การตรวจจับสิ่งผิดปกติสำหรับเวชระเบียนอิเล็กทรอนิกส์อัตโนมัติ
Automatic Outlier Detection for Identify Data Entry Errors in Electronic Medical Records

ภัทราพร พัชรตระกูล
Pattrapon Patcharatakul

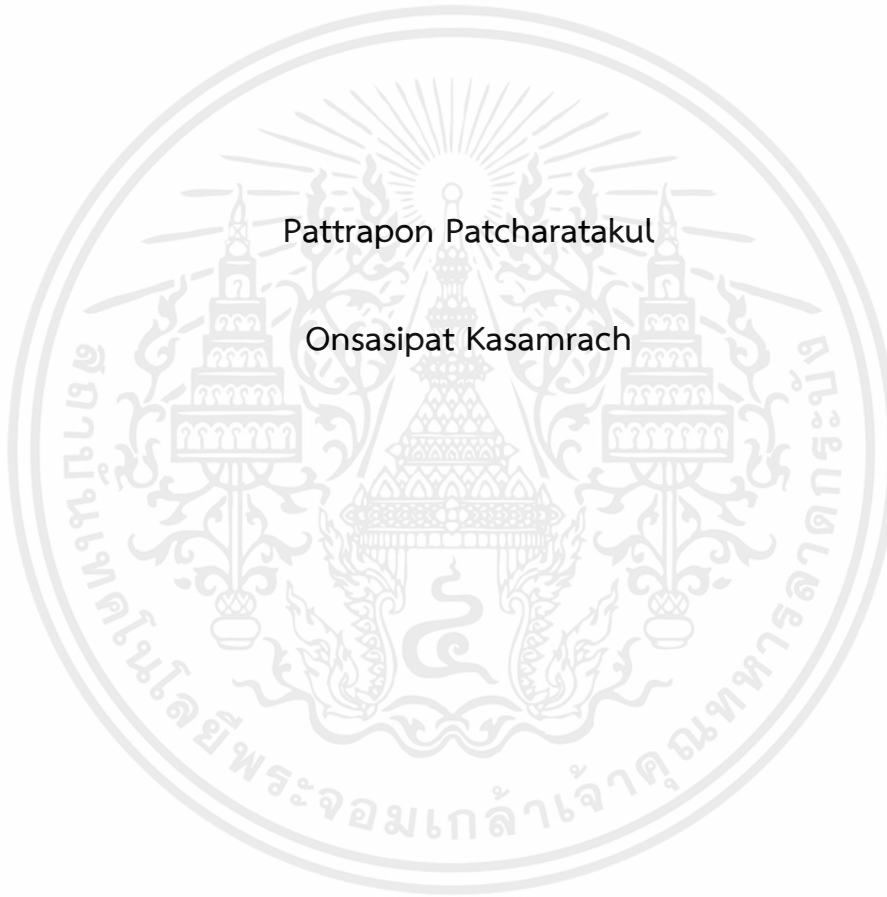
อรศศิพัชร์ เกษมราช
Onsasipat Kasamrach

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

Automatic Outlier Detection for Identify Data Entry Errors in Electronic Medical Records

Pattrapon Patcharatakul

Onsasipat Kasamrach



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2020

หัวข้อปริญญานิพนธ์	การตรวจจับสิ่งผิดปกติสำหรับเวชระเบียนอิเล็กทรอนิกส์อัตโนมัติ	
รายชื่อนักศึกษา	นางสาวภัทราพร พัชรตระกูล	รหัสนักศึกษา 62010690
	นางสาวอรศศิพัชร์ เกษมราช	รหัสนักศึกษา 62011037
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2565	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ.ดร.บุณย์ชนะ ภู่ระหงษ์	

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

.....

(รศ.ดร.บุณย์ชนะ ภู่ระหงษ์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	การตรวจจับสิ่งผิดปกติสำหรับเวชระเบียนอิเล็กทรอนิกส์อัตโนมัติ	
รายชื่อนักศึกษา	นางสาวภัทราพร พัชรตระกูล	รหัสนักศึกษา 62010690
	นางสาวอรศศิพัชร์ เกษมราช	รหัสนักศึกษา 62011037
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2566	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ.ดร.บุณย์ชนะ ภู่ระหงษ์	

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์เพื่อแก้ปัญหาข้อมูลผิดปกติในเวชระเบียนอิเล็กทรอนิกส์ ที่ส่งผลเสียทั้งทำให้ผู้คนเข้าใจข้อมูลผิดพลาดและยังทำให้การนำข้อมูลไปใช้ต่อยอดเกิดความคลาดเคลื่อน ผู้จัดทำจึงได้นำอัลกอริทึมสำหรับการตรวจหาสิ่งผิดปกติ (Anomaly Detection) มาเป็นวิธีแก้ปัญหา โดยวิธีที่เลือกมาใช้คือวิธี Local outlier factor และวิธี Isolation forest ซึ่งจะประเมินประสิทธิภาพโดยใช้ Calinski-Harabasz Index และ Davies-Bouldin Index เป็นตัวชี้วัดประสิทธิภาพของการปฏิบัติงาน ผลที่ได้จากการทดลองพบว่าวิธี Local outlier factor มีประสิทธิภาพที่ดีสุดสำหรับการตรวจจับข้อมูลผิดปกติ เมื่อจำนวนข้อมูลมีปริมาณมาก

Thesis Title	Automatic Outlier Detection for Identify Data Entry Errors in Electronic Medical Records	
Student	Miss Pattrapon Patcharatakul	Student ID. 62010690
	Miss Onsasipat Kasamrach	Student ID. 62011037
Degree	Bachelor of Engineering	
Program	Information Engineering	
Year	2023	
Thesis Advisor	Assoc.Prof.Dr.Boonchana Purahong	

ABSTRACT

This research aims to compare 2 anomaly detection methods between an Isolation forest and a Local outlier factor to identify data errors in electronic medical records problems which can lead people to misunderstand the information and use the information inaccurately. To evaluate the performance of the methods, we use Calinski-Harabasz Index and Davies-Bouldin Index. The result of this experiment suggested that the Local outlier factor is the most effective method when a database contains a large amount of data.

กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์ในครั้งนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากผู้จัดทำได้รับความช่วยเหลือและสนับสนุนจาก รศ.ดร.บุณย์ชนะ ภู่ระหงษ์ ที่เป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ในครั้งนี้ ได้กรุณาสละเวลาให้คำปรึกษาตลอดระยะเวลาการทำโครงงาน และขอขอบคุณ นางสาวปภาดา เปี่ยมจินดา นักศึกษาปริญญาเอกจากสถาบันวิทยสิริเมธี ที่ได้ช่วยชี้แนะแนวทางในการทำงาน อีกทั้งช่วยแนะนำแหล่งข้อมูลสำหรับการค้นคว้า

สุดท้ายนี้ขอขอบคุณครอบครัว และเพื่อนนักศึกษา ที่คอยเป็นกำลังใจและให้ความช่วยเหลือโดยตลอด รวมถึงผู้มีพระคุณทุกท่านที่ไม่ได้กล่าวนามไว้ในที่นี้ที่มีส่วนช่วยเหลือจนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

ภัทราพร พัชรตระกูล
อรศศิพัชร์ เกษมราช

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ข้อมูลผิดพลาดในเวชระเบียนอิเล็กทรอนิกส์มีสาเหตุมาจากปัจจัยหลายประการเช่น ความผิดพลาดจากขั้นตอนรวบรวมข้อมูลหรือความผิดพลาดที่เกิดจากมนุษย์ เป็นต้น ก่อให้เกิดเป็นปัญหาของคนทำงานด้านนี้ที่ต้องเผชิญมานาน ไม่เพียงแค่ทำให้เกิดความเข้าใจผิด แต่ยังส่งผลทำให้การวินิจฉัยคลาดเคลื่อนและลดทอนความน่าเชื่อถือของผลลัพธ์ลง ดังนั้นวิธีแก้ปัญหาที่ได้เลือกมาใช้ก็คืออัลกอริทึมสำหรับการตรวจหาสิ่งผิดปกติ (Anomaly Detection) โดยใช้การเรียนรู้โดยไม่มีผู้สอน (Unsupervised learning) ตรวจจับสิ่งที่ไม่สอดคล้องกับกลุ่มข้อมูลทั่วไปและพิจารณาในการจัดการกับข้อมูลเหล่านั้น

ในการทดลองนี้จะนำเสนอสองวิธีมาเปรียบเทียบกันคือ Local outlier factor และ Isolation forest ซึ่งจะใช้ Calinski-Harabasz Index และ Davies-Bouldin Index เป็นตัวชี้วัดประสิทธิภาพของการปฏิบัติงาน

1.2 จุดประสงค์

- เพื่อลดความผิดพลาดในชุดข้อมูลเวชระเบียนอิเล็กทรอนิกส์
- เพื่อศึกษาวิธี Local outlier factor หรือ Isolation forest สำหรับหาคำตอบว่า วิธีไหนจะเหมาะกับการแก้ปัญหามากกว่ากัน

1.3 ขอบเขตของการทดลอง

ทำการทดลองเพื่อพิสูจน์ว่าระหว่างวิธี Local outlier factor หรือ Isolation forest จะเหมาะสมกับการแก้ปัญหาของการทดลอง โดยใช้ Calinski-Harabasz Index และ Davies-Bouldin Index เป็นตัวชี้วัดประสิทธิภาพของการปฏิบัติงาน

1.4 ผลที่คาดว่าจะได้รับ

- ลดการวินิจฉัยผิดพลาดที่เกิดจากความผิดพลาดของเวชระเบียนอิเล็กทรอนิกส์
- แสดงผลเชิงประจักษ์ได้ว่าวิธี Local outlier factor หรือ Isolation forest ดีกว่า

1.5 อุปกรณ์ที่ต้องใช้

1.5.1 ซอฟต์แวร์

- Python
- Google Colaboratory
- Scikit-Learn

1.6 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงาน	2565							2566		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
Research Review	←	→								
Data Preprocessing			←	→						
Training Model					←	→				
ทดลองและตรวจสอบแก้ไข							←	→		
จัดทำเอกสารและรายงาน				←	→	→	→	→	→	→

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Calinski-Harabasz Index

Calinski-Harabasz Index หรืออีกชื่อที่รู้จักว่า Variance Ratio Criterion ถูกนำเสนอโดย Calinski และ Harabasz ในปีค.ศ. 1974 เป็นวิธีที่ใช้ทำการเปรียบเทียบประสิทธิภาพและเทคนิคการจัดกลุ่มข้อมูล ด้วยอัตราส่วนของผลรวมระหว่างเมทริกซ์การกระจายตัวระหว่างกลุ่มและเมทริกซ์การกระจายภายในกลุ่ม

$$CH = \left[\frac{\sum_{k=1}^{K} n_k ||c_k - c||^2}{K-1}\right] / \left[\frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k} ||d_i - c_k||^2}{N-K}\right]$$

โดย ค่า CH เป็นค่า Calinski-Harabasz Index

ค่า nk และ ck เป็นจำนวนจุดข้อมูล

ค่า c เป็นค่า global centroid

ค่า N เป็นผลรวมทั้งหมดของจุดข้อมูล

หากมีค่า Calinski-Harabasz Index มากจะแสดงถึงกลุ่มจุดข้อมูลมีความหนาแน่นและแยกจากกันได้ดี ส่งผลให้มีประสิทธิภาพที่ดีกว่า

2.1.2 Davies-Bouldin Index

Davis-Bouldin Index คือเกณฑ์การวัดที่ชี้วัดคุณภาพเทคนิคการจัดกลุ่ม ถูกนำเสนอโดย David L. Davies และ Donald W. Bouldin ในปีค.ศ. 1979 โดยทั่วไปแล้วจะใช้เพื่อประเมินการแบ่งจัดกลุ่มข้อมูลด้วยอัลกอริทึมแบบการแบ่งกลุ่มข้อมูลแบบเคมีน (K-Mean clustering) เพื่อหาจำนวนของกลุ่มข้อมูล ด้วยการแบ่งเป็นอัตราส่วนของผลรวมการกระจายตัวข้อมูลในกลุ่มและระยะห่างระหว่างกลุ่ม ซึ่งการคำนวณค่าเป็นไปดังสมการต่อไปนี้

$$\frac{1}{K}\sum_{k=1}^{K} \underset{k \neq 1}{max} \left\{\frac{S(U_k) + S(U_l)}{d(U_k, SU_l)}\right\}$$

โดย ค่า S(Uk)+S(Ul) เป็นระยะห่างของข้อมูลภายในกลุ่ม k และกลุ่ม l

ค่า d(Uk, Ul) เป็นระยะห่างระหว่างจุดกึ่งกลางกลุ่ม k และกลุ่ม l

หากมีค่าน้อยจะส่งผลให้กลุ่มระยะห่างกันมากและมีการกระจัดกระจายในกลุ่มน้อย ทำให้ได้การแบ่งแยกของกลุ่มดีที่สุด

2.1.3 Isolation forest

Isolation forest เป็นวิธีตรวจจับค่าความผิดปกติโดยการสุ่มจุดบนข้อมูลแล้วประกอบขึ้นเป็นกลุ่มของต้นไม้ทวิภาค (Binary tree) ที่มีโครงสร้างอัลกอริทึม โดยมีสมมุติฐานว่าข้อมูลปกติจะสามารถแตกกิ่งสาขาเป็นจำนวนพอ ๆ กัน ดังนั้นหากพบโหนด (Node) ที่แตกสาขาน้อยกว่าค่าเฉลี่ยอย่างมีนัยสำคัญ ก็จะถือว่าเป็นข้อมูลที่ผิดปกติ ในการสร้าง Isolation forest จากชุดข้อมูล (Dataset) จะมีพารามิเตอร์สำคัญอยู่ 3 ตัว ได้แก่

