ระบบจำแนกตั๋วงานอัจฉริยะด้วยเทคนิควิเคราะห์ข้อความไทย-อังกฤษ และการเรียนรู้ของเครื่อง

SMART IT SERVICE DESK TICKET CLASSIFICATION SYSTEM USING THAI-ENGLISH TEXT ANALYSIS AND MACHINE-LEARNING TECHNIQUES

เฉลิมชัย พิเดช

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2563

ระบบจำแนกตั๋วงานอัจฉริยะด้วยเทคนิควิเคราะห์ข้อความไทย-อังกฤษ และการเรียนรู้ของเครื่อง

เฉลิมชัย พิเดช

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2563

SMART IT SERVICE DESK TICKET CLASSIFICATION SYSTEM USING THAI-ENGLISH TEXT ANALYSIS AND MACHINE-LEARNING TECHNIQUES

CHALERMCHAI PIDEJ

A Master's Project Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Information Technology)

Faculty of Science, Srinakharinwirot University

2020

สารนิพนธ์

เรื่อง

ระบบจำแนกตั๋วงานอัจฉริยะด้วยเทคนิควิเคราะห์ข้อความไทย-อังกฤษ และการเรียนรู้ของเครื่อง

ของ

เฉลิมชัย พิเดช

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร.ศุภชัย ไทยเจริญ)

..... ประธาน

(อาจารย์ ดร.ธรรมศักดิ์ เธียรนิเวศน์)

..... กรรมการ

(อาจารย์ ดร.โสภณ มงคลลักษมี)

ชื่อเรื่อง	ระบบจำแนกตั๋วงานอัจฉริยะด้วยเทคนิควิเคราะห์ข้อความไทย-อังกฤษและการเรียนรู้ของเครื่อง
ผู้วิจัย	เฉลิมชัย พิเดช
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2563
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. ศุภชัย ไทยเจริญ

ตั๋วงาน คือ การเขียนคำร้องในระบบและส่งให้กับผู้สนับสนุนด้านไอทีช่วยแก้ไขปัญหา ข้อความในตั๋วงานจะมีทั้งประโยคที่เป็นภาษาไทย ภาษาอังกฤษ และภาษาไทยผสมภาษาอังกฤษ โดยบางครั้งผู้ใช้งานเขียนข้อความที่สั้นมากทำให้ไม่สามารถเข้าใจความหมายของข้อความและไม่สามารถทำการระบุประเภทของตั๋วงานได้อย่างถูกต้อง ในงานวิจัยนี้ได้ประยุกต์ใช้การประมวลผลภาษาธรรมชาติ วิทยาการข้อมูล หลักการเรียนรู้ของเครื่อง เพื่อช่วยในการแก้ปัญหาการแยกประเภทตั๋วงาน โดยการวิเคราะห์คำที่ไม่สำคัญและใช้เครื่องมือการตัดคำภาษาไทยในขั้นตอนการประมวลผลภาษาธรรมชาติ ที่สามารถให้ผลทดลองมีความถูกต้องสูงถึง 91 เปอร์เซ็นต์

คำสำคัญ : ตั๋วงาน, การจำแนกประเภทตั๋วงาน, วิทยาการข้อมูล, หลักการเรียนรู้ของเครื่อง, การประมวลผลภาษาธรรมชาติ

Title	SMART IT SERVICE DESK TICKET CLASSIFICATION SYSTEM USING THAI-ENGLISH TEXT ANALYSIS AND MACHINE-LEARNING TECHNIQUES
Author	CHALERMCHAI PIDEJ
Degree	MASTER OF SCIENCE
Academic Year	2020
Thesis Advisor	Assistant Professor Dr. Supphachai Thaicharoen

An IT ticket is a textual description of the IT-related complaints of the users or the problems that are submitted to IT helpdesk support personnel for resolutions. The text sentences in the ticket could be written in English-only, Thai-only, and a combination of the Thai and English languages. In addition, the description is occasionally too short to be able to specify the correct category of problems. This paper presents an application of natural language processing, data science and machine-learning methods to facilitate the classification of ticket problems on a real-world dataset. By eliminating important words and performing word tokenization in the pre-processing step, the performance of ticket classifications from the experiments is as high as 91% accuracy.

Keyword : Service desk ticket, Ticket classification, Data science, Machine learning, Natural Language Processing

กิตติกรรมประกาศ

สารนิพนธ์นี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์จาก ผศ.ดร.ศุภชัย ไทยเจริญ อาจารย์ที่ปรึกษา ที่ให้คำปรึกษาและคำแนะนำในการทำสารนิพนธ์และการสนับสนุนข้อมูลทางด้านวิชาการ

ขอกราบขอบพระคุณองค์กรที่ผู้ทำวิจัยทำงานอยู่ ที่ให้ความอนุเคราะห์และอนุญาตให้ทำการเก็บรวบรวมข้อมูลเพื่อนำมาใช้ในการวิจัย อีกทั้งขอขอบคุณเพื่อนร่วมงานที่สนับสนุนในหลาย ๆ ด้าน ทำให้งานสารนิพนธ์นี้เสร็จลุล่วงไปด้วยดี

ขอกราบขอบพระคุณคณะผู้จัดงาน NCITE2021 ที่ช่วยเหลือและให้คำปรึกษาในการนำเสนองานวิจัยนี้ได้เป็นอย่างดี

เฉลิมชัย พิเดช

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญรูปภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตการวิจัย 2

1.4 ประโยชน์ที่ได้รับจากงานวิจัย 3

บทที่ 2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง 4

2.1 ตั๋วงานและกระบวนการในการสร้างตั๋วงาน 4

2.1.1 ตั๋วงาน (Service Desk Ticket) 4

2.1.2 กระบวนการในการสร้างตั๋วงาน (Ticket Creation Process) 4

2.1.3 โปรแกรมตั๋วงาน 5

2.1.4 ปัญหาที่พบในระบบตั๋วงาน 7

2.2 เทคนิคการประมวลผลภาษาธรรมชาติ 7

2.2.1 การทำความสะอาดข้อความ (Text Cleansing) 7

2.2.2 การตัดคำในข้อความ (Tokenization) 8

2.2.3 การตัดส่วนท้ายของคำ (Stemming) 8

2.2.4 การแปลงคำให้เป็นรากศัพท์ (Lemmatization) 8
2.2.5 การหาคำสำคัญในข้อความ (TF-IDF) 8
2.3 อัลกอริทึมการทำนายและการวัดประสิทธิภาพ (Algorithm and Confusion Matrix) 9
2.3.1 อัลกอริทึมสุ่มป่าไม้แบบแยกประเภท (Random Forest Classification) 9
2.3.2 การประเมินสมรรถนะแบบจำลองการทำนาย (Confusion Matrix) 9
2.4 งานวิจัยที่กี่ยวข้อง 11
บทที่ 3 วิธีดำเนินการวิจัย 14
3.1 การทบทวนวรรณกรรมและงานวิจัย 14
3.2 การวิเคราะห์และศึกษาข้อความของตั๋วงานจากข้อมูลจริง 15
3.2.1 เริ่มต้นการวิเคราะห์และตรวจสอบชุดข้อมูลของตั๋วงาน 15
3.2.2 เลือกข้อมูลที่จะนำมาใช้ 17
3.2.3 จัดรูปแบบชุดข้อมูล 18
3.2.4 ลบคำเฉพาะ (Unique Term) 21
3.3 ประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ 22
3.3.1 ทำความสะอาดข้อความ (Text Cleansing) 22
3.3.2 การตัดข้อความออกเป็นคำ (Word Tokenization) 23
3.3.3 ลบคำที่ไม่สื่อความหมาย (Stopword Removal) 23
3.3.4 การหาคำสำคัญในข้อความ (TF-IDF) 27
3.3.4.1 การหาจำนวนคำในข้อความตั๋วงาน (Word Count) 27
3.3.4.2 การหาความถี่ของคำในเอกสาร (Term Frequency) 28
3.3.4.3 การหาความถี่ของเอกสารผกผัน (Inverse Data Frequency) 29
3.3.4.4 การหาคำสำคัญในข้อความ (TF-IDF) 30
3.4 สร้างแบบจำลองการแยกประเภทตั๋วงาน 31

3.4.1 แบ่งชุดข้อมูล (Train-Test Split) 31
3.4.2 การประเมินสมรรถนะแบบจำลอง (Confusion Matrix) 35
3.4.3 ผลการทดสอบการจำแนกตั๋วงาน 35
3.4.3.1 อธิบายตั๋วงานที่คลุมเคลือโดยแบบจำลองไม่สามารถแยกประเภทได้ 37
3.4.3.2 อธิบายตั๋วงานที่คลุมเคลือแต่แบบจำลองสามารถแยกประเภทได้ 38
บทที่ 4 ผลการดำเนินการวิจัย 39
4.1 การลบคำเฉพาะ (Unique Term) 39
4.2 การตัดคำด้วย PyThaiNLP 39
4.3 การทำความสะอาดข้อมูลด้วย Regular Expression 41
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ 45
5.1 สรุปผลการวิจัย 45
5.2 อภิปรายผล 47
5.2.1 การลบคำเฉพาะ 47
5.2.2 การตัดคำด้วย PyThaiNLP 47
5.2.3 การใช้ Regular Expression เพื่อกรองคำที่สมบูรณ์มากที่สุด 47
5.3. ข้อเสนอแนะ 48
บรรณานุกรม 49
ประวัติผู้เขียน 51

สารบัญตาราง

หน้า

ตาราง 1 การประเมินสมรรถนะแบบจำลองการทำนาย 10
ตาราง 2 การประเมินประสิทธิภาพของแบบจำลองในรูปแบบค่าต่างๆ 10
ตาราง 3 สูตรคำนวณในการประเมินสมรรถนะของโมเดลรูปแบบค่าต่างๆ 10
ตาราง 4 สูตรคำนวณการประเมินสมรรถนะของแบบจำลองที่มีมากกว่า 2 คลาสขึ้นไป 11
ตาราง 5 ความหมายของประเภทตั๋วงาน 21
ตาราง 6 อธิบายความหมายของ Regular Expression 23
ตาราง 7 แสดงผลการทดสอบการทำนายข้อความตั๋วงานที่คลุมเคลือ 37
ตาราง 8 อธิบาย RegEx ของแต่ละบรรทัด 43

สารบัญรูปภาพ

หน้า

ภาพประกอบ 1 กระบวนการการสร้างตั๋วงาน 5
ภาพประกอบ 2 หน้าต่างโปรแกรมการสร้างตั๋วงาน 5
ภาพประกอบ 3 ตั๋วงานที่ถูกสร้างรอการนำไปดำเนินการ 6
ภาพประกอบ 4 ตั๋วงานที่ถูกแก้ไขแล้วตามเวลาที่กำหนด 6
ภาพประกอบ 5 ตัวอย่างประโยคจากผู้ใช้งานที่คลุมเครือ 7
ภาพประกอบ 6 การประมวลผลภาษาธรรมชาติ 7
ภาพประกอบ 7 ตาราง Confusion Matrix 9
ภาพประกอบ 8 ภาพรวมวิธีดำเนินการวิจัย 14
ภาพประกอบ 9 นำเข้าชุดข้อมูลเพื่อทำการวิเคราะห์และศึกษา 16
ภาพประกอบ 10 เลือกดูจำนวนตั๋วงานที่มีภาษาไทยในข้อความ 16
ภาพประกอบ 11 ข้อความตั๋วงานมีความคลุมเครือ 17
ภาพประกอบ 12 เลือกข้อมูลที่จะนำมาใช้ 17
ภาพประกอบ 13 แบ่งคอลัมน์เป็น MasterCate และ DetailCate 18
ภาพประกอบ 14 ประเภทของตั๋วงานที่นอกขอบเขต 19
ภาพประกอบ 15 ประเภทตั๋วงานถูกยุบรวมเป็น OutOfScope 19
ภาพประกอบ 16 แปลงประเภทของตั๋วงานเป็นตัวเลข 20
ภาพประกอบ 17 ตั๋วงานที่คลุมเครือในแต่ละประเภทตั๋วงาน 20
ภาพประกอบ 18 ลบคำเฉพาะที่เป็นชื่อบุคคล และ ชื่อคอมพิวเตอร์ 22
ภาพประกอบ 19 ฟังก์ชั่นการประมวลผลภาษาธรรมชาติ 22
ภาพประกอบ 20 การกรองข้อความด้วย Regular Expression 23
ภาพประกอบ 21 แสดงจำนวนคำที่พบมากที่สุดในรูปแบบ Word Cloud 24

ภาพประกอบ 22 ภาพรวมของข้อความตั๋วงาน 24
ภาพประกอบ 23 กราฟจำนวนข้อความของตั๋วงาน 25
ภาพประกอบ 24 กราฟจำนวนคำที่พบมากที่สุด 25
ภาพประกอบ 25 ข้อมูลทางสถิติจำนวนคำของตั๋วงาน 26
ภาพประกอบ 26 กราฟจำนวนคำของตั๋วงาน 26
ภาพประกอบ 27 การหาจำนวนคำในเอกสาร 28
ภาพประกอบ 28 ได้ผลการนับคำในเอกสารทั้งหมด 28
ภาพประกอบ 29 การหาความถี่ของคำในเอกสาร 28
ภาพประกอบ 30 น้ำหนัก (Weight) ความถี่ของคำในเอกสาร 29
ภาพประกอบ 31 การหาความถี่ของเอกสารผกผัน 29
ภาพประกอบ 32 น้ำหนัก (Weight) ความถี่ของเอกสารผกผัน 29
ภาพประกอบ 33 น้ำหนักของคำสำคัญในข้อความ 30
ภาพประกอบ 34 นำหมายเลขตั๋วงานมาตรวจสอบกับตั๋วงาน 30
ภาพประกอบ 35 การแบ่งข้อมูลเพื่อใช้ในการทดสอบการทำนาย 31
ภาพประกอบ 36 แปลงชุดข้อมูลจากภาษาธรรมชาติไปเป็นชุดตัวเลข (TF-IDF) 32
ภาพประกอบ 37 หาค่า max_features ที่ดีที่สุด 32
ภาพประกอบ 38 กราฟแสดงค่า max_features ที่ดีที่สุด 33
ภาพประกอบ 39 หาข้อความตั๋วงานที่มีน้ำหนักเท่ากับศูนย์ 33
ภาพประกอบ 40 แสดงตั๋วงานที่มีน้ำหนักเท่ากับศูนย์ 34
ภาพประกอบ 41 สร้างแบบจำลองในการทำนาย 34
ภาพประกอบ 42 แบบจำลองในการประเมินผลลัพธ์การทำนาย 35
ภาพประกอบ 43 แสดงผลการทำงานในรูปแบบตาราง Confusion Matrix 36
ภาพประกอบ 44 ทดสอบการทำนายข้อความตั๋วงานที่คลุมเคลือ 36

ภาพประกอบ 45 แสดงน้ำหนักของคำของตัวงานที่คลุมเคลือและทำนายไม่ถูก 37
ภาพประกอบ 46 แสดงน้ำหนักของคำของตัวงานที่คลุมเคลือแต่ทำนายถูก 38
ภาพประกอบ 47 เปรียบเทียบการตัดคำระหว่าง PyThaiNLP กับ NLTK 40
ภาพประกอบ 48 ผลลัพธ์การตัดคำด้วย NLTK 40
ภาพประกอบ 49 คุณลักษณะที่ได้จากการตัดคำโดย NLTK 41
ภาพประกอบ 50 ผลลัพธ์การกรองข้อความโดยใช้ RegEx 41
ภาพประกอบ 51 จำนวนคำที่ไม่สมบูรณ์ 1 และ 2 ตัวอักษร หรือ สระกับตัวอักษร 42
ภาพประกอบ 52 RegEx เพื่อกรองจำนวนคำที่ไม่สมบูรณ์ 42
ภาพประกอบ 53 ผลลัพธ์ของการทำความสะอาดข้อมูลโดย RegEx 43
ภาพประกอบ 54 ผลการทำนายจากการทำความสะอาดข้อมูลโดย RegEx 44
ภาพประกอบ 55 วัดประสิทธิภาพแบบจำลองด้วยเทคนิค Cross Validation 44
ภาพประกอบ 56 จำนวนคำที่เหลือจากการลบคำเฉพาะ 45
ภาพประกอบ 57 ผลการทำนายจากการตัดคำ 46
ภาพประกอบ 58 เปรียบเทียบจำนวนคำก่อนหลังการกรองโดย RegEx 46
ภาพประกอบ 59 คำที่มีผลต่อการทำนายของแบบจำลอง 48

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ผู้วิจัยได้ทำงานที่เกี่ยวข้องกับการแก้ปัญหาเครื่องคอมพิวเตอร์ ปัญหาด้านเครือข่าย ปัญหาการใช้งานโปรแกรม หรือปัญหาใดๆที่เกี่ยวข้องกับไอที เมื่อผู้ใช้งาน (User) ต้องการแจ้งปัญหาถึงฝ่ายสนับสนุนด้านไอที (IT Service Desk) ผู้ใช้งานสามารถแจ้งปัญหาได้หลายช่องทาง เช่น การโทร (Call) การเดินมาติดต่อกับแผนกไอทีโดยตรง (Walk-in) การส่งอีเมล (Mailing) การพูดคุยผ่านโปรแกรมสนทนา (Chatting Application) เมื่อสนับสนุนด้านไอทีได้รับเรื่องแล้ว จะทำการบันทึกปัญหาจากผู้ใช้งานลงในโปรแกรมระบบตั๋วงาน (Ticket Application) ที่ชื่อว่า โปรแกรมโอทีอาเอส (OTRS Application) โดย OTRS ย่อมาจาก Open-Source Ticket Request System เมื่อทำการบันทึกปัญหาจากผู้ใช้งานเรียบร้อยแล้ว โปรแกรมจะมีรายการการแจ้งปัญหาแสดงขึ้นมาในระบบ โดยรายการดังกล่าวจะถูกเรียกว่าตั๋วงาน (Ticket) ซึ่งตั๋วงานแต่ละรายการจะต้องถูกจำแนกออกเป็นประเภทต่างๆ โดยต่อไปนี้จะถูกเรียกว่า ประเภทตั๋วงาน (Service Category) บางครั้ง เช่น ถ้าผู้ใช้งานแจ้งปัญหาการเข้าใช้งานอินเตอร์เน็ตไม่ได้ ตั๋วงานนี้จะจัดอยู่ในประเภทปัญหาด้านเครือข่าย เป็นต้น ผู้ใช้งานเองสามารถที่จะแจ้งปัญหาผ่านโปรแกรมระบบตั๋วงานได้โดยตรงได้เช่นกัน แต่การที่ผู้ใช้งานเขียนตั๋วงานขึ้นเองนั้น ข้อความในตั๋วงานบางครั้งมีความคลุมเครือ มีรูปแบบการเขียนที่ไม่แน่นอน และจับใจความไม่ได้ บางทีข้อความอาจถูกเขียนด้วยภาษาไทย หรือภาษาอังกฤษ หรือเขียนผสมมาทั้งภาษาไทยและภาษาอังกฤษปนกัน อาจเป็นเพราะว่าผู้ใช้งานเขียนด้วยความเข้าใจของตัวเอง หรือเขียนแค่ให้ฝ่ายสนับสนุนไอทีรับทราบว่ามีปัญหาแต่ไม่รู้ว่าจะระบุว่าชัดเจนอย่างไร ตัวอย่างข้อความในตั๋วงานที่ผู้ใช้งานเขียนขึ้นมา เช่น “Ex ใช้งานไม่ได้” เป็นต้น ส่งผลให้ฝ่ายสนับสนุนด้านไอทีต้องใช้เวลาในการตีความข้อความของตั๋วงานอย่างมาก มีหลายครั้งที่จะต้องติดต่อกลับไปยังผู้ใช้งานเพื่อขอข้อมูลเพิ่มเติม เพื่อที่ฝ่ายสนับสนุนด้านไอทีจะได้ทำการแยกประเภทตั๋วงานได้ถูกต้อง ปัญหานี้ผู้ทำวิจัยได้หาทางแก้ไขปัญหาโดยการจัดอบรมให้กับผู้ใช้งานให้มีความเข้าใจถึงการใช้งานระบบด้านไอทีขององค์กร เพื่อที่จะได้เขียนตั๋วงานได้อย่างถูกต้อง แต่ด้วยปัจจัยหลายๆอย่าง รวมถึงพฤติกรรมของผู้ใช้งานเอง ทำให้แนวทางการแก้ปัญหานี้ไม่ได้ผลเท่าที่ควร ทำให้ตั๋วงานแบบนี้ก็ยังคงมีเข้ามาอยู่เช่นเดิม ผู้ทำวิจัยจึงคิดที่จะนำเอาเทคโนโลยีสารสนเทศมาช่วยในการแก้ปัญหานี้ จากที่ได้อ่านและค้นคว้างานวิจัยที่เกี่ยวข้องที่เกี่ยวข้องกับตั๋วงานจากหลากหลายแหล่งข้อมูล จึงได้พบเทคนิคที่จะนำมาช่วยในการ

