การพยากรณ์ผลการแข่งขันฟุตบอลพรีเมียร์ลีกอังกฤษด้วยการจัดหมู่และการถดถอย

PREDICTING PREMIER LEAGUE OUTCOMES USING CLASSSIFICATION AND REGRESSION

พิศิษฏ์ ชินกุลประสาน
PISIT CHINGULPRASAN

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์
ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2568

KMITL-2025-SC-M-017-040

PREDICTING PREMIER LEAGUE OUTCOMES USING CLASSSIFICATION AND REGRESSION

PISIT CHINGULPRASAN

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

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

2025

KMITL-2025-SC-M-017-040

หัวข้อการค้นคว้าอิสระ การพยากรณ์ผลการแข่งขันฟุตบอลพรีเมียร์ลีกด้วยการจัดหมวดหมู่และการถดถอย
นักศึกษา นาย พิศิษฏ์ ชินกุลประสาน
รหัสนักศึกษา 66056059
ปริญญา วิทยาศาสตรมหาบัณฑิต(วิทยาการข้อมูลและการวิเคราะห์)
ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
พ.ศ. 2568
อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร. ณชญาดา กมลมิธิชม

บทคัดย่อ

ผู้วิจัยได้เห็นถึงความท้าทาย และโอกาสในการนำความรู้มาประยุกต์ใช้กับข้อมูลเกี่ยวกับกีฬาที่ได้รับความนิยมสูงอย่างฟุตบอลพรีเมียร์ลีกอังกฤษ งานวิจัยนี้จึงถูกริเริ่มขึ้นโดยมีวัตถุประสงค์หลักเพื่อใช้ประโยชน์จากข้อมูลสถิติการแข่งขันในการสร้างแบบจำลองด้วยเทคนิคการเรียนรู้ของเครื่องจักร (Machine Learning) สำหรับการทำนายจำนวนประตู (Goals For) และผลการแข่งขัน (Match Result) โดยเชื่อมั่นว่าการวิเคราะห์ข้อมูลเชิงลึกด้วยวิธีการทางสถิติ และการเรียนรู้ของเครื่องจักร จะช่วยให้สามารถเข้าใจปัจจัยที่มีผลต่อผลการแข่งขัน และจำนวนประตู

การดำเนินงานวิจัยตั้งแต่การรวบรวม และเตรียมข้อมูลอย่างเป็นระบบ การทำความสะอาดและแปลงข้อมูลให้พร้อมสำหรับการวิเคราะห์ การคัดเลือกคุณลักษณะที่มีอิทธิพลต่อผลลัพธ์ด้วยหลากหลายเทคนิค (สหสัมพันธ์ (Correlation), ข้อมูลร่วมกัน (Mutual Information) และความสำคัญของคุณลักษณะ (Feature Importance)) ไปจนถึงการสร้าง และประเมินประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องจักรทั้งแบบจำลองการถดถอยปัวซอง และแบบจำลองการจำแนกประเภท (Decision Tree, Random Forest, XGBoost และ LightGBM) ผลลัพธ์ที่ได้จากการศึกษาได้ชี้ให้เห็นถึงศักยภาพของแบบจำลอง และความสำคัญของการเลือกคุณลักษณะที่เหมาะสม รวมถึงการปรับปรุงประสิทธิภาพของแบบจำลองผ่านการจูนค่าพารามิเตอร์ และการเพิ่มพารามิเตอร์สังเคราะห์ เพื่อให้การทำนายผลการแข่งขันมีความแม่นยำ และครอบคลุมยิ่งขึ้น

คำสำคัญ : การเรียนรู้ของเครื่องจักร, จำนวนประตู, ผลการแข่งขัน, Correlation, Decision Tree, Feature Importance, LightGBM, Mutual Information, Random Forest, XGBoost

Independent Study Title	Predicting Premier League Outcomes using classification and regression
Student Name	Pisit Chingulprasan
Student ID	66056059
Degree	Master of Science (Data Science and Analytics) KMITL Digital Analytics and Intelligence Center
Year	2025
Advisor	Assoc. Prof. Dr. Nachayadar Kamolmitisom

ABSTACT

This research was initiated to address the challenges and opportunities in applying knowledge to highly popular sports data, specifically English Premier League football. Our primary objective is to leverage statistical match data to build Machine Learning models for predicting Goals For and Match Results. We believe that in-depth data analysis using statistical methods and machine learning will provide a deeper understanding of the factors influencing match outcomes and goals.

The research methodology encompassed systematic data collection and preparation, data cleaning and feature engineering into a ready for prediction, and the selection of features through various techniques, including correlation, mutual information, and feature importance. This was followed by the creation and performance evaluation of diverse machine learning models, including Poisson regression and classification models (Decision Tree, Random Forest, XGBoost, and LightGBM). The study's results highlighted the models' potential and the importance of appropriate feature selection, as well as demonstrating performance improvements through hyperparameter tuning and the addition of synthetic features to enhance the accuracy and comprehensiveness of match predictions.

Keywords: Correlation, Decision Tree, Feature Importance, Goals For, LightGBM, Machine Learning, Match Result, Mutual Information, Random Forest, XGBoost

กิตติกรรมประกาศ

การศึกษาวิจัยเรื่อง "การทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีกด้วยการเรียนรู้ของเครื่องจักร" ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความกรุณาอย่างยิ่งจาก รศ.ดร.ณชญาดา กมลมิธิชม (อาจารย์ที่ปรึกษา) และคณะกรรมการทุกท่าน ที่ได้ให้คำแนะนำ คำปรึกษา และตรวจทาน แก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ตลอดระยะเวลาการดำเนินงานวิจัย ความรู้ และประสบการณ์อันทรงคุณค่าของท่านเป็นแรงผลักดัน และเป็นแนวทางสำคัญที่ทำให้งานวิจัยนี้มีความสมบูรณ์และเป็นไปตามวัตถุประสงค์ที่ตั้งไว้ ผู้วิจัยขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ข้าพเจ้าขอขอบคุณแหล่งข้อมูลสถิติฟุตบอลจากเว็บไซต์ fbref.com ที่ได้เอื้อเฟื้อข้อมูลอันเป็นประโยชน์อย่างยิ่งในการศึกษาและวิเคราะห์ และขอขอบคุณงานวิจัย และทฤษฎีต่างๆ ที่เกี่ยวข้อง ซึ่งได้ทำการศึกษา และอ้างอิงถึงในบทที่ 2 อันเป็นพื้นฐานสำคัญของงานวิจัยนี้

ข้าพเจ้าขอขอบคุณครอบครัวและเพื่อนๆ ที่คอยให้กำลังใจ และให้การสนับสนุนมาโดยตลอด ที่ให้คำปรึกษาด้านวิชาการ และให้กำลังใจ ทำให้ข้าพเจ้ามีความมุ่งมั่น ผ่านอุปสรรคจนสามารถดำเนินวิจัยได้อย่างลุล่วง

สุดท้ายนี้ข้าพเจ้าหวังเป็นอย่างยิ่งว่าผลการศึกษาวิจัยนี้จะเป็นประโยชน์ต่อผู้ที่สนใจ และสามารถนำไปต่อยอดในอนาคตได้

พิศิษฏ์ ชินกุลประสาน

สารบัญ

สารบัญ(ต่อ)

สารบัญตาราง

สารบัญตาราง(ต่อ)

สารบัญรูปภาพ

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

ในปัจจุบันกีฬาฟุตบอลเป็นกีฬาที่ได้รับความนิยมมากในโลก โดยเฉพาะพรีเมียร์ลีกอังกฤษ (Premier League) ซึ่งเป็นลีกฟุตบอลที่มีผู้ติดตามชม และให้ความสนใจอย่างแพร่หลายทั่วโลก การทำนายผลการแข่งขันผลฟุตบอลจึงเป็นเรื่องที่ได้ความสนใจจากทั้งวงการกีฬา และวงการภาคธุรกิจ เช่นการวิเคราะห์ผลการแข่งขัน, การวิเคราะห์ข้อมูลแผนการเล่นของทีมนั้นๆ หรือไม่ว่าจะเป็นการเดิมพันทางด้านธุรกิจ ซึ่งในปัจจุบันนี้การพัฒนาการเล่น ไม่ว่าจะเป็นข้อมูลวิธีการเล่น ระบบการเล่น สถิติต่างๆ เป็นเครื่องมือสำคัญที่ช่วยสนับสนุนการพัฒนาทางด้านกีฬา และการตัดสินใจในหลายมิติด้วยการเติบโตของเทคโนโลยี และข้อมูลขนาดใหญ่ (Big Data) ทำให้เกิดข้อมูล และสถิติต่างๆที่เกี่ยวข้องกับฟุตบอลจำนวนมาก การใช้ความรู้ทางการเรียนรู้ของเครื่องจักร(Machine Learning) ทำให้สามารถวิเคราะห์ และประมวลผลข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ และเทคโนโลยีในปัจจุบันทำให้เกิดการสร้างแบบจำลอง เพื่อใช้ทำนายผลการแข่งขัน ซึ่งเป็นสมมุติฐานที่น่าสนใจ และท้าทาย เนื่องจากการทำนายผลการแข่งขันมีความซับซ้อนของข้อมูลหลายอย่าง เช่น สถิติการแข่งขัน ฟอร์มการเล่น และปัจจัยภายนอกที่ยากจะคาดเดา โดยการวิจัยนี้จะมุ่งเน้นในการนำเทคนิคของเรียนรู้ของเครื่องจักรมาประยุกต์ใช้เพื่อทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีก โดยอ้างอิงจากข้อมูล และสถิติต่างๆ

1.2 วัตถุประสงค์

1) เพื่อสร้างแบบจำลองของเรียนรู้ของเครื่องจักรที่สามารถทำนายจำนวนประตู

2) เพื่อสร้างแบบจำลองของเรียนรู้ของเครื่องจักรที่สามารถทำนายผลการแข่งขันฟุตบอล

3) เพื่อศึกษา และเปรียบเทียบแบบจำลองของการเรียนรู้ของเครื่องจักรในประเภทการถดถอย (Regression) และการจำแนกประเภท (Classification) ในการทำนายจำนวนประตูผลการแข่งขัน

4) เพื่อวิเคราะห์หาปัจจัยและคุณลักษณะต่างๆ ที่มีผลต่อการการทำนายจำนวนประตู และผลการแข่งขัน

1.3 ขอบเขตงานวิจัย

ในการวิจัยครั้งนี้ ได้รวบรวมข้อมูลทุติยภูมิทางสถิติที่เกี่ยวข้องจากการแข่งขันฟุตบอลพรีเมียร์ลีกตั้งแต่ฤดูกาล 2016-2017 จนถึงฤดูกาล 2023-2024 จากเว็บไซต์ fbref.com ข้อมูลที่ใช้ประกอบด้วยสถิติเชิงลึกด้านต่างๆ เช่น การยิงประตู การส่งบอล การป้องกัน การครองบอล การสร้างโอกาสในการยิง และการทำประตู ข้อมูลถูกรวบรวมและรวมเข้าด้วยกันโดยใช้ Match ID ในการเตรียมข้อมูล มาทำเป็นตารางโดยใช้ Library PANDAS จากนั้นจะทำการเตรียมข้อมูลให้พร้อม (Data Preparation) โดยการทำความสะอาด (Data Cleaning) ข้อมูลต่างๆ ด้วยโปรแกรมไพธอน (Python) และเลือกคุณลักษณะต่างๆที่สำคัญต่อการสร้างแบบจำลองการเรียนรู้ของเครื่องจักร ที่ใช้ทำนายผลการแข่งขัน และทดสอบประสิทธิภาพของแบบจำลอง เพื่อประเมินความแม่นยำ โดยการเปรียบเทียบการทำนายโดยใช้การเรียนรู้แบบมีผู้สอน (Supervised Learning) ด้วยวิธีการถดถอย (โดยการทำนายประตูที่ยิงได้โดยใช้ข้อมูลต่างๆ) และการจำแนก (โดยการทำนายผลของการแข่งขัน เช่น ชนะ แพ้ และเสมอ) เพื่อหาว่าการทำนายวิธีไหนที่ให้ผลลัพธ์ที่ดีที่สุด และพัฒนาแบบจำลองโดยการใช้การปรับจูนไฮเปอร์พารามิเตอร์ของแบบจำลอง และการเพิ่มคุณลักษณะสังเคราะห์

1.4 วิธีการดำเนินวิจัย

ในการสร้างแบบจำลองเพื่อทำนายจำนวนประตู และผลการแข่งขันด้วยการเรียนรู้ของเครื่องจักร ผู้วิจัยได้วางแผนขั้นตอนในการดำเนินการศึกษา เพื่อให้ได้ตามวัตถุประสงค์ดังนี้

1) กำหนดหัวข้อการวิจัย วางแผนการดำเนินงาน และเขียนเค้าโครง

2) ศึกษางานวิจัยที่เกี่ยวข้อง เพื่อกำหนดสมมติฐาน

3) ดำเนินการเก็บ และรวบรวมตารางสถิติต่างๆข้อมูลจาก fbref.com โดยใช้ไพธอน

4) สำรวจข้อมูล และเตรียมข้อมูลด้วยการทำความสะอาดข้อมูล

5) วิเคราะห์ข้อมูล และจัดทำการเลือกคุณลักษณะ (Feature Selection) เพื่อสร้างแบบจำลอง

6) สร้าง และทดสอบแบบจำลองการเรียนรู้ของเครื่องจักรทั้งแบบจำลองการถดถอย และแบบจำลองการจำแนกประเภท

7) ประเมินประสิทธิภาพของแบบจำลอง และเปรียบเทียบเพื่อหาแบบจำลองที่เหมาะสม

8) พัฒนาแบบจำลองให้มีประสิทธิภาพดีขึ้น

9) สรุปผลการวิจัย และนำเสนอแนวทางการพัฒนาต่อไปในอนาคต

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ผู้วิจัยสามารถสร้างแบบจำลองที่ใช้ทำนายจำนวนประตู หรือผลการแข่งขันฟุตบอลด้วยการเรียนรู้แบบเครื่องจักร ทั้งแบบจำลองการถดถอย และแบบจำลองการจำแนกประเภท ที่มีประสิทธิภาพ และนำไปประยุกต์ใช้กับกีฬา หรือการแข่งขันประเภทอื่นๆ เช่น การแข่งขันเกมส์ (E-Sport) หรือ การแข่งขันบาสเก็ตบอล (NBA Basketball)

2) การวิจัยนี้ช่วยให้ผู้วิจัยเข้าใจถึงการวิเคราะห์ข้อมูลเชิงสถิติ ปัจจัยที่มีผล และการประยุกต์ใช้การเรียนรู้แบบเครื่องจักรในการทำนายผลการแข่งขันฟุตบอล

บทที่ 2

ทฤษฎีและการวิจัยที่เกี่ยวข้อง

การศึกษา และวิจัยอิสระในหัวข้อ “การทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีกด้วยการเรียนรู้ของเครื่องจักร” ผู้วิจัยได้รวบรวม แนวคิด และทฤษฎีที่เกี่ยวข้อง โดยรายละเอียดหัวข้อมีดังนี้

2.1 ประวัติของฟุตบอลพรีเมียร์ลีก

2.2 การเรียนรู้ของเครื่องจักร (Machine Learning)

2.3 การถดถอยเชิงเส้น (Linear Regression)

2.4 การถดถอยปัวซอง (Poisson Regression)

2.5 การจำแนกประเภทด้วยต้นไม้ตัดสินใจ (Decision Tree Classification)

2.6 การจำแนกประเภทด้วยป่าแบบสุ่ม (Random Forest Classification)

2.7 การจำแนกประเภทด้วยเอ็กซ์ตรีมเกรเดียนท์บูสติ้ง (XGBoost)

2.8 การจำแนกประเภทด้วยไลต์จีบีเอม (LightGBM)

2.9 การตรวจสอบค่าสุดโต่ง หรือค่าที่ผิดปกติ (Outlier Detection)

2.10 การคัดเลือกคุณลักษณะที่สำคัญ (Feature Selection)

2.11 การจัดการกับข้อมูลที่ไม่สมดุล (Imbalanced Data)

2.12 การประเมินประสิทธิภาพของแบบจำลอง (Model Evaluation)

2.13 การปรับจูนไฮเปอร์พารามิเตอร์แบบจำลอง (Hyperparameter Tuning)

2.14 วิจัยที่เกี่ยวข้อง (Related Paper/Related Research)

2.1 ประวัติของฟุตบอลพรีเมียร์ลีก

ฟุตบอลพรีเมียร์ลีก (Premier League) ถือเป็นลีกฟุตบอลที่ได้อยู่ในระดับสูงสุดของประเทศอังกฤษ (England) และเป็นหนึ่งในลีกที่ได้รับความนิยมสูงสุดในโลก ทั้งในแง่ของการแข่งขัน ผู้ชมทั่วโลก และรายได้ที่เกิดจากการถ่ายทอดสด โฆษณา หรือไม่ว่าจะเป็นผู้ให้การสนับสนุน โดยที่จุดเริ่มต้นของการแข่งขันฟุตบอลอังกฤษ เกิดขึ้นเมื่อปลายศตวรรษที่ 19 โดยปี ค.ศ. 1888 ได้มีการก่อตั้ง เดอะฟุตบอลลีก (The Football League) ขึ้นซึ่งเป็นลีกฟุตบอลอาชีพแห่งแรกของโลก โดยประกอบสโมสรที่ทำการแข่งขันเพียง 12 ทีมเท่านั้นในช่วงเริ่มต้น จากนั้นลีกก็เติบโตตามลำดับ

พรีเมียร์ลีกถือกำเนิดขึ้นในปี ค.ศ. 1992 โดยมีเหตุผลสำคัญมาจากการที่สโมสรชั้นนำต้องการเพิ่มรายได้ มีอิสระในการบริหารจัดการทางด้านการเงิน และการตลาดมากขึ้น การแยกตัวออกจากฟุตบอลลีกเดิม เป็นการสร้างลีกให้มีการแข่งขันที่สูงขึ้น ดึงดูดเงินลงทุนจากภาคส่วนต่างๆ ทั้งการถ่ายทอดสด และสปอนเซอร์ และกลายมาเป็นลีกที่สร้างรากฐานให้กับระบบลีกฟุตบอลในอังกฤษในปัจจุบัน

พรีเมียร์ลีกเติบโตอย่างรวดเร็ว กลายเป็นลีกที่มีผู้ชมมากที่สุดในโลก โดยมีการแข่งขันถ่ายทอดสดไปไม่น้อยกว่า 200 ประเทศ มีผู้เข้ารวมกว่า 4.7 พันล้านคนต่อฤดูกาล รายได้หลักของลีกมาจากลิขสิทธิ์ ถ่ายทอดสด การขายสินค้า และการตลาดระดับโลก โดยที่การเข้ามาของนักเตะต่างชาติ ผู้จัดการทีมต่างชาติ และพัฒนาการต่างๆทางด้านการกีฬา และธุรกิจทำให้พรีเมียร์ลีกกลายเป็นศูนย์กลางของฟุตบอลระดับโลก

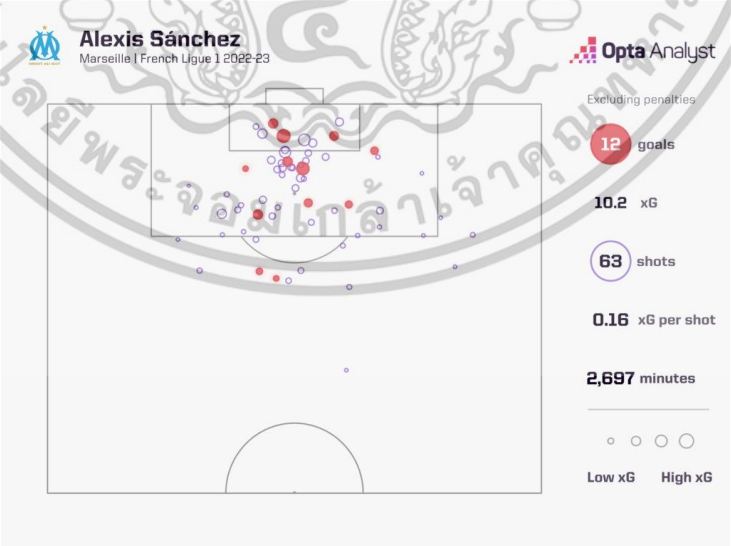
ภาพที่ 2.1 พัฒนาการของโลโก้พรีเมียร์ลีกตั้งแต่อดีตถึงปัจจุบัน
(ที่มา: https://turbologo.com/articles/premier-league-logo/)

2.1.1 ค่าที่เกี่ยวข้องกับสถิติที่เกี่ยวกับการใช้งานเครื่องจักร

ในยุคของฟุตบอลสมัยใหม่การวิเคราะห์สถิติขั้นสูง (Advanced Statistics) มีบทบาทสำคัญในการทำความเข้าใจ และประเมินประสิทธิภาพของทีม และผู้เล่น ค่าสถิติค่าความคาดหวังที่จะได้ประตู (Expected Goals: xG), ค่าความคาดหวังที่จะได้แอสซิสต์ (Expected Assists: xAG) และค่าความคาดหวังที่จะเสียประตู (Expected Goals Against: xGA) ซึ่งเป็นตัวนิยมที่ใช้อย่างกว้างขวางในฟุตบอลสมัยใหม่ และเป็นข้อมูลสำคัญที่สามารถนำมาใช้เป็นคุณลักษณะ ในการสร้างแบบจำลองการเรียนรู้ของเครื่องจักร เพื่อทำนายผลการแข่งขัน

1) ค่าความคาดหวังที่จะได้ประตู (Expected Goals: xG)

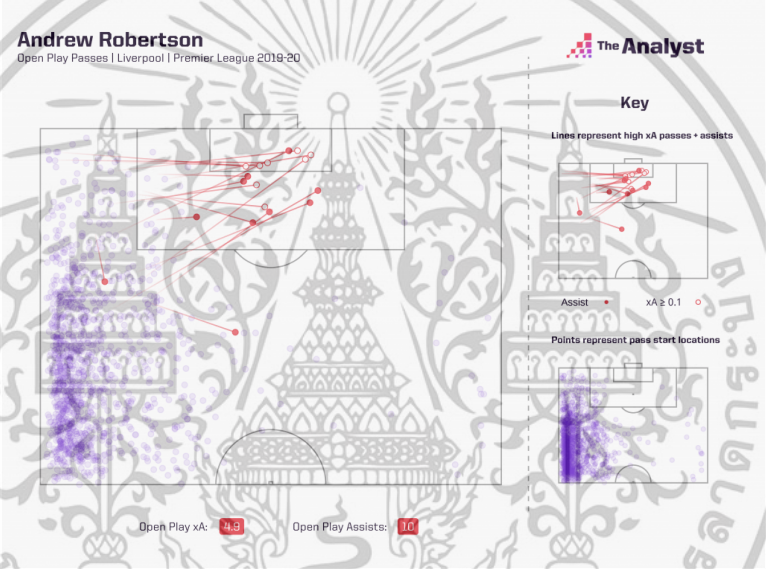
เป็นค่าสถิติที่ใช้วัดคุณภาพของโอกาสในการทำประตู โดยจะคำนวณความน่าจะเป็นในการยิงประตูจากตำแหน่ง และสถานการณ์ต่างๆ ซึ่งนำไปสู่การเป็นประตู โดยการคำนวณค่าความคาดหวังที่จะได้ประตู จะพิจารณาปัจจัยต่างๆ เช่น ระยะห่างจากประตู มุมในการยิง ประเภทของการยิง (หัว หรือเท้า) สถานการณ์ก่อนที่จะยิง หรือไม่ว่าจะเป็นจำนวนผู้เล่นที่ป้องกันในระหว่างการยิงประตู โดยความสำคัญของค่าความคาดหวังที่จะได้ประตู คือช่วยในการประเมินว่า สร้างโอกาสในการทำประตูได้มากน้อยเพียงใด และประสิทธิภาพในการเปลี่ยนโอกาสเป็นประตู เมื่อเทียบกับประตูที่เกิดขึ้นจริง



ภาพที่ 2.2 ตัวอย่างการแสดงค่าความน่าจะเป็นของการทำประตู (Expected Goals - xG)
(ที่มา: https://theanalyst.com/2023/08/what-is-expected-goals-xg)

2) ค่าความคาดหวังที่จะได้แอสซิสต์ (Expected Assists: xA)

เป็นค่าสถิติคล้ายกับค่าความคาดหวังที่จะได้ประตู แต่ใช้กับการจ่ายบอลที่นำไปสู่โอกาสในการทำประตู โดยจะวัดความน่าจะเป็นที่การจ่ายบอลจากตำแหน่งและสถานการณ์ต่างๆ จะนำไปสู่การยิงประตู ที่ควรจะนำไปสู่จะพิจารณาปัจจัยต่างๆ เช่น ประเภทของการจ่ายบอล (การเปิดบอล การแทงทะลุช่อง) ตำแหน่งของผู้จ่ายบอล และตำแหน่งของผู้รับบอล และสถานการณ์ก่อนการจ่ายบอล โดยความสำคัญของค่าความคาดหวังที่จะได้แอสซิสต์ คือช่วยในการประเมินการสร้างโอกาสให้เพื่อนร่วมทีมทำประตู และประสิทธิภาพของการจ่ายบอลที่นำไปสู่ประตู



ภาพที่ 2.3 ตัวอย่างการแสดงค่าความน่าจะเป็นของการแอสซิสต์ (Expected Assists - xA

(ที่มา: https://theanalyst.com/2021/03/what-are-expected-assists-xa/)

3) ค่าความคาดหวังที่เสียประตู (Expected Goals Against: xGA)

เป็นค่าสถิติที่วัดคุณภาพของโอกาสในการทำประตูที่ทีมเสียไป โดยจะคำนวณจากค่าความน่าจะเป็นที่โอกาสเหล่านั้นจะกลายเป็นประตูของคู่แข่ง โดยปัจจัยใช้ปัจจัยเดียวกับค่าความคาดหวังที่จะได้ประตุ แต่เป็นของอีกทีมนึง หรือทีมคู่แข่ง โดยความสำคัญของค่าความคาดหวังที่จะเสียประตู คือช่วยในการประเมินประสิทธิภาพในการป้องกันของทีม โดยดูว่าทีมมีโอกาสเสียประตูมากน้อยเพียงใดเมื่อเทียบกับประตูที่เกิดขึ้นจริง

2.2 การเรียนรู้ของเครื่องจักร (Machine Learning)

เป็นสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence) ที่มุ่งเน้นการพัฒนาอัลกอริทึม (Algorithm) ที่ช่วยให้คอมพิวเตอร์ หรือเครื่องจักรสามารถเรียนรู้จากชุดข้อมูล โดยที่ไม่ต้องเขียนโปรแกรมอย่างชัดเจน เฉพาะเจาะจง โดยกระบวนการเรียนรู้สามารถระบุ และทำความเข้าใจรูปแบบในชุดข้อมูล เพื่อนำไปสู่การทำนายหรือการตัดสินใจของเครื่องจักร โดยประเภทของการเรียนรู้ แบ่งออกเป็น

2.2.1 การเรียนรู้แบบมีผู้ฝึกสอน (Supervised Learning)

การเรียนรู้ของเครื่องจักรที่มีการใช้ข้อมูลแบบที่มีผลเฉลย หรือป้ายกำกับ (Label) โดยข้อมูลจะแบ่งเป็นคุณลักษณะ และผลเฉลยที่เป็นผลลัพธ์ที่ถูกต้อง แบบจำลองจะทำการเรียนรู้ และเปรียบเทียบผลลัพธ์ที่ทำนายผลออกมา เทียบกับผลเฉลยที่มีอยู่แล้ว ซึ่งการเรียนรู้แบบมีผู้ฝึกสอนแบ่งได้เป็น 2 วิธี คือ การถดถอย (Regression) ใช้กับชุดข้อมูลที่เป็นแบบค่าต่อเนื่อง (Continuous values) เช่น การทำนายราคาบ้าน การทำนายยอดขาย ส่วนอีกวิธีนึงเป็นการจำแนกประเภท (Classification) ใช้กับข้อมูลที่เป็นลักษณะการแบ่งกลุ่มหรือหมวดหมู่ (Categorical data) เช่น การจำแนกว่าข้อความเป็นสแปมหรือไม่ หรือไม่ว่าจะเป็นการทำนายผลการแข่งขันฟุตบอล

2.2.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

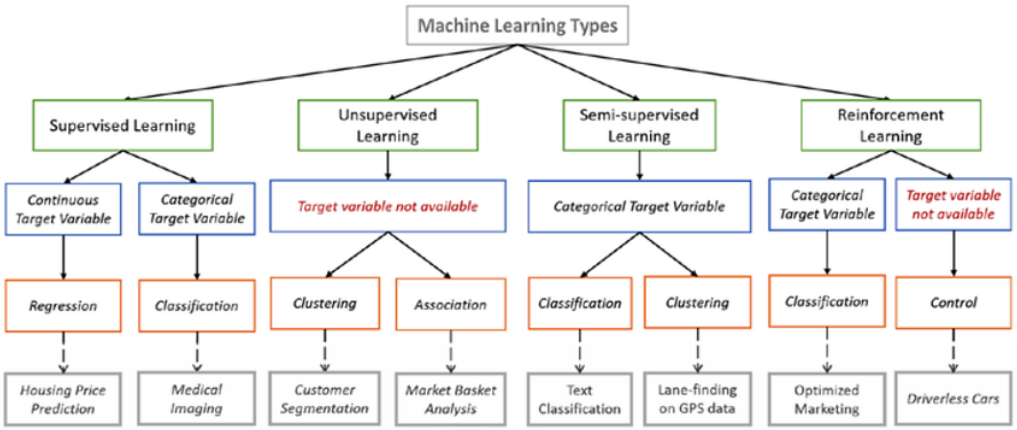
การเรียนรู้ของเครื่องจักรที่ไม่มีผลเฉลย หรือไม่มีป้ายกำกับ แบบจำลองจะต้องเรียนรู้ที่จะหาคำตอบ โครงสร้าง และรูปแบบของข้อมูลโดยไม่มีการแนะนำ โดยส่วนใหญ่จะใช้กับการจำแนกจัดกลุ่ม (Clustering) เช่น การจำแนกกลุ่มของลูกค้า การแบ่งกลุ่มจัดกลุ่มลูกค้า

2.2.3 การเรียนรู้แบบกึ่งสอน (Semi-Supervised Learning)

การเรียนรู้ของเครื่องจักรที่มีป้ายกำกับเพียงบางส่วน และข้อมูลที่ไม่มีป้ายกำกับช่วยในการเรียนรู้ เหมาะกับกรณีที่ข้อมูจำนวนมาก เช่น การจำแนกประเภทของเว็บไซต์ การจำแนกความรู้สึก

2.2.4 การเรียนรู้แบบเสริมแรง (Reinforcement Learning)

การเรียนรู้โดยให้แบบจำลองเรียนรู้ผ่าน การกระทำ (Action) และผลตอบแทน (Reward) โดยที่จะให้แบบจำลองทำการลองผิดลองถูกของตัวแบบจำลองเอง โดยมีการให้ผลตอบแทนเมื่อแบบจำลองทำถูกต้อง และมีการลงโทษ (Penalty) เมื่อแบบจำลองทำผิด จนกระทั่งไปถึงจุดที่แบบจำลองได้ผลลัพธ์ที่มีผลตอบแทนสูงที่สุด