Algorithm 1 : $iForest(X, t, \psi)$

Inputs: X - input data, t - number of trees, ψ - sub-sampling size

Output: a set of t *iTrees*

Initialize $Forest$
set height limit $l = ceiling(\log_2 \psi)$
for $i = 1$ to t **do**
  $X' \leftarrow sample(X, \psi)$
  $Forest \leftarrow Forest \cup iTree(X', 0, l)$
end for
return $Forest$

รูปที่ 2.1 อัลกอริทึม Isolation Forest

1. X เป็นชุดข้อมูลที่เราจะนำมาฝึกฝน
2. t เป็นจำนวน tree ภายใน forest
3. ψ เป็นขนาดกลุ่มตัวอย่างย่อย (Subsampling size)

ข้อมูลที่เราจะตรวจสอบว่ามันเป็นข้อมูลผิดปกติหรือไม่นั้น จะถูกส่งไปยัง Isolation Tree แต่ละต้น เพื่อคำนวณหาคะแนนความผิดปกติ (Anomaly score) และค่านี้จะใช้ในการตัดสินว่าข้อมูลนั้นเป็นข้อมูลผิดปกติหรือไม่

Algorithm 2 : $iTree(X, e, l)$

Inputs: X - input data, e - current tree height, l - height limit

Output: an iTree

if $e \geq l$ or $|X| \leq 1$ **then**
  return $exNode\{Size \leftarrow |X|\}$
else
  let Q be a list of attributes in X
  randomly select an attribute $q \in Q$
  randomly select a split point p from max and min values of attribute q in X
  $X_l \leftarrow filter(X, q < p)$
  $X_r \leftarrow filter(X, q \geq p)$
  return $inNode\{Left \leftarrow iTree(X_l, e+1, l),$
    $Right \leftarrow iTree(X_r, e+1, l),$
    $SplitAtt \leftarrow q,$
    $SplitValue \leftarrow p\}$
end if

รูปที่ 2.2 อัลกอริทึม Isolation Tree

โดย ค่า l เป็นค่าสูงสุดของ $\log_2 \psi$

ค่า Q เป็นค่าแอตทริบิวต์ (Attribute) ของข้อมูลแต่ละตัวอย่าง

ในการสร้าง Isolation Tree ให้สุ่ม X มาตามจำนวนที่ขนาดกลุ่มตัวอย่างย่อยกำหนด แล้วให้สุ่มแอตทริบิวต์ Q มา 1 แอตทริบิวต์จากนั้นให้สุ่มค่าจุดแยกตัว (Split point) จากแอตทริบิวต์อีกทีเพื่อใช้ในการแบ่งข้อมูล ทำซ้ำจนกว่าจะเข้าเงื่อนไขว่าทุกตัวอย่างข้อมูลจะแยกออกจากกันและ tree มีความลึกเกิน ค่า l จะสามารถเขียนได้เป็น Q = { q1, q2, q3, ..., qx } ในการสร้าง Isolation Tree 1 ต้น

จากนั้นคำนวณหาค่าคะแนนความผิดปกติจากสมการต่อไปนี้

$$c(\Psi) = 2H(\Psi - 1) - (2(\Psi - 1)/\Psi)$$

$$H(i) = \ln(i) + 0.5772$$

โดย ค่า $c(\Psi)$ เป็นค่าเฉลี่ยของระยะทางระหว่างโหนด (Average Path Length)
ค่า h(*x*) เป็นค่าเฉลี่ยของตัวอย่างทั้งหมด

นำค่า $c(\Psi)$ ที่ได้จากสมการไปลดความซ้ำซ้อน (Normalize) ของ E(h(x)) ต่อในสมการต่อไป

$$s(x,\Psi) = 2^{-\frac{E(k(x))}{c(\Psi)}}$$

จากทั้งหมดที่กล่าวมาจะได้ค่า $s(x,\Psi)$ ซึ่งค่าคะแนนความผิดปกติ ซึ่งจะเป็นค่าที่บอกว่าเป็นข้อมูลผิดปกติหรือไม่ หากค่ามีความเข้าใกล้ 1 แสดงว่าเป็นข้อมูลที่ผิดปกติ

2.1.4 Local outlier factor (LOF)

Local outlier factor (LOF) เป็นอัลกอริทึมที่ไว้หาค่าความผิดปกติจากการเปรียบเทียบความหนาแน่นของข้อมูลโดยคำนวณจากการค้นหาค่าข้อมูลที่ใกล้สุด (K-Nearest neighbors) คัดแยกจุดที่มีความหนาแน่นสูงและต่ำ สิ่งใช้ในการคำนวณจะเริ่มจากคำนวณระยะทางของแต่ละจุด ด้วยระยะห่างแมนฮัตตัน (Manhattan distance) ต่อจากนั้นคำนวณค่าข้อมูลที่ใกล้สุดหรือก็คือระยะห่างระหว่างจุดที่เราสนใจกับจุดใกล้เคียง (Neighbours) เมื่อได้ค่าทั้งหมดแล้วให้คำนวณหาค่า local reachable densities (lrd) ของแต่ละจุด ซึ่งจะเป็นตัวบอกว่าจะมีระยะห่างเท่าไหร่จากจุดที่เราสนใจไปจนถึงจุดต่อไป ซึ่งการคำนวณค่าเป็นไปดังสมการต่อไปนี้

$$lrd_k(o) = \frac{||N_k(o)||}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)}$$

โดย ค่า $||N_k(o)||$ เป็นจำนวนของจุดใกล้เคียง
ค่า reachdist เป็นระยะที่เข้าถึงได้ (Reachable distance) สามารถหาค่าได้จากสมการต่อไปนี้

$$reachdist_k(o \leftarrow o') = max\{dist_k(o), dist(o, o')\}$$

เมื่อสามารถหาค่า lrd ได้แล้ว ให้นำค่าทั้งหมดที่หาได้มาคำนวณหาค่า local outlier factor สำหรับหาข้อมูลผิดปกติจากสมการดังต่อไปนี้

$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}}{||N_k(o)||}$$

จากค่า local outlier factor หากมีค่านี้มากกว่า 1 จะมีความเป็นไปได้ว่าเป็นข้อมูลผิดปกติ วิธีนี้เหมาะสำหรับการตรวจจับความผิดปกติของข้อมูลที่มีการกระจายตัวไม่สม่ำเสมอ

2.2 เทคโนโลยีที่เกี่ยวข้อง

2.2.1 Python

Python เป็นภาษาโปรแกรมคอมพิวเตอร์ที่ใช้อย่างแพร่หลาย มุ่งเน้นให้ผู้ใช้สามารถอ่านภาษาสคริปต์ได้ง่าย โดยมีโครงสร้างและไวยากรณ์ของภาษาที่ไม่ซับซ้อน มีความสามารถใช้ชนิดข้อมูลแบบไดนามิก จัดการหน่วยความจำอัตโนมัติ และยังสามารถทำงานบนแพลตฟอร์มได้หลากหลายและใช้งานร่วมกับภาษาการเขียนโปรแกรมยอดนิยมอื่น ๆ จึงทำให้มีหลายองค์กรใหญ่ระดับโลกนิยมนำไปใช้

