**รายงานปฏิบัติงานสหกิจศึกษา**

**ณ ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่**

# การวิเคราะห์ความคิดเห็นจากลูกค้าเพื่อพัฒนามาตรการการดำเนินการอย่างรวดเร็วสำหรับโรงพยาบาลระดับตติยภูมิ

# (Voice of Customer Sentiment Analysis to Develop a Rapid Action Protocol for the Tertiary Care Hospital)

**นาย จาตุรนต์ วงศ์เศรษฐี**

**610510679**

**สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต**

**สาขาวิชาวิทยาการคอมพิวเตอร์**

**คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่**

**ปีการศึกษา 2564**

# การวิเคราะห์ความคิดเห็นจากลูกค้าเพื่อพัฒนามาตรการการดำเนินการอย่างรวดเร็วสำหรับโรงพยาบาลระดับตติยภูมิ

**(Voice of Customer Sentiment Analysis to Develop a Rapid Action Protocol for the Tertiary Care Hospital)**

นาย จาตุรนต์ วงศ์เศรษฐี

610510679

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2564

คณะกรรมการสอบสหกิจศึกษา

ประภาพร เตชอังกูร ..... ประธานกรรมการ

( อ.ดร.ประภาพร เตชอังกูร )

จักรเมธ ..... กรรมการ

( ผศ.ดร.จักรเมธ บุตรกระจ่าง )

วันที่ 29 เดือน ก.ค. พ.ศ. 2564

# หนังสือยินยอมให้ข้อมูลเพื่อการศึกษา และเผยแพร่ผลการศึกษาสหกิจศึกษา

วันที่ 30 กันยายน 2564

หนังสือฉบับนี้ ข้าพเจ้า นพ. ธีรพัฒน์ ตันพิริยะกุล ในนาม บริษัท ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่อยู่ 110/392 อาคารศรีพัฒน์ ถนนอินทวโรรส ตำบลศรีภูมิ อำเภอเมือง จังหวัดเชียงใหม่ 50200 ขอทำหนังสือฉบับนนี้เพื่อเป็นหลักฐานแสดงว่า ข้าพเจ้าได้รับทราบและยินยอมให้ นายจาตุรนต์ วงศ์เศรษฐี รหัสนักศึกษา 610510679 สังกัดภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่ ผู้เข้าร่วมโครงการสหกิจศึกษา ในภาคการเรียนที่ 1 ปีการศึกษา 2564 ณ.หน่วยงานของข้าพเจ้า ตั้งแต่วันที่ 19 เมษายน 2564 ถึงวันที่ 30 กันยายน 2564 ผู้ศึกษาสามารถเก็บข้อมูล ณ หน่วยงานของข้าพเจ้าตามคำชี้แจงของผู้ศึกษา และอนุญาตให้นำผลการศึกษาและปฏิบัติงานเผยแพร่สู่สาธารณะได้

ทั้งนี้หากผู้ศึกษาได้กระทำภายในขอบเขตอำนาจของหนังสือยินยอมฉบับนี้ให้มีผลสมบูรณ์และชอบด้วยกฎหมายทุกประการ และหากมีผลกระทบหรือเกิดความเสียหายขึ้นจะไม่มีการเรียกร้องแต่อย่างใด เพื่อเป็นหลักฐานแห่งความยินยอมนี้ ข้าพเจ้าได้ลงลายมือชื่อและประทับตราไว้

นพ. ธีรพัฒน์ ตันพิริยะกุล
ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่

# กิตติกรรมประกาศ

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่ ตั้งแต่วันที่ 19 เมษายน 2564 ถึงวันที่ 30 กันยายน 2564 ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์ที่มีคุณค่าจากการฝึกปฏิบัติงานสหกิจศึกษารายงานสกิจศึกษาฉบับนี้สำเร็จได้ด้วยดี เนื่องด้วยการความร่วมมือจากหลายฝ่าย ดังนี้

1) กวิสรา ทองดีเลิศ ตำแหน่ง โปรแกรมเมอร์
2) ปรียานุช มูลถี ตำแหน่ง โปรแกรมเมอร์
3) อาจารย์ ดร.ประภาพร เตชอังกูร อาจารย์ที่ปรึกษาสหกิจศึกษา
4) ผู้ช่วยศาสตราจารย์ ดร.จักรเมธ บุตรกระจ่าง กรรมการสอบสหกิจศึกษา

รวมถึงบุคคลท่านอื่นที่ไม่ได้กล่าวนามทุกท่านที่ได้กรุณาให้ความรู้ ปรึกษาคำแนะนำช่วยเหลือและเป็นประโยชน์แก่ข้าพเจ้าในระหว่างที่ปฏิบัติงาน ข้าพเจ้าจึงขอขอบคุณทุกท่านที่ได้มีส่วนร่วมในการให้ความรู้ คำแนะนำ และการช่วยเหลือจนรายงานฉบับนี้เสร็จสมบูรณ์

สุดท้ายนี้ขอขอบพระคุณภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่ และศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่ให้โอกาสในการมาฝึกงานแบบสหกิจศึกษาในครั้งนี้ ทำให้ได้รับความรู้ และประสบการณ์นอกจากเรียนที่มหาวิทยาลัย ซึ่งข้าพเจ้าหวังอย่างยิ่งว่าจะได้นำไปใช้ประโยชน์ในชีวิตประจำวัน และในการทำงานต่อไป

จาตุรนต์ วงศ์เศรษฐี

**หัวข้อสหกิจศึกษา** การวิเคราะห์ความคิดเห็นจากลูกค้าเพื่อพัฒนามาตรการการดำเนินการอย่างรวดเร็วสำหรับโรงพยาบาลระดับตติยภูมิ
**สถานประกอบการ** ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่
**ผู้ดำเนินการศึกษา** 610510679 นายจาตุรนต์ วงศ์เศรษฐี
**หลักสูตร** วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
**อาจารย์ที่ปรึกษาสหกิจ** อาจารย์ ดร.ประภาพร เตชอังกูร

# บทคัดย่อ

เมื่อไม่นานมานี้ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้วางแผนที่จะปรับปรุงสภาพแวดล้อมของโรงพยาบาลและบริการทางการแพทย์ตามความคิดเห็นของลูกค้าที่รับเข้ามามากมายในแต่ละวัน ก่อนหน้านี้ทางศูนย์ศรีพัฒน์ฯ ใช้พนักงานในการจำแนกความพึงพอใจของลูกค้าทำให้ในกระบวนการนี้เกิดความล่าช้าเป็นอย่างมากเนื่องจากมีความคิดเห็นของลูกค้าเป็นจำนวนมากส่งผลให้เกิดการปรับปรุงโรงพยาบาลล่าช้าตามไปด้วย เพื่อแก้ไขปัญหานี้งานวิจัยนี้จึงมีจุดมุ่งหมายเพื่อปรับปรุงการวิเคราะห์ความความคิดเห็นของลูกค้าของศูนย์ศรีพัฒน์ฯ โดยใช้เทคนิคการเรียนรู้ของเครื่องได้แก่ ต้นไม้ตัดสินใจ การสุ่มป่าไม้ ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเค นาอีฟเบย์ ต้นไม้ที่ไล่ระดับสี สถาปัตยกรรมเพอร์เซปตรอนแบบหลายชั้น การวิเคราะห์การถดถอยโลจิสติก และ ซัพพอร์ตเวกเตอร์-แมชชีน โดยการฝึกฝนโมเดลการเรียนรู้ของเครื่องด้วยข้อมูลความคิดเห็นของลูกค้าจากฐานข้อมูลของศูนย์ศรีพัฒน์ฯ ในการจำแนกประเภทสามารถวิเคราะห์ความคิดเห็นในเชิงบวกหรือเชิงลบได้ ยิ่งไปกว่านั้นเราได้สร้างการจำแนกประเภทที่สามารถระบุความคิดเห็นเชิงลบว่าเป็นความเสียหายเล็กน้อยหรือรุนแรง

ในการศึกษานี้ผลการจำแนกความคิดเห็นว่าดีหรือไม่ดีพบว่าการใช้โมเดลที่ได้จากการโหวตของโมเดลที่สร้างจากเทคนิค การสุ่มป่าไม้ มีประสิทธิภาพที่ดีกว่าเทคนิคการเรียนรู้ของเครื่องแบบอื่นๆ โดยการจำแนกประเภทได้ผลลัพธ์ค่าความถูกต้องอยู่ที่ 0.930 และค่า F1-Score อยู่ที่ 0.929 นอกจากนี้เราได้สร้างโมเดลเพื่อจำแนกความคิดเห็นเชิงลบในระดับเล็กน้อย หรือรุนแรง โดยใช้การโหวตของโมเดลที่สร้างจากเทคนิค การสุ่มป่าไม้ ต้นไม้ที่ไล่ระดับสี สถาปัตยกรรมเพอร์เซปตรอนแบบหลายชั้น ซัพพอร์ตเวกเตอร์แมชชีน และการถดถอยโลจิสติก ได้ประสิทธิภาพที่ดีที่สุดด้วยค่า F1-Score อยู่ที่ 0.853 และค่าความถูกต้องอยู่ที่ 0.853

| | |
|---|---|
| **Title** | Voice of Customer Sentiment Analysis to Develop a Rapid Action Protocol for the Tertiary Care Hospital |
| **Company** | Sriphat Medical Center Faculty of Medicine Chiang Mai University |
| **Name** | 610510679 Mr. Jaturon Wongsettee |
| **Degree** | Bachelor of Science in Computer Science |
| **Advisor** | Prapaporn Techa-Angkoon, Ph.D. |

# Abstract

Recently, Sriphat Medical Center, Faculty of Medicine, Chiang Mai University has planned to improve hospital environment and medical services according to customer feedback. There are many feedbacks from customer everyday. Previously, Sriphat Medical Center used their customer service teams to classify customer satisfaction. The process of classifying feedbacks was delayed due to the large number of feedbacks. As a result, the hospital improvement was delayed. To address this issue, this research aims to improve the sentiment analysis of Sriphat Medical Centers' customer feedbacks using machine learning techniques: Decision Tree, Random Forest, k-Nearest Neighbors, Naïve Bayes, Gradient Boosted Trees, Multi-Layer Perceptron, Logistic Regression, Support Vector Machines. By training the machine learning models with the customer feedback data from Sriphat Medical Center's database, the classifier can analyze the opinions as positive or negative. Moreover, we built the classifier that can determine the negative opinions as mild or severe damage.

In this study, the result of classifying opinions as good or bad showed that using voting of models generated from Random Forest technique performed better than other machine learning techniques. The classifier achieved the accuracy with 0.930 and F1-Score with 0.929. Furthermore, we built the models to classify the negative opinions as mild or severe level. Using the voting technique from Random Forest, Gradient Boosting Tree, Multi-Layer Perceptron, Support Vector Machines, and Logistic Regression achieved better performance with F1-Score at 0.853 and accuracy at 0.853.

# สารบัญ

**หัวข้อ** **หน้า**

# สารบัญ(ต่อ)

**หัวข้อ** **หน้า**

# สารบัญรูป

# สารบัญตาราง

ตารางที่ หน้า

# บทที่ 1
# บทนำ

ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่เล็งเห็นถึงความสำคัญของความคิดเห็นและผูกพันของลูกค้าที่มีองค์กร ดังนั้นต้องตอบสนองความต้องการของลูกค้าให้ได้มากที่สุดจึงพัฒนาระบบจำแนกประเภทความคิดเห็น และแจ้งเตือนได้ทันที ทำให้ประหยัดเวลากว่าการใช้คนจำแนกและหาทางแก้ไขข้อผิดพลาดขององค์กรก่อนที่ลูกค้าจะแปรเปลี่ยนความผูกพันเป็นความไม่พอใจ

## 1.1. ข้อมูลสถานประกอบการ

### 1.1.1. ชื่อองค์กร

ชื่อภาษาไทย ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่

ชื่อภาษาอังกฤษ Sripat Medical Center Faculty of Medicine Chiang Mai University

### 1.1.2. ระยะเวลาประกอบการ

ตั้งแต่วันที่ 19 เมษายน 2564 ถึง 30 กันยายน 2564

### 1.1.3. ลักษณะองค์กร

ศูนย์ศรีพัฒน์ฯ เป็นหน่วยงานหนึ่งในคณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่ มีพันธกิจหลักในการบริหารจัดการเพื่อให้บริการด้านสุขภาพ โดยสร้างนวัตกรรมในการดูแลรักษาพยาบาล และผลิตภัณฑ์บริการทางการแพทย์ด้วยทีมแพทย์ผู้เชี่ยวชาญเฉพาะทางในแต่ละสาขา

## 1.2. ตำแหน่งและลักษณะงานที่ได้รับมอบหมาย

### 1.2.1. ตำแหน่งงานที่ปฏิบัติ

ผู้พัฒนาซอฟต์แวร์ (Software Developer)

### 1.2.2. งานที่ได้รับมอบหมาย

ทำ Classification Models สำหรับจำแนกประเภทความคิดเห็นของลูกค้า เพื่อแจ้งเตือนผู้ดูแลระบบแบบ Real-Time เมื่อลูกค้าแสดงความคิดเห็นด้านลบที่มีผลกระทบร้ายแรง และนำข้อมูลความคิดเห็นของลูกค้าแสดงใน Dashboard บน Web Application

**1.2.3. ลักษณะงานที่ปฏิบัติ**

1. รับความต้องการของผู้ใช้
2. ศึกษาลักษณะความคิดเห็นลูกค้าในอดีตเพื่อทำการจำแนกประเภท และทำความสะอาดข้อมูล
3. ทำ Classification Models สำหรับจำแนกประเภทความคิดเห็นของลูกค้า และทำระบบแจ้งเตือนผ่าน Line Notification
4. ทำ Web Application เพื่อแสดง Dashboard สำหรับข้อมูลความคิดเห็นของลูกค้า
5. ทดสอบความสมบูรณ์ของระบบ และทำการแก้ไข

**1.3. หลักการและเหตุผล**

Voice of Customer เป็นการรับความคิดเห็น และความปรารถนาของลูกค้าที่มีต่อองค์กร เพื่อให้ตอบสนองความต้องการของลูกค้าได้ และทำให้ลูกค้าเกิดความผูกพันธ์กับองค์กร แต่เป็นเรื่องยากที่จะรับรู้ความปรารถนาของลูกค้าได้ทุกคน และตอบสนองได้ทันท่วงทีก่อนที่ลูกค้าไม่พอใจมากขึ้นจนแตกหักกับองค์กร เนื่องจากลูกค้ามีจำนวนที่มากเกินกว่าบุคลากรภายในองค์กรจะทำการรับฟังได้ทุกความปรารถนาในเวลาอันสั้น ทำให้ต้องมีเครื่องมือในการช่วยเหลือเพื่อทุ่นแรง และเพิ่มความรวดเร็วในการรับฟังความปรารถนาของลูกค้า

เพื่อแก้ไขข้อผิดพลาดให้ได้ทันท่วงที ดังนั้นผู้จัดทำจึงได้จัดทำ Web Application เพื่อใช้ในการจำแนกความคิดเห็นด้านดี และด้านไม่ดีของลูกค้า โดยด้านไม่ดีแบ่งระดับความเสียหายออกเป็นไม่รุนแรง และรุนแรง จากนั้นให้ความคิดเห็นด้านไม่ดีในระดับรุนแรงแจ้งเตือนให้ผู้ที่รับผิดชอบทราบในทันที

**1.4. วัตถุประสงค์**

1. เพื่อให้สามารถรับรู้ความคิดเห็นด้านลบของลูกค้าที่มีผลกระทบร้ายแรงได้แบบทันที
2. เพื่อให้สามารถจัดการกับความคิดเห็นด้านลบของลูกค้าที่มีผลกระทบร้ายแรงได้ทันท่วงที

