การเพิ่มประสิทธิภาพการจำแนกตั๋วงานสำหรับระบบจัดการบริการด้านไอที โดยใช้การเรียนรู้ของเครื่องจักร

สิรภพ แสนบุตร[1] และ ฐิติรัตน์ ศิริบวรรัตนกุล[2*]
สถาบันบัณฑิตพัฒนบริหารศาสตร์ ถ.เสรีไทย แขวงคลองจั่น เขตบางกะปิ กรุงเทพฯ 10240

[*] Corresponding Author: thitirat@as.nida.ac.th
[1] นักศึกษาระดับปริญญาโท คณะสถิติประยุกต์
[2] ผู้ช่วยศาสตราจารย์ คณะสถิติประยุกต์

ข้อมูลบทความ

ประวัติบทความ :
รับเพื่อพิจารณา : 22 เมษายน 2564
แก้ไข : 12 ตุลาคม 2564
ตอบรับ : 20 ตุลาคม 2564
DOI : 10.14456/kmuttrd.2021.20

คำสำคัญ :
การจำแนกประเภทตั๋วงาน / การเรียนรู้ด้วยเครื่องจักร / การประมวลผลภาษาธรรมชาติ / ระบบจัดการบริการด้านไอที

บทคัดย่อ

งานวิจัยนี้มีจุดประสงค์เพื่อศึกษาแนวทางการเพิ่มประสิทธิภาพการจำแนกตั๋วงานสำหรับระบบจัดการบริการด้านไอทีโดยใช้การเรียนรู้ของเครื่องจักร ซึ่งผู้วิจัยได้ทำการทดลองกับข้อมูลตั๋วงานด้านไอทีจำนวน 10,608 ตั๋วงาน โดยนำข้อความที่ผู้ใช้บริการแจ้งงานเข้ามาผ่านตั๋วงานหนึ่งๆ เข้าสู่กระบวนการประมวลผลภาษาธรรมชาติ ได้แก่ การสกัดคำ การกำจัดคำ การสะกดคำ และอื่น ๆ จากนั้นจึงนำผลที่ได้มาเรียนรู้และทำนายผ่านตัวแบบทั้งหมด 6 ตัวแบบ ได้แก่ Multinomial Naïve Byes, Support Vector Machine, Logistics Regression, Random Forest, Stacking Model with Bagging และ XG-Boosting บนการสกัดคุณลักษณะทั้งหมด 2 แบบ ได้แก่ Count Vectorization และ TF-IDF เพื่อเปรียบเทียบประสิทธิภาพ ผลการศึกษาพบว่าตัวแบบที่ใช้เทคนิค Ensemble ดังเช่น XG-Boosting บนการสกัดคุณลักษณะแบบ TF-IDF สามารถประยุกต์ใช้งานกับข้อมูลชุดอื่นและให้ผลการทำนายแม่นยำที่สุดด้วยดัชนี F1-Score เท่ากับ 81.21% ทั้งยังช่วยลดเวลาในกระบวนการจำแนกประเภทตั๋วงานทั้งหมดให้เหลือน้อยกว่า 1 วัน ตลอดจนเพิ่มระดับความพึงพอใจเฉลี่ยของผู้ใช้บริการได้สูงสุดถึง 8.17%

Using Machine Learning to Improve Ticket Classification for IT Service Management System

Siraphop Seanbudh[1] and Thitirat Siriborvornratanakul[2*]
National Institute of Development Administration, Serithai Road, Klong-Chan, Bangkapi, Bangkok 10240

** Corresponding Author: thitirat@as.nida.ac.th*
[1] Master's Student, Graduate School of Applied Statistics.
[2] Assistant Professor, Graduate School of Applied Statistics.

Article Info

Article History:
Received: April 22, 2021
Revised: October 12, 2021
Accepted: October 20, 2021
DOI : 10.14456/kmuttrd.2021.20

Keywords:
Ticket Classification / Machine Learning / Natural Language Processing / IT Service Management

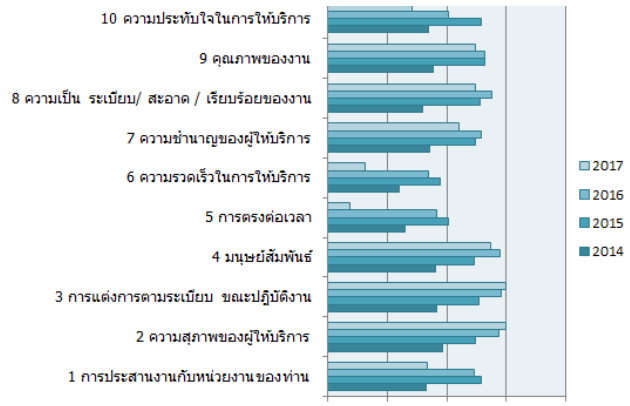
Abstract

In this research, we propose a maching learning based system for automatic IT support ticket classification. A total of 10,608 tickets (written in Thai) were used where each ticket was processed by several techniques of natural language processing, including word tokenization, word removal and word spelling, among others. The processed data were then fed to six different machine learning models—Multinomial Naïves Bayes, Support Vector Machine, Logistics Regression, Random Forest, Stacking Model with Bagging, and XG-Boosting. Each machine learning model was tested with two feature extraction schemes—Count Vectorization and TF-IDF. Our experiments revealed that the most accurate solution was obtained through the model utilizing ensemble technique, particularly XG-Boosting with TF-IDF feature extraction. Applying this best solution to an unknown dataset, we obtained good results, both quantitatively and qualitatively. In the case of quantitative results, we achieved the highest F1-Score of 81.21%. In the case of qualitative results, this system speeded up the IT ticket classication process from more than one day with manual classification to less than one day with our system. Consecutively, the overall users' satisfaction regarding IT service management system increased by 8.17%.

1. บทนำ

ระบบจัดการบริการด้านไอที (IT Service Management หรือ ITSM) เป็นระบบที่ถูกใช้งานอย่างแพร่หลายทั่วโลกในฐานะแพลตฟอร์มตัวกลางระหว่างผู้ให้บริการด้านเทคโนโลยีสารสนเทศกับผู้ใช้บริการ โดยหน้าที่ของระบบนั้นมีได้หลากหลาย อาทิ รับ-ส่งปัญหา แจ้งปัญหาการใช้งาน แจ้งความประสงค์ต่าง ๆ ด้านเทคโนโลยีสารสนเทศ เป็นต้น ทั้งนี้ เพื่อเป้าหมายสูงสุดในการสนับสนุนความต้องการและจัดการระบบไอทีภายในองค์กรให้มีประสิทธิภาพ ในงานวิจัยชิ้นนี้ ผู้วิจัยทำการศึกษาปัจจัยที่ส่งผลต่อการเพิ่มประสิทธิภาพและลดเวลาที่ใช้ในการจำแนกประเภทตั๋วงานสนับสนุนด้านไอที (IT Support Ticket) ของระบบจัดการบริการด้านไอที โดยผู้วิจัยได้เลือกกรณีศึกษาของปัญหาที่เกิดขึ้นจริงในกระบวนการทำงานของฝ่ายเทคโนโลยีสารสนเทศในบริษัทเอกชนแห่งหนึ่ง ซึ่งความเป็นมาเริ่มต้นคือในทุก ๆ ช่วงไตรมาสสุดท้ายของปี พนักงานภายในบริษัททุกหน่วยงานจะต้องทำการกรอกแบบสอบถามเพื่อประเมินความพึงพอใจในการใช้บริการจากผู้ให้บริการด้านไอทีซึ่งเป็นพนักงานฝ่ายเทคโนโลยีสารสนเทศของบริษัทเอง การให้คะแนนจะเป็นลักษณะการให้เลือกตั้งแต่ 1 จนถึง 5 คะแนนเรียงตามระดับความพึงพอใจจากน้อยไปมาก โดยการประเมินนี้มี 10 ตัวชี้วัด ได้แก่ การประสานงานกับหน่วยงานของท่าน ความสุภาพของผู้ให้บริการ การแต่งกายตามระเบียบขณะปฏิบัติงาน มนุษยสัมพันธ์ การตรงต่อเวลา ความรวดเร็วในการให้บริการ ความชำนาญของผู้ให้บริการ ความเป็นระเบียบ/สะอาด /เรียบร้อยของงาน คุณภาพของงาน และ ความประทับใจในการให้บริการ จากสถิติข้อมูลคะแนนประเมินย้อนหลังตั้งแต่ พ.ศ. 2557 ถึง พ.ศ. 2560 (รูปที่ 1) จะเห็นว่าผลคะแนนประเมินเฉลี่ยต่ำที่สุด 2 ลำดับแรกเป็นประเด็นที่เกี่ยวข้องกับเรื่อง “เวลา” คือความตรงต่อเวลาและความรวดเร็วในการให้บริการ โดยนอกจากสองหัวข้อนี้จะได้คะแนนความพึงพอใจเฉลี่ยน้อยที่สุดแล้ว ยังมีคะแนนน้อยกว่าตัวชี้วัดอื่น ๆ อย่างมีนัยสำคัญ อีกทั้งยังมีแนวโน้มที่ลดลงอย่างเห็นได้ชัดในช่วงปี พ.ศ. 2558 ถึง พ.ศ. 2560 ถือเป็นประเด็นปัญหาที่ควรต้องได้รับการพิจารณาแก้ไขอย่างเร่งด่วน