ภาพที่ 2.4 แผนผังประเภทของการเรียนรู้ของเครื่อง

(ที่มา: https://ai.plainenglish.io/different-types-of-machine-learning-algorithms-28974016e108)

2.3 การถดถอยเชิงเส้น (Linear Regression)

แบบจำลองที่เป็นอัลกอริทึมการเรียนรู้ของเครื่องจักรแบบมีผู้สอน ที่ใช้สำหรับปัญหาการทำนายการถดถอย วัตถุประสงค์หลัก คือการสร้างแบบจำลองทางคณิตศาสตร์ที่แสดงความสัมพันธ์เชิงเส้นระหว่างตัวแปรอิสระ (Independent Variables) หนึ่งตัวหรือมากกว่า กับตัวแปรตาม (Dependent Variable) เพื่อใช้ทำนายค่าของตัวแปรตามจากข้อมูลตามตัวแปรอิสระ

2.3.1 การถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression)

การถดถอยเชิงเส้นอย่างอย่างเป็นกรณีที่มีตัวแปรอิสระเพียงหนึ่งตัว โดยสมการของแบบจำลองจะอยู่ในรูปของสมการ

$$Y = \beta_0 + \beta_1 X + \epsilon$$

โดยที่:

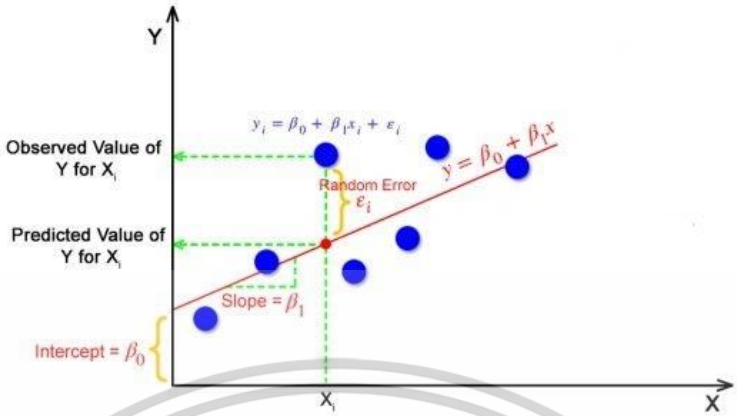
Y คือ ตัวแปรตามที่ต้องการทำนาย

X คือ ตัวแปรอิสระที่ใช้ในการทำนาย

β_0 คือ ค่าจุดตัดแกน Y (Intercept) ค่า Y เมื่อ X = 0

β_1 คือ ค่าสัมประสิทธิ์ความชัน (Slope) ที่แสดงอัตราการเปลี่ยนแปลงของ Y ต่อการเปลี่ยนแปลงหนึ่งหน่วยของ X

ϵ คือ ค่าความคลาดเคลื่อน (Error term หรือ Residual) ซึ่งแสดงถึงค่าความแตกต่างค่า Y ที่เป็นค่าจริง (y_i) กับ ค่า Y ที่เกิดจากทำนาย ($\hat{y}_i$)



ภาพที่ 2.5 หลักการทำงานของการถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression)
(ที่มา https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/)

2.3.2 การถดถอยเชิงเส้นแบบพหุคูณ (Multiple Linear Regression)

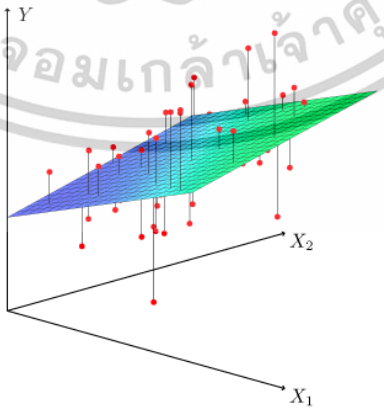
เมื่อมีตัวแปรอิสระมากกว่าหนึ่งตัว แบบจำลองจะขยายเป็นการถดถอยเชิงเส้นแบบพหุคูณ (Multiple Linear Regression) ซึ่งมีรูปของสมการ

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

โดยที่:

$X_1, X_2, \ldots\ldots, X_n$ คือ ตัวแปรอิสระ n ตัว

$\beta_1, \beta_2, \ldots\ldots, \beta_n$ คือ ค่าสัมประสิทธิ์ที่สอดคล้องกันของแต่ละตัวแปรอิสระ



ภาพที่ 2.6 หลักการทำงานของการถดถอยเชิงเส้นพหุคูณ (Multiple Linear Regression)
(ที่มา: https://statsandr.com/blog/multiple-linear-regression-made-simple/)

การประมาณค่าพารามิเตอร์ในแบบจำลองการถดถอยเชิงเส้นใช้วิธีกำลังสองน้อยที่สุด (Ordinary Least Squares: OLS) ซึ่งมีหลักการคือ การค่าสัมประสิทธิ์ที่ทำให้ผลรวมของกำลังสองของค่าความคลาดเคลื่อนมีค่าน้อยที่สุด

วิธีการที่นิยมใช้ในการประมาณค่าสัมประสิทธิ์ $\beta_1, \beta_2, \ldots\ldots, \beta_n$ ในแบบจำลองการถดถอยเชิงเส้น คือวิธีการกำลังสองน้อยที่สุด (Ordinary Least Squares: OLS) วัตถุประสงค์ของ OLS เพื่อลดความแตกต่างของผลรวมความแตกต่างระหว่าง ค่า Y ที่เป็นค่าจริง กับค่า Y ที่ทำนายได้ หรือเรียกว่า ค่าความคลาดเคลื่อน โดยค่าผลรวมของค่าความความเคลื่อน (Sum of Square Error: SSE) ที่ใช้หาค่าที่เหมาะสมที่สุดคือ

$$SSE = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

โดยที่:

m คือ จำนวนข้อมูลทั้งหมด

y_i คือ ค่าจริงของตัวแปรตามสำหรับข้อมูลตัวที่ i

$\hat{y}_i$ คือ ค่าทำนายของตัวแปรตามสำหรับข้อมูลตัวที่ i

และค่าประมาณของสัมประสิทธิ์ β ที่ได้จากวิธีกำลังสองน้อยที่สุดคือ

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

โดยที่:

X คือ เมทริกซ์ออกแบบที่ประกอบด้วยตัวแปรอิสระd

Y คือ เวกเตอร์ของตัวแปรตาม

$\hat{\beta}$ คือ เวกเตอร์ของสัมประสิทธิ์

X^T คือ เมทริกซ์สลับเปลี่ยน (Transpose) ของ X

$(X^T X)^{-1}$ คือ เมทริกซ์ผกผัน (Inverse)ของ $(X^T X)$

2.3.3 สมมติฐานของการถดถอยเชิงเส้น (Linear Assumption)

เพื่อให้แบบจำลองการถดถอยเชิงเส้นให้ผลลัพธ์ที่น่าเชื่อถือ และการอนุมานทางสถิติมีความถูกต้องจำเป็นต้องตรวจสอบสมมติฐาน ดังนี้

1) ความเป็นเชิงเส้น (Linearity): ความสัมพันธ์ระหว่างตัวแปรอิสระ และตัวแปรตามต้องเป็นเชิงเส้น

2) ความเป็นอิสระ (Independence): ค่าความคลาดเคลื่อนต้องเป็นอิสระจากกัน ไม่มีความสัมพันธ์ระหว่างความคลาดเคลื่อน

3) ความสม่ำเสมอของความแปรปรวน (Homoscedasticity*):* ความแปรปรวนของค่าความเคลื่อนต้องมีค่าคงที่

4) การแจกแจงแบบปกติ (Normality): ค่าความคลาดเคลื่อนต้องมีการแจกแจงแบบปกติ

5) ไม่มีปัญหาพหุสัมพันธ์ (Multicollinearity): ตัวแปรอิสระต้องไม่มีสัมพันธ์ที่สูงระหว่างกัน

2.4 การถดถอยแบบปัวซอง (Poisson Regression)

แบบจำลองสถิติที่ใช้ในการวิเคราะห์ข้อมูลที่มีลักษณะเป็นจำนวนนับ (Count data) ซึ่งแตกต่างจากข้อมูลต่อเนื่องที่ใช้ในการถดถอยเชิงเส้นทั่วไป ข้อมูลจำนวนนับ คือข้อมูลที่บอกจำนวนครั้งของการเกิดเหตุกาณ์ในช่วงเวลา หรือพื้นที่ที่กำหนด เช่น การเกิดอุบัติเหตุต่อวัน จำนวนลูกค้าที่เข้าร้านต่อชั่วโมง ข้อมูลจำนวนนับมีคุณลักษณะที่มีการแจกแจงที่ไม่ปกติ ส่วนใหญ่จะเป็นการแจกแจงที่เบ้ขวา (Right-skewed), ค่าต้องไม่เป็นจำนวนลบ และความแปรปรวนไม่คงที่ มักจะเปลี่ยนแปลงตามค่าเฉลี่ย

การถดถอยแบบปัวซองใช้การแจกแจกปัวซอง (Poisson Distribution) เป็นพื้นฐาน ซึ่งเป็นการแจกแจงความน่าจะเป็นแบบไม่ต่อเนื่องที่เหมาะสมสำหรับการอธิบายจำนวนครั้งของเหตุกาณ์ที่เกิดขึ้นในช่วงเวลา หรือพื้นที่ที่กำหนด ฟังก์ชันความน่าจะเป็นของแจกแจงปัวซองมีสมการ คือ

$$P(Y = k) = \frac{e^{-\lambda}\lambda^{k}}{k!}$$

โดยที่:

Y คือ ตัวแปรสุ่มที่แทนจำนวนครั้งของการเกิดเหตุการณ์

k คือ จำนวนครั้งของเหตุการณ์ที่เราสนใจ

λ คือ ค่าเฉลี่ย และค่าความแปรปรวนของเหตุการณ์

e คือ ค่าคงที่ของออยเลอร์ (2.718)

แบบจำลองการถดถอยแบบปัวซอง จะศึกษาความสัมพันธ์ระหว่างตัวแปรอิสระ และค่าเฉลี่ยของจำนวนนับ โดยใช้ฟังก์ชันเชื่อมโยง (Link function) เชื่อมโยงระหว่างค่าเฉลี่ยของตัวแปรตาม กับการรวมเชิงเส้นของตัวแปรอิสระ ฟังก์ชันที่เชื่อมโยงที่ใช้คือ ฟังก์ชันลอการิทึม (Log link function) อยู่ในรูปสมการ

$$\ln(\lambda_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_n X_{in}$$

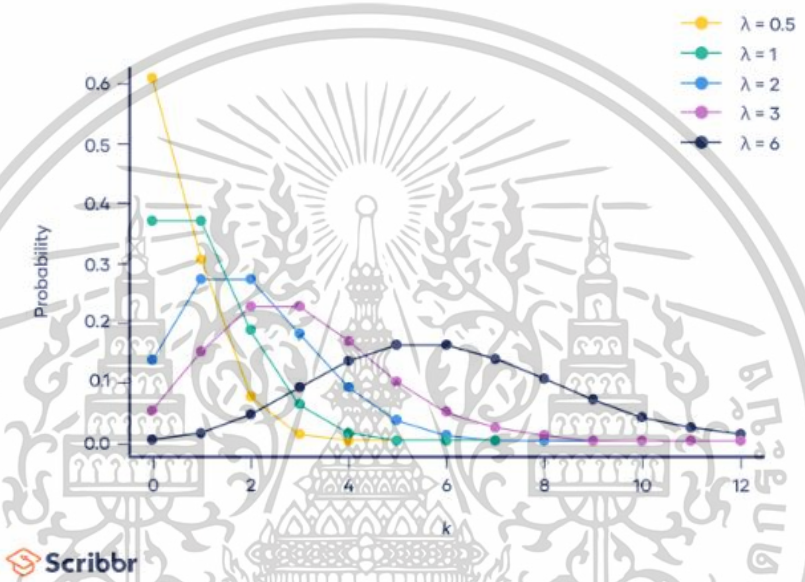
$$\lambda_i = e^{\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_n X_{in}}$$

โดยที่:

λ_i คือ ค่าเฉลี่ยที่คาดหวังของจำนวนนับสำหรับข้อมูลที่ i

$\beta_0, \beta_1, \beta_n$ คือ สัมประสิทธิ์การถดถอย

X_0, X_{i1}, Xi_n คือ ค่าตัวแปรอิสระสำหรับข้อมูลที่ i



ภาพที่ 2.7 ความน่าจะเป็นของปัวซองเมื่อเปลี่ยนไปตามค่าเฉลี่ยการเกิดเหตุการณ์

(ที่มา: https://www.scribbr.com/statistics/poisson-distribution/)

ข้อจำกัด และปัญหาของการถดถอยแบบปัวซอง แม้ว่าจะเหมาะสำหรับข้อมูลจำนวนนับ แต่ในทางปฏิบัติอาจพบปัญหาที่ทำให้แบบจำลองเกิดความไม่เหมาะ ได้แก่

1) ปัญหาความแปรปรวนเกิน (Overdispersion): เกิดขึ้นเมื่อความแปรปรวนของข้อมูลมีค่าสูงกว่าค่าเฉลี่ย ซึ่งผิดกับสมมติฐานของปัวซองที่ค่าเฉลี่ยน และความแปรปรวนต้องเท่ากัน

2) ปัญหาข้อมูลศูนย์เกิน (Zero-inflation): เกิดขึ้นเมื่อข้อมูลมีค่าศูนย์มากกว่าเกินกว่าค่าเฉลี่ยของปัวซอง

เพื่อแก้ไขปัญหาดังกล่าวจึงได้พัฒนาแบบจำลองที่ซับซ้อนขึ้น ดังนี้

2.4.1 การถดถอยปัวซองแบบทวินามเชิงลบ (Negative Binomial Regression)

แบบจำลองที่ใช้สำหรับข้อมูลจำนวนนับแบบเดียวกับการถดถอยแบบปัวซอง โดยมีการเพิ่มพารามิเตอร์การกระจาย (Dispersion Parameter) เข้าไปในแบบจำลอง ทำให้ความสัมพันธ์ระหว่างค่าเฉลี่ย และความแปรปรวน มีความยืดหยุ่นมากขึ้น โดยความแปรปรวนเฉลี่ยสมการ คือ

$$Var(Y) = \mu + \alpha\mu^2$$

โดยที่:

α คือ พารามิเตอร์การกระจาย

μ คือ ค่าเฉลี่ยของตัวแปรตาม

แบบจำลองการถดถอยปัวซองแบบทวินามเชิงลบจึงความสามารถในการจัดการกับปัญหาความแปรปรวนเกิน ได้ดีกว่าแบบจำลองปัวซอง หากค่าพารามิเตอร์การกระจายเข้าใกล้ศูนย์ แบบจำลองทั้งสองแบบทำงานเหมือนกัน

2.4.2 การถดถอยปัวซองแบบศูนย์พอง (Zero-Inflated Poisson)

แบบจำลองที่ได้รับการออกแบบมาโดยเฉพาะเพื่อวิเคราะห์ข้อมูลนับที่มี “จำนวนค่าศูนย์เกิน” ซึ่งเกินกว่าที่แบบจำลองปัวแซงปกติ โดยแบบจำลองจะสร้างกระบวนการคู่ขนานกัน

1) กระบวนการไบนารี่ (Binary Process): กระบวนการนี้จะจำลองความน่าจะเป็น ($\boldsymbol{\pi}$) ที่เหตุการณ์ใดๆ จะเป็นค่าศูนย์เสมอ (Structural zeros) โดยทั่วไปแบบจำลองจะใช้แบบจำลองการถดถอยโลจิสติก (Logistic Regression) เพื่อหาความน่าจะเป็นระหว่างกระบวนการ ซึ่งนั้นหมายความว่าถ้าเหตุการณ์อยู่ในกลุ่มนี้คืออยู่ในกลุ่มที่ไม่มีเหตุการณ์นั้นเกิดขึ้นเลย

2) กระบวนการนับ (Count Process) กระบวนการนี้หลังจากผ่านกระบวนการไบนารี่มาแล้ว ผลลัพธ์ไม่เป็นศูนย์เสมอ จะใช้การแจกแจงแบบปัวซองแบบปกติในการความน่าจะเป็น ฟังก์ชันความน่าจะเป็นของการถดถอยปัวซองแบบศูนย์พอง จะอยู่ในรูป

$$P(Y = 0) = \pi + (1 - \pi)e^{-\mu}$$

$$P(Y = k) = (1 - \pi)\frac{e^{-\lambda}\lambda^k}{k!} \text{ เมื่อ } k = 1,2,3, \dots, n$$

แบบจำลองการถดถอยปัวซองแบบศูนย์พอง จะแบ่งกระบวนการออกเป็น 2 ส่วน โดยที่ส่วนแรกใช้ตรวจสอบความน่าจะเป็นว่าเหตุการณ์นั้นอยู่ในกลุ่มของไม่มีเหตุการณ์นั้นเกิดขึ้นเลย และหลังจากที่ทำส่วนแรกเสร็จกลุ่มที่มีโอกาสเกิดเหตุการณ์จะใช้แบบจำลอง

แบบปัวซองแบบปกติ

2.4.3 สมมติฐานการถดถอยปัวซอง

1) ตัวแปรตามมีการแจกแจงแบบปัวซอง: ตัวแปรตามต้องมีค่าเป็นจำนวนนับ และจำนวนเต็มที่ไม่ติดลบ

2) ความเป็นอิสระของเหตุการณ์ (Independence of Observations): เหตุการณ์ในแต่ละเหตุกาณ์แต่ละครั้งต้องเป็นอิสระต่อกัน ไม่มีความสัมพันธ์ หรือการพึ่งพาอาศัยกันระหว่างเหตุการณ์

3) ความเท่ากันของค่าเฉลี่ย และความแปรปวน (Equisdispersion): การแจกแจงแบบปัวซองต้องมีค่าเฉลี่ย และความแปรปรวนเท่ากัน

4) ความสัมพันธ์เชิงเส้นระหว่าง ลอการิทึมของค่าเฉลี่ยกับตัวแปรอิสระ

5) ไม่มีปัญหาพหุสัมพันธ์ (Multicollinearity)

2.5 ต้นไม้ตัดสินใจ (Decision Tree)

แบบจำลองการเรียนรู้ของเครื่องจักรแบบมีผู้ฝึกสอนที่มีโครงสร้างแบบลำดับชั้น (Hierarchical) คล้ายกับต้นไม้ (Tree-based) โดยสามารถใช้งานได้ทั้งงานจำแนกประเภท และการถดถอย องค์ประกอบหลักของต้นไม้ตัดสินใจ ประกอบด้วย โหนดราก (Root Node) ใช้เป็นโหนดเริ่มต้นที่มีข้อมูลทั้งหมด, โหนดภายใน (Internal Node) ใช้เป็นโหนดที่ทำการตัดสินใจแบ่งข้อมูล, กิ่ง (Branches) ใช้เป็นเส้นเชื่อมที่แสดงผลของการตัดสินใจ, และโหนดใบ (Leaf Nodes) ใช้เป็นโหนดสุดท้ายที่ให้ผลการทำนาย

กระบวนการสร้างต้นไม้ตัดสินใจเป็นไปในลักษณะเวียนซ้ำ (Recursive) โดยเริ่มต้นจากโหนดราก ในแต่ละโหนด จะทำการเลือกคุณลักษณะที่เหมาะสมที่สุด และกำหนดเงื่อนไขการแบ่งข้อมูล (Splitting criterion) เพื่อแบ่งข้อมูลออกเป็นกลุ่มย่อย เป้าหมายเพื่อทำให้ข้อมูลในโหนดลูก (Child Nodes) มีความบริสุทธิ์ (Purity) หรือมีความเป็นเนื้อเดียวกัน (Homogeneous) สูงที่สุดตามค่าของตัวแปรตาม กระบวนการนี้จะทำซ้ำไปเรื่อยๆ ในแต่ละกิ่งจนกว่าเงื่อนไขการหยุด (Stopping criteria) ที่กำหนดไว้ เช่น โหนดสามารถแยกข้อมูลออกมาได้สมบูรณ์หรือจำนวนข้อมูลในโหนดน้อยกว่าเกณฑ์ที่กำหนดสำหรับการจำแนก และความลึกของต้นไม้ถึงขีดจำกัด เกณฑ์ที่นิยมใช้ได้แก่

1) Entropy และ Information Gain เป็นการวัดความไม่บริสุทธิ์ หรือความไม่แน่นอนของของข้อมูลในโหนด เป็นการเลือกคุณลักษณะที่ใช้เป็นจุดจำแนก โดย Entropy สำหรับโหนด t หาได้จาก

$$Entropy(t) = -\sum_{i=1}^{c} p(i|t) log_2 p(i|t)$$

โดยที่:

C	คือ จำนวนคลาส หรือจำนวนประเภทในข้อมูล
$p(i\|t)$	คือ สัดส่วนของข้อมูลในโหนด

และ Information Gain (เมื่อแบ่งแยกโหนด t ด้วยคุณลักษณะ A)

$$Gain(t, A) = H(t) - \sum_{v \in Values(A)} \frac{|t_v|}{|t|} H(t_v)$$

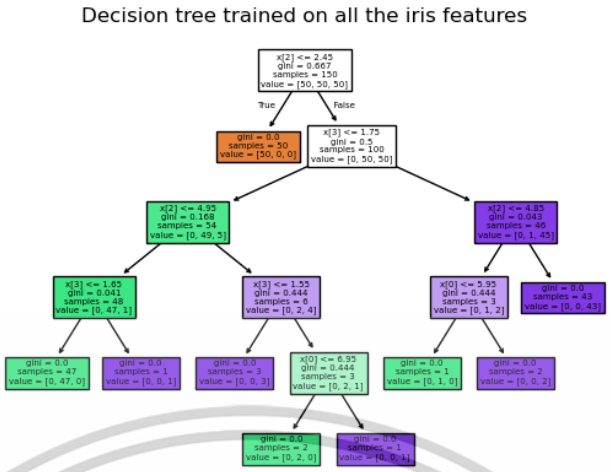
โดยที่:

Values(A)	คือ เซตของค่าที่ไปได้ของแอตทริบิวต์ A
t_v	คือ เซตของข้อมูลในโหนด t ที่มีแอตทริบิวต์ A เป็น v
$\|t\|$	คือ จำนวนข้อมูลในโหนดของ t
$\|t_v\|$	คือ จำนวนข้อมูลในโหนดของ t_v

2) Gini Index เป็นการวัดค่าความน่าจะเป็นของการจำแนกข้อมูลผิดคลาสแบบสุ่ม เพื่อเลือกคุณลักษณะที่ทำให้ Gini index น้อยที่สุด

$$Gini(t) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

แม้ว่า Decision Tree จะมีข้อได้เปรียบในด้านการตีความ (Interpretability) โดยเฉพาะสำหรับต้นไม้ขนาดเล็ก แต่ก็มีแนวโน้มที่จะเกิดปัญหา Overfitting ได้ง่ายหากต้นไม้มีความลึกมากเกินไป และมีความไม่เสถียร (Instability) คือการเปลี่ยนแปลงเล็กน้อยในข้อมูลฝึกฝนอาจนำไปสู่โครงสร้างต้นไม้ที่แตกต่างกันอย่างมาก



ภาพที่ 2.8 โครงสร้างของต้นไม้ตัดสินใจ (Decision Tree)
(ที่มา: https://scikit-learn.org/stable/auto_examples/tree/plot_iris_dtc.html)

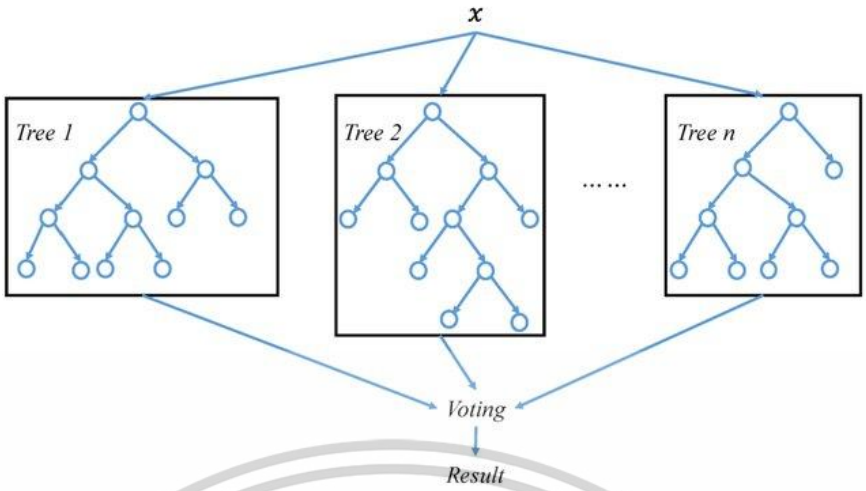
2.6 ป่าแบบสุ่ม (Random Forest)

แบบจำลองการเรียนรู้ของเครื่องจักรแบบรวมหมู่ (Ensemble Learning) ที่ใช้ต้นไม้ตัดสินใจหลายๆ ต้นมาทำงานร่วมกัน สามารถใช้ได้ทั้งการจำแนกประเภท และการถดถอย โดยหลักการพื้นฐานการสร้างต้นไม้ตัดสินใจหลายๆต้นมาจำนวนมาก (Forest) แต่ละต้นไม้จะถูกฝึกด้วยข้อมูลส่วนย่อยต่างๆ จากคุณลักษณะที่เลือกมา โดยผลลัพธ์สุดท้ายของการจำแนกประเภทจะรวบรวมผลลัพธ์จากหลายๆต้นในป่า มาตัดสินในการลงคะแนนเสียงส่วนมาก (Major Voting) หรือค่าเฉลี่ย (Average) ขึ้นอยู่กับลักษณะของงาน กระบวนการสร้างป่าแบบสุ่มประกอบด้วยกระบวนการ

1) การสุ่มตัวอย่างข้อมูล (Bootstrapping): แต่ละต้นไม้จะฝึกด้วยข้อมูลที่ได้จากการสุ่มตัวอย่างข้อมูล และใส่คืน

2) การสุ่มเลือกคุณลักษณะ (Feature Randomness): ในขณะที่แต่ละต้นไม้สร้างจุดแบ่งแต่ละโหนด จะพิจารณาเลือกคุณลักษณะที่เหมาะสมที่สุดในการแยกข้อมูล แทนที่จะพิจารณาข้อมูลทั้งหมด

แบบจำลองแบบป่าแบบสุ่มมีข้อได้เปรียบสำคัญคือช่วย ลดปัญหา Overfitting ได้ดีกว่าแบบจำลองแบบต้นไม้ตัดสินใจเพียงต้นเดียว มีความแม่นยำสูงกว่า และมีความเสถียรมากกว่า นอกจากนี้ยังสามารถให้ ความสำคัญของคุณลักษณะ (Feature Importance) ซึ่งบอกได้ว่าคุณลักษณะใดมีส่วนสำคัญต่อการทำนาย แต่แบบจำลองแบบป่าแบบสุ่มมีความซับซ้อนในการตีความมากกว่าแบบจำลองแบบต้นไม้ตัดสินใจ และต้องใช้ทรัพยากรในการประมวลผลสูงกว่า



ภาพที่ 2.9 โครงสร้างของแบบจำลองสุ่มป่า (Random Forest)
(ที่มา:
https://www.researchgate.net/publication/335483097_A_hybrid_ensemble_method_for_pulsar_candidate_classification/figures?lo=1)

2.7 เอ็กซ์ตรีมเกรเดียนท์บูสติ้ง (Extreme Gradient Boosting)

แบบจำลองที่มีประสิทธิภาพสูง และได้รับความนิยมอย่างแพร่หลาย เป็นเทคนิคการเรียนรู้ของเครื่องจักรแบบรวมหมู่ อีกรูปแบบหนึ่งที่ทำงานในลักษณะลำดับ (Sequential) โดยจะสร้างแบบจำลอง (มักเป็นแบบจำลองต้นไม้ตัดสินใจขนาดเล็ก (Weak Learners)) ทีละต้น และแต่ละต้นไม้ที่สร้างขึ้นมาใหม่จะพยายามแก้ไขข้อผิดพลาด (Errors) ที่เกิดขึ้นจากแบบจำลองรวมของต้นไม้ก่อนหน้า

แนวคิดหลักของแบบจำลอง คือการปรับปรุงแบบจำลองซ้ำๆ โดยการหาแบบจำลองที่สามารถเข้ากับ ความชัน (Gradient) ของฟังก์ชันการสูญเสีย (Loss Function) ที่ใช้ในการวัดข้อผิดพลาดของแบบจำลองรวมชุดปัจจุบัน สำหรับงานถดถอยความชันนี้มักจะสัมพันธ์กับค่าความคลาดเคลื่อน (Residuals) ส่วนงานจำแนกประเภท จะเกี่ยวข้องกับความชันของการสูญเสีย (Loss Function) ที่เหมาะสม เช่น Cross-Entropy

แบบจำลองได้รับการพัฒนาเพิ่มเติมจากแบบจำลองโดยทั่วไป โดยมีการเพิ่มประสิทธิภาพในหลายด้าน ได้แก่

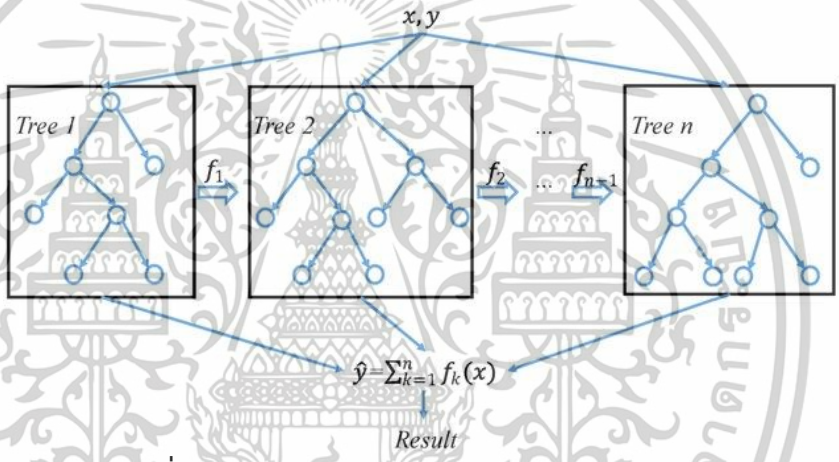
1) การใช้ Regularization (L1 และ L2): เพื่อช่วยในการคัดเลือกคุณลักษระ และลดความซับซ้อนของแบบจำลอง

2) การจัดการข้อมูลที่ขาดหายไป: แบบจำลองจะมีกลไกภายในการเรียนรู้วิธีการจัดการกับข้อมูลที่ขาดหายไป ได้โดยอัตโนมัติโดยไม่ต้องทำการเติมค่าสูญหาย (Imputation) ล่วงหน้า

แบบจำลองจะพิจารณาทิศทางการแบ่งข้อมูลที่ดีที่สุดเมื่อมีค่าข้อมูลที่ขาดหายไป ในระหว่างการสร้างต้นไม้

3) การทำการตัดแต่งกิ่ง (Tree Pruning) ย้อนกลับ: แบบจำลองจะสร้างต้นไม้ให้ลึกถึงค่าที่กำหนดไว้ล่วงหน้าโดยพารามิเตอร์ที่ชื่อว่า “max_depth” ก่อน จากนั้นจะทำการ Pruning ย้อนกลับ จะตัดกิ่งที่ไม่ก่อให้เกิดการปรับปรุงความผิดพลาดอย่างมีนัยสำคัญ วิธีนี้ช่วยหลีกเลี่ยงปัญหาการ Pruning ก่อนเวลาอันควร (Pre-Pruning) ซึ่งอาจทำให้พลาดข้อมูลสำคัญ

