกอริทึมการเรียนรู้ของเครื่องสำหรับการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคาร มีวัตถุประสงค์เพื่อใช้ข้อมูลที่มีอยู่ในการประเมินและทำนายว่าลูกค้ารายใดมีแนวโน้มที่จะยกเลิกบริการธนาคารในอนาคต ข้อมูลซึ่งใช้สำหรับวิจัยประกอบด้วยข้อมูลทางประชากรศาสตร์ของลูกค้า ประวัติการทำธุรกรรม การใช้บริการธนาคารต่าง ๆ รวมถึงพฤติกรรมการใช้ผลิตภัณฑ์หรือบริการของธนาคาร

การประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่อง เช่น Logistic Regression, Random Forest, CatBoost, AdaBoost, Gradient Boosting และ XGBoost ช่วยเพิ่มความแม่นยำในการคาดการณ์การยกเลิกบริการ โดยการประเมินผลการทำงานด้วย Accuracy, Precision, Recall, F1-Score และ Precision-Recall Curve ได้ผลที่น่าพอใจ

5.1 สรุปผลการวิจัย

การวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาการใช้แบบจำลองการเรียนรู้ของเครื่อง ในการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารโดยทำการเปรียบเทียบประสิทธิภาพของ 6 โมเดล ได้แก่ Random Forest, AdaBoost, XGBoost, Gradient Boosting, CatBoost และ Logistic Regression

ผลการทดลองพบว่า XGBoost และ Gradient Boosting เป็นโมเดลที่มีประสิทธิภาพสูงสุด โดยมีค่า Recall และ F1 Score สูงถึง 0.98 และมากกว่า 0.93 ตามลำดับ แสดงถึงความแม่นยำและความสมดุลระหว่าง Precision และ Recall ที่ดีเยี่ยม

โมเดล AdaBoost มีค่า Recall 0.91 ซึ่งเหมาะสำหรับงานที่ต้องการลดการพลาดข้อมูลสำคัญ แม้ Precision จะต่ำกว่าบางโมเดล

ปัจจัยที่อาจมีผลต่อประสิทธิภาพของโมเดล ได้แก่

1) ความไม่สมดุลของข้อมูล (Imbalanced Data) ซึ่งอาจทำให้โมเดลบางตัวมีค่า Precision ต่ำ

2) ประสิทธิภาพของอัลกอริทึมในการเรียนรู้และประมวลผลข้อมูล

3) การตั้งค่าพารามิเตอร์ของแต่ละโมเดลที่อาจยังไม่เหมาะสม

5.2 ข้อเสนอแนะ

5.2.1 ข้อเสนอแนะในการปรับปรุงโมเดล

5.2.1.1 จัดการกับความไม่สมดุลของข้อมูล (Imbalanced Data)

1) ใช้เทคนิค Undersampling เพื่อลดจำนวนตัวอย่างจากกลุ่มที่มีมากเกินไป

2) ใช้การปรับแต่งค่าการให้ความสำคัญของคลาส (Class Weight) เพื่อให้โมเดลให้ความสำคัญกับกลุ่มที่มีจำนวนน้อย

5.2.1.2 การปรับแต่งพารามิเตอร์ (Hyperparameter Tuning)

1) ทดลองใช้ Grid Search เพื่อหาค่าพารามิเตอร์ที่เหมาะสม

2) ใช้ Cross-Validation เพื่อลดโอกาสของการเกิด Overfitting

5.2.2 ข้อเสนอแนะในการนำไปใช้จริง

5.2.2.1 นำโมเดลที่ดีที่สุดไปใช้ในการคาดการณ์ลูกค้าที่มีแนวโน้มจะยกเลิกบริการ

โมเดล XGBoost สามารถนำไปใช้เป็นระบบแจ้งเตือนให้กับฝ่ายบริการลูกค้า หรือฝ่ายบริหารความสัมพันธ์ลูกค้า (CRM) เพื่อดำเนินมาตรการรักษาลูกค้าไว้ก่อนที่พวกเขาจะตัดสินใจยกเลิกบริการ

5.2.2.2 วิเคราะห์ปัจจัยที่ทำให้ลูกค้ายกเลิกบริการ

ใช้ผลลัพธ์ของโมเดลร่วมกับการวิเคราะห์ข้อมูลอื่น ๆ เช่น พฤติกรรมการใช้บริการ, ระยะเวลาที่เป็นลูกค้า, การร้องเรียน หรือความพึงพอใจของลูกค้า เพื่อหาแนวทางปรับปรุงการให้บริการ

5.2.2.3 นำผลลัพธ์ไปใช้ในการวางแผนกลยุทธ์การตลาด

ใช้ข้อมูลที่ได้จากการวิเคราะห์เพื่อออกแบบโปรโมชัน หรือมาตรการจูงใจลูกค้าให้อยู่กับธนาคารต่อไป เช่น ข้อเสนอพิเศษ, ส่วนลดค่าธรรมเนียม หรือบริการให้คำปรึกษาทางการเงินแบบเฉพาะบุคคล

บรรณานุกรม

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

Li, Y., & Chen, X. (2019). CatBoost: Gradient boosting with categorical features support. arXiv preprint arXiv:1905.08408.