2.2.2 Google Colaboratory

Google Colaboratory หรืออีกชื่อว่า Google Colab เป็นบริการ Jupyter Notebook บนคลาวด์ (Cloud) จาก Google Research ที่อนุญาตให้สามารถเขียนและดำเนินงานโปรแกรมภาษา Python ได้ผ่านเบราว์เซอร์ (Browser) เหมาะกับงานประเภทการวิเคราะห์ข้อมูลและ Machine learning ซึ่งการใช้บริการนี้ไม่จำเป็นจะต้องติดตั้งและยังมีทรัพยากรเช่นชิปประมวลผลกราฟิก (GPU : Graphics Processing Unit) ให้ใช้ได้ฟรี

2.2.3 ชุดข้อมูล (Dataset)

ชุดข้อมูลที่ใช้ในการทดลองครั้งนี้ประกอบด้วยชุดข้อมูลของผู้ป่วยโรค COVID-19 ภายใต้โครงการ CHIVID (Community Isolation-Based Electronic Health Record during COVID-19) ซึ่งเป็นระบบอำนวยความสะดวกในการเฝ้าระวังและติดตามสังเกตอาการของผู้ป่วยในระยะทางไกล ทั้งกลุ่มผู้ป่วยที่แยก "กักตัวที่บ้าน" (Home Isolation : HI) "กักตัวในชุมชน" (Community Isolation : CI) และ "โรงพยาบาลสนาม" โดยข้อมูลได้ถูกเก็บมาในช่วงเวลาระหว่างเดือนสิงหาคมปี 2021 จนถึงเดือนเมษายนปี 2022 ทั้งหมดจำนวน 17,995 คน ชุดข้อมูลประกอบด้วยอายุ, เพศ, BMI (body mass Index), โรคประจำตัว, ความรุนแรงของโรคและปริมาณออกซิเจนในเลือดประจำวันและอาการผู้ป่วย ซึ่งข้อมูลที่ก่อให้เกิดปัญหามากที่สุดคือข้อมูลปริมาณออกซิเจนในเลือดประจำวัน จากปัญหาที่มีค่าข้อมูลผิดปกติหรือเบี่ยงเบนไปจากช่วงค่าปกติโดยตรวจหาค่าผิดปกติจากการเปรียบเทียบข้อมูลกับผู้ป่วยในอดีตที่มีลักษณะอาการคล้ายคลึงกัน

2.2.4 Scikit-Learn

Scikit-Learn คือชุดคำสั่งเสริมของภาษา Python ที่มีไลบรารี (Library) ใช้ร่วมกับ NumPy, SciPy และ Matplotlib และเป็นเครื่องมือประกอบไปด้วยอัลกอริทึมทางคณิตศาสตร์, สถิติ และ วัตถุประสงค์ทั่วไป สำหรับพื้นฐานในงานด้าน Machine learning นอกจากนี้ยังมีฟังก์ชันการวิเคราะห์การถดถอย (Regression), การจำแนกข้อมูล (Classification), การจัดกลุ่ม (Clustering), การเลือกแบบจำลอง (Model selection) และ การจัดการข้อมูล (Preprocessing) ช่วยอำนวยความสะดวกในการทำงานผ่านทางอินเตอร์เฟสใน Python

บทที่ 3
วิธีการดำเนินงาน

3.1 การติดตั้งเครื่องมือ

ทำการติดตั้งไลบรารี่ (Library) ซึ่งเสมือนโปรแกรมสำเร็จรูปที่ช่วยเก็บฟังก์ชัน (Function) การทำงานเฉพาะทางไว้ทำให้ไม่จำเป็นต้องสร้างขึ้นมาใหม่เองหมดแต่สามารถนำไลบรารี่ที่มีการพัฒนาไว้อยู่แล้วมาใช้งานได้เลย ซึ่งรายชื่อไลบรารี่ที่จำเป็นต้องติดตั้งลงใน Google Colaboratory ก่อนจะเริ่มทำการทดลองมีดังต่อไปนี้

- pandas
- numpy
- matplotlib.pyplot
- seaborn
- sklearn.ensemble
- sklearn.neighbors
- sklearn.metrics

เมื่อติดตั้งไลบรารีสำเร็จ ให้นำเข้าข้อมูลไฟล์ third_clean.csv ซึ่งเป็นกลุ่มตัวอย่างชุดข้อมูล SpO2 (Saturation of Peripheral Oxygen หรือค่าความอิ่มตัวของออกซิเจนในเลือด) ของผู้ป่วยโรค COVID-19 ภายใต้โครงการ CHIVID ทั้งหมด 176,182 ข้อมูล

```
df = pd.read_csv('/content/drive/MyDrive/Senior Project/third_clean.csv')
```

รูปที่ 3.1 แสดงการเขียนโค้ดสำหรับอ่านไฟล์ข้อมูลใน Google Colaboratory

3.2 การทำความสะอาดข้อมูล (Data Cleansing)

เนื่องจากรูปแบบข้อมูลของไฟล์ third_clean.csv ยังไม่อยู่ในสภาพที่พร้อมใช้งานจึงต้องมีการทำความสะอาดข้อมูลก่อนที่จะนำมาใช้ในการทดลอง ซึ่งมีขั้นตอนดังต่อไปนี้

3.2.1 กำจัดข้อมูลที่ไม่ต้องการออกจากไฟล์

spo2
[98]
[98.0,nan,98.0,98.0]
[nan,97.0,97.0]
[nan,97.0,98.0]
[nan,98.0,97.0]
[nan,98.0,97.0]
[nan]
[nan]
[100]

รูปที่ 3.2 ตัวอย่างข้อมูลหัวข้อ spo2 ภายในไฟล์ third_clean.csv

จากรูปที่ 3.2 ภายในแถวข้อมูลของหัวข้อ spo2 ซึ่งข้อมูลควรจะเป็นค่า SpO2 แสดงถึงระดับเปอร์เซ็นต์ออกซิเจนที่เป็นตัวเลขเท่านั้นจึงต้องมีการลบอักขระอื่นนอกจากตัวเลขออก

```
df['spo2'] = df['spo2'].str.replace('[','').str.replace(']','').str.replace(' ','').str.split(',')
```

รูปที่ 3.3 แสดงการเขียนโค้ดสำหรับลบอักขระ

นอกจากนั้นยังมีข้อมูลประเภทไม่ใช่ตัวเลขที่กลายเป็นตัวแปร nan ภายในแถวข้อมูล ทำให้เป็นอุปสรรคต่อการทดลองในขั้นตอนต่อไปได้ จึงจำเป็นต้องลบออกเช่นกัน

```
for j in range(len(df['spo2'])):
  df['spo2'][j] = [ i for i in df['spo2'][j] if i != 'nan']
```