4) รองรับการประมวลผลแบบขนาน (Parallel Processing): แบบจำลองถูกออกแบบมาให้สามารถใช้ประโยชน์จาก CPU หลาย Core ได้อย่างมีประสิทธิภาพในการฝึกฝนต้นไม้ ทำให้กระบวนการฝึกทำได้รวดเร็วกว่า



ภาพที่ 2.10 โครงสร้างของแบบจำลอง XGBoost

(ที่มา https://www.researchgate.net/publication/335483097_A_hybrid_ensemble_method_for_pulsar_candidate_classification/figures?lo=1)

2.8 ไลต์เกรเดียนท์บูสติ้ง (Light Gradient Boosting)

แบบจำลองที่ถูกพัฒนาโดยไมโครซอร์ฟ (Microsoft) โดยหลักการพื้นฐานของแบบจำลองเกรเดียนท์บูสติ้ง เน้นการเพิ่มประสิทธิภาพด้านความเร็ว (Speed) และ ความสามารถในการรองรับข้อมูลขนาดใหญ่ (Scalability) โดยเฉพาะสำหรับข้อมูลที่มีมิติสูง แบบจำลองการปรับปรุงด้านประสิทธิภาพผ่านเทคนิคการปรับปรุงหลักๆ ดังนี้:

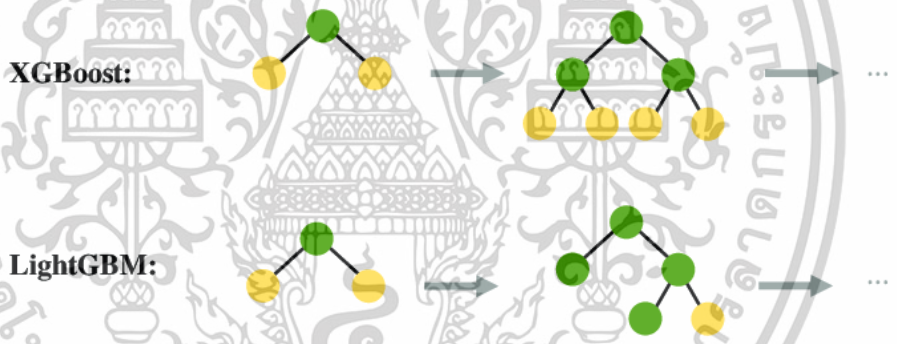
1) Histogram-based Splitting: แทนที่จะหาจุดแบ่งที่ดีที่สุดจากค่าจริงทั้งหมดของแต่ละคุณลักษณะแบบจำลองจะแปลงค่าของคุณลักษณะต่อเนื่องให้อยู่ในรูปแบบของฮิสโตแกรม (Histograms) แบ่งค่าที่เป็นช่วง ทำให้กระบวนการค้นหาการแบ่งชุดข้อมูลรวดเร็วขึ้นอย่างมาก

2) Leaf-wise (หรือ Best-first) Tree Growth: ต่างจากแบบจำลองทั่วไปส่วนใหญ่ที่สร้างต้นไม้แบบ Level-wise (ขยายโหนดทั้งหมดในระดับเดียวกัน) แบบจำลองจะเลือกขยายโหนดปลายทาง (Leaf) ที่ให้ ผลตอบแทน (Gain) หรือลดค่าผิดพลาดได้มากที่สุด ซึ่งทำให้ได้โครงสร้างต้นไม้ที่ไม่สมดุล แต่โดยทั่วไปจะนำไปสู่การผลลัพธ์ที่รวดเร็วขึ้น และแม่นยำขึ้น

3) Gradient-based One-Side Sampling (GOSS): เทคนิคการสุ่มเลือกข้อมูลฝึกฝน โดยให้ความสำคัญกับข้อมูลที่มีค่า Gradient สูง (ค่าที่ยังมีความผิดพลาดสูง และทำนายได้ไม่ดี) และสุ่มเลือกข้อมูลที่มีค่า Gradient ต่ำมาเพียงบางส่วน เพื่อลดจำนวนข้อมูลที่ใช้ในการฝึกฝนแต่ละต้นไม้

4) Exclusive Feature Bundling (EFB): เทคนิคที่ใช้รวมกลุ่มคุณลักษณะบางอย่างที่มีความสัมพันธ์กันน้อย หรือไม่เกิดขึ้นพร้อมกันให้อยู่ชุดเดียวกัน หมวดหมู่เดียวกัน (Bundle) เพื่อลดจำนวนคุณลักษณะ และเร่งความเร็วในการฝึกฝน

แบบจำลองไลต์เกรเดียนท์บูสติ้ง จะฝึกฝนได้เร็วกว่า และใช้หน่วยความจำน้อยกว่าแบบจำลองเอ็กซ์ตรีมเกรเดียนท์บูสติ้งโดยเฉพาะเมื่อทำงานกับชุดข้อมูลขนาดใหญ่ ในขณะที่ยังคงให้ประสิทธิภาพในการทำนายที่ใกล้เคียงกัน หรือดีกว่า



ภาพที่ 2.11 การเปรียบเทียบโครงสร้างระหว่างแบบจำลอง XGBoostและ LightGBM (ที่มา: https://www.linkedin.com/pulse/xgboost-vs-lightgbm-ashik-kumar)

2.9 การตรวจสอบค่าสุดโต่ง หรือค่าที่ผิดปกติ (Outlier Detection)

จุดข้อมูลที่มีลักษณะแตกต่างอย่างมีนัยสำคัญจากแนวโน้มส่วนใหญ่ของข้อมูลในชุดข้อมูล ค่าเหล่านี้อาจเกิดขึ้นจากหลายสาเหตุ เช่น ความผิดพลาดระหว่างการเก็บข้อมูล (Measurement Error), การบันทึกข้อมูลที่ไม่ถูกต้อง (Data Entry Error), หรืออาจเป็นค่าที่เกิดขึ้นจริงแต่เป็นเหตุการณ์ที่เกิดขึ้นได้ยาก (Genuine but Anomalous Value) การตรวจจับ และจัดการกับค่าผิดปกติถือเป็นขั้นตอนสำคัญในกระบวนการเตรียมข้อมูล (Data Preprocessing) เนื่องจากค่าผิดปกติสามารถส่งผลกระทบอย่างมากต่อการวิเคราะห์ทางสถิติและการฝึกฝนแบบจำลองการเรียนรู้ของเครื่องจักร ทำให้ค่าสถิติพื้นฐาน (เช่น ค่าเฉลี่ย, ความแปรปรวน) เกิดความเอนเอียง (Bias) และ

ทำให้ประสิทธิภาพของแบบจำลองลดลง

2.9.1 วิธีการตรวจสอบค่าผิดปกติโดยใช้ IQR (Interquartile Range)

วิธี IQR เป็นเทคนิคที่ไม่ขึ้นอยู่กับการแจกแจงของข้อมูล (Non-parametric) และใช้กราฟกล่อง (Box Plot) เป็นเครื่องมือพื้นฐานในการระบุค่าผิดปกติ IQR คือ ค่าความแตกต่างระหว่างควอไทล์ที่สาม (Q3) และควอไทล์ที่หนึ่ง (Q1) โดย Q1 คือค่าที่แบ่งข้อมูล 25% แรก และ Q3 คือค่าที่แบ่งข้อมูล 75% แรกออกจากข้อมูลที่เหลือ

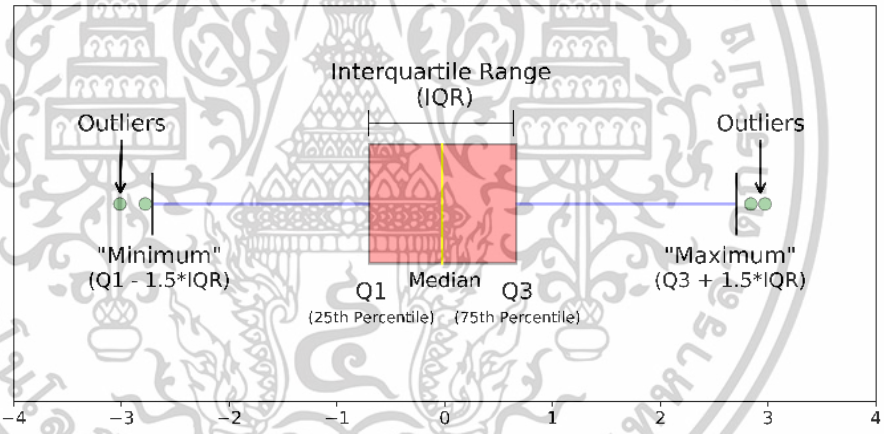
$$\text{IQR} = \text{Q3} - \text{Q1}$$

จากค่า IQR สามารถกำหนดสำหรับการระบุค่าความผิดปกติได้ โดยพิจารณาค่าที่อยู่นอกขอบล่าง และขอบบน

ขอบล่าง $\text{(Lower Bound)} = \text{Q1} - 1.5 \times \text{IQR}$

ขอบบน $\text{(Upper Bound)} = \text{Q3} + 1.5 \times \text{IQR}$

พิจารณาถ้าค่าไม่ได้ในช่วงขอบล่าง (Lower Bound) และ ขอบบน (Upper Bound) ถือว่าเป็นค่าผิดปกติ



ภาพที่ 2.12 Boxplot สำหรับการตรวจสอบค่าผิดปกติในข้อมูล

(ที่มา: https://www.kdnuggets.com/2019/11/understanding-boxplots.html)

2.10 การคัดเลือกคุณลักษณะที่สำคัญ (Feature Selection)

กระบวนการในการระบุ และเลือกชุดย่อยของคุณลักษณะ (Subset of Features) ที่มีความเกี่ยวข้อง (Relevant) มีความสัมพันธ์ (Correlated) หรือมีความสำคัญ (Important) กับ ตัวแปรตามที่ต้องการทำนาย จากชุดคุณลักษณะเริ่มต้นทั้งหมด วัตถุประสงค์หลักของการคัดเลือกคุณลักษณะในงาน คือการปรับปรุงประสิทธิภาพของแบบจำลอง (Improved Model Performance), ลดความซับซ้อนของแบบจำลอง (Reduce Model Complexity), ลดเวลาในการ

ฝึกฝน (Reduce Training Time), ป้องกันปัญหา Overfitting และเพิ่มความสามารในการตีความแบบจำลอง โดยวิธีการเลือกคุณลักษณะแบบดังนี้

1) สหสัมพันธ์ (Correlation): ใช้สัมประสิทธิ์สหสัมพันธ์ของเพียร์สัน (Pearson Correlation Coefficient) เพื่อวัดความสัมพันธ์เชิงเส้นระหว่างสองตัวแปร ค่าสัมประสิทธิ์อยู่ในช่วง [-1, 1] โดยค่าใกล้ -1 หรือ 1 บ่งชี้ถึงความสัมพันธ์เชิงเส้นที่แข็งแกร่ง (ทิศทางลบ/บวก) และค่าใกล้ 0 บ่งชี้ถึงความสัมพันธ์เชิงเส้นที่อ่อนแอ ข้อจำกัดคือวัดได้เฉพาะความสัมพันธ์เชิงเส้น

2) เลือกจากข้อมูลร่วมกัน (Mutual Information: MI): มาตรวัดปริมาณของข้อมูลร่วมกันระหว่างสองตัวแปร สามารถจับความสัมพันธ์ได้ทั้งแบบเชิงเส้น และไม่เป็นเชิงเส้น ค่าข้อมูลร่วมที่สูงบ่งชี้ว่าการทราบค่าของตัวแปรหนึ่งช่วยลดความไม่แน่นอนเกี่ยวกับค่าของอีกตัวแปรได้มาก

3) การเลือกจากความสำคัญของคุณลักษณะ (Feature Importance): ในแบบจำลอง Tree-based จะมีการคำนวณ Feature Importance โดยพิจารณาจากปริมาณการลดลงของเกณฑ์ความไม่บริสุทธิ์ (เช่น Entropy, Gini Index) ได้จากการใช้คุณลักษณะนั้นในการแบ่งโหนด คุณลักษณะที่นำไปสู่การลดความไม่บริสุทธิ์/ความแปรปรวนได้มากที่สุดจะถูกพิจารณาว่ามีความสำคัญสูง

หลังจากที่ได้คุณลักษณะจากการเลือกต่างๆ ต้องตรวจสอบปัญหา Multicollinearity เพื่อลดปัญหาคุณลักษณะที่เลือกมามีความสัมพันธ์กันเองสูง

2.11 การจัดการกับข้อมูลที่ไม่สมดุล (Imbalanced Data)

ข้อมูลที่มีการกระจายที่ไม่สมดุลระหว่างคลาสต่างๆ โดยมีจำนวนข้อมูลในแต่ละคลาสแตกต่างกันอย่างมากในการทำการจำแนกประเภท ซึ่งเกิดขึ้นเมื่อข้อมูลคลาสนึงมีจำนวนมาก ส่วนข้อมูลอีกคลาสนึงไม่จำนวนน้อยมาก ต่างกันอย่างมาก ซึ่งจะทำให้แบบจำลองมีแนวโน้มที่จะเรียนรู้คลาสที่มีจำนวนมากมากกว่าคลาสที่มีจำนวนน้อย ซึ่งอาจจะทำให้เกิดผลลัพธ์ที่ลำเอียง (Bias) โดยจัดการกับข้อมูลที่ไม่สมดุลได้โดยวิธีการ

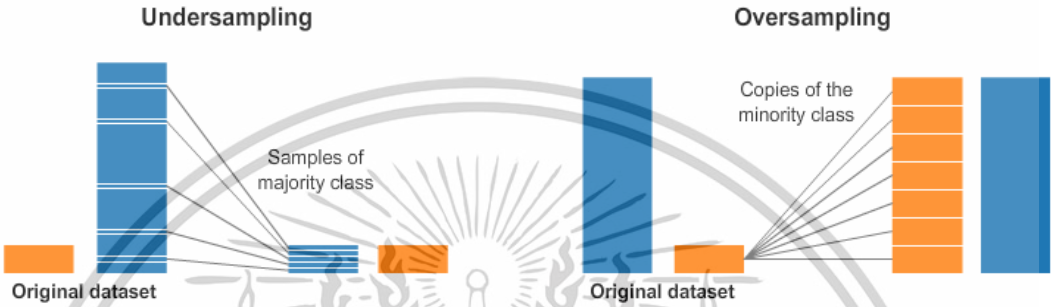
1) การลดจำนวนข้อมูล (Undersampling): เป็นการลดจำนวนข้อมูลของคลาสส่วนใหญ่ (Majority Class) ลง เพื่อให้มีจำนวนใกล้เคียงกับคลาสส่วนน้อย วิธีนี้เหมาะสมเมื่อมีข้อมูลจำนวนมากพอในคลาสส่วนใหญ่ ข้อดีคือช่วยลดขนาดของชุดข้อมูลและลดเวลาในการฝึกฝนแบบจำลอง ข้อเสียคืออาจสูญเสียข้อมูลที่มีประโยชน์ซึ่งอยู่ในตัวอย่างที่ถูกสุ่มออกไปจากคลาสส่วนใหญ่

2) การเพิ่มข้อมูล (Oversampling): เป็นการเพิ่มจำนวนข้อมูลของคลาสส่วนน้อย (Minority Class) เพื่อให้มีจำนวนใกล้เคียงกับคลาสส่วนใหญ่ วิธีนี้เหมาะสมเมื่อชุดข้อมูลมีขนาดเล็กหรือขนาดปานกลาง ข้อดีคือไม่สูญเสียข้อมูลจากคลาสส่วนใหญ่ และช่วยให้แบบจำลองได้เรียนรู้ลักษณะของคลาสส่วนน้อยมากขึ้น ลดปัญหา Bias ข้อเสียคืออาจเพิ่มโอกาสในการเกิด Overfitting โดยเฉพาะ

หากใช้วิธีการเพิ่มข้อมูลแบบง่ายๆ การสุ่มเพิ่มจำนวนข้อมูลสามารถทำได้หลายวิธี เช่น

2.1) การทำซ้ำข้อมูลเดิม (Simple Duplication): คัดลอกข้อมูลของคลาสส่วนน้อย

2.2) การสังเคราะห์ข้อมูลใหม่ (Synthetic Minority Over-sampling Technique: SMOTE): สร้างตัวอย่างข้อมูลใหม่ของคลาสส่วนน้อยขึ้นมาโดยการสุ่มเลือกตัวอย่างจากคลาสส่วนน้อยและตัวอย่างเพื่อนบ้านที่ใกล้เคียง จากนั้นสังเคราะห์ข้อมูลขึ้นมาระหว่างจุดข้อมูลเหล่านั้น



ภาพที่ 2.13 การจัดการข้อมูลไม่สมดุลด้วยเทคนิค Over/Undersampling (ที่มา: https://medium.com/analytics-vidhya/undersampling-and-oversampling-an-old-and-a-new-approach-4f984a0e8392)

2.12 การประเมินประสิทธิภาพของแบบจำลอง (Model Evaluation)

เป็นการประเมินแบบจำลองเพื่อใช้วัดค่า ค่าที่แบบจำลองทำนายได้ใกล้เคียงกับค่าจริง มากน้อยเพียงใด โดยเน้นวัดความผิดพลาดของค่าที่ทำนาย โดยถ้าเป็นแบบจำลองการถดถอยจะมีเมตริกที่ใช้วัดดังนี้

2.12.1 ค่าที่ใช้สำหรับแบบจำลองปัวซอง

1) ค่าความน่าจะเป็นแบบลอการึทึม (Log-Likelihood): วัดความน่าจะเป็นที่ข้อมูลจะเกิดขึ้นภายใต้แบบจำลองที่สร้างขึ้น โดยใช้สมการลอการึทึม โดยใช้เปรียบเทียบแบบจำลอง ว่าแบบจำลองอันไหนเหมาะสมกับชุดข้อมูลมากกว่า ถ้าแบบจำลองไหนมีค่าสูงกว่า แสดงว่าแบบจำลองนั้นเหมาะสมมากกว่า

$$\log L(\theta) = \sum_{i=1}^{n} \ln P(x_i|\theta)$$

โดยที่:

x_i คือ ข้อมูลตัวที่ i

θ คือ พารามิเตอร์ของแบบจำลอง

2) ค่าสารสนเทศของอาไคเกะ (Akaike Information Criterion: AIC): เป็นเกณฑ์การประเมินแบบจำลองที่คำนึงถึงทั้งความแม่นยำ และความซับซ้อนของแบบจำลอง โดยถ้าแบบจำลองไหนมีค่า AIC ที่ต่ำกว่าแสดงว่าแบบจำลองนั้นแบบจำลองดีกว่า

$$AIC = 2k - 2\ln(L)$$

โดยที่ k คือ จำนวนคุณลักษณะ

L คือ ค่า Loglikelihood

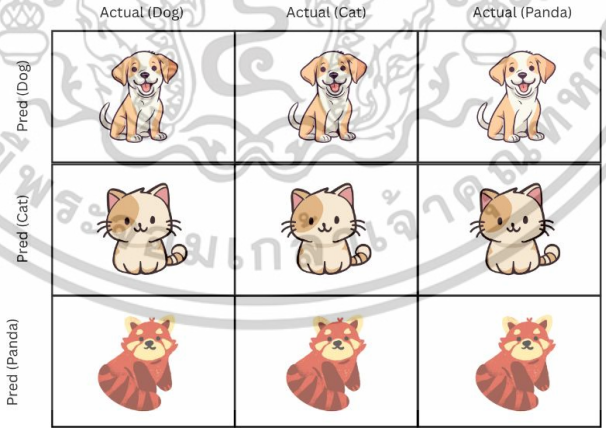
3) ค่าสารสนเทศของเบย์ (Bayesian Information Criterion: BIC): เป็นเกณฑ์การประเมินแบบจำลองที่คำนึงถึงทั้งความแม่นยำ และความซับซ้อนของแบบจำลอง เหมือนกับ AIC แต่ให้ค่าปรับโทษที่มากกว่า โดยถ้าแบบจำลองไหนมีค่า BIC ที่ต่ำกว่าแสดงว่าแบบจำลองนั้นแบบจำลองดีกว่า

$$BIC = k\,x\ln(n) - 2\ln(L)$$

โดยที่ n คือ จำนวนข้อมูล

2.12.2 ค่าที่ใช้สำหรับแบบจำลองการจำแนกประเภท

เมื่อแบบจำลองเป็นแบบจำแนกประเภทจะใช้เมตริกการ โดยการอธิบายด้วยเมทริกซ์คอนฟิวชั่น (Confusion Matrix) ที่ใช้สรุปผลการทำนายของแบบจำลองแบบจำแนกประเภท โดยการเปรียบเทียบระหว่างค่าที่แบบจำลองการทำนาย (Predicted) กับค่าจริง (Actual) จำนวนขนาดของเมทริกซ์จะขึ้นอยู่กับจำนวนของคลาส



ภาพที่ 2.14 เมทริกซ์ความสับสน (Confusion Matrix)

True Positive (TP): จำนวนข้อมูลที่คลาสจริงเป็น Dog และทำนายถูกต้องได้เป็น Dog

True Negative (TN): จำนวนข้อมูลที่คลาสจริงเป็น Cat/Panda และทำนายถูกต้องได้เป็น Cat/Panda

False Positive (FP): จำนวนข้อมูลที่คลาสจริงเป็น Cat/Panda และทำนายผิดพลาดได้เป็น Dog

False Negative (FN): จำนวนข้อมูลที่คลาสจริงเป็น Dog และทำนายผิดพลาดได้เป็น Cat/Panda

1) ความเที่ยงตรง (Precision): สัดส่วนของจำนวน True Positive ต่อจำนวนที่ทำนายเป็น Positive ทั้งหมด วัดความถูกต้องเมื่อแบบจำลองทำนายว่าเป็น Positive

$$Precision_i = \frac{TP_i}{TP_i + FP_i}$$

2) ความแม่นยำในการทำนายถูกจริง (Recall): สัดส่วนของจำนวน True Positive ต่อจำนวน Actual Positive ทั้งหมด วัดความสามารถของแบบจำลองในการค้นหา Positive จริงทั้งหมด

$$Recall_i = \frac{TP_i}{TP_i + FN_i}$$

3) ค่าเอฟวันสกอร์ (F1-Score): เป็นค่าเฉลี่ยแบบถ่วงน้ำหนักระหว่าง Precision และ Recall ให้ค่าสมดุลระหว่างสองเมตริกนี้ เหมาะสำหรับใช้กับข้อมูลที่ไม่สมดุล

$$F1_i = 2\,x\,\frac{Precision_i\,x\,Recall_i}{Precision_i + Recall_i}$$

4) ความแม่นยำ (Accuracy): สัดส่วนของจำนวนการทำนายที่ถูกต้องทั้งหมด ต่อจำนวนข้อมูลทั้งหมด โดยใช้บอกว่าแบบจำลองสามารถจำแนกประเภทข้อมูลถูกต้องกี่เปอร์เซ็นต์

$$Accuracy_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

และถ้าเป็น Accuracy รวมของแบบจำลองสมมติว่ามี 3 Classes (A,B,C)

$$Accuracy = \frac{TP_A + TP_B + TP_C}{Total}$$

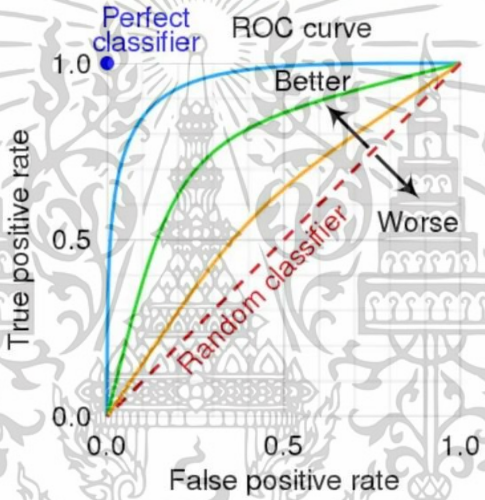
5) ค่าเฉลี่ยแบบ Weighted Average

$$Weighted\ Precision = \frac{(Precision_A x\ Support_A) + (Precision_B x\ Support_B) + (Precision_c x\ Support_c)}{Total}$$

$$Weighted\ Recall = \frac{(Recall_A x\ Support_A) + (Recall_B x\ Support_B) + (Recall_c x\ Support_c)}{Total}$$

$$Weighted\ F1 - Score = \frac{(F1_A x\ Support_A) + (F1_B x\ Support_B) + (F1_c x\ Support_c)}{Total}$$

6) กราฟ ROC (Receiver Operating Characteristic Curve): เป็นการวัดความสามารถในการจำแนกประเภทของแบบจำลอง โดยคำนวณจากค่า Sensitivity (True Positive Rate: TPR) เทียบกับ ค่า False Positive Rate (FPR) โดยจะสร้างกราฟขึ้นและคำนวณพื้นที่ใต้กราฟเรียกว่า AUC (Area Under the Curve) เพื่อใช้บ่งบอกว่าแบบจำลองมีความสามารถในการจำแนกมากแค่ไหน โดยที่ใกล้ 1 แปลว่าแบบจำลองมีความสามารถในการจำแนกได้อย่างดีเยี่ยม



ภาพที่ 2.15 กราฟ ROC Curve สำหรับวัดประสิทธิภาพการจำแนก
(ที่มา: https://spotintelligence.com/2024/06/17/roc-auc-curve-in-machine-learning/)

2.13 การปรับจูนไฮเปอร์พารามิเตอร์แบบจำลอง (Hyperparameter Tuning)

กระบวนการในการค้นหา ชุดค่าของไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด สำหรับแบบจำลอง เพื่อให้แบบจำลองที่ได้มีประสิทธิภาพสูงสุดในการทำนายบนชุดข้อมูลที่ไม่เคยเห็นมาก่อน กระบวนการนี้มีความสำคัญอย่างยิ่งเนื่องจากค่าของไฮเปอร์พารามิเตอร์มีผลกระทบอย่างมากต่อประสิทธิภาพของแบบจำลอง ความสามารถในการ Generalize และปัญหา Overfitting หรือ Underfitting วิธีการปรับจูนไฮเปอร์พารามิเตอร์ที่นิยมใช้ ได้แก่:

2.13.1 การค้นหาแบบกริด (Grid Search)

เป็นวิธีการที่กำหนด ช่วงของค่า (Value Range) หรือ รายการของค่าที่เป็นไปได้ (List of Possible Values) สำหรับไฮเปอร์พารามิเตอร์แต่ละตัวที่ต้องการปรับจูน จากนั้น

Grid Search จะทำการฝึกฝนและประเมินประสิทธิภาพของแบบจำลองสำหรับ ทุกชุดค่าผสม (Combination) ที่เป็นไปได้ของไฮเปอร์พารามิเตอร์ที่กำหนดไว้ใน Grid ชุดค่าผสมที่ให้ประสิทธิภาพสูงสุดบนชุดข้อมูลสำหรับประเมิน (เช่น ชุด Validation Set หรือโดยใช้ Cross-Validation) จะถูกเลือก วิธี Grid Search รับประกันว่าจะได้ชุดค่าที่ดีที่สุดจาก Grid ที่กำหนด แต่มีข้อเสียคือใช้เวลาในการคำนวณสูงมาก โดยเฉพาะเมื่อจำนวนไฮเปอร์พารามิเตอร์หรือจำนวนค่าใน Grid เพิ่มขึ้น มักใช้ร่วมกับ การตรวจสอบแบบ Cross-Validation (Cross-Validation) เพื่อให้การประเมินประสิทธิภาพของแต่ละชุดค่าไฮเปอร์พารามิเตอร์มีความน่าเชื่อถือและลดความเสี่ยงของการ Overfitting กับชุด Validation เพียงชุดเดียว

2.13.2 การค้นหาแบบสุ่ม (Random Search)

เป็นวิธีการที่กำหนดช่วงของค่า (Range) หรือการแจกแจง (Distribution) สำหรับไฮเปอร์พารามิเตอร์แต่ละตัว จากนั้นทำการสุ่มเลือก ชุดค่าผสมของไฮเปอร์พารามิเตอร์ มาจำนวนหนึ่ง (กำหนดจำนวนครั้งในการสุ่มไว้ล่วงหน้า) และทำการฝึกฝนและประเมินแบบจำลองสำหรับแต่ละชุดค่าผสมที่สุ่มได้ วิธี Random Search มักจะมีประสิทธิภาพในการค้นหาชุดค่าไฮเปอร์พารามิเตอร์ที่ดีได้เร็วกว่า Grid Search โดยเฉพาะในกรณีที่มีไฮเปอร์พารามิเตอร์จำนวนมาก หรือมีไฮเปอร์พารามิเตอร์บางตัวที่มีผลกระทบต่อประสิทธิภาพมากกว่าตัวอื่น เนื่องจาก Random Search มีโอกาส "สุ่มเจอ" ค่าที่ดีในไฮเปอร์พารามิเตอร์ที่สำคัญได้แม้ว่าจะสุ่มจำนวนครั้งไม่มากนัก

2.13.3 การเพิ่มคุณลักษณะด้วย Feature Engineering

การเพิ่มคุณลักษณะเข้าไปหลังจากกระบวนการของการเลือกคุณลักษณะแบบความสำคัญของคุณลักษณะ โดยมีจุดประสงค์เพื่อเพิ่มประสิทธิภาพในการทำนายของแบบจำลอง

2.14 วิจัยที่เกี่ยวข้อง (Related Paper/Related Research)

2.14.1 Predicting the Outcome of English Premier League Matches using Machine Learning

ผู้วิจัย: Omoregie และ Monday (2021)

หลักการทำงาน: ใช้ข้อมูลจากฤดูกาล 2021–2024 เช่น จำนวนประตู, ใบเหลือง, ใบแดง, การครองบอล และการยิงตรงกรอบ เพื่อฝึกสอนแบบจำลอง Gaussian Naive Bayes ในการทำนายผลการแข่งขัน

แบบจำลองที่ใช้: Gaussian Naive Bayes

ผลลัพธ์: ความแม่นยำประมาณ 62%

แหล่งที่มา:

https://www.researchgate.net/publication/349367241_Predicting_the_Outcome_of_English_Premier_League_Matches_using_Machine_Learning

2.14.2 Predicting Soccer Match Results in the English Premier League

ผู้วิจัย: Ulmer และ Fernandez (2014)

หลักการทำงาน: ใช้ข้อมูลประวัติการแข่งขันฟุตบอลพรีเมียร์ลีกอังกฤษเพื่อสร้างชุดคุณลักษณะ และนำไปฝึกสอนแบบจำลอง Machine Learning หลากหลายประเภท เช่น Linear Classifier, Naive Bayes, Hidden Markov Model, Support Vector Machine (SVM), และ Random Forest เพื่อทำนายผลการแข่งขันเป็นสามประเภท: ชนะ, เสมอ, หรือแพ้

แบบจำลองที่ใช้: Linear Classifier, Naive Bayes, Hidden Markov Model, SVM, Random Forest

ผลลัพธ์: Random Forest มีประสิทธิภาพดีที่สุด โดยสามารถบรรลุความแม่นยำสูงสุดที่ 63%

แหล่งที่มา:

https://cs229.stanford.edu/proj2014/Ben%20Ulmer%2C%20Matt%20Fernandez%2C%20Predicting%20Soccer%20Results%20in%20the%20English%20Premier%20League.pdf