Schapire, R. E. (1990). The strength of weak learnability. Machine Learning, 5(2), 197-227.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189-1232.

Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R News, 2(3), 18-22.

XGBoost Contributors. (2016). XGBoost: A scalable and efficient gradient boosting library.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(5), e1249.

Chollet, F. (2017). Deep Learning with Python. Manning Publications.

ภาคผนวก

ภาคผนวก ก
ชุดข้อมูลซึ่งใช้สำหรับวิเคราะห์

ข้อมูลชุดข้อมูลซึ่งใช้สำหรับวิเคราะห์นี้ถูกรวบรวมจากธนาคารแห่งหนึ่ง โดยมีข้อมูลลูกค้าทั้งที่ยังคงใช้บริการและที่ยกเลิกบริการ ข้อมูลที่ใช้ประกอบด้วยตัวแปรต่าง ๆ เช่น อายุ เพศ รายได้ สถานภาพสมรส ประวัติการใช้บริการทางการเงิน และจำนวนผลิตภัณฑ์ที่ใช้ รายละเอียดของแต่ละคุณลักษณะในชุดข้อมูลแสดงอยู่ในตารางด้านล่าง

ชื่อคุณลักษณะ	คำอธิบาย
cust_no	รหัสลูกค้า
open_acct_date	วันที่เปิดบัญชี
email_validate_ind	การยืนยันอีเมล
verify_sub_channel	ช่องทางยืนยันตัวตน
current_age	อายุของลูกค้า (ปี)
gender	เพศของลูกค้า (ชาย/หญิง)
occupation	อาชีพ
income_source	แหล่งที่มารายได้
income_range	ช่วงของรายได้
marital_status	สถานภาพสมรส
acct_status	ลูกค้ายกเลิกบริการหรือไม่ (1 = ไม่ยกเลิก, 0 = ยกเลิก)
is_open_tgt	การใช้งานผลิตภัณฑ์อื่น
phone_os	ระบบปฏิบัติการมือถือ (Android, iOS)
max_login_date	วันที่เข้าใช้งานล่าสุด
luckydrawy2024	โปรโมชัน luckydrawy2024
s99_24	โปรโมชัน s99_24
conline_event2024	โปรโมชัน conline_event2024
no_of_txn2024_c	จำนวนธุรกรรม
trxn_amt2024_c	ปริมาณธุรกรรม
bal_31dec24	ยอดเงินล่าสุด
first_deposit_date	วันที่เอาเงินเข้าครั้งแรก
First_deposit_amt	จำนวนเงินที่ฝากครั้งแรก

ภาคผนวก ข
รายละเอียดของอัลกอริทึมที่ใช้

การศึกษานี้ใช้โมเดลการเรียนรู้ของเครื่องหลายประเภทที่มีความสามารถแตกต่างกันเพื่อการทำนายการยกเลิกบัญชีเงินฝากของลูกค้าธนาคารโดยการเลือกใช้ 6 โมเดลหลักที่มีความหลากหลายทั้งในแง่ของแนวทางการเรียนรู้และการจัดการข้อมูลเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดในการทำนายการยกเลิกบริการ โดยรายละเอียดของโมเดลต่าง ๆ มีดังนี้

1. ป่าแบบสุ่ม (Random Forest)

การทำงาน Random Forest เป็นเทคนิคการเรียนรู้แบบ ensemble ที่สร้างหลาย ๆ ต้นไม้ตัดสินใจ (Decision Trees) แล้วนำมารวมผลเพื่อให้ได้ผลลัพธ์ที่แม่นยำขึ้น การตัดสินใจของโมเดลจะพิจารณาจากผลรวมของการทำนายจากต้นไม้แต่ละต้น โดยการตัดสินใจของแต่ละต้นไม้จะใช้ข้อมูลที่แตกต่างกันในการฝึก ดังนั้นโมเดลนี้มีความสามารถในการลดปัญหา overfitting ที่มักเกิดขึ้นจากการใช้ต้นไม้เพียงต้นเดียว

ข้อดี

-ประสิทธิภาพสูงในการจัดการกับข้อมูลที่มีความซับซ้อน

-สามารถจัดการกับข้อมูลที่มีลักษณะหลากหลายทั้งเชิงปริมาณและเชิงคุณลักษณะได้ดี

ข้อจำกัด

-การฝึกโมเดลที่มีต้นไม้จำนวนมากอาจทำให้การคำนวณช้าลง

2. AdaBoost (Adaptive Boosting)

การทำงาน AdaBoost เป็นอัลกอริทึมการเรียนรู้แบบ ensemble ที่ปรับปรุงจากการเรียนรู้เชิงเสริม (boosting) ที่จะเพิ่มน้ำหนักให้กับตัวอย่างข้อมูลที่โมเดลทำนายผิด จากนั้นจะฝึกโมเดลใหม่เพื่อให้โฟกัสที่ตัวอย่างเหล่านั้นโดยเฉพาะ วิธีนี้ช่วยให้โมเดลมีความแม่นยำมากขึ้น