**1.5. ประโยชน์ที่ได้รับ**

1. ช่วยให้พนักงานลดเวลางานที่ใช้ในการจำแนกประเภทความคิดเห็นของลูกค้า
2. เพิ่มความรวดเร็วในการแก้ไขปัญหาที่ลูกค้าไม่พึงพอใจ
3. รักษาความสัมพันธ์ของลูกค้าที่มีต่อองค์กร

**1.6. ขอบเขต**

ระบบจำแนกประเภทความคิดเห็น และแจ้งเตือนได้ทันทีนั้นสามารถเข้าถึงได้เฉพาะผู้ที่มีตำแหน่งเป็นผู้ดูและระบบ แลประกอบด้วยฟีเจอร์หลัก ๆ ดังต่อไปนี้

1. จำแนกประเภทความคิดเห็นของลูกค้า ประเภทความคิดเห็นของลูกค้าประกอบไปด้วยความคิดเห็นด้านดี และด้านไม่ดีของลูกค้า โดยด้านไม่ดีแบ่งระดับความเสียหายออกเป็นไม่รุนแรง และรุนแรง เพื่อที่จะให้จำแนกได้แบบอัติโนมัติจึงต้องใช้ Machine Learning มาจำแนกเมื่อลูกค้าป้อนความคิดเห็นเข้าสู่ระบบ
2. แจ้งเตือนแบบ Real-Time เมื่อจำแนกประเภทได้ความคิดเห็นด้านไม่ดีระดับความเสียหายรุนแรงจะต้องแจ้งเตือนผู้ดูแลระบบผ่าน Line Notification ทันที
3. แสดง Dashboard หน้า Dashboard ของผู้ดูแลระบบจะต้องมีกราฟแสดงอัตราส่วนระหว่างความคิดเห็นประเภทด้านดี และไม่ดี กราฟแสดงอัตราส่วนระหว่างความคิดเห็นด้านไม่ดีระดับไม่รุนแรง และระดับรุนแรง กราฟเปรียบเทียบความคิดเห็นประเภทด้านดี และไม่ดีในแต่ละปีงบประมาณ และกราฟเปรียบเทียบความคิดเห็นประเภทด้านดี และไม่ดีในแต่ละเดือนตามปีงบประมาณนั้นๆ
4. ค้นหาข้อมูล ข้อมูลกราฟที่แสดงบนหน้า Dashboard ของผู้ดูแลระบบจะต้องสามารถค้นหาตามช่วงเวลาที่ต้องการได้

**1.7. เครื่องมือและอุปกรณ์ที่ใช้**

**1.7.1. ฮาร์ดแวร์ที่ใช้ในการพัฒนา**

เครื่องคอมพิวเตอร์ Server หน่วยประมวลผล Intel(R) Xeon(R) Gold 6226 CPU @ 2.70GHz 8 Core หน่วยความจำหลัก 16 GiB

เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผล Intel(R) Core(TM) i3-4010U CPU @ 1.70GHz 2 Core หน่วยความจำหลัก 8 GB DDR3 หน่วยความจำรอง 466 GB กราฟิกการ์ด Intel(R) HD Graphics Family (128 MB Memory)

**1.7.2. ซอฟต์แวร์ที่ใช้ในการพัฒนา**

1. Visual Studio Code โปรแกรมสำหรับเขียนหรือแก้ไข Source Code
2. Anaconda 3 โปรแกรมสำหรับกำหนด Python Environment modules
3. Figma ใช้สำหรับออกแบบหน้าต่าง Website
4. Laravel 7 Framework ใช้ในการพัฒนา Web Application รูปแบบ MVC
5. Google Chrome เป็นเว็บเบราว์เซอร์ที่ใช้ในการแสดงผล Web application

6. Xampp เป็นโปรแกรมสำหรับสร้างฐานข้อมูล Localhost

7. Line Application ใช้สำหรับรับข้อความแจ้งเตือนความคิดเห็นแก่ผู้ดูแลระบบ

**1.7.3. ภาษาที่ใช้ในการพัฒนา**

1. Python ใช้สำหรับพัฒนา Classification Models

2. HTML, CSS, Javascript, PHP ใช้สำหรับพัฒนา Web Application และติดต่อฐานข้อมูล

**1.8. แผนปฏิบัติงานสหกิจ**

ระยะเวลาการพัฒนาระบบและทำงานอื่นๆ ที่ได้รับมอบหมาย เริ่มตั้งแต่วันที่ 19 เมษายน 2564 ถึง 30 กันยายน 2564 รวมเป็นระยะเวลา 5 เดือน 19 วัน แสดงรายละเอียดดังตาราง 1.1

**ตารางที่ 1. 1 แผนการดำเนินงานและระยะเวลาในการพัฒนาระบบ**

| ลำดับที่ | หัวข้องาน | พ.ศ. 2564 | | | | | |
|---|---|---|---|---|---|---|---|
| | | เม.ย. | พ.ค. | มิ.ย. | ก.ค. | ส.ค. | ก.ย. |
| 1 | ศึกษาการใช้งาน Laravel 7 Framework | | | | | | |
| 2 | ทำงานอื่นที่ได้รับมอบหมาย | | | | | | |
| 3 | รวบรวมความต้องการของผู้ใช้ และออกแบบแบบจำลองของผลลัพธ์ | | | | | | |
| 4 | ศึกษาเทคนิคในการทำ Classification Models | | | | | | |
| 5 | ทำ Classification Models สำหรับจำแนกประเภทความคิดเห็นของลูกค้า | | | | | | |
| 6 | ทำ Dashboard และระบบแจ้งเตือนแบบ Real-Time | | | | | | |
| 7 | ทดสอบและแก้ไขระบบ | | | | | | |

| ลำดับที่ | หัวข้องาน | พ.ศ. 2564 | | | | | |
|---|---|---|---|---|---|---|---|
| | | เม.ย. | พ.ค. | มิ.ย. | ก.ค. | ส.ค. | ก.ย. |
| 8 | เขียนบทความวิชาการ และจัดทำเอกสาร รายงานปฏิบัติสหกิจศึกษา | | | | | | |

# บทที่ 2
# หลักการและทฤษฎีที่เกี่ยวข้อง

## 2.1. การรับฟังเสียงของลูกค้า (Voice of Customer)

การรับฟังเสียงของลูกค้า [1] เป็นกระบวนการที่ธุรกิจสามารถเข้าใจความต้องการของลูกค้าได้อย่างแท้จริง กระบวนการ “รับฟังเสียงของลูกค้า” ที่มีประสิทธิภาพมักเริ่มจากการจำแนกลูกค้าเป็นกลุ่มต่างๆ เช่น กลุ่มวัยทำงาน กลุ่มผู้สูงอายุ กลุ่มที่ใช้โซเชียลมีเดีย (Social Media) กลุ่มพื้นที่เมือง หรือส่วนภูมิภาค โดยอำนวยความสะดวกในการมีช่องทางที่หลากหลาย ขึ้นอยู่กับความสะดวกในแต่ละช่องทางการแสดงความ-คิดเห็นของลูกค้า เช่น ทางโทรศัพท์ ทางจดหมาย ทางเว็บไซต์ทางโซเชียลมีเดีย (Social Media) และทางพนักงาน เป็นต้น ระบบรับฟังเสียงที่ดีควรมีระบบการบันทึกข้อมูล ระบบการป้อนข้อมูลที่เหมาะสมและระบบการประมวลผลที่รวดเร็ว เพื่อทำให้องค์กรสามารถกำหนดวิธีการและกลยุทธ์การตลาดได้อย่างถูกต้องแม่นยำ ในอันที่จะทำให้ลูกค้าเป้าหมายซื้อและใช้ผลิตภัณฑ์หรือบริการได้ตรงตามความต้องการ ทั้งการซื้อในปัจจุบันและการใช้ต่อไปในอนาคต

กระบวนการค้นหา “เสียงของลูกค้า” [1] เริ่มจากการรวบรวมข้อมูล ข้อเท็จจริงหรือพฤติกรรมที่เกี่ยวกับลูกค้าอย่างต่อเนื่อง ข้อมูลที่สำคัญ ได้แก่ ความต้องการของลูกค้าที่อยากได้สินค้าหรือบริการประเภทต่างๆ ปัญหาและอุปสรรคในการรับบริการที่เป็นประโยชน์ในการพัฒนาปรับปรุงบริการต่อไป “เสียงของลูกค้า” อาจได้มาจากการสำรวจตลาดและพฤติกรรมผู้บริโภค การสอบถามกลุ่มตัวอย่างหรือกลุ่มเป้าหมาย การบันทึกการขายหรือรายงานการขาย บันทึกข้อร้องเรียนของลูกค้า และข้อมูลภาคสนาม

หลายองค์กรขนาดใหญ่มีการจัดทำเสียงของลูกค้าเป็นระบบงาน (Web-based Application) [1] เพื่อประโยชน์ในการปรับปรุงพัฒนาการบริการให้สะดวกรวดเร็วและทันท่วงที มีการพัฒนาระบบงานให้สอดคล้องกับขั้นตอนการทำงานขององค์กร เพื่อความเป็นมาตรฐานสำหรับการบริหารงานองค์กร มีการกำหนด หลักการและแนวคิดเกี่ยวกับข้อตกลงระดับการให้บริการ (Service Level Agreement) เพื่อเป็นข้อตกลงร่วมกันในการให้บริการแก่ลูกค้า และบางแห่งได้กำหนดปริมาณการจัดการตอบเรื่องร้องเรียนจากเสียงของลูกค้าเป็น ดัชนีชี้วัดความสำเร็จ (KPI) ในการประเมินผลงานของบุคลากรภายในองค์กร

## 2.2. การจำแนกประเภท (Classification)

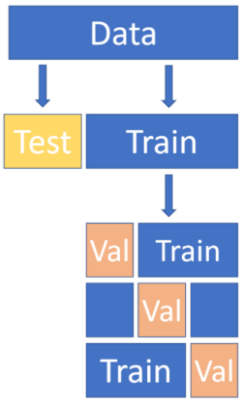
การจำแนกประเภท [2] เป็นการจำแนกข้อมูลออกเป็นประเภทต่างๆ ตามที่ กำหนดคำตอบ (Label) ได้กำหนดไว้ โดย การเรียนรู้ของเครื่อง (Machine Learning) ในส่วนของ การจำแนกประเภทจะให้คำตอบเป็นคำตอบที่ถูกกำหนดไว้เท่านั้น ไม่สามารถให้คำตอบที่นอกเหนือจากที่กำหนดไว้ในชุด-

ฝึกฝนหรือออกมาเป็นตัวเลขที่ผ่านการคำนวณได้ แต่การทำ การจำแนกประเภท ก็อาจจะไม่ถูกต้องเสมอไป บางครั้งเราอาจทำนายผิดให้ผู้ชายกลายเป็นผู้หญิงก็เป็นได้ เพราะฉะนั้นทุกๆครั้งที่ทำ การจำแนกประเภท ต้องทำการประเมินโมเดลเสมอ

### 2.3. การทำความสะอาดข้อมูล (Data cleansing)

การทำความสะอาดข้อมูล (Data cleansing) [3] คือ กระบวนการตรวจสอบ การแก้ไข หรือการลบ เพื่อให้รายการข้อมูลที่ไม่ถูกต้องออกไปจากชุดข้อมูล ตารางหรือฐานข้อมูล ซึ่งเป็นหลักสำคัญของฐานข้อมูล เพราะหมายถึงความไม่สมบูรณ์ ความไม่ถูกต้อง ความไม่สัมพันธ์กับข้อมูลอื่น ๆ เป็นต้น จึงทำให้ผู้เชี่ยวชาญหลาย ๆ คนมองว่าการล้างข้อมูลเป็นสิ่งสำคัญที่สุดในการจัดการกับคุณภาพของข้อมูล

### 2.4. การแยกชุดข้อมูล (Training/Test Set Split)




**รูปที่ 2. 1 การแยกชุดข้อมูล**

การแยกชุดข้อมูล [4] จากรูปที่ 2.1 อธิบายได้ว่า

- ชุดข้อมูลสอน (Training Set) คือชุดข้อมูลที่ใช้สอนโมเดลให้มีความสามารถในการแบ่งแยกประเภท
- ชุดข้อมูลตรวจสอบ (Validation Set) ใช้สำหรับทดสอบหาผลลัพธ์เปรียบเทียบหลังจากสอนโมเดลว่าโมเดลทำงานได้ดีแค่ไหน และหลังจากจูนแต่ละครั้งโมเดลไหนทำงานได้ดีกว่ากัน
- ชุดข้อมูลทดสอบ (Test Set) ใช้สำหรับทดสอบหลังจากได้โมเดลที่ดีที่สุดมาแล้วว่าโมเดลจะทำงานได้ดีแค่ไหนกับข้อมูลที่ไม่เคยเห็นมาก่อน

### 2.5. การแบ่งข้อมูลเป็นจำนวน k ส่วน (k-fold Cross-Validation)

การแบ่งข้อมูลเป็น k ส่วน โดยในแต่ละส่วนของข้อมูลจะถูกแบ่งเป็น ชุดข้อมูลสอน และ ชุดข้อมูลตรวจสอบ และเมื่อทดสอบจนครบ k ครั้งแล้วจะมีการนำเอา ค่าความถูกต้อง (Accuracy) ของแต่ละครั้งมาเฉลี่ยเพื่อให้ได้ค่าความถูกต้องสุดท้าย และหากผลลัพธ์มีค่าความถูกต้องสูงก็หมายความว่า

แบบจำลองนั้นมีประสิทธิภาพหรือความแม่นยำสูงนั่นเอง โดยทั่วไปค่า k ที่ใช้ในการทดลองจะใช้เป็น 10 เนื่องจากเป็นการแบ่งข้อมูลเป็น ชุดข้อมูลสอน เป็นร้อยละ 90 และ ชุดข้อมูลตรวจสอบ เป็นร้อยละ 10 และทำการสลับกันสร้างแบบจำลองเพื่อแน่ใจว่าข้อมูลทุกส่วนถูกนำมาสร้าง และทดสอบด้วยความน่าจะเป็นเท่าๆกัน

**2.6. เวกเตอร์ของการนับคำ (Count Vectorizer)**

เวกเตอร์ของการนับคำ [6] คือการนำกลุ่มของ โทเคน (token) มาสร้างเป็น เมทริกซ์ (matrix) โดยใช้กลุ่มของคำที่มีเป็นตัวอ้างอิง คำที่มีในประโยคจะถูกตั้งค่าเป็น 1 คำที่ไม่มีจะเป็น 0 เช่น มีกลุ่มของคำ [“This”, “is”, “am”, “are”, “a”, “be”, “test”, “word”, “sentence”] ประโยค “This is a test sentence” จะแปลงเป็นเมทริกซ์ได้ดังนี้ [1, 1, 0, 0, 1, 0, 1, 0 ,1]

**2.7. การปรับไฮเปอร์พารามิเตอร์ (Tune Hyperparameter)**

การปรับไฮเปอร์พารามิเตอร์อัตโนมัติ [7] เป็นทางออกหนึ่งในการแก้ปัญหาเพื่อให้ได้โมเดลที่มีประสิทธิภาพในระยะเวลาที่สั้นลง ซึ่งกระบวนการนี้จะทำให้การเลือกชุดการปรับไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุดสำหรับแบบจำลองโดยอัตโนมัติ