แก้ไขปัญหานี้คือ เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing) หรือที่เรียกว่า NLP โดยเทคนิคดังกล่าวเป็นการนำเอาภาษาเขียนที่มนุษย์เข้าใจมาประมวลผลภาษาธรรมชาติ เพื่อแปลงเป็นรูปแบบของชุดข้อมูลที่คอมพิวเตอร์สามารถเข้าใจได้ โดยใช้หลักการเรียนรู้ของเครื่อง (Machine Learning) และใช้อัลกอริทึมแบบแยกประเภท (Classification Algorithm) เพื่อแยกประเภทตั๋วงานได้อย่างถูกต้องมากที่สุด

1.2 วัตถุประสงค์

1.2.1 เพื่อนำเอาเทคโนโลยีสารสนเทศมาช่วยในการแก้ปัญหานี้โดยการวิเคราะห์ข้อความของตั๋วงาน

1.2.2 เพื่อตรวจสอบข้อความของตั๋วงานของแต่ละประเภทว่ามีข้อความที่ไม่สมบูรณ์ในรูปแบบไหนบ้าง

1.2.3 เพื่อนำชุดข้อมูลของตั๋วงานที่ได้ไปประยุกต์ใช้กับเทคนิคการประมวลผลภาษาธรรมชาติ

1.2.4 เพื่อประยุกต์ใช้หลักการเรียนรู้ของเครื่องในการแยกประเภทตั๋วงานได้อย่างถูกต้องมากที่สุด

1.3 ขอบเขตการวิจัย

ผู้ทำวิจัยได้ศึกษาและทำวิจัยจากชุดข้อมูลจริงที่ผู้ใช้งานได้บันทึกไว้ในระบบตั๋วงานขององค์กร โดยข้อมูลดังกล่าวผู้ทำวิจัยได้ทำหนังสือรับรองขอความอนุเคราะห์เก็บข้อมูลเพื่องานวิจัยจากมหาวิทยาลัยถึงองค์กรเป็นที่เรียบร้อยแล้ว ชุดข้อมูลนี้ที่เก็บอยู่ในฐานข้อมูลของโปรแกรมโอทีอาเอสอยู่ระหว่างเดือนกรกฎาคม ปี 2561 ถึงเดือนธันวาคม ปี 2563 มีจำนวนตั๋วงาน 11,819 รายการ ที่ครอบคลุมประเภทของตั๋วงานทั้งหมด โดยมีตั๋วงานที่เป็นภาษาไทย 719 รายการ ภาษาอังกฤษ 11,000 รายการ คอลัมน์ที่นำมาใช้งานคือ คอลัมน์ตั๋วงาน (Title) จะเป็นหัวข้อที่เป็นข้อความของตั๋วงาน กับ คอลัมน์ประเภทของตั๋วงาน (Service Category) จะเป็นประเภทของตั๋วงานแต่ละรายการ ซึ่งคาดว่าจำนวนตั๋วงานนี้สามารถที่จะนำไปทดลองเพื่อให้ได้ผลลัพธ์ที่ถูกต้องที่สุดและเชื่อถือได้ งานวิจัยนี้ได้ใช้เทคโนโลยีสารสนเทศเพื่อทำการแยกประเภทของตั๋วงานได้อย่างถูกต้อง ข้อความของตั๋วงานจะถูกวิเคราะห์และศึกษาเพื่อให้เข้าใจถึงรูปแบบการเขียนข้อความตั๋วงานของผู้ใช้งาน เพื่อใช้เป็นแนวทางในการหาเทคนิคและขั้นตอนที่ดีที่สุดในการแยกประเภทตั๋วงาน จากนั้นดำเนินการทำความสะอาดและจัดรูปแบบชุดข้อมูลให้อยู่ในรูปแบบที่เหมาะสมเพื่อนำไปเข้าสู่กระบวนการประมวลผลภาษาธรรมชาติที่ใช้กับภาษาไทย (PythaiNLP)

และกระบวนการประมวลผลภาษาธรรมชาติที่ใช้กับภาษาอังกฤษ (NLTK) ขั้นตอนนี้จะใช้เครื่องมือเขียนด้วยภาษาไพธอน (Python programming language) โดยประเภทของตั๋วงานในชุดข้อมูลนี้จะเลือกเฉพาะที่อยู่ในขอบเขตหน้าที่รับผิดชอบของผู้ทำวิจัยเท่านั้น

1.4 ประโยชน์ที่ได้รับจากงานวิจัย

1.4.1 ผู้ใช้งานสามารถแจ้งปัญหาผ่านตั๋วงานเพียงแค่สั้นๆ โดยไม่ต้องเสียเวลาในการบรรยายถึงปัญหาในตั๋วงาน ทำให้การแจ้งปัญหาได้อย่างรวดเร็ว

1.4.2 ผู้สนับสนุนด้านไอทีสามารถแก้ปัญหาที่ถูกระบุประเภทตั๋วงานมาแล้วได้ทันทีและไม่เสียเวลา

1.4.3 ได้กระบวนการในการจำแนกตั๋วงานอัตโนมัติ

1.4.4 เพิ่มประสิทธิภาพในการจัดการตั๋วงาน และแก้ปัญหาได้เสร็จได้ตามข้อตกลงรวมกันในสัญญา (Service Leve Agreement: SLA)

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในงานวิจัยนี้ผู้ทำวิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องและนำเสนอตามหัวข้อดังต่อไปนี้

2.1. ตั๋วงานและกระบวนการในการสร้างตั๋วงาน (IT Service Desk Ticket and Ticket Creation process)

2.2 เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing)

2.3. อัลกอริทึมการทำนายและการวัดประสิทธิภาพ (Algorithm and Confusion Matrix)

2.4 งานวิจัยที่เกี่ยวข้อง

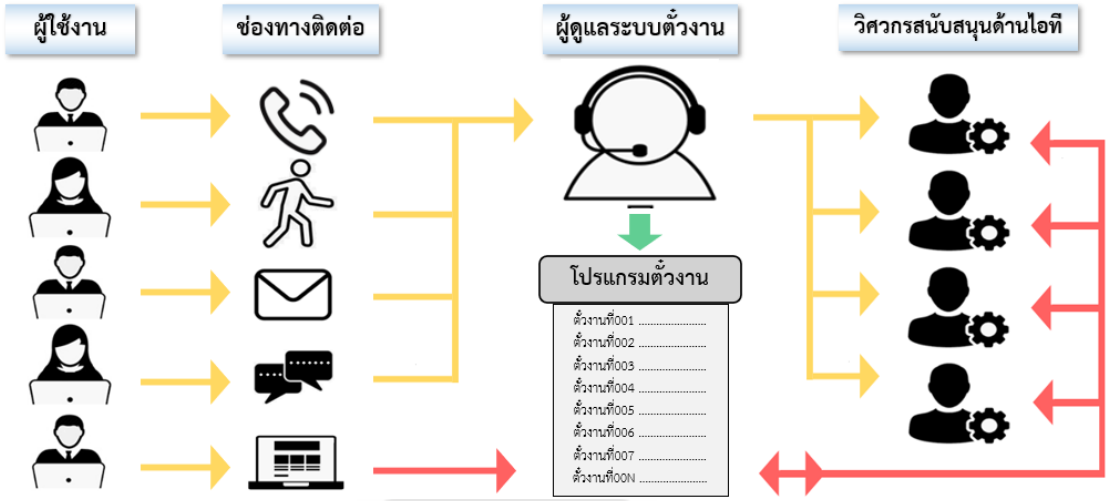
2.1 ตั๋วงานและกระบวนการในการสร้างตั๋วงาน

2.1.1 ตั๋วงาน (Service Desk Ticket)

เป็นบันทึกคำร้องจากผู้ใช้งานที่มีปัญหาในการใช้งานอุปกรณ์หรือระบบใดๆที่เกี่ยวข้องกับไอที ปัจจุบันโปรแกรมที่นำมาใช้กับตั๋วงานตามที่ได้กล่าวมานั้น มีอยู่หลากหลายโปรแกรม เช่น Zendesk, Frechservice, และ LiveAgent เป็นต้น โปรแกรมต่าง ๆ เหล่านี้มีการใช้งานคล้ายกัน แต่แตกต่างกันที่คุณลักษณะของตัวโปรแกรมเอง ในงานวิจัยนี้ผู้ทำวิจัยทำงานองค์กรที่ใช้โปรแกรมตั๋วงานที่มีชื่อว่า OTRS โปรแกรมนี้มีลักษณะการทำงานเหมือนโปรแกรมอื่นๆ คือ ทำการบันทึกปัญหาเข้าระบบตั๋วงาน โดยมีคุณลักษะตั๋วงานเช่น เวลาที่เกิดปัญหา (Create Time) ผลกระทบ (Impact) ความสำคัญ (Priority) และประเภทของตั๋วงาน (Service Category) เป็นต้น

2.1.2 กระบวนการในการสร้างตั๋วงาน (Ticket Creation Process)

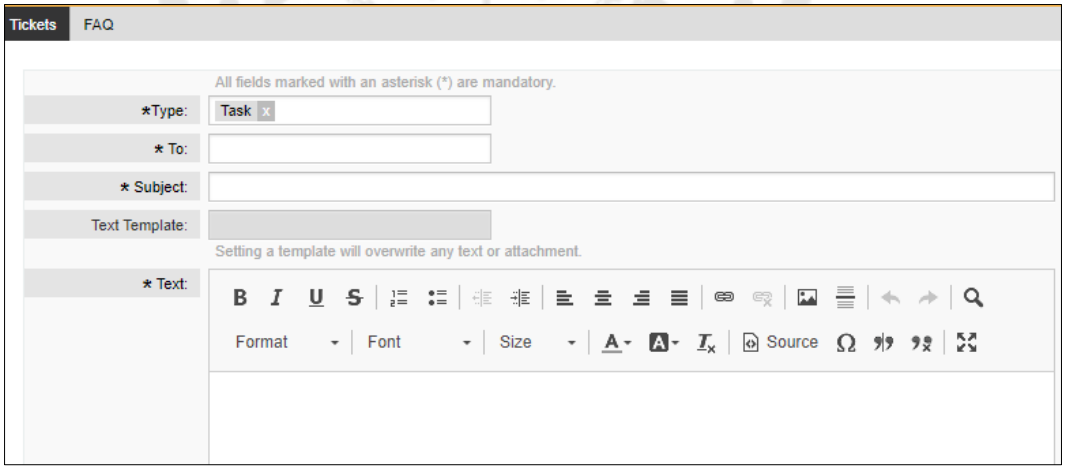
เริ่มต้นผู้ใช้งานสามารถแจ้งปัญหาให้กับทีมสนับสนุนด้านไอทีได้หลายช่องทาง ผู้ดูแลระบบตั๋วงานเมื่อได้รับแจ้งปัญหาแล้วจะทำการบันทึกปัญหาจากผู้ใช้ในระบบตั๋วงาน จากนั้นผู้ดูแลระบบตั๋วงานจะตรวจดูว่าวิศวกรสนับสนุนด้านไอทีแต่ละคนมีจำนวนตั๋วงานที่อยู่ในมือเท่าไรบ้าง เพื่อจะได้ส่งตั๋วงานให้ได้อย่างสมดุลกันตามกระบวนการลูกศรสีเหลือง (ดังภาพประกอบ 1) บางครั้งผู้ใช้งานสามารถทำการสร้างตั๋วงานได้ด้วยตัวเองผ่านระบบตั๋วงาน และถ้าหากวิศกร สนับสนุนด้านไอทีไม่มีงานในมือแล้ว ก็สามารถดึงตั๋วงานในระบบมาทำเองได้ตามกระบวนการลูกศรสีแดง (ดังภาพประกอบ 1)



ภาพประกอบ 1 กระบวนการการสร้างตั๋วงาน

2.1.3 โปรแกรมตั๋วงาน

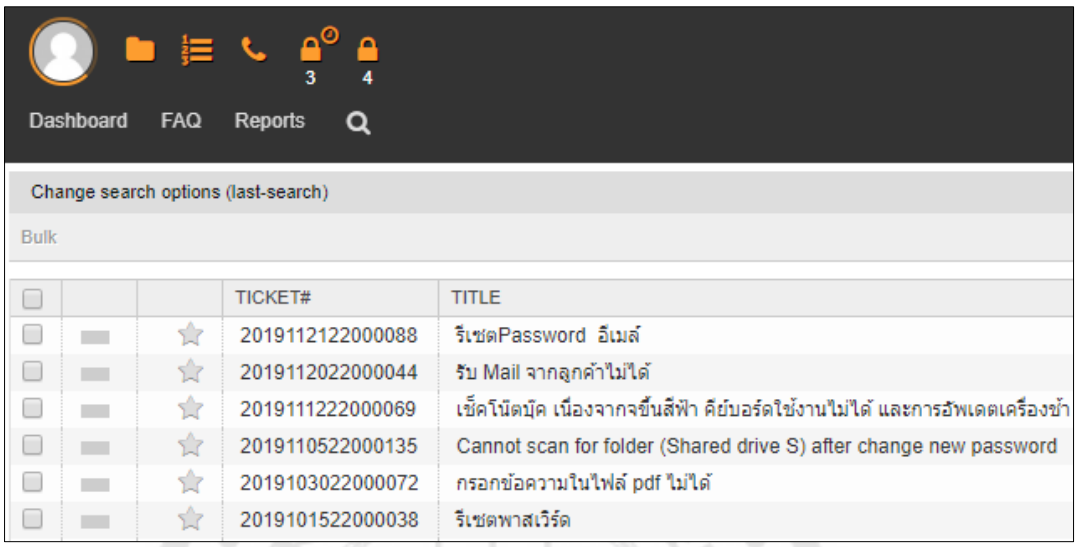
เมื่อผู้ดูแลระบบตั๋วงานหรือผู้ใช้งานทำการเข้าระบบตั๋วงานเรียบร้อยแล้วจะเห็นหน้าต่างการสร้างตัวงาน โดยหน้าต่างนี้จะมีช่องให้ใส่ข้อมูลพื้นฐานเช่น ประเภทการแจ้ง (Type) ผู้รับแจ้ง (To) หัวข้อตั๋วงาน (Subject) และ รายละเอียดตั๋วงาน (Text) ซึ่งจะคล้ายกับการส่งอีเมล์ (ดังภาพประกอบ 2)



ภาพประกอบ 2 หน้าต่างโปรแกรมการสร้างตั๋วงาน

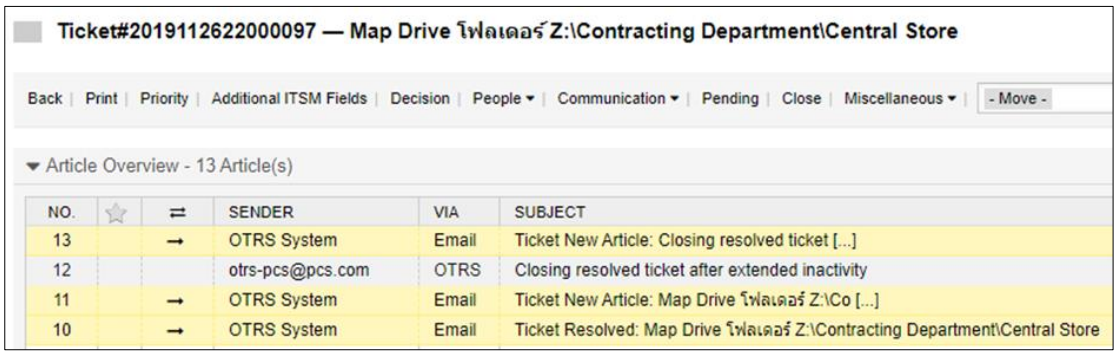
เมื่อสร้างเสร็จแล้วก็จะปรากฏตั๋วงาน ระบบจะทำการบันทึกเข้าฐานข้อมูลของตั๋วงาน และเมื่อบันทึกเสร็จแล้วจะมีตั๋วงานปรากฏขึ้นมาพร้อมกับเลขตั๋วงานที่เอาไว้ใช้อ้างอิงในการ

ติดตามความคืบหน้า ผู้ดูแลระบบตั๋วงานสามารถแจกจ่ายตั๋วงานให้กับวิศวกรด้านไอทีหรือวิศวกรด้านไอทีดึงเอาตั๋วงานมาทำเองได้ (ดังภาพประกอบ 3)



ภาพประกอบ 3 ตั๋วงานที่ถูกสร้างรอการนำไปดำเนินการ

วิศวกรด้านไอทีที่ทำการแก้ไขปัญหาจะต้องดำเนินการแก้ปัญหาให้อยู่ภายใต้กรอบระยะเวลาที่กำหนดไว้ใน SLA (Service Level Agreement) ซึ่งเป็นตามข้อกำหนดของสัญญาซึ่งเป็นไปตามมาตรฐานของ ITSM (IT Service Management) เมื่อทำตั๋วงานนั้นๆเสร็จแล้ว วิศวกรด้านไอทีจะทำการแจ้งเมลกลับ หรือ โทรศัพท์โทรแจ้งบอกกับผู้ใช้งานว่าคำร้องหรือปัญหาที่แจ้งมาได้รับการดำเนินการแก้ไขเป็นที่เรียบร้อยแล้ว จากนั้นวิศวกรด้านไอทีจะทำการปิดตั๋วงานและเป็นอันเสร็จสิ้นกระบวนการพื้นฐานของระบบตั๋วงาน ทั้งนี้กระบวนการดังกล่าวอาจจะมีขั้นตอนมากกว่านี้ ซึ่งเป็นไปตามนโยบายของแต่ละองค์กร (ดังภาพประกอบ 4)