2.14.3 Predicting Football Match Results Using a Poisson Regression Model

ผู้วิจัย: Loukas et al. (2023)

หลักการทำงาน: ใช้แบบจำลอง Double Poisson Regression เพื่อทำนายจำนวนประตูที่แต่ละทีมจะทำได้ โดยอิงจากข้อมูลการแข่งขันจริงของพรีเมียร์ลีกฤดูกาล 2022–2023

แบบจำลองที่ใช้: Double Poisson Regression

ผลลัพธ์: ความคลาดเคลื่อนของการทำนายประตู ±1 ประตู

แหล่งที่มา: https://www.mdpi.com/2076-3417/14/16/7230

2.14.4 Poisson Modeling and Predicting English Premier League Goal Scoring

ผู้วิจัย: Nguyen (2021)

หลักการทำงาน: ศึกษาความสอดคล้องของจำนวนประตูกับกระบวนการ Poisson และใช้ Poisson Regression เพื่อพยากรณ์ผลการแข่งขัน

แบบจำลองที่ใช้: Poisson Regression

ผลลัพธ์: สามารถพยากรณ์อันดับในตารางลีกได้แม่นยำ

แหล่งที่มา: https://nejsds.nestat.org/journal/NEJSDS/article/1

2.14.5 Predicting Football Results with the Poisson Regression Model

ผู้วิจัย: Artur (2021)

หลักการทำงาน: ใช้ Poisson Regression ทำนายจำนวนประตูที่แต่ละทีมจะทำได้ โดยใช้ข้อมูลสถิติทีมและความได้เปรียบในบ้าน

แบบจำลองที่ใช้: Poisson Regression Model

ผลลัพธ์: ทำนายผลการแข่งขันได้ใกล้เคียงกับผลจริง เช่น Chelsea ชนะ Leicester 2-1

แหล่งที่มา: https://artiebits.com/blog/predicting-football-results-with-statistical-modelling/

2.14.6 Enhancing Football Match Predictions through AI and Machine Learning in the English Premier League

ผู้วิจัย: Langdon Huynh (2024)

หลักการทำงาน: พัฒนาแบบจำลอง MLP, Decision Tree, Random Forest เพื่อทำนายผลพรีเมียร์ลีก ใช้ข้อมูลฤดูกาล 2021/2022 เช่น อัตราการครองบอล, การยิงตรงกรอบ

แบบจำลองที่ใช้: MLP, Decision Tree, Random Forest

ผลลัพธ์: Random Forest มีความแม่นยำสูงสุดที่ 61.54%

แหล่งที่มา: https://nhsjs.com/2024/enhancing-football-match-predictions-through-ai-and-machine-learning-in-the-english-premier-league/

2.14.7 Predicting Football Match Outcomes with Machine Learning Approaches

ผู้วิจัย: Raju et al. (2023)

หลักการทำงาน: เปรียบเทียบการทำนายผลพรีเมียร์ลีกด้วยแบบจำลอง Logistic Regression, Decision Trees, และ Random Forests จากข้อมูลการแข่งขันจริง

แบบจำลองที่ใช้: Logistic Regression, Decision Trees, Random Forests

ผลลัพธ์: Random Forest ให้ความแม่นยำสูงสุดที่ 61.3%

แหล่งที่มา:

https://www.researchgate.net/publication/376680813_Predicting_Football_Match_Outcomes_with_Machine_Learning_Approaches

2.14.8 Predicting the Outcome of Soccer Matches Using Machine Learning and Statistical Analysis

ผู้วิจัย: Ahmed et al. (2022)

หลักการทำงาน: ใช้ Support Vector Machines, Random Forest และ Naïve Bayes ในการทำนายผลการแข่งขันพรีเมียร์ลีกอังกฤษ

แบบจำลองที่ใช้: Support Vector Machines, Random Forest, Naïve Bayes

ผลลัพธ์: แม่นยำสูงสุดที่ 65%

แหล่งที่มา:

https://www.researchgate.net/publication/359023883_Predicting_the_Outcome_of_Soccer_Matches_Using_Machine_Learning_and_Statistical_Analysis

2.14.9 Predicting Final Result of Football Match Using Poisson Regression Model

ผู้วิจัย: Azhari et al. (2018)

หลักการทำงาน: ใช้ข้อมูลทางสถิติและ Poisson Regression ในการทำนายผลการแข่งขันฟุตบอล

แบบจำลองที่ใช้: Poisson Regression

ผลลัพธ์: ความแม่นยำประมาณ 60%

แหล่งที่มา:

https://www.researchgate.net/publication/329387208_Predicting_Final_Result_of_Football_Match_Using_Poisson_Regression_Model

2.14.10 Predicting the Outcome of a Soccer Match Using Machine Learning

ผู้วิจัย: Raju et al. (2023)

หลักการทำงาน: ใช้ Machine Learning เช่น SVM และ Random Forest ทำนายผลการแข่งขัน โดยใช้ข้อมูลทางสถิติย้อนหลัง

แบบจำลองที่ใช้: Support Vector Machines, Random Forest, Naïve Bayes

ผลลัพธ์: ความแม่นยำสูงสุดที่ 65%

แหล่งที่มา: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4992342

จากการทบทวนงานวิจัย พบว่างานวิจัยที่เกี่ยวข้องสามารถแบ่งได้เป็นสองกลุ่มหลัก:

1) การทำนายผลการแข่งขันแบบจำแนก (ชนะ เสมอ หรือแพ้) โดยใช้แบบจำลองแบบ ป่าแบบสุ่ม, ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machines) และ นาอีฟเบส์ (Naive Bayes) ในกลุ่มงานวิจัยแรก งานวิจัยที่ใช้ป่าแบบสุ่ม (Ulmer & Fernandez, 2014; Huynh, 2024; Raju et al., 2023; Ahmed et al., 2022) แสดงถึงประสิทธิภาพที่โดดเด่น โดยมีความแม่นยำ 61–65% เนื่องจากแบบจำลองนี้สามารถจัดการกับข้อมูลที่มีมิติสูง และความสัมพันธ์ที่ไม่เป็นเชิงเส้นได้ดี อย่างไรก็ตาม ความแม่นยำที่จำกัดชี้ให้เห็นถึงความท้าทายในการทำนายผลการแข่งขันที่มีตัวแปรสุ่มสูง เช่น ฟอร์มของผู้เล่นหรือปัจจัยที่ไม่สามารถวัดได้

2) การทำนายจำนวนประตูโดยใช้แบบจำลองเชิงสถิติ เช่น Poisson Regression และ Double Poisson Regression ในกลุ่มงานวิจัยที่สอง งานที่ใช้ Poisson Regression (Loukas et al., 2023; Nguyen, 2021; Azhari et al., 2018) เหมาะสำหรับการทำนายจำนวนประตู เนื่องจากข้อมูลจำนวนประตูมีการแจกแจงแบบ Poisson อย่างไรก็ตาม ข้อจำกัดของแบบจำลองนี้คือการจัดการกับ Overdispersion (ความแปรปรวนที่สูงกว่าค่าเฉลี่ย) ซึ่งอาจลดประสิทธิภาพในบางกรณี งานวิจัยนี้จะต่อยอดจากงานวิจัยก่อนหน้าโดย:

1) ใช้แบบจำลองที่ทันสมัย เช่น LightGBM ซึ่งมีความสามารถในการจัดการข้อมูลขนาดใหญ่และมิติสูงได้ดีกว่าแบบป่าแบบสุ่ม

2) รวมคุณลักษณะใหม่ เช่น ค่าความคาดหวังที่จะได้ประตู (xG) และค่าความคาดหวังที่จะได้แอสซิสต์ (xA) เพื่อเพิ่มความแม่นยำ (การเก็บข้อมูล และคุณลักษณะไม่เหมือนกัน)

3) พัฒนาการจัดการ Overdispersion โดยใช้ Negative Binomial Regression ร่วมกับ Poisson Regression เพื่อเพิ่มประสิทธิภาพในการทำนายจำนวนประตู

การทบทวนงานวิจัยนี้ช่วยให้เห็นถึงจุดแข็ง และข้อจำกัดของแนวทางที่มีอยู่ และเป็นแนวทางในการพัฒนาแบบจำลองที่มีประสิทธิภาพสูงขึ้นสำหรับการทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีกในงานวิจัยนี้

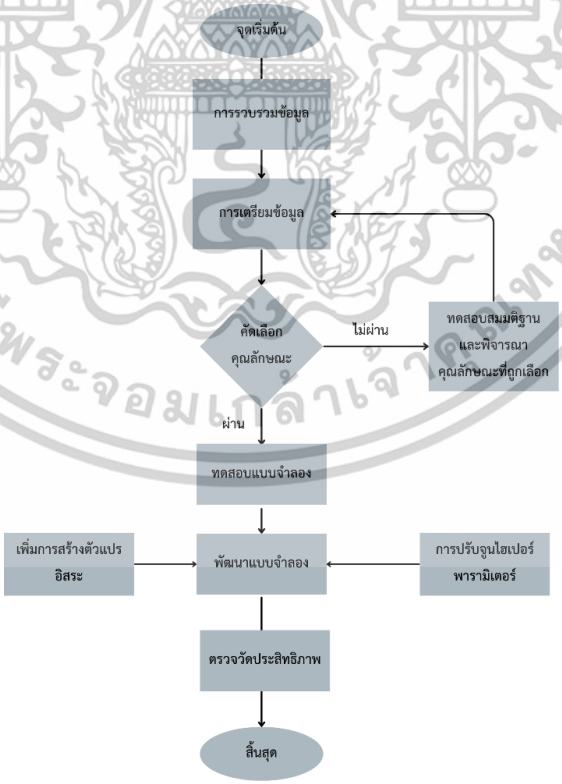
บทที่ 3
วิธีการดำเนินงานวิจัย

วิธีการดำเนินงานของวิจัยนี้ มุ่งเน้นไปที่การสร้างแบบจำลองที่สามารถใช้ทำนายผลฟุตบอลพรีเมียร์ลีกได้ โดยใช้การเรียนรู้ของเครื่องจักร และทำการเปรียบเทียบแบบจำลองในหลายๆแบบ และ การจัดการข้อมูลในแบบต่างๆ เพื่อหาแบบจำลองที่มีประสิทธิภาพที่สุด

3.1 ขั้นตอนการดำเนินงานวิจัย

การดำเนินงานวิจัยในครั้งนี้เริ่มจากการศึกษาแนวคิด และทฤษฎีที่เกี่ยวข้องกับการทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีกโดยอาศัยเทคนิคการเรียนรู้ของเครื่องจักร ซึ่งเน้นศึกษาวิธีการจัดการข้อมูลให้เหมาะสมกับการวิเคราะห์ เช่น การเตรียมข้อมูล การเลือกคุณลักษณะที่สัมพันธ์กับผลการแข่งขัน การเลือก และประยุกต์ใช้เทคนิคต่างๆ ในการสร้างแบบจำลอง การประเมินประสิทธิภาพของแบบจำลอง และกระบวนการพัฒนาแบบจำลองให้ได้ประสิทธิภาพที่สูงขึ้น

เพื่อให้สามารถเข้าใจภาพรวมของกระบวนการวิจัย และขั้นตอนการดำเนินการในแต่ละช่วง ได้จัดทำแผนภาพการทำงานดังแสดงในรูปที่ 3.1



<u>ภาพที่ 3.1</u> แผนภาพกระบวนการวิจัย

3.2 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการวิจัยครั้งนี้คือภาษาโปรแกรมไพธอน (Python) ซึ่งเป็นภาษาที่ได้รับความนิยมอย่างแพร่หลายในการวิเคราะห์ข้อมูล และการประยุกต์ด้านปัญญาประดิษฐ์ (Artificial intelligence) โดยเฉพาะการเรียนรู้ของเครื่องจักร เครื่องมือดังกล่าวถูกใช้งานผ่านแพลตฟอร์ม Google Colaboratory (Colab) ซึ่งเป็นแพลตฟอร์มที่รองรับการเขียน และรันโค้ดออนไลน์ได้อย่างสะดวก โดยมีข้อดีในด้านการเข้าถึงทรัพยากรประมวลผลที่มีประสิทธิภาพ และการเชื่อมต่อกับ Google Drive

การใช้งานไพธอน ครอบคลุมตั้งแต่การรวบรวมข้อมูล การจัดเตรียมข้อมูล การสร้าง และฝึกแบบจำลอง การประเมินผล การเปรียบเทียบโมเดล ตลอดจนการทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีก

ตารางที่ 3.1 ไลบารี่ หรือโมดูลที่ในโปรแกรมภาษาไพธอนที่ใช้ในงานวิจัย

โมดูล (Module)	ฟังก์ชัน (Function)	คำอธิบาย (Description)
numpy	-	คำนวณเชิงตัวเลข เช่น อาเรย์หรือเมทริกซ์
math	-	ฟังก์ชันคณิตศาสตร์พื้นฐาน
scipy.stats	-	ฟังก์ชันทางสถิติทั่วไป
scipy.stats	pearsonr	ค่าสหสัมพันธ์
statsmodels.api	-	การสร้างแบบจำลองแบบ OLS
statsmodels.tsa.api	-	การวิเคราะห์ข้อมูลแบบอนุกรมเวลา
statsmodels.stats.outliers_influence	variance_inflation_factor	ตรวจสอบ multicollinearity ด้วยค่า VIF
sklearn.preprocessing	OrdinalEncoder	แปลง categorical เป็นตัวเลขตามลำดับ
sklearn.feature_selection	mutual_info_regression	ใช้คัดเลือก features ด้วย mutual information
sklearn.pipeline	Pipeline	สร้างกระบวนการ และแบบจำลองร่วมกัน
sklearn.linear_model	PoissonRegressor	Poisson Regression

ตารางที่ 3.1(ต่อ) ไลบารี่ หรือโมดูลที่ในโปรแกรมภาษาไพธอนที่ใช้ในงานวิจัย

โมดูล (Module)	ฟังก์ชัน (Function)	คำอธิบาย (Description)
statsmodels.discrete.count_model	ZeroInflatedPoisson	Zero Inflation Poisson Regression
sklearn.linear_model	nbinom	Negative Binomial Poisson Regression
sklearn.tree	DecisionTreeClassifier	การจำแนกแบบต้นไม้ตัดสินใจ
sklearn.ensemble	RandomForestClassifier	การจำแนกแบบป่าแบบสุ่ม
Xgboost	XGBClassifier	การจำแนกแบบ XGBoost
lightgbm	LGBMClassifier	การจำแนกแบบ LightGBM
sklearn.metrics	accuracy_score	ค่าความแม่นยำของโมเดลในการจำแนก
sklearn.metrics	precision_score	ค่าความแม่นเฉพาะของคลาสบวก
sklearn.metrics	recall_score	ค่าความครอบคลุมของคลาสบวก
sklearn.metrics	f1_score	ค่าเฉลี่ยแบบถ่วงน้ำหนักของ Precision และ Recall
sklearn.metrics	roc_auc_score	พื้นที่ใต้กราฟ ROC
metrics	confusion_matrix	เมทริกซ์แสดงค่าผลลัพธ์จริงเทียบกับค่าที่โมเดลทำนาย
imblearn.under_sampling	RandomUnderSampler	ลดจำนวนตัวอย่างจากคลาสที่มีมากเกินไปแบบสุ่ม
imblearn.over_sampling	RandomOverSampler	เพิ่มจำนวนตัวอย่างในคลาสที่มีน้อยโดยการคัดลอกซ้ำแบบสุ่ม
imblearn.over_sampling	SMOTE	เพิ่มจำนวนตัวอย่างในคลาสที่มีน้อยโดยการสังเคราะห์
sklearn.model_selection	RandomizedSearchCV	ค้นหา hyperparameters แบบสุ่ม

ตารางที่ 3.1(ต่อ) ไลบารี่ หรือโมดูลที่ในโปรแกรมภาษาไพธอนที่ใช้ในงานวิจัย

โมดูล (Module)	ฟังก์ชัน (Function)	คำอธิบาย (Description)
sklearn.model_selection	GridSearchCV	ค้นหา hyperparameters แบบครบทุก combination
matplotlib.pyplot	-	สำหรับสร้างกราฟ
seaborn	-	สำหรับสร้างกราฟ

3.3 การรวบรวมข้อมูล

ข้อมูลที่ใช้ในการวิจัยครั้งนี้เป็น ข้อมูลทุติยภูมิ (Secondary Data) ที่รวบรวมจากเว็บไซต์ www.fbref.com ซึ่งเป็นแหล่งข้อมูลทางสถิติที่เชื่อถือได้ในวงการฟุตบอล โดยข้อมูลครอบคลุมการแข่งขันพรีเมียร์ลีกตั้งแต่ฤดูกาล 2016-2017 จนถึงฤดูกาล 2023-2024 ประเภทของข้อมูลที่รวบรวมประกอบด้วยสถิติเชิงลึกที่เกี่ยวข้องกับการแข่งขันฟุตบอลในหลากหลายมิติ ได้แก่ การยิงประตู (Shooting), การส่งบอล (Passing), ประเภทของการส่งบอล (Passing Type), ผู้รักษาประตู (Goalkeeping), การป้องกัน (Defend), การครองบอล (Possession), การสร้างโอกาสยิง และทำประตู (Shot and Goal Creating Actions), สถิติทั่วไประหว่างการแข่งขัน (Miscellaneous)

ข้อมูลข้างต้นถูกแยกเก็บเป็นตารางต่าง ๆ ตามหัวข้อที่กล่าวมา โดยครอบคลุมการแข่งขันในแต่ละฤดูกาล เพื่อให้สามารถเชื่อมโยงข้อมูลเหล่านี้เข้ากับแต่ละแมตช์การแข่งขันได้อย่างถูกต้อง ผู้วิจัยได้ดำเนินการเพิ่มรหัสประจำการแข่งขัน (Match ID) ให้กับแต่ละตารางย่อย

ในขั้นตอนต่อมา ผู้วิจัยได้ทำการรวมข้อมูลจากแต่ละตารางเข้าด้วยกันโดยใช้ Match ID เป็นคีย์หลักในการเชื่อมโยง เพื่อสร้างชุดข้อมูลหลัก (Master Dataset) ที่มีข้อมูลครบถ้วน และครอบคลุมทุกการแข่งขันผลลัพธ์ของกระบวนการนี้คือชุดข้อมูลที่มีจำนวนทั้งหมด 5,320 แถว และ 164 คุณลักษณะ (Features)

ตารางที่ 3.2 ผลการแข่งขัน และสถิติภาพรวม (Score and Fixtures)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Match_ID	ตัวระบุเฉพาะสำหรับแต่ละการแข่งขัน	Object
Match_Date	วันที่ทำการแข่งขัน	Object
Match_Time	เวลาที่เริ่มทำการแข่งขัน	Object

ตารางที่ 3.2(ต่อ) ผลการแข่งขัน และสถิติภาพรวม (Score and Fixtures)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Season	ฤดูกาลของการแข่งขัน	Object
Competition	ชื่อรายการการแข่งขัน/ลีก	Object
Round	รอบของการแข่งขัน	Object
Day	วันของสัปดาห์ที่ทำการแข่งขัน	Object
Venue	ทีมเหย้า/ทีมเยือน	Object
Team	ชื่อทีมที่กำลังวิเคราะห์สถิติในแถวนั้นๆ	Object
Formation	แผนการเล่นของทีมในนัดนั้นๆ	Object
Captain	ชื่อกัปตันทีมในนัดนั้นๆ	Object
Opponent	ชื่อทีมคู่แข่งในนัดนั้นๆ	Object
Goals_for	จำนวนประตูที่ทีมทำได้ในนัดนั้น	Int64
Goal_Against	จำนวนประตูที่ทีมเสียในนัดนั้น	Int64
xG	ค่าความน่าจะเป็นของการเกิดประตูที่ทีมควรทำได้	Float64
xGA	ค่าความน่าจะเป็นของการเกิดประตูที่ทีมคู่แข่งควรทำได้	Float64
Possession	เปอร์เซ็นต์การครองบอลของทีมในนัดนั้น	Int64
Match_Result	ผลการแข่งขันสำหรับทีม	Object
Referee	ชื่อผู้ตัดสินในนัดนั้นๆ	Object

ตารางที่ 3.3 สถิติการยิง (Shooting)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Goals	จำนวนประตูที่ทีมทำได้ในนัดนั้น โดยมีผู้เล่นเป็นผู้ทำประตู	Int64
Shots	จำนวนการยิงทั้งหมดของทีม	Int64
Shots_On_Target	จำนวนการยิงที่เข้ากรอบของทีม	Int64
Shots_On_Target %	เปอร์เซ็นต์ของการยิงที่เข้ากรอบจากการยิงทั้งหมด	Float64
Goals_Per_Shot	อัตราส่วนจำนวนประตูที่ทำได้ต่อจำนวนการยิงทั้งหมด	Float64
Goals_Per_SoT	อัตราส่วนจำนวนประตูที่ทำได้ต่อจำนวนการยิงเข้ากรอบ	Float64
Shot_Distance	ระยะทางเฉลี่ยของการยิงทั้งหมด (ตำแหน่งการยิงถึงประตู)	Float64
Freekick_Shots	จำนวนการยิงที่มาจากลูกฟรีคิก	Int64
Penalty_Goals	จำนวนประตูที่ทำได้จากลูกจุดโทษ	Int64
Penalty_Attemps	จำนวนครั้งที่ทีมได้ยิงลูกจุดโทษ	Int64
xG	ค่าความน่าจะเป็นของการเกิดประตูที่ทีมควรทำได้	Float64
Non_Pen_xG	ค่าความน่าจะเป็นของการเกิดประตูที่ไม่รวมจากลูกจุดโทษ	Float64
Non_Pen_xG_Per_Shot	อัตราส่วนของค่าความน่าจะเป็นของการเกิดประตูที่ไม่รวมจากลูกจุดโทษต่อจำนวนการยิงทั้งหมด (ไม่นับจุดโทษ)	Float64
Goal_xG_Diff	ความแตกต่างระหว่างจำนวนประตูที่ทำได้จริง	Float64
Goal_xG_NoPen_Diff	ความแตกต่างระหว่างจำนวนประตูที่ไม่นับจุดโทษกับค่าค่าความน่าจะเป็นของการเกิดประตูที่ไม่รวมจากลูกจุดโทษ	Float64

ตารางที่ 3.4 สถิติผู้รักษาประตู (Goalkeeping)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
GK_Shots_On_Target_Against	จำนวนการยิงเข้ากรอบที่ผู้รักษาประตูต้องเผชิญ	Int64
Goals_Against	จำนวนประตูที่ทีมเสียในนัดนั้น	Int64
GK_Saves	จำนวนครั้งที่ผู้รักษาประตูเซฟลูกยิงได้	Int64
Saves%	เปอร์เซ็นต์mผู้รักษาประตูเซฟลูกยิงได้	Float64
GK_Clean_Sheets	คลีนชีทในนัดนั้นหรือไม่ (0 คือไม่ 1 คือใช่)	Int64
GK_PostShot_xG	ค่าความยากของการเซฟหลังจากลูกถูกยิงแล้ว	Float64
GK_PostShot_xG_Diff	ผลต่างระหว่างจำนวนการเซฟจริงกับ Post-Shot xG	Float64
GK_PK_Att	จำนวนลูกจุดโทษที่ผู้รักษาประตูต้องเจอ	Int64
GK_PK_Allowed	จำนวนลูกจุดโทษที่ผู้รักษาประตูไม่สามารถเซฟได้	Int64
GK_PK_Saved	จำนวนลูกจุดโทษที่เซฟได้	Int64
GK_PK_Missed	จำนวนลูกจุดโทษที่ฝ่ายตรงข้ามยิงออกเอง	Int64
GK_Launched_Cmp	จำนวนการส่งบอลเกิน 40 หลาที่สำเร็จโดยผู้รักษาประตู	Int64
GK_Lanched_Att	จำนวนการส่งบอลเกิน 40 หลาที่โดยผู้รักษาประตู	Int64
GK_Launched_Cmp%	เปอร์เซ็นต์ความแม่นของการส่งบอลเกิน 40 หลาสำเร็จ	Float64
GK_Pass_Att	จำนวนการส่งบอลทั้งหมดของผู้รักษาประตู ไม่รวมตั้งเตะ	Int64
GK_Throw_Att	จำนวนการทุ่มบอลด้วยมือของผู้รักษาประตู	Int64
GK_Pass_Launch%	เปอร์เซ็นที่ผู้รักษาประตูจะส่งบอลเกิน 40 หลา	Float64
GK_Pass_AvgLen	ระยะเฉลี่ยที่ผู้รักษาประตูส่งบอล	Float64
GK_GK_Att	จำนวนการตั้งเตะจากประตูทั้งหมด	Int64
GK_GK_Launch%	เปอร์เซ็นที่การส่งบอลเกิน 40 หลาจากลูกตั้งเตะ	Float64
GK_GK_AvgLen	ระยะเฉลี่ยที่ผู้รักษาประตูส่งบอลจากลูกตั้งเตะ	Float64

ตารางที่ 3.4(ต่อ) สถิติผู้รักษาประตู (Goalkeeping)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
GK_Crosses_ Faced	จำนวนครั้งที่ผู้รักษาประตูต้องเจอกับลูกครอส	Int64
GK_Crosses_ Stopped	จำนวนครั้งที่ผู้รักษาประตูป้องกันลูกครอส	Int64
GK_Stopped%	เปอร์เซ็นที่ผู้รักษาประตูจะป้องกันลูกครอส	Float64
GK_DefAct_onPen	จำนวนครั้งที่ผู้รักษาประตูป้องกันได้จากนอกเขตลูกโทษ	Int64
GK_AvgDist	ระยะเฉลี่ยที่ผู้รักษาประตูป้องกันได้จากนอกเขตลูกโทษ	Float64

ตารางที่ 3.5 สถิติการส่งบอล (Passing)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Total_Cmp	จำนวนการส่งบอลสำเร็จ	Int64
Total_Att	จำนวนการส่งบอลทั้งหมด	Int64
Total_Cmp%	เปอร์เซ็นการส่งบอลสำเร็จ	Float64
Pass_Total_Dist	ระยะทางรวมในการส่งบอลทั้งหมด	Int64
Pass_Prog_Dist	ระยะทางรวมในการส่งบอลไปข้างหน้า	Int64
Pass_Short_Att	จำนวนการส่งบอลระยะสั้น (5-15 หลา) ทั้งหมด	Int64
Pass_Short_Cmp	จำนวนการส่งบอลระยะสั้นสำเร็จ	Int64
Pass_Short_Cmp %	เปอร์เซ็นการที่ส่งบอลระยะสั้นสำเร็จ	Float64
Pass_Med_Att	จำนวนการส่งบอลระยะกลาง (15-30 หลา) ทั้งหมด	Int64
Pass_Med_Cmp	จำนวนการส่งบอลระยะกลางสำเร็จ	Int64
Pass_Med_Cmp%	เปอร์เซ็นการที่ส่งบอลระยะกลางสำเร็จ	Float64
Pass_Long_Att	จำนวนการส่งบอลระยะยาว (มากกว่า 30 หลา) ทั้งหมด	Int64
Pass_Long_Cmp	จำนวนการส่งบอลระยะยาวสำเร็จ	Int64
Pass_Long_Cmp%	เปอร์เซ็นการที่ส่งบอลระยะยาวสำเร็จ	Float64
Assists	จำนวนการเป็นผู้ช่วยทำประตู	Int64

ตารางที่ 3.5(ต่อ) สถิติการส่งบอล (Passing)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
xAG	ค่าความน่าจะเป็นของการเกิดประตูที่มาจากการแอสซิสต์	Float64
xA	ค่าความน่าจะเป็นของการเป็นผู้ช่วยทำประตู	Float64
Key_Passes	จำนวนการส่งบอลที่นำไปสู่โอกาสในการยิง	Int64
Pass_into_Final_Third	จำนวนการส่งบอลเข้าสู่พื้นที่แดนสุดท้ายของคู่ต่อสู้	Int64
Pass_into_Pen	จำนวนการส่งบอลเข้าสู่กรอบเขตโทษคู่ต่อสู้	Int64
Crosses_into_Pen	จำนวนการครอสบอลเข้าสู่กรอบเขตโทษคู่ต่อสู้	Int64
Progressive_Passes	จำนวนการส่งบอลที่ไปข้างหน้า	Int64

ตารางที่ 3.6 สถิติประเภทการส่งบอล (Pass type)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Passes_Live	จำนวนการส่งบอลในสถานการณ์ปกติ	Int64
Passes_Dead	จำนวนการส่งบอลในสถานการณ์ลูกตาย	Int64
Freekick_Passes	จำนวนการส่งบอลจากลูกฟรีคิก	Int64
ThroughBalls	จำนวนการส่งบอลทะลุช่อง	Int64
Switches	จำนวนการส่งบอลที่ระยะมากกว่า 40 หลาในด้านกว้างของสนาม	Int64
Crosses	จำนวนการครอส	Int64
Throwins	จำนวนการทุ่มบอล	Int64
Cornerkicks	จำนวนลูกเตะมุม	Int64
CK_Inswing	จำนวนลูกเตะมุมที่เปิดเข้าหาประตู	Int64
CK_Outswing	จำนวนลูกเตะมุมที่เปิดออกจากประตู	Int64
CK_Straight	จำนวนลูกเตะมุมแบบที่เปิดพุ่งเข้าหาประตู	Int64
Passes_Cmp	จำนวนการส่งบอลสำเร็จ	Int64

ตารางที่ 3.6(ต่อ) สถิติประเภทการส่งบอล (Pass type)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Passes_Offside	จำนวนการส่งบอลที่เกิดการล้ำหน้า	Int64
Passes_Blocked	จำนวนการส่งบอลที่ถูกสกัดกั้น	Int64

ตารางที่ 3.7 สถิติโอกาสสร้างสรรค์ในการยิง และการทำประตู(Goal and Shot Creation)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Shot_Creating_ Actions	จำนวนครั้งที่ผู้เล่นมีส่วนร่วมให้เกิดโอกาสในการยิง (SCA)	Int64
SCA_Pass_Live	จำนวน SCA จากการส่งในสถานการณ์ปกติ	Int64
SCA_Pass_Dead	จำนวน SCA จากการส่งในสถานการณ์ลูกตาย	Int64
SCA_TakeOn	จำนวน SCA จากการเลี้ยงบอลผ่าน	Int64
SCA_Shot	จำนวน SCA จากการยิงประตู (เพื่อนร่วมทีมยิงซ้ำ)	Int64
SCA_Fouls_Drawn	จำนวน SCA จากการที่ได้ฟาว์ล	Int64
SCA_Def_Actions	จำนวน SCA จากการแย่งบอล	Int64
Goal_Creating_ Actions	จำนวนครั้งที่ผู้เล่นมีส่วนร่วมในการได้ประตู (GCA)	Int64
GCA_Pass_Live	จำนวน GCA จากการส่งในสถานการณ์ปกติ	Int64
GCA_Pass_Dead	จำนวน GCA จากการส่งในสถานการณ์ลูกตาย	Int64
GCA_TakeOn	จำนวน GCA จากการเลี้ยงบอลผ่าน	Int64
GCA_Shot	จำนวน GCA จากการยิงประตู (เพื่อนร่วมทีมยิงซ้ำ)	Int64
GCA_Fouls_Drawn	จำนวน GCA จากการที่ได้ฟาว์ล	Int64
GCA_Def_Actions	จำนวน GCA จากการแย่งบอล	Int64