จากการหารือร่วมกันของทีมงานฝ่ายเทคโนโลยีสารสนเทศและการติดตามกระบวนการทำงานที่เกี่ยวข้องกับตั๋วงานโดยละเอียดทุกขั้นตอน เริ่มตั้งแต่กระบวนการจำแนก กระบวนการมอบหมายงาน และจบที่กระบวนการดำเนินการ ทำให้ทีมงานพบความจริงที่ว่า ในการทำงานจริงกระบวนการที่มักกินเวลาเกินจากที่คาดการณ์ไว้มากที่สุดคือ “กระบวนการจำแนกประเภทของตั๋วงาน” โดยในขณะที่ผู้บริหารฝ่ายคาดการณ์ระยะเวลาดำเนินการของทั้งสามกระบวนการ คือ ไม่เกิน 1 วัน ไม่เกิน 3 วัน และไม่เกินระยะที่ KPI กำหนด ตามลำดับ แต่ในความเป็นจริงคือตั๋วงานปริมาณกว่า 52% ต้องรอการจำแนกเป็นเวลานานกว่า 1 วัน ส่งผลให้เกิดความล่าช้าสะสมในกระบวนการถัด ๆ ไป ทำให้ผู้ใช้บริการต้องรอนานขึ้น และในบางครั้งก็ทำให้บริษัทต้องจ่ายค่าโอทีสำหรับฝ่ายสนับสนุนไอทีเพิ่มเติมอีกด้วย จากปัญหาทั้งหมดที่กล่าวไปนี้ ผู้วิจัยจึงต้องการนำเทคนิคการเรียนรู้ของเครื่องจักร (machine learning) มาประยุกต์ใช้เพื่อวิเคราะห์ข้อความในตั๋วงานจากผู้ใช้บริการโดยอัตโนมัติ จุดประสงค์หลักคือการเพิ่มประสิทธิภาพ (ลดเวลา) ในการจำแนกประเภทตั๋วงาน ลดค่าใช้จ่ายบริษัท และลดความลำเอียงในการจำแนกตั๋วงานโดยมนุษย์ เปลี่ยนมาให้ระบบอัตโนมัติตัวเดียวช่วยวิเคราะห์ตั๋วงานทุกตั๋วให้ด้วยเกณฑ์การพิจารณาที่คงที่สม่ำเสมอแทน



รูปที่ 1 ผลคะแนนประเมินความพึงพอใจของผู้ใช้บริการเฉลี่ยในช่วงปี พ.ศ. 2557 ถึง พ.ศ. 2560

2. การทบทวนงานวิจัยที่เกี่ยวข้อง

งานวิจัยของ Altintas และ Tantug [1] เสนอตัวแบบสำหรับระบุปัญหาของตั๋วงานภาษาตุรกีอัตโนมัติจากอีเมลข้อความที่ได้รับจากผู้ใช้บริการ โดยแบ่งข้อมูลออกเป็นทั้งหมด 4 หมวด (categories) ได้แก่ หมวดเทคโนโลยีสารสนเทศ หมวดกิจการนักศึกษา หมวดสุขภาพ/วัฒนธรรม/กีฬา และ หมวดสำนักทุนการศึกษาและหอพัก โดยมีจำนวนตั๋วงานในแต่ละหมวดคือ 3380, 6586, 130 และ 113 ตั๋วงานตามลำดับ รวมทั้งหมด 10,209 ตั๋วงาน งานวิจัยนี้ประยุกต์การทำ Feature Extraction บนข้อความในอีเมลโดยใช้ TF-IDF (Term Frequency-Inverse Document Frequency) ร่วมกับการทดลองสร้าง Classification model แบบ Supervised Learning ทั้งหมด 4 ตัว ได้แก่ Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbors (KNN) และ Decision Tree และเพิ่มการทำ Term weighting ถ่วงน้ำหนักที่แตกต่างกันเพื่อประเมินความแตกต่างของประสิทธิภาพ ผลการทดลองระบุว่า SVM ให้ผลลัพธ์ความแม่นยำดีที่สุดสำหรับการจำแนกข้อมูลตั๋วงานที่มีจำนวนมาก ได้แก่ การจำแนกหมวดหมู่จากข้อมูลอีเมลทั้งหมด และการจำแนกประเภทตั๋วงานของหมวดเทคโนโลยีสารสนเทศ และหมวดกิจการนักศึกษา ในขณะที่ Naïve Bayes จะมีประสิทธิภาพดีเมื่อข้อมูลฝึกอบรม (training set) มีจำนวนน้อย ได้แก่ ข้อมูลของฝ่ายทุนการศึกษาและหอพัก ส่วน Decision Tree นั้นให้ผลลัพธ์ความแม่นยำค่อนข้างดีสำหรับข้อมูลของหมวดสุขภาพ/วัฒนธรรม/กีฬา เมื่อเปรียบเทียบกับการจำแนกของตัวแบบอื่นๆ จุดเด่นของงานวิจัยนี้ คือมีการพัฒนาตัวแบบให้เหมาะสมกับชุดข้อมูลแต่ละจำนวน ส่วนจุดด้อยคือมีจำนวนหมวดหมู่สำหรับการทำนายค่อนข้างมาก ทำให้โอกาสในการทำนายพลาดมีสูงกว่า

Dedik [2] เสนอตัวแบบสำหรับจัดหมวดหมู่ข้อความแบบอัตโนมัติของเว็บแอปพลิเคชันในบริษัทซอฟต์แวร์แห่งหนึ่งในสาธารณรัฐเช็ก โดยงานวิจัยนี้ใช้ข้อมูลฝึกอบรมทั้งภาษาอังกฤษและภาษาเช็กที่ถูกรวบรวมจากแหล่งข้อมูลทั้งหมด 3 แหล่ง คือจากบริษัทซอฟต์แวร์ที่กล่าวถึงในข้างต้นและ จาก Open-source 2 แหล่ง (Firefox และ Netbeans) รวมเป็นจำนวนมากกว่า 10,000 ตั๋วงาน ในส่วนของการจำแนกประเภทใช้ตัวแบบ SVM ซึ่งเป็น Supervised Learning ร่วมกับการสกัดคุณลักษณะด้วยวิธีการทำ TF-IDF ซึ่งได้ผลลัพธ์ความแม่นยำอยู่ที่ 57%, 53%, และ 53% สำหรับชุดข้อมูล Firefox, Netbeans, และของบริษัท ตามลำดับ จุดเด่นของงานวิจัยนี้คือสามารถนำไปใช้ได้กับหลาย ๆ โครงการโดยไม่ต้องปรับแต่งพารามิเตอร์มาก สามารถ generalize ให้เข้ากับกับโครงการอื่น ๆ ได้ดี

Vedala [3] เสนอกลไกการจำแนกประเภทตั๋วงานจากระบบ ITSM ของบริษัทที่มงานดูแลลูกค้าแห่งหนึ่ง เพื่อลดเวลาที่ใช้ในกระบวนการนี้ซึ่งปกติจะใช้เวลาอย่างน้อย 2-3 วันสำหรับตั๋วงานมากกว่า 100 งาน งานวิจัยนี้ใช้ระยะเวลา 2 ปี (ค.ศ. 2016-2017) ในการเก็บรวบรวมข้อมูลตั๋วงานทั้งหมดจำนวน 381,528 รายการ (แต่ละรายการมีข้อมูล 280 คอลัมน์) ซึ่งจะมีทั้งภาษาอังกฤษ ฟินแลนด์ เยอรมัน สวีเดน รวมถึงภาษาอื่น ๆ อีกมากมาย แล้วจึงนำข้อมูลมาทดลองสกัดคุณลักษณะด้วย TF-IDF Vectorizer ร่วมกับตัวแบบของ Logistic regression, Random forests และ Extremely randomized trees ผลการวัดประสิทธิภาพด้วยค่า Precision ค่า Recall และค่า F1-score พบว่า Random forests ให้ค่า F1-score เฉลี่ยสูงสุดในกลุ่ม โดยมีค่าถึง 90% จากการตรวจสอบความถูกต้องแบบ Cross-validation ทั้งนี้งานวิจัยชิ้นนี้ยังนำตัวแบบที่ได้ไปทดลองใช้งานจริงอีกเป็นเวลาสองสัปดาห์ ผลปรากฏว่าระยะเวลาที่ใช้ในการจำแนกตั๋วงานลดลงถึง 50% จากระยะเวลาหลักนาทีกลายเป็นหลักวินาที ช่วยเร่งให้กระบวนการทั้งหมดรวดเร็วขึ้น อีกทั้งสามารถสร้างคำแนะนำเพิ่มเติมอัตโนมัติในการแก้ปัญหาสำหรับกรณีที่เกิดขึ้นบ่อยที่สุดได้อีกด้วย จุดเด่นของงานวิจัยนี้คือมีจำนวนข้อมูลสำหรับใช้เรียนรู้ค่อนข้างมาก ทำให้ได้ผลลัพธ์ความแม่นยำสูง

Mandal และคณะ [4] เสนอระบบการมอบหมายตั๋วงานทางอีเมลอัตโนมัติแบบ end-to-end เพื่อให้สามารถส่งต่อตั๋วงานหนึ่ง ๆ ไปยังทีมงานที่รับผิดชอบได้อย่างเหมาะสมและทันท่วงที งานวิจัยนี้ทำการทดลองกับชุดข้อมูลจริงของลูกค้าทั้งหมดสามชุดรวมมากกว่า 700,000 อีเมล ใช้การสกัดคุณลักษณะแบบ TF-IDF สำหรับการเรียนรู้ด้วยเครื่องจักรดั้งเดิม (traditional machine learning) ได้แก่ LinearSVM, KNN, Logistic Regression, Multiple-Naïve Bayes, Random Forest, Adaboost และ Gradient Boosting และใช้หลักการประมวลผลภาษาธรรมชาติสองวิธี คือ Word Embedding (WE) และ Pre-trained Word Vectors (G - GloVe) นอกจากนี้ยังมีการทดลองที่ใช้ตัวแบบการเรียนรู้เชิงลึก (deep learning) ได้แก่ MLP (multilayer perceptron), CNN (convolution neural network), LSTM (long short term memory)