ในปัจจุบันมีวิธีการปรับไฮเปอร์พารามิเตอร์อัตโนมัติ [7] อยู่มากมาย ทั้งนี้ การค้นหาแบบกริด (grid search) และ การค้นหาแบบสุ่ม (random search) เป็น 2 วิธีดั้งเดิมที่รู้จักกันอย่างกว้างขวาง การค้นหาแบบกริด เริ่มต้นจากผู้สร้างแบบจำลองต้องกำหนดชุดของการปรับไฮเปอร์พารามิเตอร์ที่ประกอบด้วยค่าต่าง ๆ ของพารามิเตอร์ของแบบจำลองที่ต้องการทดสอบ จากนั้นแบบจำลองจะทำการรันทุกชุดการปรับไฮเปอร์พารามิเตอร์จนหมด แล้วจึงคืนค่าที่ดีที่สุดของการปรับไฮเปอร์พารามิเตอร์ แต่ละตัวออกมา ในขณะที่การค้นหาแบบสุ่มนั้น ผู้สร้างแบบจำลองเพียงกำหนดขอบเขตค่าของการปรับไฮเปอร์พารามิเตอร์ที่สนใจ ซึ่งการค้นหาแบบสุ่มจะทำการสุ่มชุดของการปรับไฮเปอร์พารามิเตอร์ที่มีค่าต่างๆ ขึ้นมาทดสอบกับแบบจำลอง ซึ่งการค้นหาแบบสุ่มนี้จะช่วยประหยัดเวลาได้มากกว่า และได้ชุดของการปรับไฮเปอร์พารามิเตอร์ที่มีการกระจายตัวที่มากกว่าการค้นหาแบบกริด

**2.8. เทคนิคการทำเหมืองข้อมูล (Data Mining Techniques)**

**2.8.1. ต้นไม้ตัดสินใจ (Decision Tree)**

ต้นไม้ตัดสินใจ [8] คือ แบบจำลองทางคณิตศาสตร์เพื่อการหาทางเลือกที่ดีที่สุด โดยการนำข้อมูลมาสร้างแบบจำลองการพยากรณ์ในรูปแบบของโครงสร้างต้นไม้ ดังรูปที่ 2.3 ซึ่งมีการเรียนรู้ข้อมูลแบบมีผู้สอน (Supervised Learning) สามารถสร้างแบบจำลองการจัดหมวดหมู่ (Clustering) ได้จากกลุ่มตัวอย่างของชุดข้อมูลสอนได้โดยอัตโนมัติ

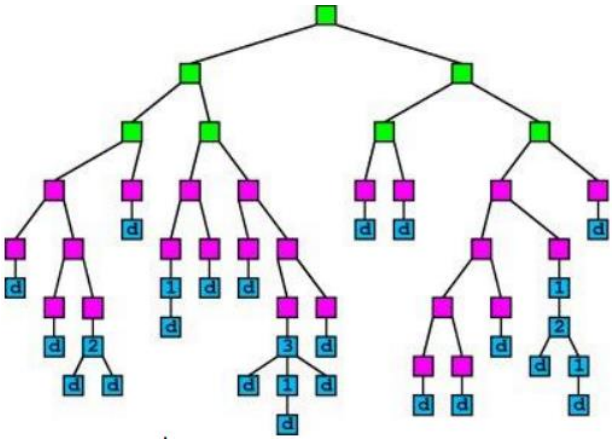
โดยปกติมักประกอบด้วยกฎในรูปแบบ “ถ้า เงื่อนไขแล้วผลลัพธ์” เช่น

“If Income = High and Married = No THEN Risk = Poor”

“If Income = High and Married = Yes THEN Risk = Good”

ส่วนประกอบของต้นไม้ตัดสินใจประกอบด้วย

1) โหนด (Node) คือคุณสมบัติต่างๆ เป็นจุดที่แยกข้อมูลว่าจะให้ไปในทิศทางใดซึ่งโหนดที่อยู่สูงสุดเรียกว่า โหนดราก (Root Node)
2) กิ่ง (Branch) คือ คุณสมบัติของคุณสมบัติในโหนดที่แตกออกมา โดยจำนวนของกิ่งจะเท่ากับคุณสมบัติของโหนด
3) ใบ (Leaf) คือ กลุ่มของผลลัพธ์ในการแยกแยะข้อมูล



**รูปที่ 2. 2 ส่วนประกอบของต้นไม้ตัดสินใจ**

### 2.8.2. การสุ่มป่าไม้ (Random Forest)

การสุ่มป่าไม้ [9] เป็นเทคนิคที่สร้างแบบจำลองที่หลากหลายโดยสุ่มตัวอย่างจาก ชุดข้อมูลสอน และสุ่มแอตทริบิวต์ (Feature) ต่างๆ ออกมาเป็นหลายๆ ชุด จากนั้นนำมาสร้างแบบจำลองด้วยเทคนิคต้นไม้ตัดสินใจหลายๆ ต้น เพียงอย่างเดียว ซึ่งแต่ละต้นก็จะให้คำตอบออกมา ในขั้นตอนท้ายสุดจะนำคำตอบเทคนิคต้นไม้ตัดสินใจแต่ละต้นมารวมกันเพื่อพิจารณาค่าที่เหมาะสมที่สุด แม้ว่าจะเป็นเทคนิคต้นไม้ตัดสินใจเหมือนกันแต่ข้อมูลและคุณลักษณะที่ใช้ในการสร้างแบบจำลองต่างกันก็ทำให้แบบจำลองที่สร้างขึ้นมามีลักษณะที่ต่างกัน

### 2.8.3. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเค (k-Nearest Neighbors : KNN)

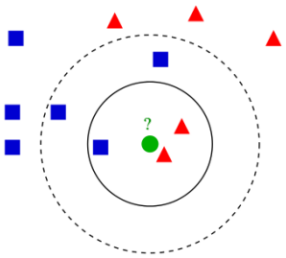
ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเค [10] เป็นวิธีการจำแนกประเภทข้อมูลที่ใช้วิธีการหาระยะห่าง ระหว่างคุณลักษณะของแต่ละข้อมูล ซึ่งวิธีหารนี้จะเหมาะสำหรับข้อมูลที่เป็นแบบตัวเลข โดยขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเค มีขั้นตอนโดยสรุปดังนี้

1) กำหนดจำนวนเพื่อนบ้าน k (นิยมกำหนดให้เป็นเลขคี่)
2) คำนวณระยะห่าง (distance) ของข้อมูลที่ต้องการพิจารณากับชุดข้อมูลสอน โดยสามารถคำนวณได้จากระยะทางยูคริเดียน (Euclidean distance) ดังสมการที่ (1)

$$\mathrm{dist}(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (1)$$

โดยที่ $\mathrm{dist}(p,q)$ หมายถึง ระยะห่างระหว่างข้อมูล $p$ กับ $q$, $p_i$ หมายถึง ค่าของข้อมูล คุณสมบัติที่ $i$ ของข้อมูลที่ $p$ และ $q_i$ หมายถึงค่าของข้อมูลคุณสมบัติที่ $i$ ของข้อมูลที่ $q$

3) จัดลำดับของระยะห่างจากน้อยไปมากและเลือกชุดข้อมูลที่น้อยที่สุดตามจำนวน $k$
4) กำหนดให้คำตอบของข้อมูลที่ต้องการทำนาย คือกลุ่มที่มีจำนวนมากที่สุดในกลุ่มของชุดข้อมูล $k$ ตัวแรก




**รูปที่ 2. 3 การจัดกลุ่มข้อมูลของขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเค**

จากรูปที่ 2.4 กำหนดให้จุดที่พิจารณาคือ วงกลมสีเขียว ควรจัดกลุ่มให้จุดที่สนใจไปอยู่ในคลาสแรกของสี่เหลี่ยมสีน้ำเงิน หรือคลาสสองของสามเหลี่ยมสีแดง ถ้า $k = 3$ แล้ววงกลมสีเขียวจะอยู่ในคลาสสอง เพราะมีสี่เหลี่ยม 1 รูป และ สามเหลี่ยม 2 รูป อยู่ในวงกลมวงใน ถ้า $k = 5$ แล้ววงกลมสีเขียวจะอยู่ในคลาสแรก เพราะมีสี่เหลี่ยม 3 รูป และ สามเหลี่ยม 2 รูป อยู่ในวงกลมวงนอก

### 2.8.4. นาอีฟเบย์ (Naïve Bayes)

นาอีฟเบย์ [10] เป็นวิธีการจำแนกประเภทข้อมูลที่มีประสิทธิภาพรูปแบบหนึ่งที่ใช้หลักความน่าจะเป็นซึ่งอยู่บนพื้นฐานของทฤษฎีเบย์และมีสมมุติฐานจากการเกิดเหตุการณ์ต่างๆ เป็นอิสระต่อกัน โดยการเรียนรู้แบบเบย์เหมาะกับกรณีของกลุ่มตัวอย่างที่มีจำนวนมาก และมีคุณสมบัติหรือแอทริบิวต์ (Attribute) ของตัวอย่างไม่ขึ้นต่อกัน และมีการจำแนกประเภทเบย์โดยมักนำวิธีการเรียนรู้แบบเบย์ไปประยุกต์ใช้งานด้านการจำแนกประเภทข้อความ (Text classification) อีกทั้งขั้นตอนวิธีในการทำงานไม่ซับซ้อนเหมือนการเรียนรู้ในรูปแบบอื่น หากกำหนดให้ความน่าจะเป็นของข้อมูลภายใต้สมมุติฐานที่ข้อมูลในกลุ่ม Vi แต่ละตัวเป็นอิสระต่อ-

กันสำหรับข้อมูล X ที่มีคุณสมบัติ n ตัว โดยที่ $X = \{a_1, a_2, \dots, a_n\}$ หรือเรียกว่า $P(a_1, a_2, a_3, \dots | V_j)$ โดยที่

$$P(a_1, a_2, a_3, \dots | V_j) = \prod_i P(a_i | v_j) \quad (2)$$

จากสมการที่ (2) คำตอบของ $P(a_1, a_2, a_3, \dots | V_j)$ หมายถึงคลาสของผลลัพธ์ $V_i$ ใดๆ โดยมักเป็นคลาสที่มีค่าความน่าจะเป็นที่มากที่สุดที่ได้จากการคำนวณจากสมการที่ (2) และใช้เป็นคำตอบสำหรับการจำแนกประเภทของข้อมูลคำนวณความน่าจะเป็นของคำตอบ $P(V_i)$ ที่พบในแต่ละคลาสจากการนำค่า $P(a_1, a_2, a_3, \dots | V_j)$ ในสมการที่ (2) มาคูณความน่าจะเป็นของคลาสนั้นๆ เพื่อหาค่า $V_{NB}$ จากสมการที่ (3)

$$V_{NB} = \arg_{vieV} \max \times \prod_{i=1}^{n} P(a_i | v_i) \quad (3)$$

การเรียนรู้แบบนาอีฟเบย์เป็นการเรียนรู้ที่ต่อเนื่องในแต่ละช่วงเวลาโดยจะมีการเรียนรู้ที่เปลี่ยนแปลงไปเนื่องจากตัวแบบของข้อมูลจะถูกปรับเปลี่ยนค่าไปตามค่าของตัวอย่างใหม่ที่มีการเพิ่มเข้ามาในแต่ละช่วงเวลาโดยรวมเข้ากับความรู้เดิมที่วิธีการทำนายค่ากลุ่มหรือคลาส โดยมีขั้นตอนวิธีหรืออัลกอริทึมในการทำงานที่สามารถปรับใช้ได้กับข้อมูลในหลายรูปแบบทั้งแบบชนิดตัวเลข และข้อความ

ขั้นตอนวิธีของนาอีฟเบย์

1) คำนวณความน่าจะเป็นของคำตอบที่พบในแต่ละคลาส จากการนำค่า $P(a_1, a_2, a_3, \dots | V_j)$ มาคูณความน่าจะเป็นของกลุ่มนั้นๆ $P(V_i)$ เพื่อหาค่า $V_{NB}$
2) นำค่าความน่าจะเป็นที่ได้มาเปรียบเทียบกัน คลาสใดที่มีค่าความน่าจะเป็นสูงสุดถือเป็นคำตอบหรือค่ากลุ่มของข้อมูล

### 2.8.5. ต้นไม้ที่ไล่ระดับสี (Gradient Boosted Trees : GBDT)

การจำแนกต้นไม้ตัดสินใจแบบค่อยเป็นค่อยไปที่มีการไล่ระดับสี (GBDT) [11] และอัลกอริทึมการถดถอยเป็นการประมวลผลทั้งหมดของต้นไม้การถดถอย (การตัดสินใจ) ที่สร้างขึ้นโดยใช้เทคนิคการไล่ระดับสีกำหนด n คุณสมบัติเวคเตอร์ $X = \{X_1 = (X_{11}, \dots, X_{1p}), \dots, X_n, \dots, X_{np})\}$ ของเวกเตอร์คุณสมบัติมิติ np และการตอบสนอง $nY = \{Y_1, \dots, y_n\}$ กระบวนการเรียนรู้ของอัลกอริทึมคือการสร้างการจำแนกต้นไม้หรือการถดถอยแบบค่อยเป็นค่อยไปโดยใช้การไล่ระดับสี โดยอาศัยข้อมูลคุณลักษณะ และการตอบสนอง จากนั้นใช้โมเดลการจำแนก และการถดถอยเพื่อจำแนกหรือทำนายตัวอย่างใหม่ที่เข้ามา

ขั้นตอนการฝึกอบรมเป็นอัลกอริทึมการไล่ระดับสีการทำงานแบบวนซ้ำ ซึ่งลดฟังก์ชันวัตถุประสงค์โดยการเลือกฟังก์ชันต้นไม้ถดถอยที่ใช้ไปในทิศทางเชิงลบ ดังสมาการที่ (4)

$$L(f) = \sum_{i=1}^{n} l(y_i, f(x_i)) + \sum_{k=0}^{m} \Omega(f_k) \quad (4)$$

โดยที่ $L(f)$ เป็น convex loss function ที่ค่าแตกต่างได้สองเท่า และ $\Omega(f) = \gamma^T + \frac{\lambda}{2}\|w\|$ เป็นเงื่อนไขการทำให้เป็นมาตรฐานที่ลงโทษความซับซ้อนของแบบจำลองที่กำหนดโดยจำนวนใบ T และบรรทัดฐาน $L_2$ ของน้ำหนัก $\|w\|$ สำหรับต้นไม้แต่ละต้น $\gamma$ และ $\lambda$ เป็นพารามิเตอร์การทำให้เป็นมาตรฐาน

**2.8.6. สถาปัตยกรรมเพอร์เซปตรอนแบบหลายชั้น (Multi-Layer Perceptron : MLP)**

สถาปัตยกรรมเพอร์เซปตรอนแบบหลายชั้นจะประกอบไปด้วยชั้นอินพุต (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นเอาต์พุต (Output Layer) แสดงดังรูปที่ 2.4