ตารางที่ 3.8 สถิติการป้องกัน (Defensive Action)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Tackles	จำนวนการเข้าสกัดบอลทั้งหมด	Int64
Tackles_Won	จำนวนการเข้าสกัดบอลสำเร็จ	Int64
Tkl_Def_Third	จำนวนการเข้าสกัดบอลในพื้นที่แดนตัวเอง	Int64
Tkl_Mid_Third	จำนวนการเข้าสกัดบอลในพื้นที่กลางสนาม	Int64
Tkl_Att_Third	จำนวนการเข้าสกัดบอลในพื้นที่แดนคู่ต่อสู้	Int64
Tkl_Drib_Won	จำนวนครั้งที่เข้าสกัดเมื่อผู้แข่งเลี้ยงบอลสำเร็จ	Int64
Tkl_DribContest	จำนวนครั้งที่เข้าสกัดเมื่อผู้แข่งเลี้ยงบอลทั้งหมด	Int64
Tkl_Drib_Won%	เปอร์เซ็นเข้าสกัดเมื่อผู้แข่งเลี้ยงบอลสำเร็จ	Float64
Tkl_Drib_Lost	จำนวนครั้งที่เข้าสกัดเมื่อผู้แข่งเลี้ยงบอลไม่สำเร็จ	Int64
Blocks	จำนวนการบล็อกลูกยิงหรือลูกส่งทั้งหมด	Int64
Shot_Blocks	จำนวนการบล็อกลูกยิง	Int64
Pass_Blocks	จำนวนการบล็อกลูกส่ง	Int64
Interception	จำนวนการตัดบอล	Int64
Tkl_Int	ผลรวมของจำนวนการเข้าสกัดบอลและการตัดบอล	Int64
Clearances	จำนวนการเตะเคลียร์บอล	Int64

ตารางที่ 3.9 สถิติครอบครองบอล (Possession)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Touches	จำนวนครั้งที่สัมผัสบอล	Int64
Touches_Def_Pen	จำนวนครั้งที่สัมผัสบอลในกรอบเขตโทษแดนตัวเอง	Int64
Touches_Def_ Third	จำนวนครั้งที่สัมผัสบอลในพื้นที่แดนตัวเอง	Int64
Touches_Mid	จำนวนครั้งที่สัมผัสบอลในพื้นที่กลางสนาม	Int64
Touches_Att_ Third	จำนวนครั้งที่สัมผัสบอลในพื้นที่แดนคู่ต่อสู้	Int64
Touches_Att_Pen	จำนวนครั้งที่สัมผัสบอลในกรอบเขตโทษคู่ต่อสู้	Int64
Touches_Live	จำนวนครั้งที่สัมผัสบอลในสถานการณ์ปกติ	Int64
Drib_Att	จำนวนครั้งที่เลี้ยงบอลผ่านคู่ต่อสู้	Int64
Drib_Succ	จำนวนการเลี้ยงบอลผ่านคู่ต่อสู้สำเร็จ	Int64
Drib_Succ%	เปอร์เซ็นต์ในการเลี้ยงบอลผ่านคู่ต่อสู้สำเร็จ	Float64
Drib_Tkld	จำนวนครั้งที่ถูกคู่ต่อสู้เข้าสกัดบอลได้ขณะเลี้ยงบอล	Int64
Drib_Tkld%	เปอร์เซ็นต์การถูกเข้าสกัดบอลสำเร็จขณะเลี้ยงบอล	Float64
Carries	จำนวนครั้งที่ครองบอลและเคลื่อนที่ไปกับบอล	Int64
Carries_TotalDist	ระยะทางรวมที่เคลื่อนที่ไปกับบอล	Float64
Carries_ProgDist	ระยะทางรวมที่เคลื่อนที่ไปกับบอลไปข้างหน้า	Float64
Carries_Prog	จำนวนครั้งที่ครองบอลและเคลื่อนที่ไปกับบอลไปข้างหน้า	Int64
Carries_into_ FinalT	จำนวนครั้งที่เคลื่อนที่ไปกับบอลเข้าสู่พื้นที่แดนคู่ต่อสู้	Int64
Carries_into_Pen	จำนวนครั้งที่เคลื่อนที่ไปกับบอลเข้าสู่กรอบเขตโทษคู่ต่อสู้	Int64
Miscontrols	จำนวนครั้งที่จับบอลพลาดทำให้เสียการครองบอล	Int64
Dispossed	จำนวนครั้งที่ถูกแย่งบอลไปได้	Int64
Received	จำนวนการรับบอล	Int64
Prog_Received	จำนวนการรับบอลที่เป็นการส่งบอลไปข้างหน้า	Int64

ตารางที่ 3.10 สถิติต่างๆ (Miscellaneous stats)

คุณลักษณะ (Feature)	ความหมายของคุณลักษณะ (Description)	ชนิดของตัวแปร (Data types)
Yellow_Cards	จำนวนใบเหลืองที่ได้รับ	Int64
Red_Cards	จำนวนใบแดงที่ได้รับ	Int64
2Yellow_Cards	จำนวนการได้ใบเหลืองที่สองซึ่งนำไปสู่ใบแดง	Int64
Fouls_Committed	จำนวนการทำฟาวล์	Int64
Fouls_Drawn	จำนวนครั้งที่ถูกทำฟาวล์	Int64
Offsides	จำนวนการล้ำหน้า	Int64
Crosses	จำนวนการครอสบอลทั้งหมด	Int64
Interception	จำนวนการตัดบอล	Int64
Tackles_Won	จำนวนการเข้าสกัดบอลสำเร็จ	Int64
Penalty_Won	จำนวนครั้งที่ทีมได้ลูกจุดโทษ	Int64
Penalty_ Conceded	จำนวนครั้งที่ทีมเสียลูกจุดโทษ	Int64
Own_Goals	จำนวนการทำเข้าประตูตัวเอง	Int64
Recoveries	จำนวนการแย่งบอลกลับคืนมาได้	Int64
Aerial_Won	จำนวนการดวลลูกกลางอากาศที่สำเร็จ	Int64
Aerial_Lost	จำนวนการดวลลูกกลางอากาศที่ไม่สำเร็จ	Int64
Aerial_Won%	เปอร์เซ็นต์การดวลลูกกลางอากาศที่สำเร็จ	Float64

3.4 การเตรียมข้อมูล และการวิเคราะห์ข้อมูล

ขั้นตอนการเตรียมข้อมูล (Data Preparation) เป็นขั้นตอนที่สำคัญอย่างยิ่ง เพื่อให้ข้อมูลมีความถูกต้อง สมบูรณ์ และเหมาะสมกับการฝึกฝนแบบจำลอง ซึ่งขั้นตอนที่ใช้ในการดำเนินวิจัยดังต่อไปนี้

3.4.1 การตรวจสอบ และทำความสะอาดข้อมูล (Data Cleaning)

หลักจากที่ข้อมูลได้ถูกรวมเข้าด้วยกัน (Merging) จากตารางต่างๆ แล้ว พบว่ามีข้อมูลที่มีคุณลักษณะซ้ำซ้อน หรือผิดปกติ จึงต้องเกิดขั้นตอนการทำสะอาดข้อมูลเสียก่อนที่จะเข้ากระบวนการจัดเตรียมข้อมูล โดยในการดำเนินวิจัยใช้ขั้นตอนการทำความสะอาดข้อมูลดังต่อไปนี้

3.4.1.1 การจัดการกับข้อมูลที่หายไป (Missing Value)

ข้อมูลที่หายไปเป็นปัญหาที่พบได้บ่อยในชุดข้อมูล และอาจส่งผลกระทบต่อความถูกต้องของข้อมูล และการสร้างแบบจำลองที่มีประสิทธิภาพ ดังนั้นการจัดการกับข้อมูลที่หายไปอย่างเหมาะสมจึงเป็นขั้นตอนที่สำคัญ ซึ่งวิธีการจัดการโดยทั่วไปประกอบด้วย การลบแถว (Row Deletion), การลบคอลัมน์ (Column Deletion) หรือจะเป็นการแทนที่ข้อมูล (Imputation) แต่จากการสำรวจข้อมูลของผู้วิจัย ไม่พบข้อมูลที่หายไปในชุดข้อมูล

3.4.1.2 การจัดการกับข้อมูลซ้ำ (Duplicate data)

ข้อมูลที่มีลักษณะเหมือนกันทุกประการในหลายๆแถว และคอลั่ม ซึ่งอาจเกิดขึ้นจากการรวบรวมข้อมูลจากหลายแหล่ง หรือเกิดข้อผิดพลาดในกระบวนการบันทึกข้อมูล ข้อมูลซ้ำอาจทำให้การวิเคราะห์ข้อมูลเกิดความเอนเอียง และส่งผลต่อประสิทธิภาพของแบบจำลอง การจัดการกับข้อมูลซ้ำสามารถทำได้โดยการระบุแถวที่ซ้ำกัน และทำการลบทิ้ง โดยต้องพิจารณาถึงผลกระทบของการลบข้อมูลซ้ำต่อขนาดของชุดข้อมูลด้วย

ตารางที่ 3.11 คุณลักษณะที่มีค่าซ้ำกันในทุกแถว

คุณลักษณะที่ 1	คุณลักษณะที่ 2	หมายเหตุ (Remark)
xG (ตารางที่ 3.1)	xG (ตารางที่ 3.2)	เลือกลบจากตารางที่ 3.2
Possession (ตารางที่ 3.1)	Possession (ตารางที่ 3.8)	เลือกลบจากตารางที่ 3.8
Tackles_Won (ตารางที่ 3.7)	Tackles_Won (ตารางที่ 3.9)	เลือกลบจากตารางที่ 3.9
Interception (ตารางที่ 3.7)	Interceptions (ตารางที่ 3.9)	เลือกลบจากตารางที่ 3.9
Passses_Offsides (ตารางที่ 3.5)	Offsides (ตารางที่ 3.9)	เลือกลบจากตารางที่ 3.9
Crosses (ตารางที่ 3.5)	Crosses (ตารางที่ 3.9)	เลือกลบจากตารางที่ 3.9
Total_Cmp (ตารางที่ 3.4)	Total_Cmp (ตารางที่ 3.5)	เลือกลบจากตารางที่ 3.5
Total_Att (ตารางที่ 3.4)	Total_Att (ตารางที่ 3.5)	เลือกลบจากตารางที่ 3.5

3.4.1.3 การจัดการกับข้อมูลไม่สอดคล้องกัน (Inconsistency data)

ข้อมูลที่มีความไม่สอดคล้องกันในด้านรูปแบบหรือหน่วยวัด เช่น รูปแบบการเขียนชื่อทีมไม่เป็นมาตรฐาน หรือใช้คำย่อที่แตกต่างกัน ซึ่งอาจส่งผลให้เกิดข้อผิดพลาดในการรวมและวิเคราะห์ข้อมูล

ตารางที่ 3.12 รายชื่อทีมมาตรฐาน

ชื่อทีมก่อนปรับ (Team name)	ชื่อตามมาตรฐาน (Standard name)
Brighton	Brighton and Hove Albion
Huddersfield	Huddersfield Town
Manchester Utd	Manchester United
Newcastle Utd	Newcastle United
Nott'ham Forest	Nottingham Forest
Sheffield Utd	Sheffield United
Tottenham	Tottenham Hotspur
West Brom	West Bromwich Albion
West Ham	West Ham United
Wolves	Wolverhampton Wanderers
Brighton	Brighton and Hove Albion

3.4.1.4 การจัดการกับข้อมูลไม่ผิดประเภท (Incorrect data type)

ข้อมูลที่ถูกจัดเก็บอยู่ในประเภทข้อมูลที่ไม่ถูกต้องตามลักษณะของข้อมูลนั้นๆ ตัวอย่างเช่น ตัวแปรที่ควรจะเป็นตัวเลข (Numerical) แต่ถูกจัดเก็บเป็นข้อความ (String) หรือตัวแปรวันที่ (Date) ถูกจัดเก็บเป็นตัวเลข ข้อมูลที่มีประเภทไม่ถูกต้องอาจทำให้การวิเคราะห์ และการสร้างแบบจำลองเกิดข้อผิดพลาด หรือไม่สามารถดำเนินการได้

3.4.1.5 การจัดการกับข้อมูลผิดปกติ (Outlier)

ข้อมูลที่มีค่าแตกต่างจากค่าส่วนใหญ่ในชุดข้อมูลอย่างมาก อาจเกิดขึ้นจากความผิดพลาดในการวัด การบันทึก หรืออาจเป็นค่าที่แท้จริงแต่มีความสุดโต่ง Outlier สามารถส่งผลกระทบอย่างมากต่อการวิเคราะห์ทางสถิติ และประสิทธิภาพของแบบจำลอง โดยเฉพาะแบบจำลองที่ไวต่อขนาดของข้อมูล การจัดการกับ Outlier สามารถทำได้โดยการระบุ Outlier ด้วยวิธีการทางสถิติ (เช่น IQR, Z-score) หรือการแสดงผลด้วยภาพ (เช่น Box Plot) จากนั้นอาจพิจารณาการลบ Outlier, การแปลงค่า หรือการใช้แบบจำลองที่ไม่ไวต่อ Outlier

3.4.2 การเตรียมข้อมูล

ขั้นตอนในการปรับปรุง และเปลี่ยนแปลงข้อมูลที่ผ่านการทำความสะอาดแล้ว ให้อยู่ในรูปแบบที่เหมาะสมสำหรับการนำไปใช้ในการสร้าง และฝึกฝนแบบจำลองการเรียนรู้ของเครื่องจักร ซึ่งขั้นตอนที่ใช้ในการดำเนินวิจัยดังต่อไปนี้

3.4.2.1 การแปลงข้อมูลหมวดหมู่ (Categorical Encoding)

ข้อมูลที่แสดงถึงกลุ่มหรือประเภท เช่น สี, ประเภททีม, หรือผลการแข่งขัน แบบจำลองส่วนใหญ่ต้องการตัวแปรอิสระที่เป็นตัวเลข ดังนั้นจึงจำเป็นต้องมีการแปลงข้อมูลหมวดหมู่ให้เป็นรูปแบบตัวเลข เทคนิคที่นิยมใช้ในการแปลงข้อมูลหมวดหมู่ ได้แก่:

1) การเข้ารหัสแบบวันฮอท (One-Hot Encoding) เป็นการแปลงค่าหมวดหมู่ให้เป็นตัวแปรบูลีน โดยสร้างคอลัมน์ใหม่สำหรับแต่ละหมวดหมู่ แล้วระบุค่าเป็น 1 หากแถวนั้นอยู่ในหมวดหมู่นั้น และเป็น 0 หากไม่ใช่ วิธีนี้เหมาะสำหรับข้อมูลที่ไม่มีลำดับขั้น (Nominal Data)

2) การเข้ารหัสแบบป้ายกำกับ (Label Encoding) เป็นการแทนค่าหมวดหมู่ด้วยหมายเลขแทน โดยกำหนดลำดับตัวเลขให้กับแต่ละกลุ่ม

3) การเข้ารหัสแบบค่าเฉลี่ย (Mean Encoding) เป็นการแทนค่าหมวดหมู่ด้วยค่าเฉลี่ยของตัวแปรเป้าหมาย (Target Variable) สำหรับแต่ละหมวดหมู่ วิธีนี้มีความยืดหยุ่น และสามารถช่วยเพิ่มประสิทธิภาพของโมเดลได้ดี

ตารางที่ 3.13 วิธีการเข้ารหัสข้อมูล

คุณลักษณะ (Feature)	วิธีการเข้ารหัส (Encoding)	หมายเหตุ (Remark)
Match_ID	-	-
Season	Label Encoding	-
Round	Label Encoding	-
Day	Label Encoding	-
Venue	Label Encoding	-
Team	Mean Encoding	-
Formation	Mean Encoding	-
Captain	-	มีหลากหลาย (High Cardinality)

ตารางที่ 3.13(ต่อ) วิธีการเข้ารหัสข้อมูล

คุณลักษณะ (Feature)	วิธีการเข้ารหัส (Encoding)	หมายเหตุ (Remark)
Opponent	Mean Encoding	
Match_Result	Label Encoding	
Referee	-	มีหลากหลาย (High Cardinality)

3.4.2.2 การปรับขนาดข้อมูล (Normalization)

การปรับขนาดข้อมูล หรือการปรับมาตรฐานข้อมูล (Standardization) เป็นกระบวนการปรับค่าของตัวแปรให้มีช่วง หรือสเกลที่คล้ายกัน โดยการวิจัยนี้ใช้การปรับด้วยค่าสูงต่ำ (Min-Max Scaling) ปรับค่าให้อยู่ในช่วง 0 ถึง 1 ซึ่งช่วยให้ข้อมูลทุกตัวแปรมีช่วงค่าที่เท่ากัน ซึ่งส่งผลให้โมเดลสามารถเรียนรู้จากข้อมูลแต่ละคุณลักษณะได้อย่างเท่าเทียมกัน ลดโอกาสการเรียนรู้ที่ลำเอียงจากค่าที่มีขนาดใหญ่กว่า หรือเล็กกว่ามาก

3.4.2.3 การจัดการกับข้อมูลที่เป็นข้อมูลหลังเกม (Data Leakage)

โดยการเก็บข้อมูล ข้อมูลส่วนใหญ่ที่ได้มาจะเป็นข้อมูลที่ได้หลังจบการแข่งขัน เช่นจำนวนการยิงประตู (Goals_For), จำนวนประตูที่เสีย (Goals Against) และยังเป็นผลเฉลยให้กับข้อมูลอย่างเช่น ผลต่างข้อมูลจำนวนการยิงประตู และการเสียประตู เพียงแค่ 2 คุณลักษณะนี้สามารถใช้บอกผลลัพธ์ของการทำนายได้เลย จึงได้ทำข้อมูลเหล่านี้เป็นค่าเฉลี่ย (Rolling Average) โดยใช้เป็น ค่าเฉลี่ย 3 นัดและ 5 นัดตามลำดับ โดยเปลี่ยนคุณลักษณะต่างๆ โดยที่ลงท้ายด้วย _roll3 แสดงว่าเป็นค่าเฉลี่ย 3 นัด และ _roll5 แสดงว่าเป็นค่าเฉลี่ย 5 นัด

3.4.2.4 การเลือกคุณลักษณะที่สำคัญ (Feature Selection)

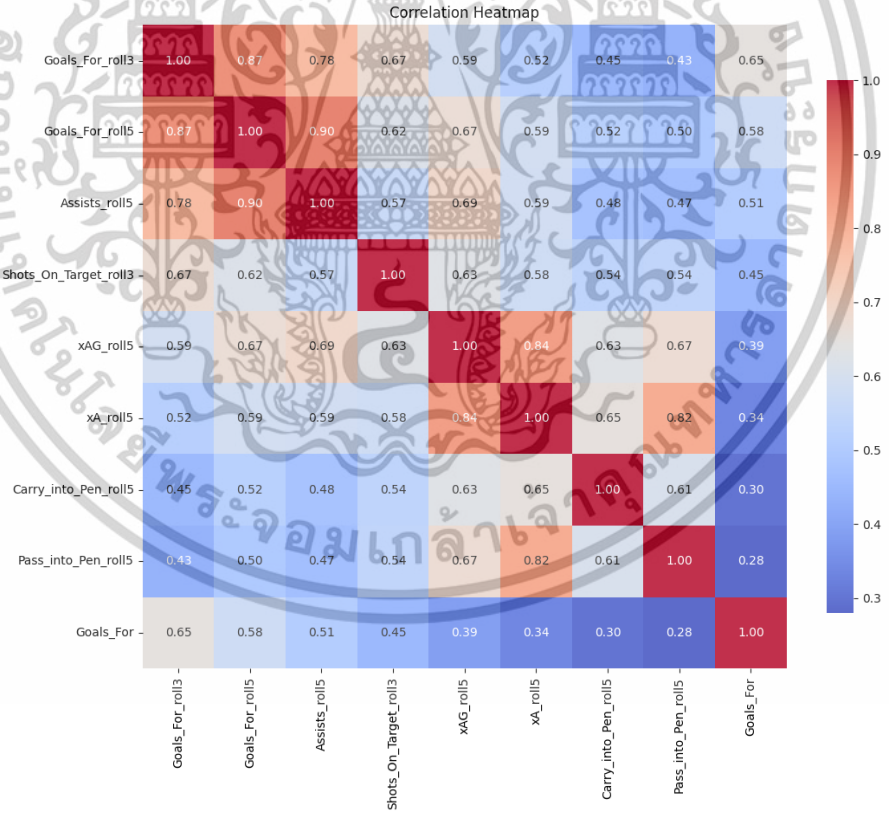
การเลือกคุณลักษณะที่สำคัญ (Feature Selection) เป็นกระบวนการระบุและเลือกชุดของคุณลักษณะ (ตัวแปรอิสระ) ที่มีความเกี่ยวข้องมากที่สุดกับตัวแปรตาม เพื่อลดความซับซ้อนของแบบจำลอง ป้องกันปัญหา Overfitting และปรับปรุงประสิทธิภาพ เทคนิคที่ใช้ในการเลือกคุณลักษณะ ดังนี้

1) เลือกจากค่าสหสัมพันธ์ (Correlation) ใช้วิธีการวิเคราะห์ค่าสหสัมพันธ์ (Correlation Coefficient) ของ Pearson เพื่อตรวจสอบความสัมพันธ์เชิงเส้นระหว่างตัวแปรอิสระกับตัวแปรตาม โดยพิจารณาเลือกเฉพาะตัวแปรที่มีค่า

สหสัมพันธ์สูง และตัดตัวแปรที่มีค่าสหสัมพันธ์ซ้ำซ้อนระหว่างกันออกเพื่อลดความซ้ำซ้อน (Multicollinearity)

ตารางที่ 3.14 การเลือกคุณลักษณะด้วยค่าสหสัมพันธ์ของตัวแปรตาม Goals_For

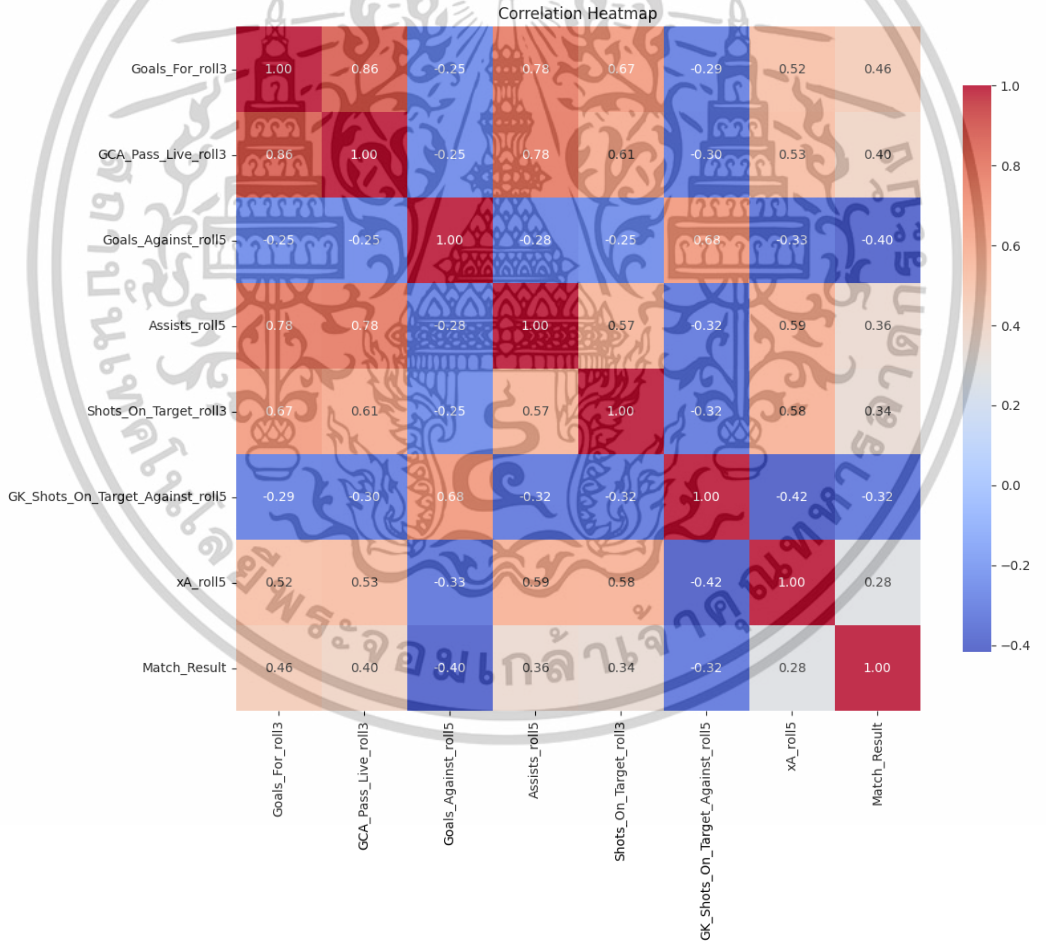
ตัวแปรตาม (Target)	เงื่อนไข (Condition)	คุณลักษณะ (Feature)
Goals_For	มีค่า Corr > 0.2 ไม่มีปัญหาเรื่อง Multicollinearity	'Goals_For_roll3', 'Goals_For_roll5', 'Assists_roll5', 'Shots_On_Target_roll3', 'xAG_roll5', 'xA_roll5', 'Carry_into_Pen_roll5', 'Pass_into_Pen_roll5'



ภาพที่ 3.2 Correlation Matrix ของคุณลักษณะ Goals_For

ตารางที่ 3.15 การเลือกคุณลักษณะด้วยค่าสหสัมพันธ์ของตัวแปรตาม Match_Result

ตัวแปรตาม (Target)	เงื่อนไข (Condition)	คุณลักษณะ (Feature)
Match_Result	มีค่า Corr > 0.2 ไม่มีปัญหาเรื่อง Multicollinearity	'Goals_For_roll3', 'GCA_Pass_Live_roll3', 'Goals_Against_roll5', 'Assists_roll5', 'Shots_On_Target_roll3', 'GK_Shots_On_Target_Against_roll5', 'xA_roll5'

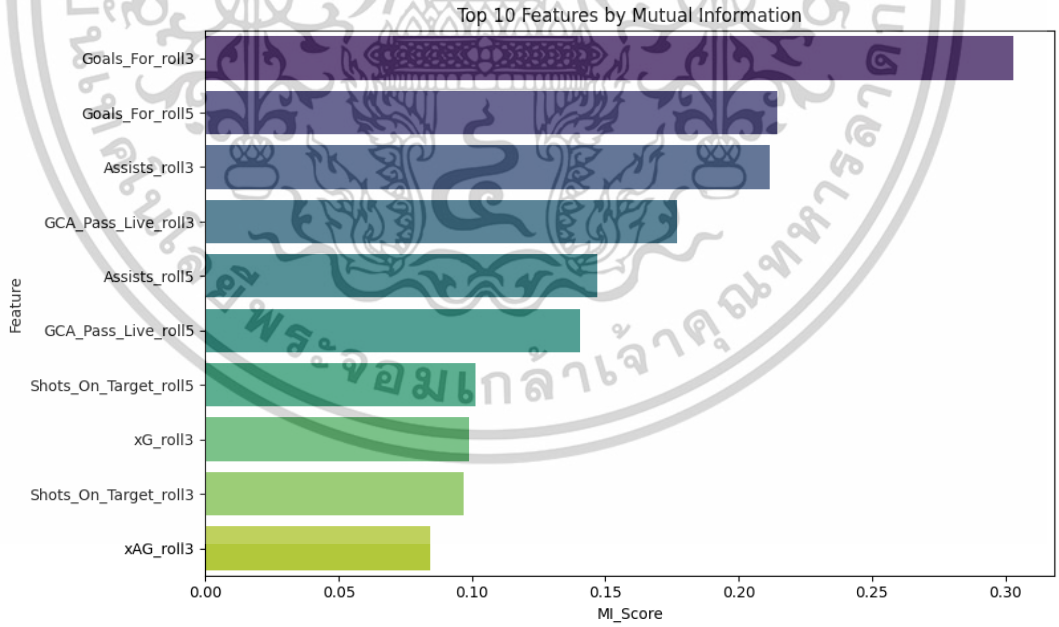


ภาพที่ 3.3 Correlation Matrix ของคุณลักษณะ Match_Result

2) เลือกจากการใช้ข้อมูลร่วมกัน (Mutual Information) ใช้การวัดปริมาณสารสนเทศที่คุณลักษณะหนึ่งมีต่อผลลัพธ์ โดยสามารถตรวจจับความสัมพันธ์ที่ไม่เป็นเชิงเส้นได้ เหมาะสำหรับข้อมูลที่มีความซับซ้อนสูง

ตารางที่ 3.16 การเลือกคุณลักษณะด้วยข้อมูลร่วมโดยมี Goals_For เป็นตัวแปรตาม

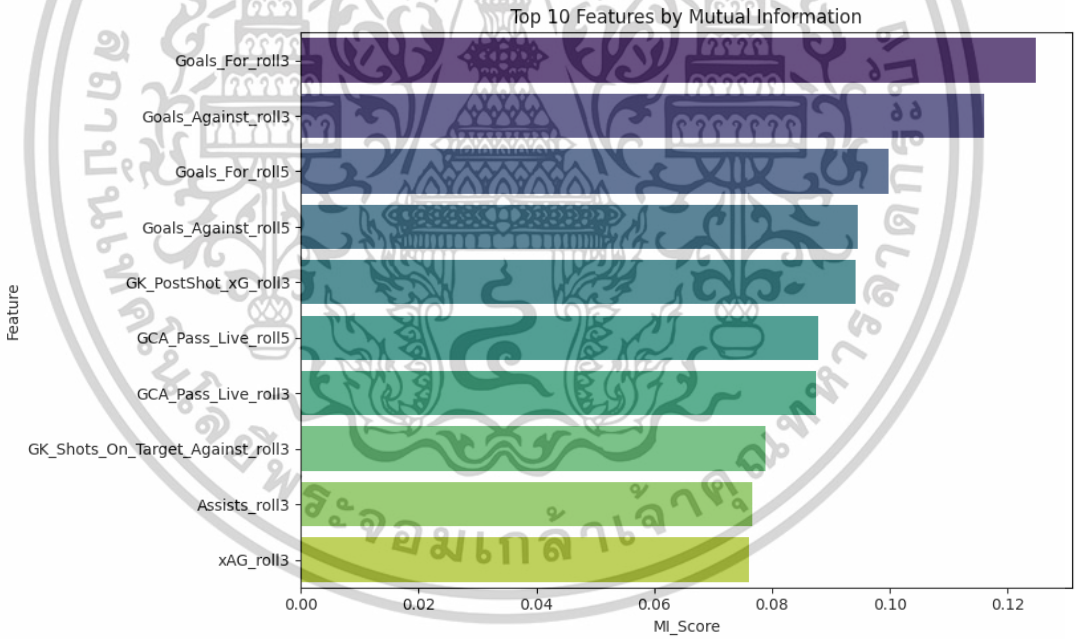
ตัวแปรตาม (Target)	เงื่อนไข (Condition)	คุณลักษณะ (Feature)
Goals_For	Top 10 MI-Score	'Goals_For_roll3', 'Goals_For_roll5', 'Assists_roll3', 'GCA_Pass_Live_roll3', 'Assists_roll5', 'GCA_Pass_Live_roll5', , 'Shots_On_Target_roll5; , 'xG_roll3', 'Shots_On_Target_roll3; , 'xAG_roll3',