ข้อดี

-สามารถทำงานได้ดีแม้จะใช้โมเดลพื้นฐานที่มีความสามารถต่ำ (weak learners)

-ใช้เวลาฝึกโมเดลน้อยกว่าเมื่อเทียบกับ Random Forest

ข้อจำกัด

-การเพิ่มน้ำหนักให้กับข้อมูลที่ทำนายผิดอาจทำให้โมเดลไวต่อการโอเวอร์ฟิต (Overfitting) หากไม่ระมัดระวัง

3. Extreme Gradient Boosting (XGBoost)

การทำงาน XGBoost เป็นอัลกอริทึมที่ได้รับการปรับปรุงจาก Gradient Boosting ซึ่งใช้การเรียนรู้แบบ boosting โดยจะสร้างโมเดลใหม่โดยการให้ความสำคัญกับข้อผิดพลาดจากโมเดลก่อนหน้านี้ในแต่ละขั้นตอน XGBoost มักจะใช้การควบคุมการเรียนรู้ที่ซับซ้อน เช่น การทำ regularization เพื่อป้องกันการโอเวอร์ฟิต (Overfitting)

ข้อดี

-ให้ผลลัพธ์ที่มีประสิทธิภาพสูงในการคาดการณ์

-รองรับการปรับแต่งหลายระดับเพื่อควบคุมการเรียนรู้และลดการเกิดโอเวอร์ฟิต (Overfitting)

ข้อจำกัด

-การตั้งค่าพารามิเตอร์อาจซับซ้อนและต้องการทดสอบและการปรับแต่งหลายขั้นตอน

4. Gradient Boosting

การทำงาน Gradient Boosting เป็นเทคนิคการเรียนรู้แบบ boosting ที่จะฝึกโมเดลใหม่บนข้อผิดพลาดที่เกิดจากโมเดลก่อนหน้านี้ โดยใช้การคำนวณ gradient descent เพื่อหาความแตกต่างระหว่างผลลัพธ์ที่ทำนายและค่าจริง โดยการฝึกทีละขั้นจะช่วยลดข้อผิดพลาดและพัฒนาโมเดลให้มีความแม่นยำมากขึ้น

ข้อดี

-ผลลัพธ์ที่มีความแม่นยำสูง สามารถใช้กับปัญหาที่มีความซับซ้อนได้ดี

-มีความยืดหยุ่นสูงในการปรับแต่งพารามิเตอร์

ข้อจำกัด

-การฝึกโมเดลต้องใช้เวลานานหากมีข้อมูลขนาดใหญ่

5. CatBoost

การทำงาน CatBoost เป็นอัลกอริทึมการเรียนรู้แบบ gradient boosting ที่ออกแบบมาเพื่อจัดการกับข้อมูลประเภทหมวดหมู่ (categorical data) โดยไม่ต้องการแปลงข้อมูลหมวดหมู่ให้เป็นตัวเลข (one-hot encoding) โมเดลนี้จึงเหมาะสำหรับข้อมูลที่มีหลายหมวดหมู่และมีความซับซ้อนในการแปลงข้อมูล

ข้อดี

-สามารถจัดการกับข้อมูลหมวดหมู่ได้โดยตรงโดยไม่ต้องแปลง

-มีประสิทธิภาพสูงในการลด overfitting และเพิ่มประสิทธิภาพ

ข้อจำกัด

-อาจต้องใช้เวลาฝึกโมเดลค่อนข้างนานหากข้อมูลมีขนาดใหญ่

6. โลจิสติกรีเกรสชัน (Logistic Regression)

การทำงาน โลจิสติกรีเกรสชันเป็นโมเดลซึ่งใช้สำหรับจำแนกประเภท โดยการทำนายความน่าจะเป็นของกลุ่มเป้าหมาย เช่น การทำนายการยกเลิกบริการของลูกค้า โดยใช้ฟังก์ชันโลจิสติกในการเปลี่ยนค่าที่ได้จากการคำนวณเชิงเส้นให้เป็นค่าความน่าจะเป็นที่อยู่ระหว่าง 0 และ 1

ข้อดี

-โมเดลที่เข้าใจง่ายและใช้งานง่าย

-ใช้เวลาในการฝึกและคำนวณน้อยเมื่อเทียบกับโมเดลที่ซับซ้อนกว่า

ข้อจำกัด

-ไม่สามารถจัดการกับข้อมูลที่มีความซับซ้อนได้ดีเท่ากับโมเดลที่ใช้ ensemble methods หรือ neural networks