และ CNN-LTSM ด้วย ทั้งนี้ผู้วิจัยมีการใช้เทคนิค Ensemble Classifier เพื่อช่วยรวมผลลัพธ์จากแต่ละคู่ของตัวแบบเข้าด้วยกัน ผลลัพธ์ที่ได้คือ LinearSVM + MLP ให้ความแม่นยำสูงสุดถึง 89.61% ซึ่งมากกว่า LinearSVM + LSTM-G ที่ให้ความแม่นยำอยู่ที่ 88.38% เล็กน้อย ในส่วนของการนำตัวแบบที่ได้ไปปรับใช้ในงาน Production ของบริษัทผู้ให้บริการรายใหญ่ทั้งหมดสามรายที่มีปริมาณอีเมลแจ้งงานเข้ามาจำนวนมากกว่า 40,000 ฉบับต่อเดือน พบว่าสามารถให้ความแม่นยำเกือบ 90% และครอบคลุมอย่างน้อย 90% ของตั๋วงานทั้งหมด ช่วยให้บริษัทประหยัดเวลาในการทำงานสุทธิไปได้สูงถึง 23,000 ชั่วโมงต่อปี จุดเด่นของงานวิจัยนี้คือการพัฒนาการเรียนรู้เชิงลึกให้เกิดประสิทธิภาพสูงสุดสำหรับข้อมูลที่บริษัทมีอยู่ แต่จุดด้อยคือตัวแบบชนิดนี้เหมาะกับข้อมูลที่มีปริมาณมาก และประสิทธิภาพจะลดลงหากปริมาณข้อมูลที่มีไม่มากพอ

Al-Hawari และ Barham [5] ได้พัฒนาตัวแบบการเรียนรู้ของเครื่องจักรเพื่อช่วยลดเวลาในการจำแนกตั๋วงาน เพิ่มความแม่นยำ รวดเร็ว และสร้างความพึงพอใจให้กับผู้ใช้บริการ โดยอาศัยข้อมูลตั๋วงานในฐานข้อมูลของมหาวิทยาลัย German Jordanian จำนวน 1,254 รายการและใช้ตัวแบบ SVM ซึ่งเป็น Supervised Learning ร่วมกับวิธีการจัดการกับข้อมูลที่เป็นข้อความด้วยการทำ TF-IDF feature vectorization ผลลัพธ์ที่ได้มีความแม่นยำในการทำนายเพิ่มขึ้นจากเดิม 53.8% กลายเป็น 81.4% และยังมีจุดเด่นในการนำไปต่อยอดเพื่อปรับปรุงพัฒนาตัวชี้วัดที่จำเป็นต่อการประเมินประสิทธิภาพในการทำงานของทีมงานฝ่ายเทคโนโลยีสารสนเทศอีกด้วย

งานวิจัยของ Paramesh และ Shreedhara [6] พัฒนาการเรียนรู้ด้วยเครื่องจักรสำหรับจำแนกหมวดหมู่ของตั๋วงานอัตโนมัติจากข้อมูลตั๋วงานขององค์กรที่มีชื่อเสียงแห่งหนึ่งจำนวน 10,742 ตั๋วงาน ซึ่งการจำแนกหมวดหมู่นี้ประกอบไปด้วยหมวดหมู่ที่แตกต่างกันทั้งหมด 18 หมวดหมู่ และมีปัญหาที่บางหมวดหมู่มีจำนวนตั๋วงานไม่สมดุลกับหมวดหมู่อื่น (imbalanced dataset) ซึ่งต้องจัดการด้วยวิธี Under sampling ก่อนจะนำชุดข้อมูลที่สมดุลกันระหว่างทุกหมวดหมู่แล้วไปสกัดคุณลักษณะแบบ TF-IDF จากนั้นจึงนำไปทดลอง 2 แนวทาง ได้แก่ แบบที่ใช้ตัวแบบ SVM และ Naïve Bayes โดด ๆ และแบบ Ensemble of Classifiers ซึ่งประกอบด้วย Bagging-NB, Bagging-SVM และ AdaBoost-SVM ผลการทดลองที่ได้พบว่าแบบ Ensemble of Classifiers ให้ประสิทธิภาพที่สูงกว่า โดยเฉพาะ Bagging-SVM ที่ให้ผลลัพธ์ความแม่นยำถึง 87.78% มากที่สุดในกลุ่มเทคนิค Ensemble ทั้งหมด รองลงมาคือ Bagged-NB และ Adaboost-SVM ที่มีความแม่นยำ 72.61% และ 86% ตามลำดับ ส่วน Naïve Bayes และ SVM นั้นมีความแม่นยำเพียง 71% และ 85% จุดเด่นของงานวิจัยนี้คือการพัฒนาตัวแบบชนิด Ensemble ทำให้ได้ประสิทธิภาพที่สูงกว่าอย่างเห็นได้ชัด ต่างจากงานวิจัยอื่นที่ยังคงใช้การเรียนรู้ด้วยเครื่องจักรแบบตัวแบบเดียว

จากการทบทวนงานวิจัยที่เกี่ยวข้องกับการจำแนกตั๋วงานทั้งหมดที่กล่าวไป ผู้วิจัยพบว่าเทคนิคการสกัดคุณลักษณะจากข้อความที่นิยมคือ TF-IDF ในกรณีที่ใช้ตัวแบบการเรียนรู้ด้วยเครื่องจักรดั้งเดิม ส่วนในกรณีของตัวแบบที่ใช้การเรียนรู้เชิงลึกนั้น นิยมใช้ Word Embedding แทน จากแนวทางที่ได้นี้ผู้วิจัยจะนำไปออกแบบการทดลองในลำดับต่อไป โดยจะมีการทดลองใช้เทคนิคสกัดคุณลักษณะจากข้อความแบบต่าง ๆ การใช้ตัวแบบการเรียนรู้ของเครื่องหลาย ๆ ชนิด และการนำเอาเทคนิค Ensemble มาใช้เปรียบเทียบผลลัพธ์กันด้วย ซึ่งผู้วิจัยได้มีการวางแผนในการนำเทคนิคต่างๆ ข้างต้น มาสร้างการทดลองในหลากหลายมิติ เพื่อให้เกิดการค้นคว้า และพัฒนาตัวแบบให้เหมาะสมกับสถานการณ์และข้อมูลที่มีอยู่มากที่สุด

3. วิธีดำเนินการวิจัย

แหล่งที่มาของข้อมูลเป็นตั๋วงานที่ถูกแจ้งเข้ามาผ่านระบบจัดการบริการด้านไอทีของฝ่ายเทคโนโลยีสารสนเทศในบริษัทเอกชนแห่งหนึ่งในกรุงเทพมหานคร ตั้งแต่ปี พ.ศ. 2558 ถึง พ.ศ. 2561 จำนวน 10,608 รายการ โดยตัวแปรต้นและตัวแปรตามของงานวิจัยนี้ ได้แก่ขั้นตอนวิธีที่ใช้ในการทดลอง และความแม่นยำในการพยากรณ์ประเภทของตั๋วงาน ตามลำดับ ส่วนตัวแปรควบคุมนั้น ได้แก่ ข้อมูลที่ใช้ฝึกสอนและทดสอบตัวแบบ รวมถึงการตั้งค่าตัวแปรต่าง ๆ ของตัวแบบ

3.1 กระบวนการเก็บรวบรวมข้อมูล

ในงานวิจัยนี้ ผู้วิจัยได้รับอนุญาตให้ใช้และเก็บรวบรวมข้อมูลจากฐานข้อมูลระบบจัดการบริการด้านไอทีของฝ่ายเทคโนโลยีสารสนเทศในบริษัทเอกชนแห่งหนึ่งในกรุงเทพมหานคร รวมเป็นจำนวนทั้งสิ้น 10,608 ตั๋วงาน ประกอบด้วยจำนวนคุณลักษณะ (attribute) ทั้งหมด 12 คุณลักษณะ ได้แก่ รหัสตั๋วงาน วันที่แจ้งงาน เวลาที่แจ้งงาน ผู้ที่แจ้งงาน แผนกของผู้ที่แจ้งงาน ราย