ภาพที่ 3.4 คุณลักษณะที่มีค่าข้อมูลร่วมสูงสุด 10 อันดับแรก กับ Goals_For

ตารางที่ 3.17 การเลือกคุณลักษณะด้วยข้อมูลร่วมโดยมี Match_Result เป็นตัวแปรตาม

ตัวแปรตาม (Target)	เงื่อนไข (Condition)	คุณลักษณะ (Feature)
Match_Result	Top 10 MI-Score	'Goals_For_roll3', 'Goals_Against_roll3', 'Goals_For_roll5', 'Goals_Against_roll5', 'GK_PostShot_xG_roll3, 'GCA_Pass_Live_roll5', 'GCA_Pass_Live_roll3', 'GK_Shots_On_Targe_Against_roll3; , 'Assists_roll3', 'xAG_roll3',

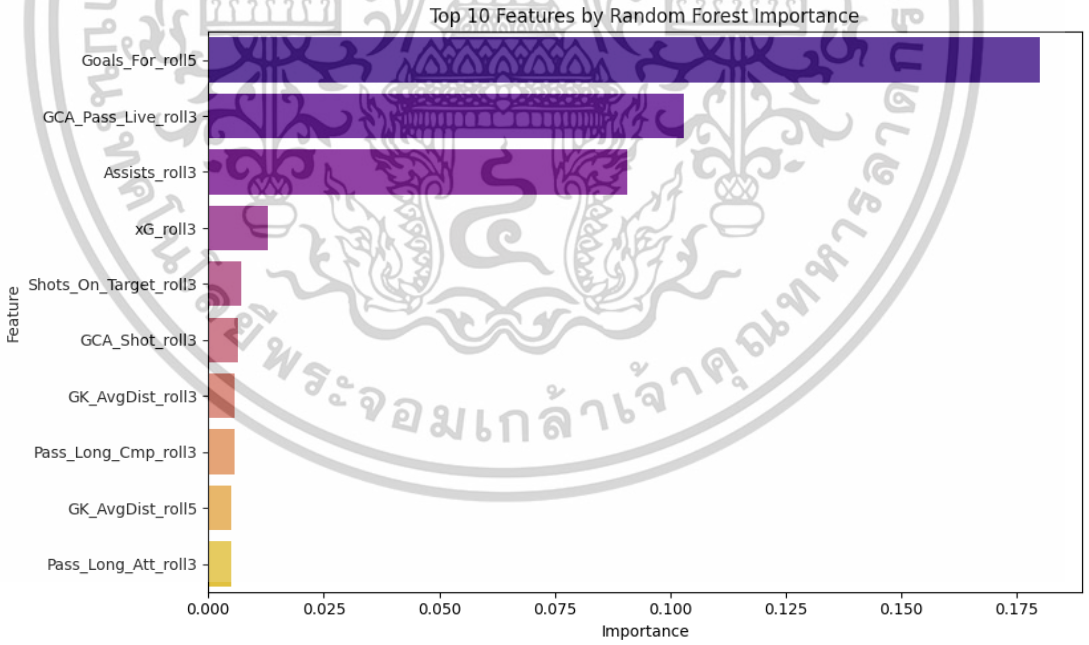


ภาพที่ 3.5 คุณลักษณะที่มีค่าข้อมูลร่วมสูงสุด 10 อันดับแรก กับ Match_Result

3) เลือกจากความสำคัญของคุณลักษณะ (Feature Importance) เป็นวิธีที่อิงกับแบบจำลอง โดยการใช้ความสำคัญของตัวแปรจากโมเดล Tree-based อย่าง Random Forest เพื่อตัดสินว่าตัวแปรใดมีอิทธิพลมากที่สุดต่อการทำนายผล

ตารางที่ 3.18 การเลือกคุณลักษณะด้วยค่าความสำคัญโดยมี Goals_For เป็นตัวแปรตาม

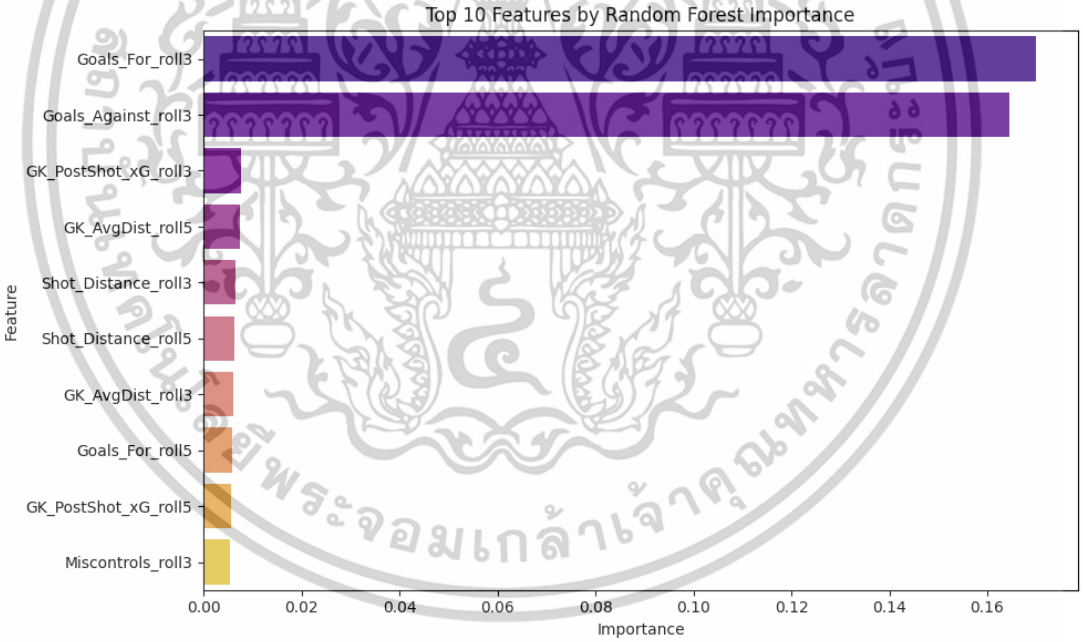
ตัวแปรตาม (Target)	เงื่อนไข (Condition)	คุณลักษณะ (Feature)
Goals_For	Top 10 Importance Score	'Goals_For_roll5', 'GCA_Pass_Live_roll3', 'Assists_roll3', 'xG_roll3', 'Shots_On_Target_roll3', GCA_Shot_roll3, 'GK_AvgDist_roll3, 'Pass_Long_Cmp_roll3, 'GK_AvgDist_roll5, 'Pass_Long_Att_roll3



ภาพที่ 3.6 คุณลักษณะที่มีค่าความสำคัญสูงสุด 10 อันดับแรก สำหรับ Goals_For

ตารางที่ 3.19 การเลือกคุณลักษณะด้วยค่าความสำคัญโดยมี Goals_For เป็นตัวแปรตาม

ตัวแปรตาม (Target)	เงื่อนไข (Condition)	คุณลักษณะ (Feature)
Match_Result	Top 10 Importance Score	'Goals_For_roll3', 'Goals_Against_roll3', 'GK_PostShot_xG_roll3', GK_AvgDist_roll5', 'Shot_Distance_roll3', Shot_Distance_roll5', GK_AvgDist_roll3', 'Goals_For_roll5', GK_Post_Shot_xG_roll5', Miscontrols_roll3'



ภาพที่ 3.7 คุณลักษณะที่มีค่าความสำคัญสูงสุด 10 อันดับแรก สำหรับ Match_Result

3.4.2.5 การแบ่งข้อมูลชุดฝึก กับชุดทดสอบ (Train-Test Split)

การแบ่งข้อมูลชุดฝึก (Training Set) กับชุดทดสอบ (Testing Set) เป็นขั้นตอนพื้นฐานในการสร้าง และประเมินแบบจำลอง โดยชุดฝึกจะถูกใช้ในการฝึกฝนแบบจำลองให้เรียนรู้ความสัมพันธ์ในข้อมูล และชุดทดสอบจะถูกใช้เพื่อประเมินประสิทธิภาพของแบบจำลองกับข้อมูลที่ไม่เคยเห็นมาก่อน โดยแบ่งข้อมูล

โดยใช้ข้อมูลฤดูกาลที่ 2023-2024 เป็นชุดทดสอบ และฤดูกาล 2016-2023 ที่เหลือเป็นชุดฝึกฝน

ตารางที่ 3.20 การแบ่งชุดข้อมูลฝึกและชุดทดสอบ (Train-Test Split)

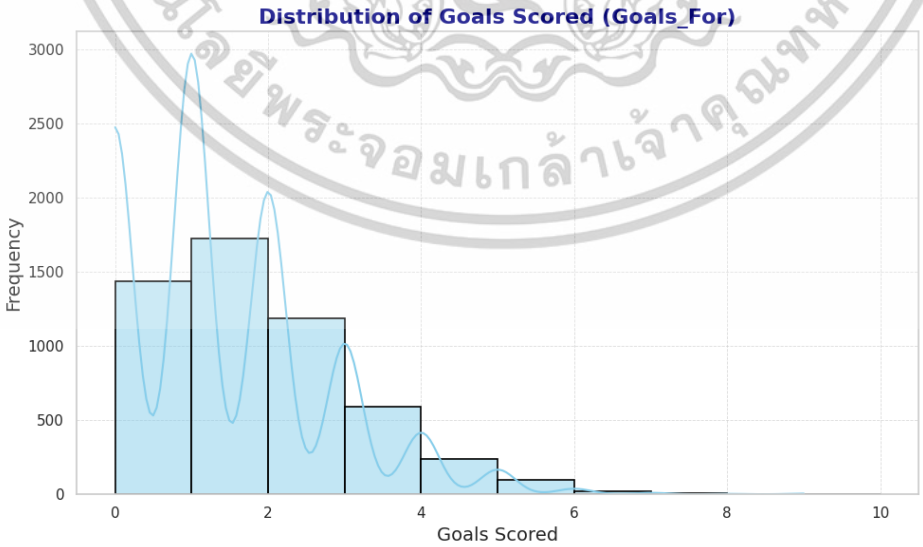
Train data	Test data	Train/Test data
4560	760	70%-30%

3.4.3 การวิเคราะห์ข้อมูลเชิงสำรวจ (Exploratory Data Analysis: EDA)

การวิเคราะห์ข้อมูลเชิงสำรวจ (EDA) เป็นขั้นตอนสำคัญเพื่อทำความเข้าใจโครงสร้างข้อมูล ลักษณะการกระจายตัวของข้อมูล และความสัมพันธ์ระหว่างตัวแปรต่าง ๆ ซึ่งมีผลต่อการเลือกแบบจำลองที่เหมาะสม โดยประกอบด้วยกระบวนการดังนี้

3.4.3.1 การวิเคราะห์การกระจายตัวของข้อมูล

การวิเคราะห์การกระจายตัวของข้อมูล (Data Distribution Analysis) ช่วยให้เข้าใจลักษณะการกระจายของค่าในแต่ละตัวแปร สำหรับตัวแปรเชิงปริมาณ สามารถใช้กราฟฮิสโทแกรม (Histogram), กราฟกล่อง (Box Plot) เพื่อดูรูปร่างของการกระจาย ค่าเฉลี่ย ค่ามัธยฐาน และการมีอยู่ของ Outlier สำหรับตัวแปรเชิงคุณภาพ สามารถใช้กราฟแท่ง (Bar Chart) หรือ กราฟพาย (Pie Chart) เพื่อดูความถี่ของแต่ละหมวดหมู่ การทำความเข้าใจการกระจายตัวของข้อมูลมีผลต่อการเลือกวิธีการจัดการข้อมูลและการเลือกแบบจำลอง

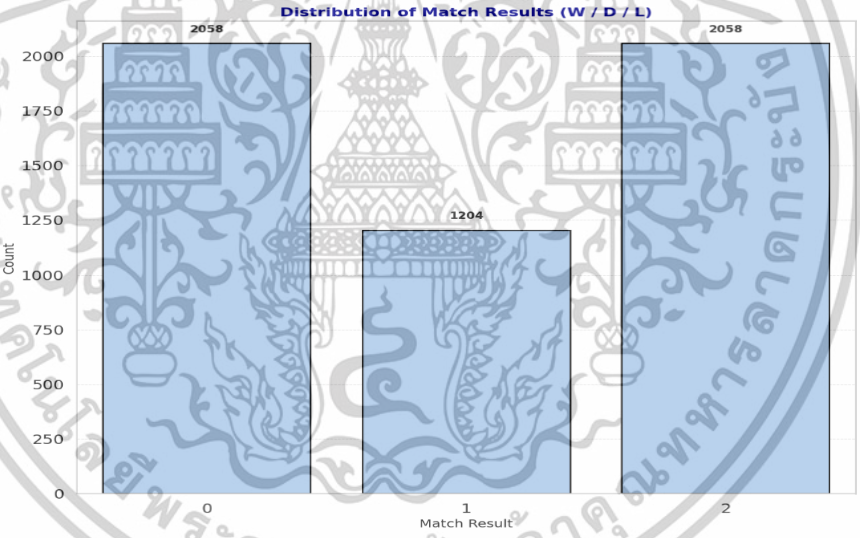


ภาพที่ 3.8 การกระจายข้อมูลของคุณลักษณะ Goals_For

จากภาพที่ 3.8 กราฟฮีสโทแกรมของตัวแปรตาม Goals_For จะเห็นได้ว่ากราฟเกิดการการเบ้ขวา (Right-Skewed) ข้อมูลส่วนใหญ่จะอยู่ในช่วงจำนวน 0 ประตู จนถึง 4 ประตู และจะมีเหตุการณ์ที่เกิดขึ้นได้ยาก หรือแทบจะไม่เกิดขึ้นเลยในช่วงจำนวน 5 ประตูขึ้นไป

3.4.3.3 การวิเคราะห์ความสมดุลของกลุ่ม (Class balance)

การวิเคราะห์ความสมดุลของกลุ่ม (Class Balance Analysis) มีความสำคัญอย่างยิ่งในปัญหาการจำแนกประเภท เป็นการตรวจสอบว่าจำนวนตัวอย่างในแต่ละประเภทของตัวแปรตามมีความสมดุลหรือไม่ หากข้อมูลมีประเภทที่ไม่สมดุล (Imbalanced Classes) อาจส่งผลให้แบบจำลองมีประสิทธิภาพไม่ดีในการทำนายประเภทของข้อมูลส่วนน้อย เทคนิคที่ใช้ในการวิเคราะห์ความสมดุลของประเภท เช่น การนับจำนวนตัวอย่างในแต่ละประเภท และการแสดงผลด้วยกราฟแท่ง หากพบปัญหาประเภทไม่สมดุล อาจต้องใช้เทคนิคการปรับสมดุลข้อมูล เช่น Oversampling หรือ Undersampling



ภาพที่ 3.9 การกระจายข้อมูลของคุณลักษณะ Match_Result

จากภาพที่ 3.9 กราฟแท่งที่แสดงการกระจายตัวของตัวแปรตาม Match_Result โดยมีการแทนค่า Match_Result เป็น W=2, D=1 และ L=0 สรุปได้ว่า การกระจายตัวของ Match_Result ในชุดข้อมูลนี้แสดงให้เห็นถึงความถี่ของการเกิดผลลัพธ์แต่ละประเภท โดยผลการแข่งขันที่ทีมเป็นฝ่ายแพ้ (L หรือค่า 0) และทีมเป็นฝ่ายชนะ (W หรือค่า 2) มีความถี่ในการเกิดขึ้นค่อนข้างใกล้เคียงกันและสูงกว่าผลเสมอ (D หรือค่า 1) เล็กน้อย ซึ่งการกระจายตัวลักษณะนี้สะท้อนถึงความน่าจะเป็นโดยทั่วไปของผลการแข่งขันฟุตบอลที่มักจะมีผลแพ้-ชนะมากกว่าผลเสมอ

แม้ว่าความแตกต่างของความถี่ระหว่างผลลัพธ์ทั้งสามอาจไม่มากนักในชุดข้อมูลนี้ แต่การทำความเข้าใจการกระจายตัวนี้มีความสำคัญสำหรับการสร้าง และประเมินแบบจำลองการจำแนกประเภท เพื่อให้แน่ใจว่าแบบจำลองสามารถจัดการกับการกระจายตัวของข้อมูลผลลัพธ์ได้อย่างเหมาะสม

3.5 การสร้าง และแบบประเมินแบบจำลอง (Model building and Evaluation)

หลังจากที่ข้อมูลได้ผ่านการเตรียมและวิเคราะห์เบื้องต้นแล้ว ขั้นตอนต่อไปคือการสร้างแบบจำลองเพื่อใช้ในทำนาย หรือวิเคราะห์ตามวัตถุประสงค์ที่ตั้งไว้ รวมถึงการประเมินประสิทธิภาพของแบบจำลอง กระบวนการสร้าง และประเมินแบบจำลองประกอบด้วยขั้นตอนสำคัญดังต่อไปนี้

3.5.1 การเลือกแบบจำลอง (Model Selection)

ถือเป็นขั้นตอนที่สำคัญในการวิจัยด้านการเรียนรู้ของเครื่อง โดยมีเป้าหมายเพื่อคัดเลือกแบบจำลองที่เหมาะสมที่สุดกับลักษณะของปัญหา และข้อมูลที่มีอยู่ เพื่อให้สามารถสร้างการทำนาย หรือจำแนกประเภทได้อย่างแม่นยำ และมีประสิทธิภาพ เกณฑ์ที่ใช้ในการเลือกแบบจำลองพิจารณาจากหลายปัจจัย ได้แก่ ประเภทของปัญหา (เช่น ปัญหาการถดถอย หรือการจำแนกประเภท), ลักษณะของข้อมูล (เช่น ขนาดข้อมูล, ประเภทของตัวแปร, ความสัมพันธ์ระหว่างตัวแปร), ผลลัพธ์จากการวิเคราะห์ข้อมูลเบื้องต้น (Exploratory Data Analysis), รวมถึงความสามารถในการตีความผลลัพธ์ของแบบจำลอง และประสิทธิภาพที่คาดหวังได้จากแต่ละแบบจำลอง โดยในการวิจัยครั้งนี้ ได้ดำเนินการเลือกใช้แบบจำลองที่หลากหลายทั้งแบบเป็นเชิงเส้น และไม่เป็นเชิงเส้นในการทำนาย โดยแบ่งตามลักษณะของข้อมูลและเป้าหมายของการทำนาย ดังนี้:

3.5.1.1 ตัวแปรตามที่ใช้ทำนายผลประตู Goals_For

เนื่องจากตัวแปรตาม Goals_For เป็นข้อมูลประเภทจำนวนนับ จึงไม่เหมาะสมกับการทำการถดถอยเชิงเส้น เพราะว่าการแจกแจงแบบไม่ปกติ และความแปรปรวนไม่คงที่ ดังนั้นตัวแปรตาม Goals_For จึงไม่เหมาะสมในการทำนายด้วยการถดถอยเชิงเส้น จึงต้องใช้แบบจำลองปัวซองที่เหมาะสมกับจำนวนนับ

3.5.1.2 ตัวแปรตามที่ใช้ทำนายผลการแข่งขัน Match_Result

สำหรับการพยากรณ์ผลการแข่งขัน (เช่น ชนะ, เสมอ, แพ้) ซึ่งเป็นข้อมูลเชิงหมวดหมู่ จะใช้แบบจำลองจำแนกประเภทแบบไม่เชิงเส้นเป็นหลัก ได้แก่ แบบจำลอง Decision Tree, Random Forest, XGBoost และ LightGBM

3.5.2 การฝึกแบบจำลอง (Model Training)

เป็นขั้นตอนสำคัญในงานวิจัยด้านการเรียนรู้ของเครื่องจักร โดยในขั้นตอนนี้แบบจำลองจะเรียนรู้จากชุดข้อมูลฝึก (Training Data) ที่ประกอบด้วยคุณลักษณะ และค่าตัวแปรตามที่เลือกจากเทคนิคการเลือกคุณลักษณะต่างๆ โดยเป้าหมายคือการให้แบบจำลองค้นหารูปแบบความสัมพันธ์ที่ระหว่างคุณลักษณะต่างๆ กับผลลัพธ์ของตัวแปรตามอย่างเป็นระบบ กระบวนการนี้ทำเพื่อให้แบบจำลองมีความสามารถในการทำความเข้าใจข้อมูลฝึกได้ดีที่สุด และเมื่อการฝึกเสร็จสิ้น จะได้แบบจำลองที่พร้อมนำไปใช้ทำนายผลลัพธ์ของข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนได้อย่างแม่นยำตามรูปแบบที่ได้เรียนรู้มา

3.5.3 การประเมินแบบจำลอง (Model Evaluation)

ขั้นตอนสำคัญที่ใช้วัดความสามารถของแบบจำลองที่ผ่านการฝึกฝนแล้วว่ามีประสิทธิภาพเพียงใดในการทำนายข้อมูลใหม่ โดยใช้ชุดข้อมูลทดสอบ (Test Set) ที่ไม่เคยถูกใช้ในการฝึก เพื่อให้การประเมินมีความเที่ยงตรง การเลือกตัวชี้วัด (Evaluation Metrics) จะพิจารณาตามประเภทของปัญหาดังนี้:

3.5.3.1 การถดถอยแบบปัวซอง (Poisson Regression Problems)

การใช้ตัวชี้วัดอย่าง AIC, BIC และ Log-likelihood เปรียบเทียบกันในแต่ละการเลือกคุณลักษณะ และเปรียบเทียบกันระหว่างแบบจำลองปัวซองแบบปกติ, แบบพหุคูณเชิงลบ และแบบศูนย์พอง ว่าแบบจำลองไหนเหมาะสมกับชุดข้อมูลมากที่สุด

3.5.3.2 การจำแนกประเภท (Classification Problems)

การใช้ตัวชี้วัดอย่าง Accuracy, Precision, Recall และ F1-Score จาก Confusion Matrix ที่สร้างขึ้น เป็นสิ่งที่จำเป็นอย่างยิ่งในการประเมินความแม่นยำและประสิทธิภาพของแบบจำลองในการจัดกลุ่มข้อมูลการประเมินนี้มีเป้าหมายเพื่อตรวจสอบความสามารถในการ Generalize ของแบบจำลอง ว่าสามารถนำไปใช้กับข้อมูลที่ไม่เคยเห็นมาก่อนได้ดีเพียงใด ซึ่งเป็นตัวชี้วัดสำคัญของประสิทธิภาพในสถานการณ์จริง

3.5.4 การเพิ่มประสิทธิภาพให้แบบจำลอง (Model Improvement)

การเพิ่มประสิทธิภาพให้แบบจำลองเป็นกระบวนการปรับปรุงประสิทธิภาพของแบบจำลองให้ดียิ่งขึ้น ซึ่งการวิจัยจะใช้วิธีดังต่อไปนี้

3.5.4.1 การปรับจูนไฮเปอร์พารามิเตอร์ (Hyperparameter Tuning)

จากการที่เลือกคุณลักษณะของตามวิธีการต่างๆ และการประเมินประสิทธิภาพของแบบจำลอง จะนั้นจะวิเคราะห์หาแบบจำลองที่คิดว่าดีที่สุด และเหมาะสมที่สุด มาปรับจูนไฮเปอร์พารามิเตอร์ โดยเพื่อเพิ่มความประสิทธิภาพให้กับแบบจำลอง โดยการใช้เทคนิคการปรับจูนไฮเปอร์พารามิเตอร์ ดังต่อไปนี้

1). Grid Search: ทำการค้นหาไฮเปอร์พารามิเตอร์ที่ทำให้แบบจำลองที่ประสิทธิภาพสูงสุด อยู่ในตัวแปรทั้งหมดที่ได้กำหนดไว้ทั้งหมด

2) Random Search: ทำการค้นหาไฮเปอร์พารามิเตอร์ที่ทำให้แบบจำลองที่ประสิทธิภาพสูงสุด อยู่ในตัวแปรทั้งหมดที่ได้กำหนดแบบสุ่ม ตามจำนวนรอบที่ได้ตั้งไว้

3.5.4.2 การปรับปรุงข้อมูล

จากเดิมข้อมูลในการใช้ทำนายของการแจกข้อมูล จะใช้การวิเคราะห์ทั้งหมด 2 บรรทัด โดยแบ่งเป็นบรรทัดของทีมเจ้าบ้าน และของทีมเยือน ซึ่งเมื่อแบบจำลองทำการจำแนก จะเกิดความผิดพลาดขึ้น สมมติว่าเจ้าบ้านทำนายออกมาว่าเป็นทีมชนะ (W) ตามความเป็นจริง ทีมเยือนต้องออกมาเป็นแพ้ (L) แต่เมื่อทำนายด้วยแบบจำลองทีมเยือนสามารถจำแนกออกได้เป็นทั้ง 3 ผลลัพธ์ ไม่ว่าจะเป็น ชนะ แพ้ และเสมอ ขึ้นอยู่กับค่าของข้อมูล เพื่อลดความสับสน และผิดพลาดจึงทำให้รวบรวมข้อมูลสำหรับการจำแนกให้เป็น 1 บรรทัด แล้วใช้ทำนายผล โดยอิงจากทีมเจ้าบ้าน โดยถ้าออกมาเป็น ชนะ นั่นหมายถึงว่าทีมเจ้าบ้านชนะ ทีมเยือน

บทที่ 4
ผลการวิจัย และการอภิปรายผล

ในบทนี้จะกล่าวถึงผลการดำเนินงานจากการสร้างแบบจำลองการเรียนรู้ของเครื่องจักรในการทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีก โดยแบ่งเป็นเป็นทั้งการถดถอย และการจัดประเภท และทำการอภิปราย และเปรียบเทียบผลลัพธ์ได้จากการทำแบบจำลองต่างๆ ที่ได้ตั้งไว้ในบทที่ 3

4.1 ผลลัพธ์ของการทำนายผลจำนวนประตู (Goals_For)

ผลลัพธ์จากการเลือกคุณลักษณะตามเทคนิคๆต่าง และแบบจำลองที่สร้างขึ้นจากแบบจำลองการถดถอยแบบปัวซอง, พหุนามเชิงลบ และศูนย์พอง โดยวัดผลด้วย AIC, BIC และ Log-Likelihood โดยผลการวิจัยเป็นดังตารางที่ 4.1 และ ตารางที่ 4.2

โดยค่าเฉลี่ยของ Goals_For อยู่ที่ 1.3822 และความแปรปวนอยู่ที่ 1.6357 ซึ่งข้อมูลชุดนี้มี Overdispersion น้อยทำให้แบบจำลองปัวซองแบบทวินามเชิงลบไม่เหมาะสมกับชุดข้อมูลนี้ และชุดข้อมูลมมีข้อมูลจำนวนนับที่มีค่า 0 อยู่ที่ 28.1% ซึ่งถือว่าแบบจำลองปัวซองแบบศูนย์พองก็ไม่เหมาะสม ในที่นี้แบบจำลองแบบปัวซองน่าจะเหมาะสมที่สุดสำหรับชุดข้อมูลนี้

ตารางที่ 4.1 ประสิทธิภาพของแบบจำลองปัวซองโดยมี Goals_For เป็นตัวแปรตาม

การเลือกคุณลักษณะ	แบบจำลอง	Log-Likelihood	AIC	BIC
สหสัมพันธ์ (Correlation)	แบบจำลองปัวซอง	-1080.485	12115.427	12173.253
	แบบจำลองทวินามเชิงลบ (alpha) = 1	-1269.963	12117.435	12181.686
	แบบจำลองศูนย์พอง	-1080.484	12117.435	12181.686

ตารางที่ 4.1(ต่อ) ประสิทธิภาพของแบบจำลองปัวซองโดยมี Goals_For เป็นตัวแปรตาม

การเลือกคุณลักษณะ	แบบจำลอง	Log-Likelihood	AIC	BIC
ข้อมูลร่วมกัน (Mutual Information)	แบบจำลองปัวซอง	-1080.429	12120.357	12191.033
	แบบจำลองทวินามเชิงลบ (alpha) = 1	-1270.048	12122.363	12199.464
	แบบจำลองศูนย์พอง	-1080.556	1211.411	12199.542
ความสำคัญของคุณลักษณะ (Feature Importance)	แบบจำลองปัวซอง	-1080.485	12115.427	12173.253
	แบบจำลองทวินามเชิงลบ (alpha) = 1	-1269.869	12117.435	12181.686
	แบบจำลองศูนย์พอง	-1080.484	12117.435	12181.686

จากตารางที่ 4.1 การเปรียบเทียบประสิทธิภาพของแบบจำลองปัวซอง แบบจำลองทวินามเชิงลบ และแบบจำลองศูนย์พอง ร่วมกับเทคนิคการเลือกคุณลักษณะสามวิธี ได้แก่ สหสัมพันธ์ ข้อมูลร่วมกัน และความสำคัญของคุณลักษณะ พบว่าแบบจำลองปัวซองแสดงประสิทธิภาพที่เหนือกว่าอย่างชัดเจนในทุกเทคนิค โดยมีค่า AIC และ BIC ต่ำที่สุด (เนื่องมาจากชุดข้อมูลไม่มี Overdispersion และไม่มีจำนวนค่าศูนย์) โดยเฉพาะเมื่อใช้ร่วมกับเทคนิคสหสัมพันธ์ และความสำคัญของคุณลักษณะ (AIC: 12115.427, BIC: 12173.253) ในขณะที่แบบจำลองทวินามเชิงลบให้ประสิทธิภาพที่แย่ที่สุด ด้วยค่า Log-Likelihood ที่ต่ำมาก (-1269.869 ถึง -1270.048) สำหรับเทคนิคการเลือกคุณลักษณะพบว่าเทคนิคสหสัมพันธ์ และความสำคัญของคุณลักษณะให้ผลลัพธ์ที่ดี และใกล้เคียงกัน ส่วนเทคนิคข้อมูลร่วมกันมีประสิทธิภาพรองลงมาเล็กน้อย ผลจากการเปรียบเทียบแบบจำลองปัวซองเป็นแบบจำลองที่เหมาะสมที่สุดสำหรับการทำนายจำนวนประตูในกีฬาฟุตบอล และสามารถนำไปประยุกต์ใช้ในการวิเคราะห์และทำนายผลการแข่งขันได้อย่างมีประสิทธิภาพ

4.1.1 การประยุกต์ใช้แบบจำลองปัวซองในการทำนายจำนวนประตู

จากการสร้างแบบจำลองปัวซอง จะได้ค่าพารามิเตอร์แลมดา (λ) ซึ่งแทนค่าเฉลี่ยของการเกิดเหตุการณ์ ในกรณีนี้คือค่าเฉลี่ยของจำนวนประตูที่ทีมสามารถทำได้ต่อการแข่งขัน 1 นัด เมื่อได้ค่าเฉลี่ยของการเกิดเหตุการณ์แล้ว สามารถนำไปคำนวณความน่าจะเป็นของการทำประตูในจำนวนต่างๆ ได้โดยใช้ฟังก์ชันมวลความน่าจะเป็น (Probability Mass Function) ของการแจกแจงปัวซอง