ภาพประกอบ 4 ตั๋วงานที่ถูกแก้ไขแล้วตามเวลาที่กำหนด

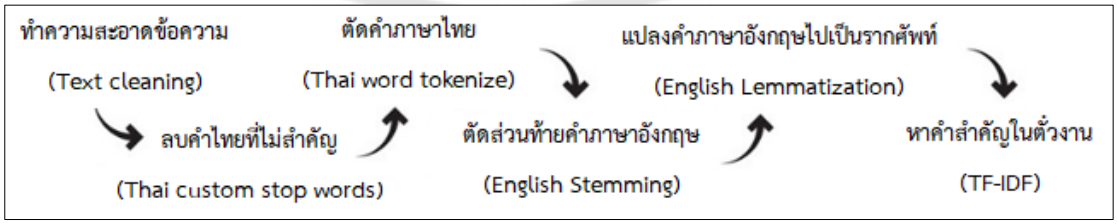
2.1.4 ปัญหาที่พบในระบบตั๋วงาน

เมื่อวิศวกรด้านไอทีได้รับตั๋วงานจากผู้ดูแลระบบตั๋วงานหรือจากผู้ใช้งาน ก็จะทำการวิเคราะห์ว่าปัญหาที่ผู้ใช้แจ้งเข้ามานั้นเป็นปัญหาประเภทใด มีความเร่งด่วนแค่ไหน มีข้อมูลครบถ้วนหรือไม่ ปัญหาที่ถูกต้องตามความเป็นจริงหรือไม่ เป็นต้น ขั้นตอนดังกล่าวค่อนข้างจะใช้เวลาพอสมควร ในบางครั้งวิศวกรด้านไอทีจะต้องโทรกลับไปถามข้อมูลกับผู้ใช้งานเพิ่มเติม เนื่องจากข้อมูลที่แจ้งเข้ามามีความคลุมเครือ หรือบางครั้งเป็นข้อความที่สั้นจนไม่สามารถเข้าใจได้ ทำให้วิศวกรด้านไอทีไม่สามารถแยกประเภทของตั๋วงานได้อย่างถูกต้อง จึงเป็นที่มาที่จะนำเอาเทคโนโลยีสารสนเทศมาช่วยในการแก้ปัญหานี้ (ดังภาพประกอบ 5)

ภาพประกอบ 5 ตัวอย่างประโยคจากผู้ใช้งานที่คลุมเครือ

2.2 เทคนิคการประมวลผลภาษาธรรมชาติ

การประมวลภาษาธรรมชาติที่เรียกว่า NLP (Natural Language Processing) เป็นส่วนหนึ่งของปัญญาประดิษฐ์หรือ AI (Artificial Intelligence) เพื่อให้คอมพิวเตอร์สามารถเข้าใจและตีความภาษามนุษย์ได้ กระบวนการโดยทั่วไปจะประกอบไปด้วย (ดังภาพประกอบ 6)



ภาพประกอบ 6 การประมวลผลภาษาธรรมชาติ

2.2.1 การทำความสะอาดข้อความ (Text Cleansing)

ประโยคหรือข้อความในภาษาของมนุษย์ไม่ได้มีแค่ตัวอักษรหรือสระ (ในภาษาไทย) แต่จะมีเครื่องหมายหรือสัญลักษณ์ต่างๆประกอบอยู่ด้วย เช่น เครื่องหมายตกใจ (!) เครื่องหมาย

แอด (@) เครื่องหมายคำพูด (“”) เป็นต้น เครื่องหมายเหล่านี้ไม่มีความหมายในการประมวลผลภาษาธรรมชาติ และจะถูกนำออกไปจากข้อความเพื่อลดจำนวนของคำ

2.2.2 การตัดคำในข้อความ (Tokenization)

การแบ่งคำในประโยคหรือข้อความออกมาเป็นคำ ๆ หรือการกำหนดขอบเขตของคำอย่างถูกต้องตามหลักของภาษานั้นๆ เช่น ข้อความภาษาอังกฤษ “I am very happy” ถูกตัดคำออกมาได้เป็น “I”, “am”, “very”, “happy” หรือข้อความภาษาไทย “ฉันมีความสุขมาก” ถูกตัดคำออกมาได้เป็น “ฉัน”, “มี”, “ความสุข”, “มาก” เป็นต้น.

2.2.3 การตัดส่วนท้ายของคำ (Stemming)

เป็นตัดส่วนส่วนท้ายของคำแบบหยาบๆ ทำให้คำลดสั้นลงเหลือแต่ส่วนหน้าของคำ เช่น “universe”, “university”, “universal” ได้ผลลัพธ์คือ “universe“ บางครั้งผลลัพธ์ของคำที่ผ่านการตัดส่วนท้ายอาจจะผิดจากความหมายเดิมได้ และไม่มีในดิกชันนารี ตัวอย่างเช่น “fly”, “flying” ได้ผลลัพธ์คือ “fri” เป็นต้น

2.2.4 การแปลงคำให้เป็นรากศัพท์ (Lemmatization)

การแปลงคำต่างๆ ให้อยู่ในรูปของรากศัพท์ของคำนั้นๆ วิธีนี้จะได้ผลลัพธ์ที่ดีว่าการทำ Stemming เพราะคำที่ได้จะมีอยู่ในดิกชันนารี เช่น “is”, “am”, “are” ได้ผลลัพธ์คือ “be” เป็นต้น

2.2.5 การหาคำสำคัญในข้อความ (TF-IDF)

โดยใช้เทคนิคที่ชื่อว่า TF-IDF (Term Frequency - Inverse Document Frequency) (Qaiser & Ali, 2018)ซึ่งการหาคำสำคัญสามารถหาได้จากนำผลคูณของทั้งสองค่า คือ ความถี่ของคำ (TF) กับความถี่เอกสารผกผัน (IDF) และจะได้ผลลัพธ์เป็นน้ำหนักของคำสำคัญออกมาตามสูตรดังนี้

$$w_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

เมื่อ tf = ความถี่ของคำในเอกสาร

idf = การผกผันในความถี่ของเอกสาร

w (weight) = ค่าน้ำหนักของคำ

t (term) = คำในเอกสาร

d (document) = เอกสาร

2.3 อัลกอริทึมการทำนายและการวัดประสิทธิภาพ (Algorithm and Confusion Matrix)

2.3.1 อัลกอริทึมสุ่มป่าไม้แบบแยกประเภท (Random Forest Classification)

เป็นหนึ่งในโมเดลของการเรียนรู้ของเครื่อง (Machine Learning) ที่ต่อยอดมาจากอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree) จากความสามารถของอัลกอริทึมต้นไม้ตัดสินใจที่มีจำนวนต้นไม้เพียง 1 ต้นในการแยกประเภท แต่การสุ่มป่าไม้เป็นการเพิ่มจำนวนต้นไม้หลายๆ ต้น เพื่อเพิ่มประสิทธิภาพให้สูงขึ้นและแม่นยำยิ่งขึ้น ต้นไม้แต่ละต้นมีการตัดสินใจอิสระต่อกัน เปรียบเหมือนการเลือกตั้งที่คนแต่ละคนจะต้องตัดสินใจเลือกผู้แทนด้วยตัวเองมากที่สุด โดยไม่ใช้ข้อมูลผู้อื่นมาใช้ในการตัดสินใจของเรา เช่นเดียวกัน การสุ่มป่าไม้จะแบ่งคุณลักษณะของต้นไม้แต่ละต้นไม่ซ้ำกัน เพื่อให้ต้นไม้มีความหลากหลายและเป็นอิสระกันมากขึ้น การสุ่มป่าไม้จะสุ่มข้อมูลจากชุดข้อมูลที่จะนำมาใช้ในการเรียนรู้ให้กับต้นไม้หลายๆรอบ โดยทั่วไปเป็นการสุ่มข้อมูลแบบบู๊ตสแตร็ป (Bootstrap) คือ หากเรามีกล่องกระดาษที่มีเบอร์ติดไว้ที่กล่องตั้งแต่ 0-9 จากนั้นเราทำการสุ่มเลือกกล่องขึ้นมา 1 กล่อง โดยเมื่อเราเลือกแล้วก็จะวางกล่องหมายเลขนั้นกลับที่เดิม ทำอย่างนี้เท่ากับจำนวนของกล่องคือ 10 ครั้ง โดยแต่ละครั้งเราอาจจะสุ่มหยิบได้กล่องหมายเลขเดิมและบางกล่องอาจจะไม่ถูกหยิบขึ้นมาเลยก็ได้ ดังนั้นเมื่อเราสุ่มข้อมูลขึ้นมาแล้วสร้างต้นไม้หนึ่งต้น จากนั้นสุ่มข้อมูลใหม่แล้วสร้างต้นไม้อีกต้นโดยทำอย่างนี้วนไป จนสุดได้นำเอาผลลัพธ์ของต้นไม้แต่ละต้นมาโหวตการทำนาย โดยอันไหนถูกโหวตมากที่สุดก็จะได้คำตอบที่ดีที่สุด วิธีการนี้อีกอย่างว่า Ensemble Method ซึ่งเป็นหนึ่งในเทคนิคของการเรียนรู้ของเครื่อง โดยนำเอาตัวทำนายหลายๆ ตัวมารวมกันแล้วโหวตเพื่อที่จะทำให้ผลลัพธ์ออกมาดีที่สุด

2.3.2 การประเมินสมรรถนะแบบจำลองการทำนาย (Confusion Matrix)

ก่อนที่จะนำแบบจำลองไปใช้งานจริง จะต้องมีการประเมินผลลัพธ์การทำนาย การวัดประสิทธิภาพส่วนใหญ่จะวัดค่าจากตาราง Confusion Matrix ซึ่งเป็นการวัดความสามารถของแบบจำลองในการแก้ปัญหาแบบแยกประเภท (ดังภาพประกอบที่ 7) และคำอธิบายความหมายของการวัดประสิทธิภาพ (ดังตารางที่ 1-4)

	Actually Positive(1)	Actually Negative(0)
Predicted Positive(1)	True Positives (TP)	False Positives (FP)
Predicted Negative(0)	False Negatives (FP)	True Negatives (TN)

ภาพประกอบ 7 ตาราง Confusion Matrix

ตาราง 1 การประเมินสมรรถนะแบบจำลองการทำนาย

การทำนายและผลลัพธ์	ความหมายของการประเมิน
True Positive (TP)	สิ่งที่แบบจำลองทำนายว่า จริง และผลลัพธ์คือ จริง
True Negative (TN)	สิ่งที่แบบจำลองทำนายว่า ไม่จริง และผลลัพธ์คือ ไม่จริง
False Positive (FP)	สิ่งที่แบบจำลองทำนายว่า จริง แต่ผลลัพธ์เป็น ไม่จริง
False Negative (FN)	สิ่งที่แบบจำลองทำนายว่า ไม่จริง แต่ผลลัพธ์เป็น จริง

เมื่อทราบถึงวิธีการวัดประสิทธิภาพของแบบจำลองแล้ว โดยทั่วไปจะมีตัววัดใช้ในงานวิจัยอยู่ 3 ค่า ที่ต้องทราบ (ตามตารางที่ 2)

ตาราง 2 การประเมินประสิทธิภาพของแบบจำลองในรูปแบบค่าต่างๆ

การทำนายและผลลัพธ์	ความหมายของการประเมิน
ความถูกต้อง (Accuracy)	วัดความถูกต้องของ Model โดยพิจารณารวมทุกคลาส
ความแม่นยำ (Precision)	วัดความแม่นยำของข้อมูล โดยพิจารณาแยกทีละคลาส
รีแคล (Recall)	วัดความถูกต้องของแบบจำลองโดยพิจารณาแยกทีละคลาส
F1-Score	หาค่าเฉลี่ยแบบ harmonic ระหว่าง precision และ recall

ตัววัดค่าดังกล่าวสามารถหาค่าได้จากสูตร (ตามตารางที่ 3)

ตาราง 3 สูตรคำนวณในการประเมินสมรรถนะของโมเดลรูปแบบค่าต่างๆ

การทำนายและผลลัพธ์	สูตรการประเมิน
ความถูกต้อง (Accuracy)	(TP+TN)/(TP+TN+FP+FN)
ความแม่นยำ (Precision)	TP / (TP + FP)
รีแคล (Recall)	TP / (TP+FN)
F1-Score	2*(precision*recall)/ (precision + recall)

การวัดประสิทธิภาพของแบบจำลองที่กล่าวมาข้างต้นเป็นการวัดแบบ 2 คลาส ในงานวิจัยนี้เป็นการจำแนกประเภทตั๋วงานที่มากกว่า 2 ประเภท จึงต้องมีการวัดค่าแบบ Micro และ Micro ด้วย (ตามตารางที่ 4)

ตาราง 4 สูตรคำนวณการประเมินสมรรถนะของแบบจำลองที่มีมากกว่า 2 คลาสขึ้นไป

การทำนายและผลลัพธ์	สูตรการประเมิน
Micro-average precision	$(TP_1$+…+TPk) / (TP1+…+TPk+FP1+…+FPk)
Micro-average recall	(TP1+TP2+…+TPn) / (TP1+…+TPn+FN1 +…+FNn)
Micro-average F1-Score	$(Pre_{micro}$ * $Recall_{micro})$ / $(Pre_{micro}$ + $Recall_{micro})$
Macro-average precision	(Prec1+Prec2+…+Precn) / n
Macro-average recall	(Recall1+Recall2+…+Recalln) / n
Macro-average of F1-Score	(F1-score1+F1-score2+F1-scroe$_n$) / n

2.4 งานวิจัยที่กี่ยวข้อง

(Paramesh, Ramya, & Shreedhara, 2018) Paramesh et al. เสนอวิธีการจำแนกประเภทตั๋วงานด้วยวิธีการใช้โมเดลการจำแนกประเภท (classification model) มากกว่าหนึ่งโมเดลร่วมกัน (Ensemble Method) ในการจำแนกประเภท โดยในการทดลองเปรียบเทียบสมรรถนะของการใช้แบบจำลองเดียว (Single Classifier) และแบบจำลองร่วม (Ensemble Classifiers) เช่น Decision Tree vs. Bagging-Tree vs. Random Forest, MNB vs. Bagging-MNB, SVM vs. Bagging-SVM, และ Voting classifier vs. Logistic Regression, KNN, MNB, และ SVM เป็นต้น Paramesh และทีมพบว่า Ensemble method ให้สมรรถนะสูงกว่าการใช้แบบจำลองเดี่ยวในการจำแนกประเภทตั๋วงาน ยกเว้น Voting classifier ที่ให้สมรรถนะเฉลี่ยของแบบจำลองเดี่ยวทุกตัวที่ในการเปรียบเทียบ.

(Paramesh & Shreedhara, 2019) Paramesh และ Shreedhara พัฒนาซอฟต์แวร์ระบบการจำแนกตั๋วงานอัตโนมัติ โดยใช้หลักการประมวลผลภาษาธรรมชาติ และหลักการเรียนรู้ของเครื่อง ตัวจำแนกประเภท Decision tree และ Naïve Bayes ถูกใช้สำหรับการจำแนกตั๋วงาน และ Bagging-Decision Tree, Bagging-Naïve Bayes และ Random Forest ซึ่งเป็น Ensemble

methods ถูกใช้ในการวิเคราะห์สมรรถนะของการจำแนกประเภท จากผลการทดลองพบว่า Ensemble methods ให้สมรรถนะสูงกว่าการใช้แบบจำลองเดี่ยว โดย Random Forest ให้สมรรถนะสูงที่สุด

(Revina, Buza, & Meister, 2020) Revina et al. ศึกษาเทคนิควิธีการต่างๆ ที่ใช้ในแต่ละขั้นของลำดับงานการจำแนกประเภทตั๋วงาน (Ticket Classification Pipeline) เพื่อหาว่าเทคนิคหรือวิธีการใดที่ให้ประสิทธิผลดีที่สุด เช่น ควรจะใช้คุณสมบัติ (Features) แบบใดในการแทนตั๋วงาน วิธีการในการเลือกคุณสมบัติ และประเภทของอัลกอลิทึมการเรียนรู้ของเครื่องที่เหมาะสม เป็นต้น เนื่องจากมีการศึกษาด้านเหล่านี้อย่างหลากหลายและแต่ละการศึกษาก็เสนอแนะที่แตกต่างกันออกไป โดยในงานวิจัย Revina และทีมงานเสนอการใช้คุณสมบัติที่มีการนำความเข้าใจทางภาษาเข้ามาผนวกรวมด้วย (Linguistic Features) คือ ความหมาย (Semantic) ความรู้สึก (Sentiment) และความรู้ความเข้าใจในองค์รวม (Meta-Knowledge) ซึ่งจากการศึกษาทีมวิจัยพบว่าการใช้คุณสมบัติที่นำความเข้าใจทางภาษามาผนวกรวมด้วยช่วยเพิ่มประสิทธิผลในการจำแนกตั๋วงานสูงกว่าการใช้ TF*IDF นอกจากนั้นยังพบว่าขั้นตอนในการเลือกคุณสมบัติมาใช้มีความสำคัญอย่างมาก และการเลือกใช้คุณสมบัติที่เหมาะสมสามารถเพิ่มสมรรถนะในการจำแนกตั๋วงานได้ ถึงแม้ว่าจะใช้อัลกอริทึมธรรมดาในการจำแนก

(Al-Hawari & Barham, 2019) Al-Hawari และ Barham พัฒนาระบบซอฟต์แวร์สำหรับช่วยแก้ปัญหาด้านไอทีในองค์กร (IT Service Desk System) โดยระบบแบ่งออกเป็น 4 ส่วน คือ ส่วนของผู้ดูแลระบบ ส่วนของผู้ใช้ที่เป็นพนักงาน ส่วนของการส่งอีเมล์อัตโนมัติ และส่วนของการจำแนกประเภทตั๋วงาน สำหรับในส่วนของการจำแนกตั๋วงาน ทีมวิจัยใช้แบบจำลองถุงคำ (Bags-of-words Model) และ TF*IDF สำหรับแทนตั๋วงาน และใช้หลักการเรียนรู้ของเครื่อง (Machine-Learning Approach) เข้ามาช่วยในการจำแนกตั๋วงาน จากผลการทดลอง ทีมวิจัยพบว่าการนำคำสนทนาแลกเปลี่ยน (Ticket Comments) ระหว่างบุคคลที่เกี่ยวข้องและคำอธิบายตั๋วงานเข้ามาผนวกรวมในเนื้อหาของตั๋วงาน สามารถช่วยเพิ่มความถูกต้องในการจำแนกตั๋วงานได้อย่างมาก

(Han & Akbari, 2018) Han และ Akbari ทำการทดลองเปรียบเทียบประสิทธิผลในการจำแนกตั๋วงานระหว่างการใช้วิธีการจำแนกประเภทพื้นฐานทั่วไป (Typical Classification Methods) คือ Rule-based, IR-based, และ SVM และวิธีการเรียนรู้เชิงลึก (deep-learning approach) คือ CNN with random words และ CNN with pre-trained word vectors จากผลการทดลอง ทีมวิจัยพบว่า CNN with pre-trained word vectors ให้สมรรถนะสูงที่สุด

(Li, Li, & Gong, 2019) Li et al. ศึกษาเปรียบเทียบประสิทธิผลของวิธีการคำนวณค่าถ่วงน้ำหนักของคำ 3 วิธีการสำหรับใช้ในการจำแนกตั๋วงานเกี่ยวกับข้อร้องเรียน (Ticket Complaints) ของผู้โดยสารที่ใช้บริการรถไฟ โดยในการทดลอง Li และทีมวิจัยเปรียบเทียบ TF*IDF, Word2Vec, และ TextRank สำหรับคำนวณค่าถ่วงน้ำหนักคำ และใช้ Naïve Bayes algorithm สำหรับสร้างแบบจำลองจำแนกประเภทตั๋วงาน ซึ่งพบว่า TF*IDF และ TextRank ให้ค่าความเที่ยงตรง (Precision) ที่ 0.778 และ 0.737 ตามลำดับ ซึ่งสูงกว่าการใช้ Word2Vec ที่ให้ค่าความเที่ยงตรง แค่ 0.320