ข้อมูลที่เข้ามาจะถูกส่งไปคำนวณในชั้นซ่อนเพื่อหาผลรวมของผลคูณของข้อมูลเข้าและค่าน้ำหนักแสดงดังสมการที่ (5)

$$y = \sum_{i=0}^{n} x_i, w_i \quad (5)$$

y คือค่าผลรวมของผลคูณข้อมูลเข้า $(x_i)$ และน้ำหนัก $(w_i)$

i *คือจำนวนข้อมูลเข้าหรือจำนวนค่าน้ำหนัก*

นำผลลัพธ์ที่ได้ไปคำนวณในฟังก์ชันการแปลงถ่ายทอดข้อมูล (Sigmoid Function) ดังสมการที่ (6)

$$o = g(y)\frac{1}{1+e^{-y}} \quad (6)$$

o คือค่าผลลัพธ์ของชั้นซ่อน

y คือค่าผลรวมของผลคูณข้อมูลเข้า $(x_i)$ และน้ำหนัก $(w_i)$

จากผลลัพธ์ของชั้นซ่อนก็จะถูกส่งไปยังชั้นเอาต์พุต ซึ่งในส่วนของชั้นเอาต์พุตจะมีการเปรียบเทียบกับค่าผลลัพธ์ที่ประมวลผลได้ และผลลัพธ์เป้าหมาย ซึ่งถ้าได้ผลลัพธ์ที่ยอมรับได้ก็จะหยุดการปรับค่าน้ำหนัก แต่ถ้ายังไม่อยู่ในช่วงของผลลัพธ์ที่ยอมรับได้ก็จะเข้าสู่กระบวนการเรียนรู้แบบแพร่ย้อนกลับ ซึ่งจะเป็นกระบวนการปรับค่าน้ำหนักจนกว่าจะได้ค่าที่เหมาะสม โดยสามารถคำนวณได้จากสมการที่ (7) และ (8)

$$\delta_k = O_k(1 - O_k)(T_k - O_k) \quad (7)$$

$$\delta_l = O_l(1 - O_l)\left(\sum_{k \in O} w_{k_l}\delta_k\right) \quad (8)$$

$\delta_k$ คือค่าความผิดพลาดที่คำนวณจากชั้นผลลัพธ์

$\delta_l$ คือค่าความผิดพลาดที่คำนวณจากชั้นซ่อน

$T_k$ คือค่าผลลัพธ์เป้าหมาย

$O_k$ คือค่าผลลัพธ์ที่ประมวลผลได้จากชั้นผลลัพธ์

$O_l$ คือค่าผลลัพธ์ที่ประมวลผลได้จากชั้นซ่อน

w คือค่าน้ำหนัก

k และ l คือดัชนีของโหนดชั้นผลลัพธ์และชั้นซ่อน

ในกรณีค่าผลลัพธ์ที่ได้จากการประมวลผลและผลลัพธ์เป้าหมายยังมีความแตกต่างกันสูงจะกระทำการปรับค่าน้ำหนักเพื่อหาค่าน้ำหนักที่เหมาะสมกับงาน การปรับค่าน้ำหนักแสดงดังสมการที่ (9) และ (10)

$$w_i^{new} = w_i^{old} + \Delta w_i \tag{9}$$

$$\Delta w_i = \delta_i x_i \tag{10}$$

$w_i^{new}$ คือค่าน้ำหนักใหม่ที่ได้จากการคำนวณ

$w_i^{old}$ คือค่าน้ำหนักเก่า

$\Delta w_i$ คืออัตราการเปลี่ยนแปลง

$\alpha$ คืออัตราการเรียนรู้ (Learning Rate)

$\delta_i$ คือค่าความผิดพลาดของผลลัพธ์

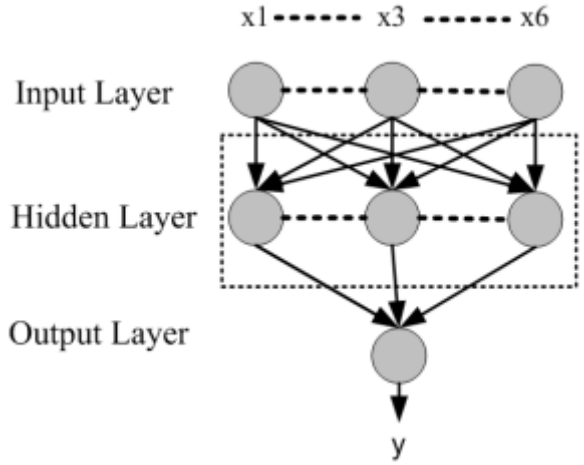
$x_i$ คือค่าข้อมูลชั้นนำเข้า

i คือจำนวนข้อมูลเข้าหรือจำนวนค่าน้ำหนัก

การทำงานของสถาปัตยกรรมเพอร์เซปตรอนแบบหลายชั้น และใช้การเรียนรู้แบบแพร่ย้อนกลับ จะกระทำฝึกสอน (Train) โดยการปรับค่าน้ำหนักไปจนกระทั่งได้ค่าความผิดพลาดที่น้อยที่สุดหรือได้ค่าความผิดพลาดที่ยอมรับได้ เมื่อได้ค่าน้ำหนักที่เหมาะสมแล้วก็จะนำไปใช้ทดสอบ (Test) และนำผลลัพธ์ที่ได้ทำการทดสอบไปคำนวณด้วยฟังก์ชัน การแปลงถ่ายทอด-ข้อมูล (Threshold Function) เพื่อให้ได้ คำตอบที่เป็นจริงหรือเท็จ ใช่หรือไม่ใช่ หรือ “0” หรือ “1” ดังสมการที่ (11)

$$f(x) = \begin{cases} 1, & x > T \\ 0, & x < T \\ \text{Random}, & x = T \end{cases} \tag{11}$$

ค่าผลลัพธ์ที่ได้จากการประมวลผลจะเป็น 1 ในกรณีที่ได้ค่าผลลัพธ์ x มีค่ามากกว่าค่าเกณฑ์ (Threshold : T) เปรียบเสมือนการตัดสินใจว่าใช่ ค่าผลลัพธ์จะเป็น 0 ในกรณีที่ได้ค่าผลลัพธ์ x มีค่าน้อยกว่าเกณฑ์เปรียบเสมือนการตัดสินใจว่าไม่ใช่ และถ้าค่าผลลัพธ์ที่ได้เท่ากับค่าเกณฑ์ให้ทำการสุ่มการตัดสินใจว่าจะเป็น “0” หรือ “1”




**รูปที่ 2. 4 สถาปัตยกรรมเพอร์เซปตรอนแบบหลายชั้น**

### 2.8.7. การวิเคราะห์การถดถอยโลจิสติก (Logistic Regression)

โมเดลการวิเคราะห์การถดถอยโลจิสติก [13] คือสมการที่ (12)

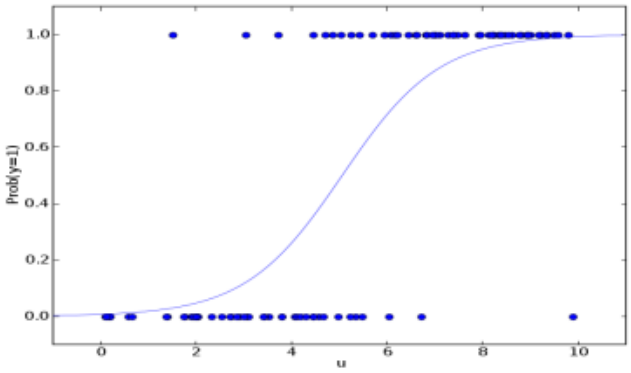
$$y_i = x_i\beta + 0\varepsilon_i \tag{12}$$

เมื่อ $x'_i = [1, x_{i1}, x_{i2}, \dots, x_{ik}]$

$\beta' = [\beta_0, \beta_1, \dots, \beta_k]$

$\varepsilon_i$ คือความคลาดเคลื่อน

สำหรับการวิเคราะห์การถดถอยโลจิสติกที่เป็นแบบไบนารี ตัวแปรตาม y มีค่าคือ 0 และ 1 ความสัมพันธ์ระหว่างตัวแปรตาม และตัวแปรทำนายจึงไม่อยู่ในรูปเชิงเส้น ซึ่งความสัมพันธ์ของตัวแปรตามและตัวแปรทำนายในการวิเคราะห์การถดถอยโลจิสติกจะอยู่ในรูปคล้ายตัว S ดังรูปที่ 2.5




**รูปที่ 2. 5 ความสัมพันธ์ตัวแปรตามและตัวแปรทำนายในการวิเคราะห์การถดถอยโลจิสติก**

ดังนั้นจะได้สมการที่ (13)

$$P(y) = \frac{e^{b_0+b_1x}}{1+e^{b_0+b_1x}} \tag{13}$$

เมื่อ $P(y)$ คือความน่าจะเป็นของการเกิดเหตุการณ์ที่สนใจ

$Q(y)$ คือความน่าจะเป็นของการไม่เกิดเหตุการณ์ที่สนใจโดยที่

$$Q(y) = 1 - P(y)$$

โดยที่ $P(y) \geq 0.5$ สรุปว่าเกิดเหตุการณ์ที่น่าสนใจ

$P(y) < 0.5$ สรุปว่าไม่เกิดเหตุการณ์ที่สนใจ

จากความสัมพันธ์ระหว่างตัวแปรตามกับตัวแปรทำนายที่ไม่เป็นเชิงเส้น จึงทำการปรับตัวอยู่ในรูปเชิงเส้นโดยเขียนให้อยู่ในรูปของ odds หรือ odd ratio ซึ่งหมายถึงอัตราส่วนระหว่างความน่าจะเป็นของการเกิดเหตุการณ์กับความน่าจะเป็นของการไม่เกิดเหตุการณ์ดังสมการที่ (14) และสมการที่ (15)

$$\log\left(\frac{P(y)}{Q(y)}\right) = \log\left(\frac{p(y)}{1-p(y)}\right) = b_0 + b_1b_2x_1 + \cdots + b_px_p \tag{14}$$

ดังนั้น

$$\log(\text{odds}) \text{ หรือ } \log it = b_0 + b_1x_1 + \cdots + b_px_p \tag{15}$$

ในการประมาณค่าสัมประสิทธิ์การถดถอย $b_i$ ของของสมการ Logistic Regression จะใช้วิธีภาวะน่าจะเป็นสูงสุด (Maximum likelihood) โดยใช้วิธีการคำนวณซ้ำๆ (Iteration) โดยเริ่มต้นจากการประมาณค่าสัมประสิทธิ์ในสมการการวิเคราะห์การถดถอยโลจิสติกเพื่อให้สามารถแก้สมการได้แล้วพิจารณาผลการทำนายเพื่อนำมาประมาณค่าสัมประสิทธิ์ใหม่ที่จะทำให้เกิดความน่าจะเป็นสูงสุดเพื่อที่จะสามารถทำนายค่า ของตัวแปรตามได้ถูกต้องใกล้เคียงกับข้อมูลจริงมากที่สุด

### 2.8.8. ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machines : SVM)

ซัพพอร์ตเวกเตอร์แมชชีน [14] เป็นอัลกอริทึมในการคัดแยกกลุ่มเพื่อจัดประเภท หรือจำแนกประเภทข้อมูลที่มีการนำมาใช้ในด้านการประมวลผลภาพ เป็นวิธีการจำแนกกลุ่มข้อมูลที่อาศัยระนาบการตัดสินใจที่เรียกว่า ระนาบเกิน หรือไฮเปอร์เพลน (Hyperplane) มาใช้ในการจำแนกกลุ่มข้อมูล โดยใช้สมการเส้นตรงในการแบ่งข้อมูลออกเป็น 2 กลุ่มแยกออกจากกัน ซัพพอร์ตเวกเตอร์แมชชีนมีรูปแบบในการเรียนรู้เป็นกระบวนการเลือกแบบจำลองที่เหมาะสมที่สุด จะทำให้ได้คำที่เหมาะสมที่สุดเป็นคำตอบ ดังนั้นซัพพอร์ตเวกเตอร์แมชชีนจึงเป็นที่นิยม และเริ่มนำไปใช้ในงานด้านการรู้จำรูปแบบซึ่งจะเลือกใช้ซัพพอร์ตเวกเตอร์แมชชีนแบบแบ่งกลุ่ม

ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการแบ่งกลุ่มข้อมูลนั้นจะใช้ ระนาบเกินที่เหมาะสมที่สุด (Optimal Hyperplane) ในการแบ่งกลุ่ม ในการสร้างระนาบเกินที่ใช้ในการแบ่งกลุ่มข้อมูลสามารถสร้างได้หลายแบบ แต่จะมีระนาบเกินที่เหมาะสมที่สุดเพียงระนาบเดียวเท่านั้นที่สามารถรักษาระยะห่างมากที่สุดระหว่างข้อมูล 2 กลุ่มที่ใกล้กันมากที่สุดได้

กำนหนดให้ $(x_i,\ y_i), \dots, (x_n,\ y_n)$ เมื่อ $x \in R^m, y \in \{-1, 1\}$ ตัวอย่างที่ใช้การสอน โดย

n คือจำนวนข้อมูลตัวอย่าง

m คือจำนวนมิติของข้อมูลเข้า

x คือข้อมูลนำเข้า

y คือประเภทหรือกลุ่มของข้อมูล ซึ่งประกอบด้วย *2* กลุ่ม มีค่า +1 หรือ -1

สำหรับปัญหาเชิงเส้น ข้อมูลมิติขนาดสูงได้ถูกแบ่งเป็น *2* กลุ่ม โดยใช้ระนาบตัดสินใจพิจารณาชุดของกลุ่มข้อมูล x โดยที่กำหนดให้กลุ่มข้อมูล $x_1$ เป็นข้อมูล $x_i$ ที่มีค่าเป็นบวก และ $x_2$ เป็นข้อมูล $x_i$ ที่มีค่าเป็นลบ การสร้างระนาบตัดสินใจเพื่อแบ่งแยกกลุ่มข้อมูลสามารถคำนวณได้ดังสมการที่ (16)

$$(w * x_1) + b > 0 \text{ ถ้า } y_i = 1 \text{ และ } (w * x_2) + b < 0 \text{ ถ้า } y_i = -1 \qquad (16)$$

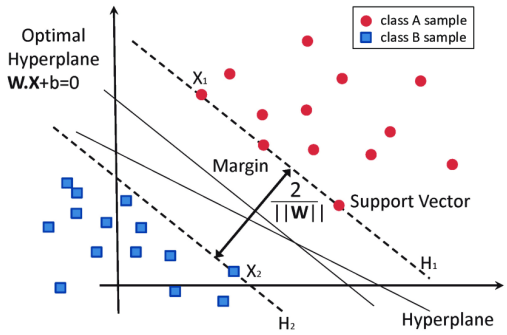
$w$ คือเวกเตอร์น้ำหนัก

$x_1$ คือเวกเตอร์ข้อมูลที่มีค่าเป็นบวก

$x_2$ คือเวกเตอร์ข้อมูลที่มีค่าเป็นลบ

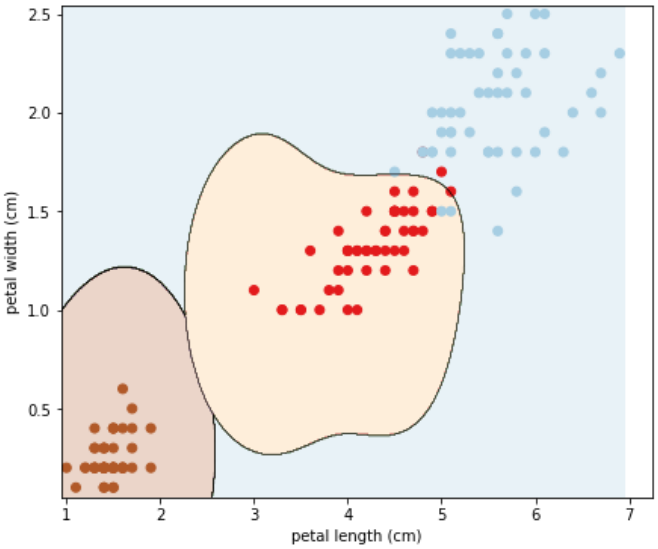
$b$ คือค่าอคติ (bias)