Possion Distribution									
k	λ = 0.1	λ = 1	λ = 2	λ = 3	λ = 4	λ = 5	λ = 6	λ = 7	λ = 8
0	0.9048	0.3679	0.1353	0.0498	0.0183	0.0067	0.0025	0	0
1	0.0905	0.3679	0.2707	0.2241	0.1839	0.1747	0.1201	0.0995	0.0952
2	0.0045	0.1839	0.2707	0.3362	0.3679	0.4368	0.3602	0.3495	0.3808
3	0.00015	0.0613	0.1805	0.3362	0.4905	0.7279	0.7204	0.814	1.0141
4	0	0.0153	0.0903	0.2522	0.4905	0.9099	1.0806	1.4242	2.0249
5	0	0.0031	0.0361	0.1513	0.3924	0.9099	1.2967	1.9883	3.2387
6	0	0.0005	0.12	0.0756	0.2629	0.7575	1.2967	2.3187	4.3132

ภาพที่ 4.1 การแสดงค่าความน่าจะเป็นของเหตุการณ์

ภาพที่ 4.1 แสดงการเปรียบเทียบความน่าจะเป็นของการทำประตูในจำนวน 0-10 ประตู สำหรับค่าแลมดาที่แตกต่างกัน เพื่อให้เห็นภาพรวมของการกระจายตัวของความน่าจะเป็นตามค่าเฉลี่ยที่เปลี่ยนแปลงไปสมมติว่าทีม A มีค่าเฉลี่ยการทำประตู (λ) เท่ากับ 1.99 ประตูจากแบบจำลองปัวซองจะสามารถคำนวณความน่าจะเป็นได้ดังนี้

- ความน่าจะเป็นที่จะทำได้ 0 ประตู ≈ 0.13 หรือ 13%
- ความน่าจะเป็นที่จะทำได้ 1 ประตู > 0.27 หรือ > 27%
- ความน่าจะเป็นที่จะทำได้ 2 ประตู < 0.27 หรือ < 27%
- ความน่าจะเป็นที่จะทำได้ 3 ประตูขึ้นไป ≈ 0.18 หรือ 18%

จากผลการคำนวณจะเห็นว่า ทีม A มีโอกาสทำประตูได้ 1 ประตูและ 2 ประตูใกล้ๆกัน (27% แต่ละกรณี) ซึ่งเป็นความน่าจะเป็นสูงสุด ในขณะที่โอกาสที่จะไม่ทำประตูเลยมีเพียง 13% เท่านั้น

4.2 ผลลัพธ์ของการทำนายผลการแข่งขัน (Match_Result)

ผลลัพธ์จากการเลือกคุณลักษณะตามเทคนิคๆต่าง และแบบจำลองที่สร้างขึ้นจากแบบจำลองการจำแนกประเภทในการทำนายผลการแข่งขัน Match_Result โดยวัดผลด้วย Accuracy, Precision, Recall และ F1-Score โดยผลการวิจัยเป็นดังตารางที่ 4.2, ตารางที่ 4.3 และตารางที่ 4.4

ตารางที่ 4.2 ประสิทธิภาพของแบบจำลองการจำแนกโดยมี Match_Result เป็นตัวแปรตาม (เลือกคุณลักษณะด้วยสหสัมพันธ์)

การเลือกคุณลักษณะ	แบบจำลอง	เทคนิค	Accuracy	Precision	Recall	F1-Score	ROC
สหสัมพันธ์ (Correlation)	ต้นไม้ตัดสินใจ (Decision Tree)	None	0.511	0.483	0.511	0.495	0.636
		Under	0.475	0.521	0.475	0.490	0.630
		Over	0.494	0.524	0.494	0.506	0.621
		SMOTE	0.525	0.523	0.525	0.523	0.647
	ป่าแบบสุ่ม (Random Forest)	None	0.571	0.511	0.571	0.529	0.696
		Under	0.526	0.529	0.526	0.526	0.697
		Over	0.575	0.531	0.575	0.544	0.694
		SMOTE	0.569	0.527	0.569	0.542	0.696
	XGBoost	None	0.561	.514	0.561	0.528	0.698
		Under	0.515	0.536	0.515	0.524	0.689
		Over	0.555	0.544	0.555	0.549	0.702
		SMOTE	0.560	0.521	0.560	0.531	0.698
	LightGBM	None	0.582	0.525	0.582	0.537	0.705
		Under	0.538	0.541	0.538	0.539	0.702
		Over	0.555	0.542	0.555	0.548	0.703
		SMOTE	0.592	0.556	0.592	0.561	0.704

ตารางที่ 4.2 การใช้เทคนิคสหสัมพันธ์ในการเลือกคุณลักษณะแสดงให้เห็นถึงข้อจำกัดที่สำคัญในการปรับปรุงประสิทธิภาพของแบบจำลอง โดยค่า Accuracy เฉลี่ยอยู่ในช่วง 0.475-0.592 แบบจำลอง LightGBM ร่วมกับ SMOTE ให้ผลลัพธ์ดีที่สุดในกลุ่มนี้ด้วยค่า Accuracy 0.592, Precision 0.556, Recall 0.592, F1-Score 0.561 และ AUC-ROC 0.704 รองลงมาคือ LightGBM (None) ด้วยค่า Accuracy 0.582 และ Random Forest (Oversampling) ด้วยค่า Accuracy 0.575 ในขณะที่แบบจำลองต้นไม้ตัดสินใจแสดงประสิทธิภาพที่ประสิทธิที่ภาพน้อยที่สุด โดยเฉพาะเมื่อใช้ Undersampling (Accuracy 0.475) ซึ่งบ่งชี้ว่าเทคนิคสหสัมพันธ์อาจไม่เหมาะสมสำหรับการจับคุณลักษณะที่ซับซ้อนในชุดข้อมูลนี้ และยังแสดงให้เห็นว่าแบบจำลองประเภทการเรียนรู้ร่วมกัน Ensemble methods (LightGBM, Random Forest, XGBoost) มีประสิทธิภาพที่สูงกว่า

แบบจำลองเดี่ยวอย่างชัดเจน โดย LightGBM ให้ค่า AUC-ROC สูงสุดในทุกเทคนิค (0.702-0.705) ตามด้วย XGBoost (0.689-0.702) และ Random Forest (0.694-0.697) ส่วนเทคนิคการจัดการข้อมูลไม่สมดุลพบว่า SMOTE และ Oversampling มีประสิทธิผลในการปรับปรุงประสิทธิภาพ ในขณะที่ Undersampling มักส่งผลให้ประสิทธิภาพลดลงในแบบจำลองส่วนใหญ่

ตารางที่ 4.3 ประสิทธิภาพของแบบจำลองการจำแนกโดยมี Match_Result เป็นตัวแปรตาม (เลือกคุณลักษณะด้วยข้อมูลร่วมกัน)

การเลือกคุณลักษณะ	แบบจำลอง	เทคนิค	Accuracy	Precision	Recall	F1-Score	ROC
ข้อมูลร่วมกัน (Mutual Information)	ต้นไม้ตัดสินใจ (Decision Tree)	None	0.546	0.519	0.546	0.526	0.652
		Under	0.506	0.552	0.506	0.522	0.656
		Over	0.531	0.531	0.531	0.530	0.651
		SMOTE	0.567	0.547	0.567	0.551	0.683
	ป่าแบบสุ่ม (Random Forest)	None	0.600	0.547	0.600	0.555	0.717
		Under	0.565	0.566	0.565	0.566	0.720
		Over	0.592	0.554	0.592	0.562	0.722
		SMOTE	0.590	0.545	0.590	0.555	0.722
	XGBoost	None	0.572	0.528	0.572	0.539	0.712
		Under	0.527	0.540	0.527	0.533	0.706
		Over	0.564	0.542	0.564	0.550	0.713
		SMOTE	0.573	0.540	0.573	0.548	0.719
	LightGBM	None	0.590	0.534	0.590	0.548	0.723
		Under	0.556	0.558	0.556	0.557	0.721
		Over	0.564	0.512	0.564	0.550	0.722
		SMOTE	0.578	0.534	0.578	0.545	0.716

ตารางที่ 4.3 การใช้เทคนิคข้อมูลร่วมกัน (Mutual Information) ในการเลือกคุณลักษณะแสดงให้เห็นการปรับปรุงประสิทธิภาพของแบบจำลองที่ดีขึ้นเมื่อเปรียบเทียบกับเทคนิคสหสัมพันธ์ โดยค่า Accuracy เฉลี่ยอยู่ในช่วง 0.506-0.600 แบบจำลอง Random Forest (None) ให้ผลลัพธ์ดีที่สุดในกลุ่มนี้ด้วยค่า Accuracy 0.600, Precision 0.547, Recall 0.600, F1-Score 0.555 และ AUC-ROC 0.717 รองลงมาคือ Random Forest (Oversampling) ด้วยค่า Accuracy 0.592 และ

LightGBM (None) ด้วยค่า Accuracy 0.590 ในขณะที่แบบจำลองต้นไม้ตัดสินใจยังคงแสดงประสิทธิภาพที่ต่ำที่สุด โดยเฉพาะเมื่อใช้ Undersampling (Accuracy 0.506) ซึ่งบ่งชี้ว่าเทคนิคข้อมูลร่วมกันสามารถจับคุณลักษณะที่สำคัญได้ดีกว่าเทคนิคสหสัมพันธ์ ผลการทดลองยังแสดงให้เห็นว่าแบบจำลองประเภทการเรียนรู้ร่วมกันยังคงมีประสิทธิภาพที่สูงกว่าแบบจำลองเดี่ยวอย่างชัดเจน โดย Random Forest ให้ค่า AUC-ROC สูงสุดในหลายเทคนิค (0.717-0.722) ตามด้วย LightGBM (0.716-0.723) และ XGBoost (0.706-0.719) ส่วนเทคนิคการจัดการข้อมูลไม่สมดุลพบว่าผลลัพธ์มีความแตกต่างกันไปตามแบบจำลอง โดยการใช้ข้อมูลดั้งเดิม (None) กับ Random Forest และ LightGBM ให้ผลดี ในขณะที่ SMOTE ช่วยปรับปรุง Decision Tree อย่างมีนัยสำคัญ และ Undersampling ยังคงส่งผลให้ประสิทธิภาพลดลงในแบบจำลองส่วนใหญ่

ตารางที่ 4.4 ประสิทธิภาพของแบบจำลองการจำแนกโดยมี Match_Result เป็นตัวแปรตาม (เลือกคุณลักษณะด้วยความสำคัญของคุณลักษณะ)

การเลือกคุณลักษณะ	แบบจำลอง	เทคนิค	Accuracy	Precision	Recall	F1-Score	ROC
ความสำคัญของคุณลักษณะ (Feature Importance)	ต้นไม้ตัดสินใจ (Decision Tree)	None	0.511	0.489	0.511	0.498	0.624
		Under	0.496	0.521	0.496	0.505	0.605
		Over	0.505	0.519	0.505	0.511	0.643
		SMOTE	0.553	0.544	0.553	0.546	0.682
	ป่าแบบสุ่ม (Random Forest)	None	0.596	0.537	0.596	0.543	0.715
		Under	0.572	0.573	0.572	0.572	0.729
		Over	0.593	0.540	0.593	0.552	0.726
		SMOTE	0.597	0.561	0.597	0.532	0.720
	XGBoost	None	0.578	0.535	0.578	0.547	0.707
		Under	0.531	0.551	0.531	0.540	0.705
		Over	0.584	0.553	0.584	0.561	0.716
		SMOTE	0.575	0.535	0.575	0.545	0.709
	LightGBM	None	0.611	0.577	0.611	0.570	0.726
		Under	0.536	0.549	0.536	0.542	0.721
		Over	0.578	0.553	0.578	0.563	0.72
		SMOTE	0.597	0.556	0.597	0.564	0.725

ตารางที่ 4.4 การใช้เทคนิคความสำคัญของคุณลักษณะ (Feature Importance) ในการเลือกคุณลักษณะแสดงให้เห็นการปรับปรุงประสิทธิภาพของแบบจำลองที่ดีที่สุดเมื่อเปรียบเทียบกับเทคนิคอื่นๆ โดยค่า Accuracy เฉลี่ยอยู่ในช่วง 0.496-0.611 แบบจำลอง LightGBM (None) ให้ผลลัพธ์ดีที่สุดในกลุ่มนี้ด้วยค่า Accuracy 0.611, Precision 0.577, Recall 0.611, F1-Score 0.570 และ AUC-ROC 0.726 รองลงมาคือ Random Forest (SMOTE) ด้วยค่า Accuracy 0.597 และ LightGBM (SMOTE) ด้วยค่า Accuracy 0.597 ในขณะที่แบบจำลองต้นไม้ตัดสินใจยังคงแสดงประสิทธิภาพที่ต่ำที่สุด โดยเฉพาะเมื่อใช้ Undersampling (Accuracy 0.496) ซึ่งบ่งชี้ว่าเทคนิคความสำคัญของคุณลักษณะสามารถคัดเลือกคุณลักษณะที่เหมาะสมได้อย่างมีประสิทธิภาพ ผลการทดลองยังแสดงให้เห็นว่าแบบจำลองประเภทการเรียนรู้ร่วมกันยังคงมีประสิทธิภาพที่สูงกว่าแบบจำลองเดี่ยวอย่างชัดเจน โดย LightGBM และ Random Forest มีค่า AUC-ROC ที่สูงใกล้เคียงกัน (0.715-0.729) ตามด้วย XGBoost (0.705-0.716) ส่วนเทคนิคการจัดการข้อมูลไม่สมดุลพบว่าการใช้ข้อมูลดั้งเดิม (None) ให้ผลดีที่สุดใน LightGBM ในขณะที่ SMOTE ช่วยปรับปรุงประสิทธิภาพของ Decision Tree และ Random Forest อย่างมีนัยสำคัญ และ Undersampling ยังคงส่งผลให้ประสิทธิภาพลดลงในแบบจำลองทุกประเภทการวิเคราะห์เปรียบเทียบประสิทธิภาพแบบจำลอง

- แบบจำลองต้นไม้ตัดสินใจ: แสดงประสิทธิภาพที่ต่ำที่สุดอย่างสม่ำเสมอในทุกเงื่อนไข โดยมีค่า Accuracy เฉลี่ยรวม 0.589 ซึ่งต่ำกว่าแบบจำลองอื่นๆ อย่างมีนัยสำคัญ ข้อจำกัดหลักคือความไวต่อการเปลี่ยนแปลงข้อมูล และแนวโน้มในการ overfitting

- แบบจำลองป่าสุ่ม: แสดงการปรับปรุงที่ชัดเจนจากแบบจำลองต้นไม้ตัดสินใจด้วยค่า Accuracy เฉลี่ยรวม 0.661 การใช้การเรียนรู้ร่วมกันช่วยลดปัญหา overfitting และเพิ่มความเสถียรของแบบจำลอง

- XGBoost: ให้ประสิทธิภาพที่สูง และมีความเสถียร ด้วยค่า Accuracy เฉลี่ยรวม 0.664 แสดงความสอดคล้องกันในทุกเมตริกการประเมิน ความเหนือกว่าในด้าน regularization ทำให้มีความทนทานต่อ noise และ outliers

- LightGBM: แสดงประสิทธิภาพสูงสุดด้วยค่า Accuracy เฉลี่ยรวม 0.672 ประสิทธิภาพที่เหนือกว่าเกิดจากอัลกอริทึม gradient boosting ที่ปรับปรุงแล้วและเทคนิค leaf-wise tree growth ที่มีประสิทธิภาพสูง

การวิจัยการทำนายผลลัพธ์ในการแข่งขันฟุตบอล (Match_Result) โดยใช้การเทคนิคต่างๆ ในการเลือกคุณลักษณะไม่ว่าจะเป็น สหสัมพันธ์, ข้อมูลร่วม และความสำคัญของคุณลักษณะ การใช้เทคนิคการจัดการข้อมูลที่ไม่สมดุล และแบบจำลองการจำแนกทั้งต้นไม้ตัดสินใจ, ป่าแบบสุ่ม, XGBoost และ LightGBM เห็นได้ว่า การใช้เทคนิคการเลือกคุณลักษณะแบบความสำคัญของ

คุณลักษณะให้ผลดีกับการวิจัยในครั้งนี้ ทำงานร่วมกับแบบจำลอง LightGBM และไม่ได้จัดการกับเทคนิคที่ไม่สมดุล แสดงประสิทธิภาพสูงสุดในการวิจัยในครั้งนี้ และแบบจำลองประเภท Gradient Boosting (XGBoost และ LightGBM) ยังให้ผลลัพธ์ที่สม่ำเสมอมากกว่าแบบจำลองประเภทอื่นๆ ซึ่งเป็นข้อมูลที่มีลักษณะซับซ้อน และความสัมพันธ์ระหว่างตัวแปรที่ไม่เป็นเชิงเส้น

4.3 การพัฒนาปรับปรุง แบบจำลอง

4.3.1 การปรับจูนไฮเปอร์พารามิเตอร์

จากผลลัพธ์ในการวิจัยในตารางที่ 4.2, ตารางที่ 4.3 และตารางที่ 4.4 พบว่าแบบจำลองในการจำแนกผลการแข่งขันคือแบบจำลองแบบ LightGBM ซึ่งมีค่า Accuracy และ F1-Score สูงที่สุดซึ่งใช้รวมกับเทคนิคการเลือกคุณลักษณะแบบความสำคัญของคุณลักษณะ และไม่ได้ใช้วิธีจัดการข้อมูลแบบ

ตารางที่ 4.5 ค่าไฮเปอร์พารามิเตอร์ที่ใช้ในการปรับแต่งแบบจำลอง LightGBM

ไฮเปอร์พารามิเตอร์	ค่า	คำอธิบาย
num_leaves	[31, 50, 100, 200]	จำนวนใบในแต่ละต้นไม้
max_depth	[-1, 5, 10, 15]	ความลึกสูงสุดแต่ละต้นไม้
learning_rate	[0.01, 0.05, 0.01, 0.02]	อัตราการเรียนรู้
n_estimators	[50, 100, 200]	จำนวนรอบในการฝึก
min_child_samples	[10, 20]	จำนวนข้อมูลขั้นต่ำในแต่ละใบ
subsample	[0.8, 1.0]	สัดส่วนของข้อมูลที่ถูกฝึก
cv	10	จำนวนรอบทั้งหมด

ตารางที่ 4.6 ค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุดสำหรับแบบจำลอง LightGBM ในแต่ละเทคนิค

ตัวแปรตาม	ไฮเปอร์พารามิเตอร์	Grid Search	Random Search
Match_Result	num_leaves	31	31
	max_depth	10	10
	learning_rate	0.01	0.01
	n_estimators	100	100
	min_child_samples	10	10
	subsample	0.8	1

ตารางที่ 4.7 การเปรียบเทียบค่าประสิทธิภาพของแบบจำลองการจำแนก LightGBM ก่อน และหลังการปรับจูนไฮเปอร์พารามิเตอร์ ของ Match_Result (เลือกคุณลักษณะด้วยสารสนเทศร่วม)

ตัวแปรตาม	เทคนิค	Accuracy	Precision	Recall	F1-Score	ROC
Match_Result	Default	0.611	0.577	0.611	0.570	0.726
	Grid Search	0.594	0.577	0.594	0.584	0.762
	Random Search	0.594	0.578	0.594	0.584	0.762

จากตารางที่ 4.7 พบว่าการปรับจูนไฮเปอร์พารามิเตอร์ของแบบจำลอง LightGBM ให้ผลลัพธ์ที่น่าสนใจ โดย Grid Search และ Random Search ให้ค่าประสิทธิภาพเหมือนกันทุกตัวชี้วัด แม้ว่าค่า Accuracy จะลดลงจาก 61.1% เป็น 59.4% แต่ค่า F1-Score เพิ่มขึ้นจาก 0.570 เป็น 0.584 และค่า ROC เพิ่มขึ้นอย่างชัดเจนจาก 0.726 เป็น 0.762 ซึ่งแสดงให้เห็นว่าแบบจำลองที่ผ่านการปรับจูนมีความสามารถในการแยกแยะคลาสที่ดีกว่า การที่ Grid Search และ Random Search ให้ผลเหมือนกันทุกตัวชี้วัดแสดงว่าทั้งสองเทคนิคสามารถค้นหาพารามิเตอร์ที่เหมาะสมได้อย่างมีประสิทธิภาพเท่าเทียมกัน โดย Random Search อาจมีข้อได้เปรียบในเรื่องความรวดเร็วในการประมวลผลเนื่องจากไม่ต้องทดลองทุกชุดพารามิเตอร์เหมือน Grid Search จึงแนะนำให้เลือกใช้ Random Search เนื่องจากให้ผลลัพธ์เดียวกันแต่ประหยัดเวลามากกว่า การปรับจูนไฮเปอร์พารามิเตอร์ช่วยปรับปรุงประสิทธิภาพของแบบจำลองในด้าน F1-Score และ ROC ซึ่งเป็นตัวชี้วัดที่สำคัญสำหรับการจำแนกประเภท

4.3.2 การเพิ่มคุณลักษณะโดยการ Feature Engineering

ผู้วิจัยได้ทำการเพิ่มคุณลักษณะเข้าไปหลังจากกระบวนการของการเลือกคุณลักษณะแบบความสำคัญของคุณลักษณะ โดยมีจุดประสงค์เพื่อเพิ่มประสิทธิภาพในการทำนายของแบบจำลอง ในการวิจัยนี้ได้นำเสนอตัวอย่าง 5 คุณลักษณะหลักในการเพิ่มประสิทธิภาพให้แบบจำลอง LightGBM ดังนี้

1) Goals_per_Shot: จำนวนประตูที่ได้ต่อจำนวนการยิง คำนวณเป็นค่าเฉลี่ยของ 3 และ 5 นัดหลังสุด

2) xG: ค่าความคาดหวังที่จะได้ประตู คำนวณเป็นค่าเฉลี่ยของ 3 และ 5 นัดหลังสุด

3) xG_diff: ผลต่างระหว่างอัตราความคาดหวังที่จะได้ประตูกับจำนวนประตูจริง คำนวณเป็นค่าเฉลี่ยของ 3 และ 5 นัดหลังสุด

4) Shot_eff: อัตราส่วนระหว่างจำนวนประตูต่อจำนวนการยิง คำนวณเป็นค่าเฉลี่ยของ 3 และ 5 นัดหลังสุด

5) SoT_rate: อัตราส่วนระหว่างจำนวนการยิงเข้ากรอบต่อจำนวนการยิงทั้งหมด คำนวณเป็นค่าเฉลี่ยของ 3 และ 5 นัดหลังสุด

ตารางที่ 4.8 ผลการเปรียบเทียบประสิทธิภาพของแบบจำลองการจำแนก LightGBM ของการเพิ่มคุณลักษณะสังเคราะห์

คุณลักษณะที่เพิ่มเข้ามา	Accuracy	Precision	Recall	F1-Score	ROC
Default	0.611	0.577	0.611	0.570	0.726
Add_all_feature	0.656	0.619	0.656	0.625	0.826
Goals_per_Shot_roll3	0.628	0.578	0.628	0.585	0.788
Goals_per_Shot_roll5	0.614	0.560	0.614	0.582	0.783
xG_roll3	0.610	0.562	0.610	0.569	0.761
xG_roll5	0.611	0.573	0.611	0.572	0.763
xG_diff_roll3	0.615	0.575	0.615	0.582	0.783
xG_diff_roll5	0.628	0.591	0.628	0.588	0.782
Shot_eff_roll3	0.609	0.568	0.609	0.572	0.770
Shot_eff_roll5	0.613	0.563	0.613	0.571	0.762
SoT_rate_roll3	0.593	0.538	0.593	0.550	0.757
SoT_rate_roll5	0.605	0.572	0.605	0.574	0.759
Goals_per_Shot_roll3 xG_diff_roll5	0.660	0.617	0.660	0.619	0.820

จากตารางที่ 4.8 การพัฒนาแบบจำลองด้วยเทคนิค Feature Engineering พบว่า การเพิ่มคุณลักษณะทั้งหมด (Add_all_feature) ส่งผลให้ได้ประสิทธิภาพการทำนายที่ดีที่สุดในทุกตัวชี้วัด โดยมีค่า Accuracy เท่ากับ 0.656, Precision เท่ากับ 0.619, F1-Score เท่ากับ 0.625 และ ROC AUC เท่ากับ 0.826 ซึ่งมีค่าสูงกว่าแบบจำลองพื้นฐาน (Default) อย่างชัดเจนในทุกตัวชี้วัด การผสมรวมคุณลักษณะ Goals_per_Shot_roll3 กับ xG_diff_roll5 ให้ผลลัพธ์การทำนายที่ดีเป็นอันดับสอง โดยสามารถบรรลุค่า Accuracy สูงสุดที่ 0.660 แต่มีค่า ROC AUC ที่ 0.820 ต่ำกว่าเล็กน้อย เมื่อพิจารณาการใช้คุณลักษณะเดี่ยว พบว่า Goals_per_Shot และ xG_diff มีศักยภาพในการทำนายสูงกว่าคุณลักษณะอื่น ๆ โดยเฉพาะอย่างยิ่งเมื่อคำนวณจากค่าเฉลี่ยของ 3 นัดหลังสุด ซึ่งตรง

ข้ามกับคุณลักษณะ SoT_rate ที่ให้ประสิทธิภาพการทำนายต่ำที่สุด การวิเคราะห์เปรียบเทียบระหว่างการใช้ค่าเฉลี่ย 3 นัดหลังสุด (roll3) กับ 5 นัดหลังสุด (roll5) แสดงให้เห็นว่าการใช้ข้อมูลระยะสั้นมีแนวโน้มให้ผลการทำนายที่แม่นยำกว่า ซึ่งสะท้อนถึงลักษณะของข้อมูลฟุตบอลที่มีความผันผวนสูงและแนวโน้มล่าสุดมีน้ำหนักมากกว่าข้อมูลในอดีตที่ห่างไกล การสังเคราะห์คุณลักษณะหลายตัวเข้าด้วยกันแสดงให้เห็นประสิทธิภาพที่เหนือกว่าการใช้คุณลักษณะเดี่ยว ซึ่งสอดคล้องกับทฤษฎีของ ensemble learning ที่ว่าการรวมข้อมูลจากหลายแหล่งสามารถลดความคลาดเคลื่อนและเพิ่มความแม่นยำในการทำนายได้ ผลการศึกษาชิ้นนี้จึงยืนยันถึงประสิทธิภาพของเทคนิค Feature Engineering ในการยกระดับสมรรถนะของแบบจำลอง LightGBM อย่างมีนัยสำคัญ โดยสามารถเพิ่มค่า Accuracy จาก 0.611 เป็น 0.656 และยกระดับค่า ROC AUC จาก 0.726 เป็น 0.826 ซึ่งถือเป็นการพัฒนาที่สำคัญและมีความหมายทางสถิติสำหรับการประยุกต์ใช้ในงานทำนายผลการแข่งขันกีฬาฟุตบอล

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

จากบทที่ 4 ได้นำเสนอผลการประเมิน และการอภิปรายผลจากการสร้างแบบจำลองการเรียนรู้ของเครื่องจักรเพื่อทำนายผลการแข่งขันฟุตบอลพรีเมียร์ลีก โดยแบ่งการศึกษาออกเป็นสองส่วนหลักคือ การถดถอยสำหรับทำนายจำนวนประตู (Goals_For) และการจำแนกประเภทสำหรับทำนายผลการแข่งขัน (Match_Result) ซึ่งผลลัพธ์จากการทดลอง และแบบจำลองและเทคนิคการเลือกคุณลักษณะต่างๆ ได้นำมาซึ่งข้อค้นพบที่สำคัญหลายประการในการทำแบบจำลองการทำนายผลการแข่งขันพรีเมียร์ลีก

5.1 การสรุปผลการวิจัย

5.1.1 การทำนายตัวแปรการถดถอย (Goals_For)

สำหรับการทำนายจำนวนประตู Goals_For ซึ่งเป็นข้อมูลประเภทจำนวนนับ ได้ทำการเปรียบเทียบแบบจำลองการถดถอยแบบปัวซอง (Poisson Regression) แบบจำลองทวินามเชิงลบ (Negative Binomial) และแบบจำลองศูนย์พอง (Zero-Inflated Poisson) ร่วมกับเทคนิคการเลือกคุณลักษณะสามวิธี ได้แก่ สหสัมพันธ์ (Correlation) ข้อมูลร่วมกัน (Mutual Information) และความสำคัญของคุณลักษณะ (Feature Importance)

ผลการวิจัยพบว่าแบบจำลองปัวซองแสดงประสิทธิภาพที่เหนือกว่าอย่างชัดเจน โดยมีค่า AIC และ BIC ต่ำที่สุดในทุกเทคนิคการเลือกคุณลักษณะ เนื่องจากชุดข้อมูลมีค่าเฉลี่ยของ Goals_For อยู่ที่ 1.3822 และความแปรปวนอยู่ที่ 1.6357 ซึ่งมี Overdispersion น้อย ทำให้แบบจำลองปัวซองเหมาะสมกับลักษณะข้อมูลนี้ ในขณะที่ชุดข้อมูลมีค่าศูนย์เพียง 28.1% ซึ่งไม่มากพอที่จะใช้แบบจำลองศูนย์พอง, การเลือกคุณลักษณะด้วยเทคนิคสหสัมพันธ์และความสำคัญของคุณลักษณะ ให้ผลลัพธ์ที่ดีที่สุดและใกล้เคียงกัน (AIC: 12115.427, BIC: 12173.253) ส่วนเทคนิคข้อมูลร่วมกันมีประสิทธิภาพรองลงมาเล็กน้อย และแบบจำลองทวินามเชิงลบให้ประสิทธิภาพที่แย่ที่สุดด้วยค่า Log-Likelihood ที่ต่ำมาก (-1269.869 ถึง -1270.048)

5.1.2 การทำนายตัวแปรจำแนกประเภท (Match_Result)

สำหรับการทำนาย Match_Result ได้ทำการเปรียบเทียบแบบจำลองการจำแนกสี่ประเภท ได้แก่ ต้นไม้ตัดสินใจ (Decision Tree), ป่าแบบสุ่ม (Random Forest), XGBoost และ LightGBM ร่วมกับเทคนิคการเลือกคุณลักษณะและเทคนิคการจัดการข้อมูลไม่สมดุล (None, Undersampling, Oversampling, SMOTE)

แบบจำลอง LightGBM ให้ประสิทธิภาพสูงสุด ด้วยค่า Accuracy เฉลี่ยรวม 0.672 รองลงมาคือ XGBoost (0.664) และ Random Forest (0.661) ส่วน Decision Tree แสดงประสิทธิภาพต่ำที่สุด (0.589), การเลือกคุณลักษณะด้วยเทคนิคความสำคัญของคุณลักษณะให้ผลลัพธ์ดีที่สุดเมื่อใช้ร่วมกับ LightGBM โดยไม่ใช้เทคนิคการจัดการข้อมูลไม่สมดุล ให้ค่า Accuracy 0.611, Precision 0.577, Recall 0.611, F1-Score 0.570 และ AUC-ROC 0.726

การเปรียบเทียบเทคนิคการเลือกคุณลักษณะพบว่า:

- ความสำคัญของคุณลักษณะ: ให้ประสิทธิภาพดีที่สุด (Accuracy เฉลี่ย 0.496-0.611)
- ข้อมูลร่วมกัน: ให้ประสิทธิภาพรองลงมา (Accuracy เฉลี่ย 0.506-0.600)
- สหสัมพันธ์: ให้ประสิทธิภาพต่ำสุด (Accuracy เฉลี่ย 0.475-0.592)