(S.P & K.S, 2019) Building Intelligent Service Desk Systems using AI (Paramesh, Shreedhara 2019) ในงานวิจัยนี้ ทำโมเดลการจำแนก Ticket อัตโนมัติ โดยการวิเคราะห์คำจาก Ticket ที่ส่งมาจากผู้ใช้งาน วิธีการจำแนกจะใช้เทคนิคการสอนให้คอมพิวเตอร์เรียนรู้ (Supervised Machine Learning) เช่น Decision trees และ Naïve Bayes และ Random Forest เป็นต้น การประเมินผลประสิทธิภาพของตัวโมเดลจะใช้ข้อมูล Tickets (Dataset) ของจริง โมเดลที่ได้ผลดีที่สุดคือ Random Forest เมื่อทำเปรียบเทียบประสิทธิของแต่ละโมเดลที่กล่าวมา

บทที่ 3
วิธีดำเนินการวิจัย

ผู้ทำวิจัยได้กำหนดระเบียบวิธีการดำเนินงานวิจัย โดยมีขั้นตอนของการดำเนินงานวิจัย และบอกถึงรายละเอียดของแต่ละขั้นตอนที่สำคัญ (ดังภาพประกอบ 8)

3.1 การทบทวนวรรณกรรมและงานวิจัย

3.2 การวิเคราะห์และศึกษาข้อความของตั๋วงานจากข้อมูลจริง

3.3 ประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ

3.4 สร้างแบบจำลองการแยกประเภทตั๋วงาน

ภาพประกอบ 8 ภาพรวมวิธีดำเนินการวิจัย

3.1 การทบทวนวรรณกรรมและงานวิจัย

จากผลการสืบค้นและศึกษางานวิจัยที่เกี่ยวข้องกับการจำแนกตั๋วงานด้วยเทคนิคการเรียนรู้ของเครื่องของผู้วิจัย พบว่าเทคนิคการเรียนรู้ของเครื่องสามารถช่วยทำให้การจำแนกตั๋วงานเป็นไปได้โดยอัตโนมัติซึ่งช่วยลดเวลา ในกระบวนการโดยรวมในการแก้ปัญหาการใช้งานคอมพิวเตอร์ในองค์กรของผู้ใช้ได้รวมถึงประสิทธิผลในการจำแนก ตั๋วงานอยู่ในระดับที่น่าพอใจคือสูงกว่า 80% อย่างไรก็ตามงานวิจัยที่มีอยู่เน้นไปที่ตั๋วงานที่ถูกเขียนด้วยภาษาๆเดียว โดยเฉพาะอย่างยิ่งคือ ภาษาอังกฤษ ยังไม่พบงานวิจัยด้านนี้ที่ใช้กับตั๋วงานภาษาไทยหรือภาษาไทยและภาษาอังกฤษ ดังนั้นในบทความนี้ทีมวิจัยนำเสนอวิธีการจำแนกตั๋วงานโดยอัตโนมัติ

ด้วยหลักการเรียนรู้ของเครื่องสำหรับตั๋วงานที่ถูกเขียนด้วยภาษามากกว่าหนึ่งภาษาโดยเน้นไปที่ภาษาไทยและอังกฤษซึ่งวิธีการที่นำเสนอมีรายละเอียดคร่าวๆดังต่อไปนี้

3.1.1 พัฒนาขั้นตอนในการประมวลผลภาษาธรรมชาติสำหรับตั๋วงานที่เขียนด้วยภาษาไทยและภาษาอังกฤษ โดยใช้ไลบรารีแบบเปิด PythaiNLP และ NLTK

3.1.2 กำหนดชุดของชนิดของตั๋วงาน (A set of pre-defined labels) ที่สอดคล้องกับการใช้งานจริง

3.1.3 แปลงตั๋วงานไปอยู่ในรูปแบบของเวกเตอร์ด้วยหลักการถุงคำ (bags-of-words) และ TF*IDF

3.1.4 ใช้อัลกอริทึมการจำแนกประเภทแบบสุ่มป่าไม้ สำหรับจำแนกประเภทตั๋วงาน จากผลการประเมินสมรรถนะด้วยค่าความถูกต้องพบว่า ถึงแม้ว่าตั๋วงานจะถูกเขียนด้วยภาษาที่ผสมระหว่างไทยและอังกฤษ ค่าความถูกต้องยังอยู่ในระดับเดียวกับงานวิจัยที่มีอยู่สำหรับตั๋วงานภาษาอังกฤษภาษาเดียว คือ สูงกว่า 80% นอกจากนั้นในบางการทดลองย่อย ค่าความถูกต้องที่ได้สูงถึง 91%

3.2 การวิเคราะห์และศึกษาข้อความของตั๋วงานจากข้อมูลจริง

จากที่ได้กล่าวไว้ในบทนำ ชุดข้อมูลนี้ได้มาจากโปรแกรม OTRS ในองค์กรที่ผู้ทำวิจัยได้ทำงานอยู่ ชุดข้อมูลนี้ได้รับการยินยอมจากทางองค์กรเพื่อนำมาใช้ในการวิจัยเรียบร้อยแล้ว เครื่องมือที่ใช้ในการวิเคราะห์และศึกษากับชุดข้อมูลนี้จะใช้ภาษาไพธอน (Python Programing Language) ที่ทำงานบน Jupyter Notebook ที่เป็นสำหรับจัดการข้อมูลเป็นจำนวนมากๆ และถูกออกแบบมาตรงตามวัตถุประสงค์เพื่อใช้งานด้านข้อมูลโดยเฉพาะ (Data Science) โดยมีไลบรารีมากมายให้เรียกใช้งาน เช่น ไลบรารี Pandas ที่ใช้จัดการข้อมูล อ่านและนำเข้าข้อมูล แสดงข้อมูล ช่วยในการเตรียมข้อมูลให้สมบูรณ์ก่อนนำไปคำนวณ และ ไลบรารี NumPy ที่นำมาใช้คำนวณทางด้านคณิตศาสตร์ เป็นต้น

3.2.1 เริ่มต้นการวิเคราะห์และตรวจสอบชุดข้อมูลของตั๋วงาน

ผู้ทำวิจัยได้เรียกข้อมูลออกมาจากโปรแกรม OTRS และทำการ Export ออกมาเป็นไฟล์ Excel จากนั้นเปิด Jupyter Notebook เพื่อเริ่มการวิเคราะห์และศึกษาข้อมูลของชุดข้อมูลตั๋วงาน เมื่อนำเข้าชุดข้อมูลเสร็จแล้ว จะเห็นชุดข้อมูลโดยมีจำนวนของตั๋วงาน 11,819 และจำนวนคอลัมน์ 72 คอลัมน์ (ดังภาพประกอบ 9)

```
#Data from OTRS for Analysis
df_All = pd.read_excel('All-Ticket_Created_2020-10-29_17_42_TimeZone_Asia_Bangkok.xlsx')
df_All = df_All.drop(df_All.index[0])
df_All.loc[656:657]
```

	Number	Ticket#	Age	Title	Created	Last Changed	Close Time	Queue	
656	10589	2020090322000131	56 d 5 h	เข้าTeams ไม่ได้ค่ะ	2020-09-03 11:48:17 (Asia/Bangkok)	2020-09-06 13:20:11 (Asia/Bangkok)	2020-09-06 13:20:11 (Asia/Bangkok)	pcs-service-desk	c succe
657	1059	2018100422000168	756 d 6 h	เซ็ต E-Mail กลาง หน่วยงาน Premier	2018-10-04 10:58:10 (Asia/Bangkok)	2018-10-14 08:45:19 (Asia/Bangkok)	2018-10-14 08:45:19 (Asia/Bangkok)	pcs-service-desk	c succe

2 rows × 72 columns

```
#Data size
df_All.shape
```

```
(11819, 72)
```

ภาพประกอบ 9 นำเข้าชุดข้อมูลเพื่อทำการวิเคราะห์และศึกษา

ชุดข้อมูลที่ได้ผู้วิจัยได้ตรวจสอบว่ามีตั๋วงานที่มีภาษาไทยอยู่ในข้อความจำนวนเท่าไร โดยการเขียนฟังชั่นในไพธอน และใช้ regular expression ในการค้นหาและคัดกรอง ข้อความในรูปแบบที่เราต้องการ โดยเราได้ผลลัพธ์ข้อความที่มีภาษาไทยเท่ากับ 791 รายการ (ดังภาพประกอบ 10)

```
TH_All['Title'] = TH_All['Title'].map(lambda x: separate_language(x))
TH_All.replace('', np.nan, inplace=True)
TH_All.dropna(inplace=True)
TH_All.loc[47:80]
```

	Title
47	ยืมสายเพื่อใช้กับห้องประชุม
68	ขอริเซ็ทการเข้าใช้อีเมล
80	ขออีเมล์สำหรับพนักงานใหม่แก้ไขอีเมล์เนื่องจากพนักงานเปลี่ยนนามสกุลและอื่นๆ

```
TH_All.shape
```

```
(791, 1)
```

ภาพประกอบ 10 เลือกดูจำนวนตั๋วงานที่มีภาษาไทยในข้อความ

เมื่อตรวจสอบตั๋วงานแต่ละรายการก็พบว่ามีข้อความเป็นจำนวนมากที่ไม่สามารถตีความหมาย และไม่สามารถแยกประเภทตั๋วงานได้ (ดังภาพประกอบ 11)

```
TH_All.loc[2255:2257]
```

	Title
2255	Outlook cannot access.
2256	EX เปิดไม่ได้ค่ะ
2257	Compu via Citrix Error - MainMenu ถูกเรียกใช้งานอยู่

ภาพประกอบ 11 ข้อความตั๋วงานมีความคลุมเครือ

3.2.2 เลือกข้อมูลที่จะนำมาใช้

จากการตรวจสอบชุดข้อมูลเบื้องต้นจากข้อ 3.2.2 ขนาดของชุดข้อมูลคือ จำนวนตั๋วงานมีทั้งหมด 11,819 รายการ และมีจำนวนคอลัมน์ 72 คอลัมน์ ขั้นตอนต่อไปคือการเลือกข้อมูลที่จะนำมาใช้ในงานวิจัย ผู้ทำวิจัยได้เลือกมา 2 คอลัมน์ จากทั้งหมด 72 คอลัมน์ คือ คอลัมน์ข้อความตั๋วงาน (Title) คอลัมน์นี้เป็นคำร้องขอหรือการแจ้งปัญหาของผู้ใช้งานในโปรแกรมตั๋วงาน บางครั้งข้อความถูกเขียนเป็น ภาษาไทย ภาษาอังกฤษ หรือมีทั้งสองภาษาในประโยคเดียวกัน และคอลัมน์ ประเภทของตั๋วงาน (Service Category) คือ ประเภทของคำร้องขอหรือการแจ้งปัญหา ในส่วนของ Service Category นี้ เราจะใช้เป็น Label ในการจำแนกประเภทของตั๋วงาน (ดังภาพประกอบ 12)

```
#Copy df_All to TH_All
Chs_All = df_All.loc[:, ['Title','Service Category']].copy()
```

```
Chs_All.loc[655:657]
```

	Title	Service Category
655	Cannot access ESS Leave	Applications - ESS-Leave
656	เข้าTeams ไม่ได้ค่ะ	Office 365 - Teams
657	เซ็ต E-Mail กลาง หน่วยงาน Premier	Computers - Laptop Move

ภาพประกอบ 12 เลือกข้อมูลที่จะนำมาใช้

3.2.3 จัดรูปแบบชุดข้อมูล

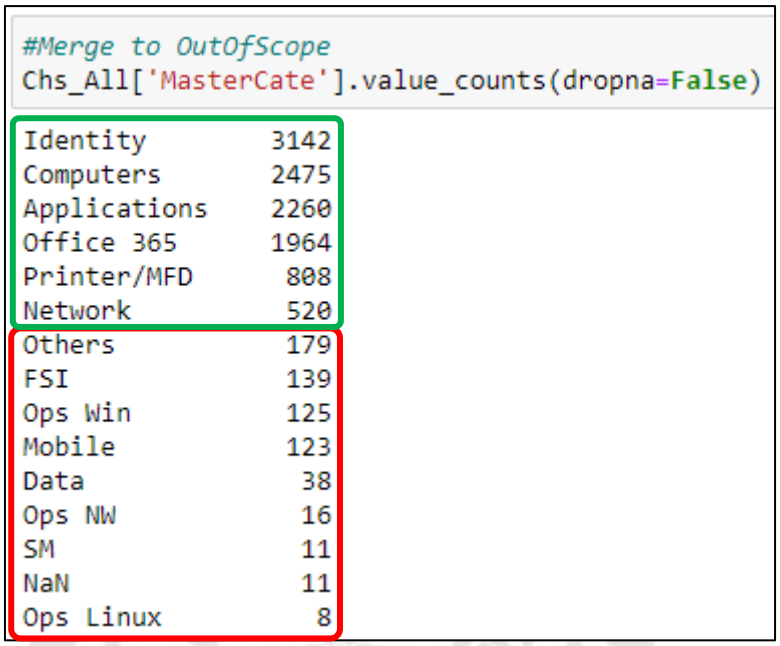
ข้อมูลชุดนี้มีประเภทของตั๋วงานทั้งหมดแบ่งออกเป็น 71 ประเภท ซึ่งขอบเขตของการแยกประเภทของตั๋วงานจะกว้างเกินไป ผู้ทำวิจัยจึงได้ตรวจสอบจึงพบว่า ประเภทของตั๋วงานจะประกอบไปด้วยประเภทหลักและประเภทรอง เช่น Computers – Laptop Move (ประเภทหลัก – ประเภทรอง) เป็นต้น ผู้ทำวิจัยจึงทำการแยกออกเป็นคอลัมน์นี้ใหม่เป็น ประเภทหลัก (MasterCate) และ ประเภทรอง (DetailCate) เพื่อลดขอบเขตของประเภทตั๋วงานให้น้อยลง (ดังภาพประกอบที่ 13)

```
# separate Service category to be Master and Detail
temp = Chs_All["Service Category"].str.split(" - ", n = 1, expand = True)
Chs_All["MasterCate"]= temp[0]
Chs_All["DetailCate"]= temp[1]
Chs_All.loc[655:657]
```

	Title	Service Category	MasterCate	DetailCate
655	Cannot access ESS Leave	Applications - ESS-Leave	Applications	ESS-Leave
656	เข้าTeams ไม่ได้ค่ะ	Office 365 - Teams	Office 365	Teams
657	เซ็ต E-Mail กลาง หน่วยงาน Premier	Computers - Laptop Move	Computers	Laptop Move

ภาพประกอบ 13 แบ่งคอลัมน์เป็น MasterCate และ DetailCate

เมื่อประเภทของตั๋วงานได้ถูกแบ่งแล้ว ผู้ทำวิจัยได้ดึงเอาคอลัมน์ MasterCate มาตรวจสอบว่ามีจำนวนประภทของตั๋วงานอยู่เท่าไร ผู้ทำวิจัยพบว่ามีจำนวนทั้งหมด 15 ประเภท ซึ่งลดลงมาก และเราจะได้ประเภทของตั๋วงานที่มีตั๋วงานสูงสุด 6 อันดับแรกคือ Identity, Computers, Applications, Office 365, Printer/MFD, Network และประเภทตั๋วงานที่เหลือถัดจากนี้จะถูกจัดอยู่นอกเหนือขอบเขตงาน (Out of Scope) ของผู้ทำวิจัย โดยพบว่ามีจำนวนของประเภทตั๋วงานที่อยู่นอกขอบเขตของงานอยู่ 9 ประเภท (ดังภาพประกอบที่ 14)



```
#Merge to OutOfScope
Chs_All['MasterCate'].value_counts(dropna=False)
```

Identity	3142
Computers	2475
Applications	2260
Office 365	1964
Printer/MFD	808
Network	520
Others	179
FSI	139
Ops Win	125
Mobile	123
Data	38
Ops NW	16
SM	11
NaN	11
Ops Linux	8

ภาพประกอบ 14 ประเภทของตัวงานที่นอกขอบเขต

ดังนั้นผู้ทำวิจัยจึงได้ยุบรวมประเภทตัวงานนอกเหนือขอบเขตที่ประกอบไปด้วย Others, FSI, Ops Win, Mobile, Data, Ops NW, SM, NaN, Ops Linux ไปเป็นประเภทตัวงานใหม่ชื่อว่า OutOfScope จึงลดจำนวนประเภทของตัวงานเหลืออยู่เพียง 7 ประเภท และได้สร้างคอลัมน์ใหม่จาก MasterCate เป็น Target (ดังภาพประกอบที่ 15)

```
Chs_All['Target'] = Chs_All['MasterCate'].replace(to_replace
                                                  =['Others', 'Data','Ops Win','Ops Linux',
                                                    'Ops NW', 'SM', 'Mobile','FSI', np.NaN]
                                                  , value='OutOfScope')
Chs_All['Target'].value_counts(dropna=False)
```

```
Identity        3142
Computers       2475
Applications    2260
Office 365      1964
Printer/MFD      808
OutOfScope       650
Network          520
Name: Target, dtype: int64
```

ภาพประกอบ 15 ประเภทตัวงานถูกยุบรวมเป็น OutOfScope

ตอนนี้ชุดข้อมูลได้ถูกลดจำนวนประเภทของตั๋วงานให้อยู่ในขนาดที่เหมาะสมเรียบร้อยแล้ว ต่อไปเป็นการแปลงประเภทข้อมูลของตั๋วงานเหล่านี้ให้อยู่ในรูปแบบของตัวเลข โดยกำหนดให้ Identity = 1, Computers = 2, Applications = 3, Office365 = 4, Printer/MFD = 5, Network = 6, OutOfScope = 7 เพื่อนำไปใช้ในแบบจำลองการทำนาย (ดังภาพประกอบที่ 16)

```
#Change to Number and Add column to Tarket
Chs_All['Target'].replace(['Identity', 'Computers', 'Applications',
                           'Office365', 'Printer', 'Network','OutOfScope' ],
                          [1, 2, 3, 4, 5, 6, 7], inplace=True)
```

```
Chs_All['Target'].value_counts(dropna=False)
```

```
1    3142
2    2475
3    2260
4    1964
5     808
7     650
6     520
Name: Target, dtype: int64
```

ภาพประกอบ 16 แปลงประเภทของตั๋วงานเป็นตัวเลข

เมื่อจัดรูปแบบของข้อมูลเรียบร้อยแล้ว ต่อไปเป็นการวิเคราะห์ชุดข้อมูลถึงรายละเอียดว่า ตั๋วงานที่ถูกสร้างด้วยผู้ใช้งานในแต่ละประเภทนั้นมีจำนวนเท่าไรและในจำนวนเหล่านี้มีข้อความของตั๋วงานที่คลุมเคลืออยู่จำนวนเท่าไร โดยผู้ทำวิจัยได้ไลบรารี่ pandas ในการเลือกข้อมูลดังกล่าวด้วยเงื่อนไข คือ คอลัมน์ Customer User ที่ไม่ใช่ชื่อของ ผู้ดูแลระบบตั๋วงานและระบุประเภทของตั๋วงานที่คอลัมน์ Target และคอลัมน์ Len ที่เป็นจำนวนของตัวอักษรทั้งหมดข้อความตั๋วงาน โดยจะเลือกรายการที่มีตัวอักษร น้อยกว่าหรือเท่ากับ 21 ตัวอักษร (ดังภาพประกอบที่ 17)

```
Dist_Users = Dist[(Dist['Customer User'] != 1) & (Dist['Customer User'] != 2) &
                  (Dist['Customer User'] != 3) & (Dist['Customer User'] != 4) &
                  (Dist['Customer User'] != 5) & (Dist['Customer User'] != 6)]
Dist_vague=Dist_Users[(Dist_Users['Target'] == 2) & (Dist_Users['Len'] <= 21)]
Dist_vague
```