ภาคผนวก ค
ตัวอย่างโปรแกรมสำหรับการประเมินผล

ด้านล่างคือตัวอย่างโปรแกรม Python ที่ใช้สำหรับการฝึกและประเมินผลโมเดล

```
# %%
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as ex
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly.subplots import make_subplots
import plotly.offline as pyo
pyo.init_notebook_mode()
sns.set_style('darkgrid')
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.ensemble import RandomForestClassifier,AdaBoostClassifier,GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from xgboost.sklearn import XGBClassifier
from sklearn.linear_model import LogisticRegression
from catboost import CatBoostClassifier
from lightgbm import LGBMClassifier
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler,MinMaxScaler
from sklearn.metrics import f1_score as f1
from sklearn.metrics import recall_score as recall
from sklearn.metrics import accuracy_score as accuracy
from sklearn.metrics import precision_score as precision
from sklearn.metrics import confusion_matrix
import scikitplot as skplt

plt.rc('figure',figsize=(18,9))
# %pip install imbalanced-learn
from imblearn.over_sampling import SMOTE
```

```
from imblearn.under_sampling import RandomUnderSampler

# %%
#parameter
showFig = False
debug = False

smote_size = 0 #if 0 default 50:50 [0.0-1.0]


# %%
df = pd.read_csv('Data_is_camp_lastest.csv')
pd.set_option('display.max_rows',None)
df.head(10)

# %%
# df.sample(10)

# %%
df.info()

# %%
df.drop(columns =
        [
         'fisrtdate_join',
         'close_acct',
],
        inplace=True)

# %%
df['verify_sub_channel'] = df['verify_sub_channel'].fillna("Unknown")
df['phone_os'] = df['phone_os'].fillna("Unknown")

# %%
df['lld'] = pd.to_datetime(df['last_login_date'], infer_datetime_format=True ,
format='%m/%d/%Y')
df['difference_lld'] = (pd.Timestamp('2025-02-21') - df['lld']) / np.timedelta64(1, 'D')
df['difference_lld'] = df['difference_lld'].fillna(-1.0)
```

```
df = df.drop([ 'lld', 'last_login_date'], axis=1)
df.head()

# %%
df['oad'] = pd.to_datetime(df['open_acct_date'], infer_datetime_format=True , format='%m/%d/%Y')
df['difference_oad'] = (pd.Timestamp('2025-02-21') - df['oad']) / np.timedelta64(1, 'D')
df['oad_mld'] = df['difference_oad'] - df['difference_lld']
df = df.drop([ 'oad', 'open_acct_date'], axis=1)
df.head()

# %%
df['verify_sub_channel'] = df['verify_sub_channel'].fillna("Unknown")
df['phone_os'] = df['phone_os'].fillna("Unknown")

# %%
df['luckydrawy2024'] = df['luckydrawy2024'].fillna(0.0)
df['c99_2024'] = df['c99_2024'].fillna(0.0)
df['c_online_event2024'] = df['c_online_event2024'].fillna(0.0)
df['no_of_txn2024_c'] = df['no_of_txn2024_c'].fillna(0.0)
df['trxn_amt2024_c'] = df['trxn_amt2024_c'].fillna(0.0)
df['bal_31dec24'] = df['bal_31dec24'].fillna(0.0)

df['first_deposit_amt'] = df['first_deposit_amt'].fillna(0.0)

# %%
df['fdd'] = pd.to_datetime(df['first_deposit_date'], infer_datetime_format=True , format='%m/%d/%Y')
df['difference_fdd'] = (pd.Timestamp('2025-02-21') - df['fdd']) / np.timedelta64(1, 'D')
df['difference_fdd'] = df['difference_fdd'].fillna(0.0)
df['nb_months_fdd'] = (df['difference_fdd'] / 30.44).astype(int)
df = df.drop(['difference_fdd', 'fdd', 'first_deposit_date'], axis=1)
df.head()

# %%
null_mask = df.isnull().any(axis=1)
null_rows = df[null_mask]
```

```
print(null_rows)

# %%
df['occupation'].value_counts().head(3).sort_values()

# %%
df['income_range'].value_counts().head(3).sort_values()

# %%
def  occu_null_check(cha):
    if pd.isnull(cha):
        return 'Company Employee'
    else:
        return cha

def  income_null_check(cha):
    if pd.isnull(cha):
        return '15,000 - 29,999 Baht'
    else:
        return cha

df['occupation'] = df['occupation'].apply(occu_null_check)
df['income_range'] = df['income_range'].apply(income_null_check)

# %%
# df.dropna(subset=['occupation'], inplace=True)
df.isna().sum()

# %%
from plotly.subplots import make_subplots
import plotly.graph_objs as go
import plotly.express as ex

# %%
df['acct_status'].value_counts()

# %%
df['verify_sub_channel'].value_counts().head(5).sort_values()
```

```
# %%
def  range_check(text):
    if text.strip() == "KBANK":
        return text.strip()
    elif text.strip() == "Branch" :
        return text.strip()
