ระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ด้วยวิธีการเรียนรู้ของเครื่อง

นายจิรภัทร ศรีอภิรัฐ

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ด้วยวิธีการเรียนรู้ของเครื่อง

นายจิรภัทร ศรีอภิรัฐ

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองโครงร่างการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง ระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ด้วยวิธีการเรียนรู้ของเครื่อง

โดย นายจิรภัทร ศรีอภิรัฐ

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม

..... คณบดีบัณฑิตวิทยาลัย / หัวหน้าภาควิชา

(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)

คณะกรรมการสอบการค้นคว้าอิสระ

..... ประธานกรรมการ

(ดร.ชูชาติ หฤไชยะศักดิ์)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.มาลีรัตน์ มะลิแย้ม)

..... กรรมการ

(อาจารย์กาญจนา วิริยะพันธ์)

ชื่อ : นายจิรภัทร ศรีอภิรัฐ

ชื่อการค้นคว้าอิสระ : ระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ด้วยวิธีการเรียนรู้ของเครื่อง

สาขาวิชา : วิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : ผู้ช่วยศาสตราจารย์ ดร.มาลีรัตน์ มะลิแย้ม

ปีการศึกษา : 2567

บทคัดย่อ

ในปัจจุบันระบบอนุมัติบัตรเครดิตมีบทบาทสำคัญต่อธุรกิจธนาคารและสถาบันการเงิน เนื่อง จากใช้ในการพิจารณาคุณสมบัติของผู้สมัคร อย่างไรก็ตาม การอนุมัติบัตรเครดิตยังคงเป็นกระบวนการที่ซับซ้อน ต้องอาศัยการวิเคราะห์ข้อมูลหลายปัจจัย รวมถึงประวัติทางการเงิน รายได้ และพฤติกรรมการใช้จ่าย และปัจจัยอื่นๆ โดยใช้บุคคลากรในการตรวจสอบและอนุมัติให้บัตรเครดิตตามกฎเกณฑ์ที่กำหนด ส่งผลให้มีความล่าช้า งานวิจัยนี้นำเสนอระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิตโดย โดยใช้การเรียนรู้ของเครื่องแบบมีผู้สอนประเภทการจำแนก เพื่อเพิ่มประสิทธิภาพและลดข้อผิดพลาดในการพิจารณาอนุมัติ ผู้วิจัยได้สร้างแบบจำลองเพื่อแบ่งกลุ่มผู้สมัครบัตรเครดิตที่บัตรเครดิต เป็น 2 ประเภท ได้แก่ กลุ่มลูกค้าที่ผ่าเกณฑ์ และกลุ่มลูกค้าที่ผ่าเกณฑ์ โดยใช้อัลกอลิทึม 10 เทคนิค เรียนรู้และทดสอบกับข้อมูลรายการสมัครสินเชื่อบัตรเครดิตบนทดสอบของผู้ใช้งาน โดยมีข้อมูลรายการสมัครบัตรเครดิต จำนวน 1,568 รายการ เพื่อหาโมโดลที่มีประสิทธิภาพมากที่สุด สำหรับนำไปใช้พัฒนาระบบพิจารณาอนุมัติบัตรแบบอัตโนมัติต่อไป ผลการวิจัยพบว่าวิธีแบบเทคนิคXGBoost ให้ค่าความถูกต้อง 98.0 % เมื่อเปรียบเทียบค่าจากตารางเมทริกซ์ความสับสน

(มีจำนวนทั้งสิ้น 109 หน้า)

คำสำคัญ : การอนุมัติบัตรเครดิต, การเรียนรู้ของเครื่อง, ระบบสนับสนุนการตัดสินใจ

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Jirapat Sriapirat
Independent Study Title : Decision support system for credit card approval using
Major Field : Data Science for Innovation
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Assistant Professor Dr. Maleerat Maliyam
Academic Year : 2024

ABSTRACT

Currently, the credit card approval system plays a crucial role in banking and financial institutions as it is used to assess applicants' qualifications. However, credit card approval remains a complex process that requires analyzing multiple factors, including financial history, income, spending behavior, and other criteria. This process relies on personnel to review and approve credit cards based on predetermined regulations, which can lead to delays. This research presents a credit card approval decision support system using supervised machine learning (Supervised Learning) in classification to improve efficiency and reduce errors in the approval process. The researcher developed a model to categorize credit card applicants into two groups: those who meet the criteria and those who do not. The study applied ten machine learning techniques. The models. The research findings indicate that the XGBoost method achieved an accuracy of 98.0 %, based on comparisons from the Confusion Matrix table.

(Total 109 Pages)

Keywords: Credit Card Approval, Machine Learning, Decision Support System

_____ Advisor

กิตติกรรมประกาศ

การค้นคว้าอิสระฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี อันเนื่องจากผู้วิจัยได้รับความช่วยเหลือ ดูแลเอาใจใส่เป็นอย่างดีจากบุคคลต่างๆ ที่เกี่ยวข้อง ทั้งนี้ผู้จัดทำค้นคว้าอิสระต้องขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.มาลีรัตน์ มะลิแย้ม ซึ่งเป็นอาจารย์ที่ปรึกษาการค้นคว้าอิสระ ที่กรุณาให้คำแนะนำและให้คำปรึกษาตั้งแต่ริเริ่มในการนำเสนอหัวข้อการค้นคว้าอิสระ ทั้งให้ความช่วยเหลือ ให้ข้อคิด ให้ความรู้ ช่วยแก้ไขปัญหา ตรวจสอบความถูกต้อง ชี้แนะความคิดเห็นให้แนวทางในการจัดทำการค้นคว้าอิสระฉบับบนี้ได้อย่างดีเยี่ยม จนทำให้การค้นคว้าอิสระฉบับนี้สามารถสำเร็จลุล่วงไปได้ ขอขอบพระคุณกรรมการสอบค้นคว้าอิสระ อาจารย์ ดร.กาญจนา วิริยะพันธ์ ที่ให้คำปรึกษาที่ช่วยแนะในการปรับปรุงการค้นคว้าอิสระฉบับนี้ ให้มีความสมบูรณ์มากยิ่งขึ้น รวมทั้งขอขอบคุณอาจารย์และบุคลากร เจ้าหน้าในภาควิชาเทคโนโลยีสารสนเทศ ที่ช่วยประสาทวิชาความรู้ที่เกี่ยวข้อง คอยประสานงาน ทำให้ผู้จัดทำค้นคว้าอิสระสามารถนำความรู้มาประยุกต์ใช้ในการจัดทำการค้นคว้าอิสระฉบับนี้

สุดท้ายนี้ผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา หัวหน้าและเพื่อนร่วมงาน เพื่อนร่วมชั้นเรียนที่ได้ให้การสนับสนุนทางด้านกำลังใจ ให้คำแนะนำ และให้ความช่วยเหลือในการดำเนินงานจนการค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงไว้ ณ ที่นี้

จิรภัทร ศรีอภิรัฐ

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง (ถ้ามี)

สารบัญรูปภาพ (ถ้ามี)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

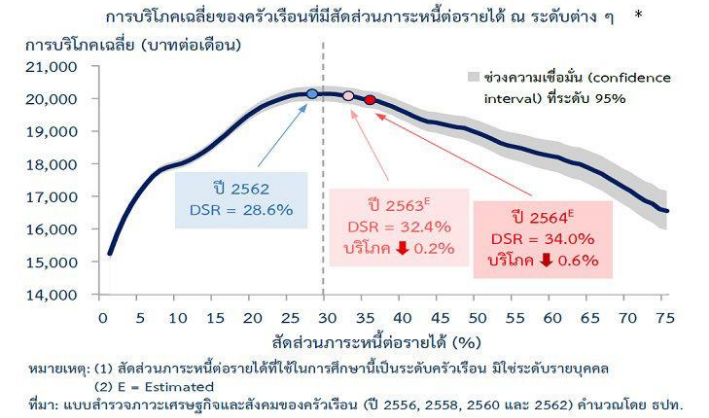
ภาพที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของงานวิจัย

ในปัจจุบันเทคโนโลยีดิจิทัลและปัญญาประดิษฐ์ (Artificial Intelligence: AI) ได้เข้ามามีบทบาทสำคัญในในทุกมิติของชีวิตมนุษย์ รวมถึงภาคธุรกิจและอุตสาหกรรมต่าง ๆ ที่นำเทคโนโลยีมาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพในการดำเนินงาน และหนึ่งในเทคโนโลยีที่หยิบยกขึ้นมาใช้นั้นคือการเรียนรู้ของเครื่อง (Machine learning) ที่เป็นสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence: AI) ที่สามารถเรียนรู้ด้วยต้นเองจากข้อมูล เป็นวิธีการวิเคราะห์ข้อมูลที่สร้างแบบจำลองเชิงวิเคราะห์โดยอัตโนมัติ ตัดสินใจเองและอาศัยการแทรกแซงของมนุษย์น้อยที่สุด การเรียนรู้ของเครื่องถูกนำมาใช้ในหลากหลายอุตสาหกรรม (วัชระ ทองสุข, 2566) เช่น ระบบแนะนำเนื้อหาของ Netflix รถยนต์ขับเคลื่อนอัตโนมัติ และระบบตรวจสอบสินเชื่อของสถาบันการเงิน เป็นต้น เทคโนโลยีเหล่านี้ช่วยเพิ่มความสะดวกสบาย ลดภาระงาน และสนับสนุนการตัดสินใจที่แม่นยำยิ่งขึ้นในองค์กรธุรกิจ ในบริบทของภาคการเงินและการธนาคาร ซึ่งมีการนำเทคโนโลยีขั้นสูงมาประยุกต์ใช้เพื่อเพิ่มประสิทธิ- ภาพในการดำเนินงาน ลดต้นทุน และเพิ่มความสะดวกสบายให้แก่ลูกค้า และหนึ่งในธุรกิจที่สำคัญของสถาบันการเงินและการธนาคารก็คือ ธุรกิจบัตรเครดิต ส่วนงานของบัตรเครดิตเป็นหนึ่งในธุรกิจด้านสินเชื่อของสถาบันการเงิน บริษัทบัตรเครดิต และผู้ให้บริการด้านการชำระเงิน โดยมุ่งเน้นให้ลูกค้าสามารถใช้จ่ายเงินล่วงหน้าได้ผ่านบัตรเครดิตภายในวงเงินที่ได้รับอนุมัติ และทำการชำระคืนตามรอบบิลที่กำหนด

ประเทศไทยได้รับผลกระทบประเทศไทยได้รับผลกระทบจากวิกฤตเศรษฐกิจหลายระลอกตั้งแต่ก่อนวิกฤตโคจากการวิเคราะห์ข้อมูลแบบสำรวจภาวะเศรษฐกิจและสังคมของครัวเรือนสำนักงานสถิติแห่งชาติ พบว่า สัดส่วนภาระหนี้ต่อรายได้ (Debt Service Ratio: DSR) โดยเฉลี่ยของครัวเรือนไทยอยู่ในจุดใกล้เส้นยาแดงที่ระดับ 30% มาตั้งแต่ก่อนเกิดโควิด (ปี 2562) ถ้าสัดส่วนนี้เพิ่มขึ้นอีกนิดเดียวจนเกิน 30% ซึ่งเป็นจุดวกกลับ (turning point) การก่อหนี้ก็จะเปลี่ยนบทบาทจากการกระตุ้นเป็นการฉุดรั้งการบริโภคของครัวเรือน



ภาพที่ 1-1 การบริโภคเฉลี่ยของครัวเรือนที่มีสัดส่วนภาระหนี้ต่อรายได้ ณ ระดับต่างๆ

สถานการณ์การแพร่ระบาดของโรคโควิด–19 วิดช่วงปี 2563 – 2564 ไปจนถึงความขัดแย้งระหว่างรัสเซีย-ยูเครน ซึ่งเป็นการเผชิญวิกฤติซ้อนวิกฤติ ทำให้ทรัพยากรที่เคยสมดุลต้องขาดแคลน มีผลให้ต้นทุนสินค้า ราคาพลังงาน และค่าครองชีพปรับตัวสูงขึ้น เศรษฐกิจไทยถูกกระทบรุนแรง รายได้ครัวเรือนหายไปมาก ครัวเรือนจำเป็นต้องกู้ยืมเพื่อพยุงกำลังซื้อและรักษาระดับการบริโภค สัดส่วนภาระหนี้ต่อรายได้โดยเฉลี่ยจึงปรับตัวเพิ่มขึ้นเร็วเกินระดับ 30% (ธนาคารแห่งประเทศไทย, 2565) จนทำให้ภาระหนี้กลายเป็นปัจจัยฉุดรั้งการบริโภคและการฟื้นตัวของเศรษฐกิจ และต่อมาในไตรมาส 3 ปี 2564 ปัญหาหนี้ครัวเรือนที่สูงถึงร้อยละ 89.3 ต่อ GDP โดยอ้างอิงจากข้อมูลสถาบันดิจิตอลและเศรษฐกิจ กล่าวว่าเมื่อเดือนมีนาคม 2565 พบว่าคนไทยมีหนี้ในปริมาณที่สูง 57% ของคนไทยที่มีหนี้มีหนี้เกิน 100,000 บาท และมีหนี้พุงสูงเกิน 90% ของ GDP มาอย่างต่อเนื่องจนติดอันดับ 7 ของโลกที่หนี้สูงมากที่สุด และเป็นหนี้หลายบัญชีโดยเฉลี่ย 3 บัญชีต่อคนและ 32% มีหนี้ 4 บัญชีขึ้นไป (ธนาคารแห่งประเทศไทย, 2566) และสัดส่วนกความสามารถในการชำระหนี้ก็ลดลงจากข้อมูลเครดิตบูโรเปิดเผยตัวเลขหนี้เสีย (NPL) สิ้น ที่ค้างชำระเกิน 90 วันขึ้นไป ไตรมาศที่ 3 ของปี 2567 พุ่งแตะถึง 1.2 ล้านล้านบาท หรือเพิ่มขึ้น 12.2 % เมื่อเทียบกับปีก่อนหน้าสูงที่สุดที่เคยมีมา สัดส่วนของหนี้บัตรเครดิคเพิ่มขึ้นมาที่ 21.5 % ภาพรวมของหนี้เสียยังคงปรับตัวสูงขึ้นอย่างต่อเนื่องเป็นผลจากที่การปรับโครงสร้างหนี้ทำได้ยาก คนยังมีรายได้ไม่เพียงพอ ไม่มีความสามรถในการชำระหนี้ ลูกค้าหนี้ยังมีรายได้ไม่มากพอและบางส่วนรายได้ยังไม่กลับ มาระดับก่อนโควิดปี 2562 (นายสุรพล โอภาสเสถียร, 2567) งานวิจัย SCB Economic Intelligence Center (EIC) แสดงให้เห็นว่ากลุ่มคนฐานล่าง 50% จนถึงปี 2565 รายได้ยังไม่ฟื้นคืนกลับมาช่วงก่อนโควิต ตรงข้ามกับกลุ่มบน top 10% ที่รายได้กลับมาอย่างรวดเร็วและยังโตเร็วกว่าเดิม และงานวิจัยของธนาคารแห่งประเทศไทยทำให้เห็นถึง

คนทำงาน 40 ล้านคน มีคนของภาคส่วนไหนบ้างรายได้ยังไม่กลับมาเท่าเดิม คนที่ทำงานของภาคเกษตร 12 ล้านคน รายได้กลับมาแล้ว 119% เมื่อเทียวกับปี 2562 แต่คนที่ทำงานนอกของภาคเกษตร 40 ล้านคน แบ่งเป็นลูกจ้าง 18 ล้านคน ทำหน้าที่อิสระอีก 10% คนทำกลุ่มนี้ที่อยู่ในภาคการท่องเที่ยวและอุตสาหกรรมรายได้ยังไม่กลับมาถึง 100% ในขณะที่บริษัทสินเชื่อ Nano finance สำหรับที่เป็นเงินกู้สำหับผู้ที่ทำอาชีพอิสระ มีรายได้ไม่แน่นอนอย่างพ่อค้าแม่ค้าแต่ดอกเบี้ยสูงกว่า 33% ต่อปี มีการปล่อยสินเชื่อกลุ่มนี้เพิ่มขึ้น 26.8% คนไทยตั้งแต่ระดับกลางลงจนถึงระดับล่าง อยู่ในสถานะการณ์ที่รายได้ไม่พอกับค่าคลองชีพที่ดีดตัวขึ้นมาจนทำให้ขาดความสามารถในการจ่ายหนี้ (THE STANDARD, 2567) กลุ่มที่ได้รับผลกระทบจากการระบาด เงินออมที่น้อยลง ทำให้ครัวเรือนเหล่านี้ขาดภูมิคุ้มกันในการรองรับเหตุการณ์เลวร้ายที่อาจเกิดขึ้นในอนาตค เช่น หากถูกเลิกจ้าง หรือค่าจ้างถูกปรับลดลงมาก นอกจากครัวเรือนจะลดการบริโภคแล้ว ยังอาจส่งผลให้ครัวเรือนผิดนัดชำระหนี้ สร้างความเสี่ยงให้กับระบบสถาบันการเงินหรือผู้ให้กู้ยืม และในกรณีเลวร้ายหากเกิดการผิดนัดชำระหนี้ในวงกว้าง ระบบการเงินได้รับความเสียหายจนไม่สามารถดำเนินการได้ตามปกติ ก็จะกระทบกิจกรรมทางเศรษฐกิจรุนแรง และอาจนำไปสู่วิกฤตเศรษฐกิจได้ในท้ายที่สุด ดังนั้นในการอนุมัติการให้สินเชื่อบัตรเครดิตแก่บุคคลใดจะต้องเป็นไปอย่างเหมาะสมเพื่อป้องกันการเกิดหนี้เสียในการไม่มีความสามารถในการชำระหนี้ตามมา จากข้อมูลธนาคารแห่งประเทศไทย ได้มีการกำหนดแนวทางการพิจารณานาหลักเกณฑ์การบริหารจัดการด้านการให้บริการแก่ ลูกค้าอย่างเป็นธรรม ในการที่ให้สถาบันการเงินกำหนดเกณฑ์ในการให้สินเชื่อบัตรเครดิตได้อย่างเหมาะสม (ธนาคารแห่งประเทศไทย, 2566) เพื่อให้สอดคล้องกับสำนักงานนายยกรัฐมนตรี กำหนดให้ปี 2565 เป็นปีแห่งการแก้ไขหนี้ครัวเรือน หนี้อื่นๆรวมถึงลูกหนี้บัตรเครดิต และในช่วงวิกฤตโควิด–19 ปัญหาหนี้ครัวเรือน ปัจจัยเสี่ยงสำคัญต่อเศรษฐกิจไทยในระยะข้างหน้า หากไม่ได้รับการแก้ไขก็จะเป็นปัจจัยฉุดรั้งการฟื้นตัวของเศรษฐกิจ และเป็นระเบิดเวลาที่อาจปะทุขึ้นจนกระทบต่อเสถียรภาพระบบการเงิน นับเป็นจุดเริ่มต้นที่ดีในการจัดการกับปัญหาและป้องกันการเกิดหนี้เสียจากการที่ออกบัตรเครดิตให้กับบุคคลที่เหมาะสม และอย่างไรก็ตามรัฐบาลก็อยากผลักดันให้เกิดเศรษฐกิจดิจิทัลเร็วขึ้นอย่างก้าวกระโดด ทั้งการทำธุรกรรมและการให้บริการรูปแบบต่างๆ ทางอิเล็กทรอนิกส์ ตลอดจน การเติบโตของเศรษฐกิจแพลตฟอร์ม (Platform Economy) การค้า e-Commerce และการใช้ Digital Banking มีการขยายตัวอย่างรวดเร็ว จากการที่รัฐบาลได้มีการวางรากฐาน “เศรษฐกิจดิจิทัล” (Digital Economy) ตามที่ได้กล่าวมาในข้างต้น และจากแนวคิด National e-Payment ส่งเสริมให้เกิด “สังคมไร้เงินสด” (Cashless Society) รัฐบาลได้สนับสนุนเพื่อพัฒนาแพลตฟอร์มเพื่อเป็นเครื่องมือในการขับเคลื่อนนโยบายและมาตรการต่างๆ จนประสบผลสำเร็จในการขับเคลื่อนเศรษฐกิจดิจิทัล สำนักงานนายกรัฐมนตรี. (2565) ธุรกิจบัตรเครดิตจึงมีความสำคัญในการคลับเคลื่อนนโยบายของรัฐบาลให้เกิดขึ้นจริง

ดอกเบี้ยบัตรเครดิตแพงเป็นอันดับต้นๆ เนื่องจากหนี้ที่เกิดจาการใช้จ่ายบัตรเครดิตเป็นสินเชื่อแบบไม่มีหลักประกัน และไม่มีข้อตกลงเลยว่าจะใช้กรอบเวลาในการจ่ายคืนเท่าไร ต่างจากหนี้บ้านหรือหนี้รถ ที่ธนาคารและผู้กู้ทำข้อตกลงร่วมกันชัดเจนว่าจะจ่ายเงินคืนให้ได้ภายในกี่ปี ธนาคารจึงบริหารจัดการได้ยากกว่า เพราะไม่สามารถรู้เลยว่าจะได้เงินกลับมาได้เมื่อไร หรือกลายเป็นหนี้เสียหรือไม่ ดอกเบี้ยบัตรเครดิตจึงสูงกว่าสินเชื่อที่มีหลักทรัพย์ค้ำประกัน เป็นเหตุผลที่แบงค์ชาติกำหนดว่า คุณต้องจ่ายเงินขั้นต่ำอย่างน้อยกี่เปอร์เซ็นต์ของยอดหนี้ เดิมทีการผู้ออกบัตร จะพิจารณาและอนุมัติในการออกบัตรเครดิตนั้นข้องข้างเสี่ยงเกิดหนี้เสียสูง ดังนั้นจึงต้องข้อมูลประกอบในการพิจารณาได้แก่ข้อมูล Credit Scoring หรือคะแนนเครดิต สำหรับประเมินความสามารถในการำระหนี้จากเจ้าของผู้ขอยื่นอนุมัติบัตรเครดิตและวงเงิน มีรายได้เพียงพอไหม พฤติกรรมจ่ายหนี้ในอดีตเป็นอย่างไร โดยในปัจจุบันของโดยทั่วไปประเทศไทยนำข้อมูล Credit Scoring มาจากศูนย์เครดิตบูโร ควบคู่กับข้อมูลคุณสมบัติอื่นๆ เช่น อัตราส่วนภาระหนี้ต่อรายได้รวม (DTI) หรือ ภาระหนี้สิน (DSR) เป็นต้น ล้วนถูกกำหนดตามแต่ละผู้ออกบัตร ได้แก่ สถาบันทางการเงิน องค์กร หน่วยงาน หรือบริษัทธุรกิจเป็นผู้กำหนดต่างกันไป ดังนั้นหนึ่งในกระบวนการสำคัญในการสำหรับธุรกิจบัตรเครดิตก็คือระบบอนุมัติบัตรเครดิต (Credit Card approval Application) ซึ่งมีบทบาทสำคัญต่อธุรกิจธนาคารและสถาบันการเงินเป็นเครื่องมือที่ใช้พิจารณาว่าผู้สมัครบัตรเครดิตมีคุณสมบัติเพียงพอและมีความสามารถในการชำระหนี้ไม่ก่อหนี้เสีย (Non-Performing Loan: NPL) ค้างในระบบหรือไม่ รวมถึงต้องพิจารณาได้อย่างรวดเร็วและแม่นยำสูง อย่างไรก็ตาม ความท้าทายหลักของธนาคารและสถาบันการเงินคือการคัดกรองลูกค้าที่เหมาะสมการอนุมัติบัตรเครดิตยังคงเป็นกระบวนการที่ซับซ้อน ซึ่งต้องอาศัยการวิเคราะห์ข้อมูลหลายปัจจัย จากระบบงานภายในและระบบงานภายนอกสถาบันการเงิน ข้อมูลเครดิตบูโร รวมถึงประวัติทางการเงิน รายได้ และพฤติกรรมการใช้จ่ายเพื่อใช้ประมวลผลข้อมูลและคำนวณค่าต่างๆ ตามเงื่อนไขเกณฑ์พิจารณาที่สถาบันการเงินถูกกำหนดไว้ในปัจจุบันการพิจารณาอนุมัติบัตรเครดิตและสินเชื่อส่วนบุคคล ซึ่งยังคงต้องอาศัยบุคลากรในการตรวจสอบข้อมูลและอนุมัติ โดยอิงตามโปรแกรมที่แสดงผลลัพธ์ตามเงื่อนไขตามที่สถาบันการเงินถูกกำหนด กระบวณการในการอนุมิติจะดำเนินการผ่านหลายส่วนงาน ใช้พนักงานที่ตรวจสอบและอนุมัติหลายขั้นตอน จึงใช้เวลาในการดำเนินการเป็นเวลานาน ส่งผลให้เกิดความล่าช้ากว่าจะรองรับผลการอนุมัติและแจ้งผลอนุมัติ โดยบัตรเครดิตและสินเชื่อส่วนบุคคลของสถาบันทางการเงิน จะแจ้งผลอนุมัติไม่น้อยกว่า 1 สัปดาห์ หลังจากผู้ขอยื่นอนุมัติยื่นขอ

จากปัญหาและเทคโนโลยีข้างต้นจึงเห็นว่าควรนำเทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning Model) เข้ามาประยุกต์ใช้ในกระบวนการอนุมัติบัตรเครดิตสามารถช่วยลดระยะเวลาการประเมินลูกค้า และเพิ่มความแม่นยำในการพิจารณาคุณสมบัติของผู้ขอสินเชื่อ โดยระบบสามารถวิเคราะห์ข้อมูลคุณสมบัติของผู้สมัครบัตรเครดิที่เกี่ยวข้อง เช่น รายได้ ประวัติการชำระหนี้ และ

พฤติกรรมทางการเงิน เพื่อจำแนกกลุ่มลูกค้าที่มีโอกาสเกิดหนี้ศูนย์และมีความสามารถในการชำระหนี้ได้อย่างถูกต้อง ในการอนุมัติบัตรเครดิตและวงเงินให้แก่บุคคลที่ไม่เหมาะสม ช่วยสนับสนุนการตัดสินใจของพนักงานในการพิจารณาอนุมัติบัตรเครดิตได้สะดวกรวดเร็วและดียิ่งขึ้น และทางผู้วิจัยจึงใช้การเรียนรู้ของเครื่องแบบมีผู้สอน (Supervised Learning) นำไปสร้างแต่ละโมโดลในการจำแนกผลการพิจารณาอนุมัติบัตรเครดิต โดยใช้เทคนิค 10 เทคนิค ได้แก่ Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM) , K-nearest Neighbors (KNN), Ensemble, XG Boost, Gradient Boosting Neuron Network และ, Naive Bayes ซึ่งโมเดลดังกล่าวที่จะได้มีการเรียนรู้และทดสอบกับชุดข้อมูล (Data Set) ที่อยู่ในชั้นกระบวนการทดสอบกระบวนการทดสอบซอฟต์แวร์โดยผู้ใช้งานจริง (User Acceptance Test) ข้อมูลที่ใช้เป็นข้อมูลประเภทของผลิตภัณฑ์ บัตรเครดิตวีซ่า (VISA) และเกณฑ์พิจารณาจากรายได้เป็นหลักและประเมินผลความถูกต้องแม่นยำของโมเดล โดยใช้เทคนิคการวัดผลที่เป็นมาตรฐาน Confusion matrix เพื่อหาโมโดลที่ดีที่สุด สำหรับนำไปใช้พัฒนาระบบระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิตต่อไป

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองและหาแบบจำลองที่ดีที่สุดในการจำแนกผลการพิจารณาอนุมัติบัตรเครดิตบัตรเครดิต มาใช้ในการพัฒนาระบบระบบสนับสนุนการตัดสินใจอนุมัติบัตร ชเครดิต (Credit Card approval Application)

1.3 ขอบเขตของการวิจัย

งานวิจัยนี้ศึกษาและสร้างแบบจำลองในการจำแนกที่มีความสามารถในการตัดสินผลการพิจารณาอนุมัติบัตรเครดิต เป็น 2 ประเภท ได้แก่อนุมัติให้ผ่าน (Approve) และไม่อนุมัติไม่ให้ผ่าน (Reject) โดยใช้ชุดข้อมูล (Dataset) จากข้อมูลจากการยื่นสมัครบัตรเครดิตและผ่านการพิจารณาแล้ว ได้แก่ข้อมูลส่วนตัว ข้อมูลการทำงาน รายได้ เครดิต หนี้ และความสามารถในการชำระหนี้เป็นต้น จากหน่วยงานแห่งหนึ่งในชั้นกระบวนการทดสอบสอบซอฟต์แวร์โดยผู้ใช้งานจริง (User Acceptance Test) ของระบบพิจารณาอนุมัติบัตรเครดิต (Credit Card Approval Application) ย้อนหลังตั้งแต่ มกราคม 2564 - ตุลาคม 2566 จำนวน 1567 รายการ โดยการใช้เทคนิค 10 เทคนิคที่เลือกมาจากงานวิจัยที่เกี่ยวข้อง ได้แก่ Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM) , K-nearest Neighbors (KNN), Ensemble, XG Boost, Gradient Boosting Neuron Network และ, Naive Bayes แก้ปัญหาการจำแนกข้อมูลไม่สมดุลโดยเทคนิคเป็นเทคนิคการสุ่มใช้ SMOTE และหาแบบจำลองที่มีประสิทธิภาพมากที่สุดในการอนุมัติบัตรเครดิต

โดยใช้ ใช้ตาราง Confusion Matrix ในการประเมินผลของแบบจำลอง เพื่อนำไปใช้พัฒนาระบบอนุมัติบัตรเครดิต

1.4 นิยามศัพท์เฉพาะ

1.4.1 เครดิตบูลโร (Credit Bureau) หมายถึง องค์กรที่รวบรวมและจัดเก็บข้อมูลเครดิตของบุคคลหรือธุรกิจจากสถาบันการเงิน เพื่อนำไปใช้ประเมินความเสี่ยงทางการเงิน (ธนาคารแห่งประเทศไทย, 2566)

1.4.2 ชั้นกระบวนการทดสอบกระบวนการทดสอบซอฟต์แวร์โดยผู้ใช้งานจริง (User Acceptance Test) หมายถึง กระบวนการที่ผู้ใช้ปลายทางทดสอบซอฟต์แวร์ก่อนนำไปใช้งานจริง เพื่อตรวจสอบว่าตรงตามความต้องการหรือไม่ (swiftlet, 2565)

1.4.3 อัตราส่วนภาระหนี้ต่อรายได้รวม (DTI) หมายถึง ตัวชี้วัดทางการเงินที่คำนวณจากสัดส่วนของภาระหนี้สินต่อรายได้รวมของบุคคล เพื่อใช้ในการพิจารณาสินเชื่อ (ananda, 2563)

1.4.4 ภาระหนี้สิน (DSR) หมายถึง อัตราส่วนที่ใช้วัดความสามารถของบุคคลหรือองค์กรในการชำระหนี้ เทียบกับรายได้ที่มี (ธนาคารกรุงไทย, 2566).

1.4.5 สินเชื่อส่วนบุคคล (Personal Loan) หมายถึง เงินกู้ที่บุคคลสามารถขอจากธนาคารหรือสถาบันการเงินเพื่อนำไปใช้ตามวัตถุประสงค์ส่วนตัว (refinn writer, 2563)

1.4.6 คะแนนเครดิต (Credit Scoring) หมายถึง หรือคะแนนเครดิต คะแนนที่แสดงถึงความน่าเชื่อถือทางการเงินของบุคคล โดยคำนวณจากประวัติการชำระหนี้และพฤติกรรมทางการเงิน (บริษัท ข้อมูลเครดิตแห่งชาติ จำกัด, 2567)

1.4.7 ผลิตภัณฑ์มวลรวมของประเทศ (GDP) หมายถึง มูลค่ารวมของสินค้าและบริการที่ผลิตในประเทศภายในระยะเวลาหนึ่ง (SET, (2564)

1.4.8 วีซ่า (VISA) หมายถึง บริษัทผู้ให้บริการเครือข่ายการชำระเงินระดับโลกที่ออกบัตรเครดิตและบัตรเดบิตร่วมกับธนาคาร (Paweennuch W, 2565)

1.4.9 แลกสะสมคะแนน (redeem point) หมายถึง กระบวนการใช้คะแนนสะสมจากบัตรเครดิตหรือโปรแกรมสะสมแต้มเพื่อแลกรับของรางวัลหรือสิทธิประโยชน์ต่าง ๆ (บริษัท เท็น เอ็กซ์ เอ็นเตอร์ไพรซ์ จำกัด, 2567)

1.4.10 เศรษฐกิจดิจิทัล (Digital Economy) หมายถึง ระบบเศรษฐกิจที่ขับเคลื่อนด้วยเทคโนโลยีดิจิทัล รวมถึงธุรกรรมออนไลน์และการใช้ข้อมูลขนาดใหญ่ (Big Data) (Indusadmin, 2560)

1.4.11 เศรษฐกิจแพลตฟอร์ม (Platform Economy) หมายถึง รูปแบบเศรษฐกิจที่ใช้แพลตฟอร์มดิจิทัลเป็นตัวกลางในการเชื่อมต่อผู้ให้บริการและผู้บริโภค เช่น Grab, Shopee และ Airbnb (สมประวิณ มันประเสริฐ, 2564)

1.4.12 การค้า e-Commerce หมายถึง การซื้อขายสินค้าและบริการผ่านระบบออนไลน์ โดยใช้แพลตฟอร์มดิจิทัลเป็นช่องทางหลัก (Mo Nuttamon, 2567)

1.4.13 National e-Payment หมายถึง โครงการพัฒนาโครงสร้างพื้นฐานด้านการชำระเงินของประเทศให้เป็นระบบอิเล็กทรอนิกส์ เช่น การโอนเงินผ่านพร้อมเพย์ (National e-Payment, 2558)

1.4.14 สังคมไร้เงินสด (Cashless Society) หมายถึง สังคมที่มีการใช้เงินสดลดลง และใช้การชำระเงินผ่านบัตรหรือระบบอิเล็กทรอนิกส์แทน (Merkle Capital, 2567)

1.4.15 ปัญญาประดิษฐ์ (AI) หมายถึง เทคโนโลยีที่พัฒนาให้คอมพิวเตอร์สามารถคิด วิเคราะห์และตัดสินใจได้คล้ายกับมนุษย์ (Dia, 2567)

1.4.16 การเรียนรู้ของเครื่อง (Machine learning) หมายถึง สาขาหนึ่งของ AI ที่ทำให้ระบบสามารถเรียนรู้จากข้อมูลและพัฒนาความสามารถโดยอัตโนมัติ (Sas, 2567)

1.4.17 จุดวกกลับ (turning point) หมายถึง ความสัมพันธ์ระหว่างภาระหนี้และการบริโภคเปลี่ยนทิศทาง เป็นการหาความสัมพันธ์ระหว่างภาระหนี้สินและการบริโภคครัวเรือน ใช้วิธีการทางเศรษฐมิติที่เรียกว่า Local linear regression ซึ่งสามารถหาความสัมพันธ์ที่ไม่เป็นเชิงเส้น (non-linear) ระหว่างตัวแปรต่างๆ จึงเหมาะกับการหาจุดวกกลับที่ความสัมพันธ์ระหว่างภาระหนี้และการบริโภคเปลี่ยนทิศทาง (ธนาคารแห่งประเทศไทย, 2565)

1.5 ประโยชน์ที่ได้รับ

ได้โมเดลที่ดีที่สุดไปใช้พัฒนาระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต เพื่อให้ธนาคารหรือหน่วยงานผู้พิจารณาบัตรเครดิต สามารถใช้ผลพิจารณาที่ได้ประกอบเพื่อช่วยในการตัดใจสินใจอนุมัติบัตรเครดิต อย่างมีประสิทธิภาพมากยิ่งขึ้น ตัดสินใจในการอนุมัติบัตรได้รวดเร็ว ลดความเสี่ยงที่อาจการเกิดหนี้ศูนย์ ลดการอนุมัติบัตรเครดิตแต่ประเภทวงเงินให้แก่บุคคลที่ไม่เหมาะสม ลดต้นทุน และรองรับการเติบโตของเศรษฐกิจดิจิทัลในอนาคต

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้องกับบัตรเครดิต

2.1.1 ความหมายและประเภทของบัตรเครดิต

บัตรเครดิตคือ หมายถึงบัตรหรือการ์ดที่สามารถนำมาใช้ซื้อ สินค้าและบริการได้ตามวงเงินที่ผู้ออกบัตร เป็นบัตรอิเล็กทรอนิกส์นี้ใช้แทนเงินสดโดยธนาคารหรือสถาบันบันธุรกิจทางด้านบัตรเครดิตที่ไม่ใช้สถาบันการเงินเป็นผู้ออกบัตร (Issuer) ให้แก่ลูกค้าผู้ถือบัตร (Card Holder) เป็นการให้สินเชื่ออุปโภคบริโภค สามารถนำบัตรมาใช้ในการใช้ชำระค่าสินค้าหรือบริการตามร้านค้าที่ตกลงกัน รวมถึงสามารถใช้บัตรถอนเงินสดผ่านเครื่อง ATM โดยหลังใช้จ่ายจะตัดยอดทั้งหมดตามรอบบิลของบัตรเครดิตนั้น มีกำหนดชำระเงินเป็นรายเดือน ตามยอดใช้จ่ายโดยสามารถชำระแบบเต็มจำนวน หรือขั้นต่ำ ตามข้อตกลงภายใต้วงเงินที่ได้อนุมัติ ทางร้านค้าผู้ให้บริการจะต้องเช็คยอดที่จ่ายกับทางธนาคารก่อนและจะได้รับรหัสอนุมัติจากธนาคาร ในสมัยก่อนจะเป็นเครื่องรูดบัตร ร้านค้าต้องโทรศัพท์ไปที่ธนาคาร แต่ปัจจุบันนี้มีเครื่องรูดบัตรที่จะ ออนไลน์กับธนาคารเพื่อให้ได้รหัสอนุมัติได้ในทันที จากผู้ขายหรือผู้ให้บริการก็จะนำสลิปไปให้ เจ้าของบัตรเซ็นชื่อ เพื่อตรวจสอบว่าเป็นเจ้าของบัตรจริงหรือไม่ โดยเทียบกับลายเซ็นต์ที่เซ็นต์ไว้ ด้านหลังของบัตรเครดิต และเก็บสำเนาไว้เพื่อส่งให้ธนาคารตรวจสอบได้ในภายหลัง บัตรเครดิคมีเครือข่ายผู้ให้บริการเช่น VISA, Master Card, American Express, China Union Pay (CUP) และ Japan Credit Bureau (JCB) ซึ่งบัตรเครดิตนั้นมีข้อดี และประโยชน์อย่างมากถ้าผู้ใช้ที่ใช้เป็น มีความสะดวกสะบายสามารถใช้จ่ายโดยไม่ต้องพกเงินสด ไม่ต้องกังวลเรื่องเงินทอน แถมยังทั้งโปรโมชั่นที่เป็นส่วนลดร้านค้า การแลกและสะสมคะแนน (redeem point) และยังมีสุทธิพิเศษอื่นๆ ที่สงเสริมการขาย เช่น ส่วนลดจากร้านค้า การโปรผ่อนชำระสินค้าดอกเบี้ย 0 % บริการที่จอดรถ ห้องรับรองสถาที่ต่าง ๆ เงินคืนจากการใช้จ่าย (cash back) เป็นต้น สิทธิที่ได้นั้นประโยชน์ขึ้นอยู่กับผู้ออกบัตรกำหนด

การคิดดอกเบี้ยบัตรเครดิต ผู้ออกบัตรจะเรียกเก็บดอกเบี้ย ค่าปรับ ค่าบริการ และค่าธรรมเนียมใด ๆ (ในที่นี้ขอเรียกโดยย่อว่าดอกเบี้ยและค่าธรรมเนียม) รวมกันได้ไม่เกิน 16% ต่อปี โดยมีวิธีคำนวณ 2 แบบตามการใช้บัตร (ธนาคารแห่งประเทศไทย, ม.ป.ป.) ดังนี

2.1.1.1 การชำระค่าสินค้าและบริการไม่เต็มจำนวนภายในวันที่กำหนดหรือชำระล่าช้า ผู้ออกบัตรสามารถคิดดอกเบี้ยตามการใช้จ่ายตั้งแต่วันที่ผู้ออกบัตรได้สำรองจ่ายให้ร้านค้า หรือตั้งแต่วันที่สรุปยอดรายการใช้จ่าย หรือตั้งแต่วันที่ครบกำหนดชำระก็ได้แต่โดยทั่วไปจะเริ่มคิดตั้งแต่วันที่ผู้ออกบัตรสำรองจ่ายเงินให้แก่ร้านค้า ซึ่งจะแบ่งเป็น 2 ส่วน คือ คิดเต็มจำนวนจนกว่าจะวันก่อนครบ

กำหนดชำระเงิน และคิดตามยอดคงค้าง (หักส่วนที่ชำระแล้วออก) นับจากวันที่ชำระจนถึงวันสรุปยอดถัดไป

2.1.1.2 การเบิกถอนเงินสด จะเริ่มคำนวณตั้งแต่วันที่เบิกถอนเงินสดออกมา ดูรายละเอียดเพิ่มเติมได้ที่หัวข้อดอกเบี้ยและค่าธรรมเนียมบัตรเครดิตนอกจากนี้ อาจมีค่าธรรมเนียมและค่าใช้จ่ายอื่น ๆ ซึ่งจะแตกต่างกันไปตามข้อกำหนดของผู้ออกบัตรแต่ละแห่ง ได้แก่

2.1.1.2.1 ค่าธรรมเนียมแรกเข้าและค่าธรรมเนียมรายปี ซึ่งเราสามารถต่อรองกับผู้ออกบัตร เพื่อขอยกเว้นการเรียกเก็บได้

2.1.1.2.2 ค่าธรรมเนียมในการชำระเงินผ่านช่องทางต่าง ๆ

2.1.1.2.3 ค่าธรรมเนียมการเบิกถอนเงินสดผ่านบัตรเครดิต โดยผู้ออกบัตรเรียกเก็บค่าธรรมเนียมประเภทนี้ได้ไม่เกิน 3 % ของจำนวนเงินสดที่เบิกถอน และต้องเสียภาษีมูลค่าเพิ่ม ซึ่งคิดจากยอดค่าธรรมเนียมการเบิกถอนด้วย

2.1.1.2.4 ค่าความเสี่ยงจากการแปลงสกุลเงินในการใช้บัตรเครดิตในต่างประเทศ 2 - 2.5 % ของอัตราแลกเปลี่ยนอ้างอิง

2.1.1.2.5 ค่าใช้จ่ายต่าง ๆ ตามที่ได้จ่ายไปจริงและพอสมควรแก่เหตุ เช่น ค่าใช้จ่ายในการติดตามทวงถามหนี้ (ซึ่งจะต้องมีการติดตามทวงถามแล้วจึงเรียกเก็บได้)

2.1.1.2.6 ค่าธรรมเนียมการเบิกถอนเงินสดผ่านบัตรเครดิต โดยผู้ออกบัตรเรียกเก็บค่าธรรมเนียมประเภทนี้ได้ไม่เกิน 3 % ของจำนวนเงินสดที่เบิกถอน และต้องเสียภาษีมูลค่าเพิ่ม ซึ่งคิดจากยอดค่าธรรมเนียมการเบิกถอนด้วย

ประวัติความเป็นมาของบัตรเครดิต กล่าวคือบัตรเครดิต (Credit Card) เป็นนวัตกรรมทางการเงินที่เปลี่ยนแปลงวิธีการใช้จ่ายของผู้คนทั่วโลก โดยมีจุดกำเนิดจากแนวคิดเรื่อง "เครดิต" หรือ "สินเชื่อ" ซึ่งหมายถึงการซื้อสินค้าหรือบริการก่อน แล้วชำระเงินภายหลัง บัตรเครดิตได้รับการพัฒนาและเปลี่ยนแปลงอย่างต่อเนื่องจากยุคแรก ๆ จนถึงปัจจุบัน ย้อนกลับไปก่อนที่จะมีบัตรเครดิตในรูปแบบปัจจุบัน เดิมระบบเครดิตมีแอบแฝงอยู่แล้วในหลายวัฒนธรรม ในสมัยโบราณ พ่อค้าหรือเจ้าของร้านมักให้ลูกค้าประจำซื้อสินค้าแบบ "เชื่อ" หรือจ่ายเงินภายหลัง ต่อมาในศตวรรษที่ 19 ระบบ "เครดิต" ถูกใช้ในบางเมืองใหญ่ เช่น ลอนดอน และนิวยอร์ก โดยพ่อค้าจะมีบัญชีเครดิตของลูกค้าเพื่อบันทึกหนี้สิน แต่จุดเริ่มต้นของบัตรเครดิตช่วงต้นศตวรรษที่ 20 โดยในปี ค.ศ. 1914 บริษัท Western Union ซึ่งเป็นบริษัทโทรเลขของสหรัฐฯ ได้ออกบัตรให้กับลูกค้าให้กับลูกค้าคนสำคัญเพื่อให้สามารถใช้บริการของบริษัทก่อนแล้วจ่ายทีหลัง ต่อมาในปี ค.ศ. 1920 ถึง ค.ศ. 1930 บริษัทน้ำมันและโรงแรมบางแห่งในสหรัฐฯ เริ่มออกบัตรสำหรับลูกค้าประจำ (Charge Cards) เพื่ออำนวยความสะดวกในการเดินทางและเข้าพัก ในปี ค.ศ. 1950 Diners Club ก่อตั้งโดย Frank McNamara และ Ralph Schneider ในสหรัฐอเมริกา โดยออกบัตร Diners Club Card ซึ่งถือเป็นบัตรเครดิตใบ