รูปที่ 3.4 แสดงการเขียนโค้ดสำหรับลบค่าที่ไม่ใช่ตัวเลข

3.2.2 แปลงข้อมูลเป็นประเภท float

ในไฟล์ third_clean.csv ข้อมูลทั้งหมดเป็นประเภท string แต่ไลบรารี่ sklearn ต้องการข้อมูลประเภท float ก่อนใช้งานจึงต้องเปลี่ยนประเภทข้อมูลก่อนขั้นตอนต่อไป

```
for j in range(len(df['spo2'])):
  df['spo2'][j] = [ float(i) for i in df['spo2'][j]]
```

รูปที่ 3.5 แสดงการเขียนโค้ดสำหรับแปลงประเภทข้อมูล

3.3 ดำเนินการทดลอง

3.3.1 กำหนดค่าพารามิเตอร์

ในการทดลองนี้ได้ทำการสุ่มค่าพารามิเตอร์มาทดลองข้อมูลแต่ละกลุ่มเพื่อหาค่าที่เหมาะสมที่สุด ค่าพารามิเตอร์มี 3 ค่า ได้แก่ Estimators, Neighbors และ Contamination อยู่ในช่วง 2 ถึง 30 , 2 ถึง 35 และ 0.025 ถึง 0.1 ตามลำดับ ซึ่งจะนำค่าพารามิเตอร์ที่เหมาะสมที่สุดมาใช้ในการวัดประสิทธิภาพของงานในขั้นตอนถัดไป โดยการใช้ Calinski-Harabasz Index และ Davies-Bouldin Index เป็นตัวชี้วัด

3.3.2 คัดแยกกลุ่มตัวอย่าง

เนื่องจากค่า SpO2 ในผู้ป่วยบางโรคมีค่าแตกต่างจากคนปกติ (ค่าปกติระดับออกซิเจนจะอยู่ที่ 96 – 100%) จึงต้องแบ่งกลุ่มข้อมูลก่อนทำการทดลอง มิฉะนั้นข้อมูลของผู้ป่วยเหล่านี้อาจถูกมองเป็นค่าผิดปกติได้ ซึ่งมี 3 กลุ่มนั่นคือ

- กลุ่มผู้ป่วยโรคอ้วน (Obesity) ผู้ป่วยสามารถเกิดภาวะแทรกซ้อนในระบบหายใจได้ โดยเกิดจากไขมันส่วนเกินเข้าไปสะสมในระบบทางเดินหายใจจนส่งผลให้หลอดลมตีบลงทำให้มีค่า SpO2 ต่ำกว่าเฉลี่ยของคนที่ BMI ปกติ (เท่ากับหรือต่ำกว่า 25 kg/m^2)

- กลุ่มผู้ป่วยโรคหอบหืด (Asthma) ผู้ป่วยกลุ่มนี้จะมีบางภาวะ เช่น เมื่อตอบสนองต่อสารภูมิแพ้ ค่า SpO2 สามารถลงต่ำไปได้ถึงช่วง 90 – 95%
- กลุ่มผู้ป่วยโรคปอดอุดกั้นเรื้อรัง (Chronic Obstructive Pulmonary Disease หรือ COPD) เนื่องจากปอดและหลอดลมของผู้ป่วยกลุ่มนี้มีการอักเสบ ทำให้ไม่สามารถรับออกซิเจนได้เต็มที่ ค่า SpO2 ของผู้ป่วยกลุ่มนี้จึงอยู่ที่ 88 - 92%

นอกจากนี้ยังแบ่งข้อมูลที่เหลือออกเป็น 2 กลุ่มนั่นคือ
- กลุ่มผู้ป่วยโรคที่ไม่เกี่ยวข้องกับค่า SpO2
- กลุ่มผู้ป่วยที่ไม่มีโรคประจำตัว

3.3.3 ทำการทดลองในแต่ละโรค

Obesity

```
[ ]  ds1 = df.loc[(df['ud_obesity'] == 1) & (df['ud_asthma'] == 0) & (df['ud_copd'] == 0) & (df['ud_none'] == 0)]
     ds1.reset_index(drop=True, inplace=True)

     ds1 = ds1[ds1['spo2'].astype(bool)]
     ds1.reset_index(drop=True, inplace=True)

     dl1 = [i for j in ds1['spo2'] for i in j]
```

```
[ ]  print(len(dl1))

     240
```

```
[ ]  dl1_array = np.array(dl1).reshape(240,1)
```

```
[ ]  obesity_if = IsolationForest(n_estimators=estimators, contamination=if_contamination).fit(dl1_array)
     obesity_if_predict = obesity_if.predict(dl1_array)
```

รูปที่ 3.6 แสดงการเขียนโค้ดขั้นตอนวิธี Isolation forest ของผู้ป่วยโรคอ้วน

Asthma

```
[ ]  ds2 = df.loc[(df['ud_asthma'] == 1) & (df['ud_obesity'] == 0) & (df['ud_copd'] == 0) & (df['ud_none'] == 0)]
     ds2.reset_index(drop=True, inplace=True)

     ds2 = ds2[ds2['spo2'].astype(bool)]
     ds2.reset_index(drop=True, inplace=True)

     dl2 = [i for j in ds2['spo2'] for i in j]
```

```
[ ]  print(len(dl2))

     2371
```

```
[ ]  dl2_array = np.array(dl2).reshape(2371,1)
```

```
[ ]  asthma_if = IsolationForest(n_estimators=estimators, contamination=if_contamination).fit(dl2_array)
     asthma_if_predict = asthma_if.predict(dl2_array)
```

รูปที่ 3.7 แสดงการเขียนโค้ดขั้นตอนวิธี Isolation forest ของผู้ป่วยโรคหอบหืด

Copd

```
[ ]  ds3 = df.loc[(df['ud_copd'] == 1) & (df['ud_obesity'] == 0) & (df['ud_asthma'] == 0) & (df['ud_none'] == 0)]
     ds3.reset_index(drop=True, inplace=True)

     ds3 = ds3[ds3['spo2'].astype(bool)]
     ds3.reset_index(drop=True, inplace=True)

     dl3 = [i for j in ds3['spo2'] for i in j]
```

```
[ ]  print(len(dl3))

     53
```

```
[ ]  dl3_array = np.array(dl3).reshape(53,1)
```

```
[ ]  copd_if = IsolationForest(n_estimators=estimators, contamination=if_contamination).fit(dl3_array)
     copd_if_predict = copd_if.predict(dl3_array)
```

รูปที่ 3.8 แสดงการเขียนโค้ดขั้นตอนวิธี Isolation forest ของผู้ป่วยโรคปอดอุดกั้นเรื้อรัง