ในการหาระนาบเกินที่เหมาะสมที่สุด จะทำการหาตำแหน่งของ ซัพพอร์ตเวกเตอร์ (Support Vector) เพื่อใช้เป็นตัวแทนของกลุ่มข้อมูลทั้งชุด ในการพิจารณาเกณฑ์แบ่งกลุ่มโดยอาศัยหลักการคือจะใช้ระนาบเกินที่เป็นระยะห่างที่สุดระหว่างข้อมูล *2* กลุ่ม ที่อยู่ใกล้กันมากที่สุดเพียงระนาบเดียวเท่านั้น ในทางทฤษฎีจะต้องไม่มีข้อมูลเกินเข้ามาในระหว่างขอบระนาบทั้งสอง จากนั้นจึงหาระนาบที่รักษาระยะห่างจากขอบมากที่สุด (Maximum Margin) และถือว่าระนาบดังกล่าวคือ ระนาบสำหรับการแบ่งกลุ่มที่เหมาะที่สุดแสดงดังรูปที่ 2.6




**รูปที่ 2. 6 ตัวอย่างระนาบการตัดสินใจแบ่งกลุ่มข้อมูลของซัพพอร์ตเวกเตอร์แมชชีน**

จากที่กล่าวข้างต้นเป็นการแบ่งกลุ่มข้อมูลด้วยระนาบการตัดสินใจแบบเชิงเส้นเท่านั้น โดยซัพพอร์ตเวกเตอร์แมชชีนมีเคอร์เนลฟังก์ชัน (Kernel Function) แบบอื่นให้ผู้ใช้สามารถประยุกต์ใช้ในการแก้ปัญหาได้หลายวิธี ดังนั้นเพื่อให้อัลกอริทึมดังกล่าวสามารถแบ่งแยกกลุ่มข้อมูลที่มีลักษณะไม่เป็นเชิงเส้น (Nonlinear Dataset) จะต้องแปลงกลุ่มข้อมูลตัวอย่างไปสู่มิติที่สูงขึ้น (Higher Dimensional Space) ซึ่งถูกเรียกว่า ฟีเจอร์สเปซ (Feature Space) โดยการแปลงดังกล่าวจะกระทำผ่านฟังก์ชันที่ไม่เป็นเชิงเส้นและสร้างฟังก์ชันวัดระยะห่างที่เรียกว่าเคอร์เนลฟังก์ชันบนฟีเจอร์สเปซ ซึ่งเหมาะสมสำหรับข้อมูลที่มีมิติข้อมูลสูงโดยมีวัตถุประสงค์ที่จะพยายามจะทำการลดความผิดพลาดในการทำนายกลุ่มข้อมูล (Minimize Error) พร้อมกับเพิ่มระยะแยกแยะโดยพยายามสร้างเส้นแบ่งกลางระหว่างกลุ่มให้มีระยะห่างระหว่างขอบเขตทั้งสองกลุ่มมากที่สุด (Maximized Margin) ใช้สำหรับข้อมูลที่มีลักษณะมิติของข้อมูลที่สูงมาก โดยโครงงานนี้ได้ใช้ เคอร์เนลฟังก์ชันคือ Gaussian RBF มีลักษณะดังรูปที่ 2.7



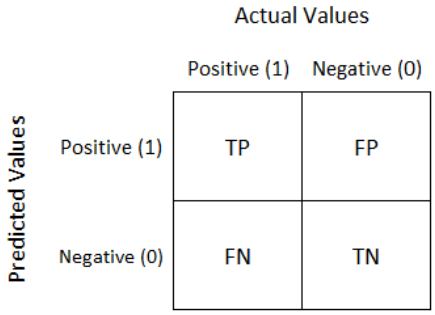

**รูปที่ 2. 7 การแบ่งกลุ่มโดยใช้เคอร์เนล Gaussian RBF**

### 2.9. ซอฟต์โหวต (Soft Vote)

ซอฟต์โหวต [15] คือ การให้โมเดลแต่ละตัวหาค่าความน่าจะเป็นของแต่ละคลาส และนำมาหาค่าเฉลี่ยแล้วสรุปว่าผลสุดท้ายควรจะเป็นคลาสไหน เช่น โมเดล KNN ทำนายว่าข้อมูลควรเป็น คลาส A = 0.8, B = 0.2 โมเดล MLP ทำนายว่าข้อมูลควรเป็น คลาส A = 0.4, B = 0.7 ดังนั้นข้อมูลในครั้งนี้ควรจะเป็น คลาส A เพราะ คลาส A = (0.8+0.4)/2 = 0.6, คลาส B = (0.2+0.7)/2 = 0.45

### 2.10. การประเมินผลโมเดล (Evaluation Model)

คอนฟิวชันแมทริกซ์ (Confusion Matrix) [16] คือตารางสำคัญในการวัดความสามารถของการเรียนรู้ของเครื่องในการแก้ปัญหาการจำแนกประเภท ดังรูปที่ 2.8




**รูปที่ 2. 8 ตัวอย่างตารางคอนฟิวชันแมทริกซ์ ขนาด 2x2**

True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” และ มีค่าเป็น “ จริง ”

True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” และ มีค่า “ ไม่จริง ”

False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” แต่ มีค่าเป็น “ไม่จริง”

False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” แต่ มีค่าเป็น “จริง”

โดยทั่วไปแล้วจะมีตัววัดที่นิยมใช้กันในงานวิจัยและการทำงานต่างๆ อยู่ 3 ค่า และสมการ คือ

Precision เป็นการวัดความแม่นยำของข้อมูล โดยพิจารณาแยกทีละคลาส โดยหาค่าได้จากสมการที่ (17)

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{17}$$

Recall เป็นการวัดความถูกต้องของโมเดล โดยพิจารณาแยกทีละคลาส โดยหาค่าได้จากสมการที่ (18)

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{18}$$

Accuracy เป็นการวัดความถูกต้องของโมเดล โดยพิจารณารวมทุกคลาส โดยหาค่าได้จากสมการที่ (19)

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{19}$$

F1-Score คือค่าเฉลี่ยแบบ ค่าเฉลี่ยฮาร์โมนิก (harmonic mean) ระหว่าง precision และ recall สร้างขึ้นมาเพื่อเป็นแมทริกซ์เดียวที่วัดความสามารถของโมเดล โดยหาค่าได้จากสมการที่ (20)

$$\text{F1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \tag{20}$$

# บทที่ 3
# ปัญหา และสมมติฐาน

### 3.1. ปัญหา

ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่ในปัจจุบันนี้เปิดให้ผู้ใช้บริการสามารถแสดงความคิดเห็นผ่านเว็บไซต์ของศูนย์ศรีพัฒน์ฯ ได้เพื่อที่จะนำความคิดเห็นนี้ไปหาข้อสรุป และดำเนินการแก้ไขปัญหาให้กับลูกค้า แต่ในการดำเนินการนั้นเป็นไปอย่างล่าช้า เพราะว่าเมื่อความคิดเห็นเข้าสู่ระบบจะถูกนำไปพักไว้ในฐานข้อมูลก่อน หลังจากนั้นต้องรอเจ้าหน้าที่เข้ามาอ่าน และวิเคราะห์เพื่อแก้ไขปัญหาต่อไป จะเห็นได้ว่ากว่าจะได้เข้าใจปัญหาจะต้องผ่านการรอคอย และขั้นตอนต่างๆ ซึ่งขัดต่อความต้องการความรวดเร็วในโลกยุคปัจจุบัน และความล่าช้านี้ทำให้ผู้ใช้บริการบางส่วนเกิดความไม่พอใจ และไม่ทำการกลับมาใช้บริการอีกทำให้ศูนย์ศรีพัฒน์ฯ เสียรายได้ ฐานลูกค้า และความไว้วางใจ

### 3.2. สมมติฐาน

จากปัญหาจะเห็นได้ว่าจำเป็นจะต้องทำการปรับเปลี่ยนขั้นตอนในเรื่องของการใช้แรงงานมนุษย์โดยให้ระบบอัตโนมัติให้เข้ามาแทนที่ซึ่งมีสมมติฐานดังนี้ เมื่อให้ระบบอัตโนมัติที่มีความรวดเร็วในการประมวนผลเข้ามาแทนที่ทำให้ความคิดเห็นถูกประมวลผลและทำการวิเคราะห์ได้ทันทีโดยไม่จำเป็นต้องนำความคิดเห็นไปเก็บพักไว้เฉยๆ ในฐานข้อมูลเพื่อรอพนักงานมาทำการอ่าน และวิเคราะห์ ดังนั้นหากมีความคิดเห็นใดต้องได้รับการแก้ไขปัญหาระบบอัตโนมัติก็จะสามารถแจ้งให้พนักงานที่รับผิดชอบได้ทันทีเพื่อแก้ไขปัญหาให้ได้ทันเวลาสามารถทำให้ผู้ใช้บริการลดความไม่พอใจลงได้

เมื่อเล็งเห็นว่าระบบอัตโนมัติสามารถใช้แก้ปัญหาได้ดังนั้นจึงจำเป็นต้องมีหลักการ และพัฒนาระบบที่มีความสามารถในการจำแนกข้อความเพื่อทำนายหมวดหมู่ที่สนใจ จากการศึกษาความสามารถนี้สามารถสร้างได้จากหลักการของการเรียนรู้ของเครื่องโดยใช้เทคนิคการจำแนกประเภท

# บทที่ 4
# ขั้นตอนวิธี

## 4.1. การรวบรวมข้อมูล

ข้อมูลที่ใช้ในการทำการทดลองครั้งนี้นำมาจากฐานข้อมูลความคิดเห็นของผู้ใช้บริการศูนย์-ศรีพัฒน์ฯ ซึ่งเป็นความคิดเห็นในรูปแบบภาษาไทย ตั้งแต่วันที่ 01 พฤษภาคม พ.ศ. 2555 ถึงวันที่ 30 มิถุนายน พ.ศ. 2564 รวบรวมได้จำนวนทั้งหมด 2,442 ระเบียน (Record) โดยนำข้อมูลออกมาเป็นไฟล์นามสกุล csv

## 4.2. การตรวจสอบ และเตรียมข้อมูล

### 4.2.1. การจัดการกับข้อมูล

จากการตรวจสอบพบว่าข้อมูลนั้นมีความไม่ถูกต้อง และความไม่สมบูรณ์ โดยมีวิธีการจัดการกับข้อมูล ดังตารางที่ 4.1. จากนั้นทำการกำหนดคำตอบแต่ละคลาสให้ข้อมูลซึ่งประกอบไปด้วย คลาส ดี (1) และ ไม่ดี (-1) โดย คลาส ไม่ดี สามารถแบ่งย่อยออกเป็น คลาส รุนแรง (1) และ ไม่รุนแรง (0)

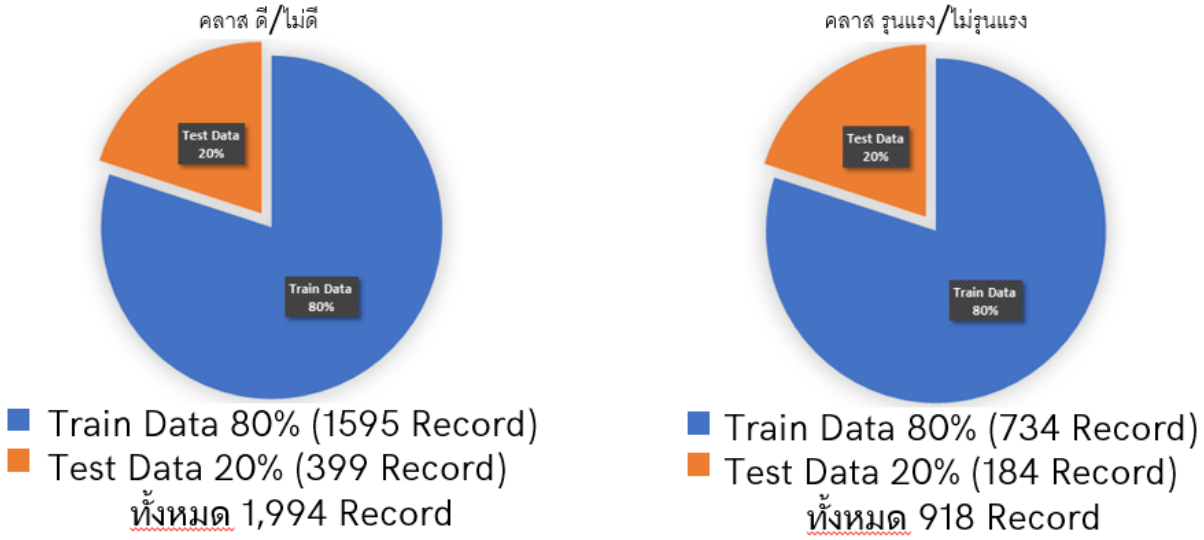
**ตารางที่ 4. 1 การจัดการข้อมูลที่ไม่สมบูรณ์**

| ลักษณะความไม่ถูกต้อง | การจัดการ |
|---|---|
| มี Row ที่เว้นว่างไว้ | ลบระเบียน |
| มีตัวเลขผสม เช่น เลขห้อง วันที่ เบอร์โทรศัพท์ และค่าใช้จ่าย | ลบตัวเลข |
| มีอักขระพิเศษผสม | ลบอักขระพิเศษ |
| มีอักขระขึ้นบรรทัดใหม่ (new line) ผสมในความคิดเห็น | ลบอักขระขึ้นบรรทัดใหม่ |

เพื่อให้สามารถนำความคิดเห็นไปคำนวณได้ดังนั้นจึงต้องแปลงข้อมูลจาก String เป็น Int64 โดยใช้วิธีการเวกเตอร์ของการนับคำจากโมดูล (Module) CountVectorizer ซึ่งเป็นไลบรารี (Library) ของ Scikit-Learn โดยจะได้ว่าชุดข้อมูลซึ่งประกอบไปด้วย คลาส ดี (1) และไม่ดี (-1) มีทั้งหมด 1,994 ระเบียน ได้จำนวนคำที่ไม่ซ้ำกันทั้งหมด 7,096 คำ และเฉลี่ยแล้วแต่-ละระเบียนมีคำประมาณ 10 คำ ชุดข้อมูลซึ่งประกอบไปด้วย คลาส รุนแรง (1) และ ไม่รุนแรง (0) มีทั้งหมด 998 ระเบียน ได้จำนวนคำที่ไม่ซ้ำกันทั้งหมด 5,127 คำ และเฉลี่ยแล้วแต่ละระเบียนมีคำประมาณ 12 คำ

### 4.2.1. การแบ่งข้อมูล

ข้อมูลที่ได้แบ่งออกเป็น 2 ส่วน ซึ่งใช้สำหรับสร้าง 2 โมเดล ดังรูปที่ 4.1 โดยโมเดลแรกใช้ข้อมูลที่ประกอบไปด้วยความคิดเห็นทั้งหมด 1,994 ระเบียน แบ่งออกเป็นคลาส ดี 997 ระเบียน และ คลาส ไม่ดี 997 ระเบียน จากนั้นทำการสุ่มอย่างเป็นระบบโดยเลือกข้อมูลทุกๆ 42 ระเบียน เพื่อให้การทดลองแต่ละครั้งแบ่งข้อมูลได้เหมือนเดิมเสมอ แล้วแบ่งเป็นข้อมูลสอนจำนวน 1,595 ระเบียน คิดเป็น 80% และข้อมูลทดสอบจำนวน 399 ระเบียน คิดเป็น 20%, โมเดลที่สองใช้ข้อมูลที่ประกอบไปด้วยความคิดเห็นคลาส ไม่ดี จำนวน 918 ระเบียน แบ่งออกเป็นคลาส รุนแรง 459 ระเบียน และคลาส ไม่รุนแรง 459 ระเบียน จากนั้นทำการสุ่มอย่างเป็นระบบโดยเลือกข้อมูลทุกๆ 42 ระเบียน แล้วแบ่งเป็นข้อมูลสอนจำนวน 734 ระเบียน คิดเป็น 80% และข้อมูลทดสอบจำนวน 184 ระเบียน คิดเป็น 20%




**รูปที่ 4. 1 การแบ่งข้อมูลในการทดลอง**

## 4.3. การเลือกใช้โมเดล

โมเดลที่ใช้ทดลองประกอบไปด้วยเทคนิค Decision Tree, Random Forest, Gradient Boosted Trees (GBDT), k-Nearest Neighbors (KNN), Naïve Bayes, Multi-Layer Perceptron (MLP), Support Vector Machines (SVM), Logistic Regression โดยโมเดลโมเดลที่ใช้นำมาจากไลบรารีของ Scikit-Learn ซึ่งความสามารถของฮาร์ดแวร์ในการวิจัยครั้งนี้ใช้ประมวนผลการฝึกสอนแต่ละโมเดลได้ในระยะเวลาที่ไม่นานมากจนเกินไปทำให้สามารถมีเวลาทดลองปรับพารามิเตอร์ได้มากขึ้น

### 4.3.1. การปรับไฮเปอร์พารามิเตอร์

ในขั้นตอนนี้ใช้วิธีการปรับไฮเปอร์พารามิเตอร์ ได้แก่ การค้นหาแบบสุ่ม (สุ่มพารามิเตอร์ 100 ครั้ง, แบ่งข้อมูลเป็นจำนวน 10 ส่วน โดยสุ่มอย่างเป็นระบบโดยเลือกข้อมูล

ทุกๆ 42 ระเบียน) และ การค้นหาแบบกริด (แบ่งข้อมูลเป็นจำนวน 10 ส่วน โดยสุ่มอย่างเป็น-ระบบโดยเลือกข้อมูลทุกๆ 42 ระเบียน) ควบคู่กันไปเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เนื่องจากหากใช้การค้นหาแบบกริดเพียงวิธีการเดียวจะทำให้กำหนดเซตของค่าแต่ละพารามิเตอร์ได้ไม่กว้าง เพราะว่าการค้นหาแบบกริดนั้นใช้เวลาในการหาที่นานมาก ดังนั้นจึงเพิ่มการค้นหาแบบสุ่มเพื่อกำหนดเซตของพารามิเตอร์ให้การค้นหาแบบสุ่มนั้นมีเซตของค่าแต่ละพารามิเตอร์มีขนาดกว้างกว่าการค้นหาแบบกริดทำให้เพิ่มโอกาสหาพารามิเตอร์ที่ดีที่สุดโดยใช้เวลาในการหาน้อยกว่าการค้นหาแบบกริด

4.3.1.1. การค้นหาแบบสุ่ม

การกำหนดค่าให้ไฮเปอร์พารามิเตอร์แต่ละโมเดลเป็นไปดังตารางที่ 4.2.

**ตารางที่ 4. 2 การกำหนดค่าไฮเปอร์พารามิเตอร์ใช้กับวิธีการค้นหาแบบสุ่ม**

| Model | Hyperparameter | Value |
|---|---|---|
| Decision Tree | criterion | [gini, entropy] |
| | splitter | [best, random] |
| | max_depth | Range(3-51) |
| | min_samples_split | Range(2-50) |
| | min_samples_leaf | Range(1-50) |
| | max_features | [auto, log2, None] |
| | random_state | 0 |
| Random Forest | n_estimators | np.linspace(200, 1000, 10) |
| | max_features | [auto, sqrt] |
| | max_depth | numpy.linspace(10, 110, 11) |
| | min_samples_split | [2, 5, 10] |
| | min_samples_leaf | [1, 2, 4] |
| | bootstrap | [True, False] |
| GBDT | n_estimators | [5,50,250] |
| | max_depth | [1,3,5,7,9,11] |
| | learning_rate | [0.01,0.1,1,10,100] |
| KNN | n_neighbors | Range(2-3,000) |
| MLP | hidden_layer_sizes | [(50,50,50), (50,100,50), (100)] |
| | activation | [tanh, relu] |
| | solver | [sgd, adam] |

| Model | Hyperparameter | Value |
|---|---|---|
| MLP | alpha | [0.0001, 0.05] |
| | learning_rate | [constant, adaptive] |
| | max_iter | [100, 250, 350, 500] |
| SVM | C | numpy.linspace(0.1, 2.0, 10) |
| | kernel | [linear, poly, rbf, sigmoid] |
| | degree | Range(2-6) |
| | gamma | [auto, scale] |
| | tol | numpy.logspace(np.log(1e-5), np.log(1e-2), num = 10, base = 3) |
| | max_iter | Range((-1)-101) |
| | Probability | True |
| Linear Regression | C | numpy.logspace(np.log(1e-5), np.log(1e-2), num = 10, base = 3) |
| | tol | numpy.linspace(0.1, 2.0, 20) |
| | Fit_intercept | [True, False] |
| | solver | [newton-cg, lbfgs, liblinear, sag, saga] |
| | max_iter | Range(50-501) |

4.3.1.2. การค้นหาแบบกริด

การกำหนดค่าให้ไฮเปอร์พารามิเตอร์แต่ละโมเดลเป็นไปดังตารางที่ 4.3.

**ตารางที่ 4. 3 การกำหนดค่าไฮเปอร์พารามิเตอร์ใช้กับวิธีการค้นหาแบบกริด**

| Model | Hyperparameter | Value |
|---|---|---|
| Decision Tree | criterion | [gini, entropy] |
| | splitter | [best, random] |
| | max_depth | [3, 10, 25, 40, 50] |
| | min_samples_split | [2, 10, 25, 50, 50] |
| | min_samples_leaf | [1, 10, 25, 50, 50] |
| | max_features | [auto, log2, None] |

| Model | Hyperparameter | Value |
|---|---|---|
| Decision Tree | random_state | 0 |
| Random Forest | n_estimators | [200, 500] |
| | max_features | [auto, sqrt, log2] |
| | max_depth | [140,150,160,170,180] |
| | criterion | [gini, entropy] |
| GBDT | loss | deviance |
| | learning_rate | [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2] |
| | min_samples_split | [140,150,160,170,180] |
| | min_samples_leaf | [gini, entropy] |
| GBDT | max_depth | [3,5,8] |
| | max_features | [log2, sqrt] |
| | criterion | [friedman_mse, mae] |
| | subsample | [0.5, 0.618, 0.8, 0.85, 0.9, 0.95, 1.0] |
| | n_estimators | 100 |
| KNN | n_neighbors | Range(2-31) |
| MLP | hidden_layer_sizes | [(50,50,50), (50,100,50), (100)] |
| | activation | [tanh, relu] |
| | solver | [sgd, adam] |
| | alpha | [0.0001, 0.05] |
| | learning_rate | [constant, adaptive] |
| | max_iter | [100, 500] |
| SVM | C | [0.1, 0.5, 1.0, 2.0] |
| | kernel | [linear, poly, rbf, sigmoid] |
| | degree | [2, 3, 5] |
| | gamma | [auto, scale] |
| | tol | [1e-5, 1e-3, 1e-2] |
| | max_iter | [-1, 50, 100] |
| | probability | True |
| Linear Regression | C | [0.1, 0.5, 1.0, 2.0] |
| | tol | [1e-5, 1e-3, 1e-2] |

| Model | Hyperparameter | Value |
|---|---|---|
| Linear Regression | Fit_intercept | [True, False] |
| | max_iter | [50, 100, 250, 300] |
| | solver | [newton-cg, lbfgs, liblinear, sag, saga] |

**4.3.2. ซอฟต์โหวต**

เพื่อการเพิ่มค่าความถูกต้องจึงให้แต่ละโมเดลช่วยกันโหวตแบบซอฟต์โหวตซึ่งเป็นการเพิ่มค่าความถูกต้องที่ใช้เงื่อนไขไม่เยอะ โดยโมเดลที่ใช้ไม่จำเป็นต้องเป็นประเภทเดียวกัน หรือพารามิเตอร์เดียวกัน จึงหวังว่าโมเดลที่ใช้แตกต่างกันในการทดลองนี้เมื่อร่วมกันโหวตแล้วจะมีแนวโน้มที่จะชดเชยข้อผิดพลาดของโมเดลบางตัวในเซตเดียวกัน โดยกำหนดให้แบ่งข้อมูลเป็นจำนวน 10 ส่วน โดยสุ่มอย่างเป็นระบบโดยเลือกข้อมูลทุกๆ 42 ระเบียน แต่ว่าค่าความถูกต้องนั้นไม่ได้เพิ่มขึ้นแบบแปรผันตรงกับจำนวนโมเดลที่ช่วยกันโหวต ดังนั้นจึงต้องหาสับเซตของโมเดล เพื่อจับกลุ่มให้ได้ทุกรูปแบบแล้วนำมาเปรียบเทียบกัน จะได้ว่าเซตของ Model = { Decision Tree, Random Forest, Gradient Boosted Trees (GBDT), k-Nearest Neighbors (KNN), Naïve Bayes, Multi-Layer Perceptron (MLP), Support Vector Machines (SVM), Linear Regression } สับเซตของเซต Model = P(Model) และมีจำนวนสมาชิก $\mathrm{n}\big(\mathrm{P}(\mathrm{Model})\big) = 2^8 - 1 = 255$ เมื่อไม่รวมเซตว่าง

# บทที่ 5
# ผลการศึกษา

## 5.1. ผลการหาไฮเปอร์พารามิเตอร์สำหรับโมเดล (ดี, ไม่ดี)

### 5.1.1. วิธีการค้นหาแบบสุ่ม

ค่าของแต่ละไฮเปอร์พารามิเตอร์ที่ดีที่สุดสำหรับโมเดลจำแนกคลาส ดี และไม่ดี โดยได้จากวิธีการค้นหาแบบสุ่มเป็นดังตารางที่ 5.1 และค่าเฉลี่ยของ Accuracy จากแต่ละโมเดลเป็นดังตารางที่ 5.2

**ตารางที่ 5. 1 ค่าไฮเปอร์พารามิเตอร์แต่ละโมเดล (ดี, ไม่ดี) จากวิธีการค้นหาแบบสุ่ม**

| Model | Hyperparameter | Value |
|---|---|---|
| Decision Tree | criterion | gini |
| | splitter | best |
| | max_depth | 37 |
| | min_samples_split | 4 |
| | min_samples_leaf | 3 |
| | max_features | None |
| | random_state | 42 |
| Random Forest | n_estimators | 822 |
| | max_features | sqrt |
| | max_depth | 70 |
| | min_samples_split | 5 |
| | min_samples_leaf | 1 |
| | bootstrap | False |
| | random_state | 42 |
| GBDT | n_estimators | 250 |
| | max_depth | 3 |
| | learning_rate | 0.1 |
| | random_state | 42 |
| KNN | n_neighbors | 104 |

| Model | Hyperparameter | Value |
|---|---|---|
| MLP | hidden_layer_sizes | (50, 100, 50) |
| | activation | tanh |
| | solver | sgd |
| | alpha | 0.05 |
| | learning_rate | constant |
| | max_iter | 500 |
| | random_state | 42 |
| SVM | C | 2.0 |
| | kernel | linear |
| | degree | 3 |
| | gamma | auto |
| | tol | 0.0027325701912391887 |
| | max_iter | 98 |
| | Probability | True |
| | random_state | 42 |
| Linear Regression | C | 0.6 |
| | tol | 0.00021775043826023207 |
| | Fit_intercept | False |
| | solver | saga |
| | max_iter | 264 |
| | random_state | 42 |

**ตารางที่ 5. 2 ค่าเฉลี่ย Accuracy แต่ละโมเดล (ดี, ไม่ดี) จากวิธีการค้นหาแบบสุ่ม**

| Model | Mean Accuracy | Time (sec) |
|---|---|---|
| Decision Tree | $0.833215 \pm 0.022619$ | 1.191 |
| Random Forest | $0.911572 \pm 0.023003$ | 395.907 |
| GBDT | $0.896525 \pm 0.020445$ | 174.468 |
| KNN | $0.778624 \pm 0.073110$ | 2.720 |
| MLP | $0.921643 \pm 0.012548$ | 4258.166 |
| SVM | $0.778078 \pm 0.026627$ | 8.864 |
| Logistic Regression | $0.926018 \pm 0.017725$ | 7.452 |

### 5.1.2. วิธีการค้นหาแบบกริด

ค่าของแต่ละไฮเปอร์พารามิเตอร์ที่ดีที่สุดสำหรับโมเดลจำแนกคลาส ดี และไม่ดี โดยได้จากวิธีการค้นหาแบบกริดเป็นดังตารางที่ 5.3 และค่าเฉลี่ยของ Accuracy จากแต่ละโมเดลเป็นดังตารางที่ 5.4

**ตารางที่ 5. 3 ไฮเปอร์พารามิเตอร์แต่ละโมเดล (ดี, ไม่ดี) จากวิธีการค้นหาแบบกริด**

| Model | Hyperparameter | Value |
|---|---|---|
| Decision Tree | criterion | entropy |
| | splitter | best |
| | max_depth | 50 |
| | min_samples_split | 10 |
| | min_samples_leaf | 1 |
| | max_features | None |
| | random_state | 42 |
| Random Forest | n_estimators | 200 |
| | max_features | log2 |
| | max_depth | 140 |
| | criterion | gini |
| | random_state | 42 |
| GBDT | loss | deviance |
| | learning_rate | 0.05 |
| | min_samples_split | 0.1 |
| | min_samples_leaf | 0.1 |
| | max_depth | 3 |
| | max_features | sqrt |
| | criterion | friedman_mse |
| | subsample | 1.0 |
| | n_estimators | 100 |
| | random_state | 42 |
| KNN | n_neighbors | 2 |
| MLP | hidden_layer_sizes | (50,100,50) |
| | activation | tanh |
| | solver | sgd |

| Model | Hyperparameter | Value |
|---|---|---|
| MLP | alpha | 0.05 |
| | learning_rate | constant |
| | max_iter | 500 |
| | random_state | 42 |
| SVM | C | 1.0 |
| | kernel | sigmoid |
| | degree | 2 |
| | gamma | scale |
| | tol | 1e-05 |
| | max_iter | -1 |
| | probability | True |
| | random_state | 42 |
| Linear Regression | C | 0.5 |
| | tol | 1e-05 |
| | Fit_intercept | False |
| | solver | sag |
| | max_iter | 100 |
| | random_state | 42 |

**ตารางที่ 5. 4 ค่าเฉลี่ย Accuracy แต่ละโมเดล (ดี, ไม่ดี) จากวิธีการค้นหาแบบกริด**

| Model | Mean Accuracy | Time (sec) |
|---|---|---|
| Decision Tree | $0.858884 \pm 0.021931$ | 17.923 |
| Random Forest | $0.931057 \pm 0.015246$ | 233.198 |
| GBDT | $0.652115 \pm 0.047911$ | 13660.185 |
| KNN | $0.736074 \pm 0.028924$ | 0.539 |
| MLP | $0.921643 \pm 0.012548$ | 3941.231 |
| SVM | $0.914108 \pm 0.014850$ | 297.030 |
| Logistic Regression | $0.925389 \pm 0.016497$ | 21.392 |

### 5.2.3. การเลือกไฮเปอร์พารามิเตอร์

เลือกตัวแทน Parameter ที่ดีที่สุดจากทั้งสองวิธีให้แต่ละ Model โดยการเปรียบเทียบค่าเฉลี่ยของ Accuracy เพื่อนำไปใช้ในกระบวนการ Vote ดังแสดงในตารางที่ 5.5.

**ตารางที่ 5. 5 เลือกไฮเปอร์พารามิเตอร์ให้แต่ละโมเดล (ดี, ไม่ดี)**

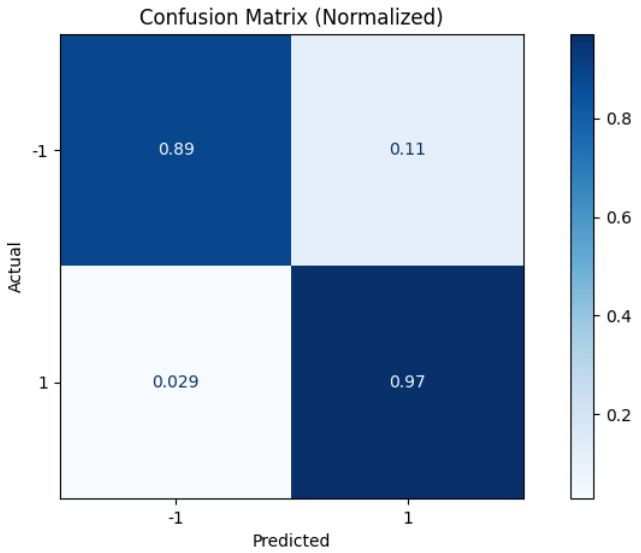
| Model | Search | Mean Accuracy |
|---|---|---|
| Decision Tree | Grid Search | $0.858884 \pm 0.021931$ |
| Random Forest | Grid Search | $0.931057 \pm 0.015246$ |
| GBDT | Random Search | $0.896525 \pm 0.020445$ |
| KNN | Grid Search | $0.736074 \pm 0.028924$ |
| MLP | Grid Search | $0.921643 \pm 0.012548$ |
| SVM | Grid Search | $0.914108 \pm 0.014850$ |
| Logistic Regression | Grid Search | $0.925389 \pm 0.016497$ |

## 5.2. ผลการหาโมเดล (ดี, ไม่ดี) ที่ดีที่สุดจากการโหวต

จากการจับกลุ่มโหวตผลปรากฏว่าโมเดล Random Forest นั้นได้ค่าเฉลี่ยของ Accuracy ดีที่สุด โดยได้ค่าเฉลี่ย Accuracy = $0.931057 \pm 0.015246$

### 5.2.1. ผลการทดสอบโมเดล (ดี, ไม่ดี) ด้วยข้อมูลทดสอบ

การทดสอบด้วยข้อมูลทดสอบโดยมีข้อมูลคลาส ดี (1) จำนวน 206 ระเบียน และ ไม่ดี (-1) จำนวน 193 ระเบียน ด้วยโมเดล Random Forest ได้คอนฟิวชันแมทริกซ์ ดังรูปที่ 5.1




**รูปที่ 5. 1 คอนฟิวชันแมทริกซ์การทดสอบด้วยข้อมูลทดสอบของโมเดล (ดี, ไม่ดี)**

จากรูปที่ 5.1 จะสามารถคำนวณหาค่า Precision, Recall, F1-Score และ Accuracy ได้ดังตารางที่ 5.6

**ตารางที่ 5. 6 ค่า Precision, Recall, F1-Score และ Accuracy ของโมเดล (ดี, ไม่ดี)**

| Model | class | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Random Forest | ไม่ดี (-1) | 0.966102 | 0.886010 | 0.924324 |
| | ดี (1) | 0.900901 | 0.970874 | 0.934579 |
| **Average** | | 0.933502 | 0.928442 | 0.929452 |
| **Accuracy** | | 0.929825 | | |

## 5.3. ผลการหาไฮเปอร์พารามิเตอร์สำหรับโมเดล (รุนแรง, ไม่รุนแรง)

### 5.3.1. วิธีการค้นหาแบบสุ่ม

ค่าของแต่ละไฮเปอร์พารามิเตอร์ที่ดีที่สุดสำหรับโมเดลจำแนกคลาส รุนแรง และไม่รุนแรง โดยได้จากวิธีการค้นหาแบบสุ่มเป็นดังตารางที่ 5.7 และค่าเฉลี่ยของ Accuracy จากแต่ละโมเดลเป็นดังตารางที่ 5.8

**ตารางที่ 5. 7 ค่าไฮเปอร์พารามิเตอร์แต่ละโมเดล (รุนแรง, ไม่รุนแรง) จากวิธีการค้นหาแบบสุ่ม**

| Model | Hyperparameter | Value |
|---|---|---|
| Decision Tree | criterion | entropy |
| | splitter | best |
| | max_depth | 19 |
| | min_samples_split | 44 |
| | min_samples_leaf | 2 |
| | max_features | None |
| | random_state | 42 |
| Random Forest | n_estimators | 288 |
| | max_features | auto |
| | max_depth | 90 |
| | min_samples_split | 5 |
| | min_samples_leaf | 1 |

| Model | Hyperparameter | Value |
|---|---|---|
| Random Forest | bootstrap | False |
| | random_state | 42 |
| GBDT | n_estimators | 50 |
| | max_depth | 5 |
| | learning_rate | 0.1 |
| | random_state | 42 |
| KNN | n_neighbors | 53 |
| MLP | hidden_layer_sizes | (50, 100, 50) |
| | activation | tanh |
| | solver | adam |
| | alpha | 0.0001 |
| | learning_rate | constant |
| | max_iter | 250 |
| | random_state | 42 |
| SVM | C | 0.944444 |
| | kernel | sigmoid |
| | degree | 3 |
| | gamma | scale |
| | tol | 0.0011758914435816995 |
| | max_iter | 99 |
| | Probability | True |
| | random_state | 42 |
| Linear Regression | C | 0.3 |
| | tol | 3.213205906864567e-06 |
| | Fit_intercept | True |
| | solver | sag |
| | max_iter | 376 |
| | random_state | 42 |

**ตารางที่ 5. 8 ค่าเฉลี่ย Accuracy แต่ละโมเดล (รุนแรง, ไม่รุนแรง) จากวิธีการค้นหาแบบสุ่ม**

| Model | Mean Accuracy | Time (sec) |
|---|---|---|
| Decision Tree | $0.757460 \pm 0.041467$ | 1.117 |
| Random Forest | $0.847445 \pm 0.033653$ | 228.963 |
| GBDT | $0.830989 \pm 0.037124$ | 198.795 |
| KNN | $0.520400 \pm 0.009530$ | 1.635 |
| MLP | $0.858330 \pm 0.031697$ | 2381.999 |
| SVM | $0.771177 \pm 0.032231$ | 8.838 |
| Logistic Regression | $0.844539 \pm 0.042021$ | 10.635 |

### 5.3.2. วิธีการค้นหาแบบกริด

ค่าของแต่ละไฮเปอร์พารามิเตอร์ที่ดีที่สุดสำหรับโมเดลจำแนกคลาส รุนแรง และไม่-รุนแรง โดยได้จากวิธีการค้นหาแบบกริดเป็นดังตารางที่ 5.9 และค่าเฉลี่ยของ Accuracy จากแต่ละโมเดลเป็นดังตารางที่ 5.10

**ตารางที่ 5. 9 ค่าไฮเปอร์พารามิเตอร์แต่ละโมเดล (รุนแรง, ไม่รุนแรง) จากวิธีการค้นหาแบบกริด**

| Model | Hyperparameter | Value |
|---|---|---|
| Decision Tree | criterion | entropy |
| | splitter | random |
| | max_depth | 40 |
| | min_samples_split | 10 |
| | min_samples_leaf | 1 |
| | max_features | None |
| | random_state | 42 |
| Random Forest | n_estimators | 200 |
| | max_features | auto |
| | max_depth | 150 |
| | criterion | Gini |
| | random_state | 42 |
| GBDT | loss | deviance |
| | learning_rate | 0.075 |
| | min_samples_split | 0.1 |
| | min_samples_leaf | 0.1 |

| Model | Hyperparameter | Value |
|---|---|---|
| GBDT | max_depth | 3 |
| | max_features | sqrt |
| | criterion | friedman_mse |
| | subsample | 0.9 |
| | n_estimators | 100 |
| | random_state | 42 |
| KNN | n_neighbors | 3 |
| MLP | hidden_layer_sizes | (50,100,50) |
| | activation | tanh |
| | solver | adam |
| | alpha | 0.0001 |
| | learning_rate | constant |
| | max_iter | 500 |
| | random_state | 42 |
| SVM | C | 2.0 |
| | kernel | Rbf |
| | degree | 2 |
| | gamma | scale |
| | tol | 1e-05 |
| | max_iter | -1 |
| | probability | True |
| | random_state | 42 |
| Linear Regression | C | 2.0 |
| | tol | 1e-05 |
| | Fit_intercept | True |
| | solver | saga |
| | max_iter | 300 |
| | random_state | 42 |

**ตารางที่ 5. 10 ค่าเฉลี่ย Accuracy แต่ละโมเดล (รุนแรง, ไม่รุนแรง) จากวิธีการค้นหาแบบกริด**

| Model | Mean Accuracy | Time (sec) |
|---|---|---|
| Decision Tree | $0.786153 \pm 0.035253$ | 15.277 |
| Random Forest | $0.803795 \pm 0.043589$ | 163.907 |
| GBDT | $0.733062 \pm 0.037417$ | 11541.639 |
| KNN | $0.550407 \pm 0.025077$ | 0.296 |
| MLP | $0.858330 \pm 0.031697$ | 2773.312 |
| SVM | $0.850167 \pm 0.032699$ | 197.144 |
| Logistic Regression | $0.840504 \pm 0.043523$ | 35.363 |

### 5.3.3. การเลือกไฮเปอร์พารามิเตอร์

เลือกตัวแทนไฮเปอร์พารามิเตอร์ที่ดีที่สุดจากทั้งสองวิธีให้แต่ละโมเดล โดยการเปรียบเทียบค่าเฉลี่ยของ Accuracy เพื่อนำไปใช้ในกระบวนการโหวตดังแสดงในตารางที่ 5.11

**ตารางที่ 5. 11 เลือกไฮเปอร์พารามิเตอร์ให้แต่ละโมเดล (รุนแรง, ไม่รุนแรง)**

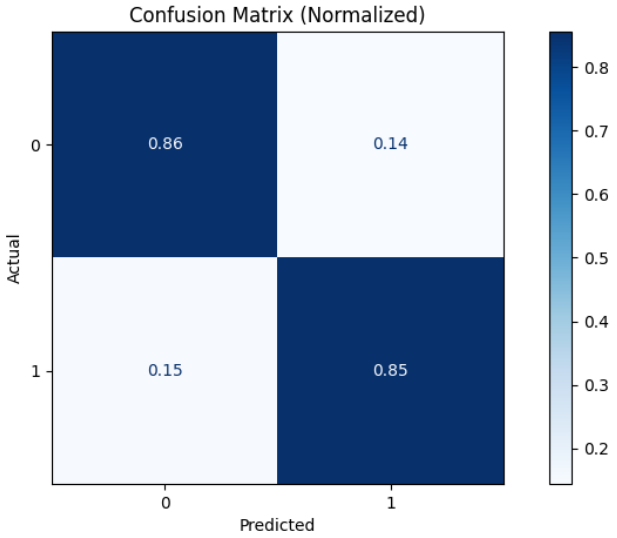
| Model | Search | Mean Accuracy |
|---|---|---|
| Decision Tree | Random Search | $0.757460 \pm 0.041467$ |
| Random Forest | Random Search | $0.847445 \pm 0.033653$ |
| GBDT | Random Search | $0.830989 \pm 0.037124$ |
| KNN | Grid Search | $0.550407 \pm 0.025077$ |
| MLP | Random Search | $0.858330 \pm 0.031697$ |
| SVM | Grid Search | $0.850167 \pm 0.032699$ |
| Logistic Regression | Random Search | $0.844539 \pm 0.042021$ |

## 5.4. ผลการหาโมเดล (รุนแรง, ไม่รุนแรง) ที่ดีที่สุดจากการโหวต

จากการจับกลุ่มโหวตผลปรากฏว่ากลุ่มของ {Radom Forest, GBDT, MLP, SVM, Logistic Regression} นั้นได้ค่าเฉลี่ยของ Accuracy ดีที่สุด โดยได้ค่าเฉลี่ย Accuracy = $0.863754 \pm 0.027188$

### 5.4.1. ผลการทดสอบโมเดล (รุนแรง, ไม่รุนแรง) ด้วยข้อมูลทดสอบ

การทดสอบด้วยข้อมูลทดสอบโดยมีข้อมูลคลาส รุนแรง (1) จำนวน 94 ระเบียน และไม่-รุนแรง (0) จำนวน 90 ระเบียน ด้วยโมเดลที่ประกอบไปด้วยสมาชิก {Radom Forest, GBDT, MLP, SVM, Logistic Regression} ได้คอนฟิวชันแมทริกซ์ ดังภาพ 5.2




**รูปที่ 5. 2 คอนฟิวชันแมทริกซ์การทดสอบด้วยข้อมูลทดสอบของโมเดล (รุนแรง, ไม่รุนแรง)**

จากรูปที่ 5.2 จะสามารถคำนวณหาค่า Precision, Recall, F1-Score และ Accuracy ได้ดังตารางที่ 5.12

**ตารางที่ 5. 12 ค่า Precision, Recall, F1-Score และ Accuracy ของโมเดล (รุนแรง, ไม่รุนแรง)**

| Model | class | Precision | Recall | F1-Score |
|---|---|---|---|---|
| {Radom Forest, GBDT, MLP, SVM, Logistic Regression} | ไม่รุนแรง (0) | 0.846154 | 0.855555 | 0.850829 |
| | รุนแรง (1) | 0.860215 | 0.851063 | 0.855615 |
| **Average** | | 0.853185 | 0.853309 | 0.853222 |
| **Accuracy** | | 0.853260 | | |

# บทที่ 6
# สรุปผลการศึกษา และวิจารณ์ผลการศึกษา

การปฏิบัติงานสหกิจ ณ ศูนย์ศรีพัฒน์ คณะแพทยศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้รับมอบหมายให้ศึกษาและพัฒนาระบบ การวิเคราะห์ความคิดเห็นจากลูกค้าเพื่อพัฒนามาตรการดำเนินการอย่างรวดเร็วสำหรับโรงพยาบาลระดับตติยภูมิ ที่ได้มีการบันทึกไว้ในระยะเวลา 6 เดือน ตั้งแต่ 19 เมษายน ถึง 30 กันยายน พ.ศ. 2564 ข้อเสนอแนะ แนวทางในอนาคต และงานอื่นๆ ที่ได้รับมอบหมายจะกล่าวในบทนี้

## 6.1. ข้อเสนอแนะ และแนวทางในอนาคต

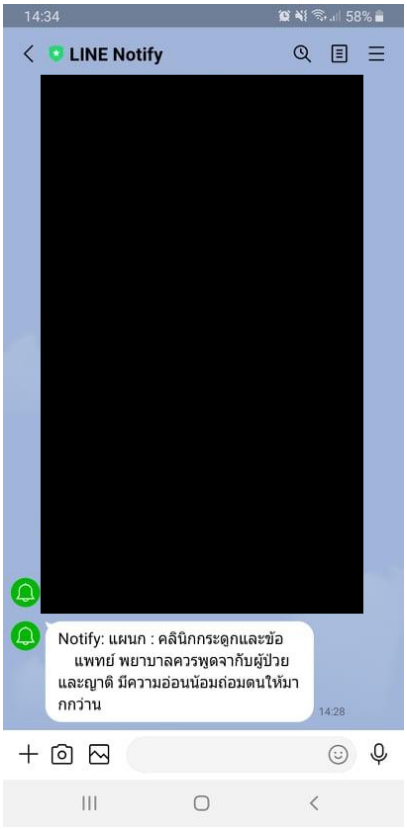
1) เนื่องจากขั้นตอนการค้นหาแบบสุ่ม และการค้นหาแบบกริดใช้เวลานานมาก ดังนั้นจึงแนะนำให้การเขียนโค้ดที่รองรับการประมวนผลบนหน่วยประมวลผลกราฟิกส์ (GPU) ที่มีประสิทธิภาพสูงจะทำให้ลดระยะเวลาในการประมวนผลได้
2) เมื่อโมเดลการเรียนรู้ของเครื่องถูกสร้างขึ้นในระยะเวลาผ่านไปมันจะล้าสมัยเนื่องจากมีข้อมูลนำเข้าเพิ่มขึ้นมาเรื่อยๆในทุกๆวัน เพื่อทำให้เกิดความแม่นยำจึงจำเป็นต้องมีการนำโมเดลกลับมาฝึกสอนใหม่เมื่อพบว่าลักษณะข้อมูลมีการเปลี่ยนแปลงไปจากเดิมจากการตรวจสอบชนิดของภาษาที่ใช้ในการแสงดความคิดเห็น และกลุ่มคำ (Word Cloud) ที่พบแตกต่างไปจากเดิม