แรกของโลก บัตรนี้ใช้สำหรับชำระค่าอาหารในร้านอาหาร และลูกค้าต้องชำระคืนเต็มจำนวนในแต่ละเดือน หลังจากนั้นได้มีบริษัทเอกชนหลายได้ออกบัตรเครดิตแห่ง ปี ค.ศ. 1958 American Express ออกบัตรเครดิตของตนเอง โดยขยายการใช้งานไปยังธุรกิจที่หลากหลายมากขึ้น การเติบโตของบัตรเครดิตในปี ค.ศ. 1966 Bank of America เปิดตัว BankAmericard ซึ่งเป็นบัตรเครดิตที่สามารถใช้ได้ในหลายร้านค้า และต่อมาได้กลายเป็น Visa ในปี 1967 ธนาคารหลายแห่งในสหรัฐฯ รวมตัวกันก่อตั้ง Interbank Card Association (ICA) และเปิดตัวบัตร Master Charge ซึ่งต่อมากลายเป็น MasterCard ปี ค.ศ. 1970 ระบบเครือข่ายบัตรเครดิตเริ่มขยายไปยังประเทศต่าง ๆ ทั่วโลก และเริ่มมีการใช้แถบแม่เหล็กเพื่อเพิ่มความปลอดภัย ปี ค.ศ. 1980 มีการพัฒนา ATM และระบบเบิกเงินสดล่วงหน้า ซึ่งทำให้บัตรเครดิตมีความสะดวกมากขึ้น ปี ค.ศ. 1990 การซื้อขายออนไลน์เริ่มเป็นที่นิยม และบัตรเครดิตกลายเป็นวิธีชำระเงินหลักมีการนำ ชิปอิเล็กทรอนิกส์ (EMV Chip) มาใช้เพื่อเพิ่มความปลอดภัย (Atom, ม.ป.ป.) ปี ค.ศ. 2000 ถึง ค.ศ. 2010 เทคโนโลยี Contactless Payment หรือการชำระเงินแบบไร้สัมผัส ถูกพัฒนาทำให้สามารถแตะบัตรเพื่อชำระเงินได้โดยบัตรเครดิตเชื่อมต่อกับ E-Wallet เช่น Apple Pay, Google Pay และ Samsung Pay และในปัจจุบันมี บัตรเครดิตเสมือน (Virtual Credit Card) ที่สามารถใช้ชำระเงินออนไลน์โดยไม่ต้องมีบัตรจริง และระบบความปลอดภัยพัฒนาไปมาก เช่น การยืนยันตัวตนผ่าน OTP, Biometric และ AI Fraud Detection (Nathaniel Soto ,2566)

สำหรับประเทศไทยบัตรเครดิตเริ่มเข้ามาในประเทศไทยครั้งแรกในช่วงปี พ.ศ. 2513 ถึง พ.ศ. 2523 ในระยะแรก บัตรเครดิตที่ใช้ในไทยเป็นบัตรที่ออกโดยบริษัทต่างชาติ เช่น Diners Club, American Express และ Visa ซึ่งให้บริการกับชาวต่างชาติและนักธุรกิจที่เดินทางเข้ามาในไทย โดยบัตรเครดิตยุคแรกสามารถใช้ได้เฉพาะในโรงแรมระดับหรู ร้านอาหาร และห้างสรรพสินค้าชั้นนำ แต่คนไทยยังไม่คุ้นเคยกับแนวคิดการใช้บัตรเครดิต เนื่องจากสังคมไทยในขณะนั้นนิยมใช้เงินสดในการทำธุรกรรม ในช่วงปี พ.ศ. 2513 ต้นปี พ.ศ. 2514 ธนาคารพาณิชย์ของไทยเริ่มเห็นโอกาสในการให้บริการบัตรเครดิต จุดเริ่มต้นธนาคารกรุงเทพ และ ธนาคารไทยพาณิชย์ เป็นธนาคารกลุ่มแรกที่เริ่มออกบัตรเครดิตให้กับลูกค้าคนไทย อย่างไรก็ตาม ในช่วงแรก บัตรเครดิตยังคงจำกัดเฉพาะกลุ่มลูกค้าระดับสูง เช่น นักธุรกิจ ข้าราชการระดับสูง และบุคคลที่มีรายได้มั่นคง หลังจากนั้น ปี พ.ศ. 2533 ถึง ปี พ.ศ. 2543 เศรษฐกิจของไทยเติบโตอย่างรวดเร็ว ธนาคารและสถาบันการเงินเริ่มแข่งขันกันออกบัตรเครดิต มีการร่วมมือกับเครือข่ายบัตรเครดิตระดับโลก เช่น Visa, MasterCard, JCB และ American Express เพื่อขยายการใช้งาน การใช้บัตรเครดิตเริ่มแพร่หลายในหมู่คนชั้นกลาง และมีการออกบัตรร่วมกับธุรกิจต่าง ๆ เช่น ห้างสรรพสินค้า สายการบิน และปั๊มน้ำมัน ในปี 2540 ประเทศไทยประสบ วิกฤตเศรษฐกิจต้มยำกุ้ง ส่งผลให้หนี้สินจากบัตรเครดิตเพิ่มขึ้น และหลายธนาคารต้องปรับนโยบายการปล่อยสินเชื่อให้รัดกุมขึ้น หลังวิกฤต ธนาคารแห่งประเทศไทย (ธปท.) ได้กำหนด

กฎระเบียบที่เข้มงวดเกี่ยวกับการออกบัตรเครดิต เช่น กำหนด รายได้ขั้นต่ำของผู้ถือบัตร และ อัตราดอกเบี้ยสูงสุด และในปัจจุบันเทคโนโลยีทางการเงินเริ่มเข้ามามีบทบาท บัตรเครดิตกลายเป็นวิธีการชำระเงินที่สำคัญทั้งในร้านค้าและออนไลน์ในประเทศไทย บัตรเครดิตกลายเป็นส่วนหนึ่งของชีวิตประจำวันของคนไทย จนกลายเป็นเครื่องมือทางการเงินที่ใช้กันอย่างแพร่หลาย โดยเฉพาะในกลุ่มวัยทำงานและกลุ่มคนชั้นสูง มีบัตรเครดิตเฉพาะกลุ่ม เช่น บัตรเครดิตสำหรับนักเดินทาง, บัตรเครดิตสะสมแต้ม, บัตรเครดิตสำหรับธุรกิจ และบัตรเครดิตที่เน้นการคืนเงิน (Cashback) บัตรเครดิตสามารถใช้จ่ายผ่านระบบออนไลน์ และสามารถแบ่งชำระ (Installment) ได้ ปัจจุบันบัตรเครดิตได้รับการพัฒนาให้ทันสมัยมากขึ้นมีบัตรเครดิตแบบ Contactless (ไร้สัมผัส) และการเชื่อมต่อบัตรกับกระเป๋าเงินดิจิทัล (E-Wallet) เช่น Apple Pay, Google Pay และ TrueMoney Wallet และมีความปลอดภัยสูงขึ้นจากเทคโนโลยีดิจิทัลและมาตรการควบคุมจากธนาคารแห่งประเทศไทย (Krungsri Consumer, 2566)

2.1.2 เกณฑ์พิจารณาการอนุมัติบัตร (Credit Card Approval Criteria)

การที่ได้มาบัตรเครดิตได้ต้องไปสมัครกับผู้ออกบัตร และจะผู้ออกบัตรต้องการพิจารณาอนุมัติบัตรเครดิตทำตามหลักเกณฑ์ตามแต่ละผู้ออกบัตร (Issuer) ของสถาบันนั้นซึ่งคุณสมบัติในการที่จะอนุมัติจะต้องสอดคล้องกับรายได้และลักษณะการใช้จ่ายมากที่สุด บัตรเครดิตถือว่ามีความเสี่ยงสูงที่จะผิดนักชำระ จึงมีดอกเบี้ยบัตรเครดิตแพงเป็นอันดับต้นๆ เนื่องจากหนี้ที่เกิดจาการใช้จ่ายบัตรเครดิตเป็นสินเชื่อแบบไม่มีหลักประกัน และไม่มีข้อตกลงเลยว่าจะใช้กรอบเวลาในการจ่ายคืนเท่าไร ผู้ออกบัตรบริหารจัดการได้ยากกว่า ไม่สามารถรู้เลยว่าจะได้เงินกลับมาได้เมื่อไร หรือกลายเป็นหนี้เสียหรือไม่ ดอกเบี้ยบัตรเครดิตจึงสูงกว่าสินเชื่อที่มีหลักทรัพย์ค้ำประกัน เป็นเหตุผลที่แบงค์ชาติกำหนดว่า คุณต้องจ่ายเงินขั้นต่ำอย่างน้อยกี่เปอร์เซ็นต์ของยอดหนี้ เดิมทีการผู้ออกบัตร จะพิจารณาและอนุมัติในการออกบัตรเครดิตนั้นข้องข้างเสี่ยงเกิดหนี้เสียสูง ดังนั้นจึงต้องข้อมูลประกอบในการพิจารณาได้แก่ข้อมูล Credit Scoring หรือคะแนนเครดิต สำหรับประเมินความสามารถในการำระหนี้จากเจ้าของผู้ขอยื่นอนุมัติบัตรเครดิตและวงเงิน มีรายได้เพียงพอไหม พฤติกรรมจ่ายหนี้ในอดีตเป็นอย่างไร โดยในปัจจุบันของโดยทั่วไปประเทศไทยนำข้อมูล Credit Scoring มาจากศูนย์เครดิตบูโรควบคู่กับข้อมูลคุณสมบัติอื่นๆ เช่น อัตราส่วนภาระหนี้ต่อรายได้รวม (DTI) หรือ ภาระหนี้สิน (DSR) เป็นต้น ล้วนถูกกำหนดตามแต่ละผู้ออกบัตร ได้แก่ สถาบันทางการเงิน องค์กร หน่วยงาน หรือบริษัทธุรกิจเป็นผู้กำหนดต่างกันไป (บริษัท บัตรกรุงไทย จำกัด, 2567) หลักเกณฑ์พิจารณาการอนุมัติบัตรเครดิต มีดังนี้

2.1.2.1 ข้อมูลส่วนบุคคล (Personal Information)

2.1.2.1.1 อายุ สำหรับบุคคลทั่วไปผู้สมัครต้องมีอายุอยู่ในช่วงที่กำหนด เช่น 20-65 ปี เพื่อให้มั่นใจว่าผู้สมัครมีศักยภาพในการใช้และชำระคืนหนี้ และอายุ 18 ปีขึ้นไป สำหรับ

บัตรเครดิตเสริม (บัตรที่ออกให้บุคคลอื่นใช้ร่วมกับเจ้าของบัตรหลัก) ส่วยอายุไม่เกิน **65 - 70 ปี** (อาจมีข้อจำกัดเรื่องการให้บัตรเครดิตกับผู้สูงอายุ)

2.1.2.1.2 สัญชาติและสถานะทางกฎหมาย ทางธนาคารกำหนดให้ผู้สมัครต้องเป็นพลเมืองของประเทศไทยหรือชาวต่างชาติที่มีถิ่นพำนักและใบอนุญาตทำงานและพำนักถาวร ต้องพิจารณาจากเอกสารที่ทางราชการออกประกอบ ได้แก่ สำเนาบัตรประชาชน / สำเนาพาสปอร์ต (Passport) สำหรับชาวต่างชาติ ใบอนุญาตทำงานและพำนักถาวร เป็นต้น

2.1.2.1.3 ที่อยู่ที่สามารถติดต่อได้ ต้องมีที่อยู่แน่นอนเพื่อลดความเสี่ยงในการติดตามหนี้ ต้องมีที่อยู่ที่แน่นอน เช่น บ้านเช่า บ้านของตัวเอง และคอนโดมิเนียม ที่อยู่ต้องสามารถตรวจสอบได้เพื่อป้องกันการฉ้อโกงแสดงเอกสารเพิ่มเติม เช่น ทะเบียนบ้าน

2.1.2.1.4 สถานภาพสมรส ธนาคารอาจพิจารณาความมั่นคงทางการเงินของครอบครัว หากแต่งงานแล้วอาจต้องแสดงรายได้ของคู่สมรส

2.1.2.2 รายได้และความสามารถในการชำระหนี้ (Income & Financial Stability)

2.1.2.2.1 รายได้ขั้นต่ำที่กำหนด โดยทั่วไป ธนาคารกำหนดรายได้ขั้นต่ำ เช่น 15,000 บาทต่อเดือนสำหรับบัตรระดับพื้นฐาน และสูงขึ้น 30,000 - 50,000 บาท/เดือน สำหรับบัตรระดับพรีเมียม รายได้ขั้นต่ำ 100,000 บาทขึ้นไป

2.1.2.2.2 แหล่งที่มาของรายได้ รายได้จากเงินเดือน การประกอบธุรกิจ หรืออาชีพอิสระต้องสามารถตรวจสอบได้ผ่านเอกสารรับรอง ได้แก่ รายการเดินบัญชี (Statement) ย้อนหลัง 3-6 เดือน หรือ เอกสารเสียภาษี (50 ทวิ หรือ ภ.พ.30) ส่วน เจ้าของธุรกิจ/อาชีพอิสระ ใช้สำเนาทะเบียนการค้า หรือเอกสารจดทะเบียนบริษัทรับรอง

2.1.2.2.3 อัตราส่วนหนี้สินต่อรายได้ (DTI: Debt-to-Income Ratio) สถาบันการเงินใช้ตัวชี้วัดนี้เพื่อประเมินความสามารถในการรับภาระหนี้เพิ่มเติม โดยปกติไม่ควรเกิน 40-50% ของรายได้ต่อเดือน หากภาระหนี้สูงเกินไป อาจถูกปฏิเสธการอนุมัติ

2.1.2.2.4 สถานภาพสมรส ธนาคารอาจพิจารณาความมั่นคงทางการเงินของครอบครัว หากแต่งงานแล้วอาจต้องแสดงรายได้ของคู่สมรส

2.1.2.2.5 ระยะเวลาทำงาน ผู้สมัครต้องมี อายุงานขั้นต่ำ 6 เดือน – 1 ปี เพื่อแสดงถึงความมั่นคงทางการเงิน กรณีเพิ่งเปลี่ยนงานใหม่ อาจต้องแสดงเอกสารเกี่ยวกับงานเก่าเพื่อให้แน่ใจว่ามีรายได้ต่อเนื่อง

2.1.2.3 ประวัติทางการเงิน (Credit History & Credit Score)

2.1.2.3.1 ข้อมูลเครดิตจากหน่วยงานเครดิตบูโร หรือ หน่วยงานราชการที่เกี่ยวข้อง ธนาคารพิจารณาประวัติการชำระหนี้ เช่น การค้างชำระหนี้ การถูกฟ้องร้อง หรือการล้มละลาย หากมี ประวัติอาจถูกปฏิเสธการสมัคร

2.1.2.3.2 เครดิตสกอร์ คะแนนเครดิตใช้เป็นเกณฑ์หลักในการประเมินความเสี่ยง โดยคะแนนที่สูงสะท้อนถึงความน่าเชื่อถือทางการเงิน ธนาคารจะตรวจสอบข้อมูลเครดิตของผู้สมัครจากหน่วยงาน เครดิตบูโร (National Credit Bureau – NCB) คะแนนเครดิตที่ดีมักจะอยู่ในช่วง 650 – 900 คะแนน

2.1.2.3.3 อัตราส่วนหนี้สินต่อรายได้ (DTI: Debt-to-Income Ratio) เปอร์เซ็นต์ของรายได้รวมต่อเดือนของตัวเองที่นำไปชำระหนี้รายเดือน อัตราหนี้สินไม่ควรเกิน 40% ของรายได

2.1.2.3.4 พฤติกรรมการชำระหนี้ ผู้สมัครต้องมีประวัติการชำระหนี้ที่ดี เช่น ไม่มีประวัติชำระล่าช้า หรือผิดนัดชำระเกิน 30 วันขึ้นไป หากผู้ที่มีประวัติการชำระเงินตรงเวลามักได้รับการอนุมัติง่ายกว่าผู้ที่เคยมีการชำระล่าช้า ถ้ามีประวัติชำระล่าช้าหลายครั้ง ธนาคารอาจเพิ่มเงื่อนไข เช่น ขอผู้ค้ำประกัน หรือกำหนดวงเงินบัตรต่ำกว่าที่ขอไว้

2.1.2.3.5 พฤติกรรมการใช้วงเงิน หากใช้วงเงินเต็มจำนวนบ่อยครั้งและไม่มีการชำระเต็มจำนวน ธนาคารอาจมองว่าเป็นความเสี่ยงสูง หรือธนาคารอาจไม่อนุมัติบัตรใหม่ บางธนาคารจำกัดการอนุมัติบัตรใหม่ให้เฉพาะผู้ที่มีบัตรเครดิตไม่เกิน 3-5 ใบ เท่านั้น

2.1.2.3.5 จำนวนบัตรเครดิตที่ถืออยู่ ผู้สมัครที่มีบัตรเครดิตหลายใบอาจได้รับการพิจารณาอย่างเข้มงวดขึ้นเนื่องจากภาระหนี้ที่อาจเพิ่มขึ้น

2.1.2.4 การพิจารณาประเภทบัตรเครดิตและวงเงิน

2.1.2.4.1 ประเภทของบัตร การพิจารณาขึ้นอยู่กับระดับของบัตร ได้แก่

ก. บัตรเครดิตพื้นฐาน (Classic, Gold) เหมาะสำหรับคนทั่วไปที่มีรายได้ปานกลาง

ข. บัตรเครดิตพรีเมียม (Platinum, Signature) ต้องมีรายได้สูงและเครดิตสกอร์ดี

ค. บัตรเครดิตร่วม (Co-branded Cards) บัตรที่ออกโดยธนาคารร่วมกับบริษัทต่าง ๆ เช่น สายการบิน, ห้างสรรพสินค้า

ง. บัตรเครดิตสำหรับธุรกิจ (Business Credit Cards ออกให้สำหรับเจ้าของกิจการ

2.1.2.4.2 วงเงินอนุมัติ วงเงินบัตรเครดิตจะถูกกำหนดโดยทั่วไปจะอยู่ที่ 1-5 เท่าของรายได้ต่อเดือน ทั้งนี้ขึ้นอยู่กับปัจจัยอื่น ๆ เช่น ประวัติสินเชื่อและเครดิตสกอร์ หากผู้สมัครมีภาระหนี้สูง วงเงินอาจถูกลดลง

ตารางที่ 2-1 เกณฑ์พิจารณาจากรายได้ของผู้ขอบัตรเครดิต

รายได้ต่อเดือน	**วงเงินอนุมัติ**
ตั้งแต่ 15,000 บาท แต่น้อยกว่า 30,000 บาท	1.5 เท่าของรายได้เฉลี่ยต่อเดือน
ตั้งแต่ 30,000 บาท แต่น้อยกว่า 50,000 บาท	3 เท่าของรายได้เฉลี่ยต่อเดือน
ตั้งแต่ 50,000 บาทขึ้นไป	5 เท่าของรายได้เฉลี่ยต่อเดือน

2.1.3 อัตราส่วนหนี้สินต่อรายได้ (DTI: Debt-to-Income Ratio)

2.1.3.1 อัตราส่วนหนี้สินต่อรายได้ (Debt-to-Income Ratio: DTI)

เป็นตัวชี้วัดทางการเงินที่ใช้ในการประเมินความสามารถของบุคคลในการชำระหนี้ โดยคำนวณจากสัดส่วนของภาระหนี้สินรายเดือนเมื่อเทียบกับรายได้รวมต่อเดือน ค่า DTI มีความสำคัญอย่างมากในการเป็นเกณฑ์พิจารณาสินเชื่อ เพราะช่วยให้สถาบันการเงินสามารถวิเคราะห์ว่าผู้ขอสินเชื่อมีภาระหนี้มากเกินไปหรือไม่ ความเสี่ยงโอกาสเกิดหนี้เสีย (Non-Performing Loan: NPL) และเป็นตัวชี้วัดว่ามีความสามารถในการชำระหนี้ใหม่เพียงพอหรือไม่ รวมถึงการให้ผลิตภัณฑ์หรือปรับอัตราดอกเบี้ยให้เหมาะสมกับระดับความเสี่ยงของลูกค้า

2.1.3.2 สูตรการคำนวณ DTI (Debt-to-Income Ratio: DTI)

$$DTI = \left(\frac{\text{ภาระหนี้สินต่อเดือนทั้งหมด}}{\text{รายได้รวมต่อเดือน}}\right) \times 100 \qquad (2\text{-}1)$$

เมื่อ ภาระหนี้สินต่อเดือนทั้งหมด แทน หนี้สิ้นรวมถึงเงินกู้ต่าง ๆ ทั้งหมดต่อเดือน เช่น ค่าผ่อนบ้าน ค่าผ่อนรถ หนี้บัตรเครดิต ค่าผ่อนสินเชื่อส่วนบุคคล ฯลฯ

รายได้รวมต่อเดือน แทน รายได้รวมทั้งหมดที่มีต่อเดือน เช่น รายได้รวมต่อเดือน รวมถึงเงินเดือน ค่าคอมมิชชั่น รายได้จากธุรกิจ และรายได้อื่น ๆ

2.1.3.3 เกณฑ์พิจารณาค่า DTI (Debt-to-Income Ratio: DTI) ในการประเมินความสามารถในการชำระหนี้ของผู้ขอสินเชื่อ ค่า DTI ที่ต่ำ (เช่น ต่ำกว่า 40%) จะช่วยให้โอกาสในการได้รับอนุมัติสินเชื่อสูงขึ้น ในขณะที่ค่า DTI ที่สูงเกินไป (มากกว่า 50-60%) อาจทำให้ถูกปฏิเสธสินเชื่อ ในการอนุมัติสินเชื่อแต่ละสถาบันการเงินอาจกำหนดเกณฑ์ DTI ที่แตกต่างกัน สินเชื่อบางประเภท เช่น สินเชื่อบ้าน อาจอนุมัติให้แม้มี DTI สูงกว่า 50% หากผู้ขอสินเชื่อมีรายได้ที่มั่นคงแต่โดยทั่วไปใช้เกณฑ์ [11] ดังตารางที่ 2.1

ตารางที่ 2-2 เกณฑ์ DTI (Debt-to-Income Ratio: DTI)

ค่า DTI (%)	ค่า DTI (%)	โอกาสอนุมัติสินเชื่อ
ต่ำกว่า 37%	ดีมาก มีภาระหนี้น้อย	สูงมาก
37% - 42%	อยู่ในเกณฑ์ดี สินเชื่อมักได้รับการอนุมัติ	สูง
43% - 49%	ภาระหนี้ค่อนข้างสูง	อาจได้รับอนุมัติ ขึ้นอยู่กับเงื่อนไขเพิ่มเติม
50% ขึ้นไป	หนี้สินมากเกินไป เสี่ยงต่อการผิดนัดชำระ	ต่ำ

2.1.4 เครดิตบูโร (National Credit Bureau – NCB)

เครดิตบูโร หรือบริษัท ข้อมูลเครดิตแห่งชาติ จำกัด เป็นสถาบันที่ทำหน้าที่เป็นศูนย์กลางในการจัดเก็บรวบรวมข้อมูลทางการเงิน ประวัติการชำระหนี้ บัญชีสินเชื่อและประวัติการและชำระสินเชื่อทุกประเภทของบุคคลธรรมดาและนิติบุคคล เป็นข้อมูลข้อมูลเชิงบวก เช่น ผิดนัดชำระ หรือข้อมูลเชิงลบ เช่นยอดสินเชื่อที่ชำระเรียบร้อยแล้ว ซึ่งส่งมาจากสถาบันการเงิน ธนาคาร และบริษัทผู้ให้บริการสินเชื่อต่างๆ เพื่อใช้เป็นข้อมูลอ้างอิงในการประเมินความสามารถในการชำระหนี้ของผู้ขอสินเชื่อ ลดโอกาสในการปล่อยสินเชื่อให้กับผู้ที่มีความเสี่ยงสูง โดยข้อมูลจากเครดิตบูโรจะถูกนำไปใช้ในกระบวนการพิจารณาอนุมัติสินเชื่อของธนาคารหรือสถาบันการเงินให้เป็นธรรมและโปร่งใสมากขึ้น โดยข้อมูลที่จัดเก็บหรือรายงานในเครดิตบูโรแบ่งออกเป็น 2 ส่วน คือ

2.1.2.1 ข้อมูลบ่งชี้ คือข้อเท็จจริงที่บ่งชี้ถึงตัวลูกค้า ได้แก่ ข้อมูลส่วนบุคคล เช่น ชื่อ-นามสกุล เลขที่บัตรประจำตัวประชาชน หมายเลขหนังสือเดินทาง และวันเดือนปีเกิด ซึ่งไม่มีการจัดเก็บหมายเลขโทรศัพท์ และข้อมูลที่อยู่ที่ลูกค้าแจ้งกับสถาบันการเงินและบริษัทที่เป็นสมาชิกเครดิตบูโร

2.1.2.2 ข้อมูลสินเชื่อที่ได้รับอนุมัติและประวัติการชำระหนี้ จำแนกเป็นรายบัญชีที่มีอยู่ในแต่ละสถาบันการเงินและบริษัทสมาชิก โดยมีข้อมูลที่สำคัญดังนี้ ประวัติการกู้เงิน สรุปข้อมูลบัญชีสินเชื่อเชื่อบ้าน บัตรเครดิต สินเชื่อส่วนบุคคล ซึ่งจะบอกว่าลูกค้ามีสินเชื่ออยู่ทั้งหมดกี่บัญชี มีจำนวนบัญชีที่ใช้สิทธิแก้ไขข้อมูลหรือโต้แย้งกี่บัญชี ประเภทและเลขที่บัญชีของสินเชื่อ ชื่อผู้ให้สินเชื่อ วงเงินที่ได้รับอนุมัติ วงเงินสินเชื่อที่ได้รับ การค้างชำระ จำนวนงวดที่เหลือ และวงเงินที่ใช้ไป สถานะของบัญชี เช่น ปกติ ปิดบัญชี พักชำระหนี้ ค้างชำระหนี้ รายละเอียดการชำระหนี้ ซึ่งจะแสดงประวัติการชำระหนี้ที่ผ่านมาหรือชำระหนี้ย้อนหลังเป็นระยะเวลาหนึ่ง (เช่น 36 เดือนย้อนหลัง) ทั้งที่ชำระตรง ชำระล่าช้า หรือผิดนัดชำระ ข้อมูลอื่น ๆ เช่น วันที่เปิดบัญชี วันที่ชำระหนี้ล่าสุด วันที่ปิดบัญชี และวันที่ปรับปรุงโครงสร้างหนี้ [9]

2.1.3 ระบบอนุมัติบัตรเครดิต (Credit Card approval Application)

หนึ่งในกระบวนการสำคัญในการสำหรับธุรกิจบัตรเครดิต ซึ่งมีบทบาทสำคัญต่อธุรกิจธนาคารและสถาบันการเงินเป็นเครื่องมือที่ใช้พิจารณาว่าผู้สมัครบัตรเครดิตมีคุณสมบัติเพียงพอและมีความสามารถในการชำระหนี้ไม่ก่อหนี้เสีย (Non-Performing Loan: NPL) ค้างในระบบหรือไม่ รวมถึงต้องพิจารณาได้อย่างรวดเร็วและแม่นยำสูง อย่างไรก็ตาม ความท้าทายหลักของธนาคารและสถาบันการเงินคือการคัดกรองลูกค้าที่เหมาะสมการอนุมัติบัตรเครดิตยังคงเป็นกระบวนการที่ซับซ้อน ซึ่งต้องอาศัยการวิเคราะห์ข้อมูลหลายปัจจัย จากระบบงานภายในและระบบงานภายนอกสถาบันการเงิน ข้อมูลเครดิตบูโร รวมถึงประวัติทางการเงิน รายได้ และพฤติกรรมการใช้จ่ายเพื่อใช้ประมวลผลข้อมูลและคำนวณค่าต่างๆ ตามเงื่อนไขเกณฑ์พิจารณาที่สถาบันการเงินถูกกำหนดไว้

2.2 การเรียนรู้ของเครื่อง (Machine Learning)

2.2.1 ความหมายของการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง (Machine Learning) เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่ พัฒนามาจากการศึกษาการเรียนรู้จำแบบเกี่ยวข้องกับการศึกษาและการสร้างอัลกอริทึมที่สามารถ ที่มุ่งเน้นให้คอมพิวเตอร์สามารถเรียนรู้จากข้อมูลศาสตร์ไว้อย่างหลากหลายแขนง และนำเข้าข้อมูลไปใช้ตัดสินใจอัตโนมัติ ปราศจากการทำงานตามลำดับคำสั่งโปรแกรมแบบกำหนดกฎตายตัว แต่สามารถปรับปรุงและพัฒนาตนเองได้จากประสบการณ์ที่ได้รับ การเรียนรู้ของเครื่องทำให้คอมพิวเตอร์สามารถทำการวิเคราะห์ข้อมูล คาดการณ์แนวโน้ม และตัดสินใจโดยอิงจากข้อมูลที่มีอยู่ ซึ่งช่วยให้เกิดการทำงานที่แม่นยำและมีประสิทธิภาพมากขึ้น ปัจจุบัน การเรียนรู้ของเครื่องถูกนำมาใช้ในหลายอุตสาหกรรม เช่น ธุรกิจ การแพทย์ การเงิน และระบบสนับสนุนการตัดสินใจในองค์กรต่างๆ [24]

2.2.2 กระบวนการวิเคราะห์ข้อมูลด้วย CRISP-DM

การเรียนรู้ของเครื่อง (Machine Learning) อาศัยกระบวนการทางสถิติและคณิตศาสตร์เพื่อนำข้อมูลที่มีอยู่มาใช้ในการเรียนรู้ ซึ่งสามารถกระบวนการมาตรฐานในการวิเคราะห์ข้อมูลด้วย CRISP-DM มาใช้ในการสร้างอัลกอริทึม

กระบวนการมาตรฐานในการวิเคราะห์ข้อมูลด้วย CRISP-DM พัฒนาขึ้นในปี ค.ศ. 1996 โดยความร่วมมือกันของ 3 บริษัท ได้แก่ DaimlerChrysler SPSS และ NCR เรียกว่า “CRISP-DM” หรือ “Cross-Industry Standard Process for Data Mining” โดยกระบวนการ CRISP-DM จะประกอบด้วย 6 กระบวนการ โดยแต่ละขั้นตอนจะเป็นขั้นตอนที่ต่อเนื่องกัน หมายความว่าผลลัพธ์จากขั้นตอน ณ ขณะนั้นจะเป็นข้อมูลตั้งต้นให้กับขั้นตอนถัดไป (Thapanee Boonchob, 2564) สามารถอธิบายกระบวนการได้ดังต่อไปนี้

2.2.2.1 Business Understanding หรือกระบวนการเข้าใจในปัญหาที่แท้จริง หรือโจทย์ที่ต้องการหาคำตอบหรือคำอธิบายเพื่อที่จะวางแผนในการทำงาน ซึ่งต้องเกิดจากการประเมิน

สถานการณ์ปัจจุบัน รวมไปถึงการมองหาแนวทางการแก้ปัญหา พร้อมกับการประเมินผลกระทบที่ได้จากการแนวทางใหม่ที่กำลังจะเกิดขึ้น

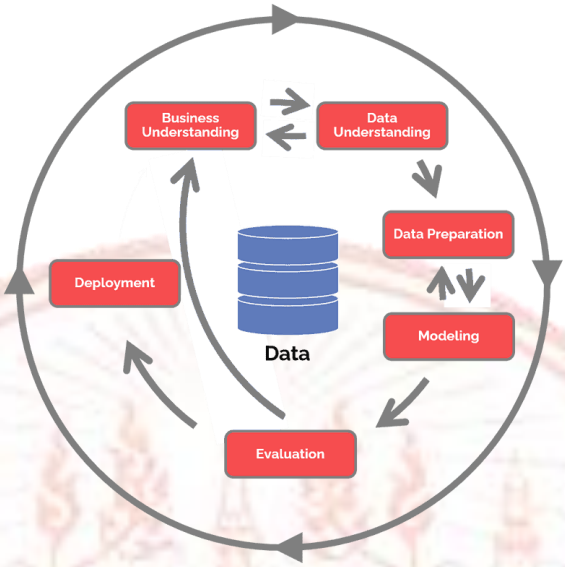
2.2.2.2 Data Understanding หรือกระบวนการเข้าใจในข้อมูลที่มี โดยขั้นตอนนี้จะเป็นขั้นตอนที่ได้จากการเก็บข้อมูลต่างๆ เช่น ฐานข้อมูล, API, หรือไฟล์ข้อมูล หลังจากนั้นจะต้องพิจารณาว่าข้อมูลที่มีนั้นเพียงพอต่อการนำไป ข้อมูลต้องมีคุณภาพและความถูกต้องเพื่อให้ผลลัพธ์ของโมเดลมีความน่าเชื่อถือ วิเคราะห์ต่อหรือไม่ หรือข้อมูลที่มีนั้นส่วนไหนที่ไม่จ าเป็นต่อการนำไปวิเคราะห์

2.2.2.3 Data Preparation หรือกระบวนการเตรียมข้อมูล เป็นขั้นตอนของการแปลงข้อมูลให้พร้อมเพื่อนำไปวิเคราะห์โดยอาจจะต้องมีการทำข้อมูลให้ถูกต้องหรือทำความสะอาดข้อมูล (data cleaning) ยกตัวอย่างเช่น การปรับขนาดข้อมูล (Normalization หรือ Standardization) การแปลงข้อมูลให้อยู่ในช่วงเดียวกัน การเติมข้อมูลที่ขาดหายไป เลือกตัวแปรหรือการเลือกคุณลักษณะ (Feature Selection) ที่มีผลต่อการเรียนรู้ และการทำข้อมูลให้ อยู่ในรูปแบบเดียวกัน เป็นต้น ซึ่งขั้นตอนนี้จะเป็นขั้นตอนที่ใช้เวลาในการทำมากที่สุดใน 6 กระบวนการของ CRISP-DM

2.2.2.4 Modeling หรือกระบวนการขั้นตอนวิเคราะห์ข้อมูล เป็นขั้นตอนที่พยายามหาคำตอบโดยการสร้างแบบจำลองขึ้น โดยนำข้อมูลที่ผ่านการเตรียมมาใช้ในการฝึกโมเดล โดยใช้เทคนิคต่างๆ เช่น Linear Regression, Decision Tree, Neural Networks ปรับค่าพารามิเตอร์ของโมเดลให้เหมาะสมเพื่อให้ได้ผลลัพธ์ที่แม่นยำ ในบางครั้งอาจจำเป็นต้องกลับไปใน ขั้นตอนการเตรียมข้อมูลอีกครั้ง เพื่อที่ว่าจะได้ตัวแบบจำลองที่เหมาะสมที่สุดกับโจทย์ที่เราต้องการหาคำตอบ

2.2.2.5 Evaluation หรือกระบวนการวัดประสิทธิภาพของแบบจำลองว่ามีความ ถูกต้องแม่นยำมากน้อยเพียงต่อการนำไปใช้งานได้ขนาดไหน ต้องอาศัยสูตรคำนวณทางคณิตศาสตร์มาช่วยอธิบาย ซึ่งจะสรุปออกมาในรูปแบบตาราง Confusion Matrix และใช้ตัวชี้วัดประสิทธิภาพ เช่น Accuracy, Precision, Recall และ F1-Score เป็นต้น และปรับปรุงโมเดลให้มีประสิทธิภาพมากขึ้น

2.2.2.6 Deployment หรือกระบวนการนำแบบจำลองไปใช้จริงในการแก้ปัญหาจากขั้นตอนที่ 1 เป็นการย้ายโมเดลจากสภาพแวดล้อมการพัฒนาและทดสอบ ไปสู่สภาพแวดล้อมในการทำงานจริงเมื่อข้อมูลใหม่เข้ามานั้นระบบจะสามารถทำงานได้ปกติหรือไม่ ประสิทธิภาพคงเดิมหรือไม่ เพื่อให้ใช้งานได้ตามวัตถุประสงค์



ภาพที่ 2-1 กระบวนการ CRISP-DM

2.2.3 ประเภทของการเรียนรู้ของเครื่อง (Machine Learning) สามารถแบ่งออกเป็น 3 ประเภทหลัก (Narut, 2562) ได้แก่

2.2.3.1 การเรียนรู้แบบมีผู้สอน (Supervised learning) เป็นเทคนิคการเรียนรู้ของเครื่องเป็นรูปแบบการเรียนรู้ที่มีชุดข้อมูลที่มีป้ายกำกับ (Labeled Data) ซึ่งประกอบด้วยอินพุต (Input) และเอาต์พุต (Output) ที่ถูกต้อง อัลกอริทึมเรียนรู้โดยใช้ข้อมูลตัวอย่างเพื่อสร้างแบบจำลองที่สามารถทำนายผลลัพธ์ใหม่ได้ โดยการแบ่งชุดข้อมูลที่มีออกเป็น 2 ชุด คือ ชุดข้อมูลที่ใช้สอนให้เครื่องเรียนรู้ (Training Data) และชุดข้อมูลที่ใช้ทดสอบ (Test Data) โดยที่ ชุดข้อมูลที่ใช้สอนให้เครื่องเรียนรู้มีเป้าหมายเพื่อที่หาแบบจำลองในการพยากรณ์และหาผลลัพธ์ที่ได้ ออกมา ส่วนชุดข้อมูลที่ใช้ทดสอบนั้นมีเป้าหมายเพื่อใช้ในการทดสอบประสิทธิภาพและความถูกต้อง ของแบบจำลองพยากรณ์ว่ามีความน่าเชื่อถือมากน้อยเพียงใด โดยหลักการของการเรียนรู้แบบมี ผู้สอนมีหลายวิธีการ เช่น Decision Tree , Naïve Bayes , Neural Network, Linear Regression, Logistic Regression เป็นต้น

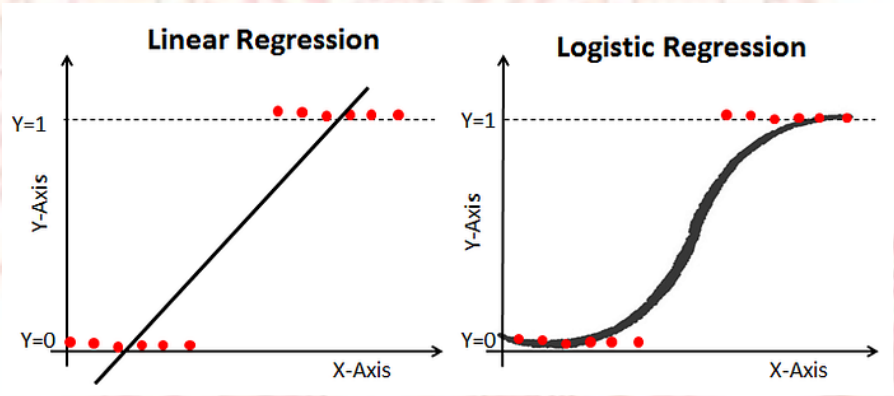
2.2.3.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) เป็นเทคนิคการเรียนรู้อีกรูปแบบหนึ่งของการเรียนรู้ของเครื่องโดยที่ไม่ จำเป็นต้องมีการสอน แต่จะเป็นการใช้ชุดข้อมูลที่มีทั้งหมดในการให้เครื่องทำการเรียนรู้ โดยไม่จำเป็น ต้องระบุผลที่ต้องการ (Unlabeled Data) โดยการเรียนรู้แบบไม่มีผู้สอน นั้นตรงกันข้ามกับการเรียนรู้แบบมีผู้สอน คือการ เรียนรู้แบบไม่มีผู้สอนนี้จะไม่มีการระบุผลที่ต้องการไว้ก่อน ให้เครื่องพยายามหาความสัมพันธ์จาก ข้อมูลเอาเอง โดยการเรียนรู้การเรียนรู้แบบไม่มีผู้สอนใช้สำหรับการจัดกลุ่มข้อมูล (Clustering) การหากฎความสัมพันธ์ (Association Rules) หรือ การลดมิติของข้อมูล

(Dimensionality Reduction) โดยมีหลายวิธีการ เช่น K-Means Clustering Hierarchical Clustering Principal Component Analysis (PCA) เป็นต้น

2.2.3.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning: RL) เป็นการใช้หลักการให้รางวัล (Reward) และบทลงโทษ (Penalty) เพื่อให้โมเดลเรียนรู้โดยการโต้ตอบกับสภาพแวดล้อม เหมาะสำหรับปัญหาที่ต้องการการตัดสินใจเป็นลำดับขั้น เช่น เกม AI และหุ่นยนต์อัตโนมัติ โดยมีหลายวิธีการ เช่น Q-Learning และ Deep Q-Networks (DQN) เป็นต้น

2.3 อัลกอริทึม

2.3.1 Logistic Regression เป็นเทคนิคการสร้างสมการทางตณิตศาสตร์ดพื่อทำการวิเคราะห์ทางสถิติที่ใช้สำหรับการจำแนกประเภท (Classification) ออกเป็น 2 กลุ่ม โดย Logistic Regression จะรับค่าตัวแปรเป้าหมายคือ Binary เช่น “ใช่” หรือ “ไม่ใช่”



ภาพที่ 2-2 เทคนิค Logistic Regression

โมเดล Logistic Regression ใช้ฟังก์ชันโลจิสติก (Logistic Function) หรือ Sigmoid Function เพื่อแปลงค่าผลลัพธ์ที่เป็นจำนวนจริงให้อยู่ในช่วง 0 ถึง 1 ซึ่งสามารถใช้ตีความเป็น ความน่าจะเป็น (Probability) ของการเป็นคลาสหนึ่ง ๆ ได้โดยพยายาม fit ชุดข้อมูลของเราให้เข้ากับ sigmoid function และเปรียบเทียบกับ Threshold (ค่าเกณฑ์, ค่าเริ่มต้นมักเป็น 0.5)

1) ถ้า $P(y = 1|X) \geq 5$ ทำนายว่าเป็น **Class 1**
2) ถ้า $P(y = 1|X) < 5$ ทำนายว่าเป็น **Class 0**

โมเดล Logistic Regression เรียนรู้ค่า Weights ($\boldsymbol{W_i}$) โดยใช้ Maximum Likelihood Estimation (MLE) และ Gradient Descent

เนื่องจากลักษณะที่เรียบง่ายและมีประสิทธิภาพของ Logistic Regression จึงไม่ต้องการพลังประมวลผลสูงมาก ทำให้เข้าใจง่ายและตีความได้ง่าย ประสิทธิภาพดีเมื่อตัวแปรอินพุตเป็นเชิงเส้น และเหมาะกับปัญหาการจำแนกประเภทที่มีข้อมูลไม่ซับซ้อน แต่ไม่เหมาะกับข้อมูลที่มีความซับซ้อนหรือไม่เป็นเชิงเส้น (อาจต้องใช้ Polynomial Features หรือ Neural Networks) 0tมีปัญหาถ้าข้อมูลมีความไม่สมดุลของคลาส (Imbalanced Data) และไวต่อ Outliers และ Feature Scaling (ต้องทำ Normalization หรือ Standardization) (Pasith Thanapatpisarn, 2564)