Other Diseases

```
[ ]  ds4 = df.loc[(df['ud_obesity'] == 0) & (df['ud_asthma'] == 0) & (df['ud_copd'] == 0) & (df['ud_none'] == 0)]
     ds4.reset_index(drop=True, inplace=True)

     ds4 = ds4[ds4['spo2'].astype(bool)]
     ds4.reset_index(drop=True, inplace=True)

     dl4 = [i for j in ds4['spo2'] for i in j]
```

```
[ ]  print(len(dl4))

     72981
```

```
[ ]  dl4_array = np.array(dl4).reshape(72981,1)
```

```
[ ]  other_if = IsolationForest(n_estimators=estimators, contamination=if_contamination).fit(dl4_array)
     other_if_predict = other_if.predict(dl4_array)
```

รูปที่ 3.9 แสดงการเขียนโค้ดขั้นตอนวิธี Isolation forest ของผู้ป่วยโรคที่ไม่เกี่ยวข้องกับค่า SpO2

Normal

```
[ ]  ds5 = df.loc[(df['ud_none'] == 1)]
     ds5.reset_index(drop=True, inplace=True)

     ds5 = ds5[ds5['spo2'].astype(bool)]
     ds5.reset_index(drop=True, inplace=True)

     dl5 = [i for j in ds5['spo2'] for i in j]
```

```
[ ]  print(len(dl5))

     100537
```

```
[ ]  dl5_array = np.array(dl5).reshape(100537,1)
```

```
[ ]  normal_if = IsolationForest(n_estimators=estimators, contamination=if_contamination).fit(dl5_array)
     normal_if_predict = normal_if.predict(dl5_array)
```

รูปที่ 3.10 แสดงการเขียนโค้ดขั้นตอนวิธี Isolation forest ของผู้ป่วยที่ไม่มีโรคประจำตัว

Obesity

```
[ ]  ds1 = df.loc[(df['ud_obesity'] == 1) & (df['ud_asthma'] == 0) & (df['ud_copd'] == 0) & (df['ud_none'] == 0)]
     ds1.reset_index(drop=True, inplace=True)

     ds1 = ds1[ds1['spo2'].astype(bool)]
     ds1.reset_index(drop=True, inplace=True)

     dl1 = [i for j in ds1['spo2'] for i in j]
```

```
[ ]  print(len(dl1))

     240
```

```
[ ]  dl1_array = np.array(dl1).reshape(240,1)
```

```
[ ]  obesity_lof = LocalOutlierFactor(n_neighbors=neighbors, contamination=lof_contamination).fit_predict(dl1_array)
```

รูปที่ 3.11 แสดงการเขียนโค้ดขั้นตอนวิธี Local Outlier Factor ของผู้ป่วยโรคอ้วน

Asthma

```
[ ]  ds2 = df.loc[(df['ud_asthma'] == 1) & (df['ud_obesity'] == 0) & (df['ud_copd'] == 0) & (df['ud_none'] == 0)]
     ds2.reset_index(drop=True, inplace=True)

     ds2 = ds2[ds2['spo2'].astype(bool)]
     ds2.reset_index(drop=True, inplace=True)

     dl2 = [i for j in ds2['spo2'] for i in j]
```

```
[ ]  print(len(dl2))

     2371
```

```
[ ]  dl2_array = np.array(dl2).reshape(2371,1)
```

```
[ ]  asthma_lof = LocalOutlierFactor(n_neighbors=neighbors, contamination=lof_contamination).fit_predict(dl2_array)
```

รูปที่ 3.12 แสดงการเขียนโค้ดขั้นตอนวิธี Local Outlier Factor ของผู้ป่วยโรคหอบหืด

Copd

```
[ ] ds3 = df.loc[(df['ud_copd'] == 1) & (df['ud_obesity'] == 0) & (df['ud_asthma'] == 0) & (df['ud_none'] == 0)]
    ds3.reset_index(drop=True, inplace=True)

    ds3 = ds3[ds3['spo2'].astype(bool)]
    ds3.reset_index(drop=True, inplace=True)

    dl3 = [i for j in ds3['spo2'] for i in j]
```

```
[ ] print(len(dl3))

    53
```

```
[ ] dl3_array = np.array(dl3).reshape(53,1)
```

```
[ ] copd_lof = LocalOutlierFactor(n_neighbors=neighbors, contamination=lof_contamination).fit_predict(dl3_array)
```

รูปที่ 3.13 แสดงการเขียนโค้ดขั้นตอนวิธี Local Outlier Factor ของผู้ป่วยโรคปอดอุดกั้นเรื้อรัง

Other Diseases

```
[ ] ds4 = df.loc[(df['ud_obesity'] == 0) & (df['ud_asthma'] == 0) & (df['ud_copd'] == 0) & (df['ud_none'] == 0)]
    ds4.reset_index(drop=True, inplace=True)

    ds4 = ds4[ds4['spo2'].astype(bool)]
    ds4.reset_index(drop=True, inplace=True)

    dl4 = [i for j in ds4['spo2'] for i in j]
```

```
[ ] print(len(dl4))

    72981
```

```
[ ] dl4_array = np.array(dl4).reshape(72981,1)
```

```
[ ] other_lof = LocalOutlierFactor(n_neighbors=neighbors, contamination=lof_contamination).fit_predict(dl4_array)
```

รูปที่ 3.14 แสดงการเขียนโค้ดขั้นตอนวิธี Local Outlier Factor ของผู้ป่วยโรคที่ไม่เกี่ยวข้องกับค่า SpO2

Normal

```
[ ]  ds5 = df.loc[(df['ud_none'] == 1)]
     ds5.reset_index(drop=True, inplace=True)

     ds5 = ds5[ds5['spo2'].astype(bool)]
     ds5.reset_index(drop=True, inplace=True)

     dl5 = [i for j in ds5['spo2'] for i in j]
```

```
[ ]  print(len(dl5))

     100537
```

```
[ ]  dl5_array = np.array(dl5).reshape(100537,1)
```

```
[ ]  normal_lof = LocalOutlierFactor(n_neighbors=neighbors, contamination=lof_contamination).fit_predict(dl5_array)
```

รูปที่ 3.15 แสดงการเขียนโค้ดขั้นตอนวิธี Local Outlier Factor ของผู้ป่วยที่ไม่มีโรคประจำตัว

บทที่ 4
ผลการดำเนินงาน

4.1 ผลจากการเปรียบเทียบเพื่อหาค่าพารามิเตอร์

ตารางที่ 4.1 ตารางแสดงจำนวนข้อมูล SpO2 แต่ละกลุ่มในการทดลอง

ชื่อกลุ่มผู้ป่วย	จำนวน
โรคอ้วน	240
โรคหอบหืด	2371
โรคปอดอุดกั้นเรื้อรัง	53
โรคอื่น ๆ	72981
ไม่มีโรคประจำตัว	100537

ตารางที่ 4.2 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคอ้วนด้วยวิธี Isolation forest