    elif text.strip() == "SCB" :
        return text.strip()
    elif text.strip() == "BAY" :
        return text.strip()
    elif text.strip() == "7-11" :
        return text.strip()
    else :
        return 'Others'

df['new_verify_sub_channel'] = df['verify_sub_channel'].apply(range_check)

# %%
df['new_verify_sub_channel'].value_counts()

# %%
df.insert(0, 'acct_status', df.pop('acct_status'))
df_copy = df
df_copy['acct_status'] = df_copy['acct_status'].replace({0:1,1:0})
df_copy['gender'] = df_copy['gender'].replace({'Female':1,'Male':0})
df_copy['phone_os'] = df_copy['phone_os'].replace({'iOS':1,'Android':0,'Unknown':-1})
df_copy['email_validate_ind'] = df_copy['email_validate_ind'].replace({'Y':1,'N':0})
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['occupation'])],axis=1)
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['income_range'])],axis=1)
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['income_source'])],axis=1)
df_copy = pd.concat([df_copy,pd.get_dummies(df_copy['marital_status'])],axis=1)
df_copy                                                                    =
pd.concat([df_copy,pd.get_dummies(df_copy['new_verify_sub_channel'])],axis=1)
df_copy.drop(columns =
        [
            'verify_sub_channel',
            'new_verify_sub_channel',
```

```
            'occupation',
            'income_source',
            'income_range',
            'marital_status',
         'cust_no'],
        inplace=True)

# %%
df_copy['phone_os'].value_counts()

# %%
df_copy.info()

# %%
df_copy['acct_status'].value_counts()

# %%
fig = make_subplots(rows=2, cols=1,shared_xaxes=True,subplot_titles=('Perason
Correaltion', 'Spearman Correaltion'))
colorscale = 'turbo'


s_val =df_copy.corr('pearson')
s_idx = s_val.index
s_col = s_val.columns
s_val = s_val.values
fig.add_trace(

go.Heatmap(x=s_col,y=s_idx,z=s_val,name='pearson',showscale=False,xgap=0.7,ygap=
0.7,colorscale=colorscale),
    row=1, col=1
)


s_val =df_copy.corr('spearman')
s_idx = s_val.index
s_col = s_val.columns
s_val = s_val.values
```

```
fig.add_trace(
    go.Heatmap(x=s_col,y=s_idx,z=s_val,xgap=0.7,ygap=0.7,colorscale=colorscale),
    row=2, col=1
)
fig.update_layout(
    hoverlabel=dict(
        bgcolor="white",
        font_size=16,
        font_family="Rockwell"
    )
)
fig.update_layout(height=1080, width=1920, title_text="Numeric Correlations")
fig.show()

# %%
df_copy.head()

# %%
df_copy.isna().sum()

# %%
df_copy['acct_status'].value_counts()

# %%
oversample = SMOTE(sampling_strategy = 0.1, random_state=42)
X,          y      =          oversample.fit_resample(df_copy[df_copy.columns[1:]],
df_copy[df_copy.columns[0]])
usampled_df = X.assign(Churn = y)

# %%
usampled_df['Churn'].value_counts()

# %%
usampled_df.info()

# %%
usampled_df.head()
```

```
# %%
usampled_df['Churn'].value_counts()

# %%
usampled_df[usampled_df.columns[16:59]].info()

# %%
ohe_data =usampled_df[usampled_df.columns[16:-1]].copy()

usampled_df = usampled_df.drop(columns=usampled_df.columns[16:-1])

# %%
fig = make_subplots(rows=2, cols=1,shared_xaxes=True,subplot_titles=('Perason
Correaltion', 'Spearman Correaltion'))

colorscale = 'turbo'

s_val =usampled_df.corr('pearson')
s_idx = s_val.index
s_col = s_val.columns
s_val = s_val.values
fig.add_trace(

go.Heatmap(x=s_col,y=s_idx,z=s_val,name='pearson',showscale=False,xgap=1,ygap=1,
colorscale=colorscale),
    row=1, col=1
)


s_val =usampled_df.corr('spearman')
s_idx = s_val.index
s_col = s_val.columns
s_val = s_val.values
fig.add_trace(
    go.Heatmap(x=s_col,y=s_idx,z=s_val,xgap=1,ygap=1,colorscale=colorscale),
    row=2, col=1
)
fig.update_layout(
```

```
    hoverlabel=dict(
        bgcolor="white",
        font_size=16,
        font_family="Rockwell"
    )
)
fig.update_layout(height=1080, width=1920, title_text="Upsmapled Correlations")
fig.show()

# %%
N_COMPONENTS = 8

pca_model = PCA(n_components = N_COMPONENTS )

pc_matrix = pca_model.fit_transform(ohe_data)

evr = pca_model.explained_variance_ratio_
total_var = evr.sum() * 100
cumsum_evr = np.cumsum(evr)

trace1 = {
    "name": "individual explained variance",
    "type": "bar",
    'y':evr}
trace2 = {
    "name": "cumulative explained variance",
    "type": "scatter",
     'y':cumsum_evr}