2.1.3.1 สมการของ Logistic Function (Sigmoid Function)

$$\sigma(z) = \frac{1}{1+e^2} \quad (2\text{-}2)$$

$$z = w_0 + w_1x_1 + w_2x_{02} + \cdots + w_nx_n \quad (2\text{-}3)$$

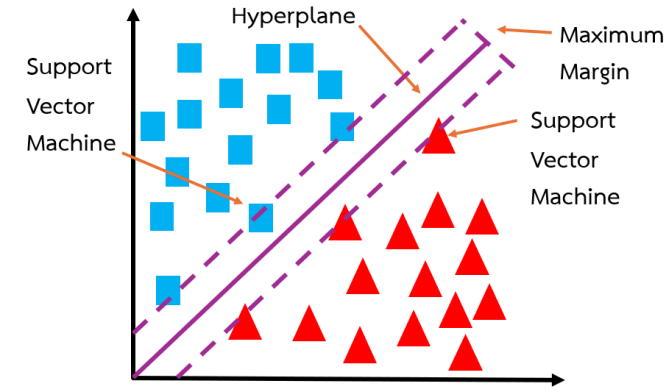
เมื่อ z เป็น สมการเส้นตรงคล้าย (Linear Regression)

w_i แทน ค่าน้ำหนัก (Weights) ที่ใช้ปรับค่าของตัวแปรอิสระ

x_i แทน ตัวแปรอินพุต (Feature Variables)

$\sigma(z)$ แทน เป็นค่าความน่าจะเป็น (อยู่ในช่วง 0 ถึง 1)

2.3.2 Support Vector Machine (SVM) เป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning Algorithm) ที่ใช้สำหรับการจำแนกประเภท (Classification) และและการทำนาย (Regression) โดยเฉพาะปัญหา Binary Classification ที่ต้องการจำแนกข้อมูลออกเป็น 2 กลุ่ม เช่น “ใช่” หรือ “ไม่ใช่” หลักการของ SVM การปรับข้อมูลใดไปอยู่ในมิติ (Dimension) ที่สูงขึ้นเพื่อการหาขอบเขตการแบ่งแยกข้อมูล (Decision Boundary) ที่ดีที่สุด ซึ่งเรียกว่า Hyperplane โดยใช้แนวคิดของ Margin ที่กว้างที่สุดระหว่างกลุ่มข้อมูล โดยพยายามหาขอบเขตที่แบ่งข้อมูลออกเป็น 2 คลาสให้ได้อย่างดีที่สุดโดยคำนึงถึงโดยให้มีระยะห่าง (Margin) คือค่าที่วัดจากระยะของ ซัพพอร์ตเวกเตอร์ไปยังไฮเปอร์เพลนที่กว้างที่สุด ยังเป็นวิธีที่ถือว่า robust ต่อข้อมูลที่มี noise พอสมควร ในกรณีที่ข้อมูลมีลักษณะไม่เป็นเชิงเส้น สามารถทำการปรับค่าของ Kernel Function เพื่อให้สามารถสร้างไฮเปอร์เพลนที่มีลักษณะไม่เป็นเส้นตรงได้ โดย Kernel Function ที่นิยมใช้มี ด้วยกัน 3 แบบคือ Polynomials , Radial Basis Function (RBF) และ Sigmoid (AI แบบเคลียร์, 2566)



ภาพที่ 2-3 เทคนิค Support Vector Machine (SVM)

2.3.3 Gradient Boosting เป็นเทคนิคการเรียนรู้ของเครื่องสำหรับแก้ปัญหาการถดถอย (Regression) และการจำแนกประเภท (Classification) เป็นอัลกอริธึมการเรียนรู้ของเครื่อง ๖ (Machine Learning) ที่ใช้แนวคิด Boosting ซึ่งเป็นการรวมโมเดลอ่อน ๆ (Weak Learners) หลายตัวเข้าด้วยกัน เพื่อให้ได้โมเดลที่แข็งแกร่ง (Strong Learner) โดยทั่วไปโมเดลอ่อน ๆ Weak Learner จะเป็น Decision Tree ขนาดเล็ก ใช้แนวคิดของ Gradient Descent ในการลดค่าความผิดพลาด (Residual Error) ข้อดีของ Gradient Boosting มีความแม่นยำสูงกว่าการใช้ Decision Tree เดี่ยว สามารถลด Overfitting ได้ดีกว่า Random Fores แต่ใช้เวลา Train นานกว่าตัวอื่น และอ่อนไหวต่อค่าพารามิเตอร์ ขั้นตอนหลักของ Gradient Boosting เริ่มต้นจาก โมเดลจะเริ่มต้นด้วยการทำนายค่าคงที่ (เช่น ค่าเฉลี่ยของตัวแปรที่ต้องการทำนาย) หรืออาจจะใช้วิธีการเบื้องต้นอื่นๆ ขึ้นอยู่กับลักษณะของปัญหา หลังจากที่โมเดลเริ่มทำการทำนายแล้ว จะคำนวณข้อผิดพลาด (หรือ residuals) โดยการหาความแตกต่างระหว่างค่าทำนายที่ได้กับค่าจริง โมเดลใหม่จะถูกสร้างขึ้นเพื่อทำนายข้อผิดพลาดที่เกิดขึ้นจากขั้นตอนก่อนหน้า โดยทั่วไปจะใช้ Decision Tree เป็นตัวสร้างโมเดลย่อย เนื่องจากมันสามารถจับความสัมพันธ์ที่ไม่เป็นเชิงเส้นได้ดี หลังจากนั้นโมเดลใหม่จะถูกเพิ่มเข้าไปในโมเดลที่มีอยู่เดิม โดยจะทำการปรับปรุงทำนายโดยการนำข้อผิดพลาดที่เกิดขึ้นมาทำนายใหม่ ค่าน้ำหนัก (weight) ที่ให้กับโมเดลใหม่จะถูกกำหนดโดยอัตราการเรียนรู้ (learning rate) ขั้นตอนนี้จะทำซ้ำหลายรอบ โดยที่แต่ละโมเดลใหม่จะช่วยปรับปรุงข้อผิดพลาดของโมเดลก่อนหน้า โมเดลทั้งหมดจะร่วมกันสร้างการทำนายที่มีความแม่นยำสูงขึ้น

2.3.3.1 หัวใจหลักของ Gradient Boosting ประกอบด้วย

2.3.3.1.1 Loss Function เป็นตัวชี้วัดข้อผิดพลาดระหว่างค่าที่โมเดลทำนายและค่าจริง ซึ่ง Gradient Boosting จะพยายามลดค่าของ Loss Function ในแต่ละรอบของการฝึก (training) โดยการเพิ่มโมเดลย่อยเข้ามาช่วยปรับปรุงผลลัพธ์ให้แม่นยำยิ่งขึ้น Gradient Boosting จะใช้ Gradient Descent ในการคำนวณค่า Gradient ของ Loss Function เพื่ออัปเดตโมเดลให้ดีขึ้นในแต่ละรอบ

2.3.3.1.2 Weak Learners เป็นโมเดลพื้นฐานที่ Gradient Boosting ใช้เพื่อเรียนรู้ข้อผิดพลาดหรือ Residuals ของโมเดลก่อนหน้า โดยใน Gradient Boosting มักใช้ Decision Tree ขนาดเล็กที่มีความลึกจำกัด (เช่น 1-3 ระดับ) เพื่อเป็น Weak Learners คุณสมบัติของ Weak Learner มีดังนี้

ก) เป็นโมเดลง่ายๆ ที่มีการเรียนรู้แบบ Greedy Algorithm เพื่อแบ่งข้อมูลตามคุณลักษณะ(features) ที่เหมาะสมที่สุด

ข) มีข้อได้เปรียบในการจัดการกับข้อมูลที่ไม่เป็นเชิงเส้น (Non-linear relationships)

ค) โมเดลต้นไม้ขนาดเล็กช่วยลดการ overfitting และเพิ่มความสามารถในการ generalize

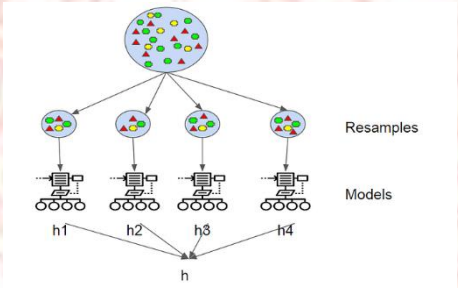
ในแต่ละรอบของ Gradient Boosting จะสร้าง Weak Learner ใหม่ขึ้นมาที่พยายามแก้ไขข้อผิดพลาดที่โมเดลก่อนหน้าทำไว้

2.3.3.1.2 Additive Mode คือกระบวนการรวมโมเดลย่อยที่สร้างขึ้นในแต่ละรอบเข้าด้วยกันอย่างต่อเนื่อง โดยผลลัพธ์ของโมเดลสุดท้ายจะเป็นการรวมผลลัพธ์จากโมเดลย่อยทั้งหมด การรวมโมเดลแบบ Additive นี้ช่วยให้ Gradient Boosting ค่อยๆ ปรับค่าทำนายให้เข้าใกล้ค่าจริงมากขึ้น โดยไม่ปรับเปลี่ยนแบบก้าวกระโดด

การทำงานร่วมกันของทั้ง 3 ส่วน Loss Function ใช้เพื่อคำนวณข้อผิดพลาดระหว่างค่าจริงกับค่าทำนาย และช่วยกำหนดทิศทาง (gradient) ของการอัปเดตในแต่ละรอ ส่วน Weak Learners ใช้ในการเรียนรู้ข้อผิดพลาด (residuals) จาก Loss Function ในแต่ละรอบ และ Additive Model รวมโมเดลย่อยทั้งหมดเข้าด้วยกันเพื่อปรับปรุงผลลัพธ์ในลักษณะเชิงเส้น opj ,2567)

2.3.4 Extreme Gradient Boosting (XGBoost) เป็นวิธีการเรียนรู้แบบกลุ่ม (Ensemble Learning) หรือการเรียนรู้แบบมีผู้เรียนหลายๆคน ช่วยๆกันเรียน (multiple-learners) โดยใช้เทคนิค

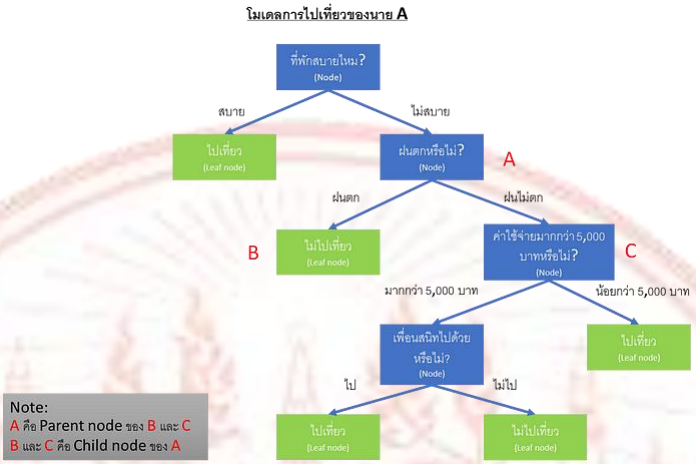
การนำการเรียนรู้ต้นไม้ตัดสินใจหลายๆ โมเดลมาต่อกัน ทำการสร้างและ เรียนรู้ โดยนำค่าผิดพลาดที่เกิดขึ้นจากโมเดลก่อนหน้ามาปรับปรุงในการสร้างโมเดลถัดไปเพื่อเพิ่ม ประสิทธิภาพ โมเดลจะถูกสร้างขึ้นจนกระทั่งไม่สามารถเพิ่มประสิทธิภาพของโมเดลได้อีก XGBoost เป็นอัลกอริทึม ที่พัฒนามาจาก Gradient Boosting โดยจะทำให้การเรียนรู้เร็วขึ้นและใช้ทรัพยากร ของเครื่องน้อยลง XGBoost สามารถทำงานกับข้อมูลจำนวนมากกว่าสิบล้าน เป็นอัลกอริทึมที่นิยมใช้ และได้ประสิทธิภาพสูงในปัจจุบัน โดยข้อดีคือสามารถลดการเกิด overfit ได้ สามารถจัดการกับ missing value แบบอัตโนมัติ สามารถทำงานแบบขนานกับจำนวน core ของ CPU ได้ และใช้ hardware อย่างมีประสิทธิภาพ เหมาะสำหรับงานที่ต้องการความแม่นยำสูง (Nut Chukamphaeng, 2561)



ภาพที่ 2-4 เทคนิค Extreme Gradient Boosting (XGBoost)

2.3.5 Decision Tree เป็นเป็นเทคนิคการนำข้อมูลมาสร้าง แบบจำลองการพยากรณ์ในรูปแบบโครงสร้างต้นไม้ ประเภทของ Decision Tree ประกอบด้วยใช้สำหรับจำแนกประเภท (Classification) และ ใช้สำหรับทำนายค่าต่อเนื่อง (Regression) เฉลี่ยค่าของผลลัพธ์จากทุกต้นไม้ โดย Decision Tree สำหรับจำแนกประเภทแทคนิคนี้สามารถสร้างแบบจำลองการจัดหมวดหมู่ได้จากกลุ่มตัวอย่างข้อมูลที่กำหนดไว้ล่วงหน้า และพยากรณ์กลุ่มของรายการที่ยังไม่เคยนำมาจัดหมวดหมู่ หลักการทำ Decision Tree คือ การแบ่งข้อมูลออกทีละ 2 ส่วน recursive binary split จาก node ล่างสุดของ tree เรียกว่า root node และไล่ขึ้นมาเรื่อยๆ จนถึง leaf node คือ node ตัวสุดท้ายที่ไม่สามารถแยกกลุ่มตัวอย่างได้อีกหรือเป็นจุดสิ้นสุดของการตัดสินใจและทำ prediction ค่า target variable ด้วยการใช้ค่า Gini Impurity, Entropy หรือ Information Gain ขึ้นอยู่ตามความเหมาะสมถ้าใช้ค่า Gini Impurity โดยหลักการจะเลือกตัวแปรที่มีค่า Gini Impurity น้อยที่สุด เพราะมันหมายถึงว่าตัวแปรนั้นทำให้ผลของการตัดใจเกิดความผิดพลาดน้อยที่สุด ถ้าแยกข้อมูลด้วย Entropy หรือ Information Gain จะขึ้นอยู่ตามค่าเงื่อนไขที่กำหนด (เช่น > หรือ ≤) ข้อดีของ Decision Tree เข้าใจง่ายและอธิบายได้ชัดเจน ไม่ต้องการทำ Feature Scaling และสามารถจัดการข้อมูลที่มี Missing Values ได้ แต่มีข้อเสียคือ Overfitting ได้ง่ายถ้าไม่มีการควบคุมความลึก

ของต้นไม้ อ่อนไหว (Sensitive) ต่อข้อมูลที่มี Noise อาจให้ผลลัพธ์ไม่ดีเมื่อมีข้อมูลที่ Overlapping (Pasith Thanapatpisarn, 2565)



ภาพที่ 2-5 เทคนิค Decision Tree แบบจำแนกประเภท (Classification)

2.1.5.1 สมการ Gini Impurity

$$Gini = 1 - \sum_{i=1}^{C} p_i^2 \qquad (2\text{-}4)$$

เมื่อ p_i แทน สัดส่วนของข้อมูลแต่ละคลาส

$Gini$ ต่ำแปลว่าข้อมูลในกลุ่มนั้นมีความบริสุทธิ์สูง

2.1.5.2 สมการ Entropy

$$Entropy = -\sum_{i=1}^{C} p_i \log_2 p_i \qquad (2\text{-}5)$$

เมื่อ p_i แทน สัดส่วนของข้อมูลแต่ละคลาส

$Entropy$ แทน ค่าวัดระดับของความไม่แน่นอนของข้อมูล

2.1.5.3 สมการ Information Gain

$$IG(S,A) = Entropy(S) - \sum_{i=1}^{C} \frac{|S_j|}{|S|} \times Entropy(S_j) \qquad (2\text{-}6)$$

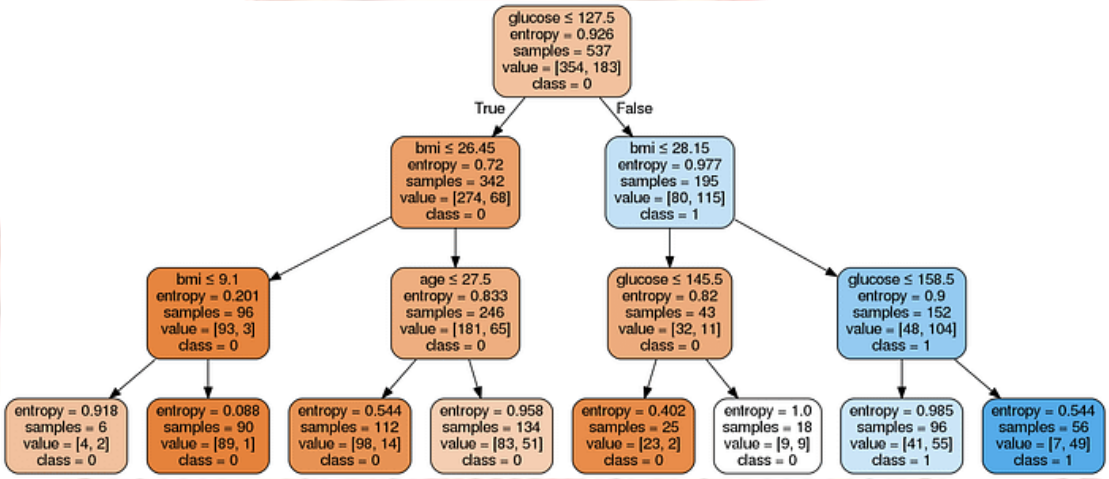
เมื่อ $\frac{|S_j|}{|S|}$ แทน คือสัดส่วนของตัวอย่างที่มีค่าของ $A = j$

S_j แทน เซตของตัวอย่างที่มีค่าของ $A = j$

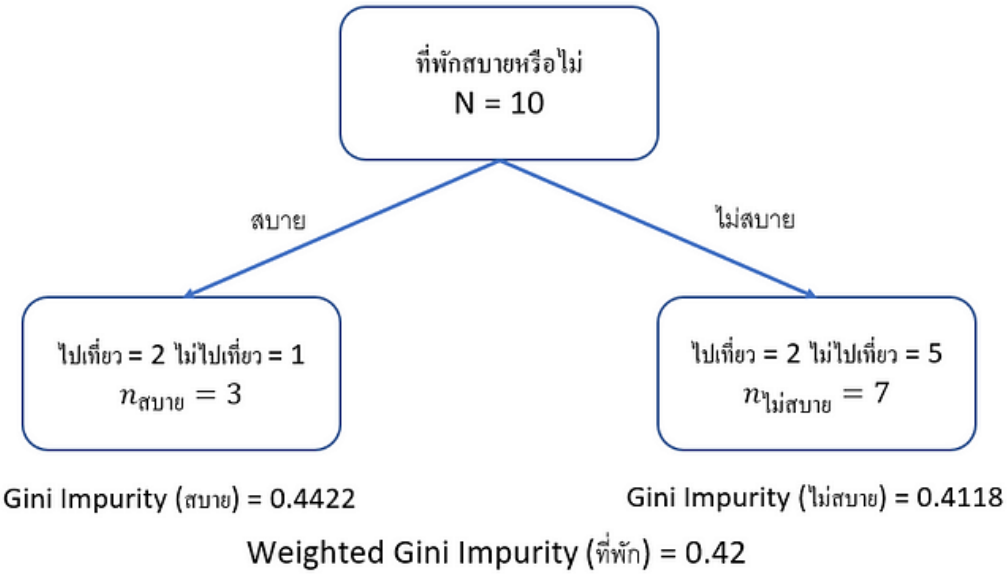
j แทน ค่าที่เป็นไปได้ของคุณสมบัติ A

$Entropy(S_j)$ แทน ค่าความไม่แน่นอนของเซต

$IG(S, A)$ ใช้ในการเลือกตัวแปรที่ดีที่สุด (ค่า Information Gain สูง แสดงว่าเป็นตัวแปรที่ดีในการแบ่งข้อมูล)



(ก) Entropy

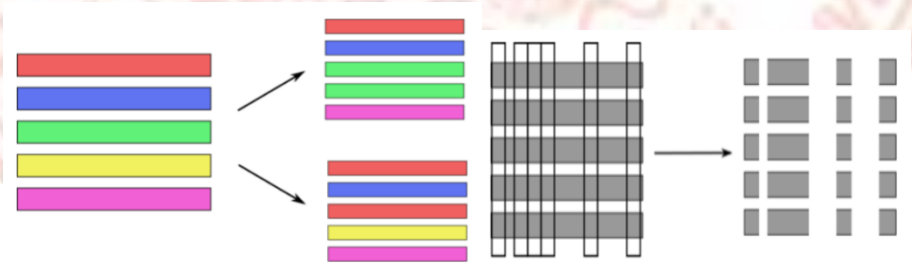


(ข) Gini Impurity

ภาพที่ 2-6 แบบจำแนกประเภท (Classification) ด้วย Entropy และ Gini Impurity

2.3.6 Random forest คือ การสร้างโมเดลจากโมเดลย่อยๆ หลาย decision tree โดยอาจตั้งแต่ 10 – 1000 โมเดล โดยที่แต่ละโมเดลย่อยนั้นจะได้รับข้อมูลที่ใช้ในการสอน (training data) ไม่เหมือนกัน หลังจากนั้นแต่ละโมเดลจำการ พยากรณ์ความแม่นยำของตัวเองออกมาด้วยการ vote output เพื่อหาคำตอบ

หลักการ Bagging จะมีการแบ่งข้อมูลออกเป็น decision tree หลายๆ ต้นแต่การทำ Bagging จะมีปัญหาเรื่องความไม่เป็นอิสระของข้อมูลเนื่องจากต่อให้เราแยกออกไปหลายๆ Tree ก็จริงแต่มันก็คือข้อมูลเดียวกัน Random Forest จึงเข้ามาแก้ปัญหาตรงนี้ โดยการทำ Random Sample Feature เป็นการแบ่งฟีเจอร์ (Feature) ของ decision tree แต่ละต้นจะมีฟีเจอร์ (Feature) ที่ไม่เหมือนกันทั้งหมด เพื่อทำให้แต่ละ Tree มีความหลากหลายและมีความอิสระกันมากขึ้น การรวมผลลัพธ์การจำแนกประเภท (Classification) ช้วิธีลงคะแนน (Voting) และ ใช้สำหรับทำนายค่าต่อเนื่อง (Regression) ใช้วิธีเฉลี่ย (Averaging) โดยการทำค่าสหสัมพันธ์ (Correlation) ระหว่าง decision tree แต่ละต้น ยิ่ง Correlation กันมากยิ่งแย่ลง สวนทางกับ decision tree ต้นเดียวยิ่งสูงยิ่งดี ข้อดีของ Random forest ลด Overfitting ได้ดี เพราะรวมผลลัพธ์จากหลายต้นไม้ มีความแม่นยำสูงโดยเฉพาะกับข้อมูลที่มีความซับซ้อนมีกการจัดการ Missing Values และ Outliers ได้ดีโดยที่ไม่ต้องทำ Feature Scaling และสามารถใช้กับข้อมูลที่มีฟีเจอร์ (Feature) จำนวนมากได้ แต่อาจใช้เวลาคำนวณสูงเมื่อมีต้นไม้จำนวนมาก ตีความได้ยากกว่าต้นไม้เดี่ยว และต้องมีการปรับ Hyperparameter ให้เหมาะสมเพื่อให้ได้ผลลัพธ์ที่ดี (PradyaSin, 2565)

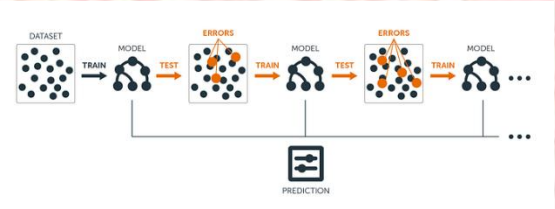


(ก) Bagging (ข) Random Sample Feature

ภาพที่ 2-7 เทคนิค Random Forest

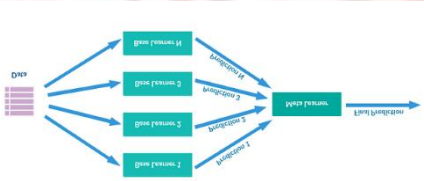
2.3.7 Ensemble คือเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ที่เป็นวิธีในการจำแนกประเภทของข้อมูลด้วยวิธีการรวมหลายโมเดลเข้าด้วยกันเพื่อให้ได้ผลลัพธ์ที่ดีกว่าโมเดลเดียว (Single Model) เพื่อ Overfitting และเพิ่มความแม่นยำ ทำให้โมเดลมีความเสถียรมากขึ้นโดยสามารถใช้ได้ทั้งจำแนกประเภท (Classification) และ ทำนายค่าต่อเนื่อง (Regression) (Pasith Thanapatpisarn, 2565) โดยบ่งออกเป็น 3 ประเภทหลัก

2.3.7.1 Bagging (Bootstrap Aggregating) เป็นวิธีการฝึกโมเดลหลายตัวด้วยข้อมูลที่สุ่มจากชุดข้อมูลเดิม (Bootstrap Sampling) แล้วใช้การเฉลี่ยผลลัพธ์ลดการเกิด Overfitting ทำให้โมเดลมีความเสถียร และใช้ได้ดีเมื่อโมเดลมีความผันผวนสูง เช่น Decision Tree หลักการทำงานของ Bagging จะทำการสุ่มตัวอย่างข้อมูล (Bootstrap Sampling) โดยการดึงข้อมูลแบบสุ่ม (Sampling with Replacement) ไป สร้างหลายโมเดลและจะสอน (Train) โมเดลด้วยชุดข้อมูลที่แตกต่างกัน อัลกอริธึมที่ใช้เช่น Random Forest



ภาพที่ 2-8 Bagging

2.3.7.2 Boosting เป็นเทคนิคที่ให้โมเดลเรียนรู้จากข้อผิดพลาดของโมเดลก่อนหน้าโดยใช้ใช้น้ำหนัก (Weight) กับตัวอย่างที่โมเดลเดิมคาดการณ์ผิด ทำให้โมเดลเรียนรู้จุดอ่อนของตัวเองช่วยเพิ่มความแม่นยำในการทำนาย หลักการทำงานของ Boosting จะสอน (Train) โมเดลแรก โมเดลแรกเรียนรู้จากข้อมูล โดยปรับน้ำหนักของตัวอย่างตัวอย่างที่ทำนายผิดจะได้รับน้ำหนักมากขึ้นและ จะสอน (Train) โมเดลถัดไป โดยใช้ข้อมูลที่ปรับน้ำหนักใหม่ รวมผลลัพธ์ของทุกโมเดล อัลกอริธึมที่นิยมใช้เช่น Gradient Boosting และ XGBoost

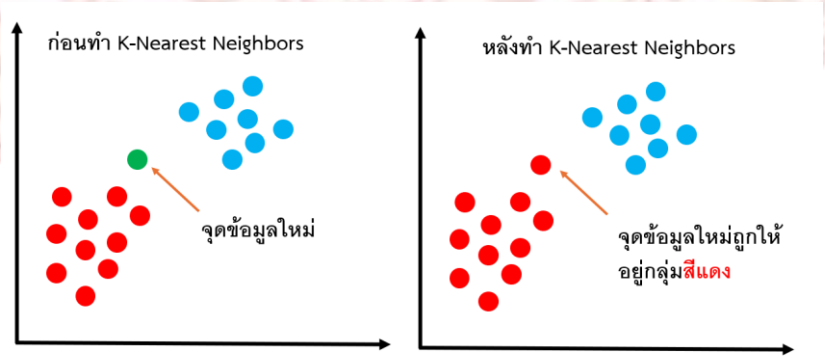


ภาพที่ 2-9 Boosting

2.3.7.3 Stacking คล้ายกับ Boosting แต่มีการแบ่งเป็นหลายๆ กลุ่ม เป็นเทคนิคที่รวมผลลัพธ์ของหลายโมเดลโดยใช้ Meta-Learner เป็นการใช้โมเดลหลายตัว (Base Models) ในการ

สอน (Train) โมเดล นำผลลัพธ์ของแต่ละโมเดลมาเป็น Feature ใหม่และใช้โมเดลอีกตัว (Meta Model) เพื่อรวมผลลัพธ์เพื่อ และทำนายครั้งสุดท้าย อัลกอริธึมที่นิยมใช้ เช่น Logistic Regression ,Random Forest, XGBoost ร่วมกัน

2.3.8 k-Nearest Neighbor (KNN) เป็นวิธีในการจำแนกประเภทของข้อมูลวิธีการหนึ่งที่เป็นที่นิยมในการจำแนกประเภทข้อมูล เพราะลักษณะพิเศษของเคเนียร์เรสเนเบอร์คือ Lazy-Learning ซึ่งเหตุที่ ถูกเรียกว่าเป็น Lazy-Learning นั้นมาจากรูปแบบในการจา แนกประเภทข้อ มูล ของเคเนียร์เรสเนเบอร์โดยไม่มีการสร้างโมเดลสำหรับจำแนกประเภทข้อมูลเตรียมไว้ล่วงหน้าเมื่อมี ข้อมูลใหม่ที่ต้องการจำแนกประเภท เพียงนำมาเทียบกับข้อมูลเดิมและดูถึงความคล้ายคลึงกันของ ข้อมูลใหม่และข้อมูลเดิมที่มีก็สามารถจำแนกประเภทของข้อมูลใหม่ได้โดยให้เป็นประเภท เดียวกับข้อมูลเดิมที่อยู่ใกล้เคียง การวัดความใกล้ชิดด้วยการวัดระยะทาง k-Nearest Neighbor มีการคำนวณระยะระหว่างจุดที่ต้องการแยกประเภทหรือ จัดกลุ่ม การพยากรณ์นั้นจะถูกกำหนดโดยการโหวตด้วยเสียงส่วนใหญ่ โดยส่วนใหญ่กำหนดให้เป็นเลขคี่ระยะห่างระหว่างการสังเกตของ i และ j สามารถวัดได้หลายทิศทาง ใช้ Euclidean distance, Manhattan distance หรือ Minkowski Distance ในการคำนวณหาระยะทางตามสมการ ข้อดีของ KNN หากเงื่อนไขในการตัดสินใจมีความซับซ้อนวิธีนี้สามารถนำไปสร้างโมเดลที่มีประสิทธิภาพได้ แต่ข้อเสียใช้เวลาคำนวณนาน ถ้า Attribute มีจำนวนมากจะเกิดข้อผิดพลาดในการคำนวณค่าและคำนวณค่าได้เฉพาะข้อมูลประเภท Nominal เช่น ข้อมูลเพศชาย-หญิง อาชีพ (Mr.P L, 2561)



ภาพที่ 2-10 เทคนิค k-Nearest Neighbor (KNN)

2.3.8.1 สมการ Euclidean distance

$$D(x_i, x_j) = \sqrt{\sum_{m=1}^{n}(x_i - x_j)^2} \quad (2\text{-}7)$$

เมื่อ n แทน จำนวนของ feature

x_i และ x_j แทน ค่าของตัวแปรแต่ละตัว

2.3.8.2 สมการ Manhattan distance

$$D(x_i, x_j) = \sum_{m=1}^{n}|x_i - x_j| \quad (2\text{-}8)$$

เมื่อ n แทน จำนวนของ feature

x_i และ x_j แทน ค่าของตัวแปรแต่ละตัว

2.3.8.3 สมการ Minkowski Distance

$$D(x_i, x_j) = \left(\sum_{m=1}^{n}|x_i - x_j|^p\right)^{\frac{1}{p}} \quad (2\text{-}9)$$

เมื่อ n แทน จำนวนของ feature

x_i และ x_j แทน ค่าของตัวแปรแต่ละตัว

p แทน พารามิเตอร์

2.3.9 Naïve Bayes เป็นเทคนิคที่สร้างโมเดลที่ใช้หลักการของ Bayes' Theorem โดยการคำนวณค่าความน่าจะเป็น (probability) ในการทำนายผล ต่างๆ ในข้อมูล ในการทำนายความน่าจะเป็นของคลาสในปัญหาการจำแนก (Classification) สามารถคาดการณ์ผลลัพธ์ได้จะทำการวิเคราะห์ความสัมพันธ์ระหว่างตัว แปรเพื่อใช้ในการสร้างเงื่อนไขความน่าจะเป็นสำหรับแต่ละความสัมพันธ์ เหมาะกับกรณีของเซตตัวอย่างที่มีจำนวนมาก ทำงานได้ดีแม้มีข้อมูลน้อย และคุณสมบัติ (Attribute) ของตัวอย่างไม่ขึ้นต่อกันซึ่งพบว่าประสิทธิภาพในการจำแนกไม้น้อยไปกว่าเทคนิคอื่นๆ แต่มีข้อเสียคือสมมติฐานว่าฟีเจอร์ (Feature) เป็นอิสระต่อกันไม่เป็นจริงเสมอ ไม่สามารถจับความสัมพันธ์ระหว่างฟีเจอร์ (Feature) ได้ดีและ เมื่อกล่าวถึงเหตุการณ์ที่เกิดขึ้น (A) คือคุณลักษณะ ถ้ามีเหตุการณ์หนึ่งเกิดขึ้นมาแล้ว (B) คือค่าคลาส สามารถเขียนให้อยู่ในรูปดังสมการ

2.3.8.1 สมการ Bayes' Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2\text{-}10)$$

เมื่อ $P(A|B)$ แทน ค่าความน่าจะเป็นที่ขอ้มูล Training Data ที่มีคลาส B และมีคุณลักษณะ A โดยที่ $A = a_1, a_2, \ldots a_M$ โดยที่ M คือ จำนวนคุณลักษณะใน Training Data หรือเป็นความน่าจะเป็นที่เหตุการณ์ A เกิดขึ้นเมื่อรู้ว่าเหตุการณ์ B เกิดขึ้น เรียกว่า Likelihood

$P(B|A)$ แทน ค่าความน่าจะเป็นที่ข้อ มูลที่มี คุณลักษณะเป็น A จะมีคลาส B หรือ เป็นความน่าจะเป็นที่เหตุการณ์ B เกิดขึ้นเมื่อรู้ว่าเหตุการณ์ A เกิดขึ้น เรียกว่า Posterior Probability

$P(A)$ แทน ความน่าจะเป็นล่วงหน้าของเหตุการณ์ A

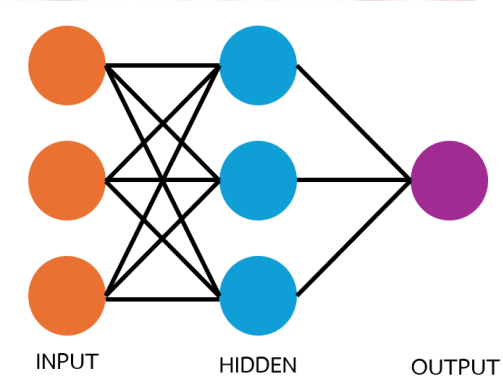
$P(B)$ แทน ค่าความน่าจะเป็นของคลาส B หรือเป็นความน่าจะเป็นล่วงหน้าของเหตุการณ์ B เรียกว่า PriorProbability

แต่การเกิดคุณลักษณะ $A = a_1, a_2, \ldots a_M$ ใน Training Data อาจจะมีจำนวนน้อยมาก หรืออาจไม่มีรูปแบบของคุณลักษณะแบบนี้เกิดขึ้นเลย ดังนั้นจึงได้ใช้หลักการที่ว่าแต่ละคุณลักษณะ เป็นอิสระต่อกันทำให้สามารถเปลี่ยนสมการ $P(A|B)$ ได้เป็น

$$P(A|B) = P(a_1|B) \times P(a_2|B) \times \ldots P(a_m|B) \quad (2\text{-}11)$$

ทำการคำนวณความน่าจะเป็นที่แต่ละคุณลักษณะจะเกิดในแต่ละคลาสแล้วนำค่าความน่าจะเป็นมาเปรียบเทียบกัน คลาสไหนที่ให้ค่าความน่าจะเป็นสูงกว่า แสดงว่าชุดข้อมูลจะมีโอกาสถูกจัดในคลาสนั้น แต่จากหลักการทำ งานของ Naïve Bayes จะพบว่าเมื่อไม่มีรูปแบบของคุณลักษณะเกิดขึ้นใน Training Data จะทา ให้ค่าความน่าจะเป็นที่คำนวณของคุณลักษณะนั้น เป็น 0 ซึงจะทำให้ค่าที่ทำนายมี ค่าเป็น 0 ไปด้วยการแก้ไขปัญหานี้จึงมีการเพิ่มค่าขมูลของคุณลักษณะที่หายไปนั้นด้วย 1 (Peachapong Poolpol, 2564)

2.3.10 Neural Network (NN) หรือโครงข่ายประสาทเทียม เป็นเทคโนโลยีที่มีที่มาจากงานวิจัยด้านปัญญาประดิษฐ์ (Artificial Intelligence : AI) เพื่อใช้ ในการคำนวณคำาฟังก์ชันจากกลุ่มข้อมูล วิธีการที่ให้โมเดลู้ตัวอย่างต้นแบบ แล้วฝึก (Train) ให้ระบบได้รู้จักที่จะคิดแก้ปัญหาที่กว้างขึ้นได้ในรูปแบบในโครงสร้าง เครือข่ายใยประสาทจะประกอบด้วยโหนด (Node) สำหรับ Input Output และการประมวลผล กระจายอยู่ในโครงสร้างเป็นชั้น ๆ ได้แก่ Input layer , Output layer และ Hidden layers การประมวลผล ของเครือข่ายใยประสาทจะอาศัยการส่งการทำงานผ่านโหนดต่าง ๆ ใน layer เหล่านี้ Neuron แต่ละตัวในเครือข่ายมี น้ำหนัก (Weight) และค่าลำเอียง (Bias) ค่าข้อมูลนำเข้า X จะถูกคูณด้วยน้ำหนัก W บวกด้วยค่า Bias ผ่านฟังก์ชันกระตุ้น (Activation Function) เพื่อสร้างเอาต์พุต แนวคิดของเครือข่ายใยประสาทได้มาจากการจำลองการทำงานของเซลล์สมองของมนุษย์ โดยหน่วยที่ย่อยที่สุดของเครือข่ายใยประสาท เรียกว่า เพอร์เซ็ปตรอน (Perceptron) ซึ่งเทียบได้กับ เซลล์สมองของมนุษย์หนึ่งนิวรอน (Neuron) เพอร์เซ็ปตรอนนี้ จะทำหน่าที่รับอินพุตซึ่งเป็นเวคเตอร์ ของจำนวนจริงเข้ามา พร้อมคำนวณค่าเหล่านี้โดยให้น้ำหนักของอินพุตแต่ละตัวแตกต่างกัน เอาต์พุตที่ได้จะถูกนำไปคำนวณคำผิดพลาด (Error) เพื่อนำมาน้ำหนักของอินพุตต่อไป ข้อดีของ Neural Network สามารถเรียนรู้รูปแบบที่ซับซ้อนได้ ใช้ได้ทั้งภาพ เสียง และข้อความ สามารถรองรับปริมาณข้อมูลขนาดใหญ่ แต่ข้อเสียคือต้องใช้พลังงานในการคำนวณสูง ต้องมีข้อมูลจำนวนมากเพื่อให้แม่นยำ และ อาจเกิด Overfitting ถ้าไม่มีการปรับแต่งพารามิเตอร์ที่ดี (Kasidis Satangmongkol, 2565)



ภาพที่ 2-11 เทคนิค Neural Network (NN)

2.3.10.1 สมการคำนวณของ Neuron

$$y = f(\sum W_i X_i + b) \quad (2\text{-}12)$$

เมื่อ Weight (W_i) แทน น้ำหนัก (weight) ของอินพุตที่ i ค่าควบคุมความสำคัญของ Feature

X_i แทน อินพุตที่ i

Bias (b) แทน ออฟเซ็ตหรือไบแอส (bias) เป็นการปรับค่าเพื่อให้ฟังก์ชันสามารถเลื่อนตำแหน่งได้

2.3.10.2 Activation Functions การทำให้เครือข่ายสามารถเรียนรู้รูปแบบที่ซับซ้อนได้ โดยทำหน้าที่แปลงค่าให้เหมาะสมก่อนส่งไปยังชั้นถัดไปกล่าวคือ Neuron ใช้ activation functions เพื่อบีบตัว output ให้อยู่ในช่วงที่ต้องการ เช่น [0, 1], [-1, +1] เป็นต้น การใช้ weighted sum ร่วมกับ activation functions ช่วยให้ neuron สามารถเรียนรู้ความสัมพันธ์แบบ non-linear ได้ดียิ่งขึ้น ถ้าเราไม่มี activation functions โมเดลของเราจะเรียนรู้ได้แค่ linear relationship เท่านั้น Activation functions ที่เรานิยมใช้ใน neural networks ได้แก่

2.3.10.2.1 Sigmoid Function หรืออีกชื่อคือ Logistic Function ให้ออกมาเป็นค่า 0 ถึง 1

$$f(x) = \frac{1}{1+e^{-x}} \quad (2\text{-}13)$$

2.3.10.2.2 ReLU (Rectified Linear Unit) Function (นิยมใช้ใน Hidden Layers) ทำให้โมเดลเรียนรู้ได้เร็วขึ้น

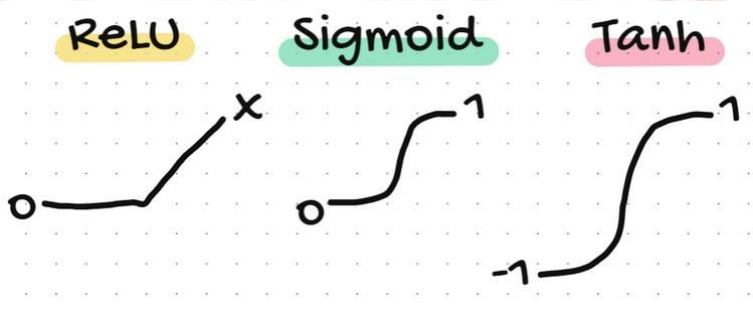
$$f(x) = \max(0, x) \quad (2\text{-}14)$$

เมื่อ ถ้า $x > 0$ ให้ค่าเดิม

ถ้า $x \leq 0$ ให้ค่า 0

2.3.10.2.1 Tanh (Hyperbolic Tangent Function) จะเปลี่ยนค่าอินพุต ให้แสดงผลค่าเอาต์พุตระหว่าง –1 ถึง 1

$$f(x) = \tanh(x) \tag{2-15}$$



ภาพที่ 2-12 Activation Functions

2.4 การแบ่งข้อมูลเพื่อทำการทดสอบประสิทธิภาพของแบบจำลอง

2.4.1 ความหมายของการแบ่งข้อมูล

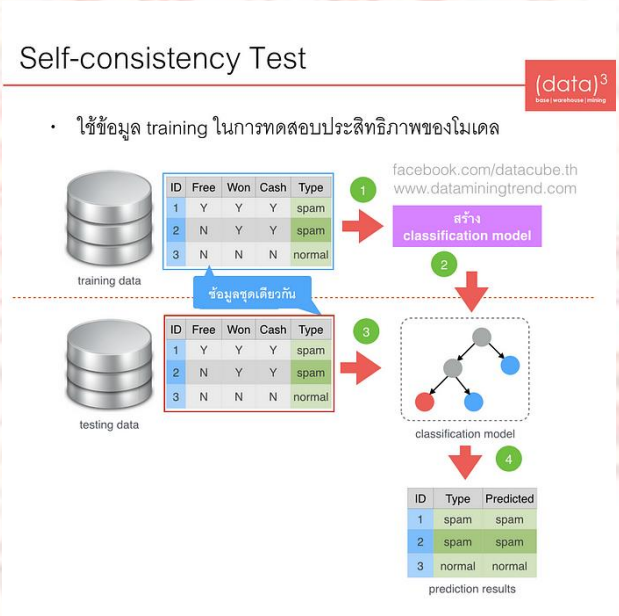
การแบ่งข้อมูล (Data Splitting) เป็นกระบวนการสำคัญในการพัฒนาและประเมินประสิทธิภาพของแบบจำลอง Machine Learning โดยช่วยให้สามารถวัดความสามารถของโมเดลในการเรียนรู้จากข้อมูลเดิมและทำนายข้อมูลใหม่ได้อย่างแม่นยำ เพื่อให้แน่ใจว่าโมเดลที่สร้างขึ้นสามารถทำงานได้ดีทั้งกับข้อมูลที่ใช้ในการฝึก (Training Data) และข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน (Unseen Data) การแบ่งข้อมูลช่วยให้สามารถวัดค่าความแม่นยำ (Accuracy) หรือค่าทางสถิติอื่น ๆ ของแบบจำลองบนชุดข้อมูลที่ไม่เคยถูกใช้ในการฝึก ใช้สำหรับเปรียบเทียบประสิทธิภาพของโมเดลหลายตัวเพื่อเลือกโมเดลที่ดีที่สุด ทำให้ได้โมเดลที่มีประสิทธิภาพและสามารถนำไปใช้งานจริงได้ การแบ่งข้อมูลยังช่วยให้สามารถใช้ข้อมูลส่วนหนึ่งในการฝึกโมเดล และอีกส่วนหนึ่งเพื่อปรับแต่งพารามิเตอร์ให้เหมาะสม (Surapong Kanoktipsatharporn, 2562) และการแบ่งข้อมูลที่ดีช่วยลดปัญหา Overfitting และ Underfitting