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
estimators = 2	191.9266	191.9266	0.0838	8.3950	0.8018	0.8018	31.1524	3.2835
estimators = 5	191.9266	191.9266	0.0838	8.3950	0.8018	0.8018	31.1524	3.2835
estimators = 10	191.9266	83.0027	0.0838	8.3950	0.8018	0.9237	31.1524	3.2835
estimators = 15	191.9266	191.9266	0.0838	8.3950	0.8018	0.8018	31.1524	3.2835
estimators = 20	191.9266	191.9266	0.0838	8.3950	0.8018	0.8018	31.1524	3.2835

estimators = 25	191.9266	191.9266	0.0838	8.3950	0.8018	0.8018	31.1524	3.2835
estimators = 30	191.9266	191.9266	0.0838	8.3950	0.8018	0.8018	31.1524	3.2835

ตารางที่ 4.3 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคหอบหืดด้วยวิธี Isolation forest

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
estimators = 2	651.7210	651.7210	651.7210	67.0867	0.4900	0.4900	0.4900	3.9428
estimators = 5	651.7210	651.7210	651.7210	67.0867	0.4900	0.4900	0.4900	3.9428
estimators = 10	651.7210	651.7210	651.7210	67.0867	0.4900	0.4900	0.4900	3.9428
estimators = 15	651.7210	651.7210	651.7210	67.0867	0.4900	0.4900	0.4900	3.9428
estimators = 20	651.7210	651.7210	651.7210	67.0867	0.4900	0.4900	0.4900	3.9428
estimators = 25	651.7210	651.7210	651.7210	67.0867	0.4900	0.4900	0.4900	3.9428
estimators = 30	651.7210	651.7210	651.7210	67.0867	0.4900	0.4900	0.4900	3.9428

ตารางที่ 4.4 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคปอดอุดกั้นเรื้อรังด้วยวิธี Isolation forest

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
estimators = 2	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
estimators = 5	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
estimators = 10	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
estimators = 15	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
estimators = 20	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
estimators = 25	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
estimators = 30	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398

ตารางที่ 4.5 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคที่ไม่เกี่ยวข้องกับค่า SpO2 ด้วยวิธี Isolation forest

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
estimators = 2	20307.7058	20307.7058	20307.7058	28355.0944	0.8111	0.8111	0.8111	0.6265
estimators = 5	20307.7058	20307.7058	20307.7058	28355.0944	0.8111	0.8111	0.8111	0.6265
estimators = 10	20307.7058	20307.7058	20307.7058	1464.9494	0.8111	0.8111	0.8111	3.7065
estimators = 15	20307.7058	20307.7058	20307.7058	1464.9494	0.8111	0.8111	0.8111	3.7065
estimators = 20	20307.7058	20307.7058	20307.7058	28355.0944	0.8111	0.8111	0.8111	0.6265
estimators = 25	20307.7058	20307.7058	20307.7058	1464.9494	0.8111	0.8111	0.8111	3.7065
estimators = 30	20307.7058	20307.7058	20307.7058	1464.9494	0.8111	0.8111	0.8111	3.7065

ตารางที่ 4.6 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยที่ไม่มีโรคประจำตัวด้วยวิธี Isolation forest

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
estimators = 2	26470.3216	26470.3216	40315.2137	40315.2137	0.7750	0.7750	0.6023	0.6023
estimators = 5	26470.3216	26470.3216	40315.2137	26470.3216	0.7750	0.7750	0.6023	0.7750
estimators = 10	26470.3216	26470.3216	26470.3216	40315.2137	0.7750	0.7750	0.7750	0.6023
estimators = 15	26470.3216	26470.3216	40315.2137	40315.2137	0.7750	0.7750	0.6023	0.6023
estimators = 20	26470.3216	26470.3216	26470.3216	26470.3216	0.7750	0.7750	0.7750	0.7750
estimators = 25	26470.3216	26470.3216	40315.2137	40315.2137	0.7750	0.7750	0.6023	0.6023
estimators = 30	26470.3216	26470.3216	26470.3216	26470.3216	0.7750	0.7750	0.7750	0.7750

จากตารางที่ 4.2 ถึง 4.6 พบว่าเมื่อเฉลี่ยผลลัพธ์จากทุกโรค กรณี Isolation forest จะมีประสิทธิภาพดีที่สุด หากกำหนดค่า Estimators และ Contamination เท่ากับ 2 และ 0.025 ตามลำดับ

ตารางที่ 4.7 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคอ้วนด้วยวิธี Local outlier factor

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
neighbors = 2	191.9266	191.9266	191.9266	191.9266	0.8018	0.8018	0.8018	0.8018
neighbors = 15	191.9266	83.0027	83.0027	8.3950	0.8018	0.9237	0.9237	3.2835
neighbors = 25	191.9266	83.0027	83.0027	8.3950	0.8018	0.9237	0.9237	3.2835
neighbors = 35	191.9266	83.0027	83.0027	8.3950	0.8018	0.9237	0.9237	3.2835

ตารางที่ 4.8 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคหอบหืดด้วยวิธี Local outlier factor

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
neighbors = 2	299.6798	299.6798	299.6798	299.6798	0.4843	0.4843	0.4843	0.4843
neighbors = 15	370.7128	370.7128	370.7128	370.7128	0.5771	0.5771	0.5771	0.5771
neighbors = 25	370.7128	370.7128	370.7128	370.7128	0.5771	0.5771	0.5771	0.5771

neighbors = 35	370.7128	370.7128	370.7128	370.7128	0.5771	0.5771	0.5771	0.5771

ตารางที่ 4.9 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคปอดอุดกั้นเรื้อรังด้วยวิธี Local outlier factor

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
neighbors = 2	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
neighbors = 15	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
neighbors = 25	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398
neighbors = 35	6.0177	6.0177	6.0177	6.0177	0.3398	0.3398	0.3398	0.3398

ตารางที่ 4.10 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยโรคที่ไม่เกี่ยวข้องกับค่า SpO2 ด้วยวิธี Local outlier factor

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
neighbors = 2	13308.4015	13308.4015	13308.4015	13308.4015	0.4063	0.4063	0.4063	0.4063
neighbors = 15	29266.7338	29266.7338	29266.7338	29266.7338	0.5116	0.5116	0.5116	0.5116

neighbors = 25	36308.0488	36308.0488	36308.0488	36308.0488	0.4419	0.4419	0.4419	0.4419
neighbors = 35	32461.3213	32461.3213	32461.3213	32461.3213	0.5824	0.5824	0.5824	0.5824

ตารางที่ 4.11 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ในแต่ละค่าพารามิเตอร์ข้อมูล SpO2 ของผู้ป่วยที่ไม่มีโรคประจำตัวด้วยวิธี Local outlier factor

	Calinski-Harabasz Index				Davies-Bouldin Index			
	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1	contamination 0.025	contamination 0.050	contamination 0.075	contamination 0.1
neighbors = 2	7144.3224	7144.3224	7144.3224	7144.3224	0.3479	0.3479	0.3479	0.3479
neighbors = 15	29547.0566	29547.0566	29547.0566	29547.0566	0.4864	0.4864	0.4864	0.4864
neighbors = 25	40040.7809	40040.7809	40040.7809	40040.7809	0.5219	0.5219	0.5219	0.5219
neighbors = 35	45174.5804	45174.5804	45174.5804	45174.5804	0.4695	0.4695	0.4695	0.4695

จากตารางที่ 4.7 ถึง 4.11 พบว่าเมื่อเฉลี่ยผลลัพธ์จากทุกโรค กรณี Local outlier factor จะมีประสิทธิภาพดีที่สุด หากกำหนดค่า Neighbors และ Contamination เท่ากับ 35 และ 0.025 ตามลำดับ

4.2 ผลการวัดประสิทธิภาพจากการเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index

ตารางที่ 4.12 ตารางเปรียบเทียบค่า Calinski-Harabasz Index และ Davies-Bouldin Index ของวิธี Isolation forest และวิธี Local outlier factor

	จำนวนข้อมูล	Calinski-Harabasz Index		Davies-Bouldin Index	
		Isolation forest	LOF	Isolation forest	LOF
โรคอ้วน	240	191.9266	191.9266	0.8018	0.8018
โรคหอบหืด	2371	651.7210	370.7128	0.4900	0.5771
โรคปอดอุดกั้นเรื้อรัง	53	6.0177	6.0177	0.3398	0.3398
โรคอื่น ๆ	72981	20307.7058	32461.3213	0.8111	0.5824
ไม่มีโรคประจำตัว	100537	26470.3216	45174.5804	0.7750	0.4695

จากตาราง 4.10 พบว่าวิธี Local outlier factor มีผลลัพธ์ประสิทธิภาพดีที่สุด เมื่อข้อมูลมีปริมาณมาก เช่น กลุ่มผู้ป่วยโรคอื่น ๆ และ กลุ่มผู้ป่วยไม่มีโรคประจำตัว แต่เมื่อข้อมูลมีจำนวนปริมาณน้อย ผลลัพธ์วิธี Isolation forest จะมีประสิทธิภาพดีกว่าเพียงเล็กน้อยหรือไม่มีความแตกต่างเลย

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ในการทดลองครั้งนี้ เมื่อกำหนดค่าพารามิเตอร์ Estimators และ Contamination ของวิธี Local outlier factor มีค่าเท่ากับ 2 และ 0.025 ตามลำดับ และกำหนดค่าพารามิเตอร์ Neighbors และ Contamination ของวิธี Isolation forest เท่ากับ 35 และ 0.025 ตามลำดับ

เมื่อมีการนำวิธีการเปรียบเทียบผลการทดลอง ได้แก่ Calinski-Harabasz Index และ Davies-Bouldin Index พบว่าผลที่ได้จากการทดลองวิธี Local outlier factor มีประสิทธิภาพดีกว่า

กรณีที่ข้อมูลมีจำนวนน้อย เช่น จำนวนข้อมูล SpO2 ของคนไข้โรคหืดที่ในการทดลองครั้งนี้มีเพียง 2371 ข้อมูล ผลลัพธ์วิธี Isolation forest มีประสิทธิภาพมากกว่า

เมื่อพิจารณาจากฐานข้อมูลในปัจจุบันที่จำนวนข้อมูลยังคงเพิ่มขึ้นอย่างต่อเนื่อง วิธี Local outlier factor มีความเหมาะสมกับการแก้ไขปัญหาของการทดลองนี้

5.2 ข้อเสนอแนะ

หากต้องการเพิ่มความแม่นยำของผลการทดลอง สามารถทำได้โดยปรับแต่งค่าพารามิเตอร์ที่ใช้ในขั้นตอนหาค่าของ Local outlier factor และ Isolation forest ให้มีความละเอียดมากขึ้น

บรรณานุกรม

[1] Arnatchai Techaviseschai. 2022. Anomaly Detection with Isolation Forest: แยกข้อมูลผิดปกติง่าย ๆ ด้วย Isolation. [ออนไลน์]. เข้าถึงได้จาก : https://bigdataexperience.org/anomaly-detection-with-isolation-forest-แยกข้อมูลผิดปกติ/

[2] Dario Radečić. 2022. Introducing Shiny for Python – R Shiny Now Available in Python. [ออนไลน์]. เข้าถึงได้จาก : https://python-bloggers.com/2022/08/introducing-shiny-for-python-r-shiny-now-available-in-python/

[3] ชิตพงษ์ กิตตินราดร. 2563. Anomaly Detection. [ออนไลน์]. เข้าถึงได้จาก : https://guopai.github.io/ml-blog13.html

[4] Supanat Jintawatsakoon. 2020. Isolation Forest — อะไรเอ่ยไม่เข้าพวก ?. [ออนไลน์]. เข้าถึงได้จาก : https://supanatj.medium.com/isolation-forest-อะไรเอ่ยไม่เข้าพวก-38aabaf612a8

[5] Chakrit Phain. 2020. Local Outlier Factor (LOF). [ออนไลน์]. เข้าถึงได้จาก : https://www.softnix.co.th/2020/03/10/anomaly-detection-part-4-local-outlier-factor/

[6] Scikit-learn. Anomaly detection with Local Outlier Factor (LOF). [ออนไลน์]. เข้าถึงได้จาก : https://scikit-learn.org/0.19/auto_examples/neighbors/plot_lof.html

[7] Yoswat Suntonarom. 2020. บันทึก training data science EP 10: Cluster – เชื่อมก้อนให้เป็นกลุ่ม. [ออนไลน์]. เข้าถึงได้จาก : https://www.bluebirz.net/th/note-of-data-science-training-ep-10-th/

[8] นภารัตน์ อมรพุฒิสถาพร. โรคปอดอุดกั้นเรื้อรัง. [ออนไลน์]. เข้าถึงได้จาก : https://www.rama.mahidol.ac.th/med/sites/default/files/public/pdf/medicinebook1/COPD.pdf

[9] สุทธิพงศ์ ผ่องแผ้ว. 2561. “การศึกษากระบวนการแบ่งกลุ่มเมล็ดพันธุ์ด้วยข้อมูลโครงสร้างของเมล็ด.” สารนิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ , มหาวิทยาลัยศรีนครินทรวิโรฒ.

[10] Liu, F.T. Ting, K.M. and Zhou, Z.H. 2008. Isolation forests. In Proceedings of International Conference on Data Mining (2008). 413-422. doi: 10.1109/ICDM.2008.17.

[11] Kapur, V.K. Wilsdon, A.G. Au, D. Avdalovic, M. Enright, P. Fan, V.S. et al. 2013. “Obesity Is Associated With a Lower Resting Oxygen Saturation in the Ambulatory Elderly: Results From the Cardiovascular Health Study.” Respiratory care. 58(3) : 831-837. doi: 10.4187/respcare.02008.