ภาพประกอบ 17 ตั๋วงานที่คลุมเครือในแต่ละประเภทตั๋วงาน

จากนั้นผู้ทำวิจัยทำการหาข้อความตั๋วงานที่คลุมเคลือด้วยมือ (Manual) จนจำนวนตั๋วงานที่คลุมเคลือตามที่ต้องการ พร้อมกับคำอธิบายความหมายประเภทของตั๋วงาน (ดังตารางที่ 5)

ตาราง 5 ความหมายของประเภทตั๋วงาน

ประเภทหลัก	ความหมายของประเภทตั๋วงาน	ตั๋วงาน	ผู้ใช้งาน	คลุมเคลือ
Identity	ปัญหาเกี่ยวกับบัญชีผู้ใช้งาน	3,142	600	0
Computers	ปัญหาเกี่ยวกับเครื่องคอมพิวเตอร์	2,475	191	14
Applications	ปัญหาเกี่ยวกับการใช้งานโปรแกรม	2,260	346	8
Office365	ปัญหาเกี่ยวกับ MS Office	1,964	252	5
Printer/MFD	ปัญหาเกี่ยวกับปริ้นเตอร์	808	88	1
Network	ปัญหาเกี่ยวกับการเชื่อมต่อเครือข่าย	520	125	1
OutOfScope	ปัญหานอกเหนือขอบเขตผู้ทำวิจัย	650	125	0

3.2.4 ลบคำเฉพาะ (Unique Term)

ผู้ทำวิจัยได้ตรวจสอบข้อความตั๋วงานในชุดข้อมูลโดยทราบว่าข้อความในตั๋วงานจะมีคำเฉพาะ (Unique Term) อยู่ 2 ประเภท คือ ชื่อบุคคลที่เป็นภาษาไทย ภาษาอังกฤษ เช่น “Chalermchai Pidej”, “เฉลิมชัย พิเดช” และชื่อคอมพิวเตอร์ เช่น “OCS-THNB5519779” เป็นต้น คำเหล่านี้เป็นคำที่ไม่สื่อความหมายและจะมีผลต่อการตัดคำในกระบวนการตัดคำ ซึ่งจะทำให้จำนวนคำที่ได้มีจำนวนมาก คำเหล่านี้จะต้องถูกลบออกจากชุดข้อมูลเพื่อลดจำนวนของคำที่จะนำไปเป็นคุณลักษณะ (Features) ในแบบจำลองการทำนาย จึงได้ทำการลบคำเหล่านั้นออกโดยตรงโดยใช้โค้ดไพธอนเป็นเครื่องมือในการดำเนินการ (ดังภาพประกอบที่ 18)

```
#Stopword ThaiName EngName SamName Email
ComName = pd.read_csv('NameOri.csv') #Read CSV Computer Name
ComName_lst = ComName['Name'].tolist() #Convert df to list
ComName_set = set(ComName_lst) #Conver list to set data type for Loop
#Remove from df
for w in ComName_set:
    stop_Train['Title'] = stop_Train['Title'].str.replace(w,'')

#Stopword Computer Name
ComName = pd.read_csv('ComNameOri.csv') #Read CSV Computer Name
ComName_lst = ComName['Name'].tolist() #Convert df to list
ComName_set = set(ComName_lst) #Conver list to set data type for Loop
#Remove from df
for w in ComName_set:
    stop_Train['Title'] = stop_Train['Title'].str.replace(w,'')
```

ภาพประกอบ 18 ลบคำเฉพาะที่เป็นชื่อบุคคล และ ชื่อคอมพิวเตอร์

3.3 ประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ

เนื่องจากข้อความในตั๋วงานประกอบด้วยข้อความที่เขียนเป็นภาษาไทยและภาษาอังกฤษ หรือภาษาไทยผสมกับภาษาอังกฤษ ในขั้นตอนการประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ ผู้ทำวิจัยได้ใช้ไลบรารีประมวลผลภาษาธรรมชาติที่ใช้กับภาษาไทยที่ชื่อว่า ไพไทยเอ็นแอลพี (PyThaiNLP) และไลบรารีที่ใช้ประมวลผลภาษาอังกฤษที่ชื่อว่า NLTK (Natural Language Toolkit) โดยขั้นตอนในการในการดำเนินการผู้ทำวิจัยได้เขียนฟังก์ชั่นในภาษาไพธอนชื่อว่า preprecess_row_text (ดังภาพประกอบที่ 19)

```
def preprocess_raw_text(raw_title):
    pattern = re.compile(r"[^\u0E00-\u0E7Fa-zA-Z ]|[']|[|]")
    cleaned_character_msg = re.sub(pattern, '', raw_title)
    words = word_tokenize(cleaned_character_msg.lower(), engine='newmm')
    meaningful_words = [w for w in words if not w in th_stop and not w in en_stop]
    cleaned_words = " ".join(filter(None, meaningful_words))
    return cleaned_words
```

ภาพประกอบ 19 ฟังก์ชั่นการประมวลผลภาษาธรรมชาติ

3.3.1 ทำความสะอาดข้อความ (Text Cleansing)

ขั้นตอนนี้เป็นการกรองเอาเฉพาะตัวอักษรและสระที่เป็นภาษาไทย และตัวอักษรภาษาอังกฤษเท่านั้น เครื่องมือที่ใช้คือไลบรารีที่ชื่อว่า RegEx (Regular Expression) เขียนได้เป็น [^\u0E00-\u0E7Fa-zA-Z']|[']|[|] โดยความหมายของเครื่องหมายได้อธิบายไว้ใน (ตารางที่ 6) และเขียนด้วยไพธอน (ดังภาพประกอบที่ 20)

```
#Regular Expression
  pattern = re.compile(r"[^\u0E00-\u0E7Fa-zA-Z' ]|['][|]")
  cleaned_character_msg = re.sub(pattern, '', raw_title)
```

ภาพประกอบ 20 การกรองข้อความด้วย Regular Expression

ตาราง 6 อธิบายความหมายของ Regular Expression

นิพจน์ทั่วไป	ความหมายของนิพจน์ทั่วไป
[^]	ลบกลุ่มของคำที่ตรงในวงเล็บข้ามออก
\u0E00--\u0E7Fa	Unicode แทนอักษรและสระทั้งหมดของภาษาไทย
a-zA-Z	อักศรภาษาอังกฤษตัวเลข ตัวใหญ่ ทั้งหมด
[ ]	ช่องว่าง (Space)
|	เงื่อนไข “หรือ” เช่น ไก่|ไข่ “ไก่ หรือ ไข่” เป็นต้น
[']	ตัดเครื่องหมายอัญประกาศออก
[|]	ตัดเครื่องหมายเส้นตรงออก

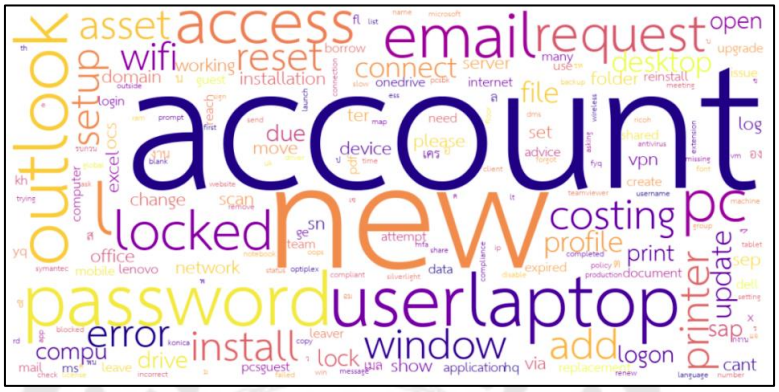
3.3.2 **การตัดข้อความออกเป็นคำ** (Word Tokenization)

ในงานวิจัยนี้ การตัดคำภาษาไทยเป็นสิ่งที่ท้าทายมาก บางครั้งประโยคหนึ่งอาจจะแปลได้หลายความหมาย เช่น “*หลวงตามหาบัว*” สามารถตัดคำได้เป็น [“หลวง”, “ตา”, “มหา”, “บัว”] หรืออีกความหมายคือ [“หลวง”, “ตาม”, “หา”, “บัว”] เป็นต้น ผู้ทำวิจัยได้ยกตัวอย่างข้อความในชุดข้อมูลนี้มาทดสอบตัดคำ เช่น “*เครื่องมุทิตา ปริ๊นงานไม่ได้หาเครื่องปริ้นไม่เจอ และ Drive share วางงานไม่ได้ค่ะ*” เมื่อผ่านการกรองคำและการตัดคำแล้วจะได้ผลลัพธ์คือ “เครื่อง”, “ปริ๊น”, “งาน”, “หา”, “เครื่อง”, “ป”, “ ริ้น”, “เจอ”, “drive”, “share”, “วาง”, “งาน” เป็นต้น

3.3.3 **ลบคำที่ไม่สื่อความหมาย** (Stopword Removal)

ในขั้นตอนนี้จะเป็นการลบคำทั่วไปที่ไม่มีความหมาย ในPyThaiNLP มีไลบรารีชื่อว่า thai_stopwords เป็นกลุ่มของคำภาษาไทยที่ไม่สื่อความหายจำนวนทั้งหมด 1,030 คำ เช่นคำว่า “ทั้งนั้น”, “จริงจัง”, “นี่แหละ”, “รวมกัน” เป็นต้น และใน NLTK มีไลบรารีชื่อว่า stopwords เป็นกลุ่มของคำภาษาอังกฤษที่ไม่สื่อความหายจำนวนทั้งหมด 179 คำ เช่นคำว่า “these”, “are”,

“into”, “be” เป็นต้น เมื่อข้อความของตั๋วงานทั้งหมดถูกตัดออกเป็นคำและลบคำไม่สื่อความหมายแล้ว ผู้วิจัยได้นำเอาผลลัพธ์ที่ได้มานำจำนวนคำที่พบมากที่สุดในรูปแบบที่ดูง่ายที่สุด โดยใช้ไลบรารีที่ชื่อว่า Word Cloud (ดังภาพประกอบที่ 20) และนับจำนวนคำของแต่ละตั๋วงานว่ามีจำนวนเท่าไร (ดังภาพประกอบที่ 21)



ภาพประกอบ 21 แสดงจำนวนคำที่พบมากที่สุดในรูปแบบ Word Cloud

เมื่อได้ชุดข้อมูลที่ทำความสะอาดเรียบร้อยแล้ว แล้ว ผู้ทำวิจัยทำการวิเคราะห์ข้อมูลเชิงลึกด้วย EDA โดยใช้ไลบรารี่ Dataprep.eda ตัวนี้สามารถวิเคราะห์ข้อมูลทางสถิติของข้อความตั๋วงาน (Title) และ จำนวนคำในแต่ตั๋วงาน (Total_Words) เพื่อทำให้เข้าใจชุดข้อมูลของตั๋วงานได้ดีมากยิ่งขึ้น (ดังภาพประกอบที่ 22-26)

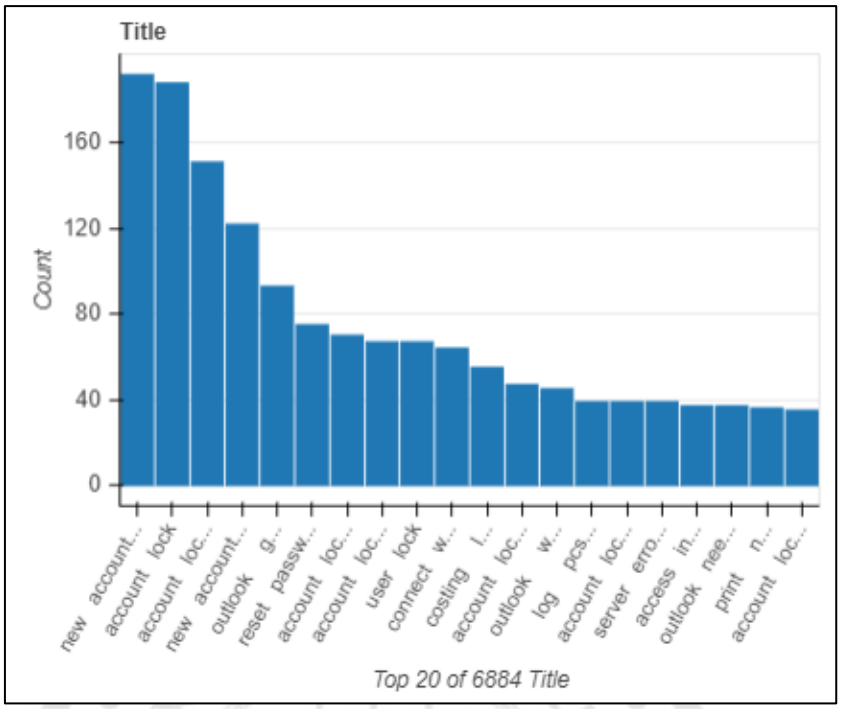
```
plot(nlp_Train,'Title')
```

Stats | Bar Chart | Pie Chart | Word

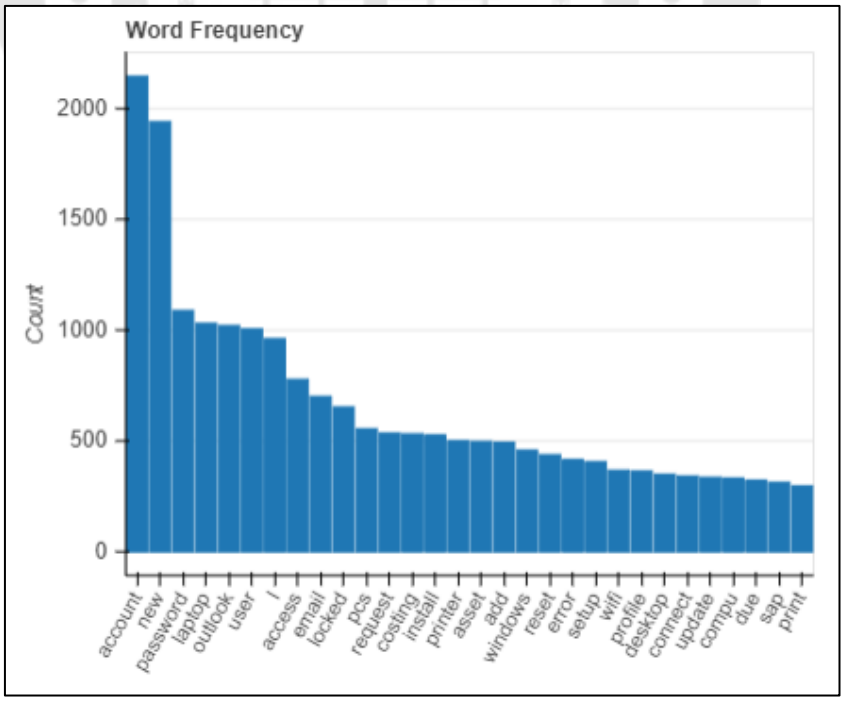
Overview

Distinct Count	6884
Unique (%)	58.2%
Missing	0
Missing (%)	0.0%
Memory Size	1.2 MB

ภาพประกอบ 22 ภาพรวมของข้อความตั๋วงาน



ภาพประกอบ 23 กราฟจำนวนข้อความของตัวงาน



ภาพประกอบ 24 กราฟจำนวนคำที่พบมากที่สุด

```
plot(nlp_Train,'Total_Words')
```

Stats | Histogram | KDE Plot | Normal Q-Q Plot | Box Plot

Overview

Distinct Count	18
Unique (%)	0.2%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Memory Size	184.7 KB
Mean	4.1298
Minimum	0
Maximum	22
Zeros	2
Zeros (%)	0.0%
Negatives	0
Negatives (%)	0.0%

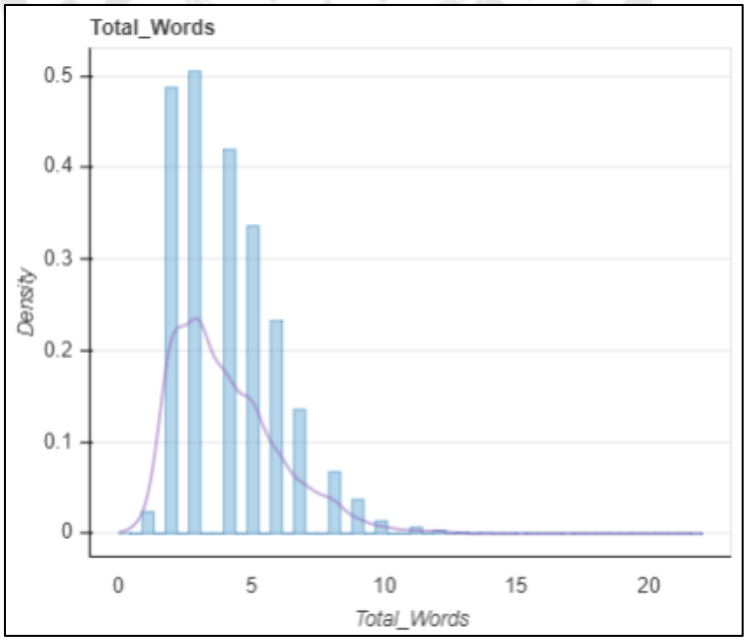
Quantile Statistics

Minimum	0
5-th Percentile	2
Q1	3
Median	4
Q3	5
95-th Percentile	8
Maximum	22
Range	22
IQR	2

Descriptive Statistics

Mean	4.1298
Standard Deviation	1.9316
Variance	3.731
Sum	48810
Skewness	1.0369
Kurtosis	1.7043
Coefficient of Variation	0.4677

ภาพประกอบ 25 ข้อมูลทางสถิติจำนวนคำของตัวงาน



ภาพประกอบ 26 กราฟจำนวนคำของตัวงาน

ผลลัพธ์ที่ได้จากการทำ Exploratory Data Analysis (EDA)

(a) สถิติเชิงพรรณนาของข้อมูลตั๋วงาน (Title)

- ข้อมูลโดยรวม (Overview) ข้อความของตั๋วงานมีทั้งหมด 11,819 รายการ เป็นข้อความที่ไม่ซ้ำกัน (Distinct Count) 6,884 รายการ คิดเป็นร้อยละ 58.2% ของชุดข้อมูล ข้อความที่สูญหาย (Missing) เท่ากับ 0

- กราฟแท่ง (Bar Chart) จำนวนข้อความตั๋วงานสูงสุด 20 อันดับจาก 6,884 ข้อความที่ไม่ซ้ำกัน จะเห็นว่าส่วนใหญ่ตั๋วงานจะอยู่ในประเภทของ Identity

- จำนวนคำที่พบ(Word Frequency) จำนวนคำที่พบมากที่สุดมีจำนวนมากว่า 2,000 คำ คือ คำว่า account ซึ่งจัดอยู่ประเภทของ Identity

(b). สถิติเชิงพรรณนาของจำนวนคำแต่ละตั๋วงาน (Total_Words)

- ข้อมูลโดยรวม (Overview) จำนวนคำทั้งหมดของแต่ละข้อความที่ไม่ซ้ำกัน 18 รายการ คิดเป็นร้อยละ 0.2% ของชุดข้อความที่สูญหายเท่ากับ 0 จำนวนคำสูงสุด(Max) 22 จำนวนคำน้อยสุด (Min) เท่ากับ 0 คือตั๋วงานที่ไม่มีคำเลยเท่ากับ 2 ตั๋วงาน