2.4.1.1 Overfitting เกิดขึ้นเมื่อโมเดลเรียนรู้รายละเอียดเฉพาะของข้อมูลฝึกมากเกินไป ทำให้ไม่สามารถทำนายข้อมูลใหม่ได้ดี

2.4.1.2 Underfitting เกิดขึ้นเมื่อโมเดลไม่ได้เรียนรู้จากข้อมูลมากพอ ทำให้ไม่สามารถจับความสัมพันธ์ของข้อมูลได้ดี

2.4.2 ประเภทของการแบ่งข้อมูล สามารถแบ่งประเภทของการแบ่งข้อมูล (Panupong Suksuwan, 2562) ได้ดังนี้

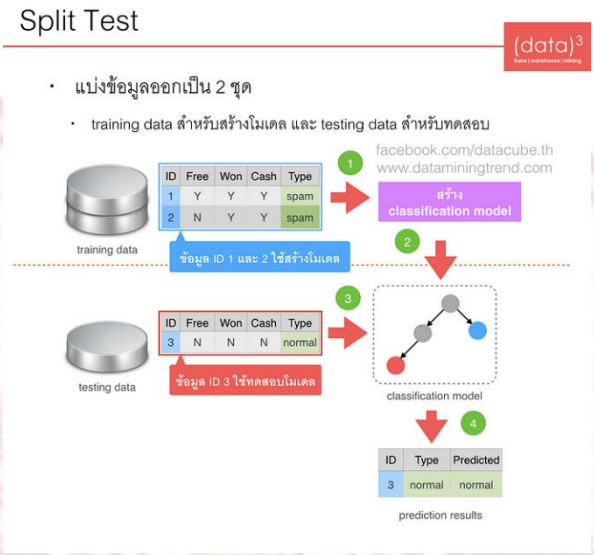
2.4.2.1 Self Consistency เป็นวิธีของการใช้ข้อมูลในการสร้างแบบจำลองกับข้อมูลในการทดสอบ เป็นชุดเดียวกัน โดยที่ประสิทธิภาพจากการที่วัดออกมาได้จะมีค่าค่อนข้างสูง อันเนื่องมาจากมีการใช้ ข้อมูลชุดเดียวกันในการสร้างแบบจำลองและใช้ในการวัดประสิทธิภาพ ดังนั้นวิธีนี้จึงเหมาะกับใช้ใน การทดสอบประสิทธิภาพเพื่อดูแนวโน้มของโมเดลที่สร้างขึ้น ถ้าได้ผลการวัดที่น้อย แสดงว่า แบบจำลองที่สร้างขึ้นมาไม่มีความเหมาะสมกับข้อมูล จึงยิ่งไม่ควรนำไปทดสอบกับชุดข้อมูลอื่น



ภาพที่ 2-13 ตัวอย่างการแบ่งข้อมูลแบบ Self-consistency Test

2.4.2.2 Split Test หรือ Hold-out Validation เป็นการแบ่งข้อมูลการแบ่งข้อมูลแบบคงที่ด้วยการสุ่มออกเป็น 2 ส่วน หรือ 3 ส่วน คือ Training Set (60-80%) ใช้สำหรับการสร้างหรือฝึกแบบจำลองเพื่อทดสอบความแม่นยำของแบบจำลองที่สร้างขึ้นมา Validation Set (10-20%) ใช้สำหรับปรับค่าพารามิเตอร์และเลือกแบบจำลองที่ดีที่สุด หรือ Test Set (10-30%) ใช้สำหรับประเมินประสิทธิภาพของแบบจำลองหลังจากการปรับแต่งแล้ว โดยการทดสอบแบบ Split Test นี้ทำการสุ่มข้อมูลเพียงครั้งเดียว ซึ่งถ้า หากครั้งที่ทำการสุ่มข้อมูลที่ใช้ในการทดสอบที่มีลักษณะคล้ายกับข้อมูลที่ใช้สร้างแบบจำลองก็จะทำ ให้ผลการวัดประสิทธิภาพได้ออกมาดีในทางกลับกันถ้าหากการสุ่มข้อมูลที่

ใช้ในการทดสอบที่มี ลักษณะแตกต่างกับข้อมูลที่ใช้สร้างแบบจำลองมากก็จะทำให้ผลการวัดประสิทธิภาพได้ออกมาไม่ดี ดังนั้นหากต้องการใช้วิธี Split Test นี้จึงควรทำการสุ่มหลายๆ ครั้ง

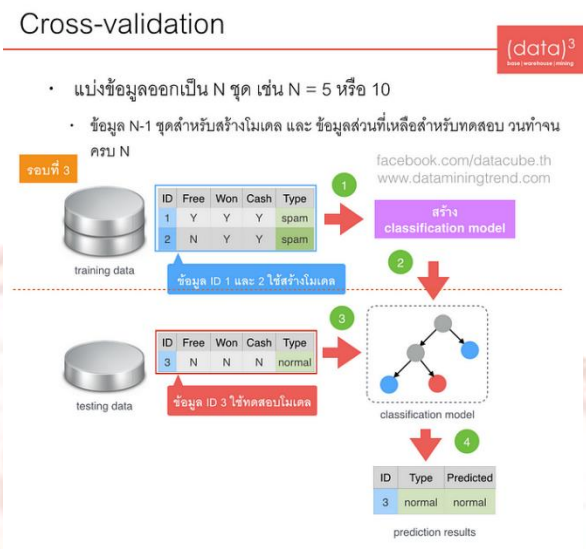


ภาพที่ 2-14 ตัวอย่างการแบ่งข้อมูลแบบ Split Test

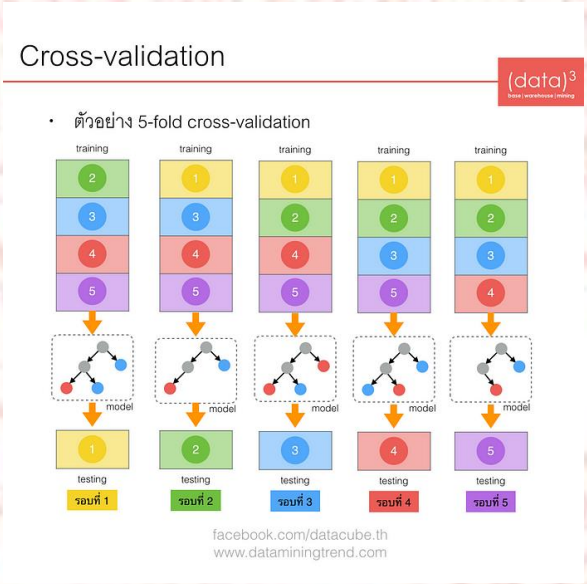
2.4.2.3 Cross Validation เป็นการแบ่งข้อมูลออกเป็นหลายส่วน โดยสามารถกำหนดได้ว่าต้องการแบ่งข้อมูลออกเป็นกี่ส่วน (k) เท่าๆกัน โดยแต่ละครั้งจะใช้ (k-1) ส่วนเป็นข้อมูลฝึก และอีก 1 ส่วนเป็นข้อมูลทดสอบ วิธีนี้ช่วยลดอคติจากการแบ่งข้อมูลเพียงครั้งเดียว และให้ผลลัพธ์ที่แม่นยำกว่า ข้อมูลหนึ่งส่วนจะถูกใช้ เป็นตัวทดสอบประสิทธิภาพของแบบจำลอง ซึ่งจะมีการทำวนไปเช่นนี้จนครบจำนวนที่แบ่งไว้โดยวิธี นี้เป็นวิธีที่หมาะและนิยมใช้ในการทำวิจัยเนื่องจากผลที่ได้มีความน่าเชื่อถือมากที่สุดใน 3 วิธีของการ แบ่งข้อมูลเพื่อนำทดสอบประสิทธิภาพของแบบจำลอง
ตัวอย่างค่า k ที่นิยมใช้มีดังนี

2.4.2.3.1 k = 5 หรือ k = 10 เป็นค่าที่นิยมมาก เนื่องจากให้สมดุลที่ดีระหว่างความแม่นยำและเวลาในการคำนวณ

2.4.2.3.2 k = Leave-One-Out (LOO) คือกรณีที่ k มีค่าเท่ากับจำนวนตัวอย่างทั้งหมด ซึ่งทำให้แบบจำลองถูกทดสอบด้วยข้อมูลทุกจุด แต่วิธีนี้อาจใช้เวลาประมวลผลสูง



(ก) การแบ่งชุดข้อมูล



(ก) 5-fold Cross Validation

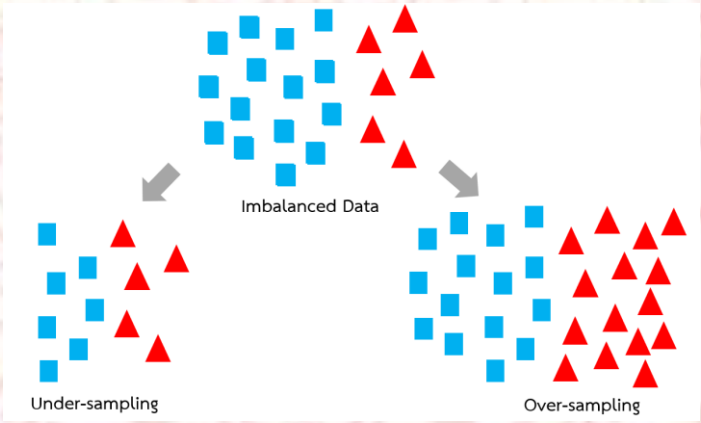
ภาพที่ 2-15 ตัวอย่างการแบ่งข้อมูลแบบ K-Cross-validation Test

2.5 การจัดการข้อมูลที่ไม่สมดุล (Imbalanced Data)

2.5.1 ความหมายข้อมูลที่ไม่สมดุล (Imbalanced Data) เป็นปัญหาที่พบได้บ่อยในงานด้าน Machine Learning โดยเฉพาะปัญหาการจำแนกประเภท (Classification) ซึ่งเกิดขึ้นเมื่อจำนวนตัวอย่างในแต่ละคลาสไม่เท่ากัน หรือมีอัตราส่วนระหว่างคลาสที่แตกต่างกันมาก ผลกระทบจากข้อมูลที่ไม่สมดุลโมเดลที่พัฒนามีแนวโน้มที่จะลำเอียงไปทางคลาสที่มีจำนวนมากกว่า ส่งผลให้ค่าความแม่นยำ (Accuracy) ไม่สามารถบอกได้ว่าโมเดลทำงานได้ดีหรือไม่ เนื่องจากโมเดลอาจละเลยคลาสที่มีจำนวนน้อย ทำให้ Recall ต่ำ การแก้ไขข้อมูลที่ไม่สมดุลสามารถทำได้โดยการเปลี่ยนแปลงอัตราส่วนของคลาสในชุดข้อมูลผ่าน (Rukchanok Piyapanichayakul, 2566) สองวิธีหลัก

2.5.1.1 Under-sampling ลดจำนวนตัวอย่างของคลาสที่มีมากเกินไป

2.5.1.2 Over-sampling ลดจำนวนตัวอย่างของคลาสที่มีมากเกินไป เหมาะกับกรณีที่มีข้อมูลจำกัดโดยไม่ทำให้ข้อมูลหายไป แต่ข้อเสียอาจเกิดปัญหา Overfitting ได้ เนื่องจากสร้างข้อมูลที่คล้ายกับตัวอย่างเดิมมากเกินไป



ภาพที่ 2-16 การจัดการข้อมูลที่ไม่สมดุล

2.5.2 SMOTE (Synthetic Minority Over-sampling Technique) เป็นเทคนิค 2 Over-sampling ที่ได้รับความนิยมมากที่สุด ในการแก้ปัญหาข้อมูลที่ไม่สมดุล (Imbalanced Data) ใช้หลักการสร้างตัวอย่างข้อมูลของคลาสที่มีจำนวนน้อย (Minority Class) ขึ้นใหม่โดยการคำนวณค่าระหว่างตัวอย่างที่มีอยู่แทนการคัดลอกตัวอย่างเก่า เป็นการสังเคราะห์ตัวอย่าง (Synthetic Data) เพื่อเพิ่มข้อมูลในชนกลุ่มน้อยให้มีจำนวนใกล้เคียงหรือเท่ากับ จำนวนข้อมูลที่มีอยู่ในชนกลุ่มมาก ทำให้ข้อมูลที่เพิ่มขึ้นมีความหลากหลายและช่วยลดปัญหา Overfitting แทนที่จะสุ่มคัดลอกตัวอย่างจากคลาสที่มีน้อย SMOTE สร้างตัวอย่างใหม่โดยใช้การคำนวณทางสถิติเพื่อสร้างจุดข้อมูลใหม่ระหว่างตัวอย่างที่มีอยู่จริง (Cory Maklin, 2565)

2.5.3 หลักการทำงานของ SMOTE มีดังนี้

2.5.3.1 เลือกตัวอย่างจากคลาสที่มีจำนวนน้อย

2.5.3.2 ใช้ k-Nearest Neighbors (KNN) เพื่อค้นหาตัวอย่างที่ใกล้เคียง

2.5.3.3 สุ่มเลือกหนึ่งใน k-Neighbors

2.5.3.4 สร้างตัวอย่างใหม่โดยใช้การคำนวณระหว่างจุดข้อมูลต้นฉบับกับเพื่อนบ้านที่เลือกได้

2.5.4 สูตรการสร้างข้อมูลใหม่ SMOTE ใหม่

$$X_{new} = X_{original} + \lambda\left(X_{neighbor} - X_{original}\right) \quad (2\text{-}15)$$

เมื่อ λ แทน ค่าที่สุ่มจากช่วง (0,1)

$X_{original}$ แทน ตัวอย่างที่มีอยู่

$X_{neighbor}$ แทน ตัวอย่างที่ถูกเลือกจากเพื่อนบ้าน

2.6 การคัดเลือกคุณสมบัติ

การคัดเลือกคุณลักษณะ (Feature selection) คือ การคัดเลือกคุณสมบัติที่ช่วย ในการลดจำนวนตัวแปรที่จะใช้ในการพยากรณ์ข้อมูลในแบบประเภทข้อมูลการจำแนกข้อมูล (classification) เพื่อให้จำนวนตัวแปรนั้นมีจำนวนให้น้อยที่สุดและไม่ส่งผลต่อประสิทธิภาพของความแม่นยำของแบบจำลองในการพยากรณ์ ทำให้โมเดลเข้าใจง่ายขึ้นและมีการทำงานที่รวดเร็วขึ้นเนื่องจากช่วยให้โมเดลเรียนรู้เฉพาะ Features ที่มีความสำคัญลงทำให้ลดเวลาประมวลผลโดยลดจำนวน ฟีเจอร์ (Feature) ต้องใช้ โดยขั้นตอนนี้เรียกว่าการคัดเลือกคุณลักษณะ หรือ เลือกฟีเจอร์ (feature selection) (Pimporn Pimpim, 2562) สามารถแบ่งได้เป็น 3 กลุ่มใหญ่ดังนี

2.6.1 Filter approach คือ การคัดเลือกคุณลักษณะด้วยการใช้วิธีคำนวณหาค่าน้ำหนักซึ่งอาจจะเป็นค่าความสัมพันธ์ระหว่างแต่ละฟีเจอร์ (Feature) กับผลของการจำแนกประเภทของข้อมูล (Target) โดยไม่ต้องฝึกสอน (Train) โมเดล โดยที่ค่าน้ำหนักที่คำนวณได้แล้วเลือกคุณลักษณะที่มีค่าน้ำหนักมากกว่าที่ต้องการมาใช้งานต่อไป ซึ่งวิธีนี้เร็วและใช้กับข้อมูลได้ทุกประเภท อาจไม่สามารถระบุ ฟีเจอร์ (Features) ที่มีความสัมพันธ์เชิงซ้อน เทคนิคในการคำนวณค่าน้ำหนักของคุณลักษณะต่างๆ มีหลายวิธี ได้แก่ Information Gain, ChiSquare หรือ Correlation

2.6.2 Wrapper approach คือ เป็นวิธีการคัดเลือกคุณลักษณะด้วยการสร้าง โมเดล (classification model) ขึ้นมาจากกลุ่มของคุณลักษณะที่ได้มีการกำหนดไว้และทำการวัดประสิทธิภาพการทำงานของโมเดล พร้อมทั้งทำการเลือกกลุ่มของคุณลักษณะที่ทำให้โมเดลมี

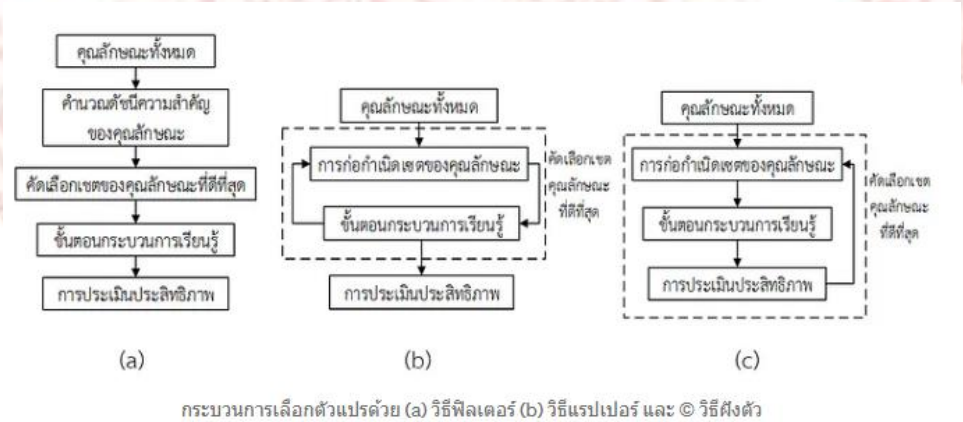
ประสิทธิภาพมากที่สุดมาใช้งาน ซึ่งวิธีนี้สามารถหาชุดฟีเจอร์ (Features) ที่เหมาะสมที่สุดได้ แต่ใช้เวลามาก เพราะต้องฝึกสอน (Train) โมเดลหลายรอบ ได้แก่ โมเดลที่ให้ค่าความถูกต้อง (accuracy) มากที่สุด การคัดเลือกคุณลักษณะด้วยวิธีนี้สามารถแบ่งได้เป็น 3 แบบใหญ่ๆ คือ

2.6.2.1 Forward Selection คือ การสร้างโมเดลโดยการเพิ่มคุณลักษณะ ของข้อมูลเข้าไปทีละ 1 คุณลักษณะ โดยที่คุณลักษณะที่มีการใส่เข้าไปนั้นจะต้องทำให้ประสิทธิภาพ ของโมเดลที่ดีขึ้น และจะทำการเพิ่มคุณลักษณะอื่นๆ เข้ามาใส่เพิ่มต่อไปเรื่อยๆจนประสิทธิภาพของ โมเดลที่ได้นั้นไม่สูงกว่าหรือเท่าเดิมจากประสิทธิภาพสูงสุดที่ทำได้ จึงจะหยุดการเพิ่มคุณลักษณะลงใน โมเดล

2.6.2.2 Backward Elimination คือ การสร้างโมเดลโดยเริ่มจากคุณลักษณะ ของข้อมูลทั้งหมดที่มีนำมาหาประสิทธิภาพของโมเดล และหลังจากนั้นจะทำการตัดคุณลักษณะของข้อมูลออกไปทีละคุณลักษณะ โดยที่ถ้าประสิทธิภาพของโมเดลดีขึ้น จะทำการตัดคุณลักษณะอื่นๆ ต่อไปเรื่อยๆจนกว่าประสิทธิภาพของโมเดลที่ได้ ไม่สูงกว่าเท่าเดิมจากประสิทธิภาพสูงสุดที่ทำได้ จึงหยุดการตัดคุณลักษณะออกจากโมเดล

2.6.2.3 Recursive Feature Elimination (RFE) คือ ใช้แบบจำลองในการคัดเลือกคุณสมบัติหรือฟีเจอร์ (Features) ที่สำคัญที่สุด แล้วตัดคุณลักษณะของที่ไม่สำคัญออก

2.6.3 Embedded Methods คือ ใช้แบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) โดยตรง ในการเลือกคุณสมบัติหรือฟีเจอร์ (Features)ขณะฝึกโมเดล



กระบวนการเลือกตัวแปรด้วย (a) วิธีฟิลเตอร์ (b) วิธีแรปเปอร์ และ © วิธีฝังตัว

รูปที่ 2-17 การเลือกคุณลักษณะ

2.7 การประเมินผลแบบจำลอง (Evaluation Model)

การประเมินผลแบบจำลอง (Model Evaluation) เป็นกระบวนการวัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) เพื่อวัดประสิทธิภาพและความแม่นยำของโมเดลที่ได้สร้างขึ้นโดยการประเมินผลจะช่วยให้เราทราบว่าโมเดลทำงานได้ดีหรือไม่ในการทำนายข้อมูลใหม่ หรือข้อมูลที่ยังไม่เคยเห็นมาก่อน ทำหน้าที่เป็นตัวชี้วัดทางสถิติและวิธีการวัดผลที่หลากหลาย เพื่อให้แน่ใจว่าแบบจำลองสามารถทำงานได้ดีทั้งกับข้อมูลที่ใช้ในการฝึก (Training Data) และข้อมูลใหม่ที่แบบจำลองไม่เคยเห็นมาก่อน (Testing Data) การประเมินผลแบบจำลองมีความสำคัญเพราะช่วยให้สามารถเลือกแบบจำลองที่ดีที่สุด ลดปัญหา Overfitting และ Underfitting และทำให้สามารถตัดสินใจได้ว่าโมเดลที่สร้างขึ้นพร้อมใช้งานจริงหรือไม่ ตัววัดประสิทธิภาพที่นิยมใช้ในการวัดประสิทธิภาพของตัวแบบพยากรณ์การจำแนกประเภท (classification) มีอยู่ 5 ค่า (Mr.P L, 2561). ดังนี้

2.7.1 ค่าความไว (Recall) คือ ค่าความถูกต้องของแบบจำลอง (Model) โดยพิจารณาแยกทีละคลาส (Class) เป็นค่าที่ใช้วัดความสามารถของแบบจำลอง ในการพยากรณ์การเกิดขึ้นจริงว่าเป็นจริงหรือเป็นการวัดว่าในกลุ่มที่แบบจำลองทำนายว่าเป็น "Positive" มีเท่าไหร่ที่ถูกต้องจริง เช่น มีผู้ที่อนุมัติบัตรเครดิตผ่านจริง 100 คน แต่ ป่วยจริง 100 คน โมเดลทำนายว่าผ่าน 80 (TP = 80) ทำนายพลาดไป 20 (FN = 20) คน ค่าความไว (Recall) คำนวณได้ว่า 80% หมายถึง โมเดลสามารถตรวจจับที่อนุมัติบัตรเครดิตผ่านเพียง 80% แต่ยังมีอีก 20% ที่พลาดไป เป็นการใช้เปอร์เซ็น ของว่าจริงโมเดลตรวจจับ เทียบกับ จำนวนของว่าจริงทั้งหมด ถ้า Recall สูงหมายความว่าโมเดลสามารถระบุข้อมูลบวกได้มาก

สูตรการคำนวณ

$$Recall = \frac{TP}{TP+FN} \quad (2\text{-}16)$$

เมื่อ TP (True Positive) แทน จำนวนที่โมเดลทำนายว่า "ใช่" และเป็น "ใช่" จริง

FN (False Negative) แทน จำนวนที่โมเดลทำนายว่า "ไม่ใช่" แต่จริงๆ เป็น "ใช่" (ทำนายผิด)

2.7.2 ค่าความถูกต้อง (Accuracy) คือ ค่าความถูกต้องและความแม่นยำของแบบจำลองกี่เปอร์เซ็นต์ โดยพิจารณารวมทุกผลลัพธ์ที่เกิดขึ้นทั้งหมด จากข้อมูลทั้งหมดที่ใช้ทดสอบ เพื่อดูว่ามีความแม่นยำในการทำนายผลถูกต้องมากหรือน้อยเพียงใด

สูตรการคำนวณ

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2\text{-}17)$$

เมื่อ TP (True Positive) แทน จำนวนที่โมเดลทำนายว่า "ใช่" และเป็น "ใช่" จริง

TN (True Negative) แทน จำนวนที่โมเดลทำนายว่า "ไม่ใช่" และเป็น "ไม่ใช่" จริง

FP (False Positive) แทน จำนวนที่โมเดลทำนายว่า "ใช่" แต่จริงๆ เป็น "ไม่ใช่" (ทำนายผิด)

FN (False Negative) แทน จำนวนที่โมเดลทำนายว่า "ไม่ใช่" แต่จริงๆ เป็น "ใช่" (ทำนายผิด)

2.7.3 ค่าความแม่นยำ (Precision) เป็นการวัดความแม่นยำของข้อมูลเมื่อมันทำนายว่าข้อมูลเป็นบวก (Positive) โดยพิจารณาแยกทีละคลาส เป็นค่าที่บอกว่าในบรรดาข้อมูลที่โมเดลทำนายว่า "ใช่" หรือ "เป็นจริง" มี กี่เปอร์เซ็นต์ที่ทำนายถูกต้องจริงๆ

สูตรการคำนวณ

$$Recall = \frac{TP}{TP+FP} \quad (2\text{-}18)$$

โดยที่

เมื่อ TP (True Positive) แทน จำนวนที่โมเดลทำนายว่า "ใช่" และเป็น "ใช่" จริง

FP (False Positive) แทน จำนวนที่โมเดลทำนายว่า "ใช่" แต่จริงๆ เป็น "ไม่ใช่" (ทำนายผิด)

2.7.4 F1-Score คือ ค่าเฉลี่ยหาค่ากลางที่ดีระหว่างค่า precision และ recall เพื่อใช้ในการวัดความสามารถในการจำแนกประเภท (Classification) ของแบบจำลอง โดยเฉพาะในกรณีที่ข้อมูลมีความไม่สมดุล เช่น เมื่อจำนวนตัวอย่างในกลุ่มหนึ่งมากกว่ากลุ่มอื่นมาก ซึ่งใช้เป็นตัวบอกถึงความเหมาะ ของการนำข้อมูลไปใช้ว่ามีความเหมาะสมเพียงใด พร้อมทั้งบอกถึงประสิทธิภาพของตัวแบบพยากรณ์

2.7.5 ROC Curve และ AUC (Area Under Curve) ทั้ง 2 เป็นเครื่องมือที่ใช้ในการประเมินประสิทธิภาพของโมเดลในการจำแนกประเภท (Classification) โดยเฉพาะในกรณีที่ข้อมูลมีความไม่สมดุล (Pasith Thanapatpisarn, 2565) โดยมีรายละเอียดดังนี้

2.7.5.1 ROC Curve คือ กราฟที่แสดงความสัมพันธ์ระหว่างการทำนายที่ถูกต้องและผิดของโมเดล หรือกราฟที่แสดงถึงการทำงานของโมเดลในการแยกแยะข้อมูลบวกและลบ โดยมักใช้สำหรับการประเมินโมเดลที่ทำนายผลเป็น 2 ประเภท

2.7.5.1.1 แกน x (Horizontal Axis) แสดงถึง False Positive Rate (FPR) หรืออัตราผลลัพธ์ที่ผิดพลาดที่โมเดลทำนายว่าเป็น "บวก" (แต่จริงๆ แล้วไม่ใช่) และเป็นกราฟที่แสดงความสัมพันธ์ระหว่างข้อมูลที่ทำนายทำนายผิด (แกน Xทำนายผิด)

สูตรการคำนวณ

$$FPR = \frac{FP}{FP+TN} \quad (2\text{-}19)$$

เมื่อ TP (True Positive) แทน จำนวนที่โมเดลทำนายว่า "ใช่" และเป็น "ใช่" จริง

FP (False Positive) แทน จำนวนที่โมเดลทำนายว่า "ใช่" แต่จริงๆ เป็น "ไม่ใช่" (ทำนายผิด)

2.7.5.1.2 **แกน y (Vertical Axis)** แสดงถึง **True Positive Rate (TPR)** หรือ **Recall** หรืออัตราผลลัพธ์ที่ถูกต้องที่โมเดลทำนายว่าเป็น "บวก" และจริงๆ แล้วมันคือ "บวก") และเป็นกราฟที่แสดงความสัมพันธ์ระหว่างข้อมูลที่ทำนายถูก (แกน Y)

สูตรการคำนวณ

$$TPR = \frac{TP}{TP+FN} \quad (2\text{-}20)$$

เมื่อ TN (True Negative) แทน จำนวนที่โมเดลทำนายว่า "ไม่ใช่" และเป็น "ไม่ใช่" จริง "ใช่" จริง

FN (False Negative) แทน จำนวนที่โมเดลทำนายว่า "ไม่ใช่" แต่จริงๆ เป็น "ใช่" (ทำนายผิด)

2.6.5.1.3 มุมมองบนกราฟมีดังนี้

ก. กราฟ ROC จะเริ่มต้นที่จุด (0, 0) ซึ่งหมายความว่า ไม่มีการทำนาย เลย

ข. กราฟจะขยับขึ้นไปทางขวาและสูงขึ้น ยิ่งไปทางมุมบนซ้าย (ใกล้จุด (0, 1)) แสดงว่าโมเดลทำงานได้ดี เนื่องจากโมเดลมีอัตรา True Positive (TPR) ที่สูง และ False Positive (FPR) ที่ต่ำ

ค. หากโมเดลไม่สามารถแยกแยะข้อมูลได้ดี จะเกิดกราฟที่ใกล้กับเส้นทแยงมุมจาก (0,0) ถึง (1,1) ซึ่งแสดงว่าโมเดลไม่สามารถแยกแยะข้อมูลบวกและลบได้ดี

2.7.5.2 AUC (Area Under the Curve) คือ พื้นที่ใต้กราฟ ROC Curve มันเป็นตัววัดประสิทธิภาพของโมเดลในการแยกแยะระหว่างคลาสบวกและลบ AUC มีค่าระหว่าง 0 และ 1 AUC สูง แสดงว่าโมเดลมีความสามารถในการแยกแยะข้อมูลได้ดี

2.7.5.2.1 การตีความ

ก. AUC มีค่าเท่ากับ 1 หมายถึงโมเดลมีประสิทธิภาพสูงสุด สามารถแยกแยะคลาสบวกและลบได้อย่างสมบูรณ์แบบ

ข. AUC มีค่าเท่ากับ 0.5 หมายถึงโมเดลไม่มีความสามารถในการแยกแยะข้อมูล แค่ทำนายเหมือนการทำนายแบบสุ่ม

ค. AUC มีค่าน้อยกว่า 0.5 หมายถึงโมเดลมีประสิทธิภาพแย่กว่าแค่การทำนายแบบสุ่ม

ง. AUC มีค่า 1 หมายถึงโมเดลที่มีประสิทธิภาพสูงสุด ซึ่งแสดงว่าโมเดลสามารถแยกแยะข้อมูล "บวก" และ "ลบ" ได้อย่างแม่นยำ

จ. AUC มีค่า 0.9 – 1 หมายถึงโมเดลที่ดีมาก มีความสามารถในการแยกแยะข้อมูลได้ดี

ฉ. AUC มีค่า 0.7 – 0.9 หมายถึงโมเดลที่มีประสิทธิภาพปานกลาง

ช. AUC มีค่า 0.5 – 0.7 หมายถึงโมเดลที่มีประสิทธิภาพต่ำ มีความสามารถแยกแยะข้อมูลได้ไม่ดี

ซ. AUC มีค่าน้อยกว่า 0.5 หมายถึงโมเดลที่แย่กว่าโมเดลสุ่ม

2.7.6 Confusion Matrix เป็นตารางที่ใช้ประเมินผลลัพธ์ของ โมเดลการจำแนกประเภท (Classification Model) โดยแสดงจำนวนค่าทำนายที่ถูกต้องและผิดพลาดในแต่ละประเภทของข้อมูล เปรียบเทียบค่าจริง (Actual) กับค่าที่โมเดลทำนาย (Predicted) สามารถใช้คำนวณค่าประสิทธิภาพของโมเดล ได้แก่ Accuracy, Precision, Recall และ F1-Score (Jirapat Klaokliang, s2564)

ตารางที่ 2-3 Confusion Matrix

	ค่าจริง (Actual)	
	True Positive	False Positive

ค่าการทำนาย (Predict)	(TP)	(FP)
	False Negative (FN)	True Negative (TN)

จากตารางที่ 2 เมื่อมีการทำนาย 2 ประเภท ผลลัพธ์ของการทำนายทั้งหมดที่เป็นไปได้ จะมีทั้งหมด 4 ค่า ดังต่อไปนี้

2.6.6.1 TP (True Positive) = จำนวนที่โมเดลทำนายว่า "ใช่" และเป็น "ใช่" จริง

2.6.6.2 TN (True Negative) = จำนวนที่โมเดลทำนายว่า "ไม่ใช่" และเป็น "ไม่ใช่" จริง

2.6.6.3 FP (False Positive) = จำนวนที่โมเดลทำนายว่า "ใช่" แต่จริงๆ เป็น "ไม่ใช่" (ทำนายผิด)

2.6.6.4 FN (False Negative) = จำนวนที่โมเดลทำนายว่า "ไม่ใช่" แต่จริงๆ เป็น "ใช่" (ทำนายผิด)

2.8 เครื่องมือและภาษาที่ใช้ในการพัฒนา

2.8.1 ภาษา Python

2.8.1.1 ความหมาย

ไพทอน (Python) คือหนึ่งในภาษาโปรแกรมระดับสูงที่ใช้กันอย่างแพร่หลาย ถูกออกแบบเพื่อให้มีโครงสร้างและ ไวยากรณ์ของภาษาที่ไม่ซับซ้อน เข้าใจง่าย มีการใช้พัฒนาแอปพลิเคชัน เว็บไซต์ รวมถึงแอปพลิเคชั่นมือถือหรือ อุปกรณ์เคลื่อนที่ด้วย หน้าที่ของ Python ก็คือการทำงานแปลชุดคำสั่งทีละบรรทัดเพื่อป้อนเข้าสู่หน่วยประมวลผล ให้คอมพิวเตอร์ทำงานตามที่เราต้องการ หรือเรียกว่าการทำงานแบบ Interpreter ด้วยภาษาที่ง่ายในการเขียน “Python” จึงมีความเหมาะสมสำหรับผู้ที่เริ่มต้นเขียนโปรแกรมไปจนถึงนักพัฒนาในองค์กรบริษัทใหญ่ อย่างเช่น Netflix, Spotify, Google, Amazon, และ Facebook เป็นต้น มี สิ่งแวดล้อมสำหรับการพัฒนาแบบเบ็ดเสร็จ (Integrated Development Environment หรือ IDE) คือซอฟต์แวร์ที่ช่วยให้นักพัฒนามีเครื่องมือที่จำเป็นในการเขียน แก้ไข ทดสอบ และดีบักโค้ดในที่เดียว ได้แก่ PyCharm IDLE Spyder และ Atom รวมถึงมี ชุดพัฒนาซอฟต์แวร์ (SDK) คือชุดเครื่องมือด้านซอฟต์แวร์ที่นักพัฒนาสามารถใช้เพื่อสร้างแอปพลิเคชันซอฟต์แวร์ในภาษาที่กำหนด SDK ส่วนใหญ่มีความเฉพาะเจาะจงสำหรับแพลตฟอร์มฮาร์ดแวร์และระบบปฏิบัติการที่แตกต่างกัน โดย Python SDK มี

เครื่องมือมากมาย เช่น ไลบรารี ตัวอย่างโค้ด และคู่มือสำหรับนักพัฒนา ซึ่งนักพัฒนาพบว่ามีประโยชน์เมื่อต้องเขียนแอปพลิเคชัน (Pacharee Toorakidsana, 2564)

2.8.1.2 การใช้งานไพทอน (Python)

2.8.1.2.1 ใช้ในด้านการพัฒนาเว็บแอปพลิเคชั่น (Web Application) ฝั่งเซิร์ฟเวอร์ประกอบด้วยฟังก์ชันแบ็คเอนด์ที่ซับซ้อนซึ่งเว็บไซต์ดำเนินการเพื่อแสดงข้อมูลต่อผู้ใช้ ตัวอย่างเช่น เว็บไซต์ต้องโต้ตอบกับฐานข้อมูล สื่อสารกับเว็บไซต์อื่น และปกป้องข้อมูลเมื่อส่งข้อมูลผ่านเครือข่าย

2.8.1.2.2 ใช้เป็นภาษาการเขียนสคริปต์ คือภาษาการเขียนโปรแกรมที่ทำให้งานที่มนุษย์ทำตามปกติเป็นไปโดยอัตโนมัติ โปรแกรมเมอร์จึงใช้สคริปต์ Python อย่างแพร่หลายเพื่อทำให้งานประจำวันหลายอย่างเป็นไปโดยอัตโนมัติ

2.8.1.2.3 ใช้ด้านวิทยาศาสตร์ข้อมูล สร้างมูลค่าคุณค่าจากข้อมูล และการทำแมชชีนเลิร์นนิง (Machine learning) จะสอนคอมพิวเตอร์ให้เรียนรู้จากข้อมูลโดยอัตโนมัติและทำนายได้อย่างแม่นยำ โดยใช้ ถาษาไพทอน (Python) สำหรับงานด้านวิทยาศาสตร์ข้อมูลต่างๆ ดังต่อไปนี้

ก. การแก้ไขและลบข้อมูลที่ไม่ถูกต้อง ซึ่งเรียกว่าการทำความสะอาดข้อมูล

ข. การแยกและเลือกคุณสมบัติต่างๆ ของข้อมูล

ค. การระบุประเภทข้อมูล ซึ่งเป็นการเพิ่มชื่อที่มีความหมายสำหรับข้อมูล

ง. การค้นหาสถิติต่างๆ จากข้อมูล

จ. การแสดงข้อมูลด้วยภาพโดยใช้แผนภูมิและกราฟ เช่น แผนภูมิเส้น กราฟแท่ง

นักวิทยาศาสตร์ข้อมูล (data scientist) ใช้ไลบรารี Python ML เพื่อฝึกฝนโมเดลแมชชีนเลิร์นนิง (Machine learning) และสร้างตัวจำแนกที่จำแนกประเภทข้อมูลได้อย่างแม่นยำ บุคคลในแวดวงต่างๆ ใช้ตัวจำแนกแบบไพทอน (Python) เพื่อทำงานด้านการจำแนกประเภท เช่น การจำแนกประเภทรูปภาพ ข้อความ และการรับส่งข้อมูลทางเครือข่าย การรู้จำเสียง และการจดจำใบหน้า นักวิทยาศาสตร์ข้อมูลยังใช้ Python สำหรับดีปเลิร์นนิ่ง ซึ่งเป็นเทคนิคแมชชีนเลิร์นนิง (Machine learning) ขั้นสูง

2.8.1.2.4 นักพัฒนาซอฟต์แวร์มักใช้ไพทอน (Python) สำหรับงานด้านการพัฒนาและการประยุกต์ใช้ซอฟต์แวร์ต่างๆ

2.8.1.3 คุณสมบัติไพทอน (Python)

2.8.1.3.1 ภาษาที่แปลผลแล้ว

2.8.1.3.2 ภาษาที่ใช้งานง่าน

2.8.1.3.3 ภาษาระดับสูง

2.8.1.3.4 ภาษาที่ระบุประเภทแบบไดนามิก

2.8.1.3.5 ภาษาเชิงอ็อบเจ็กต์

2.8.1.4 ไลบรารี (Library) คือชุดของโค้ดที่ใช้บ่อยซึ่งนักพัฒนาสามารถใช้ในโปรแกรม ไพทอน (Python) เพื่อหลีกเลี่ยงการเขียนโค้ดขึ้นใหม่ทั้งหมด ตามค่าเริ่มต้นแล้วไพทอน (Python) จะมาพร้อมกับไลบรารีมาตรฐาน ซึ่งมีฟังก์ชันที่นำกลับมาใช้ใหม่ได้มากมาย นอกจากนี้ยังมีไลบรารี ไพทอน (Python) มากกว่า 137,000 รายการสำหรับการประยุกต์ใช้ต่างๆ รวมถึงการพัฒนาเว็บ วิทยาศาสตร์ข้อมูล และแมชชีนเลิร์นนิง (ML) ไลบรารียอดนิยมได้แก่

2.8.1.4.1 Matplotlib & Seaborn นักพัฒนาซอฟต์แวร์ใช้ Matplotlib & Seabornเพื่อลงจุดข้อมูลในกราฟิกสองมิติและสามมิติ (2D และ 3D) คุณภาพสูง ซึ่งมักจะใช้ในงานทางวิทยาศาสตร์ Matplotlib ช่วยให้คุณสามารถแสดงข้อมูลเป็นภาพโดยแสดงผลในแผนภูมิต่างๆ เช่น แผนภูมิแท่งและแผนภูมิเส้น คุณยังสามารถลงจุดแผนภูมิได้หลายรายการพร้อมกัน และกราฟิกสามารถใช้งานได้ในทุกแพลตฟอร์มอีกด้วย

2.8.1.4.2 Pandas มีโครงสร้างข้อมูลที่ปรับให้เหมาะสมและยืดหยุ่น ซึ่งคุณสามารถใช้เพื่อจัดการชุดข้อมูลเวลาและข้อมูลที่มีโครงสร้าง เช่น ตารางและอาร์เรย์ ตัวอย่างเช่น คุณสามารถใช้ Pandas เพื่ออ่าน เขียน ผสานรวม กรอง และจัดกลุ่มข้อมูลได้ หลายคนจึงใช้สำหรับงานด้านวิทยาศาสตร์ข้อมูล การวิเคราะห์ข้อมูล และเทคนิคแมชชีนเลิร์นนิง (Machine learning)

2.8.1.4.3 NumPy เป็นไลบรารียอดนิยมที่นักพัฒนาใช้เพื่อสร้างและจัดการอาร์เรย์ จัดการรูปร่างเชิงตรรกะ และดำเนินการคำนวณพีชคณิตเชิงเส้นได้อย่างง่ายดาย โดย NumPy รองรับการทำงานร่วมกับภาษาต่างๆ มากมาย เช่น C และ C++

2.8.1.4.4 Scikit-learn เป็นไลบรารียอดนิยมที่นักพัฒนาสำหรับการพัฒนาโมเดล Machine Learning รองรับอัลกอริธึมต่าง ๆ เช่น Linear Regression, Decision Tree,

SVM, Random Forest, KNN ฯลฯ และมีเครื่องมือสำหรับการประเมินผล เช่น Confusion Matrix, Precision, Recall, F1-score

2.8.1.5 เฟรมเวิร์ก (Framework) คือชุดข้อมูลของแพคเกจและโมดูล โดยโมดูลคือชุดของโค้ดที่เกี่ยวข้อง และแพคเกจคือชุดของโมดูล ทั้งนี้นักพัฒนาสามารถใช้เฟรมเวิร์ก ไพทอน (Python) เพื่อสร้างแอปพลิเคชัน Python ได้รวดเร็วขึ้น เนื่องจากไม่ต้องกังวลกับรายละเอียดระดับต่ำต่างๆ เช่น วิธีการที่การสื่อสารเกิดขึ้นในเว็บแอปพลิเคชัน หรือวิธีการที่ Python จะทำให้โปรแกรมเร็วขึ้น ไพทอน (Python) มีเฟรมเวิร์ก 2 ประเภท ได้แก่

2.8.1.5.1 เฟรมเวิร์กแบบฟูลสแตกมีเกือบทุกสิ่งที่จำเป็นสำหรับการสร้างแอปพลิเคชันขนาดใหญ่

2.8.1.5.2 ไมโครเฟรมเวิร์กเป็นเฟรมเวิร์กพื้นฐานที่มีฟังก์ชันการทำงานน้อยที่สุดสำหรับการสร้างแอปพลิเคชัน ไพทอน (Python) อย่างง่าย นอกจากนี้ยังมีส่วนขยายหากแอปพลิเคชันต้องการฟังก์ชันที่ซับซ้อนยิ่งขึ้นอีกด้วย

เฟรมเวิร์ก (Framework) ที่นิยมได้แก่ Django, Flask, TurboGears, Apache MXNet, PyTorch และ FastAPI (Amazon Web Services, ม.ป.ป.)