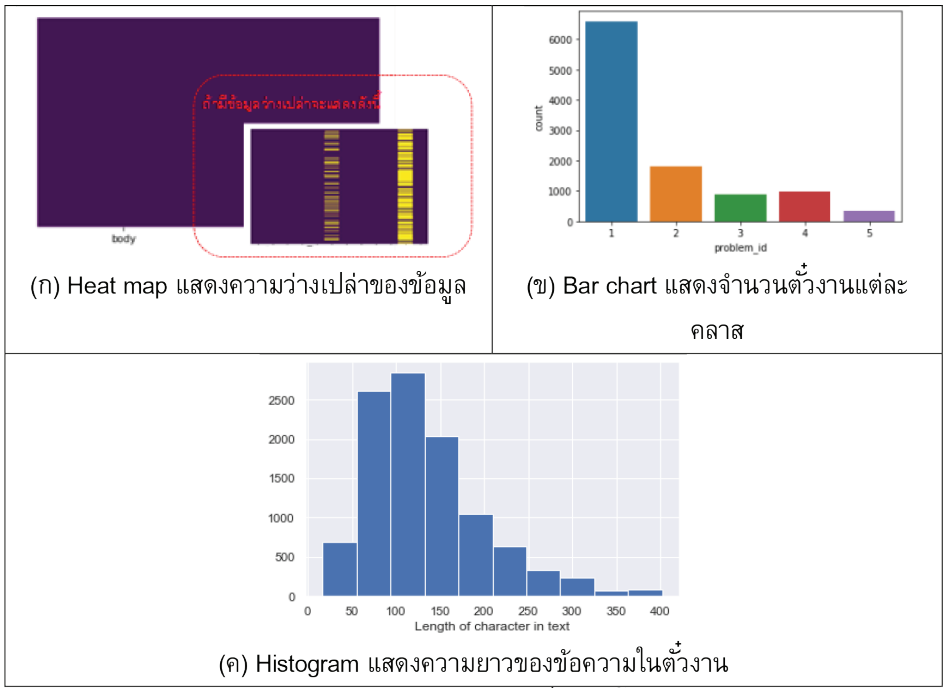
ละเอียดของปัญหา ผู้ที่รับงาน ผู้ที่ดำเนินการ รายละเอียดการดำเนินการ วันเวลาที่ดำเนินการแล้วเสร็จ วัตถุประสงค์ของงาน และประเภทของงาน (ตัวแปรเป้าหมาย) โดยทั้งหมดเป็นข้อมูลที่เก็บรวบรวมมาตั้งแต่ปี พ.ศ. 2558 ถึง พ.ศ. 2561 ในเนื้อหาของข้อมูลจะประกอบไปด้วยข้อมูลที่มีลักษณะเป็นทั้ง วันที่ เวลา ตัวเลข หมวดหมู่ และข้อความ ส่วนประเภทของงานซึ่งเป็นตัวแปรเป้าหมายที่ต้องการนำมาจำแนก (problem_id) นั้นมีอยู่ทั้งหมด 5 คลาสด้วยกัน ได้แก่ หมายเลข 1 คือ Hardware, หมายเลข 2 คือ Software, หมายเลข 3 คือ Network, หมายเลข 4 คือ Internet และ หมายเลข 5 คือ E-mail รูปที่ 2 แสดงตัวอย่างของข้อมูลจริงที่ผู้วิจัยทำการเชื่อมต่อข้อความหลาย ๆ ข้อความในข้อมูลที่เกี่ยวข้องกันทั้งหมดเข้าด้วยกัน (text concatenation) เพื่อเตรียมข้อมูลให้พร้อมก่อนนำเข้าสู่กระบวนการประมวลผลภาษาธรรมชาติ (natural language processing, NLP) ต่อไป ทั้งนี้ข้อมูลที่ถูกนำมาเชื่อมต่อกัน ได้แก่ “รายละเอียดของปัญหา” และ “รายละเอียดการดำเนินการ” เท่านั้น เนื่องจากเป็นข้อมูลส่วนที่ผู้วิจัยทดลองพบว่ามีความสัมพันธ์กับเป้าหมายอย่างมีนัยสำคัญ

3.2 กระบวนการสำรวจข้อมูล

ก่อนจะนำข้อมูลไปแปลงให้อยู่ในรูปที่พร้อมต่อการนำไปเป็นอินพุตให้กับตัวแบบนั้น ผู้วิจัยทดลองสำรวจลักษณะของข้อมูลเบื้องต้นก่อนเพื่อให้เข้าใจลักษณะและพฤติกรรมของชุดข้อมูลนี้ ซึ่งจะมีผลต่อไปถึงพฤติกรรมของตัวแบบที่เป็นการเรียนรู้ของเครื่องจักรต่อไปด้วย รูปที่ 3 แสดงผลการสำรวจข้อมูลตั๋วงานในงานวิจัยนี้ซึ่งประกอบด้วย (ก) การสร้างกราฟประเภท Heat map chart เพื่อแสดงผลความว่างเปล่าของข้อมูล (null) เมื่อสีเหลืองหมายถึงข้อมูลว่างเปล่า ทั้งนี้เพื่อให้ผู้วิจัยสามารถจัดการและแก้ไขปัญหาข้อมูลว่างนั้นได้แต่เนิ่น ๆ (ข) การสร้างกราฟประเภท Bar chart แสดงจำนวนตั๋วงานที่เกิดขึ้นในแต่ละคลาสทั้ง 5 คลาส เพื่อสำรวจความแตกต่างของปริมาณข้อมูล หรือความไม่สมดุลของปริมาณข้อมูลในแต่ละคลาส และ (ค) การสร้างกราฟประเภท Histogram แสดงผลจำนวนตั๋วงานที่มีความยาวของข้อความต่าง ๆ กัน ทั้งนี้ผลจากการสำรวจข้อมูลผู้วิจัยพบปัญหาหลัก 2 ปัญหา คือ ปัญหาความไม่สมดุลของปริมาณข้อมูลในแต่ละคลาส (รูปที่ 3.ข) และปัญหาที่ข้อมูลบางรายเป็นข้อความที่สั้นมากเกินไปสำหรับนำมาใช้ในการวิเคราะห์ข้อมูล (รูปที่ 3.ค)

	body	problem_id
5	ไม่สามารถพิมพ์งานได้ restartเครื่อง แล้วทำการต...	1
6	ระบบ SAP ไม่สามารถใช้งานได้ (ทุกด่าน) Sector D...	2
7	ขอติดตั้งระบบงาน Argento จำนวน 4 เครื่อง ดำเนิ...	2
8	เครื่อง notebook มีอาการค้างบ้างครั้งเนื่องจาก...	1
9	setup pc เตรียมอบรม sap ห้องประชุมอาคารกู้ภัย ...	1
10	ติดตั้ง wireless cisco 3 ตัวที่เอาไปเคลมมาเนี่...	3
11	login notbook ไม่ได้ ปิดเครื่องเปิดใหม่ log in...	1

รูปที่ 2 ตัวอย่างข้อมูลในชุดข้อมูล เมื่อคอลัมน์ body คือข้อความที่เกี่ยวข้องกันซึ่งถูกนำมาเชื่อมกัน และคอลัมน์ problem_id คือคลาสเป้าหมายของการจำแนกตั๋วงานนั้น ๆ



(ก) Heat map แสดงความว่างเปล่าของข้อมูล

(ข) Bar chart แสดงจำนวนตั๋วงานแต่ละคลาส

(ค) Histogram แสดงความยาวของข้อความในตั๋วงาน

รูปที่ 3 ผลการสำรวจข้อมูลตั๋วงานทั้งหมด

3.3 กระบวนการเตรียมข้อมูล

3.3.1 การจัดการกับปริมาณข้อมูลที่มีความแตกต่างระหว่างคลาสมากเกินไป หรือเรียกอีกอย่างว่าปัญหา Imbalance data โดยจากการศึกษาพบว่าแนวทางในการแก้ปัญหานี้ ได้แก่ การเพิ่มปริมาณข้อมูล (Over-sampling) โดยสุ่มเพิ่มจำนวนข้อมูลในกลุ่มรองให้มากขึ้น และ การลดปริมาณข้อมูล (Under-sampling) โดยสุ่มเลือกข้อมูลจากกลุ่มหลักมาเพียงบางส่วน เพื่อให้เหลือจำนวนใกล้เคียงกับกลุ่มรอง ในงานวิจัยนี้เนื่องจากข้อมูลตั๋วงานที่ผู้วิจัยรวบรวมมาได้มีปริมาณไม่มากนัก จึงเลือกใช้วิธีแรก เนื่องจากวิธีหลังจะยิ่งทำให้ปริมาณข้อมูลน้อยลงและอาจไม่เพียงพอต่อการเรียนรู้ของตัวแบบ

3.3.2 การทำความสะอาดข้อมูล (data cleansing) ด้วยการกำจัดตัวเลขที่ไม่มีนัยสำคัญและอักขระพิเศษที่ไม่ใช่ตัวอักษรออกจากข้อมูล เช่น #|'|"|”|“|!|\+|*|\%|$|\?|/|_|\-|=|\.|>|<|\(|\)|฿|, เป็นต้น เนื่องจากอักขระเหล่านี้ไม่มีประโยชน์ต่อการวิเคราะห์ข้อมูลและอาจทำให้การจำแนกประเภทของข้อมูลผิดพลาด

3.3.3 การแยกภาษาข้อความ (language separating) เพื่อแยกข้อความระหว่างภาษาไทยและภาษาอังกฤษออกจากกัน เนื่องจากแต่ละภาษามีกระบวนการเตรียมข้อมูลบางส่วนที่แตกต่างกัน

3.3.4 การสกัดคำ (tokenization) หรือการนำข้อความที่เป็นประโยคมาแตกออกเป็นคำแต่ละคำ แล้วเก็บคำที่ได้เอาไว้ในรายการ ในส่วนนี้ผู้วิจัยเลือกใช้เครื่องมือสำหรับสกัดคำภาษาอังกฤษคือ nltk.tokenize และในส่วนของข้อความภาษาไทยจะใช้ pythainlp.tokenize