- สถิติเชิงพรรณนา (Descriptive Statistics) ค่าเฉลี่ยของคำทั้งหมดของชุดข้อมูลตั๋วงานเท่ากับร้อยละ 4.1298% คือประมาณ 4 คำต่อตั๋วงาน ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของจำนวนของคำในแต่ละตั๋วงานของข้อมูลชุดนี้มีการระจายตัวอยู่ที่ 1.9316% คือประมาณ 2 ต่อตั๋วงาน จำนวนคำทั้งหมด (Sum) เท่ากับ 48,810 คำ

- กราฟความหนาแน่น (KDE Plot) โดยจำนวนของคำจะกระจุdตัวอยู่ช่วงจำนวนคำระหว่า 0-10 คำ

3.3.4 การหาคำสำคัญในข้อความ (TF-IDF)

เมื่อผ่านกระบวนการลบคำไม่สำคัญและการตัดคำเรียบร้อยแล้ว ต่อไปนี้จะนำข้อมูลที่ได้ไปทำการหาคำสำคัญในข้อความโดยมีกระบวนการดังนี้

3.3.4.1 การหาจำนวนคำในข้อความตั๋วงาน (Word Count)

เป็นการนับจำนวนของคำทั้งหมดที่ได้จากการตัดคำเพื่อให้ได้คำเฉพาะออกมาจากทุกข้อความตั๋วงาน ในงานวิจัยนี้ใช้ไลบรารีชื่อว่า CountVectorizer ในภาษาไพธอนเป็นเครื่องมือในการหาจำนวนคำในเอกสาร (ดังภาพประกอบที่ 27) และได้ผลการนับคำในเอกสารทั้งหมด (ดังภาพประกอบที่ 28)

```
tf_plot = pd.DataFrame({'Term': cv.get_feature_names(),
                        'Occurrences':np.asarray(wc.sum(axis=0)).ravel().tolist()})
tf_plot.sort_values(by='Occurrences',ascending=False).head()
```

ภาพประกอบ 27 การหาจำนวนคำในเอกสาร

	Term	Occurrences
9	account	2144
850	new	1939
936	password	1087
677	laptop	1029
905	outlook	1019

ภาพประกอบ 28 ได้ผลการนับคำในเอกสารทั้งหมด

3.3.4.2 **การหาความถี่ของคำในเอกสาร** (Term Frequency)

คือการหาน้ำหนักของคำโดยถ้าคำใดปรากฏบ่อยในเอกสารนั้นๆ คำนั้นจะมีความเกี่ยวข้องกับใจความของเอกสารนั้น วิธีการคำนวณหาความถี่ของคำในเอกสารจาก การนำจำนวนคำที่ปรากฏในเอกสาร หารด้วย จำนวนคำทั้งหมดในเอกสาร เมื่อใช้ภาษาไพธอนในการคำนวณหาความถี่ของคำในเอกสาร (ดังภาพประกอบที่ 29) และได้น้ำหนัก (Weight) ความถี่ของคำในเอกสาร (ดังภาพประกอบที่ 30)

```
def tf(corpus):
    dic={}
    for document in corpus:
        for word in document.split():
            if word in dic:
                dic[word] = dic[word] + 1
            else:
                dic[word]=1
    for word,freq in dic.items():
        print(word,freq)
        dic[word]=freq/sum(map(len,(document.split()
                                    for document in corpus)))
    return dic
```

ภาพประกอบ 29 การหาความถี่ของคำในเอกสาร

```
TF_Score = tf(nlp_Train.loc[152, ['Title']])
TF_Score
```

```
fw 1
please 1
delete 1
account 2
ger 1
r 2
executive 1

{'fw': 0.1111111111111111,
 'please': 0.1111111111111111,
 'delete': 0.1111111111111111,
 'account': 0.2222222222222222,
 'ger': 0.1111111111111111,
 'r': 0.2222222222222222,
 'executive': 0.1111111111111111}
```

ภาพประกอบ 30 น้ำหนัก (Weight) ความถี่ของคำในเอกสาร

3.3.4.3 **การหาความถี่ของเอกสารผกผัน** (Inverse Data Frequency)

คือการหาน้ำหนักของคำที่บ่งชี้ว่าคำทั่วไป (Common word) ถูกใช้อย่างไร ยิ่งมีการใช้งานเอกสารบ่อยขึ้นคะแนนก็จะยิ่งลดลง ยิ่งคะแนนต่ำคำนั้นก็จะยิ่งมีความสำคัญน้อยลง วิธีการคำนวณหาความถี่ของคำในเอกสารจาก การนำเอาจำนวนของเอกสารทั้งหมดมาหารด้วยจำนวนเอกสารที่พบคำ เมื่อใช้ภาษาไพธอนในการคำนวณหาความถี่ของของเอกสารผกผัน(ดังภาพประกอบที่ 31) และได้น้ำหนัก (Weight) ความถี่ของเอกสารผกผัน (ดังภาพประกอบที่ 32)

```
idf_trans=TfidfTransformer(smooth_idf=True,use_idf=True)
idf_trans.fit(wc)
idf_wgh = pd.DataFrame(idf_trans.idf_, index=cv.get_feature_names(),columns=["IDF"])
idf_wgh.sort_values(by=['IDF']).head()
```

ภาพประกอบ 31 การหาความถี่ของเอกสารผกผัน

	IDF
account	2.707586
new	2.921672
password	3.413413
outlook	3.451953
laptop	3.452936

ภาพประกอบ 32 น้ำหนัก (Weight) ความถี่ของเอกสารผกผัน

3.3.4.4 การหาคำสำคัญในข้อความ (TF-IDF)

จะเป็นการช่วยกรองคำทั่วไปออกไป ทำให้รู้ว่าข้อความของตั๋วงานกล่าวถึงปัญหาเรื่องอะไรวิธีการคำนวณหาความถี่ของคำสำคัญในข้อความในเอกสารการนำเอาค่าความถี่ของคำ (TF) มาคูณกับค่าความถี่ของเอกสารผกผัน (IDF) เมื่อใช้ภาษาไพธอนในการคำนวณหาคำสำคัญในข้อความแล้ว จะได้น้ำหนักของคำสำคัญในข้อความ โดยผู้ทำวิจัยได้เลือกเอกสารที่ทำมาแสดงตั้งแต่หมายเลขเอกสาร (Index) ตั้งแต่ 3369 ถึง 3379 เพื่อให้ได้เห็นน้ำหนักของคำสำคัญทั้งภาษาไทยและภาษาอังกฤษ (ดังภาพประกอบที่ 33) และเมื่อนำไปทดสอบกับข้อความของตั๋วงานตามหมายเลขของเอกสารเลขที่ 3369 และ 3373 จะได้ผลลัพธ์ (ดังภาพประกอบที่ 34)

accont	accoun	account	...	โรค	โอน	ใช้กับ	ใช้งาน	ใช้ไม่ได้	ใบ	ใส่	ได	ไฟ	ไวไฟ
0.0	0.0	0.000000	...	0.0	0.447214	0.0	0.0	0.0	0.0	0.447214	0.0	0.0	0.0
0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.707107	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.447214	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.707107	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.707107	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0

ภาพประกอบ 33 น้ำหนักของคำสำคัญในข้อความ

	Index(3369)
โอน	0.537691
ใส่	0.504225
ข้อมูล	0.411497
notebook	0.395668
เครื่อง	0.361609
sawangkaew	0.000000
saraburi	0.000000

	Index(3373)
new	0.733468
account	0.679724
able	0.000000
sata	0.000000
screen	0.000000
scr	0.000000
scheduled	0.000000

ภาพประกอบ 34 นำหมายเลขตั๋วงานมาตรวจสอบกับตั๋วงาน

3.4 สร้างแบบจำลองการแยกประเภทตั๋วงาน

3.4.1 แบ่งชุดข้อมูล (Train-Test Split)

การสร้างแบบจำลองเพื่อการทำนาย ถ้าชุดข้อมูลที่ใช้ในการทำนายเป็นชุดเดียวกันจะทำให้เกิดความผิดพลาดได้ เปรียบเทียบแบบจำลองเหมือนกับนักเรียนรู้ข้อสอบแล้ว อย่างนี้เรียกว่า Overfitting และเมื่อแบบจำลองเจอชุดข้อมูลชุดใหม่ที่ยังไม่เคยเจอ (Yet-unseen Data) ส่วนมากผลลัพธ์การทำนายจะลดลง เปรียบเทียบเหมือนนักเรียนที่ไม่เคยเจอข้อสอบชุดนี้ ทำข้อสอบไม่ได้ ทำให้ได้คะแนนน้อยลง เพื่อที่จะหลีกเลี่ยงปัญหานี้ ข้อปฏิบัติที่ดีคือการแบ่งชุดข้อมูลออกเป็น 2 ส่วนคือ ข้อมูลเพื่อการฝึกฝน (Training set) และข้อมูลเพื่อการทดสอบ (Test set) โดยในงานวิจัยนี้ได้ใช้ไลบรารี่ชื่อว่า train_test_split เพื่อช่วยในการจัดแบ่งข้อมูล ดังภาพประกอบ (ดังภาพประกอบที่ 35)

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
    nlp_Train['Title'], nlp_Train['Target'], test_size=0.2, random_state=2019)
```

```
pd.DataFrame({"x_train":x_train.shape,"x_test":x_test.shape,
              "y_train":y_train.shape,"y_test":y_test.shape})
```

	x_train	x_test	y_train	y_test
0	9455	2364	9455	2364

ภาพประกอบ 35 การแบ่งข้อมูลเพื่อใช้ในการทดสอบการทำนาย

อธิบายค่าพารามิเตอร์ในการแบ่งข้อมูลดังนี้

(a) test_size = 0.2 คือ แบ่งชุดข้อมูลเพื่อการทดสอบออกเป็น 20% ของจากชุดข้อมูลทั้งหมด อีก 80% ที่เหลือคือชุดข้อมูลสำหรับการสอน (train_size)

(b) random_state=2019 คือ การกำหนดตัวเลขในการสุ่มข้อมูล โดยงานวิจัยนี้ได้กำหนดไว้อยู่ที่ 2019 ถ้าไม่กำหนดตัวเลขนี้ การเมื่อกำหนดแล้ว ผลของการแบ่งข้อมูล (split) ก็จะเหมือนเดิม เมื่อไม่กำหนด random_state ข้อมูลที่ได้ก็จะต่างกันกันไปทุกครั้งที่ทำงานแบ่งข้อมูล

เมื่อแบ่งข้อมูลเพื่อใช้ในทดสอบและประเมินประสิทธิภาพการทำงานของแบบจำลองแล้ว จึงนำผ่านกระบวนการการหาคำสำคัญ (TF-IDF) เพื่อแปลงชุดข้อมูลจากภาษาธรรมชาติไปเป็นชุดตัวเลข เพื่อที่คอมพิวเตอร์สามารถนำไปคำนวนได้ (ดังภาพประกอบ 36)

```
tfidf_vectorizer = TfidfVectorizer(tokenizer=str.split,max_features=1800)
tfidf_vsm_x_train = tfidf_vectorizer.fit_transform(x_train)
tfidf_vsm_x_test = tfidf_vectorizer.transform(x_test)
```

ภาพประกอบ 36 แปลงชุดข้อมูลจากภาษาธรรมชาติไปเป็นชุดตัวเลข (TF-IDF)

TfidfVectorizer มีพารามิเตอร์ที่ผู้ทำวิจัยตั้งค่าไว้ดังนี้

(a) tokenizer=str.split เป็นการตัดคำโดยใช้คำสั่ง str.split เนื่องจากชุดข้อมูลได้ผ่านกระบวนการประมวลผลภาษามาแล้ว จึงไม่ต้องใช้การตัดคำของ TfidfVectorizer

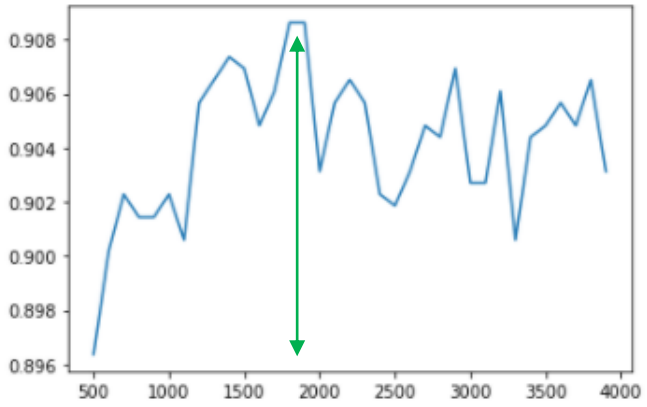
(b) max_features=1800 เป็นการกำหนดจำนวนคุณลักษณะเพื่อใช้ในการทำนาย ผู้ทำวิจัยได้เลือกจำนวนคุณลักษณะ 1,800 จากจำนวนทั้งหมด 3,981 คุณลักษณะตัวเลข 1,800 ได้มาจากการหาจำนวนที่ได้ผลลัพธ์ของการทำนายที่ดีที่สุด ผู้ทำวิจัยได้เขียนภาษาไพธอนให้แบบจำลองการทำการทำนายจากค่า max_features เริ่มต้นจาก 500 ถึง 3,900 โดยเพิ่มค่าทีละ 100 ซึ่งจะได้จำนวนรอบทั้งหมด 35 รอบ ค่าที่ได้จะเก็บไว้ใน List ชื่อว่า Accuracy เก็บค่าความถูกต้อง และ Max_Features เก็บจำนวนของคุณลักษณะที่ใช้ในการทำนาย เมื่อได้ค่าแล้วจึงนำไปทำการสร้างกราฟเพื่อแสดงว่าค่า max_features ใดที่ให้ผลการทำนายที่ดีที่สุด โดยผลที่ได้อยู่ที่ช่วงของ 1,800-1,900 (ดังภาพประกอบที่ 37-38)

```
KOB = 400
Accuracy = [ ]
Max_Features = [ ]
for x in range(35):
    KOB = KOB + 100
    tfidf_vectorizer = TfidfVectorizer(tokenizer=str.split,max_features=KOB)
    tfidf_vsm_x_train = tfidf_vectorizer.fit_transform(x_train)
    tfidf_vsm_x_test = tfidf_vectorizer.transform(x_test)
    rf = RandomForestClassifier(n_estimators=100, random_state=0, max_features=3)
    rf.fit(tfidf_vsm_x_train, y_train)
    rf_predict = rf.predict(tfidf_vsm_x_test)
    rf_accuracy = accuracy_score(y_test, rf_predict)
    Accuracy.append(rf_accuracy)
    Max_Features.append(KOB)
```

ภาพประกอบ 37 หาค่า max_features ที่ดีที่สุด

```
import matplotlib.pyplot as plt

plt.plot(Max_Features,Accuracy)
plt.show()
```



ภาพประกอบ 38 กราฟแสดงค่า max_features ที่ดีที่สุด

เมื่อหาจำนวนคุณลักษณะที่ดีที่สุดได้แล้ว ผู้ทำวิจัยได้ทำการตรวจสอบว่ามีตัวงานใดบ้างที่มีค่า TF-IDF เท่ากับ 0 และมีอยู่จำนวนเท่าใด โดยนำชุดข้อมูลที่ผ่านการทำความสะอาดแล้วที่อยู่ในตัวแปลประเภท list ที่ชื่อว่า texts_tfidf ซึ่งมีจำนวนข้อความตัวงานอยู่ที่ 11,819 รายการ ผู้ทำวิจัยพบว่ามีจำนวนตัวงานที่เป็น 0 ทั้งหมด 12 รายการ และตัวงานบางรายการที่ถูกทำความสะอาดข้อความจนมีจำนวนคำเท่ากับ 0 อยู่ 2 รายการ (ดังภาพประกอบที่ 39 - 40)

```
tfidf_all = TfidfVectorizer(tokenizer=str.split,use_idf=False,
                            max_features=1800)
tfidf_transformer = tfidf_all.fit_transform(texts_tfidf)
tfidf_plot = pd.DataFrame(tfidf_transformer.toarray(),
                          columns=tfidf_all.get_feature_names())

zero_true=(tfidf_plot.T == 0).all()
zero_weight = pd.DataFrame(zero_true)
zero_weight.loc[zero_weight[0] == True]

select_zero = [524,2245,2249,2255,2526,3390,
               3521,4380,4443,6627,7368,8649]
nlp_Train.iloc[select_zero]
```

ภาพประกอบ 39 หาข้อความตัวงานที่มีน้ำหนักเท่ากับศูนย์

	Title	Target
525	authoriser	7
2246	price sandisk	7
2250	organizational unit	1
2256	ex	4
2527	resignee hafizulbahrun	1
3391		7
3522	procurment	1
4381	updating	4
4444	cat	7
6628	related	1
7369		7
8650	thboardroom	4

ภาพประกอบ 40 แสดงตัวงานที่มีน้ำหนักเท่ากับศูนย์

ต่อไปจะเป็นการสร้างแบบจำลองเพื่อนำเอาชุดข้อมูลดังกล่าวไปทดสอบการทำนาย โดยใช้อัลกอริทึมแบบการสุ่มป่าไม้ (Decision Tree) ซึ่งเป็นแบบจำลองการทำนายการจำแนกข้อมูลแบบกำกับดูแล (Supervised Classification) ที่มีการสร้างการทำนายในรูปแบบของชุดต้นไม้ตัดสินใจหลาย ๆ ต้น (Ensemble of Decision Trees) มาช่วยในการทำนายผลลัพธ์ โดยต้นไม้ตัดสินใจ แต่ละต้นถูกสร้างเป็นอิสระต่อกัน โดยใช้ข้อมูลย่อยที่ถูกสุ่มเลือกจากชุดข้อมูลฝึกสอนเดิม โดยการสุ่มเลือกเป็นแบบ bootstrap (Random Sampling ith Replacement) นั้นคือ อาจมีข้อมูลที่ถูกเลือกซ้ำได้ ทำให้ประสิทธิภาพในการทำนายสูง ซึ่งเขียนด้วยภาษาไพทอน (ดังภาพประกอบที่ 41)

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=0, max_features=3)
rf.fit(tfidf_vsm_x_train, y_train)
rf_predict = rf.predict(tfidf_vsm_x_test)
```

ภาพประกอบ 41 สร้างแบบจำลองในการทำนาย

อธิบายค่าพารามิเตอร์ของแบบจำลอง

(a) n_estimatore = 100 คือจำนวนต้นไม้ตัดสินใจ (Decision tree) ที่ใช้ในป่านี้

(b) random_state = 0 การกำหนดตัวเลขในการสุ่มข้อมูล

(c) max_features = 3 คือการกำหนดจำนวนคุณลักษณะในการทดสอบในแต่ละต้นไม้

3.4.2 **การประเมินสมรรถนะแบบจำลอง** (Confusion Matrix)