data = [trace1, trace2]
layout = {
    "xaxis": {"title": "Principal components"},
    "yaxis": {"title": "Explained variance ratio"},
  }
fig = go.Figure(data=data, layout=layout)
fig.update_layout(                title='Explained      Variance      Using      {}
Dimensions'.format(N_COMPONENTS))
fig.show()
```

```
# %%
usampled_df_with_pcs = pd.concat([usampled_df,pd.DataFrame(pc_matrix,columns=['PC-{}'.format(i) for i in range(0,N_COMPONENTS)])],axis=1)
# usampled_df_with_pcs

# %%
fig = ex.scatter_matrix(
    usampled_df_with_pcs[['PC-{}'.format(i) for i in range(0,N_COMPONENTS)]].values,
    color=usampled_df_with_pcs['current_age'],
    dimensions=range(N_COMPONENTS),
    labels={str(i):'PC-{}'.format(i) for i in range(0,N_COMPONENTS)},
    title=f'Total Explained Variance: {total_var:.2f}%')

fig.update_traces(diagonal_visible=False)
fig.update_layout(
    coloraxis_colorbar=dict(
        title="current_age",
    ),
)
fig.show()

# %%
fig = make_subplots(rows=2, cols=1,shared_xaxes=True,subplot_titles=('Perason Correaltion', 'Spearman Correaltion'))


s_val =usampled_df_with_pcs.corr('pearson')
s_idx = s_val.index
s_col = s_val.columns
s_val = s_val.values
fig.add_trace(

go.Heatmap(x=s_col,y=s_idx,z=s_val,name='pearson',showscale=False,xgap=1,ygap=1, colorscale=colorscale),
    row=1, col=1
)
```

```
s_val =usampled_df_with_pcs.corr('spearman')
s_idx = s_val.index
s_col = s_val.columns
s_val = s_val.values
fig.add_trace(
    go.Heatmap(x=s_col,y=s_idx,z=s_val,xgap=1,ygap=1,colorscale=colorscale),
    row=2, col=1
)
fig.update_layout(
    hoverlabel=dict(
        bgcolor="white",
        font_size=16,
        font_family="Rockwell"
    )
)
fig.update_layout(height=1080, width=1920, title_text="Upsmapled Correlations With
PC\'s")
fig.show()

# %%
# usampled_df_with_pcs.info()

# %%
X_features = [
    'luckydrawy2024',
    'difference_lld',
    'bal_31dec24',
    'trxn_amt2024_c',
    'nb_months_fdd',

    'PC-0',
    'PC-1',
    'PC-2',
    'PC-3',
    'PC-4',
    'PC-5',
    'PC-6',
```

```
    'PC-7']

X = usampled_df_with_pcs[X_features]
y = usampled_df_with_pcs['Churn']

# %%
train_x,test_x,train_y,test_y = train_test_split(X,y,random_state=42)

# %%
x_small, y_small = train_x.sample(n=135675, random_state=42),
train_y.loc[train_x.sample(n=135675, random_state=42).index]

# %%
from sklearn.model_selection import RandomizedSearchCV

param_dist = {
    'n_estimators': [100, 200],
    'max_depth': [5, 10, None],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2],
    'class_weight': ['balanced', None]
}

rf = RandomForestClassifier(random_state=42, n_jobs=-1)
rf_search = RandomizedSearchCV(
    rf, param_distributions=param_dist, n_iter=10, cv=3, scoring='recall', n_jobs=-1,
random_state=42, verbose=0
)
rf_search.fit(x_small, y_small)

print("Best Params:", rf_search.best_params_)

# %% [markdown]
# Best Params: {'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 2,
'max_depth': 5, 'class_weight': 'balanced'}

# %%
param_dist = {
```

```
    'n_estimators': [50, 100, 150, 200],
    'learning_rate': [0.01, 0.1, 0.5, 1.0]
}

ada = AdaBoostClassifier(random_state=42)
ada_search = RandomizedSearchCV(ada, param_distributions=param_dist,
                                n_iter=10, cv=3, scoring='recall', n_jobs=-1,
random_state=42, verbose=0)

ada_search.fit(x_small, y_small)
print("Best Params:", ada_search.best_params_)

# %%
xgb = XGBClassifier(eval_metric='logloss')

xgb_param = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2],
    'subsample': [0.6, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0]
}

xgb_search = RandomizedSearchCV(xgb, param_distributions=xgb_param,
                                n_iter=10, cv=3, scoring='recall', n_jobs=-1,
random_state=42, verbose=0)

xgb_search.fit(x_small, y_small)
print("Best Params:", xgb_search.best_params_)

# %%
gbc = GradientBoostingClassifier(random_state=42)

gbc_param = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.6, 0.8, 1.0]
```

```
}

gbc_search = RandomizedSearchCV(gbc, param_distributions=gbc_param,
                                n_iter=10,        cv=3,        scoring='recall',        n_jobs=-1,
random_state=42, verbose=0)

gbc_search.fit(x_small, y_small)