## 6.2. งานอื่น ๆ ที่ได้รับมอบหมาย

เนื่องจากผู้ใช้งานในระบบนี้คือ ผู้รับบริการการรักษา และพนักงานที่ไม่ใช่ ผู้พัฒนา (Developer) ดังนั้นจึงจำเป็นต้องทำระบบออกมาในรูปแบบการแจ้งเตือนผ่านไลน์ (Line Notification) และ เว็บแอป-พลิเคชัน (Web Application) เพื่อความสะดวกแก่ผู้ใช้งาน และใช้งานง่าย

### 6.2.1. การแจ้งเตือนผ่านไลน์ (Line Notification)

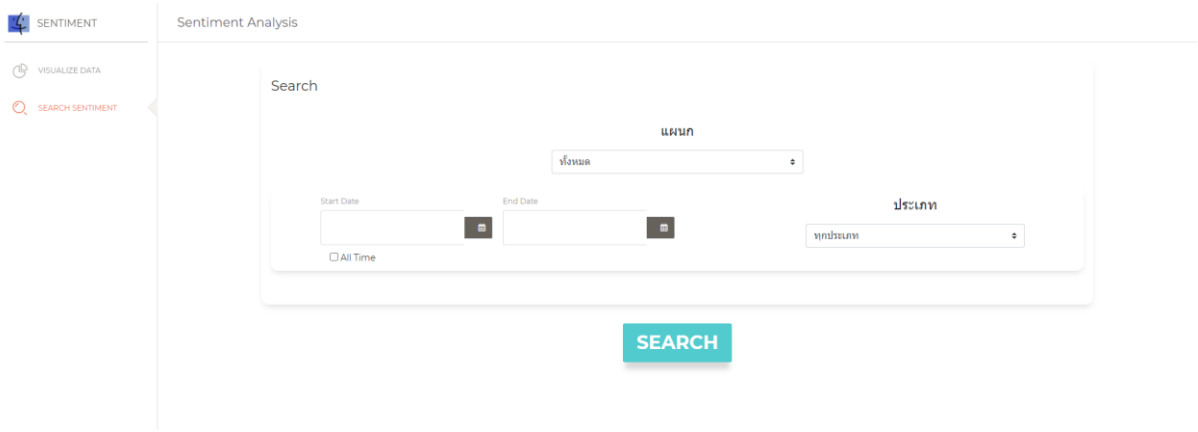
การแจ้งเตือนความคิดเห็นที่เกิดขึ้นแก่ผู้รับบริการการรักษาให้กับพนักงานที่เกี่ยวข้องทราบซึ่งจะอยู่ในรูปแบบของการแจ้งเตือนผ่านไลน์ โดยความคิดเห็นที่จะแจ้งเตือนนั้นจะเป็นความคิดเห็นในแง่ไม่ดี และรุนแรง รายละเอียดที่แจ้งประกอบไปด้วย แผนกที่เกี่ยวข้อง และความคิดเห็นที่ได้รับ ดังตัวอย่างในภาพที่ 6.1



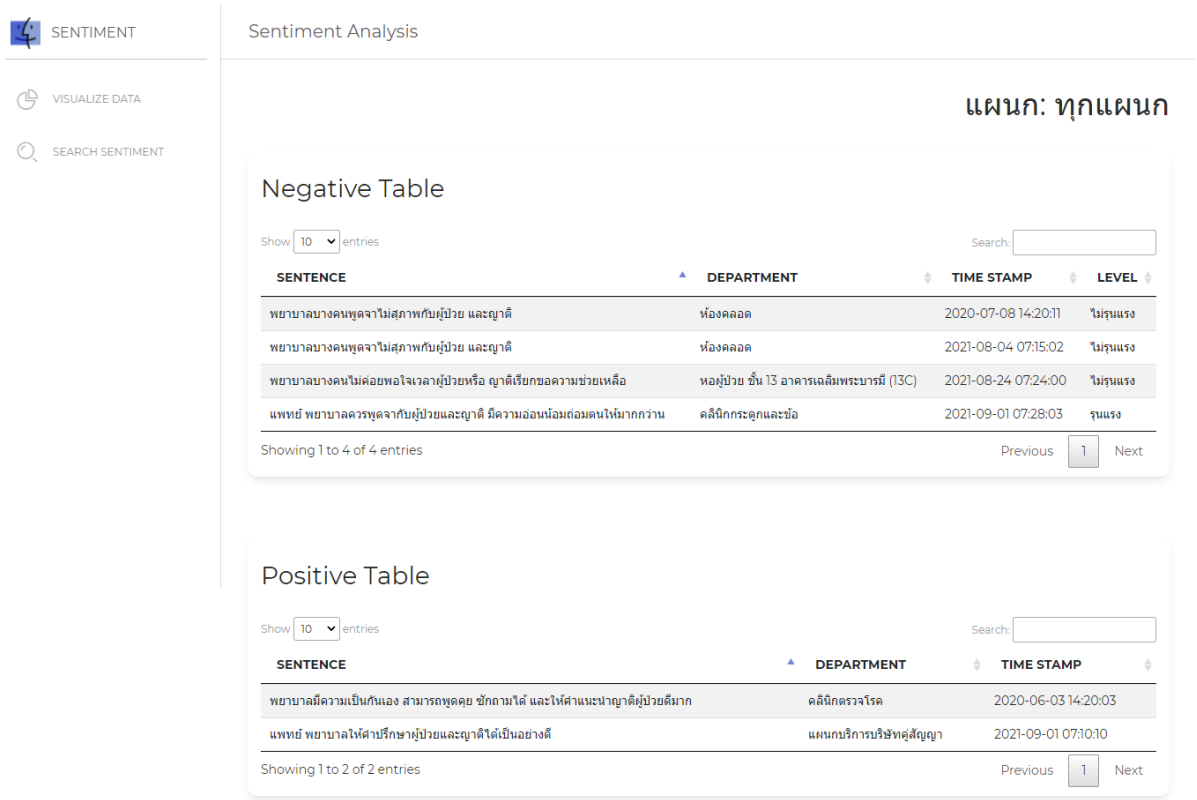

**รูปที่ 6. 1 การแจ้งเตือนผ่านไลน์**

### 6.2.2. ตารางค้นคืนข้อมูล

การค้นคืนข้อมูลจะสามารถทำได้จากการกรอกแบบฟอร์มค้นหาข้อมูล โดยสามารถค้นหาได้ตามแผนก ช่วงเวลา และประเภทความคิดเห็น ดังรูปที่ 6.2 ทำให้ได้ตารางค้นคืนข้อมูลที่แยกตามแผนก และประเภทความคิดเห็น ดังรูปที่ 6.3



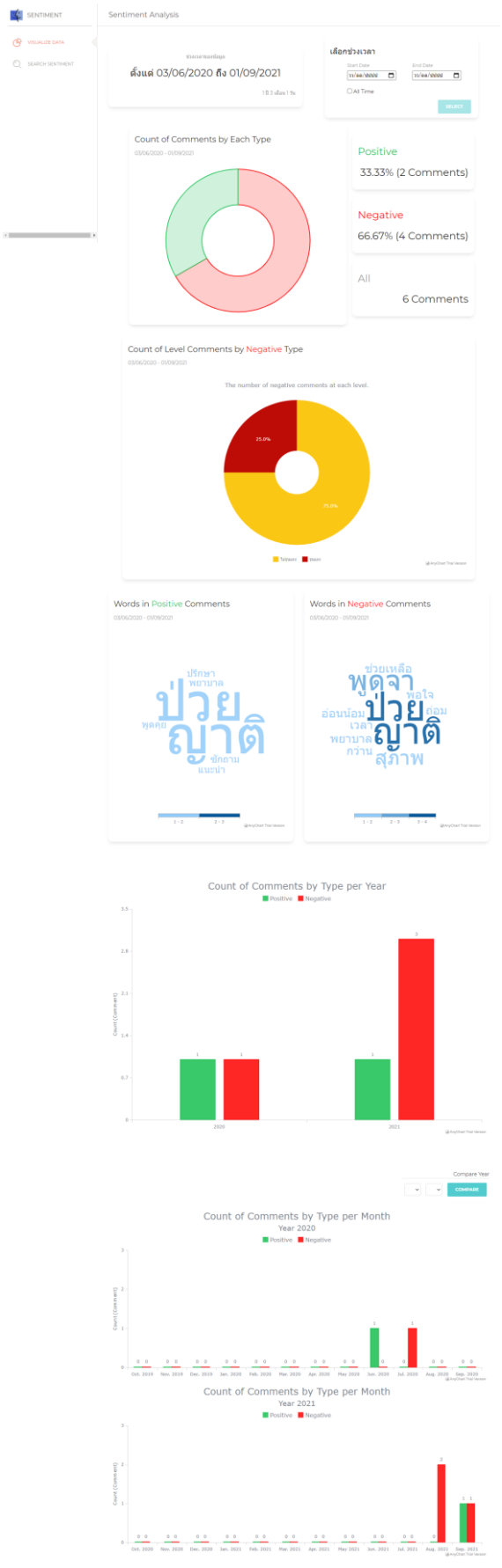

**รูปที่ 6. 2 แบบฟอร์มค้นหาข้อมูล**




**รูปที่ 6. 3 ตารางค้นคืนข้อมูล**

### 6.2.2. การนำเสนอภาพข้อมูล (Data Visualization)

นำข้อมูลที่รอบรวมได้มาแสดงในรูปแบบของ กลุ่มคำ (Word Cloud) และกราฟต่างๆ ที่สามารถค้นหาข้อมูลตามช่วงเวลาได้ ดังรูปที่ 6.4 เพื่อให้เข้าใจง่าย เห็นภาพรวมได้ชัดเจน และง่ายต่อการจดจำ




**รูปที่ 6. 4 การนำเสนอภาพข้อมูล**

# เอกสารอ้างอิง

[1] บริษัท แอ๊ฟฟินีตี้ โซลูชั่น จำกัด. การรับฟังเสียงของลูกค้า VOICE OF CUSTOMER – VOC. Available at: URL:https://www.thailandcontactcenter.com/voiceofcustomer. Accessed Sep 07, 2021.

[2] Mr.P L. Classification หรือ การจำแนกประเภท. Available at: URL: https://medium.com/mmp-liเริ่มเรียน-machine-learning-0-100-introduction-1c58e516bfcd. Accessed Sep 07, 2021.

[3] Ricco Smart Data. การทำความสะอาดข้อมูล Data cleansing หรือ Data cleaning คืออะไร. Available at: URL:https://riccosmartdata.com/data-cleansing-or-data-cleaning/. Accessed Sep 07, 2021.

[4] Keng Surapong. Training/Validation/Test Set Split. Available at: URL:https://www.bualabs.com/archives/532/what-is-training-set-why-train-test-split-training-set-validation-set-test-set/. Accessed Sep 07, 2021.

[5] สกุลรัตน์ ขุนสูงเนิน, โษฑศ์รัตต ธรรมบุษดี, สมเกียรติ วัฒนศิริชยักลุ. การใช้การวิเคราะห์เชิงทำนายสำหรับการระบุสถานะการจำหน่ายและการรอดชีวิตในผู้ป่วยภาวะติดเชื้อและผู้ป่วยภาวะช็อกจากเหตุพิษติดเชื้อบนพื้นฐานของปัจจัย. KKU RESEARCH JOURNAL (GRADUATE STUDIES), 2019 April – June;19(2):117-130.

[6] iOmelet. จัดการข้อมูลสินค้าซ้ำซ้อนด้วย ‘Product Similarity Search’. Available at: URL: https://read.montivory.com/2019/09/12/product-similarity-search/. Accessed Sep 07, 2021.

[7] Thitiya Trithipkaiwanpon. การปรับไฮเปอร์พารามิเตอร์อัตโนมัติด้วยกระบวนการแบบฝูงผึ้ง. Available at: URL:https://read.montivory.com/2019/09/12/product-similarity-search/. Accessed Sep 07, 2021.

[8] วรุจิรา ธรรมสมบัต. ระบบสนับสนุนการตัดสินใจในการเลือกใช้แพคเกจอินเทอร์เน็ตมือถือโดยใช้ต้นไม้ตัดสินใจ (วิทยาลัยราชพฤกษ์, 2012), p. 10-11.

[9] วันวิสาข์ ชนะประเสริฐ. การประยุกต์ใช้เทคนิคเหมืองข้อมูลเพื่อแนะนำอาชีพสำหรับนักศึกษาปริญญาตรีคณะโบราณคดี มหาวิทยาลัยศิลปากร (วิทยานิพนธ์ปริญญามหาบัณฑิต สนเทศศาสตร์เพื่อการศึกษา มหาวิทยาลัยศิลปากร, 2016), p. 20.

[10] ชนาธิป พันทะยักษ์. การวิเคราะห์ข้อมูลผลคะแนนสอบ O-NET กรณีศึกษา โรงเรียนบาง-ประกอกวิทยาคมด้วยเทคนิคดาต้าไมนิงค์ (วิทยานิพนธ์ปริญญาบัณฑิต คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏกาญจนบุรี, 2019), p. 6-13.

[11] Ying Hu, Oleg Kremnyov, Ivan Kuzmin. Faster Gradient-Boosting Decision Trees Available at: URL:https://wp.ironhorse.dev/tech-decoded/resources/faster-gradient-boosting-decision-trees/. Accessed Oct 06, 2021.

[12] นิเวศ จิระวิชิตชัย. การค้นหาเทคนิคเหมืองข้อมูลเพื่อสร้างโมเดลการวิเคราะห์โรคอัตโนมัติ (สถาบันวิจัยและพัฒนา มหาวิทยาลัยราชภัฏสวนสุนันทา, 2010), p. 14-15.

[13] อรทัย เจริญสิทธิ์. การวิเคราะห์การถดถอยโลจิสติกแบบไบนารีสำหรับการวิจัยทางสังคมศาสตร์. SAU JOURNAL OF SOCIAL SCIENCES & HUMANILITIES. 2017 July - December;1(2):1-9.

[14] รติพร จันทร์กลั่น. การปรับปรุงอลักอริทึมซัพพอร์ตเวกเตอร์แมชชีนสำหรับการจำแนกข้อมูล ภาพไบโอเมตริกซ์ (วิทยานิพนธ์ปริญญามหาบัณฑิต วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี, 2014), p. 7-9.

[15] Scikit Learn. Voting Classifier. Available at: URL:https://scikit-learn.org/stable/modules/ensemble.html#voting-classifier. Accessed Sep 07, 2021.

[16] chengz. วัดประสิทธิภาพ Model จาก Confusion Matrix. Available at: URL:https://medium.com/@cheng3374/วัดประสิทธิภาพ-model-จาก-confusion-matrix-69d391bcd48. Accessed Sep 07, 2021.