3.3.5 การตรวจสอบการสะกด (spell checking) เพื่อตรวจคำผิดในหลายระดับ เช่น พิมพ์ผิดเป็นคำที่ไม่มีอยู่ในพจนานุกรม พิมพ์ผิดคำ คำพ้องเสียง คำสแลง หรือ ตั้งใจพิมพ์ผิด เป็นต้น โดยผู้วิจัยจะทำการตรวจสอบว่าคำที่อยู่ในข้อความมีปรากฏอยู่ในพจนานุกรมหรือไม่ และอาจแนะนำคำใกล้เคียงที่น่าจะเป็นคำที่ถูกต้องให้โดยอัตโนมัติ ซึ่งหลักการของการตรวจคำผิดนี้จะใช้ขั้นตอนวิธีของ Peter Norvig [7] ในการหารายการคำใกล้เคียงจากพจนานุกรม จากนั้นจะแสดงรายการคำทุกคำที่เป็นไปได้ในพจนานุกรมเรียงตามความน่าจะเป็น แล้วเลือกคำที่มีความน่าจะเป็นสูงสุดจากรายการของคำทั้งหมด เช่น ถ้าป้อนข้อมูลเป็นคำว่า โน๊ตบุ๊ค โน้ตบุค โน็ตบุ๊ก ซึ่งเป็นคำ

ที่สะกดผิด ระบบจะแสดงผลลัพธ์ออกเป็นคำว่า โน้ตบุ้ก ซึ่งเป็นคำที่สะกดถูก เป็นต้น สำหรับในภาษาอังกฤษจะใช้คลังคำศัพท์จาก NLTK Corpus ซึ่งมีคำศัพท์ในคลังกว่า 2 แสนคำ และใช้ตัวอักษร พยัญชนะและสระของภาษาอังกฤษในการนำมาสุ่มหาคำใกล้เคียง สำหรับภาษาไทยนั้นผู้วิจัยใช้คลังคำศัพท์จาก Thai National Corpus (TNC) ซึ่งมีคำศัพท์ในคลังกว่า 4 หมื่นคำ และใช้ตัวอักษร พยัญชนะ สระ และวรรณยุกต์ของภาษาไทยในการนำมาสุ่มหาคำใกล้เคียง

3.3.6 การลดรูปคำศัพท์ โดยใช้วิธีที่เป็นที่นิยม 2 วิธีคือ วิธี Lemmatization ซึ่งเป็นกระบวนตัดส่วนท้ายของคำแบบหยาบ ๆ ด้วยฮิวริสติก (heuristic) ทำให้ลดรูปของคำลงเหลือแต่ส่วนหน้าของคำที่เหมือนกันในคำกลุ่มเดียวกัน และวิธี Stemming ซึ่งเป็นกระบวนการในการแปลงคำด้วยรายการรากศัพท์ในพจนานุกรม และการวิเคราะห์หลักไวยากรณ์ของภาษาอย่างเหมาะสมในการแปรคำหรือผันคำ เพื่อกำจัดการแปรผันของคำให้เหลือแต่รูปฟอร์มพื้นฐาน ซึ่งทั้ง 2 วิธีนี้จะใช้กับภาษาอังกฤษได้ดีพอสมควร

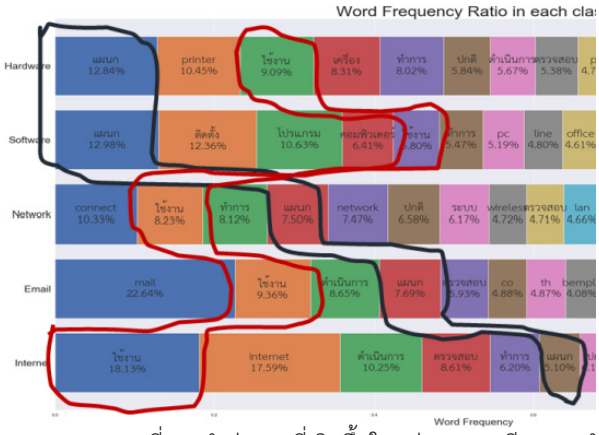
3.3.7 การกำจัดคำทั่วไป (stop words removing) เป็นการกำจัดกลุ่มคำทั่วไปที่มักพบบ่อย ๆ ในประโยคหรือเอกสารแต่ไม่ค่อยช่วยในการสื่อความหมาย อีกทั้งส่งผลให้ความซับซ้อนของโปรแกรมเพิ่มขึ้นในภายหลัง สำหรับคลังคำศัพท์ทั่วไปที่เป็นคำที่ไม่สื่อความหมายในภาษาอังกฤษผู้วิจัยเลือกใช้ nltk.stopwords.words('english') ซึ่งมีคำศัพท์ในคลังกว่า 2 ร้อยคำ ส่วนภาษาไทยจะใช้ pythainlp.corpus.common.thai_stopwords() ซึ่งมีคำศัพท์ในคลังกว่า 1 พันคำ

3.3.8 การผสานรวมรายการของคำ (merge word list) เมื่อเตรียมข้อมูลแล้วเสร็จ ผู้วิจัยจึงนำรายการของคำทั้งหมดทั้งของภาษาไทยและภาษาอังกฤษ กลับมารวมกันอีกครั้งเพื่อสร้างเป็น Bag of words ที่จะนำไปใช้งานต่อไป

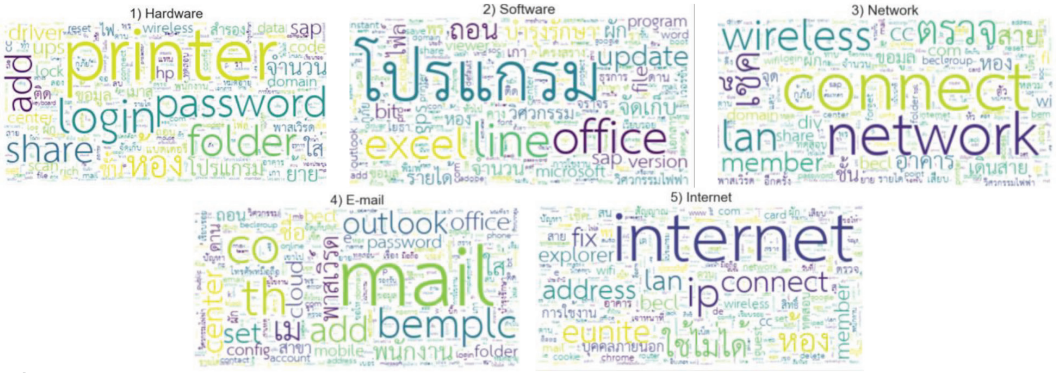
3.3.9 การปรับรวมคำที่พ้องความหมาย (synonyms normalization) เป็นการปรับให้คำที่มีความหมายในลักษณะเดียวกันแต่อาจเขียนอยู่ในรูปแบบอื่น ๆ ที่ต่างกันให้กลายเป็นคำ ๆ เดียวกันหรือกลุ่มเดียวกัน เช่น ปรับคำว่า mail เมล์ อีเมล์ เป็นคำว่า mail หรือปรับคำว่า connect คอนเน็ค เชื่อมต่อ เป็นคำว่า connect เป็นต้น โดยผู้วิจัยใช้ไลบรารีชื่อ synset ของWordNet ร่วมกับการใช้ประสบการณ์ด้านศัพท์เทคนิคของผู้วิจัยเองในการพิจารณารวบคำ

3.3.10 พิจารณา Word Frequency ร่วมกับ TF-IDF score เพื่อประกอบการตัดสินใจในการตัดคำที่ไม่จำเป็นออก โดยหากพบว่าคำใดมีปรากฏอยู่ในทุกคลาสด้วยความถี่สูง ร่วมกับมีค่า TF-IDF score ต่ำอย่างเห็นได้ชัด (ดังตัวอย่างของคำว่า ‘แผนก’ และ ‘ใช้งาน’ ในรูปที่ 4) แสดงว่าคำ ๆ นั้นเป็นคำทั่วไปที่ไม่มีความสำคัญหรืออัตลักษณ์ในการระบุเจาะจงถึงคลาสใดคลาสหนึ่ง ควรที่จะกำจัดออกจากข้อมูล ทั้งนี้รูปที่ 5 แสดงกราฟก้อนเมฆของคำ (Word cloud) เพื่อช่วยให้มองเห็นคำที่มีความสำคัญในแต่ละคลาสได้ชัดเจนยิ่งขึ้น

3.3.11 การสกัดคุณลักษณะ เป็นกระบวนการแปลงข้อมูลให้อยู่ในรูปแบบที่สามารถนำไปใช้งานได้ในตัวแบบการเรียนรู้ของเครื่อง เช่น การแปลงข้อความหรือรูปภาพให้กลายเป็นชุดของตัวเลขชุดใหม่ เป็นต้น ในงานวิจัยนี้ผู้วิจัยได้ทำการศึกษาและทดลองวิธีการสกัดคุณลักษณะ 2 แบบ ได้แก่ (1) เวกเตอร์ของการนับคำ (count vectorization) ซึ่งเป็นวิธีพื้นฐานในการแปลงคำให้กลายเป็นตัวเลข อาศัยเพียงการนับจำนวนคำที่ปรากฏอยู่บนข้อความเท่านั้น โดยวิธีนี้สามารถใส่จำนวนคำที่นับได้ลงไปในเวกเตอร์ตรง ๆ หรืออาจใส่เป็นค่าจำนวนคำที่ถูกถ่วงน้ำหนักจากความถี่ในการปรากฏก็ได้ (2) เทคนิคการคัดแยกคำตามความสำคัญ (term frequency-inverse document frequency หรือ TF-IDF) เป็นการนำ term frequency หรือจำนวนครั้งที่คำแต่ละคำปรากฏในแต่ละข้อความมาหารด้วยจำนวนคำทั้งหมดในข้อความนั้น แล้วจึงนำมาคูณกับ inverse document frequency หรือจำนวนข้อความทั้งหมดหารด้วยจำนวนข้อความที่มีแต่ละคำปรากฏอยู่ แล้วจึงใส่ฟังก์ชัน logarithm เข้าไป โดยสำหรับส่วนของ term frequency นั้นจะมีค่ามากถ้าคำไหนปรากฏอยู่ในข้อความบ่อย ๆ แต่ถ้าคำ ๆ นั้นไปปรากฏอยู่ในเกือบทุก ๆ ข้อความก็จะมีผลทำให้ค่า inverse document frequency ต่ำลงไป สามารถถ่วงดุลกับค่า term frequency ที่สูงและช่วยให้เราสามารถสกัดคำเด่น ๆ โดยหลีกเลี่ยงคำปรากฏบ่อยที่ไม่มีนัยสำคัญออกไปได้