print("Best Params:", gbc_search.best_params_)

# %%
cat = CatBoostClassifier(verbose=0)

cat_param = {
    'iterations': [100, 200, 300],
    'depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2],
    'l2_leaf_reg': [1, 3, 5]
}

cat_search = RandomizedSearchCV(cat, param_distributions=cat_param,
                                n_iter=10,        cv=3,        scoring='recall',        n_jobs=-1,
random_state=42, verbose=0)

cat_search.fit(x_small, y_small)
print("Best Params:", cat_search.best_params_)

# %%
lr = LogisticRegression(solver='saga', max_iter=1000)

lr_param = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100],
    'penalty': ['l1', 'l2'],
    'class_weight': ['balanced', None],
}

lr_search = RandomizedSearchCV(lr, param_distributions=lr_param,
                                n_iter=10,        cv=3,        scoring='recall',        n_jobs=-1,
random_state=42, verbose=0)
```

```
lr_search.fit(x_small, y_small)
print("Best Params:", lr_search.best_params_)
# %%
rf_pipe = Pipeline(steps =[ ('scale',StandardScaler()),
("RF",RandomForestClassifier(**rf_search.best_params_, random_state=4)) ])
ada_pipe = Pipeline(steps =[ ('scale',StandardScaler()),
("RF",AdaBoostClassifier(**ada_search.best_params_, random_state=42)) ])
xg_pipe = Pipeline(steps =[ ('scale',StandardScaler()),
("RF",XGBClassifier(**xgb_search.best_params_, random_state=42)) ])
gb_pipe = Pipeline(steps =[ ('scale',StandardScaler()),
("RF",GradientBoostingClassifier(**gbc_search.best_params_, random_state=42)) ])
cb_pipe = Pipeline(steps =[ ('scale',StandardScaler()),
("RF",CatBoostClassifier(**cat_search.best_params_, random_state=42)) ])
lr_pipe = Pipeline(steps =[ ('scale',StandardScaler()),
("RF",LogisticRegression(**lr_search.best_params_, random_state=42)) ])

# %%
rf_pipe.fit(train_x,train_y)
rf_prediction = rf_pipe.predict(test_x)

# %%
ada_pipe.fit(train_x,train_y)
ada_prediction = ada_pipe.predict(test_x)

# %%
xg_pipe.fit(train_x,train_y)
xg_prediction = xg_pipe.predict(test_x)

# %%
gb_pipe.fit(train_x,train_y)
gb_prediction = gb_pipe.predict(test_x)

# %%
cb_pipe.fit(train_x,train_y)
cb_prediction = cb_pipe.predict(test_x)

# %%
```

```
lr_pipe.fit(train_x,train_y)
lr_prediction = lr_pipe.predict(test_x)

# %%
fig = go.Figure(data=[go.Table(header=dict(values=[
    '<b>Model<b>',
    '<b>Accuracy On Test Data<b>',
    '<b>Precision On Test Data<b>',
    '<b>Recall On Test Data<b>',
    '<b>F1 Score On Test Data<b>'],
                                line_color='darkslategray',
    fill_color='whitesmoke',
    align=['center','center'],
    font=dict(color='black', size=18),
    height=40),

            cells=dict(values=[[
              '<b>Random Forest<b>',
              '<b>AdaBoost<b>',
              '<b>XG<b>',
              '<b>GradientBoosting<b>',
              '<b>CatBoost<b>',
              '<b>Logistic<b>'
              ], [
                np.round(accuracy(rf_prediction,test_y),2),
                np.round(accuracy(ada_prediction,test_y),2),
                np.round(accuracy(xg_prediction,test_y),2),
                np.round(accuracy(gb_prediction,test_y),2),
                np.round(accuracy(cb_prediction,test_y),2),
                np.round(accuracy(lr_prediction,test_y),2)
                ], [
                np.round(precision(rf_prediction,test_y),2),
                np.round(precision(ada_prediction,test_y),2),
                np.round(precision(xg_prediction,test_y),2),
                np.round(precision(gb_prediction,test_y),2),
                np.round(precision(cb_prediction,test_y),2),
                np.round(precision(lr_prediction,test_y),2)
                ], [
```

```
                    np.round(recall(rf_prediction,test_y),2),
                    np.round(recall(ada_prediction,test_y),2),
                    np.round(recall(xg_prediction,test_y),2),
                    np.round(recall(gb_prediction,test_y),2),
                    np.round(recall(cb_prediction,test_y),2),
                    np.round(recall(lr_prediction,test_y),2)
                    ], [
                    np.round(f1(rf_prediction,test_y),2),
                    np.round(f1(ada_prediction,test_y),2),
                    np.round(f1(xg_prediction,test_y),2),
                    np.round(f1(gb_prediction,test_y),2),