2.8.1.6 FastApi เป็นเว็บเฟรมเวิร์ก (Web Framework) สำหรับพัฒนา API ด้วยไพทอน (Python) ที่รวดเร็ว ใช้งานง่าย และรองรับมาตรฐานสมัยใหม่ เช่น ASGI (Asynchronous Server Gateway Interface), Pydantic, และ OpenAPI เฟรมเวิร์กนี้ได้รับความนิยมมากขึ้นเรื่อยๆ เนื่องจากสามารถสร้าง RESTful API ได้ง่ายและรวดเร็ว เร็วกว่า Flask และ Django และมีประสิทธิภาพเทียบเท่า Node.js และ Go ในบางกรณี (Littikrai. 2564)

2.8.1.6.1 จุดเด่นของ FastAPI

ก. เร็วมาก ประสิทธิภาพสูงใกล้เคียงกับ Node.js และ Go

ข. ใช้งานง่าย มีระบบ Type Hints จาก Python ที่ช่วยให้พัฒนาได้รวดเร็วขึ้น

ค. รองรับ Asynchronous (async/await) สามารถใช้ async กับ await เพื่อรองรับ I/O-bound operations

ง. รองรับ OpenAPI & JSON Schema อัตโนมัติ สร้าง Swagger UI และ ReDoc ให้อัตโนมัติ

จ. ตรวจสอบข้อมูลอัตโนมัติ (Data Validation) ใช้ Pydantic เพื่อตรวจสอบอินพุต

ฉ. รองรับ WebSocket และ GraphQL

ช. รองรับ Dependency Injection (DI) ช่วยให้โค้ดมีโครงสร้างที่ดีขึ้น

ซ. รองรับ OAuth2, JWT, และการยืนยันตัวตน

2.8.2 Google Colaboratory หรือ Colab เป็น Jupyter Notebook ที่ทำงานบนระบบคลาวด์ของ Google เป็น IDE ที่อนุญาตให้ผู้ใช้เขียนซอร์สโค้ดในตัวแก้ไขและเรียกใช้จากเบราว์เซอร์ ซึ่งช่วยให้นักพัฒนา นักวิจัย และผู้เรียนสามารถ เขียนและรันโค้ดไพทอน (Python) ได้ฟรี โดยไม่ต้องติดตั้งซอฟต์แวร์ใดๆ บนคอมพิวเตอร์ของตัวเอง เน้นงานแมชชีนเลิร์นนิง การวิเคราะห์ข้อมูล โครงการการศึกษา ฯลฯ มีทรัพยากร GPU และ TPU ได้ฟรีการคำนวณให้ใช้แต่มีข้อจำกัดสำหรับทรัพยากรณ์จำเป็นต้องจ่ายเงินซื้อเวอร์ชันโปรเพิ่ม หากรันโค้ดบนเครื่องเสมือนบางตัวหากปิดเบราว์เซอร์ เครื่องจะถูกลบออกหลังจากไม่มีการใช้งานเป็นระยะเวลาหนึ่งเพื่อเพิ่มทรัพยากร โดยสามารถเรียกใช้โค้ดแยกออกจากผู้ใช้และทรัพยากรอื่น ดังนั้นสามารถกู้คืนเครื่องเสมือนกลับเป็นสถานะดั้งเดิมได้หากคุณมีปัญหา สามารถแชร์และทำงานร่วมกันได้ง่าย และมีการเก็บไฟล์ไว้บน Google drive สามารถดาวน์โหลด ในรูปแบบโอเพ่นซอร์สได้แก่ Jupiter .ipynb รองรับ Jupyter และ IPython คุณสมบัติหลักที่สำคัญคือ Colab ก็คือใช้งานง่าย สามารถใช้งานร่วมกับ สามารถใช้งานร่วมกับ ใช้งาน Git hub บันทึกข้อความ รหัส ผลลัพธ์ และความคิดเห็นอธิบายในการเขียนโค้ด (John Burke, 2566)

2.8.3 เว็บแอปพลิเคชัน

เว็บแอปพลิเคชัน คือ การพัฒนาระบบงานบนเว็บหรือแอปพลิเคชันที่สามารถเข้าถึงได้ผ่านเวปเบราว์เซอร์ เช่น Google Chrome, Mozilla Firefox โดยไม่จำเป็นต้องติดตั้งลงบนอุปกรณ์ของผู้ใช้ ทำงานผ่านอินเทอร์เน็ตหรืออินทราเน็ตซึ่งทำงานในลักษณะของ Client - Server เว็บแอปพลิเคชันเป็นส่วนหนึ่งของ Web Technology ที่ช่วยให้ผู้ใช้สามารถทำงานหรือเข้าถึงข้อมูลต่าง ๆ ได้จากทุกที่ผ่านอินเทอร์เน็ต รองรับการใช้งานข้ามแพลตฟอร์ม ไม่ว่าจะเป็นคอมพิวเตอร์ สมาร์ทโฟน หรือแท็บเล็ต ต่างจากแอปพลิเคชันแบบเดิม (Desktop Application) ที่ต้องติดตั้งลงบนอุปกรณ์ การอัปเดตและบำรุงรักษาง่าย เนื่องจากเว็บแอปพลิเคชันถูกติดตั้งและรันบนเซิร์ฟเวอร์ นักพัฒนาสามารถอัปเดตระบบได้โดยไม่ต้องให้ผู้ใช้ดาวน์โหลดซอฟต์แวร์ใหม่ (Amazon Web Services, ม.ป.ป.) สามารถรองรับผู้ใช้จำนวนมากพร้อมกันผ่านการขยายเซิร์ฟเวอร์ (Scalability) เช่น การใช้ Load Balancer และ Cloud Computing และมีการรักษาความปลอดภัยของข้อมูลเนื่องจากข้อมูลของผู้ใช้ถูกจัดเก็บบนเซิร์ฟเวอร์ จึงต้องมีมาตรการป้องกัน เช่น SSL/TLS Encryption, Authentication, Authorization และ Firewall โดยหลักการทำงานผู้ใช้งานสามารถเพิ่ม แก้ไขประวัติ การบันทึกข้อมูลผ่านทางหน้าเว็บการใช้งานและข้อมูลต่างๆจะถูกจัดเก็บลงฐานข้อมูลเพื่อใช้

งานต่อไป สามารถแบ่งส่วนของเว็ปแอปพลิเคชั่นในการพัฒนาออกเป็น 2 ส่วนได้แก่ ส่วนที่ติดต่อกับผู้ใช้งานและส่วนที่ประมวนผลและจัดการข้อมูล (พิสุทธิ์ ตั้งพิษฐานสกุล, 2557)

2.8.3.1 ส่วนติดต่อผู้ใช้ (Front-End) สามารถมองเห็นและโต้ตอบผ่านเว็บเบราว์เซอร์ ภาษาและเฟรมเวิร์ก (Framework) ที่ใช้ในการสร้างส่วนติดต่อผู้ใช้ได้แก่

2.8.3.1.1 ภาษา HTML (HyperText Markup Language) ใช้สร้างโครงสร้างพื้นฐานของหน้าเว็บ

2.8.3.1.2 ภาษา CSS ใช้กำหนดรูปแบบการแสดงผล เช่น สี ขนาด และการจัดวาง ทำให้เว็บดูเป็นระเบียบและสวยงามขึ้น สามารถใช้ร่วมกับ HTML ได้โดยการกำหนดสไตล์ให้กับองค์ประกอบต่างๆ

2.8.3.1.3 ภาษา JavaScript เป็นภาษาสคริปต์ใช้เพิ่มความสามารถเชิงโต้ตอบกับผู้ใช้ เช่น ปุ่มคลิก, ฟอร์ม, แสดงผลแบบไดนามิก, และการโหลดข้อมูลโดยไม่ต้องรีเฟรชหน้า และการตรวจสอบฟอร์มและการโหลดข้อมูลแบบไม่ต้องรีเฟรชหน้าเว็บ

2.8.3.1.4 Bootstrap เป็นเฟรมเวิร์ก (Framework) สำหรับการพัฒนาเว็บไซต์ที่ช่วยให้การออกแบบเว็บทำได้ง่ายและรวดเร็วขึ้น โดยมีการรวม CSS, JavaScript และ HTML Components สำเร็จรูปมาให้พร้อมใช้งานได้แก่ มีองค์ประกอบเว็บที่ออกแบบไว้แล้ว เช่น ปุ่ม (Buttons), ฟอร์ม (Forms), การ์ด (Cards) Bootstrap ถูกพัฒนาโดย Twitter และเปิดตัวครั้งแรกในปี 2011 โดยมีเป้าหมายเพื่อช่วยให้นักพัฒนาเว็บสร้างเว็บไซต์ที่สวยงาม ตอบสนองทุกอุปกรณ์ (Responsive) ได้อย่างง่ายดาย ะบบเลย์เอาต์ที่ช่วยจัดวางองค์ประกอบบนเว็บได้อย่างเป็นระเบียบ (Grid System)

2.8.3.2 ส่วนประมวลผลและจัดการข้อมูล (Back-End) เป็นส่วนประมวลผลตรรกะและการทำงานของเว็บแอปพลิเคชัน การทำงานในส่วนนี้จะเริ่มหลังจากเว็บแอปพลิเคชันได้รับ HTTP Request มาจากผู้ใช้งาน ทำการประมวลผลและส่งข้อมูลกลับไปให้กับผู้ใช้งานเฟรมเวิร์ก (Frameworks) และไลบรารี (Library) ยอดนิยมสำหรับ Front-End ได้แก่ React.js Angular และ Vue.js (พีรพงษ์ พิมพ์อูบ, 2563)

2.8.3.3 React.js คือ JavaScript ไลบรารี (Library) ที่ใช้สำหรับสร้างยูเซอร์อินเตอร์เฟส (User Interface) ถูกพัฒนาโดย Facebook (Meta) และเปิดตัวครั้งแรกในปี 2013 โดยสามารถเขียนโค้ดในการสร้างยูเซอร์อินเตอร์เฟสที่มีความซับซ้อนแบ่งเป็นส่วนย่อยออกจากกัน แต่ละส่วนสามารถทำงานได้อย่างอิสระ และสามารถนำแต่ละส่วนไปใช้ซ้ำได้ โดยมีละกษณะจุดเด่นที่ การทำงานแบบ Component-Based Architecture แบ่ง UI ออกเป็นส่วนย่อยๆ ที่สามารถนำมาใช้ซ้ำได้ เช่น ปุ่ม,

ฟอร์ม, การ์ดข้อมูล ฯลฯ ทำให้การพัฒนาเป็นระบบและจัดการโค้ดได้ง่ายขึ้น , Virtual DOM คือการเปรียบเทียบและอัปเดต UI เฉพาะส่วนที่เปลี่ยนแปลง ซึ่งทำให้การแสดงผลเร็วขึ้นกว่าการอัปเดต DOM จริงโดยตรง และ One-Way Data Binding สามารถควบคุมการเปลี่ยนแปลงของข้อมูลได้ง่าย และลดปัญหาข้อมูลที่ไม่สอดคล้องกัน ซึ่งช่วยให้การพัฒนา UI มีประสิทธิภาพสูงและจัดการโค้ดได้ง่าย React ได้รับความนิยมอย่างมาก เนื่องจากรองรับการพัฒนาเว็บแอปพลิเคชันแบบ Single Page Application (SPA) ที่รวดเร็วและมีประสิทธิภาพ ใช้ JSX เป็น Syntax ที่ช่วยให้เราเขียนโค้ด UI ที่ดูคล้าย HTML ภายใน JavaScript ได้ ทำให้โค้ดอ่านง่ายและใช้งานสะดวกขึ้น สามารถทำงานร่วมกับ Next.js เพื่อรองรับ Server-Side Rendering (SSR) ซึ่งช่วยให้โหลดหน้าเว็บเร็วขึ้นและรองรับ SEO ได้ดี (Papimpat Nimprasert, 2567)

2.8.3.4 Vite เป็นเครื่องมือสำหรับ สร้างและพัฒนาเว็บแอปพลิเคชัน (Frontend Build Tool) ที่ออกแบบมาให้ เร็วกว่า Webpack โดยเฉพาะเมื่อใช้กับ Reac.js, Vue, Svelte และ Vanilla JavaScript รองรับ TypeScript และ JSX ไม่มีไฟล์ config ที่ซับซ้อน รองรับการ Build ที่เร็วขึ้นโดยใช้ Rollup มีระบบ Tree Shaking ในตัว โหลดเฉพาะโค้ดที่จำเป็น สามาถเริ่มต้นเซิร์ฟเวอร์ Dev ได้ทันที ไม่ต้องรอการ Bundling และมี Hot Module Replacement (HMR) ที่เร็วมาก โหลดเฉพาะส่วนที่เปลี่ยนแปลง (Supawish kaewjing, 2566)

2.8.3.5 Material-UI (MUI) เป็นไลบรารีสำหรับ React ที่ช่วยให้นักพัฒนาสามารถสร้าง UI ได้ง่ายขึ้นโดยใช้ Material Design ของ Google ซึ่งเป็นแนวทางการออกแบบที่เน้นรองรับการปรับแต่ง (Customization) สามารถเปลี่ยนธีม, สี และสไตล์ได้ง่ายต่อการใช้งานง่ายและให้ความสวยงาม ความง่ายต่อการใช้งาน และประสบการณ์ที่ดีของผู้ใช้ (UX/UI) ประสิทธิภาพสูงโหลดเร็วและรองรับการเรนเดอร์แบบ SSR (Server-Side Rendering) MUI มี Component สำเร็จรูป ที่สามารถใช้ได้เลย เช่น Button, Card, Dialog, Table, TextField และอีกมากมาย โดยทั้งหมดถูกออกแบบมาให้ดูทันสมัยและใช้งานได้กับทุกอุปกรณ์ (Responsive Design) (Launchplatfrom, 2566)

2.8.4 Program Visual studio code (VS Code) โปรแกรมแก้ไขข้อความ (text editor) ที่ได้รับความนิยมสูงมากในการพัฒนาเป็นโปรแกรม การเขียนโค้ด และแก้ไขซอร์สโค้ดที่มีขนาดเล็กแต่มีประสิทธิภาพสูงเป็นโอเพนซอร์สที่เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานหลายแพลตฟอร์ม รองรับการทำงานและสามารถใช้งานได้ฟรีทั้งบน Microsoft, macOs และ Linux ซึ่งสนับสนุน JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ง่าย และรองรับการเปิดใช้งานภาษาต่างๆหลายภาษาโปรแกรม เช่น C++, C#, Java, Python, PHP, FastAPI, React.js และ Go สามารถปรับเปลี่ยนธีมได้ มีส่วน Debugger และ Commands สามารถปรับแต่งให้เหมาะสมกับความต้องการของผู้ใช้ได้อย่างยืดหยุ่น VS Code พัฒนาโดย Microsoft VS Code คือความสามารถในการติดตั้ง Extensions หรือปลั๊กอินที่ช่วยเพิ่มฟังก์ชันการทำงาน เช่น IntelliSense

(ช่วยแนะนำการเติมคำหรือโค้ดที่ควรใช้) หรือปรับแต่งการทำงานของโปรแกรมได้อย่างหลากหลาย เช่น ธีมสี, คีย์ลัด (Keyboard Shortcuts), หรือ การตั้งค่าคอนฟิกต่างๆ เพื่อให้ตรงกับความต้องการของผู้ใช้ VS Code รองรับการดีบัคโค้ดโดยสามารถตรวจสอบข้อผิดพลาดที่เกิดขึ้นในโค้ดได้โดยตรงจากตัวโปรแกรม โดยไม่ต้องใช้เครื่องมือภายนอก มีฟีเจอร์ Git integration ที่ช่วยให้คุณสามารถจัดการเวอร์ชันของโค้ดได้โดยตรงจากภายในโปรแกรม เช่นการ commit, push, และ pull โค้ด และสามารถรองรับการแบ่งหน้าจอการทำงานออกเป็นหลายส่วน (Split View) มี Integrated Terminal ที่ช่วยให้คุณสามารถรันคำสั่งต่างๆ รองรับการจัดการโปรเจ็กต์อย่างมีระเบียบ รวมถึงมี Live Share ทำงานร่วมกับผู้อื่นในทีมแบบ Real-time (ณัฐพล แสนคำ, 2563)

2.8.5 Postman คือเครื่องมือสำหรับการทดสอบ API (Application Programming Interface) ที่ใช้กันอย่างแพร่หลายในการพัฒนาแอปพลิเคชัน โดยเฉพาะในกระบวนการพัฒนา API และการทดสอบการทำงานของ API ในสภาพแวดล้อมต่างๆ Postman ช่วยให้นักพัฒนาสามารถส่งคำขอ (requests) ไปยังเซิร์ฟเวอร์, ตรวจสอบผลลัพธ์ที่ได้รับกลับมา (responses), และการจัดการต่างๆ เช่น การตั้งค่า, การบันทึกคำขอ, การสร้างคอลเลกชัน ฯลฯ ได้อย่างสะดวกและง่ายดาย (ณัฐพล แสนคำ, 2563) ฟีเจอร์หลักของ Postman มีดังนี้

2.8.5.1 การส่งคำขอ (Request) Postman รองรับการส่งคำขอ HTTP หลากหลายประเภท เช่น

2.8.5.1.1 GET ใช้เพื่อดึงข้อมูลจากเซิร์ฟเวอร์

2.8.5.1.2 POST ใช้เพื่อส่งข้อมูลไปยังเซิร์ฟเวอร์

2.8.5.1.3 PUT ใช้เพื่ออัปเดตข้อมูลที่มีอยู่ในเซิร์ฟเวอร์

2.8.5.1.4 DELETE ใช้เพื่อลบข้อมูลจากเซิร์ฟเวอร์

2.8.5.1.5 PATCH ใช้เพื่ออัปเดตข้อมูลบางส่วนบนเซิร์ฟเวอร์

2.8.5.2 การส่งคำขอสามารถตั้งค่าตัวเลือกได้หลากหลาย เช่น

2.8.5.2.1 URL ของ API

2.8.5.2.2 Body สำหรับส่งข้อมูลที่ต้องการกับคำขอ เช่น ข้อมูลในรูปแบบ JSON หรือ XML

2.8.5.2.3 PUT ใช้เพื่ออัปเดตข้อมูลที่มีอยู่ในเซิร์ฟเวอร์

2.7.5.3 การดูและตรวจสอบคำตอบ (Response) เมื่อส่งคำขอแล้ว Postman จะแสดงผลลัพธ์จากเซิร์ฟเวอร์ที่เรียกว่า Response ซึ่งจะประกอบด้วย

2.8.5.3.1 Status Code เช่น 200 (OK), 404 (Not Found), 500 (Internal Server Error) เป็นต้น

2.8.5.3.2 Headers ซึ่งให้ข้อมูลเพิ่มเติมเกี่ยวกับการตอบกลับ เช่นประเภทของข้อมูล (Content-Type)

2.8.5.3.3 Body ซึ่งเป็นข้อมูลที่ส่งกลับจากเซิร์ฟเวอร์ โดยอาจจะเป็นในรูปแบบ JSON, XML, หรือข้อความธรรมดา

2.8.5.3.4 Time ซึ่งแสดงระยะเวลาในการตอบสนองจากเซิร์ฟเวอร์

2.8.5.4 การจัดการคอลเลกชัน (Collection) Postman ให้คุณสามารถจัดระเบียบคำขอ API โดยการสร้าง Collection ซึ่งเป็นกลุ่มของคำขอที่สามารถใช้ซ้ำได้ ตัวอย่างเช่น คุณอาจจะสร้าง Collection สำหรับการทดสอบ API ของโปรเจ็กต์หนึ่งๆ โดยสามารถจัดระเบียบคำขอ, เพิ่มคำอธิบาย, หรือกรอกค่า default เพื่อทดสอบในครั้งถัดไปได้

2.8.5.5 การตั้งค่าตัวแปร (Environment Variables) Postman รองรับการตั้งค่าตัวแปรเพื่อใช้ในการทดสอบที่แตกต่างกันในแต่ละสภาพแวดล้อม (Environment) เช่น การทดสอบในสภาพแวดล้อมที่ใช้ URL ต่างกันสำหรับ Development, Testing, และ Production การตั้งค่าตัวแปรช่วยให้สามารถเปลี่ยนแปลง URL หรือค่าอื่นๆ ได้อย่างง่ายดายโดยไม่ต้องไปแก้ไขในทุกคำขอ

2.8.5.6 การทดสอบอัตโนมัติ (Automated Testing) Postman รองรับการเขียนสคริปต์ทดสอบ (Test Scripts) โดยใช้ JavaScript ซึ่งช่วยให้คุณสามารถตรวจสอบผลลัพธ์ที่ได้รับจาก API โดยอัตโนมัติ ตัวอย่างเช่น ตรวจสอบว่า status code ที่ได้รับเป็น 200 หรือไม่, ข้อมูลใน Response body ถูกต้องตามที่คาดหวังหรือไม่ ฯลฯ

2.8.5.7 การแชร์และทำงานร่วมกัน Postman ช่วยให้การทำงานร่วมกันระหว่างทีมพัฒนาง่ายขึ้น ด้วยฟีเจอร์ที่สามารถแชร์ Collection หรือ Environment ให้กับสมาชิกในทีมอื่นๆ ได้ สามารถทำงานร่วมกันใน Team Workspace ที่แชร์ข้อมูลเกี่ยวกับ API ต่างๆ โดยไม่ต้องทำงานแยกกัน

2.8.5.8 การบันทึกคำขอและการตั้งค่า (History) Postman จะเก็บประวัติของคำขอทั้งหมดที่คุณเคยส่งใน History ทำให้คุณสามารถเรียกดูคำขอที่เคยส่งไปแล้วได้, ทดสอบคำขอนั้นใหม่, หรือคัดลอกคำขอไปใช้ในการทดสอบใหม่ในภายหลัง

2.9 งานวิจัยที่เกี่ยวข้อง

เครือวัลย์ เนตรพนา และ ศิริสรรพ เหล่าหะเกียรติ (เครือวัลย์ เนตรพนา, 2023) ได้ศึกษาการวิเคราะห์ความเสี่ยงในการผิดนัดชำระของลูกหนี้บัตรเครดิตโดยใช้อัลกอริทึมการเรียนรู้ของเครื่องหลายรูปแบบ ได้แก่ Logistic Regression, XGBoostClassifier, K-Nearest Neighbors, Random Forest, Support Vector Classifier (SVC) และ Gradient Boosting โดยใช้ Confusion Matrix เป็นเกณฑ์ในการประเมินประสิทธิภาพ ซึ่งพบว่าอัลกอริทึม XGBoost ให้ผลลัพธ์ดีที่สุดในงานวิจัยนี้

สกุลกาญจน์ ทองคำ และนุวีย์ วิวัฒนวัฒนา (สกุลกาญจน์ ทองคำ, 2022) มุ่งเน้นการทำนายการผิดนัดชำระของลูกหนี้บัตรเครดิตเช่นกัน โดยใช้อัลกอริทึม Logistic Regression, Decision Tree, AdaBoost และ Random Forest โดยผลการประเมินผ่าน Confusion Matrix ชี้ให้เห็นว่า XGBoost เป็นวิธีที่มีประสิทธิภาพสูงสุด

สุดากาญจน์ ติดตะ และ ดร.ธนภัทร ฆังคะจิตร (สุดากาญจน์ ติดตะ, 2021) พัฒนาระบบอนุมัติเงินกู้สวัสดิการพนักงานอัตโนมัติโดยใช้เทคนิค Machine Learning ได้แก่ Naïve Bayes, K-Nearest Neighbor, Support Vector Machine และ Random Forest โดย Random Forest ให้ผลลัพธ์ดีที่สุดตามการประเมินด้วย Confusion Matrix

ขวัญนภา พิมพ์ชารีย์ (ขวัญนภา พิมพ์ชารีย์, 2021) ได้ประยุกต์ใช้เหมืองข้อมูลสำหรับการพิจารณาการให้สินเชื่อในธนาคาร โดยใช้อัลกอริทึม Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) และ Decision Tree พร้อมใช้วิธีประเมินแบบ Cross Validation, Chi-Square และ Information Gain ซึ่ง SVM แสดงผลลัพธ์ที่ดีที่สุดในบริบทนี้

Makumburage Poornima and Chathurangi Peiris (Makumburage Poornima, 2019) จาก University of Colombo ศึกษาการพยากรณ์การอนุมัติบัตรเครดิตด้วยเทคนิค Machine Learning โดยใช้อัลกอริทึม ANN Model และ Support Vector Machine (SVM) ซึ่งพบว่า SVM มีประสิทธิภาพดีที่สุด

สุเมธ จุฑาจันทร์ และ สมพร ปั่นโภชา (สุเมธ จุฑาจันทร์, 2022) ทำการเปรียบเทียบผลลัพธ์ของการอนุมัติสินเชื่อด้วย 3 แบบจำลอง ได้แก่ k-Nearest Neighbors (KNN), Logistic Regression และ CART Model โดย KNN ให้ผลที่ดีที่สุดภายใต้การประเมินด้วย Confusion Matrix

Naman Dalsania, Devang Punatar และ Deep Kothari (Naman Dalsania, 2022) ในงานวิจัยนี้ ได้ศึกษาและเปรียบเทียบอัลกอริทึมการจำแนกประเภทการอนุมัติบัตรเครดิต เช่น Support Vector Machine (SVM), Adaboost , Gradient Boosting ซึ่งจากการประเมินด้วย Confusion Matrix พบว่า Gradient Boosting ให้ผลที่โดดเด่นที่สุดในงานวิจัย

Yiran Zhao (Yiran Zhao, 2022) ในงานวิจัยสองชิ้น ได้ศึกษาและเปรียบเทียบอัลกอริทึมการจำแนกประเภทการอนุมัติบัตรเครดิต เช่น Logistic Regression, Linear Support Vector และ

Naïve Bayes ซึ่งจากการประเมินด้วย Confusion Matrix พบว่า Linear Support Vector ให้ผลที่ดีที่สุดงานวิจัย

Viswanatha V และคณะ (Viswanatha V, 2023) ศึกษาการพยากรณ์การอนุมัติเงินกู้ในธนาคารด้วย Machine Learning โดยใช้ Random Forest, Naïve Bayes, Decision Tree และ KNN ซึ่งผลการวิจัยแสดงให้เห็นว่า Naïve Bayes มีประสิทธิภาพดีที่สุดในกลุ่มตัวอย่างที่ศึกษา

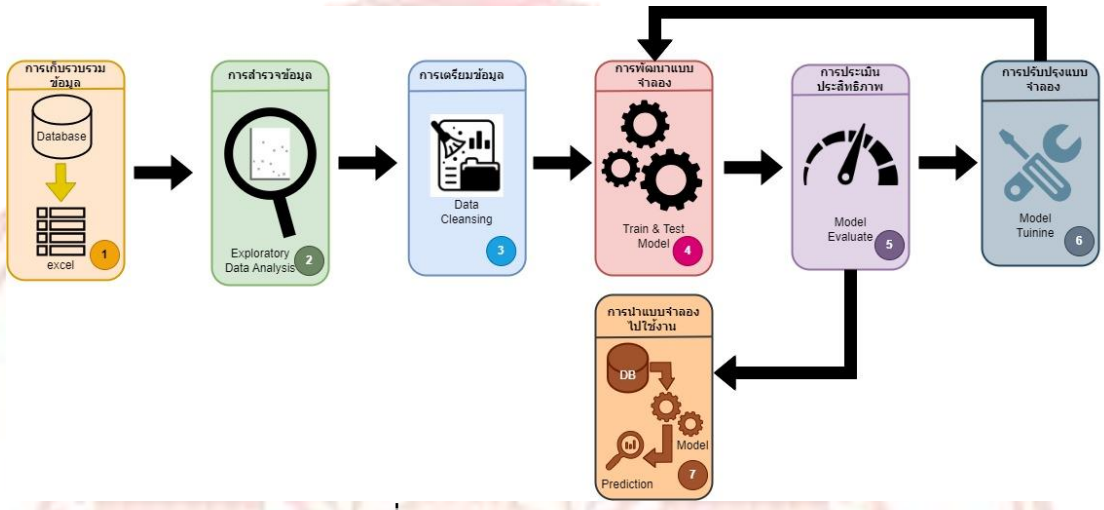
Prabaljeet Singh Saini, Atush Bhatnagar และ Lekha Rani (Prabaljeet Singh Saini, 2023) วิเคราะห์เปรียบเทียบอัลกอริทึมการจัดประเภทสำหรับการพยากรณ์การอนุมัติเงินกู้ โดยพบว่า Random Forest เป็นโมเดลที่มีความแม่นยำสูงสุดเมื่อเทียบกับ KNN, SVC และ Logistic Regression

Puneeth B. R และคณะ (Puneeth B. R, 2022) นำเสนอแนวทางการพยากรณ์ความเหมาะสมของการขอสินเชื่อโดยใช้อัลกอริทึม Logistic Regression, Decision Trees และ Random Forest โดยโมเดล Random Forest แสดงประสิทธิภาพสูงที่สุดจากการประเมิน

JiXiang Niu (JiXiang Niu, 2022) ได้เสนอการวิจัยเกี่ยวกับการพยากรณ์สินเชื่อโดยใช้แนวคิดของ Machine Learning ที่สามารถตีความได้ (interpretable ML) โดยใช้อัลกอริทึม XGBoost ซึ่งให้ผลการทำนายที่แม่นยำและสามารถอธิบายได้อย่างโปร่งใส

บทที่ 3
วิธีดำเนินการวิจัย

3.1 กรอบแนวคิดของงานวิจัย



ภาพที่ 3-1 กรอบแนวคิดของงานวิจัย

จากภาพที่ 3-1 ขั้นตอนแรกของการเริ่มดำเนินงานวิจัยคือการเก็บรวมข้อมูล (Collection data) ผู้สมัครบัตรเครดิตและผ่านการพิจารณาแล้วจากฐานข้อมูลมาในรูปแบบ CSV และเรียกข้อมูลข้อมูลมาจะอยู่ในรูปแบบดาต้าเฟรม (Data frame) ขั้นตอนต่อมานำข้อมูลที่ได้มาทำการสำรวจข้อมูล (Exploratory data analysis) ดูลักษณะของข้อมูล และจึงทำการทำการการเตรียมข้อมูล (Data Preparation) โดยการความสะอาดข้อมูล (Cleansing Data) ในการทำความสะอาดข้อมูลนั้นต้องดูว่าข้อมูลมีค่าว่างหรือขาดหายหรือไม่ ถ้าหากพบข้อมูลที่เป็นค่าว่างหรือขาดหายจะใช้ การเติมข้อมูล เมื่อทำการทำความสะอาดข้อมูลเรียบร้อยแล้วจึงมาแสดงเป็นกราฟเพื่อศึกษาแนวโน้ม ทิศทางของชุดข้อมูล จึงมาทำ Feature Engineering เพื่อหาตัวแปรที่เหมาะสมและมีความสัมพันธ์กับชุดข้อมูลเพื่อนำไปเป็นชุดข้อมูลให้ในการพัฒนาแบบจำลอง (Model Building) แบบต่างๆ ได้เรียนรู้ ชุดข้อมูลแบ่งออกเป็นชุดเรียนรู้และทดสอบโดยใช้ แบ่งข้อมูลแบบความเที่ยงตรง k กลุ่ม (k-Fold Cross Validation) โดยแบบข้อมูลออกเป็น 10 ชุด สำหรับให้แบบจำลองเรียนรู้และการทดสอบ ดำเนินการแก้ไขปัญหาข้อมูลที่ไม่สมดุลโดยใช้เทคนิค SMOTE ควบคู่กับการการประเมินประสิทธิภาพของแบบจำลอง (Model Evaluation) ถ้าหากผลการทดสอบไม่เป็นที่น่าพอใจจะต้องทำการการปรับปรุงประสิทธิภาพของแบบจำลอง (Model Improvement) ให้เหมาะสมเพื่อที่จะทำได้ผลการพยากรณ์ใกล้เคียงกับข้อมูลจริงมากที่สุด เมื่อได้แบบจำลองที่เหมาะสมแล้วจึงนำไป การพัฒนาแบบจำลอง (Model Building) และใช้ในการพยากรณ์ โดยการให้ข้อมูลปัจจุบัน

3.2 การเก็บรวบรวมข้อมูล (Collection data)

ผู้วิจัยได้ใช้ชุดข้อมูล (Dataset) ที่ทำการเก็บรวบรวมข้อมูลจากผู้สมัครที่ยื่นสมัครบัตรเครดิตและผ่านการพิจารณาแล้วจากหน่วยงานแห่งหนึ่งในชั้นกระบวนการทดสอบกระบวนการทดสอบซอฟต์แวร์โดยผู้ใช้งานจริง (User Acceptance Test) ของระบบพิจารณาอนุมัติบัตรเครดิต (Credit Card Approval Application) ย้อนหลังตั้งแต่ มกราคม 2564 - ตุลาคม 2566 ข้อความ เป็นไฟล์นามสกุล .csv เฉพาะประเภทบัตรเครดิตแพลตตินั่ม เกณฑ์รายได้ จำนวนข้อมูลทั้งสิ้น 1567 แถว โดยแบ่งออกเป็น 2 กลุ่ม ได้แก่ กลุ่มที่มีคุณสมบัติผ่านเกณฑ์พิจารณาอนุมัติบัตรเครดิต จำนวน 1,369 รายการ กลุ่มที่มีไม่มีคุณสมบัติผ่านเกณฑ์พิจารณาอนุมัติบัตรเครดิต จำนวน 198 รายการ โดยมีคุณสมบัติหรือคุณลักษณะของข้อมูล (feature) ทั้งหมด 32 คอลัมน์

3.3 การสำรวจข้อมูล (Exploratory data analysis)

หลังจากเก็บรวบรวมข้อมูล ขั้นตอนนี้เป็นการสำรวจข้อมูล โดยกระบวนการทางสถิติและการวิเคราะห์ข้อมูลที่ใช้เทคนิคต่าง ๆ เพื่อให้เข้าใจลึกซึ้งถึงลักษณะและคุณลักษณะของข้อมูล ในมุมต่างๆ ในทุกๆ ตัวแปร หรือเปรียบเทียบกันระหว่างตัวแปร ได้แก่ การตรวจสอบดูขนาดของข้อมูล ประเภทของข้อมูล ขอบเขตของข้อมูล ช่วยให้ค้นพบข้อมูลเชิงลึกที่ซ่อนอยู่ในข้อมูลและระบุปัญหาหรือข้อผิดพลาดในข้อมูลเพื่อทำการปรับปรุงและแก้ไขในขั้นตอนต่อไป เพื่อเพิ่มประสิทธิภาพในการพัฒนาแบบจำลอง

ตารางที่ 3-1 ข้อมูลที่ขาดหายไปแต่ละคอลัมน์

แต่ละคอลัมน์	จำนวนที่ขาด	แต่ละคอลัมน์	จำนวนที่ขาด	แต่ละคอลัมน์	จำนวนที่ขาด
V1_Nationality	0	V12_TotalExposureLimitAmt	417	V23_NCBAcctStatus	160
V2_Age	0	V13_TotalExposureLimitGMT	427	V24_NCBAcctType	1330
V3_TimeInJob	0	V14_MaxMainCard	22	V25_NCBOpenDateMonth	1499
V4_Income	0	V15_MaxSupCard	30	V26_NCBTDRDateMonth	1565
V5_DepRefType	1552	V16_DupCardType	43	V27_NCBOverdueCurrMonth	1558
V6_DepRefAmt	1552	V17_AML	56	V28_NCBDlq6m	1330
V7_ExistingDTI	427	V18_KYC	56	V29_NCBDlq12m	1330
V8_TotalDTI	427	V19_SLL	160	V30_NCBDlq24m	1330
V9_CreditLimit	20	V20_WriteOff	2	V31_NCBOverLimit6m	1330
V10_CreditLimitGMI	427	V21_Fraud	0	CALSS	0
V11_CreditLimitDepRef	1542	V22_Bankruptcy	160		

ตารางที่ 3-2 คุณลักษณะของชุดข้อมูล (Dataset)

ลำดับ	คุณลักษณะ	ชื่อตัวแปร	ชนิดของข้อมูล
1	สัญชาติ	V1_Nationality	TEXT
2	อายุ	V2_Age	INTEGER
3	อายุงานหรือระยะเวลาการประกอบธุรกิจ	V3_TimeInJob	INTEGER
4	รายได้ต่อเดือน	V4_Income	FLOAT
5	ประเภทอ้างอิงเงินฝากฯ	V5_DepRefType	INTEGER
6	จำนวนเงินฝากฯ	V6_DepRefAmt	FLOAT
7	DTI (ภาระหนี้เดิม)	V7_ExistingDTI	FLOAT
8	DTI (ภาระหนี้เดิม + ภาระครั้งนี้)	V8_TotalDTI	FLOAT
9	วงเงินตามเกณฑ์บัตรเครดิตที่สมัคร	V9_CreditLimit	FLOAT
10	จำนวนเท่าของวงเงินต่อรายได้	V10_CreditLimitGMI	FLOAT
11	% ของวงเงินต่อจำนวนเงินฝากฯ อ้างอิง	V11_CreditLimitDepRef	FLOAT
12	วงเงินรวมบัตรเครดิตและบัตรกดเงินสด	V12_TotalExposureLimitAmt	FLOAT
13	จำนวนเท่าของวงเงินรวมบัตรต่างๆ ต่อรายได้	V13_TotalExposureLimitGMT	FLOAT
14	จำนวนบัตรหลักที่มีในระบบ	V14_MaxMainCard	INTEGER
15	จำนวนบัตรเสริมที่มีในระบบ	V15_MaxSupCard	INTEGER
16	สมัครประเภทบัตรซ้ำ	V16_DupCardType	TEXT
17	AML (ปปง.)	V17_AML	INTEGER
18	KYC	V18_KYC	INTEGER
19	กำกับลูกหนี้รายใหญ่ (SLL)	V19_SLL	TEXT
20	Write off	V20_WriteOff	TEXT
21	ฐานข้อมูล Fraud	V21_Fraud	TEXT
22	ประวัติการถูกศาลดำเนินการบังคับคดี	V22_Bankruptcy	INTEGER
23	สถานะบัญชีใน NCB	V23_NCBAcctStatus	INTEGER
24	ประเภทบัญชีใน NCB	V24_NCBAcctType	INTEGER
25	วันที่เปิดบัญชี	V25_NCBOpenDateMonth	DATETIME
26	ประวัติการปรับโครงสร้างหนี้ใน NCB	V26_NCBTDRDateMonth	INTEGER
27	เกินวงเงินเดือนล่าสุดใน NCB	V27_NCBOverdueCurrMonth	INTEGER
28	ประวัติการชำระหนี้ NCB ย้อนหลัง 6 เดือน	V28_NCBDlq6m	INTEGER
29	ประวัติการชำระหนี้ NCB ย้อนหลัง 12 เดือน	V29_NCBDlq12m	INTEGER
30	ประวัติการชำระหนี้ NCB ย้อนหลัง 24 เดือน	V30_NCBDlq24m	INTEGER
31	ประวัติการใช้เกินวงเงิน NCB ย้อนหลัง 6 เดือน	V31_NCBOverLimit6m	INTEGER

3.4 การเตรียมข้อมูล (Data Preparation)

หลังจากการสำรวจ ทำความสะอาดข้อมูลและเก็บในรูปแบบของ .csv เนื่องจากข้อมูลที่ได้จากขั้นตอนเก็บรวบรวมเป็นข้อมูลดิบที่ดิบ (Raw Data) อาจมีข้อผิดพลาดหลายอย่าง เช่น ข้อมูลไม่ครบและประเภทของข้อมูลยังไม่เหมาะที่เอาไปใช้งานได้ เป็นต้น

3.4.1 วิธีการทำความสะอาด หลังจากสำรวจข้อมูลแล้วพบว่าข้อมูลยังไม่สมบูรณ์มีขั้นตอนดังนี้

3.4.1.1 ลบคุณลักษณะของข้อมูลฟีเจอร์ (feature) ที่ไม่จำเป็นออก

3.4.1.2 ลบข้อมูลที่ซ้ำออก เนื่องจากมีบางรายการที่มีข้อมูลซ้ำ

3.4.1.3 แปลงประเภทของข้อมูล เนื่องจากข้อมูลที่เป็นตัวอักษร ไม่อยู่ในรูปแบบที่สามารถ วิเคราะห์ได้ จึงต้องทำการแทนค่าข้อมูลให้อยู่ในรูปแบบที่สามารถวิเคราะห์ได้

3.4.1.4 การเติมข้อมูลที่ขาด หรือ ข้อมูลมีสิ่งรบกวน แก้ไขโดยการแทนค่าข้อมูลที่ถูกต้องไปแทนที่ข้อมูลเดิม

3.4.2 การนำเสนอข้อมูล (Data Visualizing) ในรูปแบบที่เข้าใจง่าย

3.4.3 การเลือกฟีเจอร์ (Feature Selection)

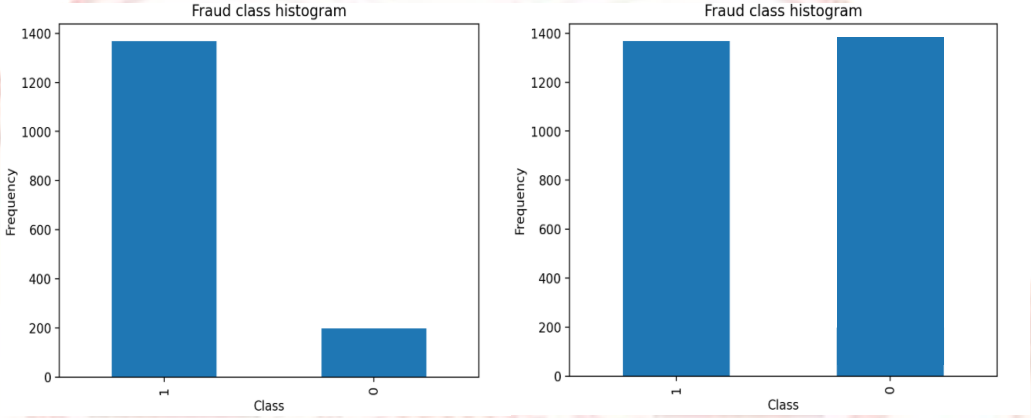
3.4.4 การจัดการข้อมูลให้เป็นระเบียบ (Data Normalizing)

3.5 การสร้างแบบจำลอง (Model Building)

3.5.1 ในงานวิจัยนี้มีการแบ่งชุดข้อมูล ข้อมูลพร้อมใช้งานสำหรับสร้างแบบจำลอง ชุดข้อมูลจำเป็นต้องแยกเป็นชุดสำหรับฝึกสอน (Training Set) และชุดสำหรับทดสอบ (Test Set) เพื่อให้แบบจำลองมีแนวโน้มที่จะทำนายผลลัพธ์ได้ อย่างถูกต้องโดยที่แบบจำลองไม่ได้เรียนรู้ข้อมูลมาก่อน และเลือกใช้วิธีสุ่มเลือกแบ่งข้อมูลแบบความเที่ยงตรง k กลุ่ม (k-Fold Cross Validation) โดยเริ่มจากการแบ่งชุดข้อมูลออกเป็นส่วนๆ ให้เท่าๆ กัน ต่อจากนั้นให้นำข้อมูลบางส่วนมาทำการ เรียนรู้และนำข้อมูลบางส่วนมาทำการทดสอบแบบจำลองที่ได้จากการเรียนรู้ โดยในการทำงานจะทำการเลือกสุ่มข้อมูล ออกเป็น k ชุดที่เท่าๆ กัน ในการทดลองครั้งแรก ข้อมูลชุดที่ 1 เป็นข้อมูลชุดทดสอบและข้อมูลชุดที่เหลือเป็นข้อมูลชุด เรียนรู้ ในการทดลองครั้งที่ 2 ข้อมูลชุดที่ 2 เป็นข้อมูลชุดทดสอบและข้อมูลชุดที่เหลือเป็นข้อมูลชุดเรียนรู้ ทำจนกระทั่ง ข้อมูลทุกชุดได้ถูกนำมาเป็นข้อมูลชุดทดสอบและข้อมูลชุดเรียนรู้ ซึ่งจะมีการทดลองทั้งหมด k ครั้ง ในงานวิจัยนี้ได้ เลือกใช้ค่า k = 10

3.5.2 การทำสมดุลข้อมูล เนื่องจากกลุ่มที่ผ่านเกณฑ์พิจารณาอนุมัติบัตรเครดิตมีมากกว่ากลุ่มที่ผ่านเกณฑ์พิจารณาอนุมัติบัตรเครดิตจำนวนมากกว่าด้านลบ ข้อมูลจึงเป็นข้อมูลที่ไม่สมดุล (Imbalanced) ทำให้ส่งผลต่อความแม่นยำการพยากรณ์ของแบบจำลอง (bias) เพื่อสร้างสมดุลให้ข้อมูลที่นำไปสร้างแบบจำลอง จึงใช้เทคนิคการสุ่มตัวอย่างชุดข้อมูลที่ไม่สมดุล SMOTE เป็นการจัดการกับปัญหาการจำแนกข้อมูลไม่สมดุล เพิ่มข้อมูลในคลาสขนาดเล็กโดยสังเคราะห์ข้อมูลใหม่จาก

ชุดข้อมูลเดิม ทำให้ข้อมูลในคลาสขนาดเล็กมีการกระจายตัวที่ดีขึ้น ช่วยให้ค่าความสำคัญของข้อมูลไม่สูญหาย และช่วยลดปัญหาการเกิด Overfitting ได้ เนื่องจากปกติทั่วไปหากอัตราส่วน 1:10 อาจไม่มีผลกระทบต่อประสิทธิภาพการทำนายข้อมูลมากนัก แต่ถ้าอัตราส่วนที่มีขนาด 1:100 หรือ 1:1000 อาจทำให้ข้อมูลเกิด Overfitting เนื่องจากคอมพิวเตอร์มองว่าข้อมูลประเภทที่น้อยกว่านั้นสามารถถูกมองข้ามได้ การทำสมดุลในกรณีนี้คือกลุ่มที่ไม่ผ่านเกณฑ์พิจารณาอนุมัติบัตรเครดิตให้ใกล้เคียงกับกลุ่มที่ผ่านเกณฑ์พิจารณาอนุมัติบัตรเครดิต



(ก) ทำสมดุลข้อมูล (ข) หลังทำสมดุลข้อมูล

ภาพที่ 3-2 ทำสมดุลข้อมูล

3.5.3 อัลกอริทึม (Algorithms)

จากการศึกษาพบว่าการใช้แบบจำลอง ได้แก่ Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM) , K-nearest Neighbors (KNN), Ensemble, XG Boost, Gradient Boosting Neuron Network และ, Naive Bayes เป็นที่นิยมในการใช้จำแนกประเภท (Classification) โดยประสิทธิภาพ ดังนั้นแบบจำลองที่จะนำมาใช้เปรียบเทียบจะประกอบด้วย แบบจำลองถดถอยโลจิสติก Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM) , K-nearest Neighbors (KNN), Ensemble, XG Boost, Gradient Boosting, Neuron Network และ, Naive Bayes ซึ่งทำการเปรียบเทียบแบบจำลองทั้งสิบแบบจำลอง โดยใช้ชุดข้อมูลและตัวชี้วัดแบบจำลองตัวเดียวกัน

3.6 การประเมินประสิทธิภาพของแบบจำลอง (Model Evaluation)

ในงานวิจัยนี้ใช้การทดสอบแบบไขว้ทบ (k - Fold Cross Validation) แบบ 10 ส่วน แล้วทำการทดสอบเพื่อประเมินประสิทธิภาพ โดยในการประเมินแบบจำลองใช้ Confusion Matrix และ Classification Report จากไลบรารี่ Scikit-learn ในการตรวจสอบความถูกต้องระหว่างผลลัพธ์ที่แบบจำลองทำนาย และผลลัพธ์จากชุดข้อมูลจริง ด้วยหลักเกณฑ์ได้แก่ ค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และ F1-Score เพื่อเปรียบเทียบหาเทคนิคการจำแนกประเภทข้อความ ที่เหมาะสมกับที่สุด 1 แบบจำลอง

3.7 การปรับปรุงประสิทธิภาพของแบบจำลอง (Model Improvement)

ในการสร้างแบบจะลองใช้ค่าเริ่มต้นตามพารามิเตอร์ที่แบบจำลองกำหนดมา กรณีแบบจำลองไม่สามารถทำนายผลที่มีความแม่นยำมากว่า 70% จะทำการทดสอบปรับค่าพารามิเตอร์ต่าง ๆ ของแต่ละแบบจำลอง เพื่อให้ได้แบบจำลองแต่ละประเภทที่ดีที่สุด

3.8 การนำแบบจำลองไปใช้งาน (Model Deployment)

เมื่อได้แบบจำลองที่มีความแม่นยำมากที่สุดโดยการเปรียบแบบ 7 แบบจำลอง จึงนำแบบจำลองที่ดีที่สุดมาใช้ในการพยากรณ์ ไปพัฒนาระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต (Credit Card approval Application) พัฒนาระบบโดยไม่มีเก็บข้อมูลไว้ในฐานข้อมูล (Data Base) มีองค์ประกอบของฮาร์ดแวร์(Hardware) ซอฟต์แวร์ (Software) และเครื่องมือ (Tool) ที่ต้องนำมาใช้พัฒนาดังนี้ดังตารางที่ 3-1

ตารางที่ 3-3 ฮาร์ดแวร์และระบบปฏิบัติการในการพัฒนาระบบ

คอมพิวเตอร์	เวอร์ชั่น
ระบบปฏิบัติการ (Operation system)	Windows 10 (64 bit)
หน่วยประมวลผล (CPU)	I7-7600kU
หน่วยความจำหลัก (RAM)	16 GB
หน่วยประมวลผลภาพ (GPU)	Intel HD Graphics 620

ตารางที่ 3-4 เครื่องมือ (Tools) และเทคโนโลยีในการพัฒนาระบบ

เครื่องมือ และเทคโนโลยี	รายละเอียด	เวอร์ชั่น
Program Python	ภาษาการพัฒนาโปรแกรมของหลังบ้านและพัฒนาโมเดล	3.13.2
JavaScript Html และ CSS	ภาษาการพัฒนาโปรแกรมของหน้าบ้าน	-
Program Visual studio code	เครื่องมือในการพัฒนาโปรแกรม	6.4.6
Framework Fast API (Backend)	เฟรมเวิร์คพัฒนาโปรแกรมในส่วนของหลังบ้าน	0.115.11
Framework React (frontend)	เฟรมเวิร์คพัฒนาโปรแกรมในส่วนของหน้าบ้าน	19.0.0
Library Material-ui	ไลท์บรารีในส่วนของโปรแกรมหน้าบ้าน	6.4.6
Library Nodejs	ไลท์บรารีในส่วนของโปรแกรมหน้าบ้าน	22.14.0
Framework Bootstrap	เฟรมเวิร์คพัฒนาโปรแกรมในส่วนของหน้าบ้าน	5.3.3
Build tool Vite	เครื่องมือสร้างโครงสร้างของโปรแกรมหน้าบ้าน	6.2.0
Google Colab	เครื่องมือในการพัฒนาโมเดล	3.10
Postman	ทดสอบเส้น API	11.35.2

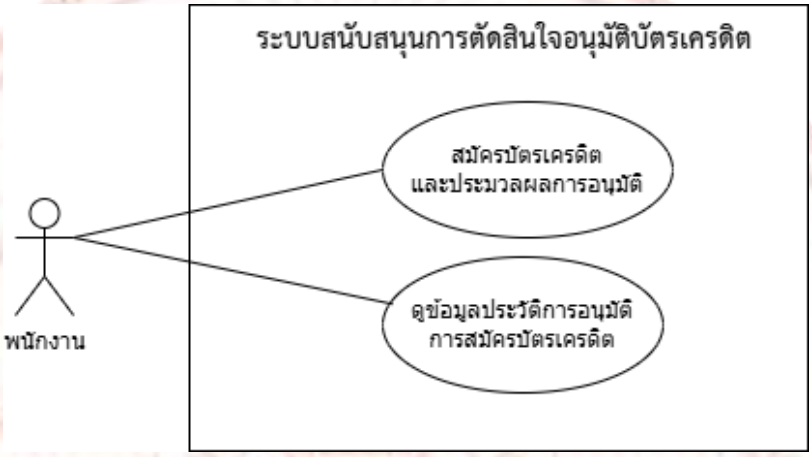
3.8.1 วิธีการดำเนินงานพัฒนาระบบ

หลังจากที่ผู้วิจัยได้ที่ได้สร้างแบบจำลองแล้วจะได้ไฟล์ แบบจำลองนามสกุล pkl ผู้วิจัยจะทำการพัฒนาระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ในรูปแบบของเว็บแอปพลิเคชัน (Web Application) ด้วยพัฒนาโปรแกรมในส่วนของหลังบ้าน (Backend) ด้วยเฟรมเวิร์ค Fast API โดยใช้ภาษา Python และในส่วนของหน้าหน้าบ้าน (frontend) ด้วยเฟรมเวิร์ค React โดยใช้ภาษา JavaScript โดยการเมื่อเจ้าหน้าที่ทำการกรอกข้อมูลคุณสมบัติของผู้สมัครบัตรเคดิต แล้วทำการกดเพื่อประมวลผลว่าคุณสมบัติที่กรอกเข้ามานั้นผ่านการพิจารณาหรือไม่ โดยจะแสดงผลการพิจารณาผ่านทางหน้าจอหลังจากประมวลผลแล้ว ถ้าผ่านจะขึ้นว่า “Approve” ถ้ำม่ผ่านจะขึ้นว่า “Reject” สามารถดูวิธีการสร้างแอพพลิเคชั่นเพิ่มเติมที่ (ภาคผนวก ข)

3.8.1 การวิเคราะห์ระบบ

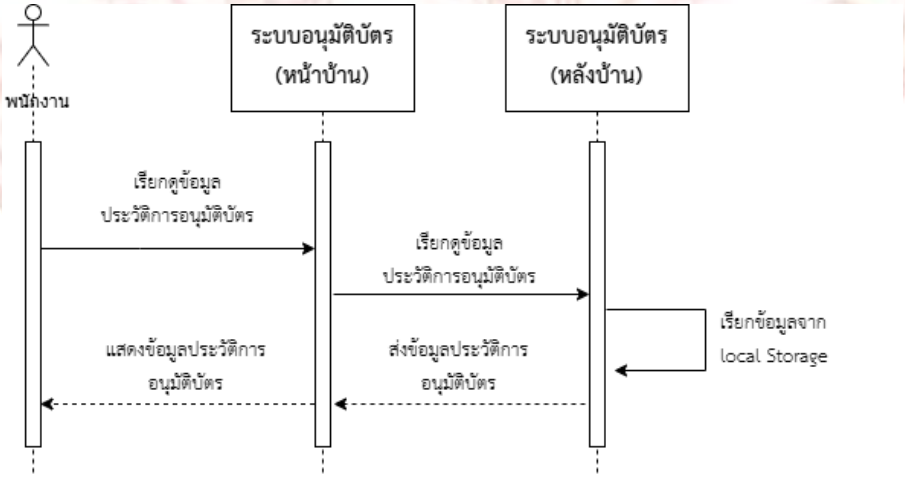
จากวิธีการดำเนินงานพัฒนาระบบ มาทำการวิเคราะห์และวางแผนปฏิบัติงานเพื่อทำการออกแบบโปรแกรมหลังบ้าน และแอปพลิเคชัน เพื่อดำเนินการพัฒนาระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ในการนำแบบจำลองที่ดีสุดมาใช้ในการอนุมัติบัตรเครดิต ดังนี้

3.9.1.1 Use Case Diagram

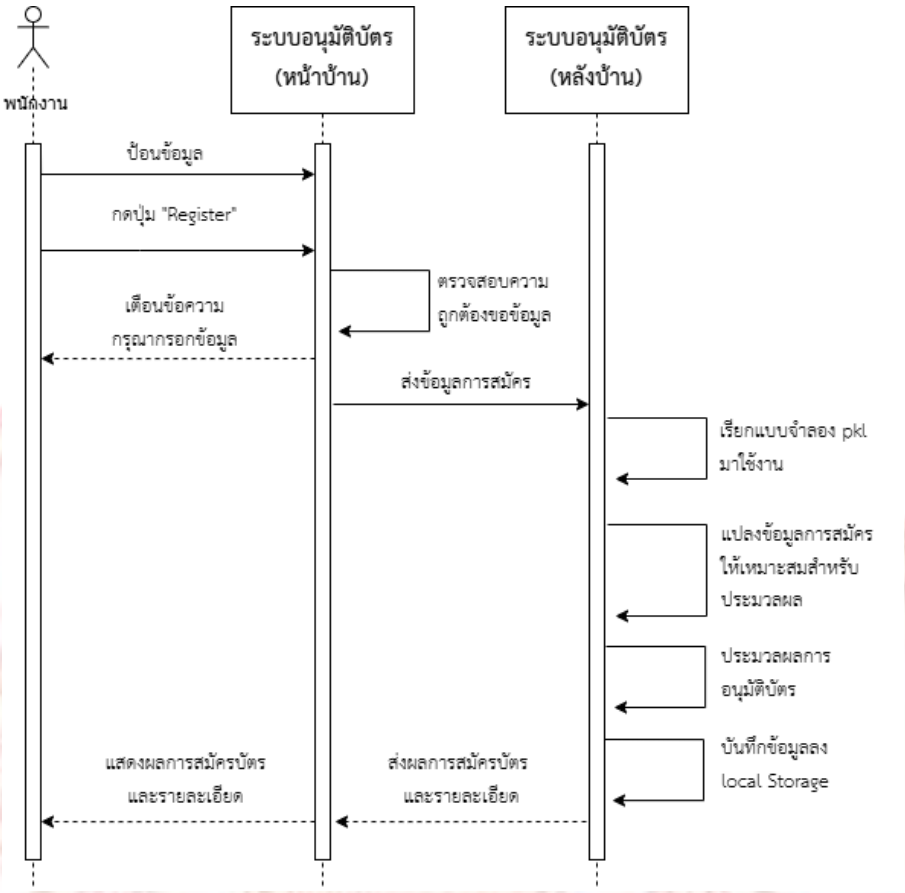


ภาพที่ 3-3 Use Case Diagram

3.9.1.1 Sequence Diagram



ภาพที่ 3-4 Sequence Diagram ดูข้อมูลประวัติการอนุมัติการสมัครบัตรเครดิต



ภาพที่ 3-4 Sequence Diagram สมัครบัตรเครดิตและประมวลผลการอนุมัติ

บทที่ 4
ผลการดำเนินงานวิจัย

ส่วนนี้จะเป็นการสร้างแบบจำลองการเรียนรู้ของเครื่องได้แก่ Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM) , K-nearest Neighbors (KNN), Ensemble, XG Boost, Gradient Boosting Neuron Network และ, Naive Bayes และทดสอบประสิทธิภาพแบบจำลองการตัดสินใจอนุมัติบัตรเครดิตเป็น 2 ประเภท อนุมัติ (Approve) และไม่อนุมัติ (Reject) โดยกำหนดอัตราส่วนส่วนระหว่างชุดข้อมูลการฝึกและชุดข้อมูลทดสอบคือ 80:20 โดยพารามิเตอร์ที่นำเข้ามาใช้ใน และการทดลองทั้งหมดถูกกำหนดให้เหมาะสม โดยจะพิจารณาจากการวัดค่าความถูกต้องแม่นยำ (Accuracy), F-measure (F1), ค่าความเที่ยงตรง (Precision) และค่าความไวหรือค่าระลึก (Recall) เป็นเกณฑ์ในการกำหนดประสิทธิภาพของแบบจำลอง นอกจากนั้นแล้วยังพิจารณาถึงเมทริกซ์ความสับสน (Confusion Matrix) รวมถึงเส้นโค้ง AUC-ROC ของแต่ละแบบจำแบบจำลอง โดยเปรียบเทียบการจำแนกของแบบจำลองที่ถูกฝึกสอนด้วยคุณลักษณะในรูปแบบต่าง ๆ ตามที่ผู้วิจัยได้ทำการออกแบบไว้ เพื่อหาแบบจำลองที่ดีสุดไปใช้พัฒนาระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต (Credit Card approval Application) ผลของการดำเนินงานสามารถอธิบายได้ดังต่อไปนี้

4.1 ผลการสร้างแบบจำลองและทดสอบประสิทธิภาพ

4.2 ผลการทดประสิทธิภาพแบบจำลองจากค่าความแม่นยำ (Accuracy)

4.3 ผลการทดสอบประสิทธิภาพแบบจำลอง Confusion Matrix และ AUC-ROC

4.4 ผลการนำแบบจำลองไปประยุกต์ใช้

4.1 ผลการสร้างแบบจำลองและทดสอบประสิทธิภาพ

ตารางที่ 4-1 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง Logistic Regression

	precision	recall	F1-score	Support
Class 0	0.25	0.72	0.38	18
Class 1	0.95	0.72	0.82	138
accuracy			0.72	156
macro avg	0.60	0.72	0.60	156
weighted avg	0.87	0.72	0.77	156

ตารางที่ 4-2 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง Decision Tree

	precision	recall	F1-score	Support
Class 0	0.68	0.94	0.79	18
Class 1	0.99	0.94	0.97	138
accuracy			0.94	156
macro avg	0.84	0.94	0.88	156
weighted avg	0.96	0.94	0.95	156

ตารางที่ 4-3 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง Random Forest

	precision	recall	F1-score	Support
Class 0	0.81	0.94	0.84	18
Class 1	0.99	0.97	0.98	138
accuracy			0.97	156
macro avg	0.90	0.96	0.93	156
weighted avg	0.97	0.97	0.97	156

ตารางที่ 4-4 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง Support Vector Machines

	precision	recall	F1-score	Support
Class 0	0.27	0.78	0.40	18
Class 1	0.96	0.72	0.83	138
accuracy			0.73	156
macro avg	0.62	0.75	0.61	156
weighted avg	0.88	0.73	0.78	156

ตารางที่ 4-5 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง K-nearest Neighbors (KNN)

	precision	recall	F1-score	Support
Class 0	0.34	0.89	0.49	18
Class 1	0.98	0.78	0.87	138
accuracy			0.79	156
macro avg	0.66	0.83	0.68	156
weighted avg	0.91	0.79	0.82	156

ตารางที่ 4-6 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง XG Boost

	precision	recall	F1-score	Support
Class 0	0.89	0.94	0.92	18
Class 1	0.99	0.99	0.99	138
accuracy			0.98	156
macro avg	0.94	0.96	0.95	156
weighted avg	0.98	0.98	0.98	156

ตารางที่ 4-7 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง Ensemble

	precision	recall	F1-score	Support
Class 0	0.81	0.94	0.87	18
Class 1	0.99	0.97	0.98	138
accuracy			0.97	156
macro avg	0.90	0.96	0.93	156
weighted avg	0.97	0.97	0.97	156

ตารางที่ 4-8 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง Gradient Boosting

	precision	recall	F1-score	Support
Class 0	0.36	0.72	0.48	18
Class 1	0.96	0.83	0.89	138
accuracy			0.82	156
macro avg	0.66	0.78	0.69	156
weighted avg	0.89	0.82	0.84	156

ตารางที่ 4-9 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง Neuron Network

	precision	recall	F1-score	Support
Class 0	0.12	1.00	0.21	18
Class 1	0.00	0.00	0.00	138
accuracy			0.12	156
macro avg	0.06	0.50	0.10	156
weighted avg	0.01	0.12	0.02	156

ตารางที่ 4-10 ความแม่นยำ Recall และ F1 Score ของแบบจำลอง naive bayes

	precision	recall	F1-score	Support
Class 0	0.12	0.94	0.22	18
Class 1	0.95	0.14	0.24	138
accuracy			0.23	156
macro avg	0.54	0.54	0.23	156
weighted avg	0.85	0.23	0.24	156

จากผลการทดสอบจากตารางที่ 4.1 - ตารางที่ 4.8 พบว่าความแม่นยำค่าเฉลี่ยถ่วงน้ำหนัก (Macro average) ของแบบจำลอง XG Boost นั้นมีค่าสูงสุดที่ 0.95 รองลงมาคือแบบจำลอง Ensemble และ Random Forest มีค่า 0.93 แบบจำลอง Decision Tree มีค่า 0.88 แบบจำลอง K-nearest Neighbors มีค่า 0.68 แบบจำลอง Support Vector Machines มีค่า 0.61 แบบจำลอง Logistic Regression มีค่า 0.60 แบบจำลอง Naive Bayes มีค่า 0.23 และแบบจำลอง Neuron Network มีค่า 0.10 ลดลงตามลำดับ ดังนั้นผลการทดสอบดังกล่าวแบบจำลอง XG Boost เป็นแบบจำลองที่มีประสิทธิภาพในการพยากรณ์มากในการจำแนกข้อมูล เนื่องจากในภาพรวม ความแม่นยำ (precision) Recall มีค่าสูงที่สุด ส่งผลให้ค่า F1-Score มีค่าสูงตามไปด้วย ซึ่ง F1-Score ถือเป็นตัวชี้วัดอีกตัวที่ต้องประกอบควบคู่กับค่าความถูกต้อง (Accuracy) โดยเป็นค่าเฉลี่ยถ่วงน้ำหนักระหว่าง Precision และ Recall โดยมียิ่งสูงยิ่งดี ด้วยเหตุนี้แล้วคะแนน F1 จึงมีความสำคัญและถูกนำมาใช้ในการประเมิน ประสิทธิภาพโดยรวมของแบบจำลองการเรียนรู้ของเครื่องรองจากค่าความถูกต้อง (Accuracy)

4.2 ผลการทดประสิทธิภาพแบบจำลองจากค่าความแม่นยำ (Accuracy)

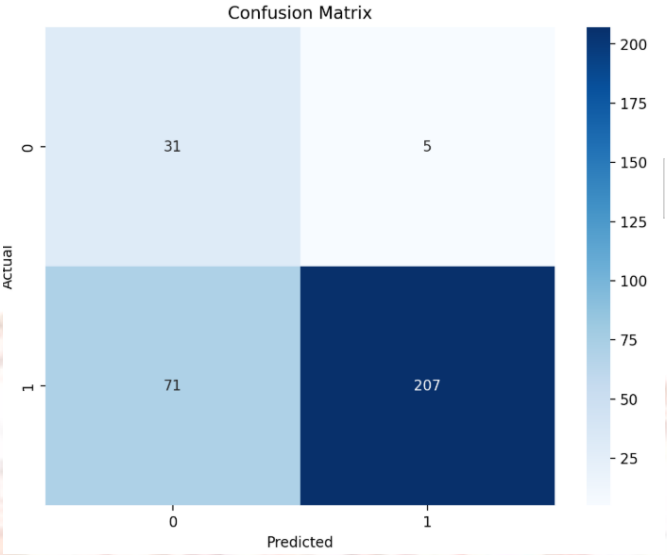
ตารางที่ 4-11 เปรียบเทียบค่าเปอร์เซ็นต์ความถูกต้องของแต่ละแบบจำลอง

แบบจำลอง	% Accuracy
XG Boost	98.0%
ensemble	96.7%
Random Forest	96.7%
Decision Tree	94.2%
Gradient Boosting	82.0%
K-nearest Neighbors	78.8%
Support Vector Machines	73.0%
Logistic Regression	72.4%
Naive Bayes	23.0%
Neuron Network	11.5%

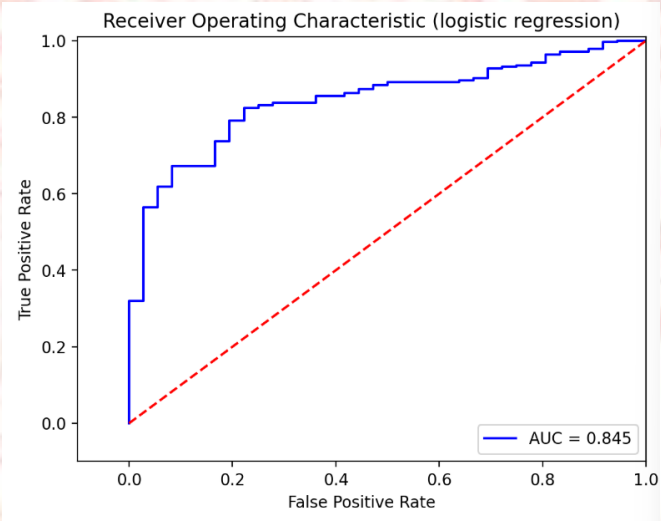
การวัดประสิทธิภาพแบบจำลองที่นำมาใช้พิจารณาคือเปอร์เซ็นต์ความถูกต้องหรือ ความแม่นยำ (% Accuracy) ในการทำนายผลของแบบจำลอง ซึ่งแสดงถึงความถูกต้องของการ ทำนายได้ตรงกับความเป็นจริงมากที่สุด หลังจากที่สร้างแบบจำลองและทดสอบประสิทธิภาพแบบจำลองการตัดสินใจอนุมัติบัตรเครดิตแล้ว การทดสอบพบว่าแบบจำลองที่มีเปอร์เซ็นต์ความถูกต้องมากที่สุดคือแบบจำลอง XG Boost ให้ค่าความถูกต้อง 98.0% รองลงมา Ensemble และ Random Forest ให้ค่า ความถูกต้อง 96.7% Decision Tree ให้ค่าควาถูกต้อง 94.2% สูงสุด Gradient Boosting ให้ค่าความถูกต้อง 82.0% K-Nearest Neighbors ให้ค่าความถูกต้อง 78.8% Support Vector Machine ให้ค่าความถูกต้อง 73.0% Logistic Regression ให้ค่าควาถูกต้อง 72.4% Naive Bayes ให้ค่าควาถูกต้อง 23.0% และ Neuron Network ให้ค่าควาถูกต้อง 11.5% ลดลงตามลำดับ ดังนั้นแบบจำลอง XG Boost มีความแม่นยำมากกว่าแบบจำลองรุ่น อื่นๆ ที่นำมาเปรียบเทียบกัน ค่าเปอร์เซ็นต์ความถูกต้องของแต่ละแบบจำลอง แสดงดังตารางที่ 4.8 งานวิจัยนี้ใช้ตัวชี้วัดจากค่าความถูกต้อง (Accuracy)

4.3 ผลการทดสอบประสิทธิภาพแบบจำลอง Confusion Matrix และ AUC-ROC

รูปที่ 4-1 - รูปที่ 4-8 แสดงเมทริกซ์ความสับสน (Confusion Matrix) ในรูปแบบฟังก์ชันแผนที่ความร้อน (heatmap function) และเส้นโค้ง AUC-ROC พบว่าสำหรับแบบจำลอง ensemble นั้นยังคงมีประสิทธิภาพที่เหนือกว่า แบบจำลองจากการเรียนรู้ของเครื่องอื่นๆ เนื่องจากในการแก้ปัญหาการแยกกลุ่มข้อมูล เนื่องจากเมทริกซ์ ความสับสนมีการเน้น (Highlight) หรือความเข้มของสีน้ำเงินเข้ม (ค่าสูง) ในแนวทแยงจากบนซ้ายไป-ล่างขวาทั้งหมด แบบจำลอง XG Boost มีค่า True Positive (TP) มีค่าสูงที่สุด และ True Negative (TN) มีค่าสูงที่สุด ส่งผลให้โมเดลทำนายถูกต้องมากขึ้น ส่วนค่า False Positive (FP) มีค่าต่ำแสดงถึงการลดการทำนายผิดพลาดว่าเป็นบวกหรือหมายถึงไม่ให้คนที่ไม่ควรได้บัตรเครดิตผ่านการอนุมัติ และสุดท้าย False Negative (FN) มีค่าต่ำที่สุดแสดงถึงลดการทำนายผิดพลาดว่าเป็นบวกหรือหมายถึงไม่ให้คนที่ไม่ควรได้บัตรเครดิตผ่านการอนุมัติ และ ค่า AUC ของทุกแบบจำลองมีค่ามากกว่า 0.5 แสดงให้เห็นถึงการแยกประเภทแต่ละกลุ่มออกจากกันได้อย่างชัดเจน ค่า AUC สูงที่สุดคือแบบจำลอง XG Boost มีค่า 0.970 ลองลงมา Gradient Boosting มีค่า 0.962 random forest มีค่า 0.891 Ensemble และ Decision tree มีค่าเท่ากันที่ 0.851 K-Neighbors Classifier มีค่า 0.892 Logistic regression มีค่า 0.845 Support Vector Machine มีค่า 0.778 Neural Network มีค่า 0.889 และ Naive Bayes มีค่า 0.554 น้อยสุด ลดลงตามลำดับ ส่งผลให้แบบจำลอง XG Boost มีอยู่สูงสุดที่ แต่เมื่อประกอบกับพิจารณาโดยโดยรวมแล้ว แบบจำลอง XG Boost ค่าความแม่นยำถูกต้อง (Accuracy) และค่าความแม่นยำ (Precision) มากกว่าแบบจำลองรุ่น อื่นๆ

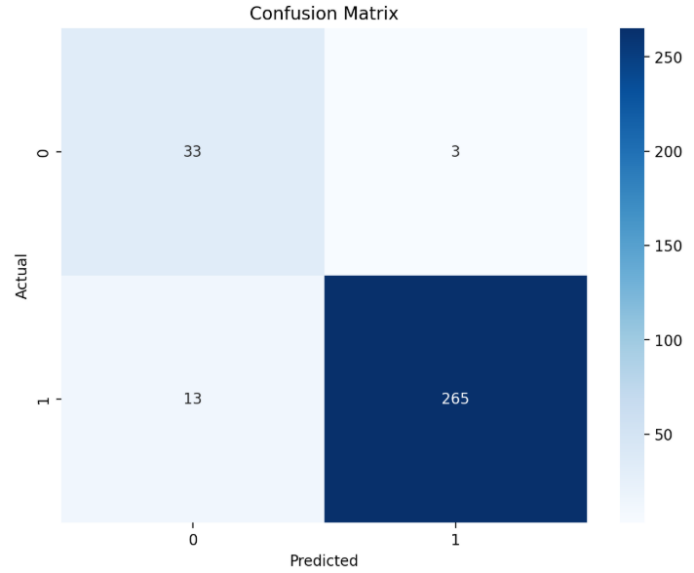


(ก) เมตริกซ์ความสับสน

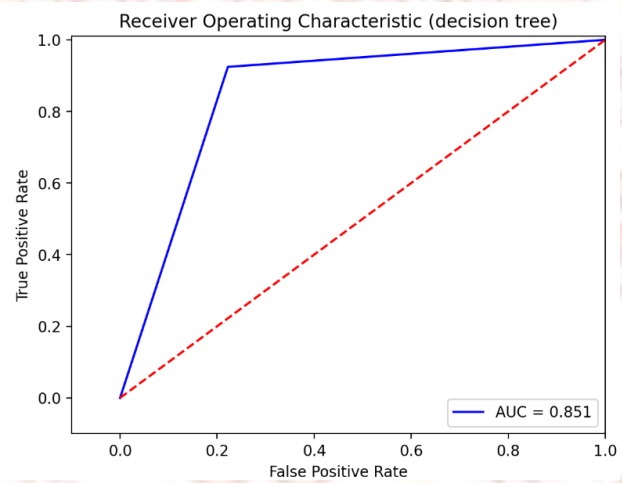


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-1 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ Logistic Regression

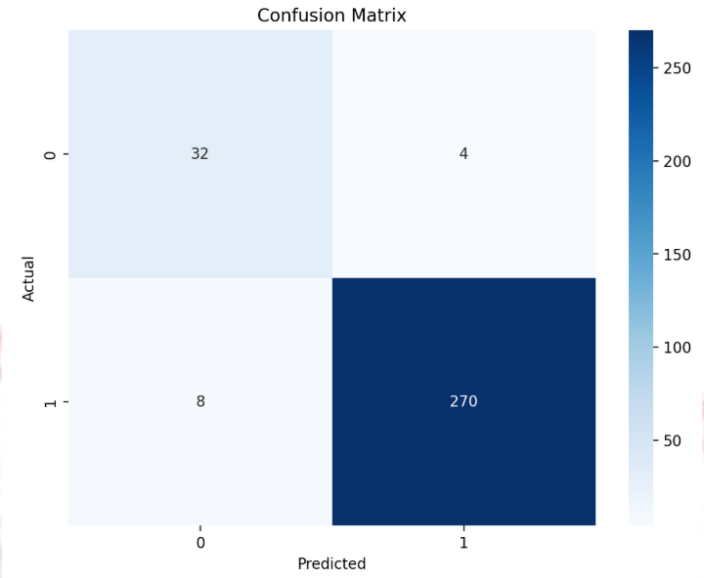


(ก) เมตริกซ์ความสับสน

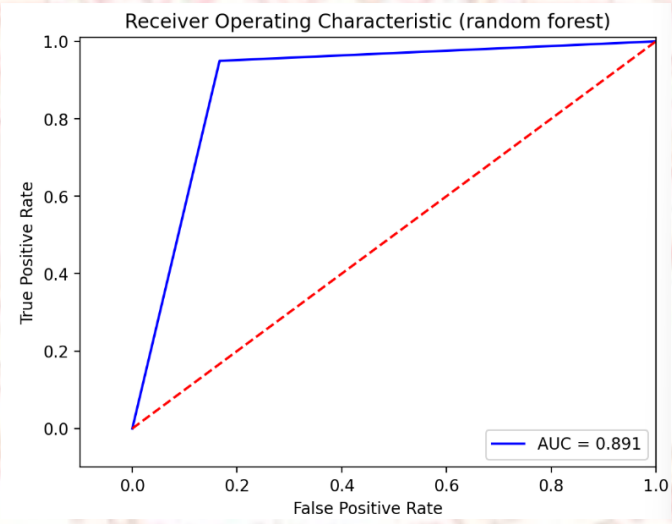


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-2 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ Decision Tree

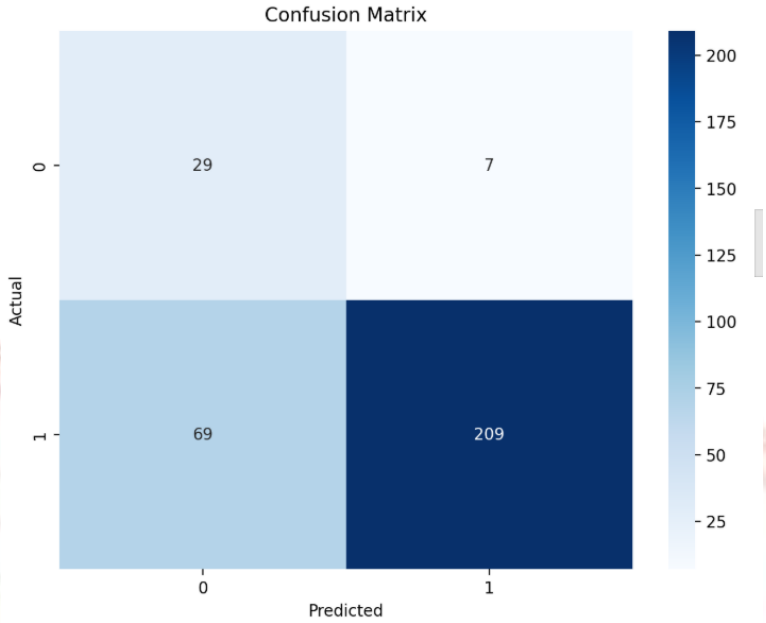


(ก) เมตริกซ์ความสับสน

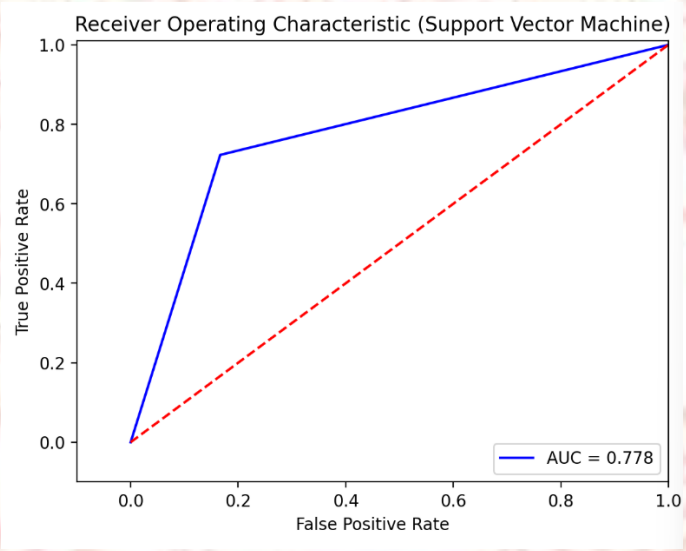


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-3 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ Random Forest

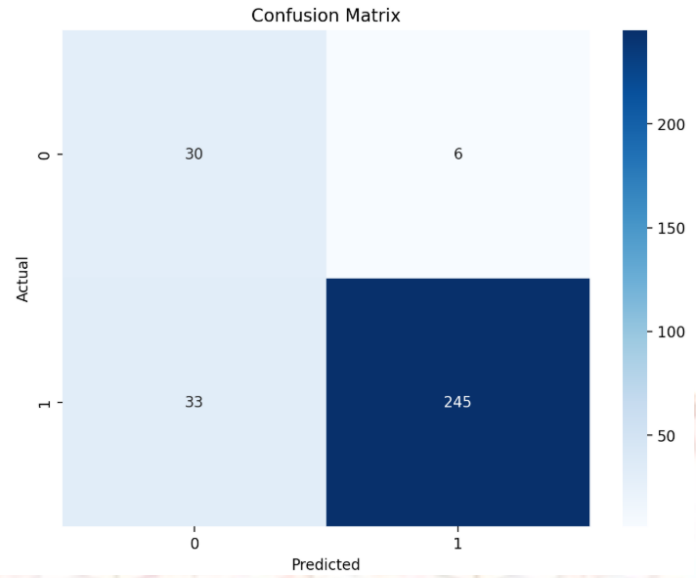


(ก) เมตริกซ์ความสับสน

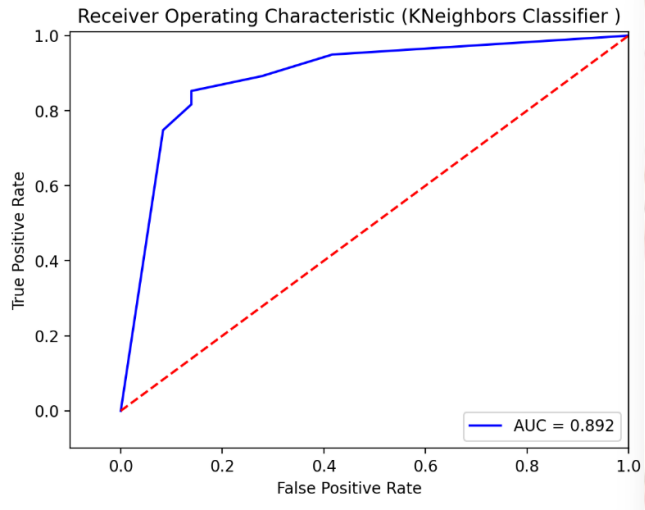


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-4 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ Support Vector Machines

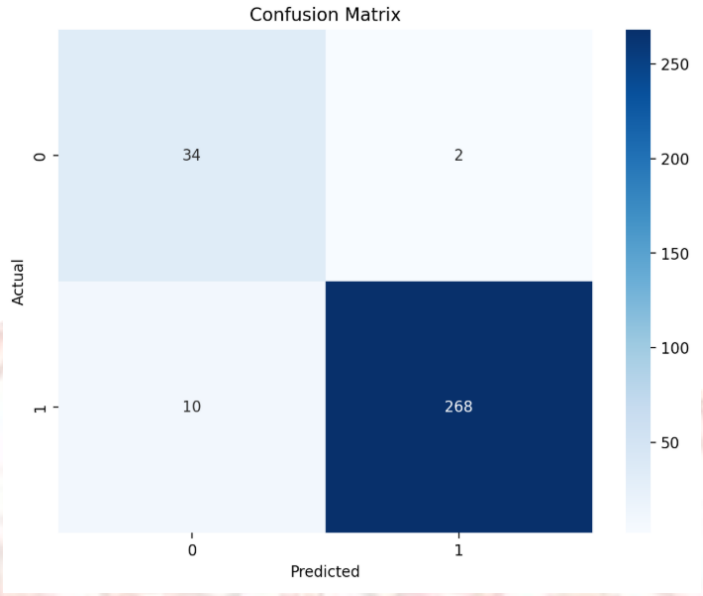


(ก) เมตริกซ์ความสับสน

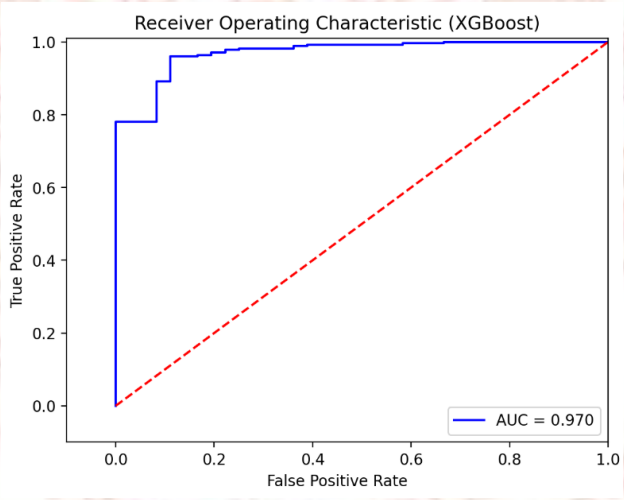


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-5 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ K-nearest Neighbors

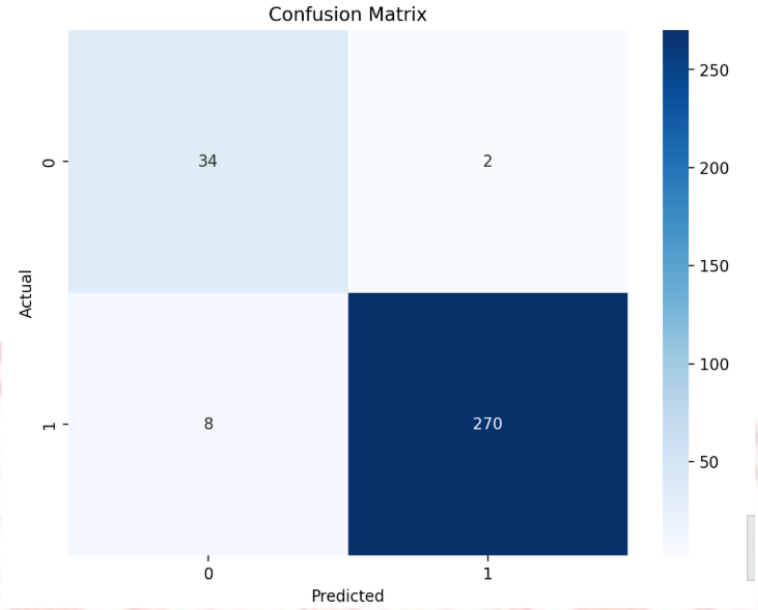


(ก) เมตริกซ์ความสับสน

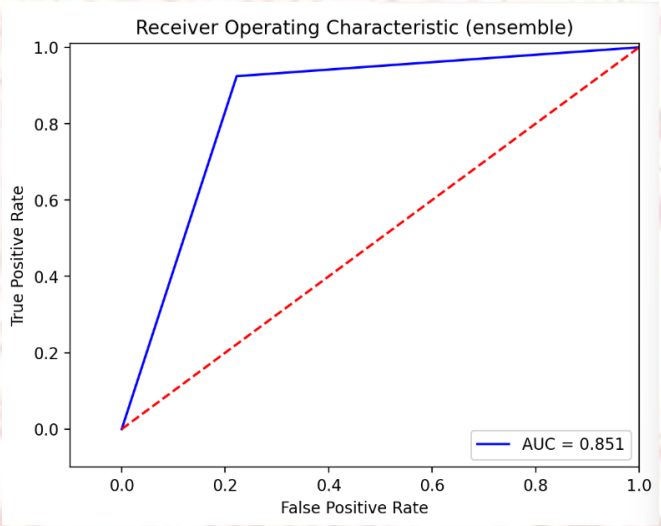


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-6 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ XG Boost

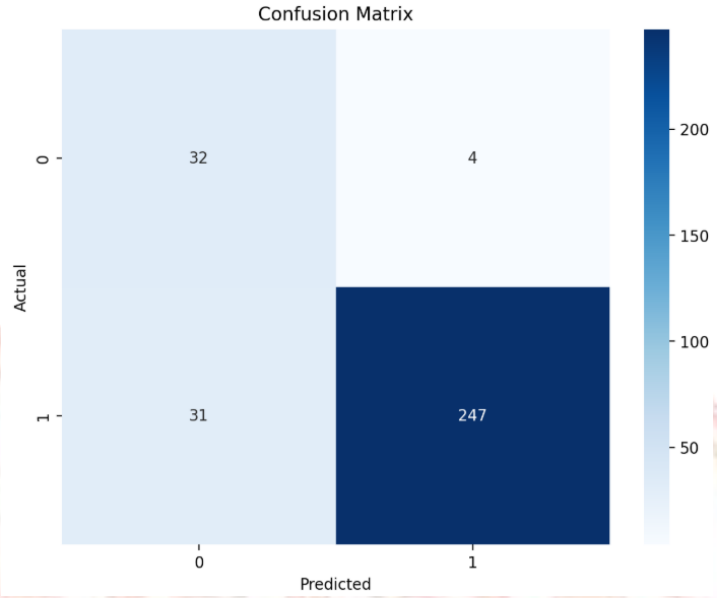


(ก) เมตริกซ์ความสับสน

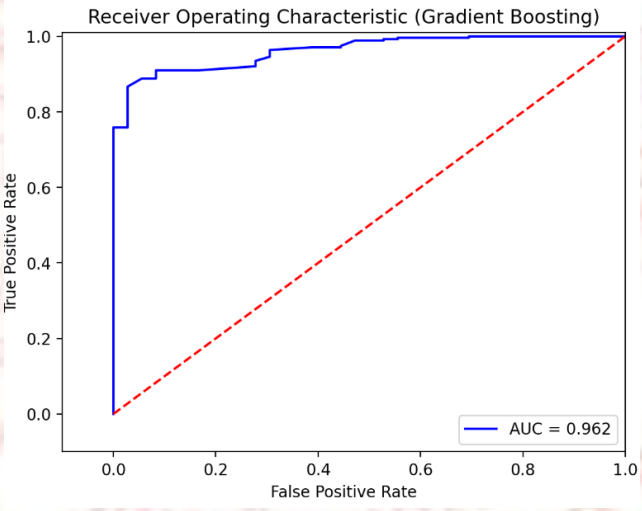


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-7 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ Ensemble

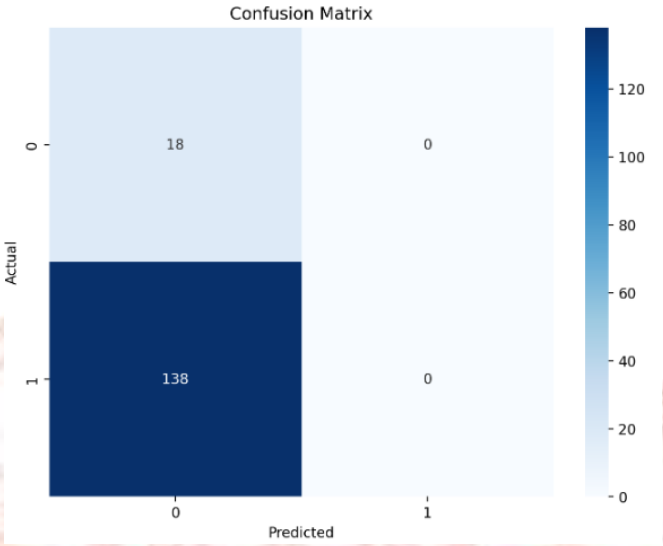


(ก) เมตริกซ์ความสับสน

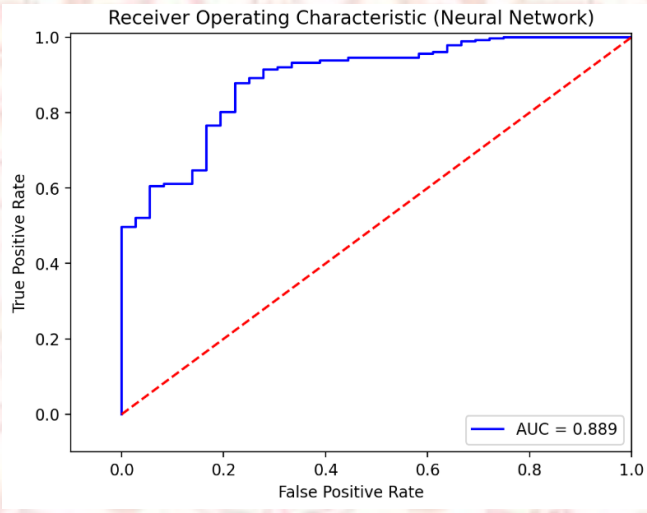


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-8 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ Gradient Boosting

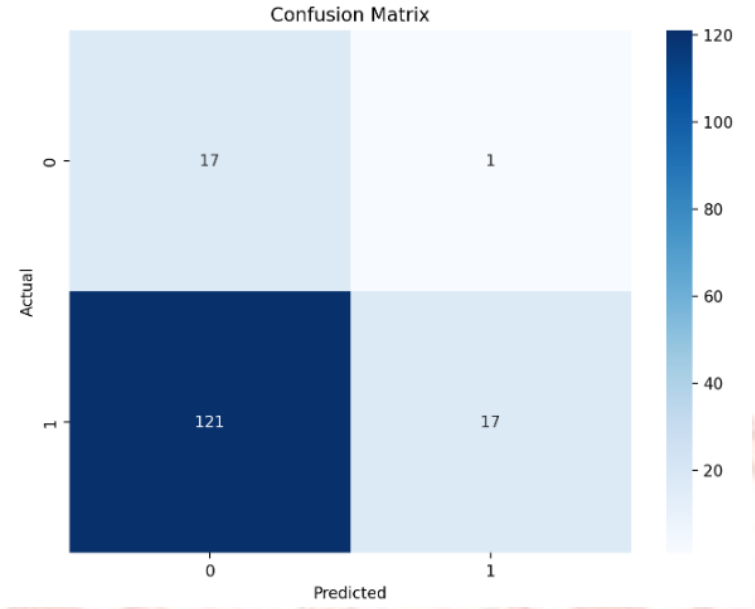


(ก) เมตริกซ์ความสับสน

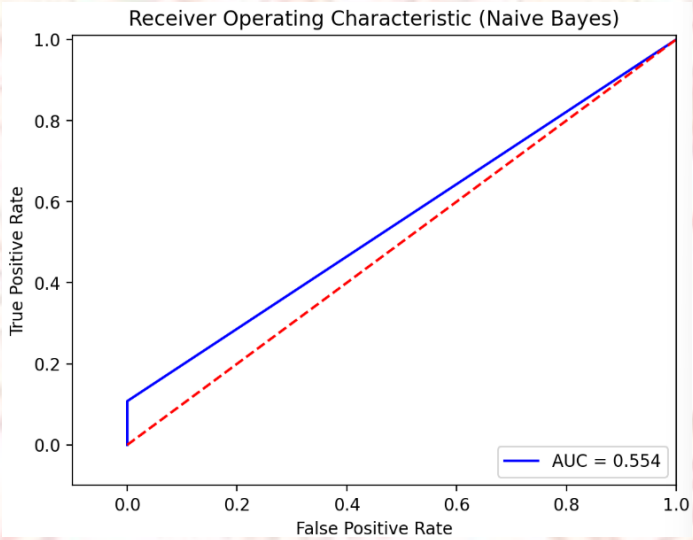


(ข) เส้นโค้ง AUC-ROC

ภาพที่ 4-9 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ Neuron Network



(ก) เมตริกซ์ความสับสน



(ข) เส้นโค้ง AUC-ROC

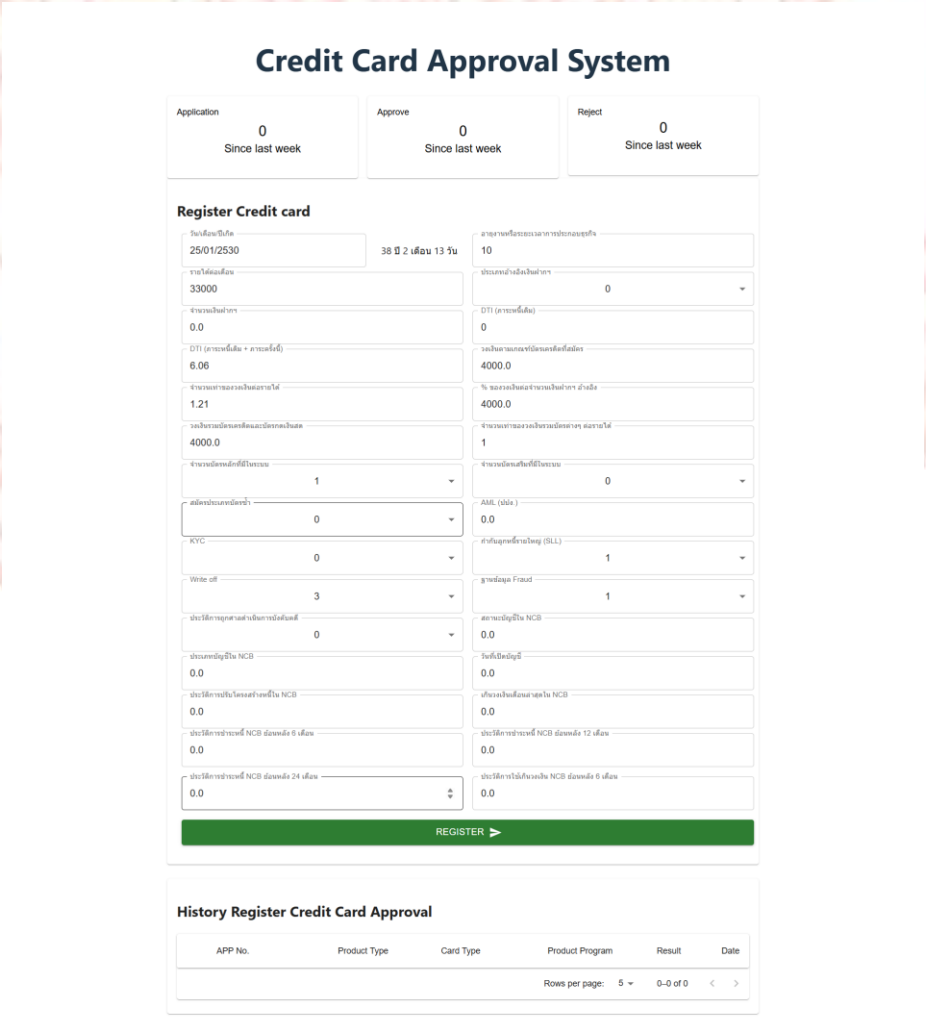
ภาพที่ 4-10 เมตริกซ์ความสับสนและเส้นโค้ง AUC-ROC ของ naive bayes

4.4 ผลการนำแบบจำลองไปประยุกต์ใช้

ขั้นตอนการพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ได้นำแบบจำลองที่ดีที่สุดก็คือ Ensemble ชื่อไฟล์ model.pkl นามสกุล .pkl ซึ่งได้ทำการฝึกอบรมและปรับแต่งค่าจนได้ความพึงพอใจของผู้วิจัยแล้ว นำพัฒนาเว็บแอพพลิเคชัน ด้วยพัฒนาโปรแกรมในส่วนของหลังบ้าน (Backend) ด้วยเฟรมเวิร์ค Fast API โดยใช้ภาษา Python และในส่วนของหน้าหน้าบ้าน (frontend) ด้วยเฟรมเวิร์ค React โดยใช้ภาษา JavaScript สามารถดูวิธีการสร้างแอพพลิเคชั่นเพิ่มเติมที่ (ภาคผนวก ข)

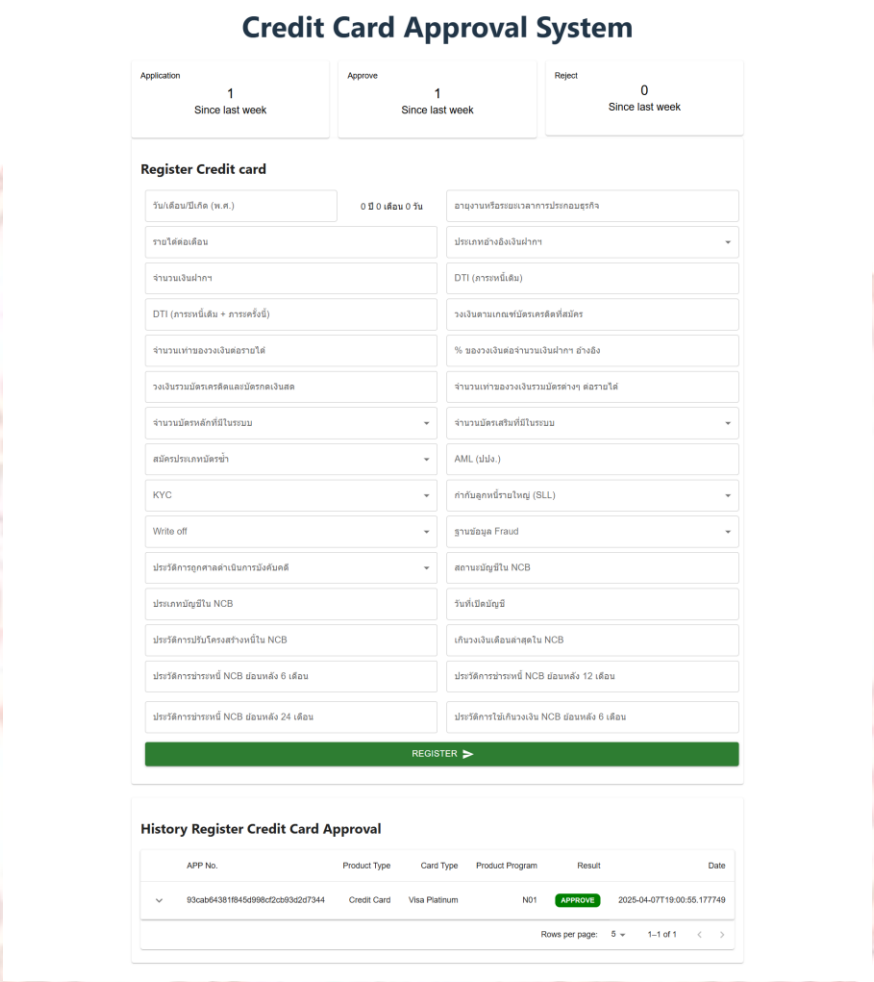
ในหน้าการกรอกและพิจารณาบัตรเครดิตเมื่อเจ้าหน้าที่ทำการกรอกข้อมูลคุณสมบัติของผู้สมัครบัตรเคดิต แล้วทำการกดเพื่อประมวลผลว่าคุณสมบัติที่กรอกเข้ามานั้นผ่านการพิจารณาหรือไม่ โดยจะแสดงผลการพิจารณาผ่านทางหน้าจอหลังจากประมวลผลแล้ว ถ้าผ่านจะขึ้นว่า “Approve” ถ้าไม่ผ่านจะขึ้นว่า “Reject”

ขั้นตอนการสมัครบัตรเครดิตเพื่อพิจารณาอนุมัติโดยการให้กรอกฟอร์มให้ครบถ้วน และกดปุ่ม “Register” ดังรูป 4-11



ภาพที่ 4-11 กรอกฟอร์มเพื่อพิจารณาอนุมัติ

เมื่อกรอกข้อมูลเรียบร้อยแล้ว กดปุ่ม “Register” เพื่อสมัครบัตรเครดิต ระบบจะประมวลผล เมื่อดำเนินการเสร็จสิ้นจะแสดงการแจ้งเตือนผลการอนุมัติทางหน้าจอ ดังรูป 4-12 อนุมัติผ่าน



ภาพที่ 4-12 ผลสมัครบัตรเครดิตหลังจากระบบประมวลผล

บทที่ 5
สรุปผล อภิปรายผล และข้อเสนอแนะ

จากการพัฒนาแบบจำลอง ประเมินคุณภาพ เปรียบเทียบ และพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต นั้นสามารถสรุปได้ดังนี้

5.1 สรุปผลและอภิปรายผล

5.3 ข้อเสนอแนะ

5.1 สรุปผลและอภิปรายผล

5.1.1 สรุปผล

การจัดทำวิจัยในครั้งนี้ผู้วิจัยได้ทำการเก็บรวบรวมข้อมูลจากผู้สมัครที่ยื่นสมัครบัตรเครดิตและผ่านการพิจารณาแล้วจากหน่วยงานแห่งหนึ่งในชั้นกระบวนการทดสอบกระบวนการทดสอบซอฟต์แวร์โดยผู้ใช้งานจริง (User Acceptance Test) มาใช้ในการจำแนกพิจารณาอนุมัติบัตรเครดิตเป็นอนุมัติและไม่อนุมัติ จากนั้นนำชุดข้อมูลมาเตรียมข้อมูลก่อนการประมวลผลทำการสำรวจข้อมูล (Exploratory data analysis) ว่าข้อมูลเป็นอย่างไรก่อนที่จะเตรียมข้อมูล (Data Preparation) ด้วยการลบฟีเจอร์ (feature) ที่ไม่จำเป็นออก ทำการลบข้อมูลที่ซ้ำ แปลงประเภทของข้อมูล และการเติมข้อมูลที่ขาด ทำการนำเสนอข้อมูล (Data Visualizing) ทำการเลือกฟีเจอร์ (Feature Selection) ในรูปแบบที่เข้าใจง่ายเพื่อแนวโน้มความสัมพันธ์ของข้อมูลในลักษณะต่างๆ ได้แก่ การแสดงค่าสหสัมพันธ์ Correlation หรือ ค่าสหสัมพันธ์ เป็นการดูทิศทางความสัมพันธ์ระหว่างตัวแปร โดยมีเพื่อในไปใช้ในการเลือกฟีเจอร์ (Feature Selection) ทำการจัดการข้อมูลให้เป็นระเบียบ (Data Normalizing) โดยการนำข้อมูลแต่ละฟีเจอร์ (Feature) ปรับให้ใหม่ให้เหมาะสมในการนำไปพัฒนาโมเดล หลังจากนั้นพบว่าข้อมูลกลุ่มที่อนุมัติและไม่อนุมัตินั้นไม่สมดุล ทำให้ต้องแก้ปัญหาการจำแนกข้อมูลไม่สมดุลด้วยเทคนิคการสุ่มแบบ SMOTE การคัดเลือกอัลกอริทึมสำหรับการจำแนก แก้ปัญหาการจำแนกข้อมูลไม่สมดุลโดยเทคนิคเป็นเทคนิคการสุ่มใช้ SMOT ทำการเลือกอัลกอริทึมที่มีพื้นฐานแตกต่างกัน 10 พื้นฐาน ได้แก่ Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM) , K-nearest Neighbors (KNN), Ensemble, XG Boost, Gradient Boosting Neuron Network และ, Naive Bayes หลังจากนั้นนำชุดข้อมูลที่เตรียมไว้ไปทำการสร้างแบบจำลอง ด้วยอัลกอริทึม 8 วิธี หลังจากที่ได้แบบจำลองก็นำไปวัดเปรียบเทียบประสิทธิภาพโมเดล แล้วสรุปผลที่ได้จากการทดลองดังนี้

5.1.1.1 จากผลการทดสอบพบว่าแบบจำลอง XG Boost มีประสิทธิภาพที่ดีกว่าแบบจำลองอีก 10 แบบจำลองประกอบด้วย Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM) , K-nearest Neighbors (KNN), Ensemble และ Gradient Boosting ในส่วนของทดสอบประสิทธิภาพแบบจำลองการตัดสินใจอนุมัติบัตรเครดิตเป็น 2 ประเภท อนุมัติ (Approve) และไม่อนุมัติ (Reject) เมื่อนำมาทดลองใช้ชุดข้อมูลโดยกำหนดพารามิเตอร์ให้เหมาะสมแล้วพบว่าส่งผลให้ แบบจำลองสามารถบรรลุประสิทธิภาพที่เปอร์เซ็นต์ความถูกต้องมากที่สุดคือแบบจำลอง XG Boost ให้ค่าความถูกต้อง 98.0% รองลงมา Ensemble และ Random Forest ให้ค่า ความถูกต้อง 96.7% Decision Tree ให้ค่าควาถูกต้อง 94.2% สูงสุด Gradient Boosting ให้ค่าความถูกต้อง 82.0% K-Nearest Neighbors ให้ค่าความถูกต้อง 78.8% Support Vector Machine ให้ค่าความถูกต้อง 73.0% Logistic Regression ให้ค่าควาถูกต้อง 72.4% Naive Bayes ให้ค่าควาถูกต้อง 23.0% และ Neuron Network ให้ค่าควาถูกต้อง 11.5% ลดลงตามลำดับ

5.1.1.2 สำหรับประสิทธิภาพในส่วนของรายงานการจำแนก (Classification) ที่จะอธิบายในด้านค่าความแม่นยำ (Precision), การเรียกคืน (Recall) และคะแนน F1(F1-Score) พบว่าแบบจำลอง Ensemble แล้ว ยังให้ประสิทธิภาพที่ดีที่สุดในการจำแนกกลุ่มลูกค้าที่สมัครบัตร เมื่อเปรียบเทียบกับแบบจำลองอื่นๆ แล้วเช่นกัน โดยสรุปได้ว่าภาพรวมค่าเฉลี่ยถ่วงนำหนัก (Macro average) คะแนน F1 ของแบบจำลอง XG Boost นั้นมีค่าสูงสุดที่ 0.95 รองลงมาคือแบบจำลอง Ensemble และ Random Forest มีค่า 0.93 แบบจำลอง Decision Tree มีค่า 0.88 แบบจำลอง K-nearest Neighbors มีค่า 0.68 แบบจำลอง Support Vector Machines มีค่า 0.61 แบบจำลอง Logistic Regression มีค่า 0.60 แบบจำลอง Naive Bayes มีค่า 0.23 และแบบจำลอง Neuron Network มีค่า 0.10 ลดลงตามลำดับ

5.1.1.3 เมื่อพิจารณาเพิ่มเติมนเมทริกซ์ความสับสน (Confusion Matrix) พบว่าแบบจำลอง XG Boost นั้นยังคงมีประสิทธิภาพที่เหนือกว่า แบบจำลองจากการเรียนรู้ของเครื่องอื่นๆ เนื่องจากในการแก้ปัญหาการแยกกลุ่มข้อมูล เนื่องจากเมทริกซ์ ความสับสนมีการเน้น (Highlight) หรือความเข้มของสีน้ำเงินเข้ม (ค่าสูง) ในแนวทแยงจากบนซ้ายไป-ล่างขวาทั้งหมด แบบจำลอง ensemble มีค่า True Positive (TP) มีค่าสูงที่สุด และ True Negative (TN) มีค่าสูงที่สุด ส่งผลให้โมเดลทำนายถูกต้องมากขึ้น ส่วนค่า False Positive (FP) มีค่าต่ำแสดงถึงการลดการทำนายผิดพลาดว่าเป็นบวกหรือหมายถึงไม่ให้คนที่ไม่ควรได้บัตรเครดิตผ่านการอนุมัติ และสุดท้าย False Negative (FN) มีค่าต่ำที่สุดแสดงถึงลดการทำนายผิดพลาดว่าเป็นบวกหรือหมายถึงไม่ให้คนที่ไม่ควรได้บัตรเครดิตผ่านการอนุมัติ

5.1.1.4 เมื่อพิจารณาเพิ่มเติมเส้นโค้ง AUC-ROC นั้นค่า AUC ของการแยกแยะแต่ละกลุ่มของแบบจำลอง XG Boost ยังมีค่าที่มากกว่า 0.5 ค่า และมีค่า AUC สูงที่สุดที่ 0.970 ลองลงมา Gradient Boosting มีค่า 0.962 random forest มีค่า 0.891 Ensemble และ Decision tree มีค่าเท่ากันที่ 0.851 K-Neighbors Classifier มีค่า 0.892 Logistic regression มีค่า 0.845 Support Vector Machine มีค่า 0.778 Neural Network มีค่า 0.889 และ Naive Bayes มีค่า 0.554 น้อยสุด ลดลงตามลำดับ

5.1.1.5 เมื่อผ่านกระบวนการสร้างแบบจำลองและเปรียบเทียบประสิทธิภาพของแต่ละแบบจำลอง แล้ว จะเข้าสู่กระบวนการนำแบบจำลองที่ได้นี้นำไปประยุกต์ใช้เพื่อพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิตในรูปแบบของเว็บแอป (Web Application) รอบรับในการประมวลผลอนุมัติบัตรเครดิตแบบเรียลไทม์มากขึ้น กล่าวคือการนำแบบจำลองที่สร้าง มาใช้พยากรณ์ผลและรู้ผลลัพธ์ได้ในทันที เพราะในความเป็นจริงแล้ว เจ้าหน้าที่ทำการกรอกข้อมูลคุณสมบัติของผู้สมัครบัตรเคดิต แล้วทำการกดเพื่อประมวลผลว่าคุณสมบัติที่กรอกเข้ามานั้นผ่านการพิจารณาหรือไม่ โดยจะแสดงผลการพิจารณาผ่านทางหน้าจอหลังจากประมวลผลแล้ว ถ้าผ่านจะขึ้นว่า “Approve” ถ้าไม่ผ่านจะขึ้นว่า “Reject” แบบทันที เพื่อสนับสนุนงานอนุมัติบัตร และให้ผลลัพท์ที่ดีกว่าเก่าอย่างมีประสิทธิภาพมากยิ่งขึ้น ตัดสินใจได้รวดเร็วขึ้น ช่วยลดขั้นตอนและภาระของพนักงานลงได้ ลดความเสี่ยงที่อาจการเกิดหนี้ศูนย์

5.1.2 อภิปรายผล

5.1.2.1 สำหรับการทดสอบประสิทธิภาพของแบบจำลองเมื่อเปรียบเทียบด้วยเมทริกซ์การเมินที่หลากหลาย ประกอบด้วย ค่าความแม่นยำถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าเรียกคืน (Recall) คะแนน F1 (F1-Score) และ เมทริกซ์ความสับสน (Confusion Matrix) พบว่าระหว่างแบบจำลอง XG Boost, Ensemble และ Random Forest มีค่าประสิทธิภาพที่ใกล้เคียงกันมากถึงแม้ว่าแบบจำลอง XG Boost จะมีค่ามากที่สุด และ XG Boost กับ Random Forest ประสิทธิภาพที่ด้อยกว่าเพียงเล็กน้อย แต่เมื่อเปรียบเทียบประสิทธิภาพโดยรวมไม่ต่างกันมาก หากมีจำนวนข้อมูลทดสอบมากขึ้น หรือมีข้อมูลที่สมบูรณ์บนชั้นข้อมูลจริง สามารถนำเข้าข้อมูลเหล่านี้เพื่อใช้ในการฝึกฝนข้อมูลใหม่อีกครั้งนั้น (Refitting model) จะพบว่าแบบจำลอง และ XG Boost และ Random Forest จะเป็น อีกหนึ่งทางเลือกที่จะนำมาพิจารณาด้วยเช่นกัน

5.1.2.2 สำหรับการทดสอบประสิทธิภาพของแบบจำลองเมื่อเปรียบเทียบด้วยเส้นโค้ง AUC-ROC พบว่าระหว่างแบบจำลอง XG Boost มีค่าสูงที่สุดที่ 0.970 แต่เมื่อประกอบกับพิจารณาโดยโดยรวมแล้ว แบบจำลอง XG Boost นั้นเป็นแบบจำลองที่โดดเด่นกว่าหรือแม่นยำกว่ามากๆ กว่าแบบจำลองคู่เปรียบเทียบอื่นๆ และมีประสิทธิภาพที่จะทำงานในระดับของการคัดแยกประเภทกลุ่ม

ลูกกลุ่มผู้สมัครบัตรเครดิตได้ดี เนื่องจากมีค่าความและแม่นยำถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) มากกว่าแบบจำลองรุ่น อื่นๆ จึงเลือกเป็นแบบจำลองที่ดีที่สุดในงานวิจัยนี้

5.1.2.3 ในการทดสอบนี้เนื่องจากชุดข้อมูล (Dataset) ที่ใช้เป็นข้อมูลที่ไม่สมดุล กล่าวคือกลุ่ม ข้อมูลประเภทกลุ่มผู้สมัครบัตรเครดิตที่มีคุณสมบัติที่ไม่ผ่าเกณฑ์พิจารณาที่เป็นข้อมูลที่ไม่สมดุล ดังนั้นการใช้ เทคนิค SMOTE ที่จะช่วยจัดการปัญหาความไม่สมดุลของข้อมูล อาจส่งผลต่อผลลัพธ์ของ ประสิทธิภาพที่ได้ของแต่ละแบบจำลอง

5.2 ข้อเสนอแนะจากงานวิจัย

5.2.1 ควรเพิ่มจำนวนข้อมูลที่ใช้สำหรับสร้างแบบจำลองสำหรับการจำแนกประเภทกลุ่มผู้สมัครบัตรเครดิตที่มีคุณสมบัติที่ไม่ผ่าเกณฑ์พิจารณาที่มากกว่านี้เพื่อกำจัดปัญหาการไม่สมดุลของข้อมูล

5.2.2 ควรเพิ่มจำนวนข้อมูลที่ใช้สำหรับสร้างแบบจำลองสำหรับการจำแนกในประเภทในประเภทบัตรอื่นเช่น บัตรมาสเตอร์การ์ด (Master Card) บัตรกดเงินสด (Revolving Card) รวมถึงอาจจะใช้เกณฑ์รายได้รูปแบบอื่นเช่น อ้างอิงเงินฝาก เป็นต้น เพื่อให้ครอบคลุมในการประมวลผลอนุมัติบัตรได้มากกว่านี้

5.2.3 เนื่องจากชุดข้อมูล (Dataset) ที่นำมาใช้อยู่ในชั้นกระบวนการทดสอบกระบวนการทดสอบซอฟต์แวร์โดยผู้ใช้งานจริง (User Acceptance Test) อาจจะมีข้อมูลที่ไม่ครบ และเสหมือนจริงน้อยกว่าบนชั้นข้อมูลจริง ส่งผลให้การทดสอบคลาดเคลื่อน

5.2.4 สำหรับแบบจำลอง Neural Network และ Naive Bayes ที่ประสิทธิภาพยังไม่แม่นยำเท่าที่ควร หากสามารถปรับ ไฮเปอร์พารามิเตอร์หรือเลือกใช้พารามิเตอร์ตัวอื่นๆของอัลกอริทึมมาปรับจูนให้ละเอียดขึ้น อาจส่งผล ต่อประสิทธิภาพของแบบจำลองที่แม่นยำในการพยากรณ์ผ

บรรณานุกรม

1. ขวัญนภา พิมพ์ชารีย์. การใช้เหมืองข้อมูลสำหรับพิจารณาการให้สินเชื่อสำหรับธนาคาร, มหาวิทยาลัยมหาสารคาม, 2021.
2. เครือวัลย์ เนตรพนาและ ศิริสรรพ เหล่าหะเกียรติ. การวิเคราะห์ความเสี่ยงในการผิดนัดชำระของลูกหนี้บัตรเครดิต โดยการใช้อัลกอริทึมการเรียนรู้ของเครื่อง, 2023 3rd Proceeding of the Data Science Conference, 2023.
3. ณัฐพล แสนคำ. (2563). [ออนไลน์]. Postman : มารู้จักกับ postman กันเถอะ. [สืบค้นวัน 20 ตุลาคา 2566]. จาก https://medium.com/lotuss-it/ postman-มารู้จักกับ-postman-กันเถอะ-157193f5a215.
4. _____. (2563). [ออนไลน์]. วิธีการใช้งาน Visual Studio Code. [สืบค้นวัน 20 ตุลาคา 2566]. จาก https://cs.bru.ac.th/สอนวิธีการใช้-visual-studio-code-2/.
5. ธนาคารกรุงไทย. (2566). [ออนไลน์]. สำรวจภาระหนี้ มีแค่ไหน ถึงอยู่ในจุดปลอดภัย. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://krungthai.com/th/financial-partner/learn-financial/1569.
6. ธนาคารแห่งประเทศไทย. (2565). [ออนไลน์]. หนี้ครัวเรือน: ปัญหาที่ทุกคนต้องร่วมด้วยช่วยกันแก้. [สืบค้นวัน 15 ตุลาคา 2566]. จาก https://www.bot.or.th/th/research-and-publications/articles-and-publications/articles/Article_1Jan2022_2.html.
7. _____. (2566). [ออนไลน์]. การให้สินเชื่ออย่างรับผิดชอบและเป็นธรรม. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.bot.or.th/content/dam/bot/fipcs/documents/FPG/2563/ThaiPDF/25630225.pdf.
8. _____. (2566). [ออนไลน์]. เครดิตบูโร. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.bot.or.th/th/satang-story/managing-debt/creditbureau.html.
9. _____. (2566). [ออนไลน์]. หนี้ครัวเรือน เป็นปัญหาคู่สังคมไทยที่หลายคนคงคุ้นหูกันอยู่แล้ว. [สืบค้นวัน 15 ตุลาคา 2566]. จาก https://projects.pier.or.th/house hold-debt/.
10. _____. (ม.ป.ป.). [ออนไลน์]. บัตรเครดิต. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.bot.or.th/th/satang-story/digital-fin-lit/creditcard.html.

บรรณานุกรม (ต่อ)

11. นายสุรพล โอภาสเสถียร. (2567) . [ออนไลน์].เครดิตบูโร” เปิดตัวเลข NPL ไทย Q3 พุ่ง 13.63 ล้านลบ. “หนี้เสียรถ-บ้าน” สูงสุด. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.kaohoon.com/news/706647.
12. บริษัท ข้อมูลเครดิตแห่งชาติ จำกัด. (2567). [ออนไลน์]. เครดิตสกอริ่ง (คะแนนเครดิต) สำหรับบุคคลธรรมดา. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.ncb.co.th/fin-knowledge/credit-score-benefits/.
13. บริษัท เท็น เอ็กซ์ เอ็นเตอร์ไพรซ์ จำกัด. (2567). [ออนไลน์]. สะสมแต้มแลกของรางวัล สร้างแคมเปญการตลาดเพื่อเพิ่มยอดขาย. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://crm-x.m/สะสมแต้มแลกของรางวัล/.
14. บริษัท บัตรกรุงไทย จำกัด. (2567). [ออนไลน์]. 8 เหตุผลที่สมัครบัตรเครดิตไม่ผ่าน พร้อมวิธีสมัครเครดิตให้ผ่านฉลุ. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.ktc.co.th/en/article/knowledge/credit-card-application-declined.
15. พิสุทธิ์ ตั้งพิษฐานสกุล. (2557). เว็บแอพพลิเคชันสำหรับงานประเมินมูลค่าอสังหาริมทรัพย์. การค้นคว้าแบบอิสระ สาขาวิชาวิศวกรรมซอฟต์แวร์ บัณฑิตวิทยาลัย มหาวิทยาลัยเชียงใหม่.
16. พีรพงษ์ พิมพ์อูบ. (2563). การพัฒนาระบบวิเคราะห์การสั่นสะเทือนเพื่อช่วยในงานซ่อมบำรุงด้วยเว็บเบสแอปพลิเคชัน. วิทยานิพนธ์ สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ มหาวิทยาลัยสุรนารี.
17. วัชระ ทองสุข. (2566). [ออนไลน์]. Machine Learning คืออะไร สำคัญอย่างไรต่อโลกธุรกิจยุคนี้. [สืบค้นวัน 15 ตุลาคา 2566]. จาก https://www.kaohoon.com /news/706647.
18. สกุลกาญจน์ ทองคำ และนุวีย์ วิวัฒนวัฒนา. การเรียนรู้ของเครื่องเพื่อการทำนายการผิดนัดชำระของลูกหนี้บัตรเครดิต, 2022 2nd Proceeding of the Data Science Conference, 2022.
19. สมประวิณ มันประเสริฐ. (2564). [ออนไลน์]. Platform Economy... คำตอบของระบบเศรษฐกิจไทยที่ให้โอกาสเท่าเทียมกัน?. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://thestandard.co/platform-economy/.

บรรณานุกรม (ต่อ)

20. สำนักงานนายกรัฐมนตรี. (2565). [ออนไลน์]. คำกล่าว พลเอกประยุทธ์ จันทร์โอชา นายกรัฐ-มนตรี เปิดเสวนาชี้แจงผลงานของรัฐบาล. [สืบค้นวัน 15 ตุลาคา 2566]. จาก https://www.thaigov.go.th/news/contents/details/54766.
21. สุดากาญจน์ ติดตะ และ ดร.ธนภัทร ฆังคะจิตร. ระบบการพิจารณาอนุมัติเงินกู้สวัสดิการพนักงานอัตโนมัติด้วยเทคนิควิธีการเรียนรู้ของเครื่อง, มหาวิทยาลัยธุรกิจบัณฑิตย์, 2021.
22. สุเมธ จุฑาจันทร์ และ สมพร ปั่นโภชา. เปรียบเทียบผลลัพธ์ของการอนุมัติสินเชื่อด้วย 3 แบบจำลองของระเบียบวิธี Machine Learning โดยใช้โปรแกรมอาร์, การประชุมนำเสนอผลงานวิจัยบัณฑิตศึกษาระดับชาติ ครั้งที่ 16 ปีการศึกษา 2022.
23. AI แบบเคลียร์. (2566). [ออนไลน์]. SVM ตอนที่ 1: Support Vector Machine โมเดลคลาสสิคที่น่าศึกษา. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://beeying.medium.com /support-vector-machine-svm-โมเดลคลาสสิคที่น่าศึกษา-78d1ce6da765.
24. Amazon Web Services. (ม.ป.ป.). [ออนไลน์]. Python คืออะไร. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://aws.amazon.com/th/what-is/python/.
25. _____. (ม.ป.ป.). [ออนไลน์]. เว็บแอปพลิเคชันคืออะไร. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://aws.amazon.com/th/what-is/web-application/.
26. ananda. (2563). [ออนไลน์]. ขีดเส้นใต้ “จุดปลอดภัย” แบกภาระหนี้แค่ไหน ถึงจะเรียกว่า ‘พอดี. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.a nanda.co.th/blog/thegen c/safe-debt/.
27. Atom. (ม.ป.ป.). [ออนไลน์]. ทำความรู้จักกับกับต้นกำเนิดของบัตรเครดิต. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://thecreditcardthai.blogspot.com/search/labe/ประวัติและความเป็นมาของบัตรเครดิต/.
28. Cory Maklin. (2565). [ออนไลน์]. Synthetic Minority Over-sampling TEchnique (SMOTE. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c.
29. Dia. (2567). [ออนไลน์]. ปัญญาประดิษฐ์ (AI) คืออะไร หลักการทำงาน และการใช้ในอุตสาหกรรม. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.dia .co.th/articles/ what-is-artificial-intelligence/.

บรรณานุกรม (ต่อ)

30. Indusadmin. (2560). [ออนไลน์]. เศรษฐกิจดิจิทัล นโยบายขับเคลื่อนเศรษฐกิจใหม่. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.industry.go.th/th/km/3371.
31. Jirapat Klaokliang. (2564). [ออนไลน์]. ประเมินประสิทธิภาพโมเดลด้วย Confusion Matrix. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://datascihaeng.medium.com/evaluation-matrix-part2-a7d83aea6537.
32. John Burke. (2566). [ออนไลน์]. Why and how to use Google Colab. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.techtarget.com/searchenterpriseai/tutorial/Why-and-how-to-use-Google-Colab.
33. Kasidis Satangmongkol. (2565). [ออนไลน์]. เข้าใจการทำงานพื้นฐานของ Neurons ใน Neural Networks. จาก https://datarockie.com/blog/how-neurons-work/.
34. Krungsri Consumer. (2566). [ออนไลน์]. วิวัฒนาการของบัตรเครดิตจากอดีตจนก้าวเข้าสู่ยุคดิจิทัล. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.krungsriconsumer.com/customer-guide/journey-creditcard.
35. Launchplatfrom. (2566). [ออนไลน์]. Material UI (MUI) คืออะไร ทำไมสำคัญกับการทำแอปพลิเคชัน. [สืบค้นวัน 20 ตุลาคา 2566]. จาก https://launchplatform.co.th/article/UXUI/material-ui.
36. Littikrai. (2564). [ออนไลน์]. สร้าง API ด้วย Python (FastAPI) แบบเบื้องต้น. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://stackpython.co/tutorial/api-python-fastapi.
37. Merkle Capital. (2567). [ออนไลน์]. สังคมไร้เงินสด (Cashless Society) คืออะไร จะเปลี่ยนโลกไปไหนทิศทางไหน. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://merkle.capital/articles/cashless-society.
38. Mo Nuttamon. (2567). [ออนไลน์]. E-commerce | อีคอมเมิร์ซ คืออะไร ? มีประเภทอะไรบ้าง. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.foxbith.com/blog/what-is-ecommerce.
39. Mr.P L. (2561). [ออนไลน์]. Evaluate Model นั้นสำคัญอย่างไร ? : Machine Learning 101. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/mmp-li/evaluate-model-precision-recall-f1-score-machine-learning-101-89dbbada0c96.

บรรณานุกรม (ต่อ)

40. _____. (2561). [ออนไลน์]. K-NN กับ Sklearn : Machine Learning 101. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/mmp-li/ k-nn-กับ-sklearn-machine-learning-101-81350e8402f0.
41. Narut. (2562). [ออนไลน์]. Machine Learning (ML) มีกี่ประเภท ?. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/@narut.soo/machine-learning-ml-มีกี่ประเภท-7a3599be19cd.
42. Nathaniel Soto. (2566). [ออนไลน์]. ทำความรู้จักกับกับต้นกำเนิดของบัตรเครดิต. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://ijira.org/ทำความรู้จกกับกับต้นกำเ/.
43. National e-Payment. (2558). [ออนไลน์]. National e-Payment เป็นระบบการชำระเงินแบบอิเล็กทรอนิกส์. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.epayment.go.th/home/app/.
44. Nut Chukamphaeng. (2561). [ออนไลน์]. ทำไมใครๆก็ใช้ XGBoost กันจัง. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/@nutorbitx/ทำไมใครๆก็ใช้-xgboost-กันจัง-a775b53cc1eb.
45. opj. (2567). [ออนไลน์]. รู้จัก Gradient Boosting แบบเจาะลึก. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.mindphp.com/forums/viewtopic.php?t=112888.
46. Pacharee Toorakidsana. (2564). [ออนไลน์]. Python คืออะไร? เป็นภาษาที่ง่ายที่สุดจริงหรือ?. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://blog.skooldio.com/what-is-python/#:~:text=Python%20คือหนึ่งในภาษา,แปลชุดคำสั่งที.
47. Panupong Suksuwan. (2562). [ออนไลน์]. Data Analytics Life cycle. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/tni-university/data-analytics-life-cycle-2e5c29a77369.
48. Papimpat Nimprasert. (2567). [ออนไลน์]. ทำความรู้จักกับ React และการใช้งานเบื้องต้น. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.borntodev.com/2024/05/13ทำความรู้จักกับ-react/.
49. Pasith Thanapatpisarn. (2564). [ออนไลน์]. [Data Scientist Model EP.03] Logistic Regression โมเดลเริ่มต้นสำหรับการทำนายผลแบบ Classification Part1. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://datascihaeng.medium.com/data-science-model-03-d06385d789f7.

บรรณานุกรม (ต่อ)

50. _____. (2565). [ออนไลน์]. [Data Scientist Model EP.07] Decision Tree โมเดลต้นไม้ กับความเรื่องมากในการตัดสินใจ!!! Part1. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://datascihaeng.medium.com/decision-tree-part01-47ef24539fba.
51. _____. (2565). [ออนไลน์]. [Data Scientist Model EP.09] Ensemble Method “ขั้นกว่าของ Decision Tree ?” Part1. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://datascihaeng.medium.com/ensemble-method-part01-a0b79da6b459.
52. _____. (2565). [ออนไลน์]. [Fundamental Data Analytics & Data Scientist EP.21] วัดประสิทธิภาพของโมเดลด้วย Evaluation Metrics Part 2. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://nattrio.medium.com/ประเมินประสิทธิภาพโมเดลด้วย-confusion-matrix-84c63ec9bd94.
53. Paweennuch W. (2565). [ออนไลน์]. บัตร Visa คือบัตรอะไร?. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://rabbitcare.com/visa-card.credit-card.financial-guide.
54. Peachapong Poolpol. (2564). [ออนไลน์]. Naïve Bayes Classification. จาก https://peachapong-poolpol.medium.com/na%C3%AFve-bayes-classification-cb6cf905505d.
55. Pimporn Pimpim. (2562). [ออนไลน์]. 03 Feature selection การเลือกคุณลักษณะ. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/@doohpim/ 03-feature-selection-การเลือกคุณลักษณะ-446f8f0c094b.
56. PradyaSin. (2565). [ออนไลน์]. Random Forest คืออะไร. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://medium.com/@pradyasin/random-forest-คืออะไร-74d2a0af3d7.
57. refinn writer. (2563). [ออนไลน์]. สินเชื่อส่วนบุคคล (Personal Loan) คืออะไร มาทำความเข้าใจกัน. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.refinn.com/blog/what-is-personal-loan.
58. Rukchanok Piyapanichayakul. (2566). [ออนไลน์]. ปัญหาข้อมูลไม่สมดุล (Imbalanced Data in Classification Model). [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://ntcloudsolutions.ntplc.co.th/knowledge/ imbalanced-data-classification/.