รูปที่ 4 กราฟ Bar chart แสดงผลความถี่ของคำต่าง ๆ ที่เกิดขึ้นในแต่ละคลาสเรียงตามลำดับจากมากไปหาน้อย



รูปที่ 5 กราฟ Word cloud แสดงความสำคัญของคำในคลาส โดยความสำคัญของคำจะแปรผันตรงกับขนาดของคำดังกล่าวที่ปรากฏในภาพ

3.4 กระบวนการสร้างตัวแบบ

3.4.1 การลดจำนวนมิติของข้อมูลด้วยการวิเคราะห์องค์ประกอบหลัก (principal component analysis, PCA) เพื่อลดปัญหาความกระจัดกระจาย (sparse) ของเมทริกซ์คุณลักษณะที่ได้มาภายหลังจากกระบวนการสกัดคุณลักษณะ โดยปกติแล้วจะมีหลักการในการเลือกจำนวนของคุณลักษณะที่ดีที่สุดซึ่งสามารถอธิบายความแปรปรวนได้อยู่ในระหว่างช่วง 95-99% เพื่อไม่ให้จำนวนคุณลักษณะมีมากหรือน้อยจนเกินไป สำหรับงานวิจัยนี้ผู้วิจัยเลือกลดมิติของข้อมูลด้วยวิธีการวิเคราะห์องค์ประกอบหลัก และเลือกจำนวนของคุณลักษณะที่ดีที่สุดที่สามารถอธิบายความแปรปรวนได้อยู่ที่ 95% และสามารถลดมิติของข้อมูลลงไปได้จากเดิมที่มีจำนวนทั้งหมด 3,673 มิติให้เหลือเพียง 893 มิติสำหรับการสกัดคุณลักษณะแบบ Count vectorization และเหลือเพียง 1,546 มิติสำหรับการสกัดคุณลักษณะแบบ TF-IDF

3.4.2 การจัดมาตราส่วนข้อมูลสำหรับเปลี่ยนค่าลบให้กลายเป็นบวก (data scaling) เนื่องจากขั้นตอนวิธีการเรียนรู้ด้วยเครื่องจักรบางตัวไม่สามารถรองรับข้อมูลที่มีมาตราส่วนหลากหลายได้เอง จึงจำเป็นที่ผู้วิจัยต้องช่วยทำการปรับช่วงของข้อมูลให้ก่อน โดยกระบวนการนี้เป็นการจัดการภายหลังจากการทำ PCA แล้วเกิดค่าติดลบ (negative value) ซึ่งตัวแบบ Multiple Naïve Bayes ไม่รองรับขึ้น จึงต้อง Rescale ข้อมูลให้กลับมาอยู่ในช่วงค่าบวก (positive value) เป็นต้น สำหรับงานวิจัยนี้ผู้วิจัยเลือกวิธีการจัดมาตราส่วนข้อมูลแบบ MinMaxScaler เพื่อปรับค่าของข้อมูลที่ติดลบให้กลายเป็นค่าที่อยู่ในช่วงระหว่าง 0.0 ถึง 1.0

3.4.3 การแบ่งข้อมูลสำหรับชุดที่ใช้สอนตัวแบบและชุดที่ใช้ทดสอบตัวแบบ (training/test set split) นั้น ผู้วิจัยใช้การสุ่มแบ่งข้อมูลแบบแบ่งชั้นภูมิ (stratified random sampling) โดยจะสุ่มข้อมูลจากแต่ละคลาสขึ้นมาตามสัดส่วนของจำนวนข้อมูลในคลาสนั้น ๆ ซึ่งสามารถกำหนดค่าสัดส่วนในการแบ่งข้อมูลชุดสอนและชุดทดสอบได้ตามความเหมาะสม สำหรับงานวิจัยนี้ผู้วิจัยเลือกแบ่งสัดส่วนของข้อมูลชุดสอนและชุดทดสอบเป็น 80:20

3.4.4 การสร้างตัวแบบ (model fitting) ในงานวิจัยนี้ผู้วิจัยทดลองตัวแบบการเรียนรู้ด้วยเครื่องจักรรวม 6 ตัว ได้แก่

1) Multinomial Naïve Bayes (MNB) เป็นตัวแบบที่เหมาะสำหรับการจำแนกประเภทมากกว่า 2 ประเภทขึ้นไปที่มีคุณลักษณะไม่ต่อเนื่อง และเป็นตัวแบบเฉพาะของ Naïve Bayes Classifier ที่ใช้การแจกแจงแบบพหุนามสำหรับแต่ละคุณลักษณะ ซึ่งตัวแบบ Naïve Bayes นี้เป็นตัวแบบที่อาศัยหลักการความน่าจะเป็น (probability) ของโอกาสที่จะเกิดเหตุการณ์ต่าง ๆ ตามทฤษฎีของเบย์ (Bayes's theorem) ซึ่งมีขั้นตอนวิธีไม่ซับซ้อน สามารถสร้างได้ง่าย และใช้เวลาน้อยในการทำงาน

2) Support Vector Machine (SVM) - เป็นตัวแบบที่ใช้ในการวิเคราะห์และจำแนกข้อมูล โดยอาศัยหลักการของการหาสัมประสิทธิ์ของสมการเพื่อสร้างเส้นแบ่งแยกกลุ่มข้อมูลที่ถูกป้อนเข้าสู่กระบวนการสอนให้ระบบเรียนรู้ ทั้งนี้ในการสร้างสมการเส้นตรงเพื่อแบ่งเขตข้อมูล 2 หรือมากกว่า 2 กลุ่มออกจากกันนั้น SVM จะพยายามสร้างเส้นแบ่งตรงกึ่งกลางระหว่างกลุ่มให้มีระยะห่างระหว่างขอบเขตของทั้งสองกลุ่มมากที่สุด นอกจากนี้เรายังสามารถนำ Kernel function มาช่วยใน SVM เพื่อแปลงข้อมูลจาก feature space หนึ่งไปยังอีก feature space หนึ่งได้

3) Logistic Regression (LR) - เป็นเทคนิคการวิเคราะห์สถิติเชิงคุณภาพที่มีวัตถุประสงค์เพื่อศึกษาความสัมพันธ์ว่าตัวแปรอิสระหรือตัวแปรทำนายใดบ้างที่สามารถอธิบายตัวแปรเกณฑ์ (ตัวแปรตาม) ได้ โดยLogistic Regression แบ่งออกเป็น 2 ประเภท คือ Binary Logistic Regression สำหรับตัวแปรตามที่มีเพียง 2 ค่า เช่น ตัวแปรที่มีค่าเป็น 0 กับ 1 และ Multinomial Logistic Regression สำหรับตัวแปรเชิงกลุ่มที่มีค่ามากกว่า 2 ค่าขึ้นไป สำหรับตัวแบบนี้คำนวณได้จากสมการของ Logistic Function

4) Random Forest (RF) - เป็นตัวแบบที่ถูกพัฒนาขึ้นจาก Decision Tree ต่างกันที่ Random Forest มีต้นไม้หลาย ๆ ต้นประกอบกัน ทำให้ประสิทธิภาพและความแม่นยำในการทำงานสูงขึ้น หลักการของ Random Forest คือทำการฝึกฝนตัวแบบที่เหมือนกันหลาย ๆ ครั้งบนข้อมูลชุดเดียวกัน โดยแต่ละครั้งของการฝึกฝนจะเลือกส่วนของข้อมูลที่ฝึกฝนไม่เหมือนกัน แล้วค่อยเอาการตัดสินใจของตัวแบบเหล่านั้นมาโหวตกันในตอนท้าย

5) Stacking Model with Bagging (SM) – เป็นตัวแบบที่มีการแบ่งตัวฝึกฝนเป็นหลาย ๆ กลุ่ม แล้วเอาข้อมูลทั้งหมดให้กลุ่มแรกเรียนรู้ จากนั้นนำคำตอบของกลุ่มแรกมารวมกันแล้วส่งต่อให้กลุ่มต่อมาเรียนรู้ต่อ ๆ กันไป ซึ่งผลในที่นี้จะส่งต่อไปให้กับ Bagging Classifier เพื่อทำการเลือกโหวต (majority vote) และดูว่าคำตอบส่วนมากเป็นอะไร หรือควรจะตอบอย่างไรที่เหมาะที่สุด เทคนิคนี้สามารถลดความแปรปรวนและลด overfit ได้

6) XG-Boosting (XGB) - เป็นวิธี Ensemble learning ประเภท Boosting ที่นำเอา Decision Tree มาฝึกฝนต่อกันหลาย ๆ ต้น โดยที่แต่ละต้นของ Decision Tree จะเรียนรู้ความผิดพลาด (error) จาก Weak classifier ของต้นไม้ตัวก่อนหน้า ทำให้ความแม่นยำในการทำนายเพิ่มมากขึ้นเรื่อย ๆ เมื่อมีการเรียนรู้ของต้นไม้ต่อเนื่องกันจนมีความลึกมากพอ ทั้งนี้ตัวแบบจะหยุดเรียนรู้เมื่อไม่เหลือความผิดพลาดจากต้นไม้ก่อนหน้าให้สามารถเรียนรู้ต่อได้แล้ว เทคนิคนี้สามารถช่วยลด Bias ได้

3.4.5 การปรับค่าพารามิเตอร์ (parameters tuning) เป็นการหาค่าที่ดีที่สุดของพารามิเตอร์ต่าง ๆ โดยผู้วิจัยเลือกใช้วิธีการทำ Grid search ด้วยโมดูล GridSearchCV ของ sklearn.model_selection (scikit-learn 0.23.2 บน Python 3.7.6) กำหนดค่าของแต่ละพารามิเตอร์ที่ต้องการจะทดสอบแล้วลองประมวลผลตัวแบบในทุก ๆ ชุดของค่าพารามิเตอร์ที่เป็นไปได้ดังต่อไปนี้

3.4.5.1 Mutinomial Naive Bayes ผู้วิจัยปรับพารามิเตอร์ fit_prior ช่วง [True, False], class_prior ช่วง [None, [.1,.9],[.2, .8]] และ alpha ช่วง numpy.linspace(0.1, 1.5, 1)

3.4.5.2 SVM ผู้วิจัยปรับพารามิเตอร์ C ช่วง [0.1,1, 10, 100], gamma ช่วง [1,0.1,0.01,0.001], shrinking ช่วง [True,False], kernel ช่วง ['rbf', 'poly', 'sigmoid'] และ decision_function_shape ช่วง ['ovo','ovr']

3.4.5.3 Logistic Regression ผู้วิจัยปรับพารามิเตอร์ C ช่วง numpy.logspace(-4, 4, 20) และ penalty ช่วง ['l1', 'l2']

3.4.5.4 Random Forest ผู้วิจัยปรับพารามิเตอร์ max_features ช่วง ['log2', 'sqrt','auto'] และ criterion ช่วง ['entropy', 'gini']

3.4.5.5 XGBoost ผู้วิจัยปรับพารามิเตอร์ min_child_weight ช่วง [1, 5, 10], gamma ช่วง [0.5, 1, 1.5, 2, 5], subsample ช่วง [0.6, 0.8, 1.0], colsample_bytree ช่วง [0.6, 0.8, 1.0] และ max_depth ช่วง [3, 4, 5]

3.4.6 การทำนายผล (model prediction) เป็นการนำตัวแบบที่เป็นผลลัพธ์จากการเรียนรู้บนข้อมูลชุดฝึกฝน มาใช้ในการทำนายผลลัพธ์ของตั๋วงานหนึ่ง ๆ

3.5 กระบวนการคัดเลือกคุณลักษณะที่สำคัญ

3.5.1 ประเมินประสิทธิภาพของตัวแบบ (evaluation) เมื่อเสร็จจากหัวข้อที่ 3.4 แล้ว ผู้วิจัยจะทำการประเมินประสิทธิภาพของตัวแบบแต่ละตัวโดยใช้ค่าจาก Confusion matrix มาคำนวณเป็นตัวชี้วัด ได้แก่ ค่า Accuracy (เปอร์เซ็นความแม่นยำของระบบทั้งหมด) ค่า Precision (เปอร์เซ็นความแม่นยำที่สนใจแค่ในส่วนที่ทำนาย) ค่า Recall (เปอร์เซ็นความแม่นยำที่สนใจแค่ในส่วนของความเป็นจริง) และค่า F1-Score (ค่าเฉลี่ยของ Precision และ Recall) สำหรับทุกคลาส

3.5.2 การคัดเลือกคุณลักษณะที่สำคัญ (feature importance) เป็นกลุ่มของเทคนิคที่ใช้ในการคำนวณหาคะแนนความสำคัญของคุณลักษณะ โดยหากคุณลักษณะใดมีความสำคัญต่อการพยากรณ์ผลลัพธ์มากก็จะได้คะแนนความสำคัญนี้มากไปด้วย ในกระบวนการนี้ ผู้วิจัยมีจุดประสงค์คือการนำตัวแบบที่ได้ (จากหัวข้อ 3.4) มาคัดกรองอีกครั้ง เพื่อให้แน่ใจว่าคุณลักษณะที่ตัวแบบใช้ในการทำนายผลลัพธ์คือคุณลักษณะที่มีความเด่นอย่างแท้จริง โดยในงานวิจัยนี้จะเป็นการใช้คำสั่ง “model.feature_importances_” ของตัวแบบ XG-Boosting (หรือตัวแบบอื่น ๆ ที่มีคุณสมบัติการคัดเลือกคุณลักษณะที่สำคัญให้ใช้งาน) ซึ่งมีรายละเอียดการพิจารณาคัดเลือกตามลำดับต่อไปนี้

1) เรียงลำดับคะแนนความสำคัญของแต่ละคุณลักษณะตั้งแต่น้อยที่สุดลงหามากที่สุด ขั้นนี้อ้างอิงจากเทคนิค Recursive Feature Elimination (RFE) ของ Guyon และคณะ [8]

2) นำทุกคะแนนความสำคัญที่ได้จากข้อ 1 มาสร้างเป็นเกณฑ์ (Threshold) เพื่อเลือกทุกคุณลักษณะที่มีคะแนนเกินเกณฑ์ให้ตัวแบบนำไปใช้ในการฝึกฝน ทำนาย และแสดงผลค่าความแม่นยำ ขั้นนี้อ้างอิงจากเทคนิค Recursive Feature Elimination (RFE) ของ Guyon และคณะ [8]

3) ทำซ้ำข้อ 2 จนกว่าจะครบทุกคุณลักษณะ ขั้นนี้อ้างอิงจากเทคนิค Recursive Feature Elimination (RFE) ของ Guyon และคณะ [8]

4) วิเคราะห์และตัดสินใจเลือกคุณลักษณะเพียงบางตัวออกมาใช้ โดยผู้วิจัยเลือกจากค่าความแม่นยำที่เพิ่มขึ้นหรือลดลงแค่เพียงเล็กน้อยในช่วงที่ยอมรับได้ เนื่องจากหากทดลองลดจำนวนคุณลักษณะลงเรื่อย ๆ แล้วพบว่าทำให้ค่าความแม่นยำลดลงมากอย่างเห็นได้ชัด เป็นการแสดงให้เห็นว่าคุณลักษณะนั้นมีความสำคัญสูงไม่ควรเอาออก

5) เมื่อเลือกค่าความแม่นยำที่ยอมรับได้แล้ว ให้ดูว่า ณ ค่าความแม่นยำที่เลือกนั้นเกิดจากการใช้คุณลักษณะที่มีคะแนนความสำคัญเท่าไรมาเป็นเกณฑ์ แล้วจึงค่อยคัดเลือกคุณลักษณะทั้งหมดที่มีคะแนนความสำคัญมากกว่าเกณฑ์ดังกล่าวนั้นเข้าไปใช้งานและฝึกฝนในตัวแบบอื่น ๆ ต่อไป

3.5.3 ภายหลังการคัดเลือกคุณลักษณะสำคัญ ผู้วิจัยทำการประเมินตัวแบบตามตัวชี้วัดในขั้น 3.5.1 ซ้ำอีกครั้ง เพื่อเปรียบเทียบให้แน่ใจว่าภายหลังกระบวนการคัดเลือกคุณลักษณะสำคัญ ตัวแบบที่ได้จะมีความเปลี่ยนแปลงไปในทิศทางที่(ผลลัพธ์)ดีขึ้นจริง

4. ผลการวิจัย

ตารางที่ 1 แสดงผลลัพธ์ความแม่นยำของตัวแบบที่สร้างด้วยการเรียนรู้ของเครื่องจักร 6 แบบและด้วยการสกัดคุณลักษณะ 2 แบบ จากผลการทดลองพบว่าการสกัดคุณลักษณะแบบ TF-IDF ได้ค่า F1-Score สูงกว่าการสกัดคุณลักษณะแบบ Count vectorization กล่าวคือการใช้ TF-IDF เพื่อคัดเลือกคำจากข้อความในชุดข้อมูลนั้นสามารถช่วยเพิ่มประสิทธิภาพในการทำนายชนิดของตั๋วงานได้มากกว่า Count vectorization นั่นเอง ทั้งนี้เหตุที่ผู้วิจัยเลือกพิจารณาค่า F1-Score เพราะค่านี้เป็นดัชนีวัดผลที่ดีมากกว่า Accuracy ในกรณีที่ต้องการเน้นวัดผลความแม่นยำสำหรับคลาสบวก (positive class) โดยเฉพาะ ทั้งนี้ผู้

วิจัยพบว่าการใช้หรือไม่ใช้เทคนิค PCA ให้ผลลัพธ์ที่ไม่ต่างกันมากนัก ยกเว้นตัวแบบ Multiple Naïve Bayes ที่ให้ผลลัพธ์ไม่แม่นยำเมื่อใช้เทคนิค PCA ดังนั้นผู้วิจัยจึงตัดสินใจไม่ใช้เทคนิค PCA สำหรับตัวแบบ Multiple Naïve Bayes เพียงตัวแบบเดียว อย่างไรก็ตามในงานนี้ผู้วิจัยไม่ได้เตรียมข้อมูลอินพุตให้มีลักษณะ center (zero-mean) และมีค่า variance ของทุก dimensions ใกล้เคียงกันแต่อย่างใด หากทำการทดลองเพิ่มเติมและเตรียมอินพุตให้อยู่ในลักษณะดังกล่าวอาจให้ผลลัพธ์จาก PCA ที่แตกต่างออกไปได้