5.1.3 การพัฒนาปรับปรุงแบบจำลอง

1) การปรับจูนไฮเปอร์พารามิเตอร์: การปรับจูนไฮเปอร์พารามิเตอร์ของแบบจำลอง LightGBM ด้วยเทคนิค Grid Search และ Random Search พบว่าทั้งสองเทคนิคให้ผลลัพธ์เหมือนกันทุกตัวชี้วัด, ค่า F1-Score เพิ่มขึ้นจาก 0.570 เป็น 0.584, ค่า ROC เพิ่มขึ้นอย่างชัดเจนจาก 0.726 เป็น 0.762 แต่ ค่า Accuracy ลดลงจาก 61.1% เป็น 59.4% ซึ่งการปรับจูนไฮเปอร์พารามิเตอร์ช่วยปรับปรุงประสิทธิภาพของแบบจำลองในด้าน F1-Score และ ROC ซึ่งเป็นตัวชี้วัดที่สำคัญสำหรับการจำแนกประเภท

2) การเพิ่มคุณลักษณะโดยการ Feature Engineering: ทำการเพิ่มคุณลักษณะใหม่ 5 ประเภทหลัก ได้แก่:

2.1) Goals_per_Shot: จำนวนประตูที่ได้ต่อจำนวนการยิง

2.2) xG: ค่าความคาดหวังที่จะได้ประตู

2.3) xG_diff: ผลต่างระหว่างอัตราความคาดหวังที่จะได้ประตูกับจำนวนประตูจริง

2.4) Shot_eff: อัตราส่วนระหว่างจำนวนประตูต่อจำนวนการยิง

2.5) SoT_rate: อัตราส่วนระหว่างจำนวนการยิงเข้ากรอบต่อจำนวนการยิงทั้งหมด

การเพิ่มคุณลักษณะทั้งหมด (Add_all_feature) ส่งผลให้ได้ประสิทธิภาพการทำนายที่ดีที่สุดในทุกตัวชี้วัด โดยมีค่า Accuracy เท่ากับ 0.656, Precision เท่ากับ 0.619, F1-Score เท่ากับ 0.625 และ ROC AUC เท่ากับ 0.826 การสังเคราะห์คุณลักษณะ Goals_per_Shot_roll3 กับ xG_diff_roll5 ให้ผลลัพธ์การทำนายที่ดีเป็นอันดับสอง โดย

สามารถบรรลุค่า Accuracy สูงสุดที่ 0.660, การใช้ข้อมูลระยะสั้น (3 นัดหลังสุด) มีแนวโน้มให้ผลการทำนายที่แม่นยำกว่าการใช้ข้อมูลระยะยาว (5 นัดหลังสุด)

การพัฒนาด้วย Feature Engineering สามารถเพิ่มค่า Accuracy จาก 0.611 เป็น 0.656 และยกระดับค่า ROC AUC จาก 0.726 เป็น 0.826 ซึ่งถือเป็นการพัฒนาที่สำคัญและมีความหมายทางสถิติ

5.2 ข้อเสนอแนะ

จากการศึกษาวิจัย มีข้อเสนอแนะที่สามารถนำไปปรับปรุงหรือพัฒนาต่อในอนาคตดังนี้

5.2.1 การรวบรวมข้อมูลและคุณลักษณะเพิ่มเติม

- ข้อมูลเกี่ยวกับสถิติของผู้เล่นรายบุคคล เช่น การยิงประตู การจ่ายบอล การป้องกัน ของผู้เล่นในแต่ละนัด
- ข้อมูลเกี่ยวกับอาการบาดเจ็บ ขวัญกำลังใจ และสภาพทีม ณ ขณะนั้น
- ข้อมูลสภาพอากาศ สนามแข่งขัน และปัจจัยภายนอกอื่นๆ
- ข้อมูลการเปลี่ยนแปลงผู้เล่นและกลยุทธ์ของทีม

5.2.2 การวิเคราะห์ด้วยอนุกรมเวลา (Time Series)

สถิติของกีฬาฟุตบอลมีการเปลี่ยนแปลงตามช่วงเวลา เช่น ฟอร์มการเล่นของทีม ความได้เปรียบในการเล่นที่บ้านกับนอกบ้าน การใช้การวิเคราะห์อนุกรมเวลาเพื่อสะท้อนถึงฟอร์มการเล่นของทีมในขณะนั้น อาจช่วยให้แบบจำลองสามารถจับรูปแบบการเปลี่ยนแปลงของประสิทธิภาพทีมได้ดีขึ้น ทำให้ความแม่นยำในการทำนายสูงขึ้น

5.2.3 การใช้แบบจำลองประเภทการเรียนรู้เชิงลึก (Deep Learning)

แบบจำลองการเรียนรู้เชิงลึกประเภทต่างๆ อย่างเช่น โครงข่ายประสาทเทียม (Neural Network) และ Recurrent Neural Network (RNN) อาจเหมาะสมสำหรับข้อมูลการวิเคราะห์ที่มีลักษณะเป็นลำดับหรือเชิงซับซ้อน โดยเฉพาะในการจับความสัมพันธ์ที่ไม่เป็นเชิงเส้นระหว่างตัวแปรต่างๆ

5.2.4 การศึกษาเปรียบเทียบข้ามลีก

ขยายการศึกษาไปยังลีกฟุตบอลอื่นๆ เพื่อทดสอบความสามารถในการประยุกต์ใช้แบบจำลองข้ามบริบทที่แตกต่างกัน และเพื่อพัฒนาแบบจำลองที่มีความทั่วไปมากขึ้น

การวิจัยในครั้งนี้ได้แสดงให้เห็นถึงศักยภาพของการใช้เทคนิคการเรียนรู้ของเครื่องจักรในการทำนายผลการแข่งขันฟุตบอล โดยเฉพาะการใช้แบบจำลอง LightGBM ร่วมกับเทคนิค Feature Engineering ที่สามารถให้ประสิทธิภาพการทำนายที่น่าพอใจ ซึ่ง

สามารถนำไปประยุกต์ใช้ในการวิเคราะห์ และทำนายผลการแข่งขันกีฬาฟุตบอลได้อย่างมีประสิทธิภาพ

เอกสารอ้างอิง

นวลวรรณ สุนทรภิษัช. (2020). **การเรียนรู้ของเครื่อง** กรุงเทพมหานคร: สำนักพิมพ์มหาวิทยาลัยเกษตรศาสตร์.

รศ.ดร. กิติ์สุชาต พสุภา (2021). **ระบบอัจฉริยะชั้นสูง: ทฤษฎี อัลกอรึทึม และการประยุกต์ใช้** กรุงเทพมหานคร: สำนักพิมพ์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กอบเกียรติ สระอุบล. (2020). เรียนรู้ Data Science และ AI: Machine Learning ด้วย Python. กรุงเทพมหานคร: สำนักพิมพ์มีเดียเนทเวิร์ค

กัลยา วานิชย์บัญชา. (2017). **หลักสถิต**. กรุงเทพมหานคร: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย

Ahmed et al. (2022). **Predicting the Outcome of Soccer Matches Using Machine Learning and Statistical Analysis**. ResearchGate

Artur. (2021). **Predicting Football Results with the Poisson Regression Model**. artiebits.com blog

Azhari, H. R., et al. (2018). **"Predicting Final Result of Football Match Using Poisson Regression Model,"** *Journal of Physics: Conference Series* 1114:

Huynh, Langdon. (2024). **Enhancing Football Match Predictions through AI and Machine Learning in the English Premier League**. NHSJS website

Omoregie, S., and Monday, H. (2021). **Predicting the Outcome of English Premier League Matches using Machine Learning.** ResearchGate

Raju et al. (2023). **Predicting Football Match Outcomes with Machine Learning Approaches.** ResearchGate.

Ulmer, J. & Fernandez, A. (2014). **Predicting Soccer Match Results in the English Premier League**. Stanford University

ภาคผนวก

ภาคผนวก

แบ่งการทำงานของ Google colab หลักๆแบ่งออกเป็นส่วนๆดังต่อไปนี้

1. การนำเข้าไลบารี่ที่ใช้ในการวิจัย

```
import warnings
warnings.filterwarnings('ignore')

# Data manipulation
import numpy as np
import pandas as pd

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Machine Learning & Modeling
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    roc_auc_score, confusion_matrix,
    mean_absolute_error, mean_squared_error
)
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier

# Imbalanced data
from imblearn.under_sampling import RandomUnderSampler
```

```
from imblearn.over_sampling import SMOTE, RandomOverSampler
from imblearn.pipeline import Pipeline

# Feature Selection
from sklearn.feature_selection import mutual_info_regression

# Statistical Modeling
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.discrete.count_model import ZeroInflatedPoisson
import statsmodels.discrete.discrete_model as discrete

# Statistical Distributions
from scipy.stats import poisson, nbinom
```

2. การเลือกคุณลักษณะด้วยสหสัมพันธ์, การใช้ข้อมูลร่วม, ความสำคัญของคุณลักษณะ และ VIF

```
def select_features_by_correlation(df, target_col, threshold=0.2, top_k=30):
    print(f"🔍 Selecting features by correlation (threshold: {threshold})")
    numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
    feature_cols = [col for col in numeric_cols if col != target_col]
    corr_matrix = df[numeric_cols].corr()
    target_corr =
corr_matrix[target_col].abs().drop(target_col).sort_values(ascending=False)

    selected = target_corr[target_corr > threshold].head(top_k).index.tolist()
    print(f"✅ Selected {len(selected)} features by correlation")
    return selected
```

```
def select_features_by_mutual_info(df, target_col, top_k=30):
    print(f"🔍 Selecting features by mutual information (top {top_k})")
    numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
    feature_cols = [col for col in numeric_cols if col != target_col]
    X = df[feature_cols]
    y = df[target_col]
    mi_scores = mutual_info_regression(X, y, random_state=42)
    mi_df = pd.DataFrame({
        'Feature': feature_cols,
        'MI_Score': mi_scores
    }).sort_values('MI_Score', ascending=False)
    selected = mi_df.head(top_k)['Feature'].tolist()
    # Plot top 10
    plt.figure(figsize=(10, 6))
    sns.barplot(data=mi_df.head(10), x='MI_Score', y='Feature', palette='viridis')
    plt.title('Top 10 Features by Mutual Information')
    plt.tight_layout()
    plt.show()
    print(f"✅ Selected {len(selected)} features by mutual information")
    return selected

def select_features_by_importance(df, target_col, top_k=30):
    print(f"🔍 Selecting features by Random Forest importance (top {top_k})")
    numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
    feature_cols = [col for col in numeric_cols if col != target_col]
    X = df[feature_cols]
    y = df[target_col]
```

```
    rf = RandomForestRegressor(n_estimators=100, random_state=42)
    rf.fit(X, y)
    importance_df = pd.DataFrame({
        'Feature': feature_cols,
        'Importance': rf.feature_importances_
    }).sort_values('Importance', ascending=False)
    selected = importance_df.head(top_k)['Feature'].tolist()
    # Plot top features
    plt.figure(figsize=(10, 6))
    top_10 = importance_df.head(10)
    sns.barplot(data=top_10, x='Importance', y='Feature', palette='plasma')
    plt.title('Top 10 Features by Random Forest Importance')
    plt.tight_layout()
    plt.show()
    print(f"✅ Selected {len(selected)} features by Random Forest importance")
    return selected


def apply_vif_filtering(df, features, target_col, vif_threshold=20, corr_threshold=0.5,
corr_diff_tol=0.05):
    print(f"🔧 Applying smart VIF filtering (VIF threshold: {vif_threshold}, Corr
threshold: {corr_threshold})")
    X = df[features].copy()
    y = df[target_col]
    removed_features = []
    while True:
        vif_df = pd.DataFrame()
        vif_df["Feature"] = X.columns
        vif_df["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
```

```
        vif_df["Corr_with_target"] = [abs(df[f].corr(y)) for f in X.columns]
        # Step 1: หา candidate ที่ VIF > threshold และ Corr < corr_threshold
        drop_candidates = vif_df[(vif_df["VIF"] > vif_threshold) &
(vif_df["Corr_with_target"] < corr_threshold)]
        # Step 2: หาคู่ฟีเจอร์ที่มี VIF สูงมากและ Corr กับ target ใกล้เคียงกัน (auto drop ตัว
correlation ต่ำกว่า)
        dropped_in_pair = False
        for i, f1 in enumerate(vif_df["Feature"]):
            for f2 in vif_df["Feature"][i+1:]:
                # เช็คว่าทั้งคู่ VIF สูงและ correlation ต่างกันน้อย
                if (vif_df.loc[vif_df["Feature"] == f1, "VIF"].values[0] > vif_threshold and
                    vif_df.loc[vif_df["Feature"] == f2, "VIF"].values[0] > vif_threshold):
                    corr1 = vif_df.loc[vif_df["Feature"] == f1, "Corr_with_target"].values[0]
                    corr2 = vif_df.loc[vif_df["Feature"] == f2, "Corr_with_target"].values[0]

                    if abs(corr1 - corr2) < corr_diff_tol:
                        # ตัด feature correlation ต่ำกว่าออก
                        to_drop = f1 if corr1 < corr2 else f2
                        if to_drop in X.columns:
                            X = X.drop(columns=[to_drop])
                            removed_features.append(to_drop)
                            print(f"   Removed {to_drop} due to high VIF and similar corr with
pair")
                            dropped_in_pair = True
                            break
            if dropped_in_pair:
                break
        # ถ้าไม่มี candidate ที่จะตัดและไม่มีคู่ให้ตัดแล้ว ให้หยุด
```

```
        if drop_candidates.empty and not dropped_in_pair:
            break
        # ถ้าไม่มีคู่ให้ตัดแต่มี candidate ให้ตัดตัวที่ VIF สูงสุดใน candidate
        if not drop_candidates.empty:
            max_row = drop_candidates.loc[drop_candidates["VIF"].idxmax()]
            max_feature = max_row["Feature"]
            max_vif = max_row["VIF"]
            max_corr = max_row["Corr_with_target"]
            if max_feature in X.columns:
                X = X.drop(columns=[max_feature])
                removed_features.append(max_feature)
                print(f"   Removed {max_feature} (VIF: {max_vif:.2f}, Corr: {max_corr:.2f})")
    final_features = X.columns.tolist()
    print(f"✅ Final features after smart VIF filtering: {len(final_features)}")
    print(f"   Removed {len(removed_features)} features due to high VIF & low corr or
pair similarity")
    print(vif_df.sort_values("VIF", ascending=False))
    return final_features
```

3. การทำ Rolling Data ข้อมูลเป็นค่าเฉลี่ย 3 และ 5

```
def create_rolling_features(df, numeric_cols, group_cols=['Team', 'Season'],
                            exclude_cols=['Goals', 'Goal_Creating_Actions', 'Non_Pen_xG'],
                            windows=[3, 5]):
    rolling_data = {}
    for col in numeric_cols:
        if col in df.columns and col not in exclude_cols:
            for w in windows:
                new_col_name = f'{col}_roll{w}'
```

```
            rolling_series = (
                df.groupby(group_cols)[col]
                .rolling(window=w, min_periods=1)
                .mean()
                .reset_index(level=group_cols, drop=True)
                .fillna(0)
            )
            rolling_data[new_col_name] = rolling_series
    return rolling_data
```

4. การทำนายการถดถอยปัวซอง, ทวินามเชิงลบ และศูนย์พอง

```
def poisson_regression(x_train, y_train, x_test, y_test):
  results = []
  x_train_const = sm.add_constant(x_train, has_constant='add', prepend=False)
  x_test_const = sm.add_constant(x_test, has_constant='add', prepend=False)
  # ========== 1. Poisson GLM ==========
  print("Fitting Poisson GLM...")
  poisson_model = sm.GLM(y_train, x_train_const, family=sm.families.Poisson())
  poisson_result = poisson_model.fit()
  # Predictions
  y_train_pred_pois = poisson_result.predict(x_train_const)
  y_test_pred_pois = poisson_result.predict(x_test_const)
  # λ for PMF
  lambda_pred_pois = poisson_result.predict(x_test_const)
  # Log-likelihood calculations
  llf_train_pois = poisson_result.llf
  llf_test_pois = np.sum(poisson.logpmf(y_test, np.maximum(y_test_pred_pois, 1e-
10)))
```

```
# Null model log-likelihood
mu_null = np.mean(y_train)
llf_null_pois = np.sum(poisson.logpmf(y_train, mu_null))
llf_null_test_pois = np.sum(poisson.logpmf(y_test, mu_null))
n = len(y_train)  # number of observations
k = poisson_result.df_model + 1  # add 1 for intercept
bic_pois = -2 * poisson_result.llf + np.log(n) * k
results.append({
  'Model': 'Poisson GLM',
  'Pseudo_R2': pseudo_r2_mcfadden(llf_test_pois, llf_null_test_pois),
  'LogLik': llf_test_pois,
  'AIC': poisson_result.aic,
  'BIC': bic_pois
  #'BIC': poisson_result.bic
  })

# ========== 2. Negative Binomial GLM ==========
print("Fitting Negative Binomial GLM...")
print(" 🔍 วินิจฉัยปัญหา Negative Binomial")
print("="*50)
# 1. ตรวจสอบความจำเป็นของ Negative Binomial
mean_y = np.mean(y_train)
var_y = np.var(y_train)
dispersion_ratio = var_y / mean_y
print(f" 📊 ข้อมูลพื้นฐาน:")
print(f"Mean: {mean_y:.4f}")
print(f"Variance: {var_y:.4f}")
print(f"Dispersion Ratio (Var/Mean): {dispersion_ratio:.4f}")
```

```
if dispersion_ratio < 1.2:
    print(" ไม่มี Overdispersion --> Poisson เหมาะสม")
elif dispersion_ratio > 2.0:
    print(" Overdispersion สูง --> Negative Binomial ควรดีกว่า")
else:
    print(" Overdispersion เล็กน้อย --> ต้องทดสอบเพิ่มเติม")
negbin_model = discrete.NegativeBinomial(y_train, x_train_const)
negbin_result = negbin_model.fit(disp=False)
y_train_pred_nb = negbin_result.predict(x_train_const)
y_test_pred_nb = negbin_result.predict(x_test_const)
# λ for PMF
lambda_pred_nb = negbin_result.predict(x_test_const)
# Log-likelihood calculations
llf_train_nb = negbin_result.llf
# Negative Binomial test log-likelihood
alpha = negbin_result.scale  # dispersion parameter
print(f"Dispersion parameter (alpha): {alpha}")
if alpha > 0:
    r = 1/alpha
    mu = np.maximum(y_test_pred_nb, 1e-10)
    p = r / (r + mu)
    p = np.clip(p, 1e-10, 1-1e-10)  # Avoid numerical issues
    llf_test_nb = np.sum(nbinom.logpmf(y_test, r, p))
    # Null model
    mu_null_nb = np.mean(y_train)
    p_null = r / (r + mu_null_nb)
    llf_null_nb = np.sum(nbinom.logpmf(y_train, r, p_null))
    llf_null_test_nb = np.sum(nbinom.logpmf(y_test, r, p_null))
```

```
else:
    llf_test_nb = np.nan
    llf_null_nb = np.nan
    llf_null_test_nb = np.nan

results.append({
    'Model': 'Negative Binomial GLM',
    'Pseudo_R2': pseudo_r2_mcfadden(llf_test_nb, llf_null_test_nb),
    'LogLik': llf_test_nb,
    'AIC': negbin_result.aic,
    'BIC': negbin_result.bic
})

# ========== 3. Zero-Inflation GLM ==========
print('-'*100)
print("Fitting Zero-Inflated Poisson...")
  # 1. ตรวจสอบ Zero-inflation
zero_count = np.sum(y_train == 0)
total_count = len(y_train)
observed_zero_prop = zero_count / total_count
# Expected zeros from regular Poisson
mean_y = np.mean(y_train)
expected_zero_prop = np.exp(-mean_y)
zero_ratio = observed_zero_prop / expected_zero_prop
print(f" 📊 Zero Analysis:")
print(f"Observed zeros: {zero_count}/{total_count} ({observed_zero_prop:.1%})")
print(f"Expected zeros (Poisson): {expected_zero_prop:.1%}")
print(f"Excess zero ratio: {zero_ratio:.2f}")
```

```
if zero_ratio < 1.2:
    print(" ไม่มี zero-inflation --> ZIP ไม่จำเป็น")
    is_zero_inflated = False
elif zero_ratio > 2.0:
    print(" Zero-inflation สูง --> ZIP ควรดีกว่า")
    is_zero_inflated = True
else:
    print(" Zero-inflation เล็กน้อย --> ต้องทดสอบ")
    is_zero_inflated = True
train_data = pd.DataFrame(x_train_const)
train_data['y'] = y_train
# Fit ZIP model
zip_model = ZeroInflatedPoisson(
    endog = y_train,
    exog = x_train_const,
    # inflation สามารถใช้ constant หรือ variables เดียวกันได้
    exog_infl=np.ones((len(y_train), 1))  # constant inflation
)
zip_result = zip_model.fit(disp=False)
# Predictions
y_train_pred_zip = zip_result.predict(x_train_const)
y_test_pred_zip = zip_result.predict(x_test_const, exog_infl=np.ones((len(y_test),
1)))
  # λ for PMF
#lambda_pred_zip = zip_result.predict(x_test_const)
# Log-likelihood calculations
llf_train_zip = zip_result.llf
# ZIP test log-likelihood (approximation)
```

```
    # ใช้ predicted mean เหมือน Poisson สำหรับการประมาณ
    llf_test_zip = np.sum(poisson.logpmf(y_test, np.maximum(y_test_pred_zip, 1e-10)))
    results.append({
        'Model': 'Zero-Inflated Poisson',
        'Pseudo_R2': pseudo_r2_mcfadden(llf_test_zip, llf_null_test_pois),
        'LogLik': llf_test_zip,
        'AIC': zip_result.aic,
        'BIC': zip_result.bic,
    })
    results_df = pd.DataFrame(results)
    return results_df,lambda_pred_pois


train_data = df_gf_corr[df_gf_corr['Season'] != '2023-2024']
test_data = df_gf_corr[df_gf_corr['Season'] == '2023-2024']
x_train_GF = train_data.drop(columns=['Goals_For','Season','Team','Opponent'])
y_train_GF = train_data['Goals_For']
x_test_GF = test_data.drop(columns=['Goals_For','Season','Team','Opponent'])
y_test_GF = test_data['Goals_For']
result_poi,y_pred_corr = poisson_regression(x_train_GF, y_train_GF, x_test_GF,
y_test_GF)
result_poi
```

5. การจำแนกประเภทของแบบจำลอง Decision Tree, Random Forest, XGBoost และ LightGBM

```
def classification_models(x_train, x_test, y_train, y_test, random_state=42):
    # เตรียมวิธีการสุ่มตัวอย่าง
    sampling_methods = {}
    # ไม่ใช้การสุ่มตัวอย่าง
```

```
    sampling_methods["Original"] = (x_train, y_train)


    # Under-sampling
    try:
        x_under, y_under =
RandomUnderSampler(random_state=random_state).fit_resample(x_train, y_train)
        sampling_methods["UnderSampling"] = (x_under, y_under)
    except:
        pass
    # Over-sampling
    try:
        x_over, y_over =
RandomOverSampler(random_state=random_state).fit_resample(x_train, y_train)
        sampling_methods["OverSampling"] = (x_over, y_over)
    except:
        pass
    # SMOTE
    try:
        x_smote, y_smote = SMOTE(random_state=random_state).fit_resample(x_train,
y_train)
        sampling_methods["SMOTE"] = (x_smote, y_smote)
    except:
        pass
    # เตรียมโมเดล
    models = {
        "Decision Tree": DecisionTreeClassifier(random_state=random_state,
max_depth=10),
```

```
        "LightGBM": LGBMClassifier(random_state=random_state, verbose=-1, n_jobs=-
1),
        "Random Forest": RandomForestClassifier(random_state=random_state,
n_estimators=100, n_jobs=-1),
        "XGBoost": XGBClassifier(random_state=random_state, eval_metric='logloss',
n_jobs=-1)
    }
    results = []
    conf_matrices = {}
    # ฝึกและประเมินโมเดล
    for method_name, (x_res, y_res) in sampling_methods.items():
        for model_name, model in models.items():
            try:
                # ฝึกโมเดล
                model.fit(x_res, y_res)
                y_pred = model.predict(x_test)
                # คำนวณ ROC AUC
                roc_auc = np.nan
                try:
                    y_proba = model.predict_proba(x_test)
                    if len(np.unique(y_test)) > 2:
                        roc_auc = roc_auc_score(y_test, y_proba, multi_class='ovr')
                    else:
                        roc_auc = roc_auc_score(y_test, y_proba[:, 1])
                except:
                    pass
                # คำนวณเมตริกต่างๆ
                results.append({
```

```
                'Model': model_name,
                'Sampling': method_name,
                'Accuracy': accuracy_score(y_test, y_pred),
                'Precision': precision_score(y_test, y_pred, average='weighted',
zero_division=0),
                'Recall': recall_score(y_test, y_pred, average='weighted',
zero_division=0),
                'F1-Score': f1_score(y_test, y_pred, average='weighted',
zero_division=0),
                'ROC AUC': roc_auc,
                'Train_Size': len(x_res)
            })
            # เก็บ confusion matrix
            conf_matrices[(model_name, method_name)] = confusion_matrix(y_test,
y_pred)
        except Exception as e:
            # กรณีที่เกิดข้อผิดพลาด
            results.append({
                'Model': model_name,
                'Sampling': method_name,
                'Accuracy': np.nan,
                'Precision': np.nan,
                'Recall': np.nan,
                'F1-Score': np.nan,
                'ROC AUC': np.nan,
                'Train_Size': len(x_res) if 'x_res' in locals() else np.nan
            })
    # สร้าง DataFrame และเรียงลำดับตาม Model
```

```
    results_df = pd.DataFrame(results)
    results_df = results_df.sort_values(['Model', 'F1-Score'], ascending=[True, False])

    # จัดเรียงคอลัมน์ให้ดูเป็นระเบียบ
    column_order = ['Model', 'Sampling', 'Accuracy', 'Precision', 'Recall', 'F1-Score', 'ROC
AUC', 'Train_Size']
    results_df = results_df[column_order].reset_index(drop=True)
    return results_df, conf_matrices


train_data = df_mr_corr[df_mr_corr['Season'] != '2023-2024']
test_data = df_mr_corr[df_mr_corr['Season'] == '2023-2024']
x_train_MR = train_data.drop(columns=['Match_Result','Season', 'Team', 'Opponent'])
y_train_MR = train_data['Match_Result']
x_test_MR = test_data.drop(columns=['Match_Result','Season', 'Team', 'Opponent'])
y_test_MR = test_data['Match_Result']
MR_corr,MR_matrix = classification_models(x_train_MR, x_test_MR, y_train_MR,
y_test_MR)
MR_corr
```

6. การปรับจูนไฮเปอร์พารามิเตอร์

```
pipeline = Pipeline([
    ('clf', LGBMClassifier(objective='multiclass', num_class=3, class_weight='balanced',
verbose=-1))
])
# พารามิเตอร์สำหรับ Grid Search
param_grid = {
    'clf__num_leaves': [31, 50, 70],
    'clf__max_depth': [-1, 10, 20],
```

```
    'clf__learning_rate': [0.01, 0.05, 0.1],
    'clf__n_estimators': [100, 200],
    'clf__min_child_samples': [10, 20],
    'clf__subsample': [0.8, 1.0]
}
# ทำ GridSearchCV โดยใช้ Accuracy เป็นเกณฑ์
grid_search = GridSearchCV(
    estimator=pipeline,
    param_grid=param_grid,
    scoring='f1_weighted',  # <<== ใช้ Accuracy เป็นหลักเกณฑ์ในการเลือกโมเดล
    cv=10,
    verbose=1,
    n_jobs=-1
)
# เทรนโมเดล
grid_search.fit(x_train_MR, y_train_MR)
# ดึงโมเดลที่ดีที่สุด
best_model = grid_search.best_estimator_
# พยากรณ์
y_pred = best_model.predict(x_test_MR)
y_pred_prob = best_model.predict_proba(x_test_MR)
# ประเมินผลแบบ weighted
acc = accuracy_score(y_test_MR, y_pred)
prec = precision_score(y_test_MR, y_pred, average='weighted')
rec = recall_score(y_test_MR, y_pred, average='weighted')
f1 = f1_score(y_test_MR, y_pred, average='weighted')
roc_auc = roc_auc_score(y_test_MR, y_pred_prob, multi_class='ovr',
average='weighted')
```

```
# แสดงผล
print("Best Hyperparameters:", grid_search.best_params_)
print(f"Accuracy (test set):      {acc:.4f}")
print(f"Precision (weighted):     {prec:.4f}")
print(f"Recall (weighted):        {rec:.4f}")
print(f"F1 Score (weighted):      {f1:.4f}")
print(f"ROC AUC Score (weighted): {roc_auc:.4f}")


# สร้าง Pipeline
pipeline = Pipeline([
    ('clf', LGBMClassifier(objective='multiclass', num_class=3, class_weight='balanced',
verbose=-1))
])
# กำหนดพารามิเตอร์แบบ prefix ด้วยชื่อขั้นตอนใน pipeline
param_dist = {
    'clf__num_leaves': [31, 50, 70],
    'clf__max_depth': [-1, 10, 20],
    'clf__learning_rate': [0.01, 0.05, 0.1],
    'clf__n_estimators': [100, 200],
    'clf__min_child_samples': [10, 20],
    'clf__subsample': [0.8, 1.0]
}
# ใช้ RandomizedSearchCV
random_search = RandomizedSearchCV(estimator=pipeline,
                                   param_distributions=param_dist,
                                   n_iter=50,
                                   scoring='f1_weighted',  # หรือ f1_macro ถ้า class imbalance
```

```
                    cv=10,
                    random_state=42,
                    verbose=1,
                    n_jobs=-1)
# การฝึกโมเดล
random_search.fit(x_train_MR, y_train_MR)
#  ทำนายผลบน test set
rand_best_model = random_search.best_estimator_
y_pred = rand_best_model.predict(x_test_MR)
y_pred_prob = rand_best_model.predict_proba(x_test_MR)

# วัดผล Performance
acc = accuracy_score(y_test_MR, y_pred)
prec = precision_score(y_test_MR, y_pred, average='weighted')
rec = recall_score(y_test_MR, y_pred, average='weighted')
f1 = f1_score(y_test_MR, y_pred, average='weighted')
roc_auc = roc_auc_score(y_test_MR, y_pred_prob, multi_class='ovr',
average='weighted')

#  แสดงผลลัพธ์
print(random_search.best_params_)
print(f"Accuracy: {acc:.4f}")
print(f"Precision: {prec:.4f}")
print(f"Recall: {rec:.4f}")
print(f"F1 Score: {f1:.4f}")
print(f"ROC AUC Score: {roc_auc:.4f}")
```

ประวัติผู้เขียน

ชื่อ	นาย พิศิษฏ์ ชินกุลประสาน
วันเกิด	27 พฤศจิกายน 2535
ที่อยู่ปัจจุบัน	126/96 หมู่บ้าน ศุภาลัยวิวล์ กรุงเทพนนท์ ถ.กรุงเทพนนท์ ตำบล บางเขน อำเภอ เมืองนนทบุรี จังหวัด นนทบุรี 11000
ประวัติการศึกษา	(2558) วิศวกรรมศาสตรบัณฑิต สาขา วิศวกรไฟฟ้า มหาวิทยาลัยพระจอมเกล้าพระนครเหนือ (2566) วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการข้อมูลและการวิเคราะห์ เกรดเฉลี่ย 4.00 มหาวิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง