การศึกษาตัวจำแนกประเภทการเรียนรู้ของเครื่องเพื่อทำนายโรคหลอดเลือดสมอง

น.ส.ฐิติพร อ้ายดี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

A Study of Classifiers in Machine Learning for Stroke Prediction

Miss Thitiporn Eyedee

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Mechanical Engineering
Department of Mechanical Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

หัวข้อวิทยานิพนธ์	การศึกษาตัวจำแนกประเภทการเรียนรู้ของเครื่องเพื่อทำนายโรคหลอดเลือดสมอง
โดย	น.ส.ฐิติพร อ้ายดี
สาขาวิชา	วิศวกรรมเครื่องกล
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร.วิบูลย์ แสงวีระพันธุ์ศิริ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.รัชทิน จันทร์เจริญ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.วิบูลย์ แสงวีระพันธุ์ศิริ)

..... กรรมการ
(ดร.สุรัฐ ขวัญเมือง)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร.อานันท์ สุตาพันธ์)

ฐิติพร อ้ายดี : การศึกษาตัวจำแนกประเภทการเรียนรู้ของเครื่องเพื่อทำนายโรคหลอดเลือดสมอง . (A Study of Classifiers in Machine Learning for Stroke Prediction) อ.ที่ปรึกษาหลัก : ศ. ดร.วิบูลย์ แสงวีระพันธุ์ศิริ

โรคหลอดเลือดสมองเป็นโรคที่มีอัตราการเสียชีวิตสูงและยังเป็นสาเหตุที่ทำให้เกิดการพิการ การทำวิทยานิพนธ์นี้มีจุดประสงค์ศึกษาตัวจำแนกประเภทในการเรียนรู้ของเครื่องที่มีประสิทธิภาพกับการทำนายโรคหลอดเลือดสมอง โดยใช้ตัวจำแนกประเภทกับข้อมูลที่ได้จากบันทึกของศูนย์โรคหลอดเลือดสมองแบบครบวงจรโรงพยาบาลจุฬาลงกรณ์ พิจารณากับปัจจัยเฉพาะและทำการทดลองเพื่อความแม่นยำในการทำนายผล การศึกษาตัวจำแนกประเภทมีทั้ง K-Nearest Neighbors, Support Vector Machine, Random Forest และ Adaboost ปรับจูนพารามิเตอร์ที่เหมาะสมกับข้อมูลทางการแพทย์ที่มีอยู่ ตัวจำแนกประเภทแบบ Random Forest ให้ผลลัพธ์ค่าเฉลี่ยความแม่นยำ 78% ในข้อมูลโรงพยาบาลจุฬาลงกรณ์ รวมถึงวิเคราะห์ปัจจัยเสี่ยงที่ทำให้เกิดโรคหลอดเลือดสมองด้วยวิธีการเรียนรู้ของเครื่องจากข้อมูลชุดที่ทำการศึกษา โดยใช้ TreeExplainer ประมาณค่าของ shapley value เพื่อแสดงผลความสำคัญของปัจจัยเฉพาะ ทั้งนี้เพื่อต่อยอดแนวทางในการปรับใช้ข้อมูลที่จะเก็บเพิ่มขึ้นได้ในอนาคต

สาขาวิชา วิศวกรรมเครื่องกล ลายมือชื่อนิสิต .....

ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6270344821 : MAJOR MECHANICAL ENGINEERING

KEYWORD: Machine Learning, Classification, Stroke, Prediction

Thitiporn Eyedee : A Study of Classifiers in Machine Learning for Stroke Prediction. Advisor: Prof. VIBOON SANGVERAPHUNSIRI, Ph.D.

Stroke has a high mortality rate and causes disability worldwide. Studying machine learning classifiers for stroke prediction is the purpose of this research. The Chulalongkorn Stroke Center of Excellence provides stroke patient data for this study. These data features and experiments were generated machine learning model with high true positive rate. K-Nearest Neighbors, Support Vector Machine, Random Forest and Adaboost are the classifiers that operated in this study. In the dataset from the King Chulalongkorn Memorial hospital, Random Forest produced high average precision of 78 percent. Moreover, machine learning were used to analyze the primary risk factor for stroke. TreeExplainer plots estimated shapley value for feature importance. This study will monitor and develop model for future data.

Field of Study: Mechanical Engineering Student's Signature .....

Academic Year: 2021 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงได้อย่างดียิ่งจากอาจารย์ที่ปรึกษา ศาสตราจารย์ ดร.วิบูลย์ แสงวีระพันธุ์ศิริ ที่ให้การสนับสนุนและคอยชี้แนะแนวทางที่เป็นประโยชน์ต่อการทำงานวิจัย ขอบคุณสมาชิกในห้องปฏิบัติการ The Regional Center of Robotics Technology ทั้งรุ่นพี่ รุ่นน้อง เพื่อน ที่ให้ความช่วยเหลือและเป็นกำลังใจในการทำวิจัยนี้ ขอบคุณมิตรสหายและครูอาจารย์ที่ให้คำปรึกษาและความรู้ในการดำเนินงาน สุดท้ายขอบคุณคุณพ่อที่คอยเลี้ยงดูและส่งเสริมให้ข้าพเจ้าจนสำเร็จทางการศึกษา แม้จะรอดูอยู่บนอีกภพภูมิ ขอบคุณคุณแม่ที่เป็นอีกแรงผลักดันสำคัญเช่นกัน

ฐิติพร อ้ายดี

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูป

หน้า

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของวิทยานิพนธ์

โรคหลอดเลือดสมองหรือ stroke [1] เกิดจากหลอดเลือดสมองตีบตันหรือเกิดเลือดออกในสมองนำไปสู่ภาวะสมองขาดเลือด ทุกปีจะมีผู้ป่วยโรคนี้ 16 ล้านคนทั่วโลก จัดเป็นโรคที่มีอัตราการเสียชีวิตสูง หากเข้ารับการรักษาได้ทันเวลาก็ต้องใช้ค่าใช้จ่ายมาก ไม่ว่าจะเป็นการเข้ารับการผ่าตัด การกายภาพบำบัดฟื้นฟูภายหลังเกิดอาการ ในช่วงไม่กี่ปีที่ผ่านได้มีการนำการเรียนรู้ของเครื่อง (Machine Learning) [2] มาช่วยวิเคราะห์ปัจจัยเสี่ยงก่อนที่จะเกิดอาการ [3] ซึ่งจะเป็นประโยชน์อย่างมากในทางการแพทย์

การเรียนรู้ของเครื่องเป็นวิทยาศาสตร์ด้านหนึ่งที่สนใจว่าคอมพิวเตอร์เรียนรู้อย่างไรจากกลุ่มข้อมูล เกี่ยวข้องกับการหาความสัมพันธ์ทางสถิติและคณิตศาสตร์ เพิ่มประสิทธิภาพในการทำงานด้วยการใช้ algorithms สร้างโมเดลขึ้นมาจากข้อมูลจำนวนมาก ประเภทของการเรียนรู้จากคอมพิวเตอร์แบ่งเป็น Supervised Learning และ Unsupervised Learning ซึ่งในงานวิจัยนี้มีเป้าหมายทำนายจาก output ที่เราสนใจ คือ การเป็นโรคหลอดเลือดสมอง จัดเป็นการเรียนรู้แบบ supervised [4]

ในขั้นตอนการทำงานจะใช้การจำแนกประเภท (classification) เช่น K-Neighbors Classifier (KNN), Support Vector Machine (SVM), Random Forest (RF), AdaBoost ร่วมด้วยกับเทคนิคต่าง ๆ ที่จะมาช่วยในการใช้การเรียนรู้ของเครื่อง[5] ทำให้ผลลัพธ์ของโมเดลมีความถูกต้องมีความแม่นยำมากขึ้น

จากชุดข้อมูลที่นำมาศึกษา [6] มีข้อมูลทั้งหมด 5,110 คนประกอบไปด้วยข้อมูลที่เป็นตัวเลขได้แก่ 1.อายุ 2.โรคความดันสูง 3.โรคหัวใจ มีค่า 0 กับ 1 ที่แทนการเป็นโรคและไม่เป็นโรคตามลำดับ 4.ระดับน้ำตาลเฉลี่ยในเลือด 5.ค่า BMI 6.โรคหลอดเลือดสมอง ซึ่งเป็นข้อมูลที่เราต้องการทำนาย ข้อมูลทางcategorical ได้แก่ 1.เพศ 2.อาชีพ 3.ที่อยู่อาศัย 4.การสมรส 5.การสูบบุหรี่ เป็นปัจจัยที่คาดว่าจะทำให้เกิดโรคหลอดเลือดสมอง นอกเหนือจากนี้ข้อมูลทางพันธุกรรม[7] การออกกำลังกายและสาเหตุอื่นที่คาดว่ามีผลกับการเป็นโรคหลอดเลือดสมองจะไม่ได้รวมอยู่ในการทำนาย ในงานวิทยานิพนธ์นี้จะนำมาประยุกต์ใช้ต่อกับบันทึกของศูนย์โรคหลอดเลือดสมองแบบครบวงจร โรงพยาบาลจุฬาลงกรณ์

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาตัวจำแนกประเภท (classifier) ในการเรียนรู้ของเครื่อง (machine learning) ที่มีประสิทธิภาพกับการทำนายโรคหลอดเลือดสมอง

1.3 ขอบเขตของการวิจัย

1) ศึกษาวิธีการจำแนกประเภท (classifier) แบบต่าง ๆ ที่เหมาะสมกับการจำแนกประเภท (classification)
2) นำผลการศึกษามาประยุกต์ใช้กับการทำนายผลโรคหลอดเลือดสมองโดยพิจารณาปัจจัยเฉพาะ (features)

1.4 ขั้นตอนการดำเนินงานของวิทยานิพนธ์

1) ศึกษาวิธีการเรียนรู้ของเครื่อง (Machine Learning) เพื่อจำแนกข้อมูลในลักษณะ Classification
2) ศึกษาและเลือกใช้ classifiers แบบต่าง ๆ และพัฒนาวิธีการนำมาประยุกต์ใช้กับการทำนายโรคหลอดเลือดสมอง
3) ทดสอบความแม่นยำในการทำนายผลการเป็นโรคหลอดเลือดสมอง
4) ปรับจูนและพิจารณา classifiers ที่มีผลกับการทำนายผลการเป็นโรคหลอดเลือดสมอง โดยใช้ข้อมูลทางการแพทย์ที่มีอยู่
5) ความเป็นไปได้ในการปรับเพิ่ม features หรือปัจจัยเฉพาะของข้อมูลที่มีอยู่โดยใช้ข้อมูล ที่ได้จากแบบข้อมูลที่บันทึกของศูนย์โรคหลอดเลือดสมองแบบครบวงจร โรงพยาบาลจุฬาลงกรณ์
6) สรุปผลและเขียนรูปเล่มวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่ได้รับ

1) ได้วิธีการเรียนรู้ของเครื่องที่เหมาะสมกับการจำแนกประเภท
2) ได้ตัวจำแนกประเภท (classifiers) ในการทำนายโรคหลอดเลือดสมอง

บทที่ 2
การสืบค้นวรรณกรรม

เพื่อเป็นแนวทางในการทำงานวิจัย ผู้วิจัยได้รวบรวมงานวิจัยต่าง ๆ ที่มีความเกี่ยวข้องและเป็นประโยชน์ในการนำไปใช้ทำงานวิจัยดังนี้

2.1 Analysis of main risk factors causing stroke in Shanxi Province based on machine learning models, 2021 [8]

ข้อมูลที่ใช้ศึกษาในงานวิจัยนี้มาจากประเทศจีน แบ่งเป็นแหล่งข้อมูลชุดแรกจากคนไข้ที่รักษาตัวในโรงพยาบาล 2,000 คน ช่วงปี 2018 และข้อมูลชุดที่ 2 จากฐานข้อมูลประชากร จำนวน 27,583 คน ในปี 2017 ถึง 2020 โมเดลที่ใช้ train และ test แบ่งเป็นอัตราส่วน 4:1 ผลลัพธ์ของปัจจัยเสี่ยงที่ก่อให้เกิดโรคหลอดเลือดสมองเรียงลำดับได้ดังนี้ Hypertension > BMI > Hyperlipidemia > Diabetes Mellitus > พฤติกรรมการสูบบุหรี่ จากทั้งหมด 177 ปัจจัยเสี่ยง ตัวจำแนกประเภทที่ใช้มี 3 แบบ คือ Random Forest, SVM และ MLP

ตารางที่ 1 ผลเปรียบเทียบของแต่ละโมเดล

Table 3
The compared performance in different models.

Models	Average precision	Average recall	Average accuracy
Random Forest (criterition: entropy, max depth of trees: 4, min sample splitting: 12)	0.8435(±0.0123)	0.8532(±0.0097)	0.8503(±0.0130)
SVM (kernel: rbf, penalty: l2,C = 1e-1)	0.7623(±0.0215)	0.7538(±0.0263)	0.7573(±0.0250)
MLP (hidden layer = (150,100,50,1))	0.8124(±0.0351)	0.8172(±0.0283)	0.8146(±0.0294)

Table 4
Result of random forest model.

	Precision	Recall	f1-score	Support
Low risk	0.8007(±0.0071)	0.9531(±0.0044)	0.8703(±0.0041)	1962
Mid risk	0.8213(±0.0082)	0.7850(±0.0118)	0.7901(±0.0064)	1367
High risk	0.9124(±0.0131)	0.7182(±0.0102)	0.8026(±0.0076)	1426
Accuracy			0.8400(±0.0100)	
Macro avg	0.8421(±0.0081)	0.8179(±0.011)	0.8271(±0.0034)	4755
Weighted avg	0.8311(±0.0095)	0.8400(±0.010)	0.8344(±0.0062)	4755

ตารางที่ 1 โมเดลที่ใช้ตัวจำแนกประเภท Random Forest ที่เซตค่าพารามิเตอร์ max depth of trees = 4 และ min sample splitting = 12 ให้ผลค่าเฉลี่ย precision สูงสุด ผลลัพธ์มีค่า 0.8435 ลำดับที่ 2 คือตัวจำแนกประเภท MLP ที่กำหนดค่า layer = (150,100,50,1) ผลลัพธ์มี

ค่า 0.8124 และลำดับที่ 3 คือตัวจำแนกประเภท SVM ที่เซตค่า kernel เป็นแบบ rbf, penalty แบบ l2 และ C = 1e-1 ผลลัพธ์มีค่า 0.7623

2.1.1 Model Interpretation

Model interpretation เป็นสิ่งสำคัญสำหรับการวิเคราะห์ข้อมูลทางการแพทย์ เช่น ค่า feature importance หรือเรียกอีกชื่อว่า Gini importance หรือ Mean Decrease (MDI) เป็นค่าเฉลี่ยของ node impurity ที่ลดลงในแต่ละตัวแปรและ weighted โดยความน่าจะเป็นของตัวแปรที่เข้าถึง node สำหรับ Random-Forest model กำหนดให้ response คือ Y และคำนวณค่าเฉลี่ยตัวแปรที่เกี่ยวข้องกับ feature X_i ด้วย N trees ได้เป็นสมการนี้:

$$\mathrm{Imp}(\mathrm{X}_i) = \frac{1}{N}\sum_{T=1}^{N}\sum_{j\in T: v(s_j)=X_m} p(j)\Delta i(s_j, \mathrm{j})$$

$\mathrm{Imp}(\mathrm{X}_i)$ คือ ค่า feature importance ของ feature X_i, $p(j)\Delta i(s_j, \mathrm{j})$ คือ น้ำหนัก purity ที่ลดลงของ feature X_i ใน nodes ทั้งหมดของ j, $p(j)$ คือ ความน่าจะเป็นของตัวอย่างที่ใช้ใน node

$$p(j) = \frac{N_j}{N} = \frac{\text{the amount of samples reaching the node j}}{\text{total amount of samples}}$$

และ $i(s_j, \mathrm{j})$ เป็น impurity measure ที่ node j ด้วยตัวแบ่งที่ node j ดังนั้นจะได้ว่า $v(s_j) = \mathrm{X}_m$ ที่ node j, splitting identifier คือ ตัวแปร X_m

ในกรณีของโมเดล Decision-Tree จะสามารถเขียน feature importance ได้ดังนี้

$$\mathrm{Imp}(\mathrm{X}_i) = \sum_{v(s_j)=X_m} p(j)\Delta i(s_j, \mathrm{j})$$

2.1.2 Permutation Importance

Permutation importance ใช้อธิบายว่า feature แต่ละตัวมีผลอย่างไรกับการ prediction ทั้งหมด เป็นการประมาณค่าความเปลี่ยนแปลงของ model prediction's accuracy โดยการสับเปลี่ยน feature importance ในแบบที่ i^{th} feature

$$Per_{\mathrm{Im}\,p}(i) = s - \frac{1}{K}\sum_{j=1}^{K} s_{i.j}$$

$Per_{\mathrm{Im}\,p}(i)$ คือ Permutation feature importance ของ feature ตัวที่ i^{th} , j^{th} ทำซ้ำ K ครั้ง สลับกันใน i^{th} feature, $s_{i.j}$ คือ model accuracy ที่ถูกดัดแปลงในชุดข้อมูล $\hat{D}_{i,j}$ ด้วย i^{th} feature ที่สลับกัน

2.1.3 SHAP (Shapley Additive explanations)

วิธีการ SHAP เป็นวิธีการที่แสดงความสำคัญของแต่ละ features และแต่ละ features มีความสัมพันธ์มากน้อยกับการทำนายผลลัพธ์หรือตัวแปรตาม

เป้าหมายคือการอธิบายการทำนายของตัวแปร x^i โดยการประมวลผลที่เกี่ยวข้องกันของแต่ละ feature กับโมเดลที่ใช้ทำนาย

$$\Phi_i = \sum_{S \subseteq F \setminus \{x^i\}} \frac{\left|S\right|!\ n - \left|S\right| - 1\ !}{n!}\left[v(S \cup \{x^i\}) - v(\mathrm{S})\right]$$

โดยที่ $S \subseteq \{x^1, x^2, ..., x^n\} \setminus x^i, F = \{x^1, x^2, ..., x^n\}.$

ในสมการ $F \setminus \{x^i\}$ คือ ซับเซตที่เป็นไปได้ทั้งหมดที่ไม่มี feature x^i, S แทน เซต sub-feature ที่ไม่รวมในผลลัพธ์, $v(S \cup \{x^i\})$ คือผลลัพธ์ (output) ของโมเดล เช่น precision, recall, accuracy และอื่น ๆ หลังจาก feature x^i เพิ่มเข้าไปในซับเซต S, $v(S)$ เป็นเอาท์พุตของโมเดลที่อยู่ในรูปซับเซต S เมื่อเปรียบเทียบกับ permutation feature importance แล้วค่า SHAP สามารถอธิบายความเกี่ยวข้องของ feature ในแต่ละตัวอย่างได้ดีกว่าทั้งทาง marginal positive และnegative แต่ละ features

2.2 Performance Analysis of Machine Learning Approaches in Stroke Prediction, 2020 [6]

งานวิจัยนี้ใช้แหล่งข้อมูลจากคลินิกประเทศบังกลาเทศจำนวน 5,110 คน เลือกใช้ตัวจำแนกประเภททั้งหมด 10 แบบ คือ Logistics Regression (LR), Stochastic, Gradient Descent (SGD), Decision Tree Classifier (DT), AdaBoost Classifier, Gaussian Classifier, Quadratic

Discriminant Analysis (QDA), Multi-layer Perceptron Classifier (MLP), K-Neighbors Classifier (KNN), Gradient Boosting Classifier (GBC) และ XGBoost Classifier

ตารางที่ 2 ผลของตัวจำแนกประเภทที่ใช้ในการทำนายโรคหลอดเลือดสมอง

CN	Accuracy	Class Label	Precision	Recall	F-1	AUC	FP Rate	FN Rate
LR	78%	**No Stroke**	0.83	0.82	0.83	0.76	30%	16%
		Stroke	0.70	0.71	0.71			
SGD	65%	**No Stroke**	0.68	0.95	0.80	0.73	48%	5%
		Stroke	0.76	0.26	0.39			
DTC	91%	**No Stroke**	0.94	0.92	0.93	0.80	12%	5%
		Stroke	0.87	0.90	0.89			
AdaBoost	94%	bNo Stroke	0.92	0.98	0.95	0.79	4%	5%
		Stroke	0.97	0.85	0.91			
Gaussian	78%	**No Stroke**	0.86	0.77	0.81	0.77	33%	12%
		Stroke	0.97	0.78	0.72			
QDA	79%	**No Stroke**	0.87	0.79	0.73	0.75	31%	11%
		Stroke	0.69	0.80	0.84			
MLP	79%	**No Stroke**	0.91	0.79	0.85	0.81	33%	9%
		Stroke	0.71	0.88	0.78			
KNeighbors	87%	**No Stroke**	0.97	0.83	0.89	0.81	25%	2%
		Stroke	0.77	0.96	0.95			
GBC	96%	**No Stroke**	0.93	0.99	0.96	0.85	0.1%	4%
		Stroke	0.99	0.87	0.93			
XGB	96%	**No Stroke**	0.94	0.99	0.97	0.90	0.5%	4%
		Stroke	0.99	0.89	0.94			
Weighted Voting	97%	**No Stroke**	0.93	1.00	0.97	0.93	0.9%	3%
		Stroke	1.00	0.90	0.95			

ตารางที่ 2 แสดงตัวจำแนกประเภทที่ให้ค่าความถูกต้อง (accuracy) สูงสุด คือ Weighted Voting มีค่า 97% ตามมาด้วยตัวจำแนกประเภท GBC และ XGB ที่ให้ผลลัพธ์การจำแนกประเภทมีค่า 96% ซึ่งมีค่าใกล้เคียงกัน ทั้งนี้ยังมีตัวบ่งชี้ความสามารถของตัวจำแนกประเภทได้แก่ precision, recall, F1-score, และ AUC

2.2.1 Area Under Curve (AUC)

ตัวบ่งชี้ความสามารถของตัวจำแนกประเภท ถ้า AUC = 1, คือผลที่เป็นบวกและลบถูกแยกได้โดยตัวจำแนกประเภทได้ถูกต้อง กรณี AUC = 0 ทั้งผลที่เป็นลบและบวกถูกจำแนกเป็นทางบวก และเมื่อ 0.5 < AUC < 1 จะหมายถึงการแยกความต่างของผลที่เป็นบวกจากผลทางลบ

2.3 Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry, 2020 [9]

งานวิจัยนี้นำข้อมูลในปี 2006 ถึงปี 2018 มีข้อมูลทั้งหมด 58,493 ข้อมูล แหล่งที่มาจาก Taiwan Stroke Registry (TSR) แบ่งประเภทของโรคหลอดเลือดสมองเป็น Ischemic และ Hemorrhagic ใช้ตัวจำแนกประเภท Support Vector Machine (SVM), Random Forest (RF), Artificial Neural Network (ANN) และ Hybrid Artificial neural Network (HANN) เพื่อใช้ในการเปรียบเทียบผลลัพธ์

2.3.1 Cross-Validation

Cross-Validation เป็นวิธีการที่ป้องกันการเกิด imbalance dataset งานวิจัยนี้เลือกใช้แบบ 10-fold cross-validation แบ่งเป็น training data 70% และ test data 30% ซึ่งตัวที่ใช้วัดประสิทธิภาพการทำงานของโมเดลมีดังนี้

ถ้าคำนึงถึงความแม่นยำของการทำนายค่า positive predictions เราสามารถคำนวณหาค่า precision ดังนี้

$$precision = \frac{TP}{TP + FP}$$

TP คือ จำนวนของ true positives หมายถึงเป็นโรคหลอดเลือดสมอง โมเดลสามารถทำนายได้ถูกต้องว่าเป็นโรคหลอดเลือดสมอง และ FP คือ จำนวนของ false positives หมายถึงไม่เป็นโรคหลอดเลือดสมอง แต่โมเดลทำนายว่าเป็นโรคหลอดเลือดสมอง

$$accuracy = \frac{TP + TN}{total\ population}$$

TN คือ จำนวนของ true negatives หมายถึงไม่เป็นโรคหลอดเลือดสมอง โมเดลสามารถทำนายได้ถูกต้องว่าไม่เป็นโรคหลอดเลือดสมอง สมการหาค่า accuracy ข้างต้นสามารถใช้คู่กับตัวชี้วัดที่ชื่อว่า recall ซึ่งสามารถเรียกอีกอย่างว่า sensitivity หรือ true positive rate (TCR) โดยที่ค่า recall สามารถคำนวณหาค่าได้ดังนี้

$$recall = \frac{TP}{TP + FN}$$

FN คือ จำนวนของ false negatives เป็นโรคหลอดเลือดสมอง แต่โมเดลทำนายว่าเป็นไม่โรคหลอดเลือดสมอง

นอกจากนี้ยังมีตัวชี้วัดการทำนายอีกตัวชื่อว่า F1-score ซึ่งสามารถคำนวณได้ดังนี้

$$F1\text{-}score = \frac{2 * precision * recall}{precision + recall}$$

2.3.2 Feature Selection

Feature Selection คือ วิธีการที่ช่วยในการเลือก input feature ที่เหมาะสมสำหรับวิธีการจำแนกประเภทแบบต่าง ๆ (classifier) เพื่อช่วยลดการเกิด overfitting

จากงานวิจัยในตัวอย่างนี้จะเลือกใช้ extremely randomized trees (extra-trees) algorithm โดยทำซ้ำหลายครั้ง เพื่อให้ได้ปัจจัยเฉพาะที่เหมาะสม โดย extra-trees algorithm ได้กำหนด Gini impurity = $1-\sum_{i=1}^{j} p_i^2$, j คือ จำนวน class ทั้งหมด, p_i เศษส่วนของ labels ด้วยคลาส i แต่ละครั้งที่ iteration กำหนดจาก $threshold_{\min} = \min(\sigma) + sd(\sigma)$, ค่า σ คือ เซตของ feature importance โดยที่ feature importance ที่มีค่าเป็น 0 จะถูกตัดออก โดยทำซ้ำกัน 10 รอบ (10-times hold-out)

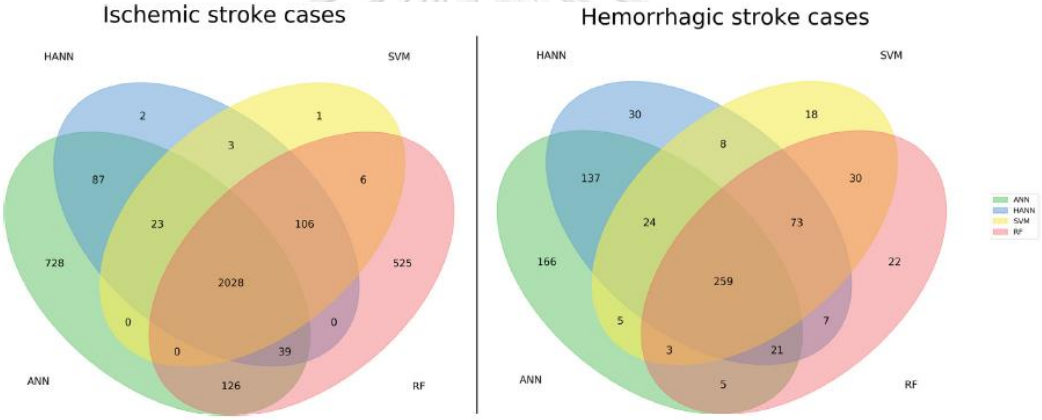


Fig. 6. Venn diagram of failed prediction cases from four machine learning models.

รูปที่ 1 เวนไดอาแกรมการทำนายที่ผิดพลาดจากโมเดล 4 แบบ

จากรูปที่ 1 เวนไดอาแกรมแสดงลักษณะของการทำนายที่ผิดพลาดจากโมเดล 4 แบบ คิดเป็น 55.2% ในข้อมูลประเภท Ischemic stroke ที่ตัวจำแนกประเภททั้ง 4 แบบทำนายผิดพลาดเหมือนกัน 2,028 กรณี จากข้อมูลการทำนายที่ผิดพลาดทั้งหมด 3,647 กรณี และคิดเป็น 32.1% ในข้อมูลประเภท Hemorrhagic stroke พบการทำนายผิดพลาดร่วมกัน 259 กรณี จากข้อมูลการทำนายที่ผิดพลาดทั้งหมด 808 กรณี สรุปคือตัวจำแนกประเภทที่ให้ประสิทธิภาพในการทำงานที่ดีที่สุดคือ SVM ทั้งในข้อมูลแบบ Ischemic stroke และแบบ Hemorrhagic stroke และตัวจำแนกประเภทที่ให้ผลแย่ที่สุดคือ ANN classifier

2.4 Interpretable Classifier Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model, 2015 [10]

งานวิจัยนี้ใช้ข้อมูลคนไข้จาก MarketScan Medicaid Multi-State Database (MDCD) แบ่งการจำแนกข้อมูลแสดงในรูปที่ 2 โดยโมเดลในการทำนายจะอยู่ในรูปของ sparse decision lists ประกอบไปด้วยชุดของ if...then... statements อธิบายเพิ่มเติมคือ if หมายถึงส่วนของชุดของมูล features และ then เกี่ยวข้องกับผลการทำนายของสิ่งที่สนใจ

if hemiplegia **and** age > 60 **then** *stroke risk* 58.9% (53.8%–63.8%)
else if cerebrovascular disorder **then** *stroke risk* 47.8% (44.8%–50.7%)
else if transient ischaemic attack **then** *stroke risk* 23.8% (19.5%–28.4%)
else if occlusion and stenosis of carotid artery without infarction **then** *stroke risk* 15.8% (12.2%–19.6%)
else if altered state of consciousness **and** age > 60 **then** *stroke risk* 16.0% (12.2%–20.2%)
else if age ≤ 70 **then** *stroke risk* 4.6% (3.9%–5.4%)
else *stroke risk* 8.7% (7.9%–9.6%)

รูปที่ 2 decision list ในการจำแนกกลุ่มข้อมูล

ข้อมูลในงานวิจัยนี้ใช้วิธีการ 5 folds cross-validation และมีจุดแบ่งของอายุ (split points) ตามที่แสดงในรูปที่ 2 ซึ่งจะแบ่งตัวแปรอายุหรือปัจจัยเฉพาะอายุเป็นตัวแปรแบบไบนารี คือ น้อยกว่าหรือมากในจุดแบ่งอายุที่ 60 ปี พิจารณาร่วมด้วยกับการใช้ยารักษาโรคและโรคประจำตัว เพศ คำอธิบาย hemiplegia เป็นอาการที่เป็นผลจากการเป็นโรคหลอดเลือดสมอง, cerebrovascular disorder บ่งชี้การเป็น prior stroke, และ transient ischaemic attack สามารถเรียกอีกอย่างว่า mini-strokes ข้อมูลทั้งหมดเหล่านี้เป็นประวัติการเป็นโรคหลอดเลือดสมองที่เป็นประโยชน์ในการศึกษา ทั้งนี้ยังได้มีแบ่งกลุ่มย่อยในกลุ่มคนไข้เพศชายและเพศหญิง

2.4.1 Bayesian Rule Lists (BRL)

BRL เหมาะสำหรับ multi-class classification เซตของ labels ที่เป็นไปได้คือ $1,\ldots,L$ ในการทำนายความเสี่ยงของโรค stroke มี 2 labels คือ เป็น stroke และ ไม่เป็น stroke กลุ่มข้อมูลที่ใช้เทรน คือ $(x_i, y_i)_{i=1}^{n}$ โดยที่ค่า $x_i \in \mathbb{R}^d$ เป็น features ของ observation i และ y_i เป็น labels, $y_i \in 1,\ldots,L$ กำหนด $\mathrm{x} = x_1,\ldots,x_n$ และ $\mathrm{y} = y_1,\ldots,y_n$

Bayesian association rules กฎความสัมพันธ์ของกฎ $a \longrightarrow b$ คือ ตัวแปรที่เกิดขึ้นก่อน a และตัวที่แปรที่เป็นผลที่เกิดขึ้น b สำหรับการ classification ตัวแปรที่เป็นตัวทำนายคือ y

$$a \longrightarrow y \sim Multinomial(\theta)$$

กำหนดให้ Multinomial คือ การแจกแจงแบบอเนกอนาม (Multinomial distribution) ซึ่งความน่าจะเป็นของอเนกนามนำไปสู่การแจกแจงก่อนการสังเกตว่า

$$\theta \mid \alpha \sim Dirichlet(\alpha)$$

กำหนดให้ Dirichlet คือ Dirichlet distribution, α คือ พารามิเตอร์ในการ distribution

ให้ observations (x, y) นำไปclassified โดยกฎนี้ กำหนด $N_{\cdot,l}$ เป็นจำนวนของ observations ที่มี label $y_i = l$ และ $N = N_{\cdot,1},...,N_{\cdot,L}$ จะได้การแจกแจงภายหลัง

$$\theta \mid x, y, \alpha \sim Dirichlet(\alpha + N)$$

วิธีการ BRL เป็นวิธีการที่เป็นประโยชน์ในการแสดงความเชื่อถือและสร้างความเข้าใจกันระหว่างวิธีการเรียนรู้ของเครื่องและบุคลากรทางการแพทย์ที่จะนำโมเดลไปใช้ เรียกได้ว่าเป็น Interpretable model โดยโมเดลจะแสดงหลักการทำงานและวิธีการตัดสินใจในการทำนายผลได้อย่างชัดเจน

บทที่ 3
การเรียนรู้ของเครื่อง

ทฤษฎีที่เกี่ยวข้องกับการเรียนรู้ของเครื่อง (Machine Learning) จะกล่าวถึงในแต่ละหัวข้อดังต่อไปนี้

3.1 Hypothesis equation

$$\hat{y} = h_\theta(x) = \theta_0 + \theta_1 x \tag{3.1}$$

รูปทั่วไปของสมการไฮเปอเทซิสแทนด้วย $h_\theta(x)$ ประกอบไปด้วยค่า θ_0 และ θ_1 เพื่อประมาณค่าเอาท์พุต $\hat{y}$, ฟังก์ชัน h_θ คือการส่งค่าอินพุตของข้อมูลหรือค่า x ไปยังค่าเอาท์พุต y

3.2 Cost Function

การวัดค่าความถูกต้องของฟังก์ชันไฮเปอเทซิสจะใช้ cost function เป็นผลจากค่าเฉลี่ยของไฮโปเทสิส เมื่อเทียบอินพุต x กับค่าเอาท์พุต y ที่ออกมาจริง

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 \tag{3.2}$$

ค่า $\overline{x}$ คือ ค่าเฉลี่ยของกำลังสอง $h_\theta(x_i) - y_i$ หรือผลต่างระหว่างค่าที่ต้องการทำนายกับค่าจริง ฟังก์ชันนี้สามารถเรียกอีกชื่อว่า ฟังก์ชั่น squatted error หรือ mean squared error

3.3 Logistic Regression

ในกระบวนการจำแนกประเภทค่าเอาท์พุตเวคเตอร์ y จะมีค่าแค่ 0 หรือ 1

$$y \in \{0,1\}$$

0 คือ คลาสที่มีผลเป็นลบ (negative class) และ 1 คือ คลาสที่มีผลเป็นบวก (positive class) จะเรียกว่าเป็นปัญหาการจำแนกประเภทไบนารี (Binary Classification Problem)

$h(\mathrm{x})$ คือ ความน่าจะเป็นที่ $\mathbf{x}$ เป็น 1

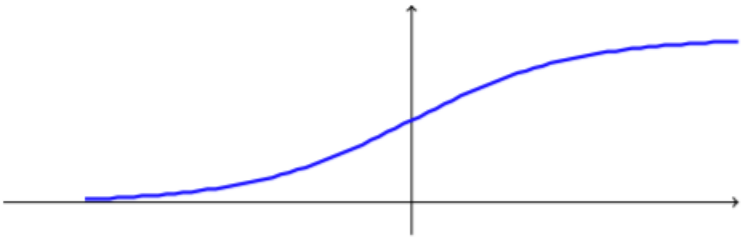
$$0 \leq h_\theta(\mathrm{x}) \leq 1 \tag{3.3}$$

อยู่ในรูปใหม่ที่เรียกว่า Sigmoid Function หรือเรียกว่า Logistic Function ชื่อ sigmoid หมายถึง รูปร่าง S รูปร่างของฟังก์ชันแสดงในรูปที่ 3

$$h_\theta(\mathrm{x}) = \mathrm{g}(\theta^T x) \tag{3.4}$$

$$z = \theta^T x \tag{3.5}$$

$$g(\mathrm{z}) = \frac{1}{1+e^{-z}} \tag{3.6}$$



รูปที่ 3 กราฟ Sigmoid Funtion [5]

3.3.1 Decision boundary

เป็นเส้นที่แบ่งพื้นที่ค่า $y = 0$ และ ค่า $y = 1$ ถูกสร้างจากฟังก์ชันไฮเปอเทซิส

$$\begin{aligned} h_\theta(x) \geq 0.5 \quad &\rightarrow y = 1 \\ h_\theta(x) < 0.5 \quad &\rightarrow y = 0 \end{aligned} \tag{3.7}$$

ฟังก์ชันไฮเปอเทซิสแบ่งเป็นการจำแนกประเภท 0 หรือ 1

$$g(\mathrm{z}) \geq 0.5 \ when \ z \geq 0 \tag{3.8}$$

อินพุตมีค่ามากกว่าหรือเท่ากับ 0 เอาท์พุตมีค่ามากกว่าหรือเท่ากับ 0.5

จาก

$$\begin{aligned} z = 0, e^{0} = 1 \quad &\rightarrow \quad g(z) = \frac{1}{2} \\ z \rightarrow \infty, e^{-\infty} \rightarrow 0 \quad &\rightarrow \quad g(z) = 1 \\ z \rightarrow -\infty, e^{\infty} \rightarrow \infty \quad &\rightarrow \quad g(z) = 0 \end{aligned} \tag{3.9}$$

เมื่อแทนค่า $\theta^T X$ ใน g จะได้

$$h_\theta(x) = g(\theta^T x) \geq 0.5 \quad when \;\; \theta^T x \geq 0 \tag{3.10}$$

อธิบายได้เป็น

$$\begin{aligned} \theta^T x \geq 0 \quad &\Rightarrow \quad y = 1 \\ \theta^T x < 0 \quad &\Rightarrow \quad y = 0 \end{aligned} \tag{3.11}$$

Cost function จะอยู่ในรูป

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) \tag{3.12}$$

$$Cost(h_\theta(x), y) = -\log(h_\theta(x)) \quad \text{if} \;\; y = 1 \tag{3.13}$$

$$Cost(h_\theta(x), y) = -\log(1 - h_\theta(x)) \quad \text{if} \;\; y = 0 \tag{3.14}$$

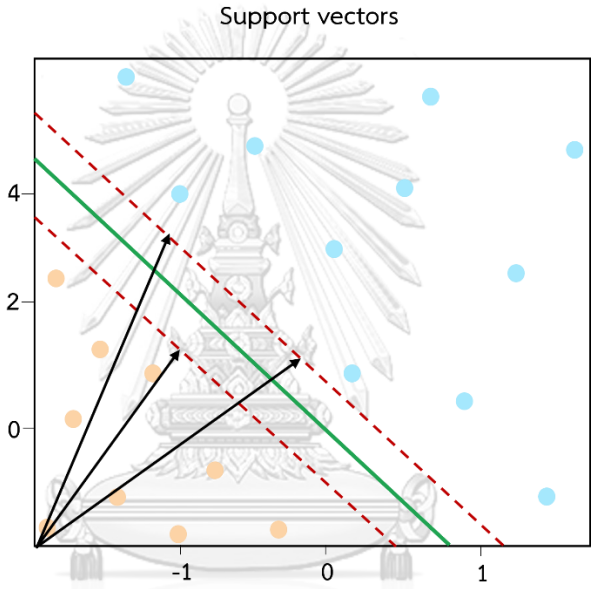
ยิ่ง hypothesis มีค่า y มากขึ้น ค่าของ cost function เอาท์พุตจะมากขึ้นตาม ถ้า hypothesis เท่ากับ y ค่า cost function จะเท่ากับ 0

$$\begin{aligned} Cost(h_\theta(x), y) = 0 \qquad &\text{if} \;\; h_\theta(x) = y \\ Cost(h_\theta(x), y) \rightarrow \infty \quad &\text{if} \;\; \mathrm{y} = 0 \;\; and \;\; h_\theta(x) \rightarrow 1 \\ Cost(h_\theta(x), y) \rightarrow \infty \quad &\text{if} \;\; \mathrm{y} = 1 \;\; and \;\; h_\theta(x) \rightarrow 0 \end{aligned} \tag{3.15}$$

กรณีที่คำตอบที่ถูกต้อง y คือ 0 ดังนั้น cost function จะมีค่า 0 ถ้าฟังก์ชันไฮเปอเทซิสมีเอาท์พุตเป็น 0 เช่นกัน แต่ถ้าฟังก์ชันไฮเปอเทซิสเข้าใกล้ 1 ดังนั้น cost function จะเข้าใกล้ ∞

3.4 Support Vector Machine (SVM)

ตัวจำแนกประเภทแบบ SVM ใช้หน่วยความจำน้อยมาก จึงใช้เวลาใน prediction หลัง train โมเดลน้อย เหมาะสำหรับข้อมูลที่มีหลายมิติ เมื่อใช้ร่วมกับ kernel methods ทำให้มีประโยชน์หลากหลาย และปรับตัวได้ดีกับข้อมูลหลายประเภท



รูปที่ 4 ภาพแสดง Support Vectors

จากในรูปที่ 4 เส้นเขียว หรือ the decision boundary แบ่งข้อมูลเป็น 2 คลาส SVM ใช้ margin เป็นเส้นคู่ขนาน 2 เส้น เห็นได้จากเส้นสีม่วงประที่อยู่ทั้ง 2 ฝั่งของขอบเขตในรูป เวคเตอร์จากจุด (0,0) ไปยังจุดต่างๆ เรียกว่า support vectors ซึ่งเป็นที่มาของชื่อตัวจำแนกประเภท

cost function ของ SVM เขียนได้ดังนี้

$$J\ \theta\ = \frac{1}{m}\sum_{i=1}^{m} y^{(i)} \cos t_1\ \theta^T x^{(i)}\ +\ 1-y^{(i)}\ \cos t_0\ \theta^T x^{(i)}\ + \frac{\lambda}{2m}\sum_{j=1}^{n} \Theta_j^2 \qquad (3.16)$$

คูณ m ตลอดสมการ เพื่อกำจัด m factor ใน denominators แต่จะไม่มีผลในส่วนของ optimization และใช้ factor C แทน λ จะได้ว่า

$$J\ \theta\ = C\sum_{i=1}^{m} y^{(i)} \cos t_1\ \theta^T x^{(i)}\ +\ 1-y^{(i)}\ \cos t_0\ \theta^T x^{(i)}\ + \frac{1}{2}\sum_{j=1}^{n} \Theta_j^2 \tag{3.17}$$

จาก $C = \frac{1}{\lambda}$ หากต้อง regularize มากขึ้น จะต้องลดการ overfitting โดยลดค่า C หากเป็นกรณีต้องการลด regularize จะต้องลดการ underfitting โดยการเพิ่มค่า C

$$h_\theta(x) = \begin{cases} 1 & if\ \Theta^T x \geq 0 \\ 0 & otherwise \end{cases} \tag{3.18}$$

เรียกว่าเป็น discriminant function ค่าเอาท์พุต เป็น 1 หรือ 0

3.4.1 Large Margin Intuition

$$\begin{aligned} &\text{ถ้า } y = 1,\ \Theta^T x \geq 1 \\ &\text{ถ้า } y = 0\ ,\ \Theta^T x \leq -1 \end{aligned} \tag{3.19}$$

เมื่อ c มีค่ามาก

$$\sum_{i=1}^{m} y^{(i)} \cos t_1\ \theta^T x^{(i)}\ +\ 1-y^{(i)}\ \cos t_0\ \theta^T x^{(i)}\ = 0 \tag{3.20}$$

แทนค่าใน cost function จะกลายเป็น

$$\begin{aligned} J(\theta) &= C \cdot 0 + \frac{1}{2}\sum_{j=1}^{n} \Theta_j^2 \\ J(\theta) &= \frac{1}{2}\sum_{j=1}^{n} \Theta_j^2 \end{aligned} \tag{3.21}$$

ระยะของ decision boundary ที่อยู่ใกล้ตัวอย่างเรียก margin ซึ่งหลักการของ SVM จะพยายามให้มีค่ามากที่สุด

3.4.2 Vector Inner Product

มีเวคเตอร์ u และ v จากพิกัด (0,0) ไปยัง (v_1, v_2) ความยาวเวคเตอร์ $v = \|v\| = \sqrt{{v_1}^2 + {v_2}^2}$

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{3.22}$$

ρ = ความยาว projection ของเวคเตอร์ v บนเวคเตอร์ u

$$u^T v = p \cdot || u || \tag{3.23}$$

จาก $u^T v = || u || \cdot || v || \cos\theta$, θ มุมระหว่าง u และ v เมื่อแทน $p = || v || \cos\theta$ จะได้ว่าสมการข้างต้น

$$u^T v = v^T u \tag{3.24}$$

เนื่องจากเวคเตอร์ u และ v มีความยาวเท่ากัน และแทนค่า $u^T v = p \cdot || u ||$

$$u^T v = u_1 v_1 + u_2 v_2 \tag{3.25}$$

กรณีที่มุมระหว่าง u และ v มากกว่า 90° ดังนั้น projection ρ จะเป็นลบ

$$\begin{aligned}\min_{\Theta}\sum_{j=1}^{n}\Theta_j^2 &= \frac{1}{2}(\Theta_1^2+\Theta_2^2+\cdots+\Theta_n^2)\\ &= \frac{1}{2}(\sqrt{\Theta_1^2+\Theta_2^2+\cdots+\Theta_n^2})^2\\ &= \frac{1}{2}\,||\,\Theta\,||^2\end{aligned} \tag{3.26}$$

จากกฎเดียวกัน เราจะได้ optimization objective

$$\begin{aligned}\Theta^T x^{(\mathrm{i})} = p^{(\mathrm{i})}\cdot||\,\Theta\,|| &= \Theta_1 x_1^{(\mathrm{i})}+\Theta_2 x_2^{(\mathrm{i})}+\cdots+\Theta_n x_n^{(\mathrm{i})}\\ \text{กรณี } y=1,\ p^{(\mathrm{i})}\cdot||\,\Theta\,|| &\geq 1\\ \text{กรณี } y=0,\ p^{(\mathrm{i})}\cdot||\,\Theta\,|| &\leq 1\end{aligned} \tag{3.27}$$

เหตุผลที่ทำให้ margin มีค่ามาก เพราะเวคเตอ Θ ตั้งฉากกับ decision boundary เพื่อที่จะได้ optimization objective เป็นจริง เราจึงต้องให้ค่า projection $p^{(i)}$ มีค่ามากที่สุดเท่าที่จะเป็นไปได้

3.5 K-Nearest Neighbors (KNN)

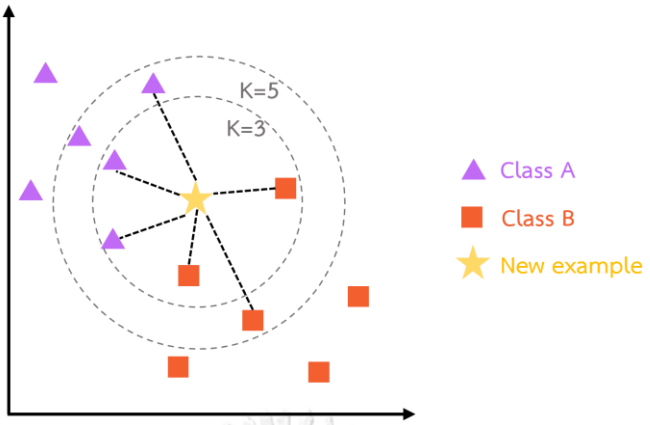
ตัวจำแนกประเภท KNN จะอธิบายหลักการทำงานโดยให้โดเมนแทนด้วย X เขียนในรูปฟังก์ชันเป็น $\rho : \mathrm{X}\times\mathrm{X}\rightarrow\mathbb{R}^d$ ที่ให้ค่าระยะทางของ 2 สมาชิกใด ๆ ใน X เช่น ถ้า $\mathrm{X}=\mathbb{R}^d$ ดังนั้น ρ ที่อยู่ในรูประยะ Euclidean

$$\rho(\mathrm{x},\mathrm{x}') = \left\|\mathrm{x}\text{ - }\mathrm{x}'\right\| = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2} \tag{3.28}$$

ให้ $S=(\mathrm{x}_1,y_1),...,(\mathrm{x}_m,y_m)$ เป็นตัวอย่างของข้อมูลที่ให้ training แต่ละ $\mathrm{x}\in\mathrm{X}$, กำหนด $\pi_1(\mathrm{x}),...,\pi_m(\mathrm{x})$ เรียงลำดับของ $\{1,...,m\}$ ตามระยะถึง x, $\rho(\mathrm{x},\mathrm{x}_i)$ สำหรับทั้งหมด $i<m$,

$$\rho(\mathrm{x},\mathrm{x}_{\pi_i(\mathrm{x})}) \leq \rho(\mathrm{x},\mathrm{x}_{\pi_{i+1}(\mathrm{x})})$$

สำหรับจำนวน k , กฎ k-NN สำหรับการจำแนกไบนารีสามารถนิยามได้ตามข้างต้น



รูปที่ 5 ภาพอธิบาย KNN

รูปที่ 5 ภาพอธิบาย KNN เริ่มจากมีข้อมูลที่ต้องการจำแนกประเภทวัดระยะทางไปยัง class A และ class B กำหนดค่า K ที่ต่างกัน เพื่อหาเพื่อนบ้าน (neighbor) [11] โดยทั่วไปทาง data science [12] นิยมเลือกค่า K เป็นเลขคี่ เมื่อจำนวนคลาสเป็นเลขคู่ ค่าที่เลือกใช้นั้นขึ้นอยู่กับข้อมูลแต่ละชุด ต้องทดลองกับโมเดลและดูผลลัพธ์ที่ได้ จำนวนของ k neighbors ใน KNN เป็น hyperparameter ที่กำหนดผลของการทำนายโมเดล ยิ่งมีค่ามาก ยิ่งต้องใช้การประมวลผลมากตาม แต่ถ้ามีค่าน้อย noise จะส่งผลกับผลลัพธ์

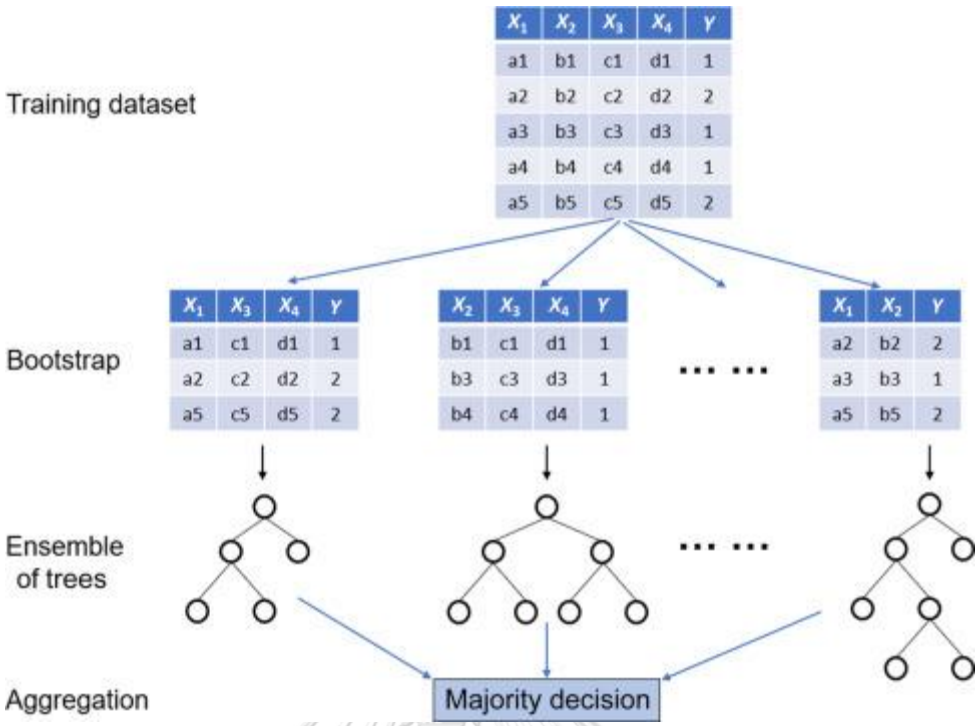
3.6 Ensemble Learning

Ensemble learning เป็นการรวมกลุ่มของตัวจำแนกประเภทมาพัฒนาให้ได้ผลลัพธ์ที่ดีขึ้นของโมเดล โดยที่ตัวจำแนกประเภทแต่ละตัวนั้นควรมีความหลากหลายและเป็นอิสระต่อกัน [13] เพื่อลดข้อผิดพลาดจากตัวจำแนกประเภทเพียงตัวเดียว ในที่นี้จะยกตัวอย่างเทคนิคที่ได้นำมาใช้ดังนี้

3.6.1 Random Forest [14]

ตัวจำแนกประเภท Random Forest (RF) เป็นการรวมของตัวจำแนกประเภทต้นไม้ (Decision Trees) กล่าวคือเป็นการ optimized ของตัวจำแนกประเภทต้นไม้ ซึ่งตัวจำแนกประเภทต้นไม้จะรวมการตัดสินใจของต้นไม้แต่ละแบบ

กำหนดให้ algorithm A และ การกระจายผ่าน θ เริ่มจากสุ่มบางตัวอย่างจาก S ตั้งชื่อตัวอย่างใหม่ที่ใช้เทรนว่าเซต S' มีขนาด m' ใช้การกระจายแบบสม่ำเสมอบน S จากนั้นสร้างลำดับ $I_1, I_2, \ldots,$ โดยที่แต่ละ I_t เป็นสับเซตของ $[d]$ ขนาด k จะได้ตัวอย่างการกระจายแบบสม่ำเสมอบนตัวแปรสุ่มจาก $[d]$ การสุ่มทั้งหมดของตัวแปรจากเวคเตอร์ θ



รูปที่ 6 แผนภาพอธิบายการทำงานของ Random Forest [15]

3.6.2 Adaboost

ตัวจำแนกประเภท Adaboost หรือชื่อเต็มของวิธีการนี้เรียกว่า Adaptive Boosting จะใช้วิธีการ booting คือการรวมตัวจำแนกประเภทที่ให้ผลลัพธ์ที่แย่ (weak learners) ให้เป็นตัวจำแนกประเภทที่ให้ผลลัพนธ์ที่ดีขึ้น (strong learners) โดยเทรนตามลำดับ โดยทีแต่ละครั้งจะพัฒนาให้ดีขึ้นกว่าการทำในครั้งก่อน ใช้ weak learner และหา hypothesis ที่ทำให้ค่า error ต่ำลง จากการกระจายและฟังก์ชั่นเป้าหมายที่ต้องการ (Empirical Risk Minimization)

กำหนดให้รับค่ากลุ่มตัวอย่างที่ใช้เทรน $S = (\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_m, y_m)$, โดยที่แต่ละค่า $i, y_i = f(\mathrm{x}_i)$

สำหรับฟังก์ชัน labeling กระบวนการ boosting จะเริ่มเป็นรอบตามลำดับ ที่รอบ t, ตัว booster ตัวแรกจะกระจายตัวอย่างใน S แสดงในรูป $D^{(\mathrm{t})}$ ซึ่ง $D^{(\mathrm{t})} \in \mathbb{R}_+^m$ และ $\sum_{i=1}^m D_i^{(\mathrm{t})} = 1$ ดังนั้นตัว booster จะส่งผ่านการกระจาย $D^{(\mathrm{t})}$ และตัวอย่าง S ไปสู่ weak learner มีสมมุติฐานว่า weaker learner จะให้ค่า weak hypothesis, h_t โดย AdaBoost จะให้ค่า weight สำหรับ h_t

$$w_t = \frac{1}{2}\log\left(\frac{1}{\varepsilon_t} - 1\right) \tag{3.29}$$

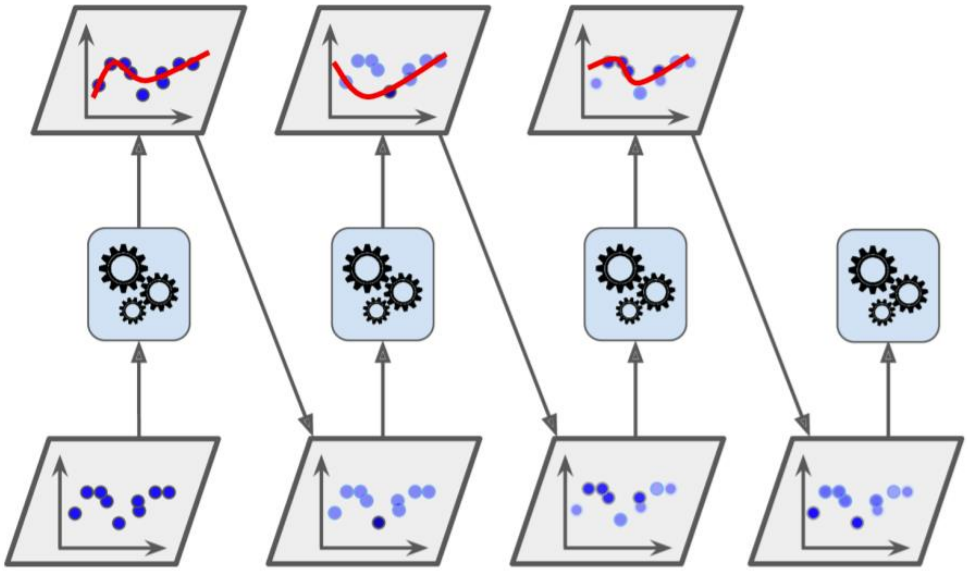
น้ำหนักของ h_t เป็นสัดส่วนผกผันกับ error ของ h_t เมื่อจบแต่ละรอบ AdaBoost จะใช้ค่าใหม่ในรอบถัดไป ทำให้ weak learner จะเน้นที่การแก้ปัญหาที่เกิดขึ้นในรอบที่แล้วและปรับแก้ในรอบใหม่ ผลลัพธ์ของอัลกอลิธึม AdaBoost จึงเป็น strong classifier ที่มาจากผลรวมน้ำหนักของ weak hypotheses

$$update\ D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(x_i))}{\sum_{j=1}^{m} D_i^{(t)} \exp(-w_t y_j h_t(x_j))} \tag{3.30}$$

เมื่อค่าทั้งหมด $i = 1,\ldots,m$

output the hypothesis

$$h_s(x) = \text{sign} \sum_{t=1}^{T} w_t h_t(x) \tag{3.31}$$



รูปที่ 7 ภาพประกอบการทำงานของ AdaBoost

3.7 Evaluating a hypothesis

ฟังก์ชัน hypothesis อาจมีค่า error ที่น้อยสำหรับตัวอย่างที่ใช้เทรน แต่ไม่ถูกต้อง เพราะ overfitting เราจึงควรแบ่งชุดข้อมูลเป็น 2 กลุ่ม คือ training set และ test set แล้วทำการคำนวณ error ของ test set

ค่าเฉลี่ยของ error สำหรับ test set

$$Test\ \ Error = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h\Theta(x_{test}^{(i)}), y_{test}^{(i)}) \tag{3.32}$$

3.8 Bias and Variance

จากการทำ cross validation จะแบ่งชุดข้อมูลเป็น 3 ชุด เช่น training set= 60%,cross validation set=20% และ Test set=20% เราจะคำนวณค่า error จาก 3 เซตที่ต่างกัน training error มีแนวโน้มลดลงเมื่อเราเพิ่มดีกรี d ของ polynomial ขณะเดียวกัน cross validation error ก็จะลดลงเมื่อเราเพิ่มดีกรี d กำหนดให้ Θ คือ พารามิเตอร์

High bias (underfitting) คือ ทั้ง $J_{train}(\Theta)$ และ $J_{CV}(\Theta)$ มีค่าสูง ดังนั้น $J_{CV}(\Theta) \approx J_{train}(\Theta)$

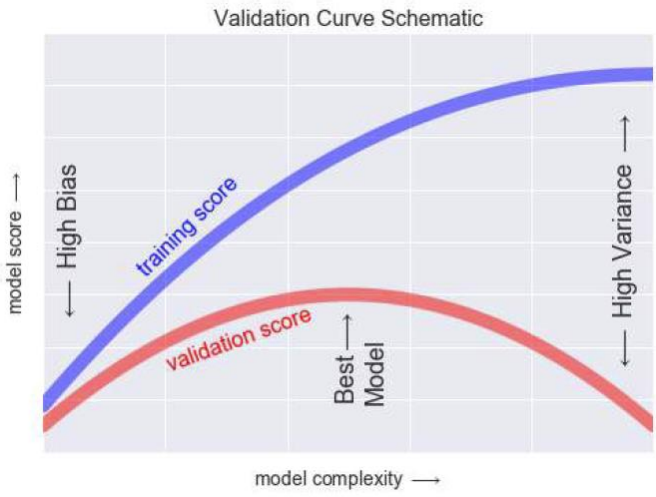
Hight variance (overfitting) คือ $J_{train}(\Theta)$ จะมีค่าต่ำและ $J_{CV}(\Theta)$ มีค่าสูงมากกว่า $J_{train}(\Theta)$

regularization parameter λ

เมื่อ λ ต่ำ $J_{train}(\Theta)$ มีค่าน้อย และ $J_{CV}(\Theta)$ มีค่าสูง

เมื่อ λ มีค่าปานกลาง ทั้ง $J_{train}(\Theta)$ และ $J_{CV}(\Theta)$ มีค่าต่ำ และ $J_{CV}(\Theta) = J_{train}(\Theta)$

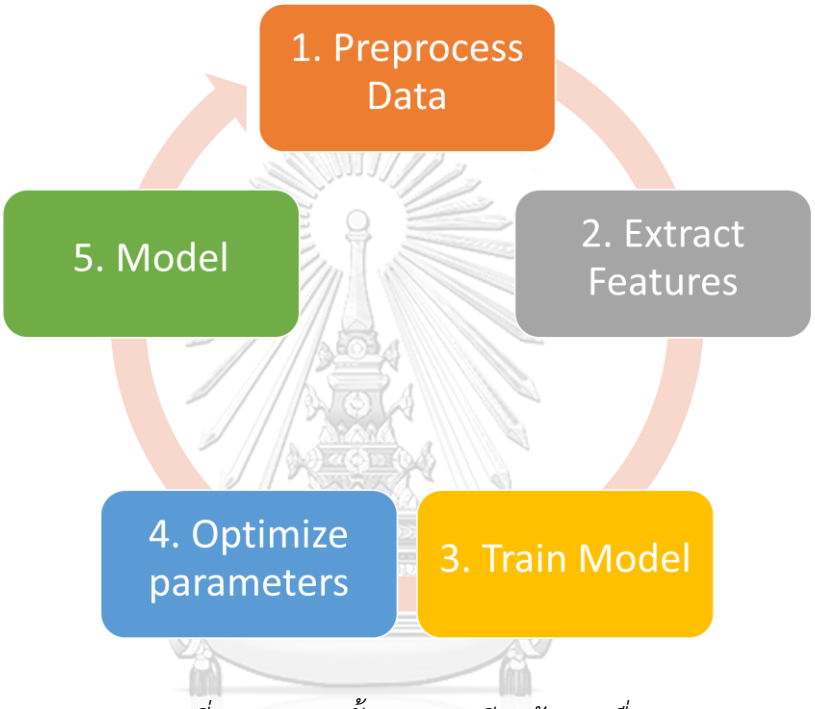
เมื่อ λ มาก $J_{train}(\Theta)$ และ $J_{CV}(\Theta)$ มีค่าสูง



รูปที่ 8 กราฟแสดงความสัมพันธ์ของ Bias และ Variance

บทที่ 4
วิธีการดำเนินงานวิจัย

ผู้วิจัยได้ศึกษาวิธีการทำงานการเรียนรู้ของเครื่องและตัวจำแนกประเภทจากการสืบค้นวรรณกรรมในบทที่ 2 และ 3 บทนี้จะเป็นขั้นตอนการดำเนินงานที่นำไปใช้ซึ่งแสดงแผนภาพขั้นตอนอย่างง่ายในรูปที่ 9

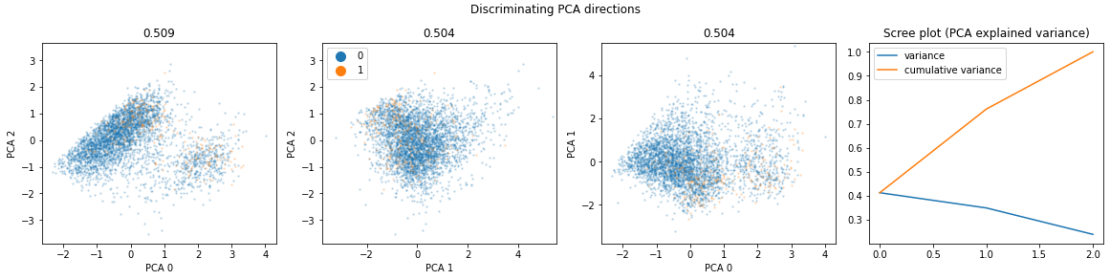


รูปที่ 9 แผนภาพขั้นตอนการเรียนรู้ของเครื่อง

4.1 ลักษณะข้อมูลที่ใช้

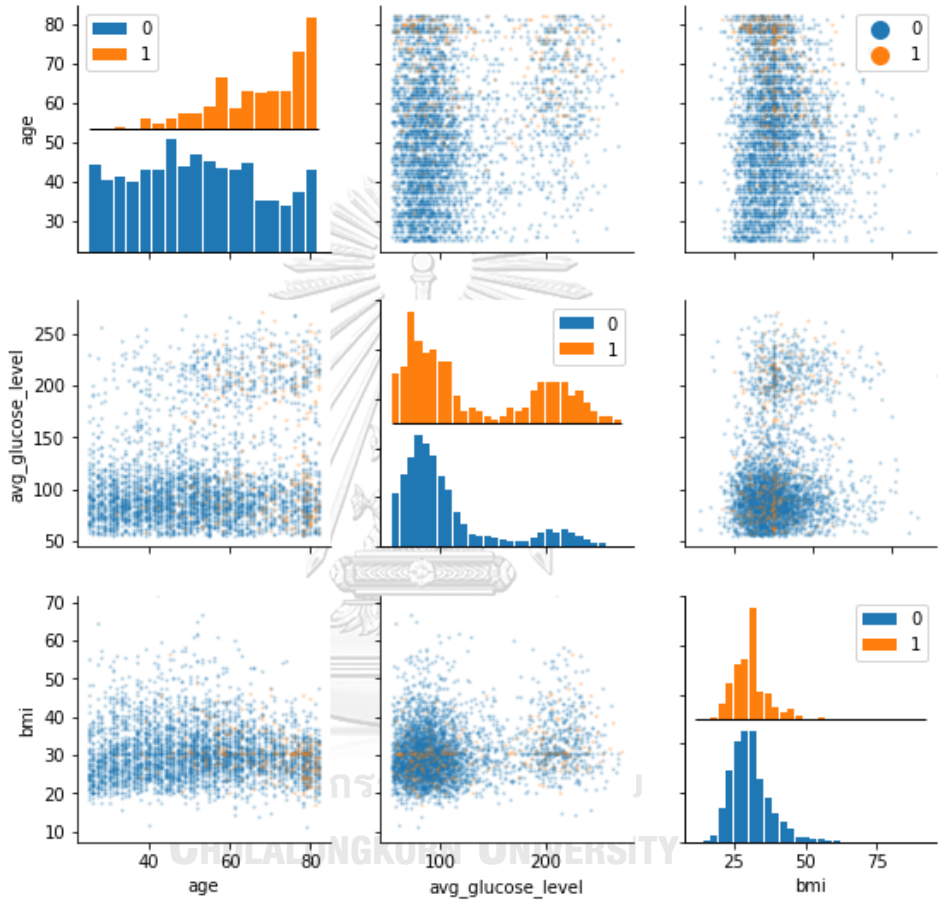
4.1.1 ข้อมูลชุดที่ 1

จำนวนเริ่มต้นมีทั้งหมด 5,110 คน แหล่งที่มาจาก [6] จำแนกออกเป็น 10 ปัจจัยเฉพาะ แสดงการจัดกลุ่มของข้อมูล (PCA) ในรูปที่ 10 และจำแนกประกอบข้อมูลได้ดังต่อไปนี้



รูปที่ 10 Principle Component Analysis ของข้อมูลชุดทดลอง

- ข้อมูลแบบตัวเลข: มี 3 ข้อมูล คือ อายุ, ระดับน้ำตาลเฉลี่ยในเลือดและค่า BMI แสดงในรูปที่ 11 และข้อมูลแบบ binary ที่เป็นตัวเลขมี 2 ข้อมูล คือ โรคความดันสูงและโรคหัวใจ มีค่า 0 กับ 1 แทนการเป็นและไม่เป็นโรคตามลำดับ โดยมีข้อมูลโรคหลอดเลือดสมองเป็นเป้าหมายที่เราต้องการทำนาย ลักษณะข้อมูลเป็นแบบ binary เช่นเดียวกัน



รูปที่ 11 กราฟแสดงการกระจายตัวของปัจจัยเฉพาะที่เป็นตัวเลข

- ข้อมูลแบบจำแนกประเภท (categorical data) 5 ข้อมูล ได้แก่

1. เพศ 2. อาชีพ 3. ที่อยู่อาศัย 4. การสมรส 5. การสูบบุหรี่

4.1.2 ข้อมูลชุดที่ 2

แหล่งที่มาจากศูนย์โรคหลอดเลือดสมองแบบครบวงจร โรงพยาบาลจุฬาลงกรณ์ จำนวนเริ่มต้นจากบันทึกข้อมูลมีทั้งหมด 99 คน ซึ่งเป็นผู้ป่วยโรคหลอดเลือดสมองแบบตีบหรืออุดตัน

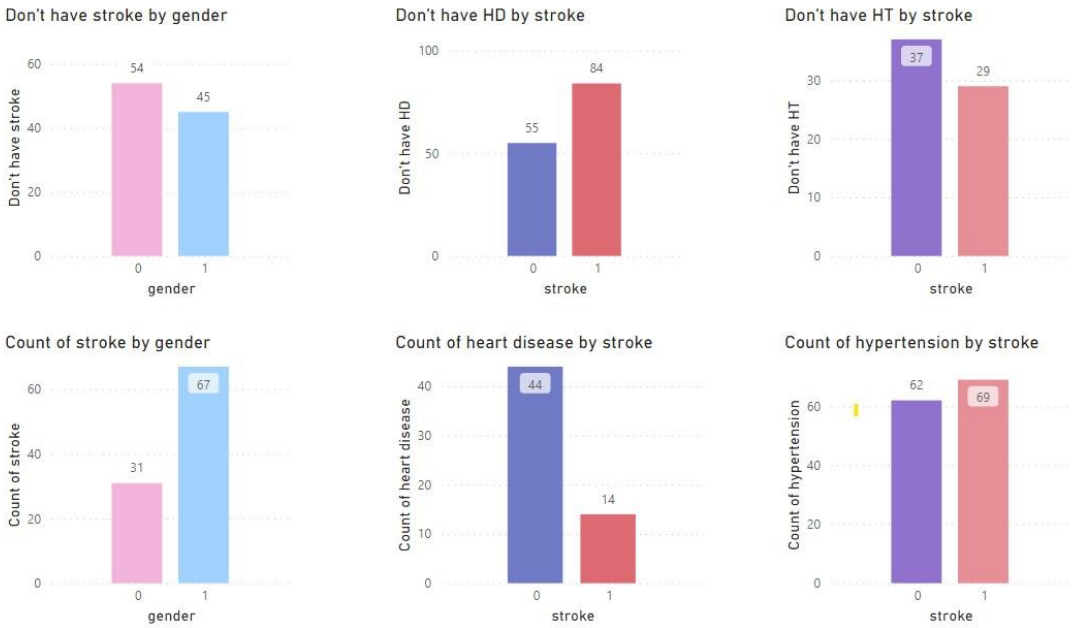
(Ischemic Stroke) แบบบันทึกมีข้อมูลเกี่ยวกับโรคประจำตัวที่ผู้ป่วยเป็น ตัวอย่างข้อมูลแสดงเพิ่มเติมในตารางที่ 3 เช่น โรคเบาหวาน (DM), โรคความดันสูง(HT), โรคหัวใจ(HD) เป็นต้น

ตารางที่ 3 คำอธิบายตัวอย่างข้อมูลในข้อมูลชุดที่ 2

ตัวย่อ	คำอธิบาย
BMI	Body Mass Index
DM	Diabetes Mellitus
DLP	Dyslipidemia
HT	Hypertension
AF	Atrial Fibrillation
Coronary HD	Coronary Heart Disease
MR	Mitral Regurgitation
TOAST	Trial of Org 10172 in Acute Stroke Treatment

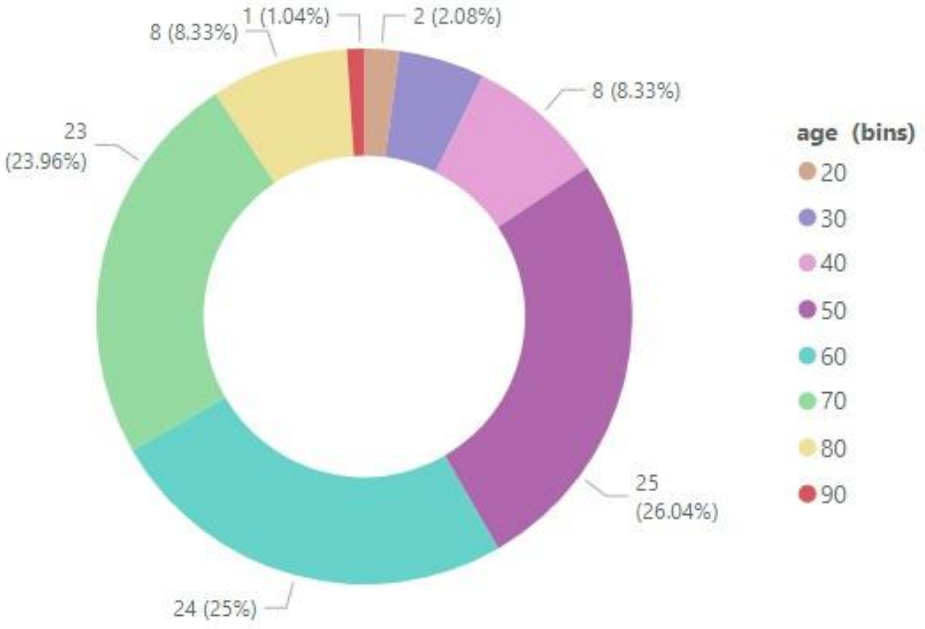
นอกเหนือจากนี้ยังมีบันทึกข้อมูลอื่น ๆ ที่ไม่เกี่ยวข้อง เช่น เวลาการเข้ามาตรวจเช็ค, คะแนนการประเมิน TOAST, ยาที่รับประทาน, โรคที่พบเป็นส่วนน้อยในกลุ่มข้อมูล จะไม่ได้นำมาใช้ในงานวิจัยนี้ จำนวนข้อมูลแต่ละประเภทที่นำมาใช้ในวิเคราะห์การทำนายโรคหลอดเลือดสมองแสดงในรูปที่ 12 จำนวนคนที่เป็นโรคมีทั้งหมด 97 คนหลังจากการตรวจเช็คข้อมูล รวมแล้วมีจำนวนข้อมูลคนที่เป็นและไม่เป็นโรคหลอดเลือดสมอง 197 คนที่จะนำมาไปใช้เทรนโมเดล

คำอธิบายเพิ่มเติมในรูปที่ 12 เพศหญิงจะแทนด้วยตัวเลข 0 และเพศชายจะแทนด้วยเลข 1 และโรค HD กับ HT จะแทนการไม่เป็นและเป็นโรคด้วยเลข 0 และ 1 ตามลำดับ



รูปที่ 12 แผนภูมิแท่งแสดงรายละเอียดจำนวนจากข้อมูลโรงพยาบาลจุฬาลงกรณ์

แต่ละช่วงของอายุจะแบ่งเป็นช่วงละ 10 ปี เช่น 20 ปี คือ ช่วงอายุ 20-29 ปี ไล่ไปทุกช่วงอายุ ซึ่งจะเห็นได้จากรูปที่ 13 ว่าช่วงอายุ 50-59 ปี เป็นช่วงอายุที่มีคนเป็นโรคหลอดเลือดสมองมากที่สุดจำนวน 25 คน คิดเป็น 26.04% จากจำนวนคนที่เป็นทั้งหมด 97 คน



รูปที่ 13 จำนวนคนที่เป็นโรคหลอดเลือดสมองจากข้อมูลโรงพยาบาลจุฬาลงกรณ์

4.2 การเตรียมการข้อมูล

เป็นขั้นสำคัญก่อนการนำข้อมูลไปใช้ในการเรียนรู้ของเครื่อง ข้อมูลแบบจำแนกประเภทจะถูกแปลงให้เป็นตัวเลขโดยใช้ sci-kit learn [16] ข้อมูลชุดที่ 1 จะอยู่ในรูปดังต่อไปนี้

1. อายุ
2. โรคความดันสูง (HT) : ไม่เป็นโรค = 0, เป็นโรค = 1
3. โรคหัวใจ (HD) : ไม่เป็นโรค = 0, เป็นโรค = 1
4. ระดับน้ำตาลเฉลี่ยในเลือด
5. ค่า BMI
6. เพศ : ผู้หญิง = 0, ผู้ชาย = 1
7. อาชีพ : ทำงานรัฐ = 0, ไม่ได้ทำงาน = 1, ทำงานเอกชน = 2, ธุรกิจส่วนตัว = 3, นักเรียน = 4
8. การสมรส : โสด = 0, แต่งงาน = 1
9. การสูบบุหรี่ : ไม่ทราบ - สูบประจำ = 0 - 4

ตัวอย่างข้อมูลแสดงในรูปที่ 14 ถึง รูปที่ 16

age	gender	HT	HD
50	Male	1	0
82	Female	0	0
45	Male	0	1
...	...	...	...
72	Male	0	1
64	Female	1	1

รูปที่ 14 ตัวอย่างชุดข้อมูลที่ใช้ในการจำแนกประเภท

Stroke
1
1
0
...
0
1

รูปที่ 15 ตัวอย่างข้อมูลที่ใช้ในการทำนายผลจากข้อมูลชุดที่ 1

TOAST
large-artery atherosclerosis
cardioembolism
small-vessel occlusion
...
stroke of other determined etiology
stroke of other undetermined etiology

รูปที่ 16 ตัวอย่างข้อมูลที่จะนำไปใช้ในการทำนายผลจากโรงพยาบาลจุฬาลงกรณ์

ข้อมูลจากโรงพยาบาลจุฬาลงกรณ์เป็นข้อมูลผู้ป่วยโรคหลอดเลือดสมองแบบ Ischemic ดังที่แสดงตัวอย่างข้อมูลในรูปที่ 16 เพื่อความสะดวกในการวิจัยผู้จัดทำจึงได้เปลี่ยนข้อมูล TOAST หรือการจำแนกประเภทของโรคหลอดเลือดสมองแบบตีบตันทุกประเภทแทนด้วยเลข 1 ความหมายเดียวกับข้อมูลชุดทดลองที่แสดงในรูปที่ 14 โดยที่ 1 หมายถึงการเป็น stroke และ 0 หมายถึงการไม่เป็น stroke

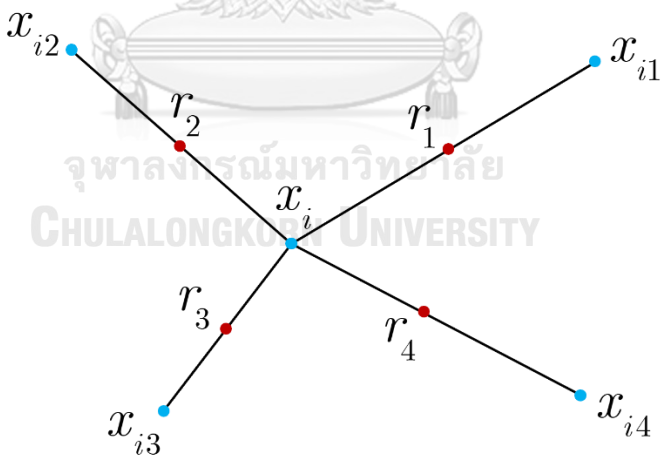
4.3 การคัดเลือกข้อมูลที่นำไปใช้

ขั้นตอนนี้จะเป็นการคัดเลือกข้อมูล เนื่องจากมีข้อมูลที่ซ้ำและมีปัจจัยเฉพาะบางตัวที่ไม่มีความสัมพันธ์กับผลลัพธ์ที่เราต้องการในการทำนายโรคหลอดเลือดสมอง คือ ที่อยู่อาศัยและหมายเลขประจำตัวในข้อมูลชุดทดลอง รวมแล้วจะเหลือข้อมูลเพียง 4,800 คนในข้อมูลชุดแรก

ข้อมูลชุดที่ 2 ได้มีการคัดปัจจัยเฉพาะบางส่วนออกจากที่กล่าวไว้ในหัวข้อ 4.1.2 ทั้งนี้ได้ตรวจสอบพบว่ามีข้อมูล 2 คนจากแบบบันทึกศูนย์โรคหลอดเลือดสมองเป็นข้อมูลที่ซ้ำกัน จึงได้ทำการคัดออกจากการนำมาใช้ศึกษาทำให้มีจำนวนทั้งหมด 197 คนสำหรับการไปสร้างโมเดล

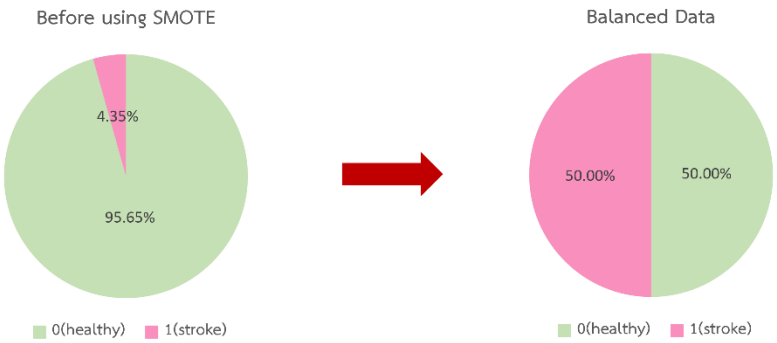
4.4 การใช้ Synthetic Minority Over-sampling Technique ในข้อมูลชุดที่ 1

การเรียนรู้จากข้อมูลที่ไม่สมดุลกัน (imbalanced data) เป็นปัญหาที่เกิดจากการจำแนกประเภทกลุ่มข้อมูลมีไม่เท่ากัน ผลที่ได้อาจทำให้เกิดค่า accuracy ที่ทำให้เข้าใจผิดพลาด (misleading accuracy) และการทำนายผลที่ biased [17] กลุ่มตัวอย่างที่เป็นโรคหลอดเลือดสมองหรือกลุ่มตัวอย่างที่เราสนใจศึกษามีจำนวนน้อยกว่ามากเมื่อเทียบกับกลุ่มตัวอย่างที่ไม่เป็นโรค วิธีการ under-sampling ของ majority class ที่เป็นข้อมูลคนปกติ รวมกับวิธีการ over-sampling ของ minority class ที่เป็นข้อมูลผู้ป่วยที่เป็นสโตรก เพื่อให้ได้ผลลัพธ์ที่ดีขึ้นในการจำแนกประเภท วิธีการ SMOTE ถูกนำไปใช้อย่างแพร่หลาย [18] ตั้งแต่ที่มีการตีพิมพ์เมื่อปี 2002 หลักการคือจำลองข้อมูลใหม่ที่ช่วยในการทำนายกลุ่มข้อมูลที่มีน้อย โดยการ interpolation ระหว่าง minority class ที่อยู่ใน defined neighborhood หลักการคือสนใจ feature space หรือข้อมูลปัจจัยเฉพาะแต่ละตัวมากกว่า data space ที่เป็นภาพรวมของข้อมูลทั้งหมด



รูปที่ 17 การสร้างข้อมูลจำลองแบบ SMOTE

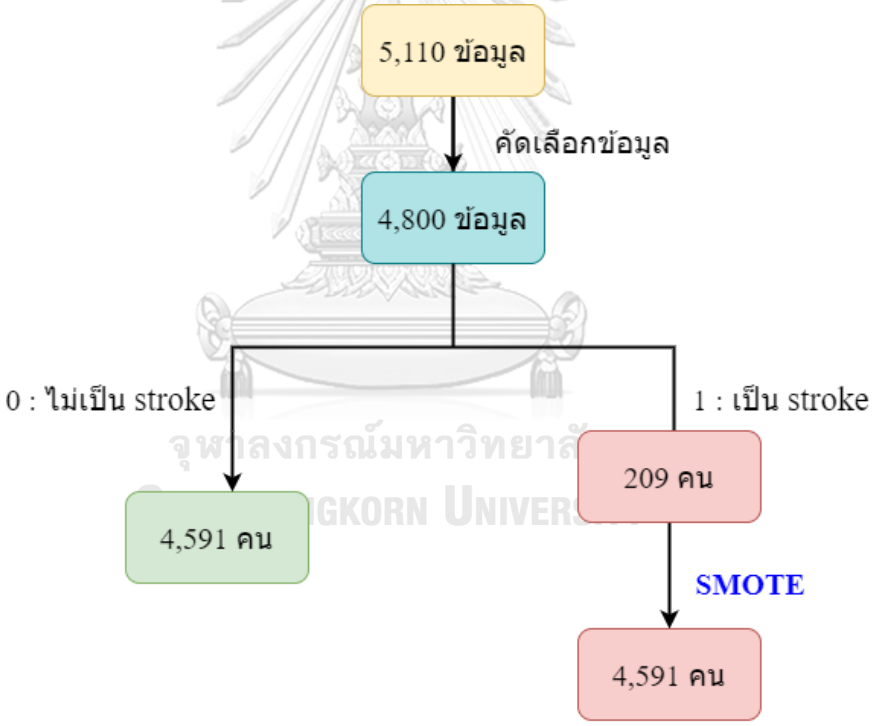
แต่ละจุด r_1 ถึง r_4 หรือจุดสีแดงในรูปอยู่ระหว่างจุด x_{i1} ถึง x_{i4} แสดงเป็นจุดสีฟ้า คือ ข้อมูลที่เกิดการประมาณค่าในช่วงเป็นข้อมูลที่เพิ่มขึ้นจากวิธีการนี้



รูปที่ 18 ข้อมูลหลังการทำให้สัดส่วนข้อมูลเท่ากัน

4.5 ข้อมูลที่เทรนโมเดล

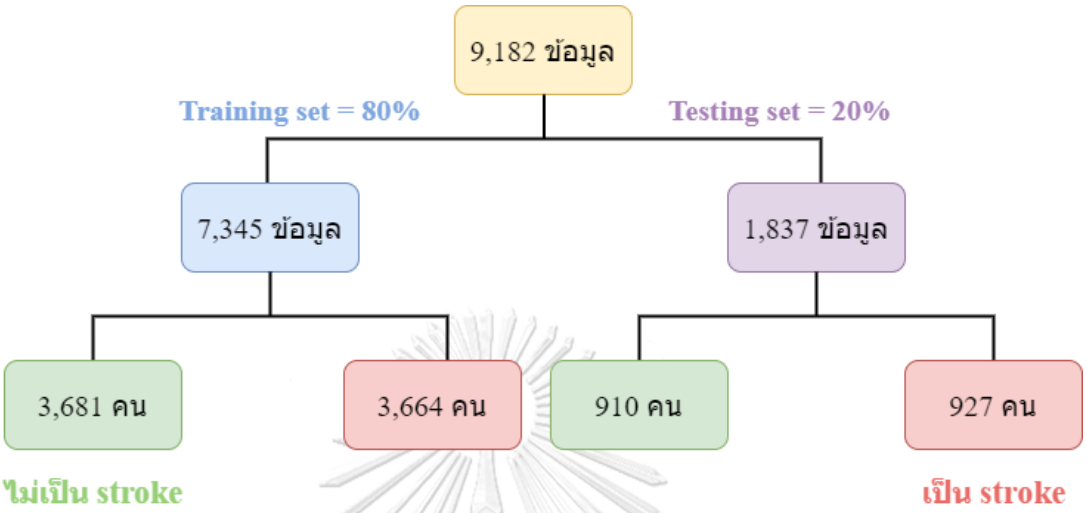
ข้อมูลชุดที่ 1 หรือข้อมูลชุดทดลอง



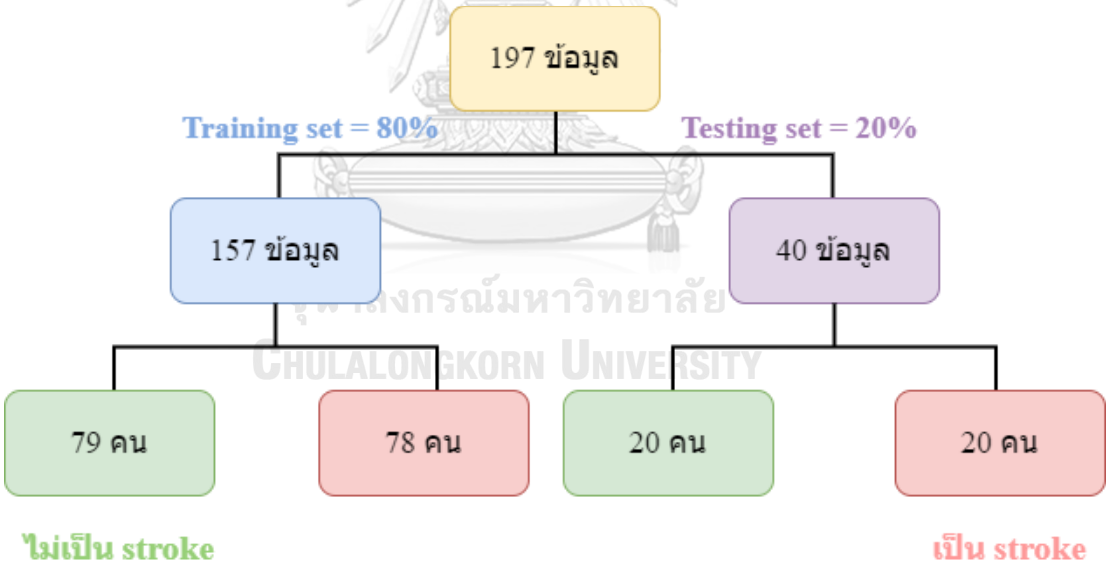
รวมทั้งหมด 9,182 ข้อมูล

รูปที่ 19 แผนผังข้อมูลชุดแรก

การแบ่งข้อมูลที่ใช้ train และ test จะอยู่ในอัตราส่วน 80:20 ทั้งข้อมูลชุดที่ 1 และ ข้อมูลชุดที่ 2 แสดงในรูปที่ 20 และ รูปที่ 21 ตามลำดับ



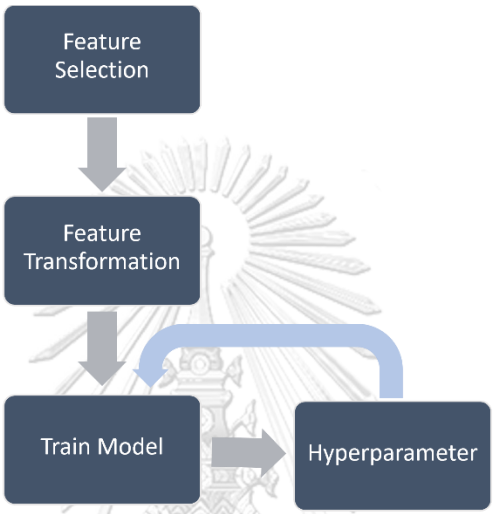
รูปที่ 20 สรุปจำนวนข้อมูลที่ใช้ข้อมูลชุดที่ 1



รูปที่ 21 สรุปจำนวนข้อมูลที่ใช้ข้อมูลชุดที่ 2

4.5 Hyperparameter Optimization

หลังจากที่ได้ทำการคัดเลือกข้อมูลที่เหมาะสมในการนำไปใช้เทรนโมเดลและทำการแปลงข้อมูลให้นำไปใช้กับตัวจำแนกประเภทต่าง ๆ ได้เกิดประสิทธิภาพ ขั้นตอนนี้จึงเป็นการกำหนดค่าพารามิเตอร์ที่ทำให้การทำนายโรคสโตรกได้ผลลัพธ์ที่แม่นยำที่สุด แผนภาพขั้นตอนอธิบายเพิ่มเติมแสดงในรูปที่ 22



รูปที่ 22 แผนภาพขั้นตอน Hyperparameter Optimization

ค่าพารามิเตอร์ที่ใช้ของ KNN แสดงในตารางที่ 4

ตารางที่ 4 พารามิเตอร์ของ KNN

Parameters	Description
n_neighbors	number of neighbors
algorithm	auto, ball_tree, kd_tree, brute
p	p = 1, manhattan_distance (l1), and euclidean_distance (l2) for p = 2
n_jobs	number of parallel jobs for neighbors search

ค่าพารามิเตอร์ที่ใช้ของ Support Vector Machine แสดงรายละเอียดในตารางที่ 5

ตารางที่ 5 พารามิเตอร์ของ SVC

Parameters	Description
C	float
kernel	linear, poly, rbf, sigmoid, precomputed or callable
degree	degree of the polynomial kernel function
gamma	scale, auto or float
probability	false and true
random_state	integer or none

ค่าพารามิเตอร์ที่ใช้ของ Random Forest แสดงรายละเอียดในตารางที่ 6

ตารางที่ 6 พารามิเตอร์ของ RF

Parameters	Description
n_estimators	the number of trees in the forest
criterion	the quality of a split (tree-specific)
max_depth	the maximum of the tree
max_features	auto, sqrt, log2
bootstrap	True and False
oob_score	out of bag samples: True and False
class_weight	balanced, balanced_subsample, dict or list of dicts

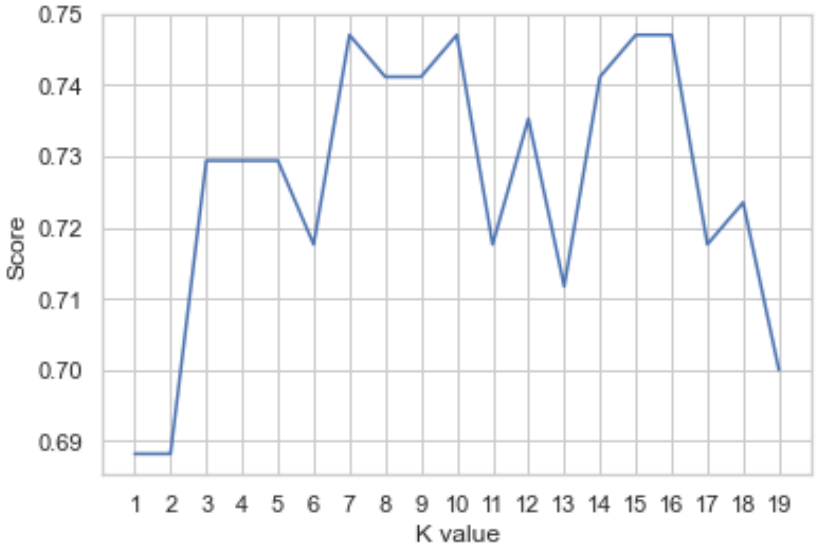
ค่าพารามิเตอร์ที่ใช้ของ AdaBoost แสดงรายละเอียดในตารางที่ 7 อธิบายเพิ่มเติมหากไม่ปรับ base estimator จะใช้ Decision Tree ที่มีค่า max depth = 1 หากใช้ตัวจำแนกประเภทตัวอื่นสามารถปรับ classes_ และ n_classes ได้

ตารางที่ 7 พารามิเตอร์ของ AdaBoost

Parameters	Description
base_estimator	the boosted ensemble
n_estimators	the number of trees in the forest
learning_rate	float, default=1.0
algorithm	SAMME, SAMME.R real boosting algorithm
random_state	integer or None

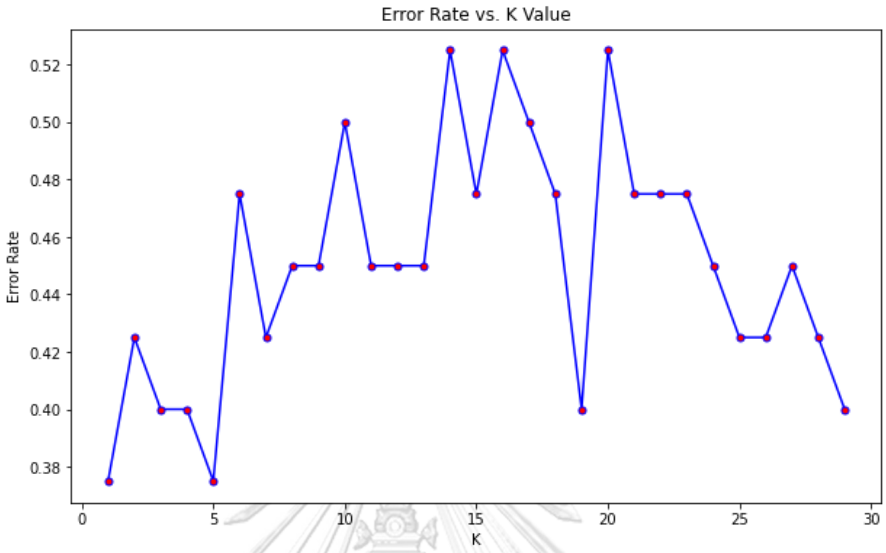
4.5 การดำเนินการเทรนโมเดล

ตัวจำแนกประเภท KNN มีค่า K ที่ใช้จะเห็นได้จากรูปที่ 23 ว่า K=7 สามารถทำให้ผลลัพธ์มีค่า accuracy ได้สูงที่สุดก่อนเป็นอันดับแรก



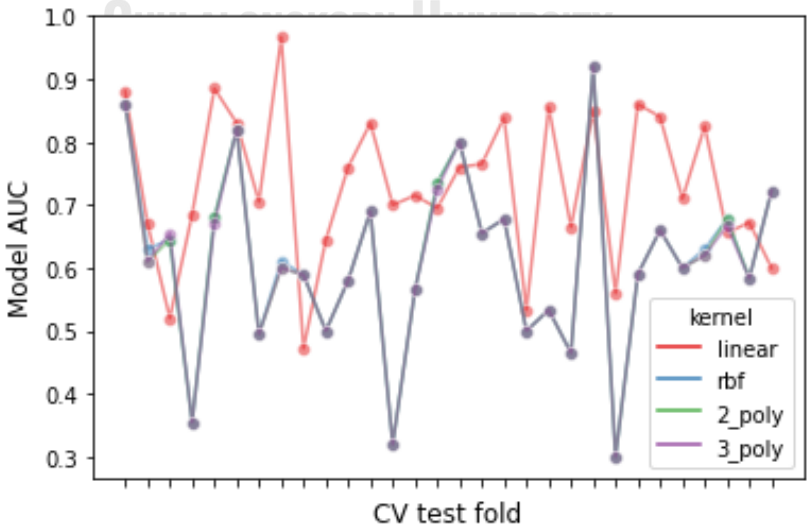
รูปที่ 23 การปรับค่า K และผลความแม่นยำของโมเดล

รูปที่ 24 แสดงกราฟระหว่าง Error rate กับค่า K ที่ใช้ ยิ่งค่า error rate น้อยหมายถึงโมเดลมีค่า accuracy สูง



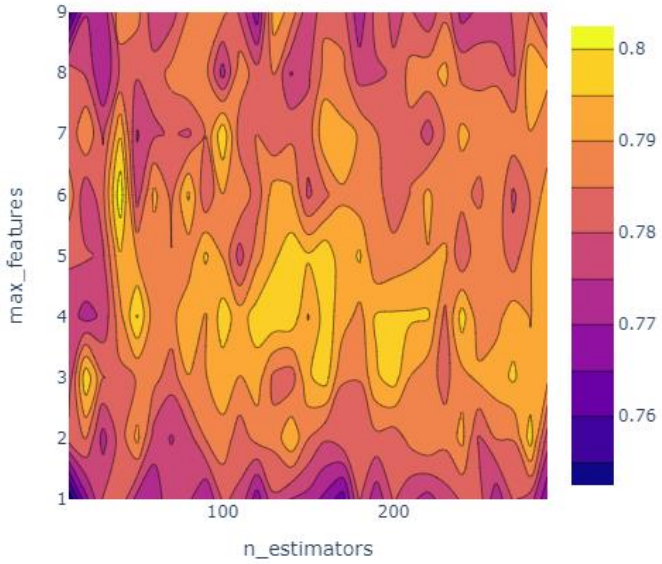
รูปที่ 24 กราฟแสดงผลของค่า K กับอัตราความผิดพลาดของโมเดล

เนื่องจาก SVC มีหลายรูปแบบจึงได้ทำการทดสอบว่ารูปแบบใดที่ให้ผลลัพธ์ดีที่สุด ในที่นี้จะใช้ค่า AUC เพื่อเปรียบเทียบ โดยที่ค่าสูงสุดของ AUC=1 หากมีค่าที่เข้าใกล้ 1 จึงหมายถึงโมเดลให้ผลลัพธ์ที่ดีกว่า รูปที่ 25 แสดงให้เห็นว่าแบบเส้นตรงให้ค่า AUC มากกว่า rbf และ poly ลำดับที่ 2 3 ที่มีความใกล้เคียงกันมาก



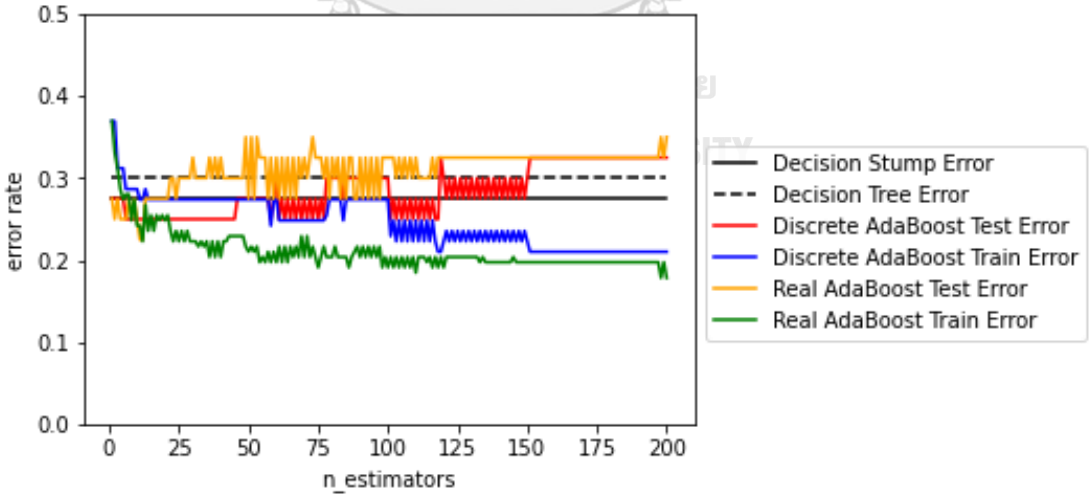
รูปที่ 25 กราฟแสดงผลจาก SVC ในรูปแบบที่ต่างกัน

แผนภาพรูปที่ 26 แสดงให้เห็นพารามิเตอร์ max_features และ n_estimators ของ RF โดยที่โซนสีเหลืองสว่างสุดมีค่าประมาณ 0.8 เป็นส่วนที่พารามิเตอร์ทั้งสองตัวทำให้โมเดลมีค่า accuracy สูงสุด



รูปที่ 26 แผนภาพผลการปรับพารามิเตอร์จาก RF

รูปที่ 27 เปรียบเทียบตัวจำแนกประเภท AdaBoost ในรูปแบบ algorithm ที่ต่างกัน



รูปที่ 27 กราฟเปรียบเทียบ algorithm ของ Adaboost

บทที่ 5

ผลการทดลอง

5.1 ผลการทดลองจากข้อมูลชุดทดลอง

หลังจากที่เตรียมการข้อมูลเรียบร้อยและปรับสัดส่วนให้ข้อมูลสมดุลกัน ข้อมูลทั้งหมดถูกปรับใช้กับค่าพารามิเตอร์ที่เหมาะสม จะได้ว่าตัวจำแนกประเภทที่ให้ค่าเฉลี่ย Precision สูงสุด คือ RF และ AdaBoost คิดเป็น 0.94 ลำดับถัดไป คือ SVC มีค่า 0.90 และ KNN มีค่า 0.89 เรียงตามลำดับ

ตารางที่ 8 ผลการทดลองจากข้อมูลชุดทดลอง

Model	Accuracy	Precision	Recall	f1-score
KNN	0.87	0.89	0.87	0.87
SVC	0.89	0.90	0.89	0.89
RF	0.94	0.94	0.94	0.94
AdaBoost	0.94	0.94	0.94	0.94
average	0.91	0.92	0.91	0.91

5.2 ผลการทดลองจากข้อมูลโรงพยาบาลจุฬาลงกรณ์

ผลการทดลองนี้ใช้วิธีการที่คล้ายกันกับข้อมูลชุดทดลอง แต่ไม่มีการปรับสมดุลของสัดส่วนข้อมูล เนื่องจากข้อมูลมีเอาท์พุตที่ต้องการทำนายเท่ากัน ทำการปรับจูนพารามิเตอร์ที่เหมาะสม จะได้ว่าตัวจำแนกประเภทที่ให้ค่าเฉลี่ย Precision สูงสุดคือ RF คิดเป็น 0.78 และอันดับถัดไปได้แก่ SVC มีค่า 0.76, AdaBoost มีค่า 0.73 และ KNN มีค่า 0.63 เรียงตามลำดับ

ตารางที่ 9 ผลการทดลองจากข้อมูลโรงพยาบาลจุฬาลงกรณ์

Model	Accuracy	Precision	Recall	f1-score
KNN	0.62	0.63	0.62	0.62
SVC	0.75	0.76	0.75	0.75
RF	0.78	0.78	0.78	0.77
AdaBoost	0.73	0.73	0.72	0.73
average	0.72	0.73	0.72	0.72

5.3 ผลการใช้งานบนplatform

ผลการดำเนินงานที่ใช้การเรียนรู้ของเครื่องในการทำวิทยานิพนธ์พบความแตกต่างของการใช้งานแต่ละ platform ได้ดังนี้

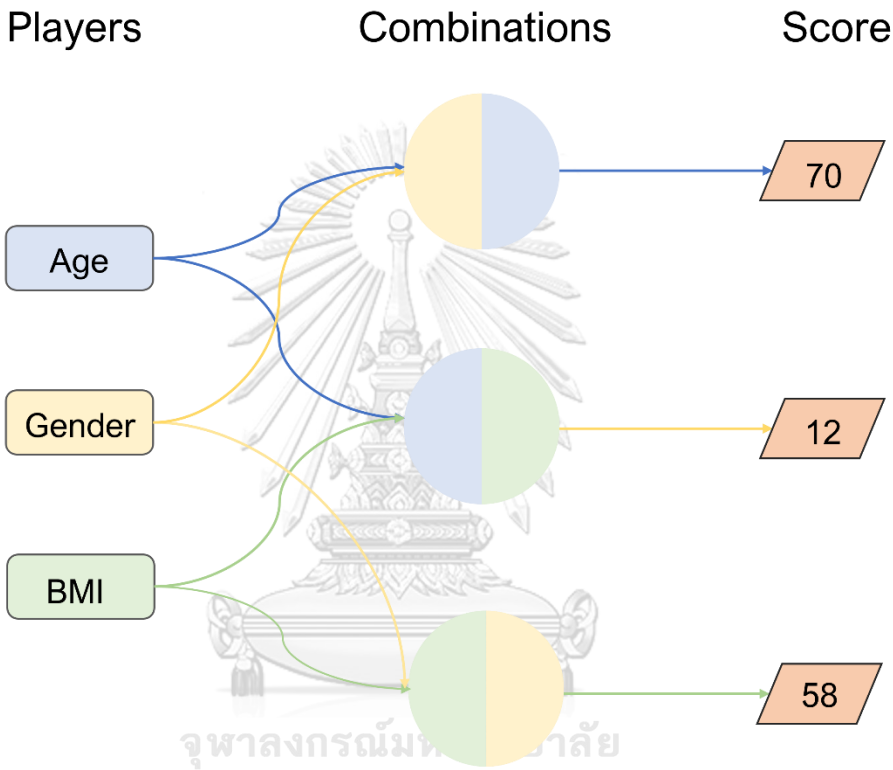
ตารางที่ 10 ผลการใช้งานบน platform

	MATLAB (desktop)	Jupyter Notebook	Google Colaboratory
ที่เก็บไฟล์	บนคอมพิวเตอร์	บนคอมพิวเตอร์	Google Drive
Hardware ที่ใช้	บนคอมพิวเตอร์	บนคอมพิวเตอร์	ไม่ใช้ เพราะใช้ cloud
จำนวน library	ตามที่มีจาก MathWorks	มีเพิ่มขึ้นตลอด	มีเพิ่มขึ้นตลอด
การติดตั้ง library	ไม่ต้อง	ติดตั้งบนเครื่อง	รันใหม่ทุกครั้ง
ภาษาที่ใช้เขียน	MATLAB	Python	Python
ความปลอดภัยของข้อมูล	มากเพราะรันบนคอมพิวเตอร์ที่ใช้	มากเพราะรันบนคอมพิวเตอร์ที่ใช้	ค่อนข้างมากเพราะรันบน cloud แต่ต้องล็อคอินผู้ใช้งาน
การเรียกดูไฟล์	ต้องมีโปรแกรม	ต้องมีโปรแกรม	สามารถดูได้โดยไม่ต้องติดตั้งโปรแกรม
File syncing	ไม่มี	ไม่มี auto saving	มี autosaving และแบคอัพเวอร์ชั่น
ความสะดวก	น้อยเพราะต้องใช้พื้นที่ค่อนข้างมากในการติดตั้งบนคอมพิวเตอร์	ปานกลางเพราะต้องติดตั้งตั้งบนคอมพิวเตอร์	ค่อนข้างสะดวกเพราะสามารถเปิดดูได้จาก browser

สรุปแล้วผู้จัดทำเลือกที่จะใช้ Google Colaboratory เป็นหลักเพราะไม่จำเป็นต้องใช้คอมพิวเตอร์เครื่องใดเครื่องนึง สามารถใช้คอมพิวเตอร์เครื่องที่มี internet เข้าถึง แต่ข้อเสียคือต้องเรียกใช้ library ใหม่ทุกครั้งที่มีการใช้งานไฟล์

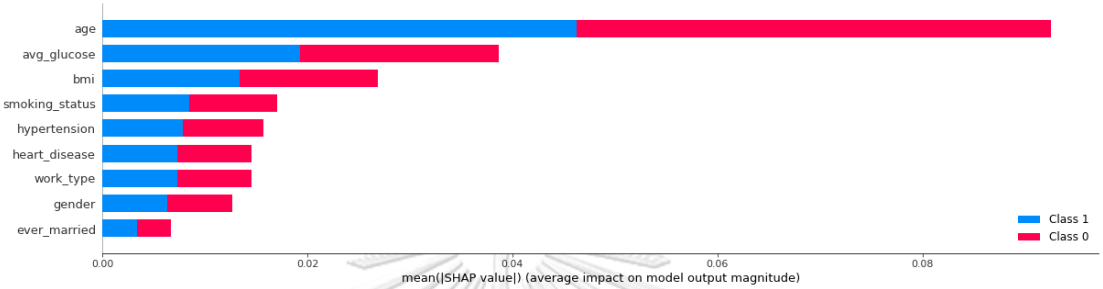
5.3 ผลวิเคราะห์ความเสี่ยงที่ก่อให้เกิดโรคหลอดเลือดสมอง

ผู้จัดทำได้เลือกใช้ค่า Shapley Additive exPlanations (SHAP) หรือ Shapley value [19] เป็นค่าประมาณของ contributions ของการจำลองหลายครั้งในโมเดล จะได้ค่า predictive error ซึ่งเป็น marginal contributions โดยที่ค่าเฉลี่ยของ marginal contribution มาจากปัจจัยเฉพาะของซับเซตทั้งหมดที่เป็นไปได้ รูปที่ 28 แสดงการจำลองวิธีการทำงานของ SHAP



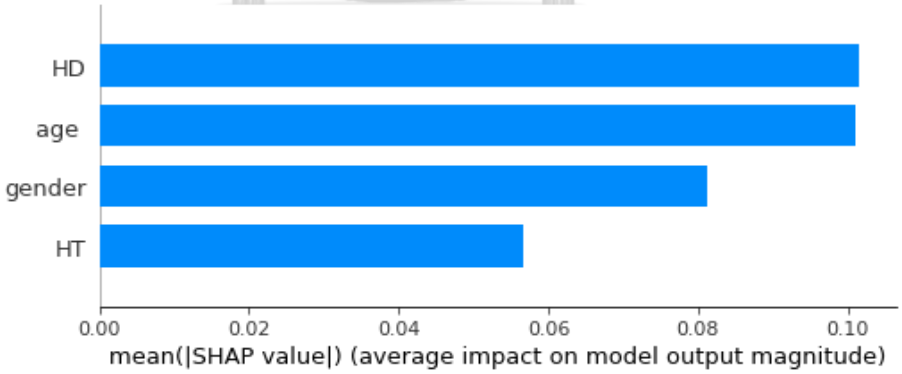
รูปที่ 28 รูปจำลองวิธีการทำงานของ SHAP

TreeExplainer ใช้ประมาณค่า SHAP สำหรับ tree-based models [20] ผลลัพธ์ที่แสดงในรูปที่ 29 กล่าวคือเมื่อมีอายุมากขึ้นยิ่งมีความเสี่ยงต่อการเป็นโรคหลอดเลือดสมอง อันดับที่ 2 คือระดับน้ำตาลในเลือด สามารถวิเคราะห์ต่อในการเป็นโรคเบาหวาน อันดับที่ 3 คือ ค่า BMI และอันดับที่มีผลรองลงมาคือ การสูบบุหรี่, โรคความดันสูง, โรคหัวใจ, การทำงาน, เพศ และสถานะการสมรส



รูปที่ 29 ปัจจัยที่ส่งผลต่อการทำนายโรคหลอดเลือดสมองจากข้อมูลชุดทดลอง

เนื่องจากปัจจัยที่นำมาวิเคราะห์ร่วมกับข้อมูลชุดทดลองมีเพียง 4 ปัจจัย รูปที่ 30 แสดงให้เห็นว่าอายุที่มากขึ้นและโรคหัวใจส่งผลมากเป็นอันดับหนึ่งที่ทำให้เกิดโรคหลอดเลือดสมอง เพศและโรคความดันโลหิตสูงเป็นปัจจัยเสี่ยงรองอันดับถัดมา

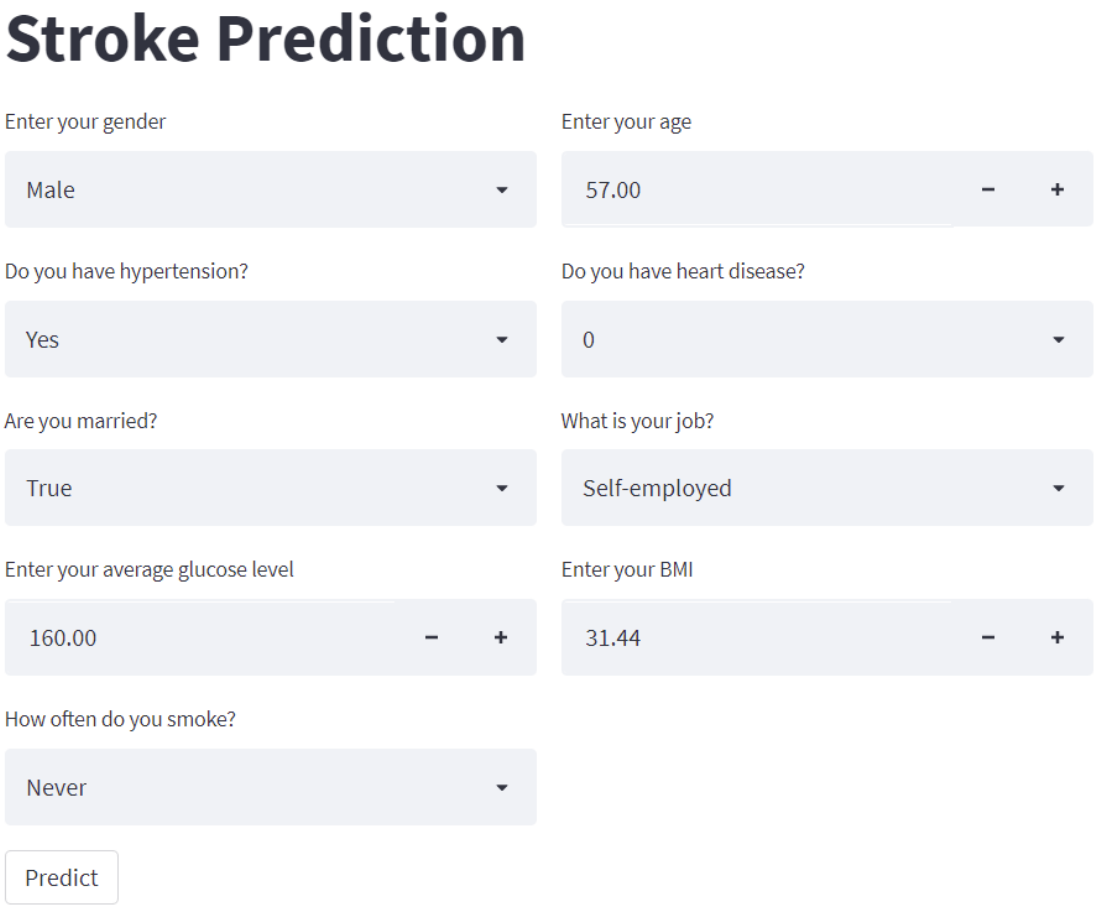


รูปที่ 30 ปัจจัยที่ส่งผลต่อการทำนายโรคหลอดเลือดสมองจากข้อมูลรพ.จุฬาลงกรณ์

เมื่อเปรียบเทียบกับผลงานวิจัยอื่น [21] และจากงานวิจัยที่เกี่ยวข้องในบทที่ 2 นั้น ลำดับปัจจัยเฉพาะที่ส่งผลต่อการเป็นโรคหลอดเลือดสมองมีความแตกต่างกัน เนื่องจากข้อมูลที่ใช้ในการทำนายมีแหล่งที่มาแตกต่างกัน ซึ่งเป็นลักษณะเฉพาะของแหล่งข้อมูลนั้น

5.4 การนำไปใช้ประโยชน์จริงในรูปแบบ Web Application

โมเดลการทำนายโรคหลอดเลือดสมองที่เสร็จสมบูรณ์ถูกนำไปพัฒนาต่อในรูปแบบเว็บแอปพลิเคชั่น เพื่อที่จะให้บุคลากรทางการแพทย์และบุคคลทั่วไปได้ใช้งานสะดวกขึ้น



รูปที่ 31 หน้าต่างการใช้งานแอปพลิเคชั่นทำนายโรคหลอดเลือดสมอง

รูปที่ 31 หน้าต่างการใช้งานแอปพลิเคชั่นทำนายโรคหลอดเลือดสมอง แสดงตัวอย่างการกรอกข้อมูลชุดใหม่ที่อยู่นอกเหนือจากโมเดลที่ทำในวิทยานิพนธ์นี้

บทที่ 6
สรุปผลงานวิจัย

6.1 สรุปผลการวิจัย

การศึกษาตัวจำแนกประเภทในการเรียนรู้ของเครื่อง K-NN, SVM, RF และ Adaboost พบว่าการทำนายโรคหลอดเลือดสมองที่มีประสิทธิภาพมากที่สุดในข้อมูลชุดแรก คือ ตัวจำแนกประเภทแบบ RF และ Adaboost มีค่าความแม่นยำที่ 94 % และในข้อมูลชุดที่ 2 จากศูนย์โรคหลอดเลือดสมองแบบครบวงจรโรงพยาบาลจุฬาลงกรณ์พบว่าตัวจำแนกประเภทแบบ RF มีค่าความแม่นยำที่ 78 % สูงสุดในกลุ่มตัวจำแนกประเภทที่ใช้ในการทดลอง

ปัจจัยเฉพาะของข้อมูลที่มีอยู่ในกลุ่มข้อมูลชุดที่ 1 มีทั้งหมด 10 ปัจจัย แต่เมื่อทำการวิเคราะห์และจัดการข้อมูลก่อนนำไปใช้เทรนโมเดล เลือกใช้เพียง 9 ปัจจัยในการทำนายโรคหลอดเลือดสมอง กรณีข้อมูลชุดที่ 2 มีจำนวนข้อมูลและปัจจัยเฉพาะที่น้อยกว่ามาก จึงได้ทำการรวบรวมข้อมูลบางประเภทที่คล้ายคลึงกันแล้วนำไปใช้ในการทำนาย ใช้ 4 ปัจจัยเสี่ยงในการวิเคราะห์ผลลัพธ์โมเดล ข้อมูลที่มีประสิทธิภาพเป็นส่วนสำคัญที่มีผลต่อโมเดลการเรียนรู้ของเครื่อง

ผลจากการทำนายการเรียนรู้ของเครื่องสามารถนำไปใช้ประกอบการวินิจฉัยระดับความเสี่ยงโรคหลอดเลือดสมองนอกเหนือจากการวิเคราะห์ทางสถิติและประสบการณ์ของแพทย์ โดยได้จัดทำในรูปแบบของ web application เป็นการประเมินความเสี่ยงเบื้องต้นสำหรับผู้สูงอายุที่มีความเสี่ยงในการเกิดโรคหลอดเลือดสมอง

6.2 ข้อเสนอแนะเพื่อการวิจัยในอนาคต

การเก็บกลุ่มตัวอย่างเพิ่มเติมทั้งในจำนวนคนและปัจจัยเฉพาะอื่นเพื่อมาเปรียบเทียบกับผลงานวิจัยนี้ เช่น โรคเบาหวาน ประวัติทางพันธุกรรม พฤติกรรมการออกกำลังกาย เป็นต้น รวบรวมนำมาใช้กับการสร้างโมเดลใหม่ในอนาคต ทั้งนี้คอยสังเกตผลจากโมเดลเดิมว่าสามารถปรับใช้ได้หรือไม่กับข้อมูลชุดใหม่ เลือกใช้ตัวจำแนกประเภทแบบอื่นในการเทรนโมเดล พัฒนาต่อยอดไปสู่การใช้ Deep Learning เช่น ภาพแสกน MRI, ภาพวิเคราะห์หน้าผู้ป่วย สามารถบอกระดับความเสี่ยงในการเป็นโรคหลอดเลือดสมอง

บรรณานุกรม

[1] M. S. Sirsat, E. Ferme, and J. Camara, "Machine Learning for Brain Stroke: A Review," *Stroke and Cerebrovascular Diseases,* vol. 29, no. 10, p. 105162, Oct 2020, doi: 10.1016/j.jstrokecerebrovasdis.2020.105162.

[2] A. Panesar, *Machine Learning and AI for Health*, 1 ed.: Apress, Berkeley, CA, 2019, p. 368.

[3] K. Kostev, T. Wu, Y. Wang, K. Chaudhuri, and C. Tanislav, "Predicting the risk of stroke in patients with late-onset epilepsy: A machine learning approach," *Epilepsy Behav,* vol. 122, p. 108211, Sep 2021, doi: 10.1016/j.yebeh.2021.108211.

[4] R. C. Deo, "Machine Learning in Medicine," *Circulation,* vol. 132, no. 20, pp. 1920-30, Nov 17 2015, doi: 10.1161/CIRCULATIONAHA.115.001593.

[5] S. S.-S. a. S. Ben-David, *Understanding Machine Learning from Theory to Algorithms*. Cambridge University Press, 2014.

[6] M. S. K. Minhaz Uddin Emon, Tamara Islam Meghla, Md. Mahfujur Rahman, and a. M. S. K. M Shamim Al Mamun, "Performance Analysis of Machine Learning Approaches in Stroke Prediction," presented at the Fourth International Conference on Electronics, Communication and Aerospace Technology (ICECA-2020), 2020.

[7] G. C. O'Connell, P. D. Chantler, and T. L. Barr, "Stroke-associated pattern of gene expression previously identified by machine-learning is diagnostically robust in an independent patient population," *Genom Data,* vol. 14, pp. 47-52, Dec 2017, doi: 10.1016/j.gdata.2017.08.006.

[8] J. Liu *et al.*, "Analysis of main risk factors causing stroke in Shanxi Province based on machine learning models," *Informatics in Medicine Unlocked,* vol. 26, 2021, doi: 10.1016/j.imu.2021.100712.

[9] C. H. Lin *et al.*, "Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry," *Comput Methods Programs Biomed,* vol. 190, p. 105381, Jul 2020, doi: 10.1016/j.cmpb.2020.105381.

[10] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model," *The Annals of Applied Statistics,* vol. 9, no. 3, 2015, doi: 10.1214/15-aoas848.

[11] A. Navlani, "KNN Classification using Scikit-learn," ed.

[12] J. VanderPlas, *Python Data Science Handbook*, 1st ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc.,, 2016.

[13] M. Ekman, *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow-Addison-Wesley Professional*, 1 ed.: Addison-Wesley Professional, 2021, p. 752.

[14] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001/10/01 2001, doi: 10.1023/A:1010933404324.

[15] S. Misra and H. Li, "Noninvasive fracture characterization based on the classification of sonic wave travel times," in *Machine Learning for Subsurface Characterization*, 2020, pp. 243-287.

[16] F. Pedregosa, Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É., "Scikit-learn: Machine Learning in Python," *Machine Learning Research,* vol. 12, pp. 2825-2830, October 2011.

[17] J. Brownlee, *Imbalanced Classification with Python*, 2021.

[18] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data - Progress and Challenges, Marking the 15-year Anniversary," *Journal of Artificial Intelligence Research,* vol. 61, pp. 863-905, 2018.

[19] Y. F. a. R. E. Schapire, "Game Theory, On-line Prediction and Boosting," in *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, 1996.

[20] S. M. Lundberg *et al.*, "From Local Explanations to Global Understanding with Explainable AI for Trees," *Nat Mach Intell,* vol. 2, no. 1, pp. 56-67, Jan 2020, doi: 10.1038/s42256-019-0138-9.

[21] T. Liu, W. Fan, and C. Wu, "A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset," *Artificial Intelligence in*

Medicine, vol. 101, p. 101723, Nov 2019, doi: 10.1016/j.artmed.2019.101723.

ประวัติผู้เขียน

ชื่อ-สกุล ฐิติพร อ้ายดี

สถานที่เกิด

วุฒิการศึกษา มหาวิทยาลัยเกษตรศาสตร์