บรรณานุกรม (ต่อ)

59. Sas. (2567). [ออนไลน์]. การเรียนรู้ของเครื่อง (machine learning) นิยามและความสำคัญ. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.sas.com/th_th/insights/analytics/machine-learning.html.
60. SET. (2564). [ออนไลน์]. GDP คืออะไร? ทำไมกระทบตลาดหุ้นทุกที?. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.setinvestnow.com/th/knowledge/article/333-investhow-gdp.
61. Supawish kaewjing. (2566). [ออนไลน์]. Vite ที่ไม่ได้เร็วแค่ชื่อ แต่.... [สืบค้นวัน 20 ตุลาคา 2566]. จาก https://developers.ascendcorp.com/vite-ที่ไม่ได้เร็วแค่ชื่อ-แต่-cced3b4eb68c.
62. Surapong Kanoktipsatharporn. (2562). [ออนไลน์]. Training Set คืออะไร ทำไมเราต้องแยกชุดข้อมูล Train / Test Split เป็น Training Set, Validation Set และ Test Set ใน Machine Learning. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://www.bualabs.com/archives/532/what-is-training-set-why-train-test-split-training-set-validation-set-test-set/.
63. swiftlet. (2565). [ออนไลน์]. รู้จักกับ User Acceptance Testing (UAT) ขั้นตอนสำคัญของการทดสอบซอฟต์แวร์!. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://swiftlet.co.th/what-is-uat/.
64. Thapanee Boonchob. (2564). [ออนไลน์]. เข้าใจ CRISP-DM ฉบับเร่งรัด. [สืบค้นวัน 20 ตุลาคา 2567]. จาก https://kamboonchob.medium.com/เข้าใจ-crisp-dm-ฉบับเร่งรัด-b0913050198f.
65. THE STANDARD TEAM. (2567). [ออนไลน์]. หนี้เสียรถพุ่ง ถูกยึดเต็มลาน สัญญาณบอก คนจมวิกฤต?. [สืบค้นวัน 22 ตุลาคา 2565]. จาก https://thestandard.co/car-debts-are-soarin/.
66. JiXiang Niu. Research on loan prediction based on interpretable machine learning, 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). 2022.
67. Makumburage Poornima, Chathurangi Peiris, Credit Card Approval Prediction by Using Machine Learning Techniques, University of Colombo School of Computing, 2019.

บรรณานุกรม (ต่อ)

68. Naman Dalsania, Devang Punatar and Deep Kothari. Credit Card Approval Prediction using Classification Algorithms, Ijraset Journal For Research in Applied Science and Engineering Technology. 2022.
69. Prabaljeet Singh Saini, Atush Bhatnagar and Lekha Rani. Loan Approval Prediction using Machine Learning: A Comparative Analysis of Classification Algorithms, 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). 2023.
70. Puneeth B. R, Ashwitha K, Arhath Kumar, et al. An Approach to Predict Loan Eligibility using Machine Learning, 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE). 2022.
71. Viswanatha V, Ramachandra A.C ,et al. Prediction of Loan Approval in Banks Using Machine Learning Approach, International Journal of Engineering and Management Research, Volume 13, Issue 4). 2023.
72. Yiran Zhao. Credit Card Approval Predictions Using Logistic Regression, Linear SVM and Naïve Bayes Classifier, 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE). 2022.

ภาคผนวก ก

คู่มือการใช้งานระบบ

คู่มือการใช้งานระบบ

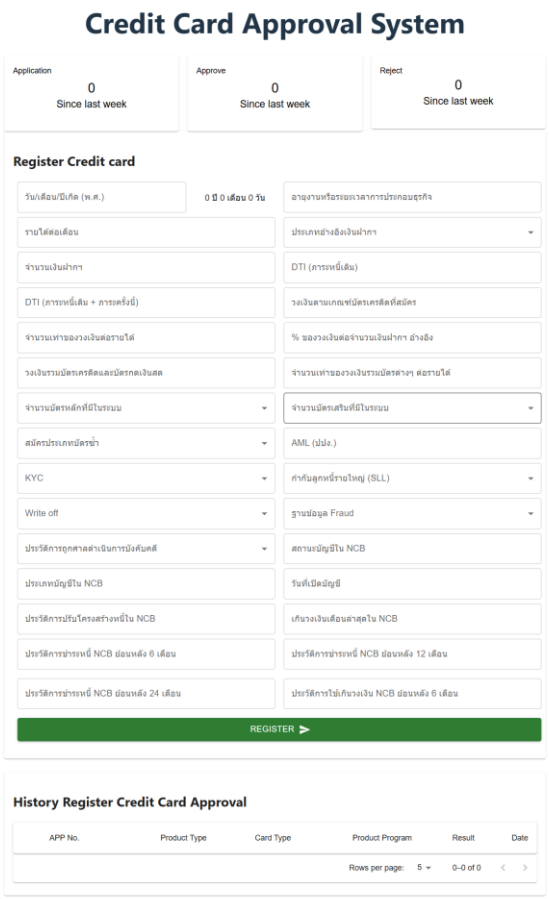
แอปพลิเคชันการพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิตขั้นตอนการใช้งานมีรายละเอียดดังนี้

1. การเข้าใช้งานระบบ

ระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิตสามารถใช้ผ่าน Web Application มีขั้นตอนการเข้าใช้งานดังนี้

1.1 เข้าผ่าน Web Application ผ่าน browser URL : http://localhost:5173/

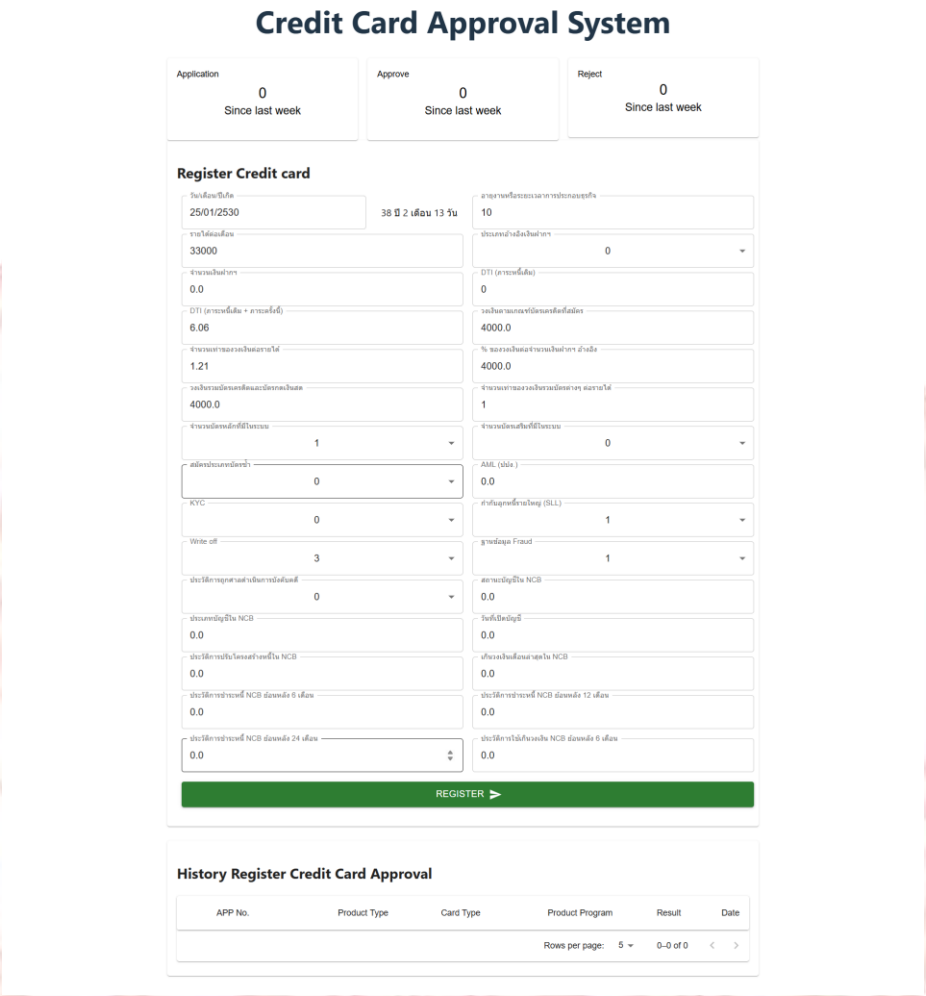
1.2 แสดงหน้าแรก ดังรูป ก-1



รูปที่ ก-1 หน้าแรก

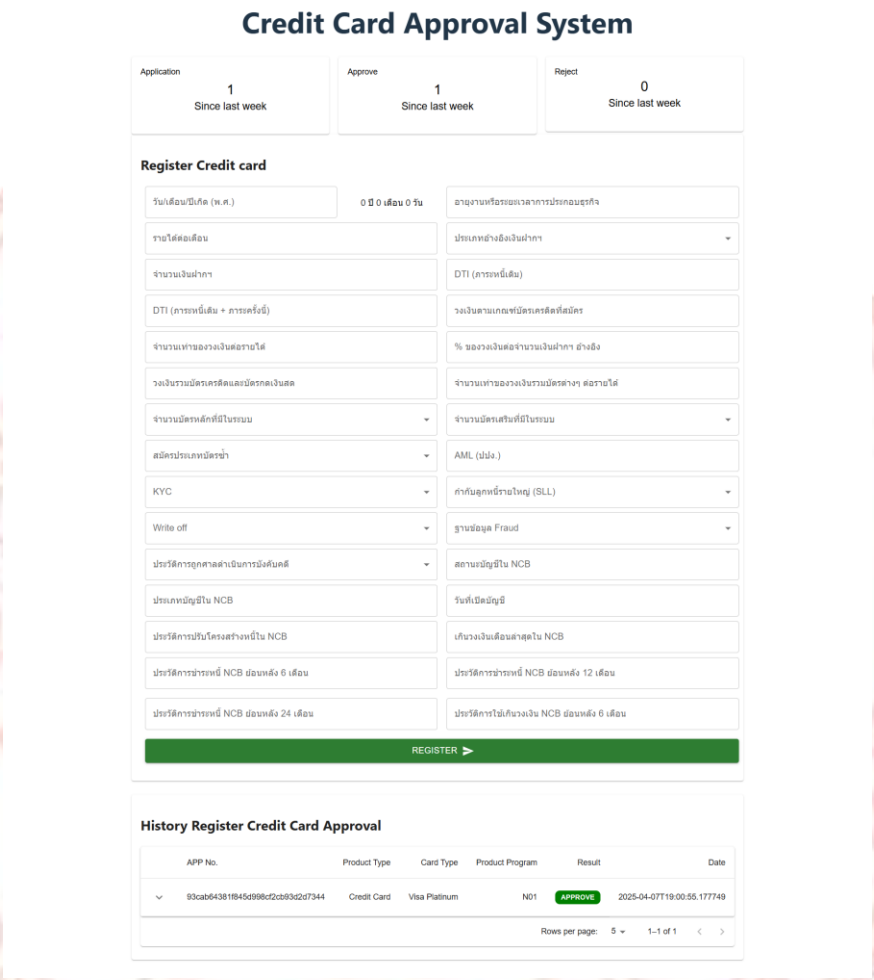
2. สมัครบัตรเครดิตเพื่อพิจารณาอนุมัติ

2.1 ให้กรอกฟอร์ม ดังรูป ก-2



รูปที่ ก-2 กรอกฟอร์ม

2.2 เมื่อกรอกข้อมูลเรียบร้อยแล้ว กดปุ่ม “**Register**” สมัครบัตรเครดิต ระบบจะประมวลผล เมื่อดำเนินการเสร็จสิ้นจะแสดงการแจ้งเตือนผลการอนุมัติทางหน้าจอ ดังรูป ก-3



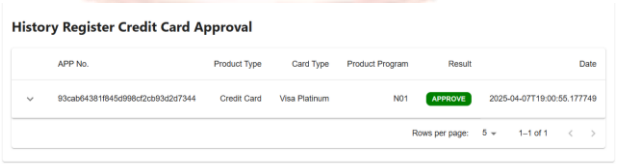
รูปที่ ก-3 แจ้งผลสมัครบัตรเครดิต

2.2 กดปุ่ม “ v ” เพื่อดูรายละเอียดข้อมูลเพิ่มเติมของรายการสมัครบัตรเครดิต ดังรูป ก-4

รูปที่ ก-4 รายละเอียดข้อมูลเพิ่มเติม

3. การแสดงรายงาน (Report) และประวัติที่คยมีการสมัครบัตรเครดิตทั้งหมด ดังรูป ก-5 และ ก-6

ภาพที่ ก-5 Dashboard สรุปข้อมูลรายการสมัครบัตรเครดิตทั้งหมด



รูปที่ ก-6 ประวัติการสมัครบัตรเครดิตทั้งหมด

ภาคผนวก ข

การพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต

การพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต

แอปพลิเคชันการพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิตมีขั้นตอนการจัดทำดังนี้

1. หน้าบ้าน (front-end)

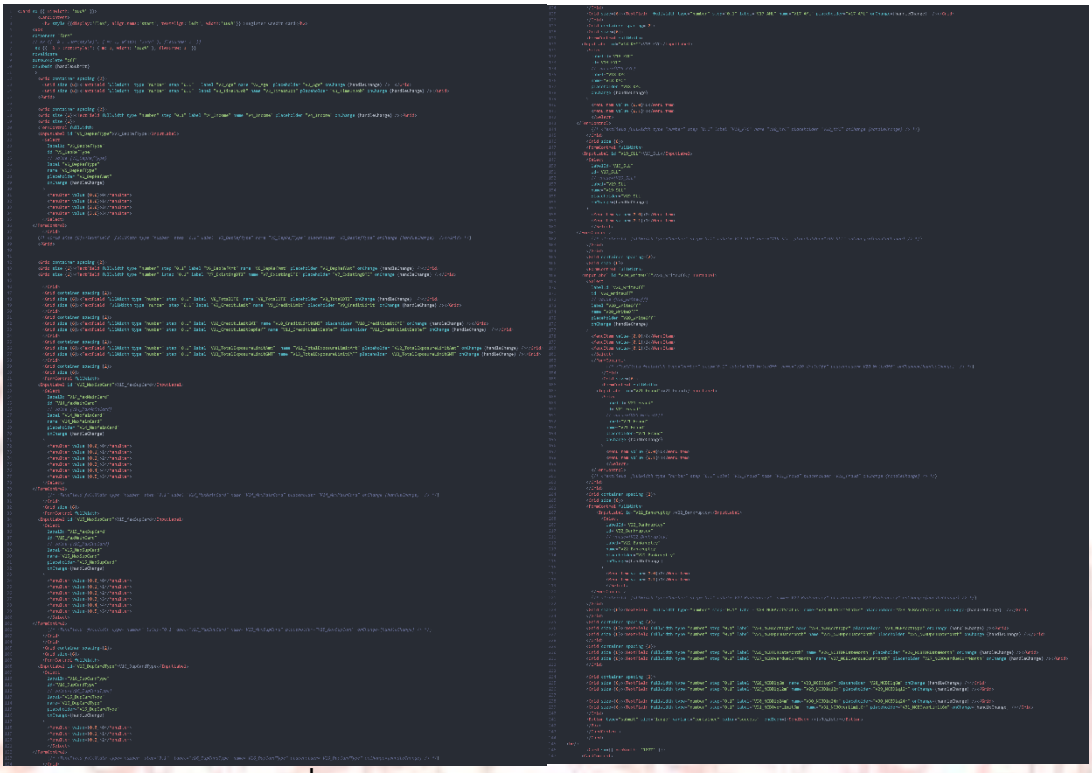
1.1 เนื่องจากการแอปพลิเคชันการพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต หลังหน้าบ้าน (front-end) มีการพัฒนาด้วยภาษาคอมพิวเตอร์ JavaScript Html Css จึงเรียกใช้ไลบรารี่ดังรูป ข-1

predict-credit-card-fontend > src > components > RegisterCreditCardForm.jsx > RegisterCreditCardForm

```
import React,  { useEffect, useState } from "react";  7.8k (gzipped: 3k)
import api from '../api';
import Form from 'react-bootstrap/Form';

import Box from '@mui/material/Box';
import TextField from '@mui/material/TextField';
import Paper from '@mui/material/Paper';
import Grid from '@mui/material/Grid2';
import { styled } from '@mui/material/styles';
import IconButton from '@mui/material/IconButton';

import Card from '@mui/material/Card';
import CardActions from '@mui/material/CardActions';
import CardContent from '@mui/material/CardContent';
import CardMedia from '@mui/material/CardMedia';

import Table from '@mui/material/Table';
import TableBody from '@mui/material/TableBody';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TableRow from '@mui/material/TableRow';
import SendIcon from '@mui/icons-material/Send';

import Stack from '@mui/material/Stack';
import Button from '@mui/material/Button';


import Collapse from '@mui/material/Collapse';
import Typography from '@mui/material/Typography';

import PropTypes from "prop-types";
import KeyboardArrowDownIcon from "@mui/icons-material/KeyboardArrowDown";
import KeyboardArrowUpIcon from "@mui/icons-material/KeyboardArrowUp";

import ReactDOM from 'react-dom'  4k (gzipped: 1.5k)
import TablePagination from "@mui/material/TablePagination";

import InputLabel from '@mui/material/InputLabel';
import MenuItem from '@mui/material/MenuItem';
import FormControl from '@mui/material/FormControl';
import Select from '@mui/material/Select';

```

รูปที่ ข-1 การอิมพอร์ตไลบรารี่เพื่อทำสร้างระบบหน้าบ้าน

1.2 สร้างฟอร์มการกรอกข้อมูลคุณสมบัติของผู้สมัครสำหรับให้แบบจำลองทำนายผลลัพธ์ ดังรูป ข-2



รูปที่ ข-2 ฟอร์มการกรอกข้อมูลคุณสมบัติของผู้สมัคร

1.3 การเขียนรับค่าข้อมูลคุณสมบัติของผู้สมัครและส่งค่า parameter ผ่าน API ไปยังฟังก์ชั่นหลังบ้าน (Request) ดังรูป ข-3

```
const register = async (userData) => {

  //console.log(userData)
  try {
    const response = await api.post('/predict', userData);
    //console.log(response.data);
    fetchList();

  } catch (error) {
      console.error("Registration error:", error);
      throw error;
  }
};

```

รูปที่ ข-3 การส่งค่าไปยังหลังบ้าน

1.4 การเขียนแสดงผลการพิจารณาที่ได้รับการคืนค่าจากหลังบ้าน (Response) เมื่อได้ผลลัพธ์จากการทำนายของแบบจำลองแล้วดังรูป ข-4

```
function Row(props) {
  const { row } = props;
  const [open, setOpen] = React.useState(false);

  return (
      <React.Fragment>
          <TableRow sx={{ '& > *': { borderBottom: 'unset' } }}>
              <TableCell>
                  <IconButton
                      aria-label="expand row"
                      size="small"
                      onClick={() => setOpen(!open)}
                  >
                      {open ? <KeyboardArrowUpIcon /> :
                      <KeyboardArrowDownIcon />}
                  </IconButton>
              </TableCell>
              <TableCell component="th" scope="row">
              {row.app_id}
              </TableCell>
              <TableCell align="right">Credit Card</TableCell>
            <TableCell align="right">Visa Platinum</TableCell>
            <TableCell align="right">N01</TableCell>
            <TableCell align="right">
            <Typography
                  className=""
                  style={{
                      backgroundColor:
                      ((row.result === 'APPROVE' && 'green') ||
                      (row.result === 'REJECT' && 'red')),
                      color:'white',
                      fontWeight: 'bold',
                      fontSize: '0.75rem',
                      borderRadius: 8,
                      padding: '3px 10px',
                      display: 'inline-block'
                  }}
                >{row.result}</Typography>
              </TableCell>
            <TableCell align="right"

              >{row.date_process}</TableCell>
            </TableRow>
            <TableRow>
                <TableCell style={{ paddingBottom: 0,
                    paddingTop: 0 }} colSpan={6}>
                    <Collapse in={open} timeout="auto" unmountOnExit>
                        <Box sx={{ margin: 1 }}>
                            <Typography variant="h6"
                                gutterBottom component="div">
                                Detail
                            </Typography>
                            <Table size="small"
                                aria-label="purchases">
                                <TableHead>
                                    <TableRow>
                                        <TableCell>
                                            Rule No.
                                        </TableCell>
                                        <TableCell align="right">
                                            Rule
                                        </TableCell>
                                        <TableCell align="right">
                                            value
                                        </TableCell>
                                    </TableRow>
                                </TableHead>
                                <TableBody>
                                    {row.allRule.map
                                        ((allRule, index) => (
                                        <TableRow key=
                                            {index}>
                                            <TableCell
                                                component="th"
                                                scope="row">
                                                {allRule.rule_id}
                                            </TableCell>
                                            <TableCell align="right">
                                                {allRule.rule_name}
                                            </TableCell>
                                            <TableCell align="right">
                                                {allRule.value}
                                            </TableCell>
                                        </TableRow>
                                    ))}
                                </TableBody>
                            </Table>
                        </Box>
                    </Collapse>
                </TableCell>
            </TableRow>
        </React.Fragment>
    );
}
```

รูปที่ ข-4 การแสดงค่าที่ได้รับจากหลังบ้าน

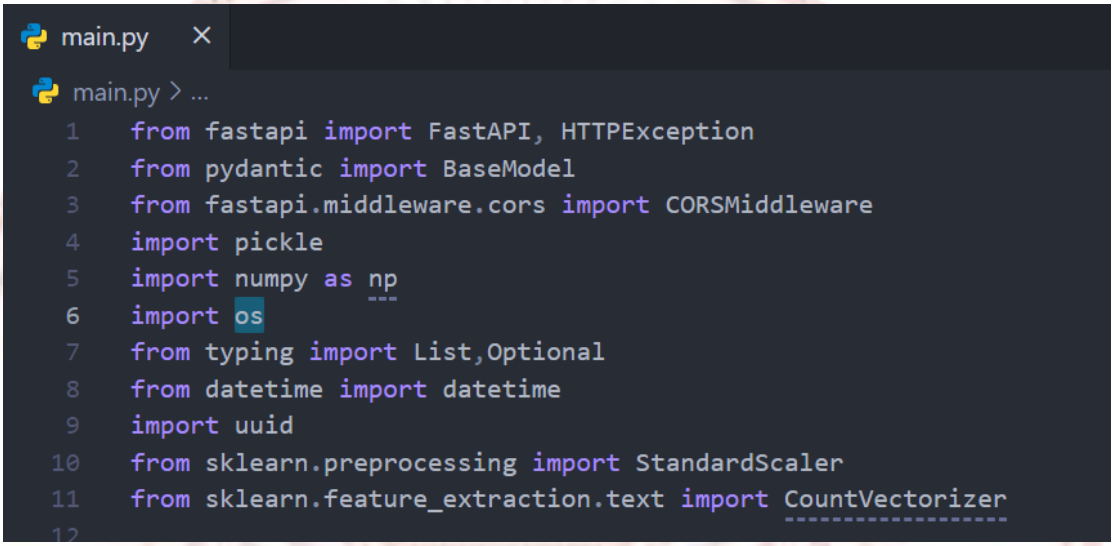
1.5 การ run แอปพลิเคชันหน้าบ้านเพื่อใช้งานดังรูป ข-5

```
PS D:\predict_credit_card\predict-credit-card-fontend> cd .\predict-credit-card-fontend\
PS D:\predict_credit_card\predict-credit-card-fontend\predict-credit-card-fontend> npm run dev

> predict-credit-card-fontend@0.0.0 dev
> vite


  VITE v6.2.1  ready in 740 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

รูปที่ ข-5 run แอปพลิเคชันหน้าบ้าน

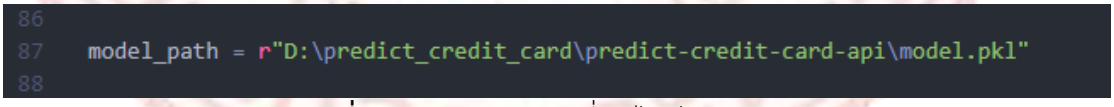
2. หลังบ้าน (backend-end)

2.1 เนื่องจากการแอปพลิเคชันการพัฒนาสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต หลังบ้าน (backend-end) มีการพัฒนาด้วยภาษาคอมพิวเตอร์ Python จึงเรียกใช้ไลบรารี่ (Library) numpy, sklearn และ os ดังรูป ข-6

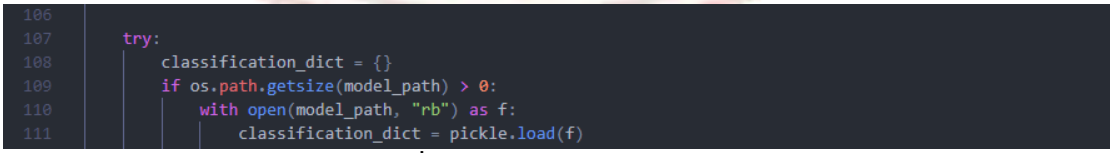


รูปที่ ข-6 การอิมพอร์ตไลบรารี่เพื่อทำสร้างระบบหลังบ้าน

2.2 ทำการเรียกใช้ไฟล์แบบจำลองชื่อไฟล์ model.pkl นามสกุล .pkl มาใช้งาน ดังรูป ข-7 และ ข-8



รูปที่ ข-7 กำหนด path ที่วางไฟล์แบบจำลอง



รูปที่ ข-8 เรียกใช้ไฟล์แบบจำลอง

2.3 สร้างฟังก์ชันเพื่อทำนายผลการอนุมัติบัตรเครดิต โดยทำการรับ parameter จากหน้าบ้าน (Request) ที่ส่งเข้ามาในฟังก์ชั่น และแปลงข้อมูลอยู่ในรูปแบบบรรทัดฐานเดียวกัน (normalization) ให้เหมาะสมในการเอาเข้าแบบจำลองเพื่อทำนายผล และทำการคืนค่าไปยังหน้าบ้าน (Response) เพื่อแสดงผล ดังรูป ข-9, ข-10, ข-11, ข-12, ข-13 และ ข-14

```python
@app.post("/predict")
def create_item(req: Rule):
```

รูปที่ ข-9 เรียกฟังก์ชันและรับ parameter

```python
class Rule(BaseModel):
    V2_Age: Optional[float]
    V3_TimeInJob:   Optional[float]
    V4_Income:  Optional[float]
    V5_DepRefType:  Optional[float]
    V6_DepRefAmt:   Optional[float]
    V7_ExistingDTI: Optional[float]
    V8_TotalDTI:    Optional[float]
    V9_CreditLimit: Optional[float]
    V10_CreditLimitGMI: Optional[float]
    V11_CreditLimitDepRef:  Optional[float]
    V12_TotalExposureLimitAmt:  Optional[float]
    V13_TotalExposureLimitGMT:  Optional[float]
    V14_MaxMainCard:    Optional[float]
    V15_MaxSupCard: Optional[float]
    V16_DupCardType:    Optional[float]
    V17_AML:    Optional[float]
    V18_KYC:    Optional[float]
    V19_SLL:    Optional[float]
    V20_WriteOff:   Optional[float]
    V21_Fraud:  Optional[float]
    V22_Bankruptcy: Optional[float]
    V23_NCBAcctStatus:  Optional[float]
    V24_NCBAcctType:    Optional[float]
    V25_NCBOpenDateMonth:   Optional[float]
    V26_NCBTDRDateMonth:    Optional[float]
    V27_NCBOverdueCurrMonth:    Optional[float]
    V28_NCBDlq6m:   Optional[float]
    V29_NCBDlq12m:  Optional[float]
    V30_NCBDlq24m:  Optional[float]
    V31_NCBOverLimit6m: Optional[float]
```

รูปที่ ข-10 รับค่า parameter

```
data_dict = req.model_dump()

numerical_columns = ['V2_Age', 'V3_TimeInJob', 'V4_Income',
                     'V5_DepRefType', 'V6_DepRefAmt', 'V7_ExistingDTI', 'V8_TotalDTI',
                     'V9_CreditLimit', 'V10_CreditLimitGMI', 'V11_CreditLimitDepRef',
                     'V12_TotalExposureLimitAmt', 'V13_TotalExposureLimitGMT',
                     'V14_MaxMainCard', 'V15_MaxSupCard', 'V16_DupCardType', 'V17_AML',
                     'V18_KYC', 'V19_SLL', 'V20_WriteOff', 'V21_Fraud', 'V22_Bankruptcy',
                     'V23_NCBAcctStatus', 'V24_NCBAcctType', 'V25_NCBOpenDateMonth',
                     'V26_NCBTDRDateMonth', 'V27_NCBOverdueCurrMonth', 'V28_NCBDlq6m',
                     'V29_NCBDlq12m', 'V30_NCBDlq24m', 'V31_NCBOverLimit6m']

scaler=StandardScaler()

features = [data_dict['V2_Age'], data_dict['V3_TimeInJob'], data_dict['V4_Income'],
            data_dict['V5_DepRefType'], data_dict['V6_DepRefAmt'], data_dict['V7_ExistingDTI'],
            data_dict['V8_TotalDTI'],data_dict['V9_CreditLimit'], data_dict['V10_CreditLimitGMI'],
            data_dict['V11_CreditLimitDepRef'],data_dict['V12_TotalExposureLimitAmt'],
            data_dict['V13_TotalExposureLimitGMT'],data_dict['V14_MaxMainCard'],
            data_dict['V15_MaxSupCard'], data_dict['V16_DupCardType'], data_dict['V17_AML'],
            data_dict['V18_KYC'],  data_dict['V19_SLL'], data_dict['V20_WriteOff'],
            data_dict['V21_Fraud'], data_dict['V22_Bankruptcy'], data_dict['V23_NCBAcctStatus'],
            data_dict['V24_NCBAcctType'],   data_dict['V25_NCBOpenDateMonth'],
            data_dict['V26_NCBTDRDateMonth'], data_dict['V27_NCBOverdueCurrMonth'],
            data_dict['V28_NCBDlq6m'], data_dict['V29_NCBDlq12m'], data_dict['V30_NCBDlq24m'],
            data_dict['V31_NCBOverLimit6m']]

x=scaler.fit_transform([features])
```

รูปที่ ข-11 แปลงข้อมูลอยู่ในรูปแบบบรรทัดฐานเดียวกัน

```
142
143         prediction = classification_dict.predict(x)[0]
144         print(prediction)
145
146         result = 'APPROVE' if prediction > 0.5 else 'REJECT'
```

รูปที่ ข-12 การทำนายผล

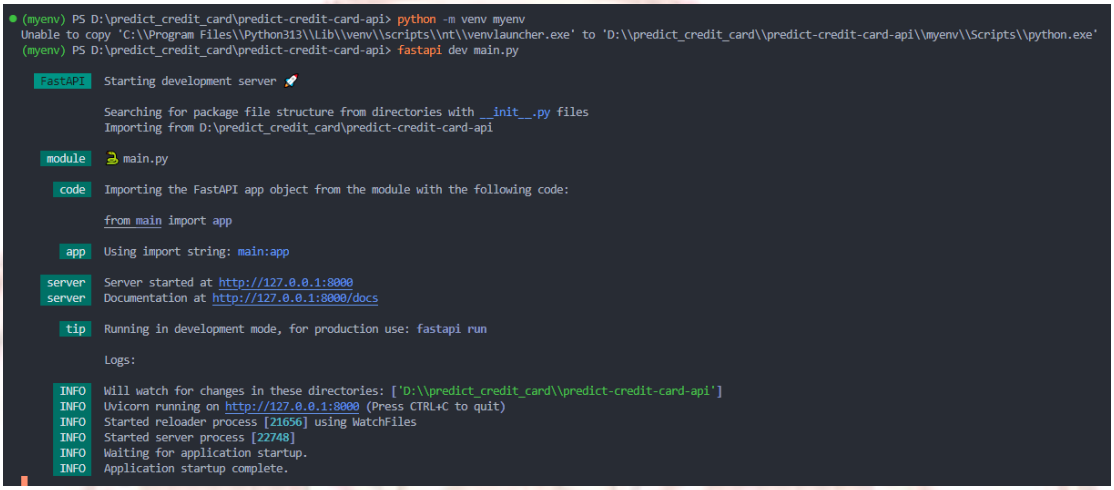
```
            AllRuleList = [];
            AllRuleList.append(AllRule(rule_id="2",rule_name="V2_Age",value=data_dict['V2_Age']))
            AllRuleList.append(AllRule(rule_id="3",rule_name="V3_TimeInJob",value=data_dict['V3_TimeInJob']))
            AllRuleList.append(AllRule(rule_id="4",rule_name="V4_Income",value=data_dict['V4_Income']))
            AllRuleList.append(AllRule(rule_id="5",rule_name="V5_DepRefType",value=data_dict['V5_DepRefType']))
            AllRuleList.append(AllRule(rule_id="6",rule_name="V6_DepRefAmt",value=data_dict['V6_DepRefAmt']))
            AllRuleList.append(AllRule(rule_id="7",rule_name="V7_ExistingDTI",value=data_dict['V7_ExistingDTI']))
            AllRuleList.append(AllRule(rule_id="8",rule_name="V8_TotalDTI",value=data_dict['V8_TotalDTI']))
            AllRuleList.append(AllRule(rule_id="9",rule_name="V9_CreditLimit",value=data_dict['V9_CreditLimit']))
            AllRuleList.append(AllRule(rule_id="10",rule_name="V10_CreditLimitGMI",value=data_dict['V10_CreditLimitGMI']))
            AllRuleList.append(AllRule(rule_id="11",rule_name="V11_CreditLimitDepRef",value=data_dict['V11_CreditLimitDepRef']))
            AllRuleList.append(AllRule(rule_id="12",rule_name="V12_TotalExposureLimitAmt",value=data_dict['V12_TotalExposureLimitAmt']))
            AllRuleList.append(AllRule(rule_id="13",rule_name="V13_TotalExposureLimitGMI",value=data_dict['V13_TotalExposureLimitGMI']))
            AllRuleList.append(AllRule(rule_id="14",rule_name="V14_MaxMainCard",value=data_dict['V14_MaxMainCard']))
            AllRuleList.append(AllRule(rule_id="15",rule_name="V15_MaxSupCard",value=data_dict['V15_MaxSupCard']))
            AllRuleList.append(AllRule(rule_id="16",rule_name="V16_DupCardType",value=data_dict['V16_DupCardType']))
            AllRuleList.append(AllRule(rule_id="17",rule_name="V17_AML",value=data_dict['V17_AML']))
            AllRuleList.append(AllRule(rule_id="19",rule_name="V18_KYC",value=data_dict['V18_KYC']))
            AllRuleList.append(AllRule(rule_id="19",rule_name="V19_SLL",value=data_dict['V19_SLL']))
            AllRuleList.append(AllRule(rule_id="20",rule_name="V20_WriteOff",value=data_dict['V20_WriteOff']))
            AllRuleList.append(AllRule(rule_id="21",rule_name="V21_Fraud",value=data_dict['V21_Fraud']))
            AllRuleList.append(AllRule(rule_id="22",rule_name="V22_Bankruptcy",value=data_dict['V22_Bankruptcy']))
            AllRuleList.append(AllRule(rule_id="23",rule_name="V23_NCBAcctStatus",value=data_dict['V23_NCBAcctStatus']))
            AllRuleList.append(AllRule(rule_id="24",rule_name="V24_NCBAcctType",value=data_dict['V24_NCBAcctType']))
            AllRuleList.append(AllRule(rule_id="25",rule_name="V25_NCBOpenDateMonth",value=data_dict['V25_NCBOpenDateMonth']))
            AllRuleList.append(AllRule(rule_id="26",rule_name="V26_NCBTDRDateMonth",value=data_dict['V26_NCBTDRDateMonth']))
            AllRuleList.append(AllRule(rule_id="27",rule_name="V27_NCBOverdueCurrMonth",value=data_dict['V27_NCBOverdueCurrMonth']))
            AllRuleList.append(AllRule(rule_id="28",rule_name="V28_NCBDlq6m",value=data_dict['V28_NCBDlq6m']))
            AllRuleList.append(AllRule(rule_id="29",rule_name="V29_NCBDlq12m",value=data_dict['V29_NCBDlq12m']))
            AllRuleList.append(AllRule(rule_id="30",rule_name="V30_NCBDlq24m",value=data_dict['V30_NCBDlq24m']))
            AllRuleList.append(AllRule(rule_id="31",rule_name="V31_NCBOverLimit6m",value=data_dict['V31_NCBOverLimit6m']))


            id=str(uuid.uuid4().hex)
            res = Response(app_id=id, response_code="ok", response_desc="success",rule=[req], result=result, date_process=datetime.now(), allRule=AllRuleList
)
            memory_db["responseList"].append(res)
            return res;
        else :
            return HTTPException(status_code=999, detail="no model");

    except Exception as e:
        return HTTPException(status_code=500, detail=f"Error occurred: {e}");

```

รูปที่ ข-13 การคืนค่าไปยังหน้าบ้าน

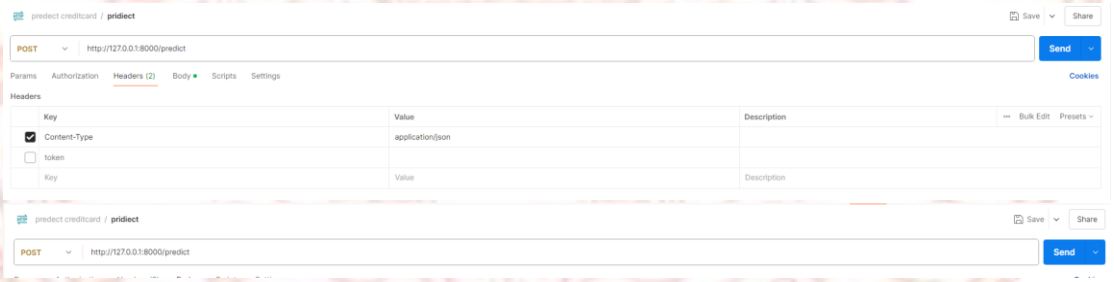
```
class AllRuleList(BaseModel):
    allRuleList: List[AllRule]

class Response(BaseModel):
    app_id: Optional[str]
    response_code: Optional[str]
    response_desc: Optional[str]
    rule:  Optional[List[Rule]] = []
    result: Optional[str]
    date_process: Optional[datetime]
    allRule:  Optional[List[AllRule]] = []
```

รูปที่ ข-14 รูปแบบที่คืนค่า

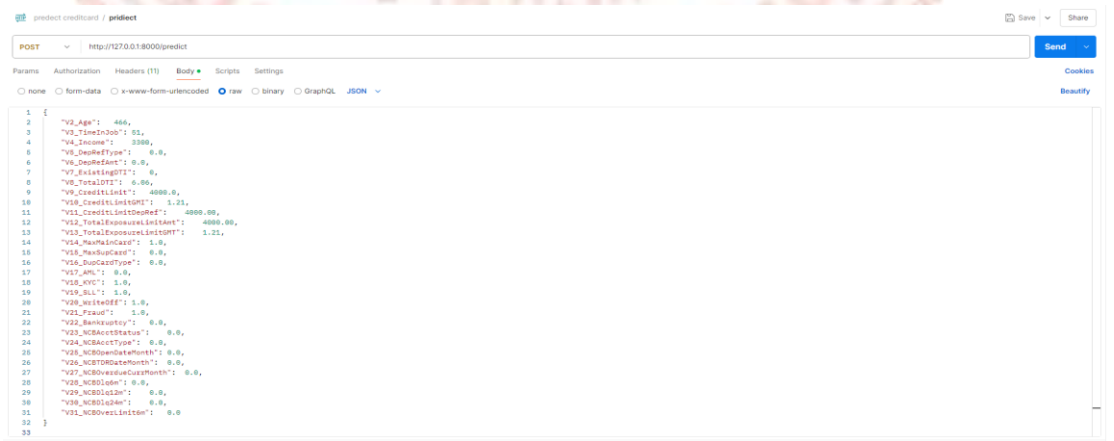
2.4 การทดสอบ API หลังบ้าน (backend-end) ด้วยเครื่องมือ Postman ดังรูป ข-15, ข-16, ข-17 และ ข-18



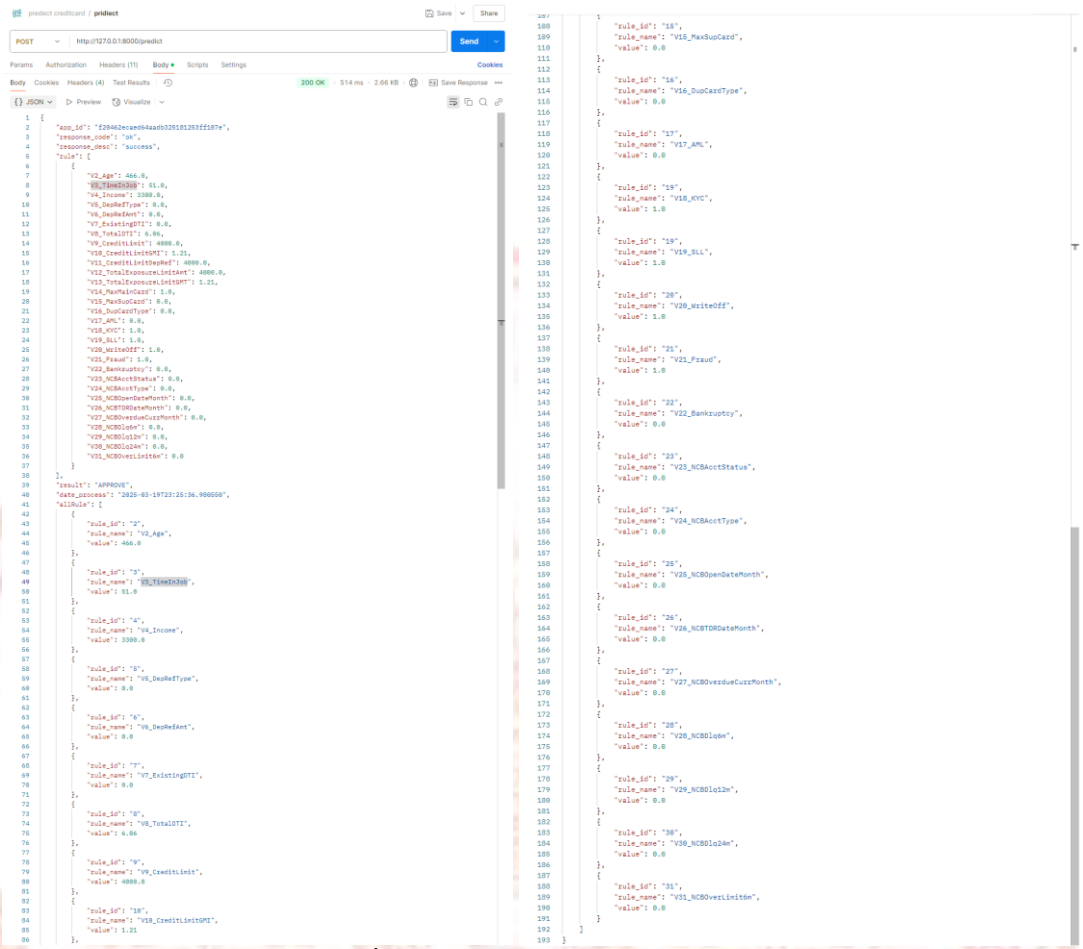
รูปที่ ข-15 run แอปพลิเคชันหลังบ้าน



รูปที่ ข-16 ตั้งค่าเครื่องมือ Postman



รูปที่ ข-17 กำหนด parameter เสมือนรับจากหน้าบ้าน



รูปที่ ข-18 ค่าเสมือนคืนไปยังหน้าบ้าน

ประวัติผู้เขียน

ชื่อ : นายจิรภัทร ศรีอภิรัฐ

ชื่อการค้นคว้าอิสระ : ระบบสนับสนุนการตัดสินใจอนุมัติบัตรเครดิต ด้วยวิธีการเรียนรู้ของเครื่อง

สาขาวิชา : วิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ประวัติ

เกิดวันอาทิตย์ที่ 31 ตุลาคม พ.ศ. 2536

ประวัติการศึกษา

พ.ศ. 2562 สำเร็จการศึกษาระดับปริญญาตรี
สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยีสารสนเทศ
มหาวิทยาธนบุรี

ประการทำงาน

ปัจจุบันทำงานอยู่ที่บริษัท แอคเซลเล้นซ์ (ประเทศไทย) จำกัด ตำแหน่ง JAVA Developer

สถานที่ติดต่อ

43/323 หมู่บ้านทวีทอง
เพชรเกษม 110 เขตหนองแขม แขวงหนองค้างพลู
จังหวัดกรุงเทพฯ 10160
อีเมล jamechocolatecake@gmail.com