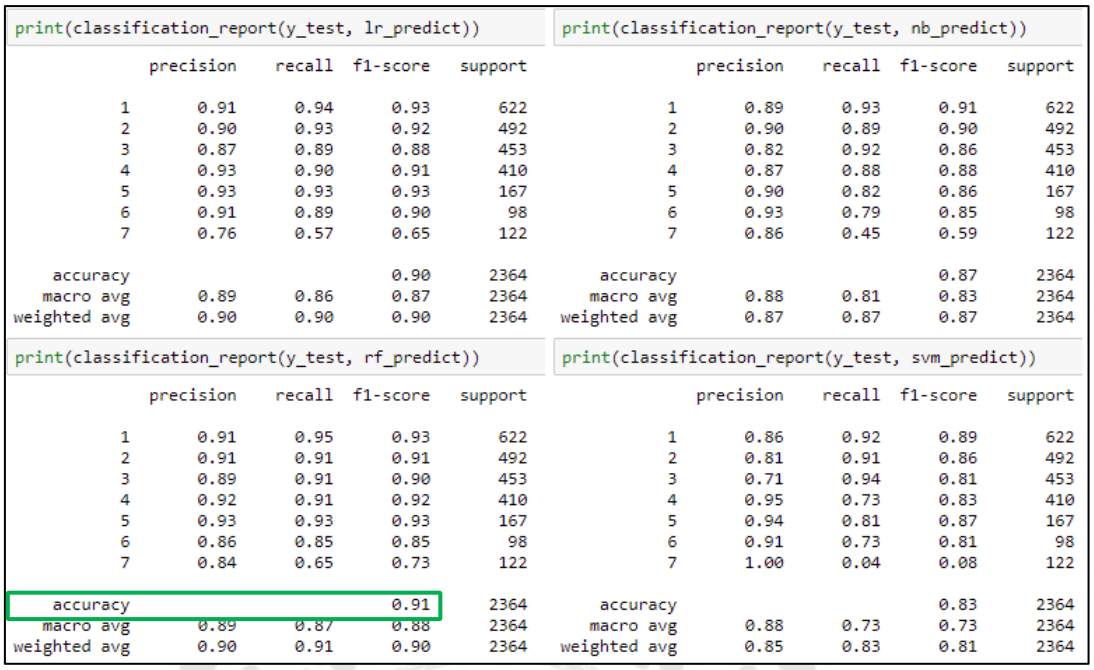
การประเมินผลลัพธ์การทำนายหรือผลลัพธ์จากโปรแกรมเปรียบเทียบกับผลลัพธ์จริงๆ ผู้ทำวิจัยใช้ภาษาไพธอนในการประเมินสมรรถนะของแบบแบบจำลอง (ดังภาพประกอบที่ 42)

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
rf_accuracy = accuracy_score(y_test, rf_predict)
rf_precision = precision_score(y_test, rf_predict, average=None)
rf_recall = recall_score(y_test, rf_predict, average=None)
rf_f1 = f1_score(y_test, rf_predict, average=None)
```

ภาพประกอบ 42 แบบจำลองในการประเมินผลลัพธ์การทำนาย

3.4.3 **ผลการทดสอบการจำแนกตั๋วงาน**

ในงานวิจัยนี้ได้ทำการทดสอบกับแบบจำลองหลายตัวเพื่อนำมาเปรียบเทียบผลว่าแบบจำลองของอัลกริทึมใดให้ผลลัพธ์ที่ดีที่สุด แบบจำลองที่นำมาทดสอบคือ linear regression ได้ผลลัพธ์ความถูกต้องเท่ากับ 90% , Naive Bayes ได้ผลลัพธ์ความถูกต้องเท่ากับ 87% และ Support-vector machine ได้ผลลัพธ์ความถูกต้องเท่ากับ 83% และแบบจำลอง Random Forest ความถูกต้องในการจำแนกตั๋วงานได้ดีถึง 91% และแสดงรูปแบบของตาราง Confusion Matrix (ดังภาพประกอบ 43)



```
print(classification_report(y_test, lr_predict))
```

	precision	recall	f1-score	support
1	0.91	0.94	0.93	622
2	0.90	0.93	0.92	492
3	0.87	0.89	0.88	453
4	0.93	0.90	0.91	410
5	0.93	0.93	0.93	167
6	0.91	0.89	0.90	98
7	0.76	0.57	0.65	122
accuracy			0.90	2364
macro avg	0.89	0.86	0.87	2364
weighted avg	0.90	0.90	0.90	2364

```
print(classification_report(y_test, nb_predict))
```

	precision	recall	f1-score	support
1	0.89	0.93	0.91	622
2	0.90	0.89	0.90	492
3	0.82	0.92	0.86	453
4	0.87	0.88	0.88	410
5	0.90	0.82	0.86	167
6	0.93	0.79	0.85	98
7	0.86	0.45	0.59	122
accuracy			0.87	2364
macro avg	0.88	0.81	0.83	2364
weighted avg	0.87	0.87	0.87	2364

```
print(classification_report(y_test, rf_predict))
```

	precision	recall	f1-score	support
1	0.91	0.95	0.93	622
2	0.91	0.91	0.91	492
3	0.89	0.91	0.90	453
4	0.92	0.91	0.92	410
5	0.93	0.93	0.93	167
6	0.86	0.85	0.85	98
7	0.84	0.65	0.73	122
accuracy			0.91	2364
macro avg	0.89	0.87	0.88	2364
weighted avg	0.90	0.91	0.90	2364

```
print(classification_report(y_test, svm_predict))
```

	precision	recall	f1-score	support
1	0.86	0.92	0.89	622
2	0.81	0.91	0.86	492
3	0.71	0.94	0.81	453
4	0.95	0.73	0.83	410
5	0.94	0.81	0.87	167
6	0.91	0.73	0.81	98
7	1.00	0.04	0.08	122
accuracy			0.83	2364
macro avg	0.88	0.73	0.73	2364
weighted avg	0.85	0.83	0.81	2364

ภาพประกอบ 43 แสดงผลการทำงานในรูปแบบตาราง Confusion Matrix

ผู้ทำวิจัยได้ทดสอบแบบจำลองกับข้อความที่คลุมเครือจำนวน 29 รายการ โดยการดึงเอารายการดังกล่าวจากหมายเลขตั๋วงาน (index) ของรายการนั้นๆมาทดสอบ (ดังภาพประกอบที่ 44) และได้ผลลัพธ์การทำนาย (ดังตารางที่ 7)

```
index_1 = [166,668,1556,1967,2035,2077,2255,3199,3246,
           3418,3847,4450,5109,5314,5401,5698,5732,6087,
           6709,6854,7415,7990,8186,8243,8488,8969,9131,10776,11131]
for x in index_1:
    get_tck = nlp_Train['Title'].iloc[x]
    cvert_tck = pd.Series(get_tck, dtype="string")
    ticket = tfidf_vectorizer.transform(cvert_tck)
    predicted = rf.predict(ticket)

    print(get_tck)
    print(predicted)
```

ภาพประกอบ 44 ทดสอบการทำนายข้อความตั๋วงานที่คลุมเคลือ

ตาราง 7 แสดงผลการทดสอบการทำนายข้อความตั๋วงานที่คลุมเคลือ

Index	ข้อความตั๋วงานจริง	ข้อความผ่าน NLP	ผลลัพธ์จริง	ทำนาย	ผลลัพธ์
669	เปิดเครื่องไม่ติด	เครื่อง ติด	2	2	ถูกต้อง
2256	EX เปิดไม่ได้ค่ะ	ex	4	1	ไม่ถูก
5699	โปรแกรมค้า	โปรแกรม ค้า	4	4	ถูกต้อง
6855	URGENT	urgent	6	7	ไม่ถูก
7991	compu setup	setup	3	5	ไม่ถูก
8244	Picture not working	picture working	3	3	ถูกต้อง

จะเห็นได้ว่าข้อความตั๋วงานที่คลุมเคลือทั้งหมด 29 รายการ โดยแบบจำลองสามารถทายถูก 26รายการ ทำนายไม่ถูก 3 รายการ (ดังภาพประกอบที่ 45)

Index(2256)		Index(6855)		Index(7991)	
able	0.0	urgent	1.0	setup	1.0
servers	0.0	able	0.0	able	0.0
sawangkaew	0.0	server	0.0	sale	0.0
save	0.0	save	0.0	sawangkaew	0.0
satidpitakul	0.0	satidpitakul	0.0	save	0.0
sata	0.0	sata	0.0	satidpitakul	0.0
saraburi	0.0	saraburi	0.0	sata	0.0

ภาพประกอบ 45 แสดงน้ำหนักของคำของตั๋วงานที่คลุมเคลือและทำนายไม่ถูก

3.4.3.1 อธิบายตั๋วงานที่คลุมเคลือโดยแบบจำลองไม่สามารถแยกประเภทได้

- Index [2256] ตั๋วงานนี้จัดอยู่ในประเภทที่ 4 คือ “Office365” แต่แบบจำลองทำนายว่าอยู่ในประเภทที่ 1 คือ “Identity” เมื่อตรวจสอบดูจะไม่พบคุณลักษณะที่แบบจำลองนี้สามารถทำนายตั๋วงานนี้ได้ นั้นคือ ไม่มีคำที่มีน้ำหนัก (Weight) เพื่อนำไปใช้ในการทำนาย

- Index [6855] ตั๋วงานนี้จัดอยู่ในประเภทที่ 6 คือ “Network” แต่แบบจำลองทำนายว่าอยู่ในประเภทที่7 คือ “OutOfScope” เมื่อตรวจสอบดูพบคำว่า Urgent ซึ่งเป็นคุณลักษณะที่ไม่บ่งบอกถึง Network เลย

- Index [7991] ตั๋วงานนี้จัดอยู่ในประเภทที่ 3 คือ “Application” แต่แบบจำลองทำนายว่าอยู่ในประเภทที่ 5 คือ “Printer” เมื่อตรวจสอบดูพบคำว่า Set ซึ่งเป็นคุณลักษณะที่ไม่บ่งบอกถึง Application เลย

ข้อความตั๋วงานที่คลุมเคลือทั้งหมด 29 รายการ โดยแบบจำลองสามารถทายถูก 26 รายการ และยกตัวอย่างมาให้ดู 3 รายการ (ดังภาพประกอบที่ 46)

	Index(669)		Index(5699)		Index(8244)
ติด	0.785284	ค้า	0.781053	picture	0.851215
เครื่อง	0.619136	โปรแกรม	0.624465	working	0.524817
able	0.000000	able	0.000000	able	0.000000
sawangkaew	0.000000	saksit	0.000000	saksit	0.000000
satidpitakul	0.000000	save	0.000000	save	0.000000
sata	0.000000	satidpitakul	0.000000	satidpitakul	0.000000
saraburi	0.000000	sata	0.000000	sata	0.000000

ภาพประกอบ 46 แสดงน้ำหนักของคำของตั๋วงานที่คลุมเคลือแต่ทำนายถูก

3.4.3.2 อธิบายตั๋วงานที่คลุมเคลือแต่แบบจำลองสามารถแยกประเภทได้

- Index [669] ตั๋วงานนี้จัดอยู่ในประเภทที่ 2 คือ “Computer” และแบบจำลองถูกต้อง เมื่อตรวจสอบจะพบคุณลักษณะที่มีน้ำหนัก (Weight) ที่บ่งบอกถึงประเภทของตั๋วงาน

- Index [5699] ตั๋วงานนี้จัดอยู่ในประเภทที่ 4 คือ “Offce365” และแบบจำลองถูกต้อง เมื่อตรวจสอบจะพบคุณลักษณะที่มีน้ำหนัก (Weight) ที่บ่งบอกถึงประเภทของตั๋วงาน

- Index [8244] ตั๋วงานนี้จัดอยู่ในประเภทที่ 3 คือ “Application” และแบบจำลองถูกต้อง เมื่อตรวจสอบจะพบคุณลักษณะที่มีน้ำหนัก (Weight) ที่บ่งบอกถึงประเภทของตั๋วงาน

บทที่ 4
ผลการดำเนินการวิจัย

ในการวิจัยเพื่อประยุกต์ใช้เทคโนโลยีสารสนเทศในการจำแนกประเภทของตั๋วงาน จากการทดลอง ผู้ทำวิจัยได้วิเคราะห์กระบวนการและรายละเอียดในทุกขั้นตอนในการทำวิจัยแล้วพบว่า กระบวนการและการจัดการชุดข้อมูลมีผลต่อประสิทธิภาพการทำนายอยู่พอสมควร เพื่อให้ทราบถึงความสำคัญของกระบวนการ ผู้ทำวิจัยจึงได้ชี้แจงรายละเอียดตามหัวข้อดังนี้

4.1. การลบคำไม่สำคัญ (Stopword)

4.2. การตัดคำด้วย PyThaiNLP

4.3. การทำความสะอาดข้อมูลด้วย Regular Expression

4.1 การลบคำเฉพาะ (Unique Term)

ข้อความของตั๋วงานในชุดข้อมูลนี้จะมีจะมีคำเฉพาะ (Unique Term) อยู่ 2 ประเภท คือ ชื่อบุคคลที่เป็นภาษาไทย ภาษาอังกฤษ เช่น “Chalermchai Pidej”, “เฉลิมชัย พิเดช” และชื่อคอมพิวเตอร์ เช่น “OCS-THNB5519779” เป็นต้น ผู้ทำวิจัยได้ทดลองนำชื่อเหล่านี้ออกจากข้อความของตั๋วงานก่อนจะนำไปตัดคำ (Tokenize) เพื่อลดจำนวนของคำก่อนนำไปประมวลผล อีกอย่างหนึ่งคือคำเฉพาะเหล่านี้จะไม่มีผลต่อการนำไปประมวลผล จากการเปรียบเทียบผลการทดลอง ผลลัพธ์ที่ได้ไม่แตกต่างกัน เป็นเพราะว่าในขั้นตอนของการหาคำสำคัญ (TF-IDF) ผู้ทำวิจัยได้กำหนดค่าพารามิเตอร์ (Parameter) ที่ชื่อว่า max_features อยู่ที่ 1,800 ซึ่งเป็นจำนวนคุณลักษณะที่เหมาะสมที่สุดเพื่อใช้ในการทำนาย

4.2 การตัดคำด้วย PyThaiNLP

จากการวิเคราะห์โครงสร้างของแต่ภาษาไทยและภาษาอังกฤษพบว่า โครงสร้างภาษาอังกฤษนั้นจะมีรูปแบบเป็นที่คำอยู่แล้ว โดยการตัดคำในภาษาอังกฤษจะใช้ย่อหน้าประโยค คำเดี่ยว ๆ หรือช่องว่างระหว่างคำ (space) ในการตัดคำ ทำให้การตัดคำในข้อความตั๋วงานที่เขียนด้วยภาษาอังกฤษอย่างเดียวจะผลลัพธ์ของคำที่ออกมาตามดิกชันนารี แต่เมื่อผู้ใช้งานได้เขียนข้อความในตั๋วงานที่มีคำภาษาไทยอยู่ด้วยหรือเขียนเป็นภาษาไทยทั้งหมด การตัดคำด้วยไลบรารี NLTK จึงได้ผลลัพธ์ของคำภาษาไทยออกมาไม่ค่อยดี ผู้วิจัยจึงเลือกทำการตัดคำโดยใช้ PyThaiNLP ซึ่งผลลัพธ์ที่ได้จะดีกว่า NLTK (ดังภาพประกอบที่ 47) และเมื่อนำชุดข้อมูลที่ตัดคำ

ด้วย NLTK ไปทดสอบกับแบบจำลองพบว่าค่าความถูกต้อง (Accuracy) ลดลงเหลือ 90% (ดังภาพประกอบที่ 48)

```
from pythainlp import word_tokenize
preprocess_raw_text(Chp41.to_string())
```

```
'title         excel     export    ออกจาก ระบบ    pcs    costing    target
```

```
from nltk.tokenize import word_tokenize
preprocess_raw_text(Chp41.to_string())
```

```
'title ไม่เปิด excel export ออกจากระบบ pcs costing ได้target'
```

ภาพประกอบ 47 เปรียบเทียบการตัดคำระหว่าง PyThaiNLP กับ NLTK

```
print(classification_report(y_test, rf_predict))
```

	precision	recall	f1-score	support
1	0.88	0.95	0.91	622
2	0.92	0.90	0.91	492
3	0.90	0.89	0.90	453
4	0.90	0.90	0.90	410
5	0.95	0.93	0.94	167
6	0.84	0.86	0.85	98
7	0.86	0.64	0.73	122
accuracy			0.90	2364
macro avg	0.89	0.87	0.88	2364
weighted avg	0.90	0.90	0.90	2364

ภาพประกอบ 48 ผลลัพธ์การตัดคำด้วย NLTK

จากการตรวจสอบ NLTK เป็น Library ที่ไม่เหมาะกับข้อความที่มีภาษาไทยปนอยู่ด้วย เป็นเพราะว่าโครงสร้างของภาษาไทยถูกเขียนติดกันทั้งประโยค แต่หลักการทำงานพื้นฐานของ NLTK เป็นการตัดคำจากช่องว่าง (Space) คล้ายกับการใช้คำสั่ง str.split() ในภาษาไพธอน จึงไม่สามารถทำการตัดประโยคภาษาไทยได้ ทำให้ได้คุณลักษณะที่เป็นคำไทยที่จะนำไปทดสอบกับแบบจำลองที่เป็นออกมาในรูปแบบของประโยคเกือบทั้งหมด (ดังภาพประกอบที่ 49)

```
tfidf_all = TfidfVectorizer(tokenizer=str.split,use_idf=False, max_features=1800)
tfidf_transformer = tfidf_all.fit_transform(texts_tfidf)
tfidf_plot = pd.DataFrame(tfidf_transformer.toarray(), columns=tfidf_all.get_feature_names())
tfidf_plot.iloc[3369:3379]
```

เข้า ครื่อง	เข้า ใช้ งาน	เข้า ไดร์	เครสมี ปัญหา ไม่ได้ รับ อีเมล์ จากคู่ คำ	เครื่อง	เครื่อง คุณ	เครื่อง ช้ำ	เจอ ปัญหา ดึง ข้อมูล จาก คอม พิว	เช็ค	เช็ค โน้ ตบุ๊ค	เช็ต	เช็ต ปริ้นเตอร์ ชั้น	เช็ ตอี เมลก ลาง	เปลี่ยนชื่อผู้ เข้าใช้งาน เครื่อง คอมพิวเตอร์ แทนพนัง งานที่ลา ออก	เปิด เครื่อง ไม่ติด	เปิด ใช้ งาน ไม่	เปิด ไฟล์
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ภาพประกอบ 49 คุณลักษณะที่ได้จากการตัดคำโดย NLTK

4.3 การทำความสะอาดข้อมูลด้วย Regular Expression

กระบวนการทำความสะอาดข้อมูลด้วย Regular Expression ช่วยให้เรากรองเอาอักษรหรือสัญลักษณ์เราไม่ต้องการหรือไม่มีผลต่อการนำไปใช้ออกไป ในงานวิจัยนี้ผู้ทำวิจัยได้ทดลองกรองคำในชุดข้อมูลให้ออกมาได้สมบูรณ์ที่สุด โดยคาดว่าจะช่วยให้ได้ผลลัพธ์การทำนายได้ดียิ่งขึ้น โดยจากเดิมผู้ทำวิจัยได้ใช้ชุด RegEx ดังนี้ [^\u0E00-\u0E7Fa-zA-Z']|['][] มีความหมายว่า ให้กรองคำที่ไม่ใช่อักษรและสระภาษาไทยและภาษาอังกฤษตัวพิมพ์เล็กและพิมพ์ใหญ่ หรือเครื่องหมายอัญประกาศ หรือ เครื่องหมายเส้นตรง ออกจากข้อความตั้งงาน ผลลัพธ์ที่ได้ (ดังภาพประกอบที่ 50)