                    np.round(f1(cb_prediction,test_y),2),
                    np.round(f1(lr_prediction,test_y),2)
                    ]
                    ]))
                ])

fig.update_layout(title='Model Results On Test Data')
fig.show()

# %%
ohe_data =df_copy[df_copy.columns[16:]].copy()
pc_matrix = pca_model.fit_transform(ohe_data)
original_df_with_pcs = pd.concat([df_copy,pd.DataFrame(pc_matrix,columns=['PC-{}'.format(i) for i in range(0,N_COMPONENTS)])],axis=1)

# %%
z=confusion_matrix(rf_prediction,test_y)
fig = ff.create_annotated_heatmap(z, x=['Not Churn','Churn'], y=['Predicted Not Churn','Predicted Churn'], colorscale='Viridis',xgap=3,ygap=3)
fig['data'][0]['showscale'] = True
fig.update_layout(title='Prediction On Original Data With Random Forest Model Confusion Matrix')
fig.show()

# %%
z=confusion_matrix(ada_prediction,test_y)
```

```
fig = ff.create_annotated_heatmap(z, x=['Not Churn','Churn'], y=['Predicted Not Churn','Predicted Churn'], colorscale='Viridis',xgap=3,ygap=3)
fig['data'][0]['showscale'] = True
fig.update_layout(title='Prediction On Original Data With AdaBoost Model Confusion Matrix')
fig.show()

# %%
#
z=confusion_matrix(unsampled_data_prediction_XG,original_df_with_pcs['acct_status'])
z=confusion_matrix(xg_prediction,test_y)
fig = ff.create_annotated_heatmap(z, x=['Not Churn','Churn'], y=['Predicted Not Churn','Predicted Churn'], colorscale='Viridis',xgap=3,ygap=3)
fig['data'][0]['showscale'] = True
fig.update_layout(title='Prediction On Original Data With XgBoost Model Confusion Matrix')
fig.show()

# %%
#
z=confusion_matrix(unsampled_data_prediction_GB,original_df_with_pcs['acct_status'])
z=confusion_matrix(gb_prediction,test_y)
fig = ff.create_annotated_heatmap(z, x=['Not Churn','Churn'], y=['Predicted Not Churn','Predicted Churn'], colorscale='Viridis',xgap=3,ygap=3)
fig['data'][0]['showscale'] = True
fig.update_layout(title='Prediction On Original Data With GradienBoosting Model Confusion Matrix')
fig.show()

# %%
#
z=confusion_matrix(unsampled_data_prediction_CB,original_df_with_pcs['acct_status'])
z=confusion_matrix(cb_prediction,test_y)
fig = ff.create_annotated_heatmap(z, x=['Not Churn','Churn'], y=['Predicted Not Churn','Predicted Churn'], colorscale='Viridis',xgap=3,ygap=3)
fig['data'][0]['showscale'] = True
```

```
fig.update_layout(title='Prediction On Original Data With CatBoost Model Confusion Matrix')
fig.show()

# %%
#
z=confusion_matrix(unsampled_data_prediction_LR,original_df_with_pcs['acct_status'])
z=confusion_matrix(lr_prediction,test_y)
fig = ff.create_annotated_heatmap(z, x=['Not Churn','Churn'], y=['Predicted Not Churn','Predicted Churn'], colorscale='Viridis',xgap=3,ygap=3)
fig['data'][0]['showscale'] = True
fig.update_layout(title='Prediction On Original Data With LogisticRegression Model Confusion Matrix')
fig.show()

# %%
from sklearn.metrics import PrecisionRecallDisplay

# %%
unsampled_data_prediction_ADA = ada_pipe.predict_proba(test_x)
skplt.metrics.plot_precision_recall(test_y, unsampled_data_prediction_ADA)
plt.legend(prop={'size': 20})


# %%
unsampled_data_prediction_ADA = rf_pipe.predict_proba(test_x)
skplt.metrics.plot_precision_recall(test_y, unsampled_data_prediction_ADA)
plt.legend(prop={'size': 20})

# %%
unsampled_data_prediction_ADA = xg_pipe.predict_proba(test_x)
skplt.metrics.plot_precision_recall(test_y, unsampled_data_prediction_ADA)
plt.legend(prop={'size': 20})

# %%
unsampled_data_prediction_ADA = gb_pipe.predict_proba(test_x)
skplt.metrics.plot_precision_recall(test_y, unsampled_data_prediction_ADA)
plt.legend(prop={'size': 20})
```

```
# %%
unsampled_data_prediction_ADA = cb_pipe.predict_proba(test_x)
skplt.metrics.plot_precision_recall(test_y, unsampled_data_prediction_ADA)
plt.legend(prop={'size': 20})

# %%
unsampled_data_prediction_ADA = lr_pipe.predict_proba(test_x)
skplt.metrics.plot_precision_recall(test_y, unsampled_data_prediction_ADA)
plt.legend(prop={'size': 20})
```

ประวัติผู้เขียน

ชื่อ	นาย กฤตภาส โสมากุล
วัน เดือน ปีเกิด	20 กรกฎาคม 2537
ที่อยู่ปัจจุบัน	399/2 ซอย8 ตำบลในเมือง อำเภอเมือง จังหวัดนครราชสีมา 30000
ประวัติการศึกษา	2560 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมชีวการแพทย์ เกรดเฉลี่ย 2.94 มหาวิทยาลัยศรีนครินทรวิโรฒ