นอกจากนี้ผลการทดลองในตารางที่ 1 ยังแสดงให้เห็นด้วยว่าการทำนายด้วยตัวแบบที่ใช้เทคนิค Ensemble ให้ผลลัพธ์ความแม่นยำสูงกว่าตัวแบบชนิดพื้นฐานที่มีโดด ๆ เพียงตัวเดียว โดยเฉพาะตัวแบบ XG-Boosting ที่ให้ผลลัพธ์ค่า F1-Score สูงถึง 90.88% รองลงมาคือตัวแบบ Stacking Model with Bagging ให้ผลลัพธ์ค่า F1-Score 87.62% แต่ในทางตรงกันข้าม ตัวแบบที่ใช้เทคนิค Ensemble ก็จำเป็นต้องใช้เวลาในการสอนและการทำนายค่อนข้างนานกว่า เนื่องจากมีความซับซ้อนในการคำนวณมากกว่า อย่างไรก็ตามผู้วิจัยเห็นว่าระยะเวลาที่ใช้ในการสอนและการทำนายตัวแบบยังคงอยู่ในขอบเขตเวลาที่ยอมรับได้และไม่มากเกินกว่าที่เป้าหมายคาดหวังไว้

ตารางที่ 1 ผลลัพธ์ค่าความแม่นยำของตัวแบบทั้งหมด 6 ตัวแบบบนการสกัดคุณลักษณะทั้ง 2 แบบ

	Count vectorization			
	Accuracy	**Precision**	**Recall**	**F1-score**
MNB	82.28%	75.48%	78.74%	76.74%
SVM	86.10%	84.22%	74.59%	77.99%
LR	84.68%	82.01%	75.09%	77.96%
RF	85.44%	82.07%	80.44%	81.04%
SM	88.12%	89.47%	82.64%	85.27%
XGB	90.72%	90.85%	86.82%	**88.67%**

	TF-IDF			
	Accuracy	**Precision**	**Recall**	**F1-score**
MNB	84.97%	84.09%	75.65%	79.01%
SVM	89.54%	87.53%	85.51%	86.34%
LR	86.24%	85.66%	79.93%	82.50%
RF	85.25%	81.83%	79.08%	80.36%
SM	89.87%	88.66%	86.39%	87.62%
XGB	92.55%	91.15%	89.14%	**90.88%**

ตารางที่ 2 ผลลัพธ์ค่าความแม่นยำของตัวแบบเมื่อนำไปทดลองใช้งานกับตั๋วงานชุดอื่น

	XG-Boosting with TF-IDF			
	Accuracy	**Precision**	**Recall**	**F1-score**
All	83.23%	79.43%	83.40%	**81.21%**
Hardware class		85.00%	82.93%	83.95%
Software class		84.62%	82.80%	83.70%
Network class		80.00%	80.00%	80.00%
E-mail class		79.43%	86.67%	82.54%
Internet class		68.75%	84.62%	75.86%

5. สรุปผลงานวิจัย

จากผลการทดลองทั้งหมดในข้างต้น สรุปผลได้ว่าผู้วิจัยได้ทำการคัดเลือกตัวแบบที่ให้ผลลัพธ์ความแม่นยำสูงสุด ซึ่งก็คือตัวแบบ XG-Boosting บนการสกัดคุณลักษณะแบบ TF-IDF มาประยุกต์ใช้กับตั๋วงานชุดอื่นจำนวนกว่า 300 ตั๋วงานนอกเหนือจากตั๋วงานชุดที่ใช้ทดลองกับตัวแบบ โดยผู้วิจัยนำแบบจำลองที่พัฒนาขึ้นมาทำนายประเภทของตั๋วงานแต่ละตั๋ว แล้วนำผลการทำนายส่งต่อให้กับทีมงานฝ่ายสนับสนุนเทคโนโลยีสารสนเทศ

(IT Support) เป็นผู้ตรวจสอบความถูกต้องของผลการทำนาย ผลการประเมินประสิทธิภาพของตัวแบบในสถานการณ์การทำงานจริงนี้ถูกแสดงอยู่ในตารางที่ 2

จากผลลัพธ์ค่าความแม่นยำในตารางที่ 2 จะเห็นว่าเมื่อนำตัวแบบ XG-Boosting บนการสกัดคุณลักษณะแบบ TF-IDF มาประยุกต์ใช้กับตั๋วงานชุดอื่นแล้ว ค่าความแม่นยำจะลดลงเล็กน้อย และ ค่า Precision ของคลาส Internet จะต่ำที่สุดเนื่องจากมีการทำนายคลาส Network ผิดมาเป็นคลาส Internet อยู่จำนวนหนึ่ง ซึ่งผู้วิจัยคาดว่าเกิดจากข้อมูลตั๋วงานมีคำสำคัญหลายคำที่ปรากฏซ้ำกันอยู่ในคลาส Network และคลาส Internet แต่แม้การประยุกต์ใช้งานจะให้ค่าความแม่นยำลดลงเล็กน้อย ในแง่ของประสิทธิภาพในการจำแนกตั๋วงานนั้นยังคงอยู่ในเกณฑ์ที่ยอมรับได้ อีกทั้งตัวแบบที่พัฒนาขึ้นนี้ยังมีประโยชน์ในเชิงประจักษ์อีก 2 ประการ ได้แก่

1) เพิ่มระดับความพึงพอใจของผู้ใช้บริการในประเด็นการตรงต่อเวลาและประเด็นความรวดเร็วในการให้บริการ จากเดิมที่คะแนนความพึงพอใจเฉลี่ยอยู่ที่ 4.47 และ 4.53 ตามลำดับ ก็เพิ่มขึ้นมาเป็น 4.90 และ 4.80 คิดเป็นความพึงพอใจที่เพิ่มขึ้น 8.17% และ 7.38% ตามลำดับ โดยระดับความพึงพอใจนี้เกิดจากการที่ผู้วิจัยนำตัวแบบของงานวิจัยนี้มาทดลองทำนายกับข้อมูลตั๋วงานชุด Validate set แล้วนำผลลัพธ์การทำนายนี้ไปสุ่มสอบถามจากผู้ใช้บริการจำนวน 30 คน พร้อมกับให้คะแนนความพึงพอใจในด้านความแม่นยำและระยะเวลาที่ใช้ของการทำนายครั้งนี้

2) ลดระยะเวลาที่ใช้ในกระบวนการจำแนกตั๋วงานลงจากเดิมที่ 52% ของตั๋วงานทั้งหมดใช้เวลามากกว่า 1 วันในการจำแนก เหลือเพียงใช้เวลาน้อยกว่า 1 วันสำหรับจำแนกตั๋วงานทั้งหมด โดยจากการจับเวลาที่ตัวแบบใช้ในการจำแนกตั๋วงานจำนวน 300 รายการ พบว่าใช้เวลา 6 นาที (5 นาที 52 วินาที) คิดเป็นเวลาประมวลผลเฉลี่ยตั๋วงานละ 1.2 วินาที

ทั้งนี้การนำไปประยุกต์ใช้จะเป็นลักษณะของการใช้ตัวแบบที่ได้จากการทดลองมาทำนายชุดข้อมูลจำนวนหนึ่งในช่วงระยะเวลาหนึ่งกับกลุ่มตัวอย่างตามที่กำหนด เพื่อทดสอบผลลัพธ์ตามสมมติฐานที่ตั้งไว้ มิได้มีการนำไปสร้างเว็บไซต์หรือซอฟต์แวร์แล้ว Deploy ขึ้น Production ของบริษัทอย่างเป็นทางการ

6. เอกสารอ้างอิง

1. Altintas, M. and Tantug, C., 2014, “Machine Learning Based Ticket Classification in Issue Tracking Systems,” *Proceeding of the International Conference on Artificial Intelligence and Computer Science (AICS 2014),* Bandung, INDONESIA, pp. 195- 207.

2. Dedik, B.V., 2015, Automatic Ticket Triage Using Supervised Text Classification, Master of Informatics Thesis, Faculty of Informatics, Masaryk University, 83 p.

3. Vedala, D., 2018, Building a Classification Engine for Ticket Routing in IT Support Systems, Master of Computer Science and Engineering Thesis, School of Science, Aalto University, 48 p.

4. Mandal, A., Malhotra, N., Agarwal, S., Ray, A. and Sridhara, G., 2018, “Cognitive System to Achieve Human-level Accuracy in Automated Assignment of Helpdesk Email Tickets,” *Service-Oriented Computing (ICSOC 2018)*, Bengaluru, India, pp. 332-341.

5. Al-hawari, F. and Barham, H., 2019, “A Machine Learning based Help Desk System for IT Service Management,” *Journal of King Saud University - Computer and Information Sciences,* 17 p. https://doi.org/10.1016/j.jksuci.2019.04.001

6. Paramesh, S.P. and Shreedhara, K.S., 2019, “IT Help Desk Incident Classification Using Classifier Ensembles,” *ICTACT Journal on Soft Computing - Department of Computer Science and Engineering, University B.D.T College of Engineering,* 9 (4), pp. 1980-1987. https://doi.org/10.21917/ijsc.2019.0276

7. Norvig, P., 2016, How to Write a Spelling Corrector [Online], Available http://norvig.com/spell-correct.html. [20 December 2020]

8. Guyon, I., Weston, J., Barnhill, S. and Vapnik, V., 2002, “Gene Selection for Cancer Classification using Support Vector Machines,” *Machine Learning,* 46, pp. 389-422.