```
nlp_Train.iloc[2031:2041]
```

	Title	Target
2032	set printer ชลบุรี ออฟฟิศ ละมุง พี่	5
2033	ปัญหา สแกน งาน ไฟ ล งาน มายัง อีเมล	4
2034	ต้องการ ถ่ายโอน ข้อมูล โน้ตบุค เครื่อง เครื่อง	4

ภาพประกอบ 50 ผลลัพธ์การกรองข้อความโดยใช้ RegEx

เมื่อผู้วิจัยได้ลองเลือกข้อความตั๋วงานที่มีภาษาไทยในข้อความจะพบว่า การตัดคำของ PyThaiNLP ยังไม่สมบูรณ์ โดยจะเห็นว่ามีคำที่มี 1 และ 2 ตัวอักษร หรือ สระกับตัวอักษร อยู่เป็นจำนวนมาก ผู้วิจัยจึงเลือกใช้ (ดังภาพประกอบที่ 51)

```
480                                                    เด.รส.มีปัญหา.อีเมล.คู่.ค้า
714                                  .....เอกสาร.ปลดล็อก.....ไซ.ด.ตะ...จุฬาฯ
846                                                    ล็อกอิน.คอม.พิว.เต.อร
940                                 เอกสาร.ตรวจ.อาชญากรรม.....ได.รฟ.....หาย
1002                                                   ได.รเวอร...ป.ริ้น.เต.อร.....
1839                          ขอยืม.โน๊ต.บุ.ค...สำหรับ....วันที่....ตค...จำนวน....เครื่อง
2032                                                   .....ชลบุรี...ออฟฟิศ.ละมุง.พี่
2033                                                   ปัญหา.สแกน.งาน...ไฟ.ล.งาน.มายัง.อีเมล
2034                                 ต้องการ.ถ่ายโอน.ข้อมูล.โน้ตบุค.เครื่อง.เครื่อง
2042    ปริ้น.บัตร.พนักงาน.โปรแกรม.คอม.พิวปริ้น.สี่.....ดึง.ข้อมูล.....คอม.พิว.ตัว...
Name: Title, dtype: object
```

ภาพประกอบ 51 จำนวนคำที่ไม่สมบูรณ์ 1 และ 2 ตัวอักษร หรือ สระกับตัวอักษร

ผู้ทำวิจัยจึงทดลองใช้ RegEx ในการกรองคำดังกล่าวออก และทำการทดสอบกับแบบจำลองเพื่อดูว่าผลลัพธ์การทำนายดีขึ้นหรือไม่ โดยเขียนได้ RegEx เพิ่ม(ดังภาพประกอบที่ 52 และ (ตารางที่ 8) อธิบาย RegEx ของแต่ละบรรทัด

```
regEx1 = re.sub(r"[^\u0E00-\u0E7Fa-zA-Z' ]|['][|]", '', cleaned_words)
regEx2 = re.sub(r"\s[a-z][a-z]\s|\s[a-z]\s", ' ', regEx1)
regEx3 = re.sub(r"\s[a-z][a-z]\s|\s[a-z]\s", ' ', regEx2)
regEx4 = re.sub(r"\A[a-z][a-z]\s|\A[a-z]\s", ' ', regEx3)
regEx5 = re.sub(r"\A([a-z]{2})\s", ' ', regEx4)
regEx6 = re.sub(r"\s([a-z]{2})\Z", ' ', regEx5)
regEx7 = re.sub(r"\s[ก-ฯ][ก-ฯ]\s|\s[ก-ฯ]\s", ' ', regEx6)
regEx8 = re.sub(r"\s[ก-ฯ][ก-ฯ]\s|\s[ก-ฯ]\s", ' ', regEx7)
regEx9 = re.sub(r"\s[ก-ฯ][ก-ฯ]\s|\s[ก-ฯ]\s", ' ', regEx8)
regEx10 = re.sub(r"\A([ก-ฯ]{2})\s", ' ', regEx9)
regEx11 = re.sub(r"\s(.[ก-ฯ]*?)\Z", ' ', regEx10)
```

ภาพประกอบ 52 RegEx เพื่อกรองจำนวนคำที่ไม่สมบูรณ์

ตาราง 8 อธิบาย RegEx ของแต่ละบรรทัด

ตัวแปล	ความหมายของ RegEx
regEx1	กรองอักขระพิเศษทั้งหมดออกจากข้อความตั๋วงาน
regEx2	กรองคำภาษาอังกฤษที่ไม่สมบูรณ์ระหว่างประโยคออกครั้งที่ 1
regEx3	กรองคำภาษาอังกฤษที่ไม่สมบูรณ์ระหว่างประโยคออกครั้งที่ 2
regEx4	กรองคำภาษาอังกฤษไม่สมบูรณ์หน้าประโยคออก
regEx5	กรองคำภาษาอังกฤษไม่สมบูรณ์หน้าประโยคออก
regEx6	กรองคำภาษาอังกฤษไม่สมบูรณ์ท้ายประโยคออก
regEx7	กรองคำภาษาไทยที่ไม่สมบูรณ์ระหว่างประโยคออกครั้งที่ 1
regEx8	กรองคำภาษาไทยที่ไม่สมบูรณ์ระหว่างประโยคออกครั้งที่ 2
regEx9	กรองคำภาษาไทยที่ไม่สมบูรณ์ระหว่างประโยคออกครั้งที่ 3
regEx10	กรองคำภาษาไทยไม่สมบูรณ์หน้าประโยคออก
regEx11	กรองคำภาษาไทยไม่สมบูรณ์ท้ายประโยคออก

สำหรับภาษาอังกฤษที่ถูกตัดคำแล้วมีคำที่มี 1 และ 2 ตัวอักษร ก็ทำเช่นเดียวกัน จากนั้นนำชุดข้อมูลที่ได้จากการลบคำไม่สมบูรณ์ไปทดสอบกับแบบจำลองการทำนาย ซึ่งผลลัพธ์ของการทำความสะอาดข้อมูลนี้แสดงผล (ดังภาพประกอบที่ 53) พร้อมทั้งผลลัพธ์การทำนาย (ดังภาพประกอบที่ 54)

```
480                                                    .มีปัญหา.อีเมล์.คู่.
714                                          ....เอกสาร.ปลดล็อก.....
846                                                    ล็อกอิน.
940                                    เอกสาร.ตรวจ.อาชญากรรม.....ไดร์....
1002                                                   ไดร์.เวอร์...รื้น.....
1839                                ขอยืม.โน๊ต...สำหรับ...อบรม...วันที่.....จำนวน.....
2032                                          .....ชลบุรี...ออฟฟิศ.ละมุง.
```

ภาพประกอบ 53 ผลลัพธ์ของการทำความสะอาดข้อมูลโดย RegEx

```
print(classification_report(y_test, rf_predict))
```

	precision	recall	f1-score	support
1	0.91	0.94	0.93	622
2	0.90	0.92	0.91	492
3	0.90	0.91	0.90	453
4	0.91	0.90	0.90	410
5	0.92	0.92	0.92	167
6	0.89	0.88	0.88	98
7	0.84	0.63	0.72	122
accuracy			0.90	2364
macro avg	0.89	0.87	0.88	2364
weighted avg	0.90	0.90	0.90	2364

ภาพประกอบ 54 ผลการทำนายจากการทำความสะอาดข้อมูลโดย RegEx

นอกเหนือจากขอบเขตของงานวิจัย ผู้ทำวิจัยได้ทำการลองทดสอบประสิทธิภาพของแบบจำลองเพิ่มเติมด้วยเทคนิคที่ชื่อว่า Cross validation โดยใช้ไลบรารี่ชื่อ cross_validate ซึ่งเป็นการแบ่งชุดของมูลออกเป็นส่วนย่อย ๆ (Fold) โดยชุดข้อมูลที่ถูกแบ่งจะสามารถกำหนดจำนวนครั้งในการทดสอบได้ (k) เมื่อทำการทดสอบแล้วจะได้ค่าความถูกต้องในแต่ละรอบออกมาแล้วก็จะนำมาหาค่าเฉลี่ย โดยการทดสอบนี้ได้ค่าเฉลี่ยความถูกต้องอยู่ที่ 0.89540 (ดังภาพประกอบที่ 55)

```
from sklearn.model_selection import cross_validate
from sklearn import datasets
from sklearn.ensemble import RandomForestClassifier

rf_Train = RandomForestClassifier(n_estimators=100, random_state=0, max_features=3)
cvAcc = cross_validate(rf_Train, tfidf_vsm_x_train, y_train, cv=10)

cvAcc = cvTrain['test_score'].mean()

print("Accuracy of Training set: " '%.5f'%(cvAcc))

Accuracy of Training set: 0.89540
```

ภาพประกอบ 55 วัดประสิทธิภาพแบบจำลองด้วยเทคนิค Cross Validation

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

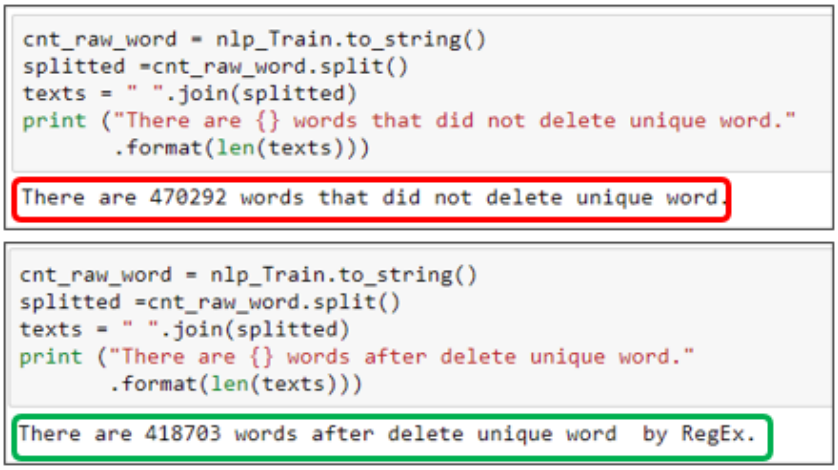
ในการวิจัยเพื่อประยุกต์ใช้เทคโนโลยีสารสนเทศในการจำแนกประเภทของตั๋วงาน จากการทดลองผู้ทำวิจัยได้สรุปเป็นหัวข้อได้ดังนี้ครับ

1. สรุปผลการวิจัย
2. อภิปรายผล
3. ข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้เป็นการจำแนกประเภทของตั๋วงานด้วยกระบวนการประมวลผลภาษาธรรมชาติ ร่วมกับแบบจำลองการทำนายโดยใช้อัลกอริทึมแบบสุ่มป่าไม้แบบแยกประเภทด้วยเทคนิคมาตรฐาน (Standard Process) และได้ผลลัพธ์การทำนายที่ดี ผู้ทำวิจัยจึงได้วิเคราะห์และหากระบวนการใหม่เพื่อให้ได้ผลการทำนายที่ดีขึ้น โดยผลการวิจัยสรุปได้ 3 วิธีดังนี้

วิธีที่ 1 ลบคำเฉพาะ อย่างที่กล่าวไว้ในบทก่อนหน้านี้ ชุดข้อมูลนี้ประกอบไปด้วยคำเฉพาะเช่น ชื่อบุคคล ที่เป็นภาษาไทยหรือภาษาอังกฤษ และ ชื่อคอมพิวเตอร์ โดยคำเหล่านี้เป็นคำที่ไม่สื่อความหมายและไม่สามารถนำใช้กระบวนการประมวลผลภาษาธรรมชาติในการตัดคำได้อย่างเหมาะสม คำเหล่านี้ก่อนทำการคัดกรองออกจะทำให้ได้จำนวนคำมากถึง 470,292 คำ เลยทีเดียวเชียว และเมื่อทำการนำคำเหล่านี้ออกไปแล้ว จะเหลืออยู่ที่ 418,703 คำ โดยลดลงไปจำนวน 51,589 คำ (ดังภาพประกอบที่ 56)



ภาพประกอบ 56 จำนวนคำที่เหลือจากการลบคำเฉพาะ

วิธีที่ 2 การตัดคำด้วย PyThaiNLP จากบทที่แล้ว จะได้ผลลัพธ์การทำนายได้ดีขึ้นมาอีก 1% (ดังภาพประกอบที่ 57)

```
print(classification_report(y_test, rf_predict))
```

	precision	recall	f1-score	support
1	0.92	0.94	0.93	622
2	0.91	0.92	0.91	492
3	0.91	0.91	0.91	453
4	0.90	0.91	0.91	410
5	0.94	0.93	0.94	167
6	0.85	0.87	0.86	98
7	0.84	0.66	0.74	122
accuracy			0.91	2364
macro avg	0.89	0.88	0.88	2364
weighted avg	0.91	0.91	0.91	2364

ภาพประกอบ 57 ผลการทำนายจากการตัดคำ

วิธีที่ 3 Regular Expression การกรองคำที่ไม่สมบูรณ์เช่นคำที่มี 1 หรือ 2 ตัวอักศร หรือ สระกับตัวอักศรในภาษาไทย และ มี 1 หรือ 2 ตัวอักศรในภาษาอังกฤษ โดยก่อนที่จะลบคำไม่สมบูรณ์ออก จำนวนคำที่ได้จากการตัดคำออกเฉพาะออกแล้วจะเหลือเท่ากับ 418,703 คำ และหลังจากการกรองแล้วจะเหลืออยู่ 381,602 คำ โดยคำไม่สมบูรณ์ถูกกรองไปเป็นจำนวน 37,101 คำ (ดังภาพประกอบที่ 58)

```
cnt_raw_word = nlp_Train.to_string()
splitted =cnt_raw_word.split()
texts = " ".join(splitted)
print ("There are {} words are not filtered by RegEx."
       .format(len(texts)))
```

```
There are 418703 words are not filtered by RegEx.
```

```
cnt_raw_word = nlp_Train.to_string()
splitted =cnt_raw_word.split()
texts = " ".join(splitted)
print ("There are {} words were filtered by RegEx."
       .format(len(texts)))
```

```
There are 381602 words were filtered by RegEx.
```

ภาพประกอบ 58 เปรียบเทียบจำนวนคำก่อนหลังการกรองโดย RegEx

5.2 อภิปรายผล

5.2.1 การลบคำเฉพาะ

เป็นการช่วยลดจำนวนคำก่อนนำไปหาน้ำหนักของคำด้วยวิธี TF-IDF เทคนิคนี้เป็นเทคนิคแรกที่นำมาใช้ เป็นการลดจำนวนคำที่ไม่มีผลต่อการทำนาย และเมื่อผู้วิจัยได้ทำการปรับค่าพารามิเตอร์ของ TF-IDF ที่ชื่อว่า max_features ให้เป็น 1,800 แล้ว การลบคำเฉพาะนี้ก็ช่วยให้ผลลัพธ์ของการทำนายดีขึ้น ซึ่งเป็นเพราะว่า TF-IDF ได้เลือกเอาคุณลักษณะจากคำที่มีค่าน้ำหนักสูงสุด 1,800 คุณลักษณะมาใช้นั่นเอง

5.2.2 การตัดคำด้วย PyThaiNLP

ผู้ทำวิจัยใช้วิธีการตัดคำเพื่อให้ได้คำที่สมบูรณ์ที่สุด เนื่องจากชุดข้อมูลที่นำมาใช้ในงานวิจัยคือข้อความของตั๋วงานที่เป็นภาษาไทย หรือ ภาษาอังกฤษ และ ภาษาไทยผสมภาษาอังกฤษ จึงเป็นเรื่องที่ท้าทายอย่างมากในการหาเทคนิคที่มีประสิทธิภาพในการจัดการกับชุดข้อมูลนี้ ข้อแรกที่จะต้องพิจารณา คือต้องทำการศึกษาโครงสร้างของภาษาเพื่อนำมาวิเคราะห์หาเครื่องมือและเทคนิคที่เหมาะสม โดยภาษาไทยจะมีเครื่องมือชื่อว่า PyThaiNLP และภาษาอังกฤษจะมีเครื่องมือชื่อว่า NLTK แต่สิ่งที่สำคัญที่สุดในงานวิจัยนี้คือ ผู้วิจัยจะทำอย่างไรที่จะนำเครื่องมือเหล่านี้มาใช้จัดการกับชุดข้อมูลตั๋วงานอย่างเหมาะสมและได้มีประสิทธิภาพที่สุด จากผลการทดลองพบว่าผู้ทำวิจัยได้เลือก PyThaiNLP โดยใช้ newmm engine เป็นการตัดคำในขั้นตอนแรก และใช้ thai_stopwords ในการลบคำภาษาไทยที่ไม่สำคัญออก จากนั้นใช้ NLTK Corpus ในการลบคำภาษาอังกฤษที่ไม่สำคัญออก และกำหนดจำนวนคุณลักษณะจากการหาคำสำคัญของชุดข้อมูลไว้ที่ 1,800 คุณลักษณะ เมื่อนำชุดข้อมูลไปทดสอบการทำนายจึงได้ผลการทำนายสูงถึง 91% ซึ่งเป็นผลลัพธ์ที่ดีที่สุดจากกระบวนการอื่นๆทั้งหมด

5.2.3 การใช้ Regular Expression เพื่อกรองคำที่สมบูรณ์มากที่สุด

ผู้วิจัยได้ทำการทดลองกรองคำและนำชุดข้อมูลที่ได้ไปทดสอบกับแบบจำลองการทำนายแล้วพบว่าได้ผลลัพธ์ลดลงมา 1% เพื่อให้ทราบว่าทำไมเทคนิคนี้ถึงทำให้ผลลัพธ์ลดลงมา จึงได้ทำการวิเคราะห์และพบว่า จำนวนคำที่ไม่สมบูรณ์เช่นคำที่มี 1 หรือ 2 ตัวอักษร หรือ สระกับตัวอักษร ในภาษาไทย และ มี 1 หรือ 2 ตัวอักษร ในภาษาอังกฤษ มีความสำคัญเป็นคุณลักษณะที่มีผลต่อการนำไปใช้ในการทำนาย เป็นเพราะว่าเครื่องมือที่นำมาใช้และเทคนิคยังไม่ดีพอสำหรับการใช้งานกับข้อมูลชุดนี้ (ดังภาพประกอบที่ 59)

Index(1001)		Index(1382)		Index(1666)		Index(2032)		Index(2393)	
ได	0.461016	รอ	0.363945	note	0.380881	งาน	0.700514	จอ	0.796064
รั้น	0.418657	เวลา	0.343909	ใบ	0.374043	สแกน	0.365728	คอม	0.605213
ป	0.404309	waiting	0.330712	pr	0.374043	ไฟ	0.324761	able	0.000000
เค	0.392852	ใบ	0.330712	book	0.362894	ปัญหา	0.316057	sak	0.000000
อร	0.389493	approval	0.325474	battery	0.346581	ล	0.304075	sata	0.000000
ricoh	0.377740	form	0.320854	เอกสาร	0.314158	อีเมล	0.278745	saraburi	0.000000

ภาพประกอบ 59 คำที่มีผลต่อการทำนายของแบบจำลอง

5.3. ข้อเสนอแนะ

จากการที่ผู้ทำวิจัยได้วิเคราะห์และทดลองเทคนิคหลาย ๆ อย่าง ผู้ทำวิจัยได้มีข้อเสนอแนะอยู่ 2 เรื่องด้วยกัน คือ เรื่องของชุดข้อมูลและเครื่องมือในการประมวลผลธรรมชาติภาษาไทยดังนี้

ชุดข้อมูลตั๋วงาน เป็นชุดข้อมูลที่มีลักษณะเฉพาะของตัวมันเอง จากที่ผู้ทำวิจัยได้ทำงานอยู่ในองค์กรที่อนุญาตให้เก็บข้อมูลอย่างเป็นทางการนั้น ผู้ทำวิจัยได้เข้าใจถึงพฤติกรรมของผู้ใช้งานที่เขียนข้อความตั๋วงานในระบบ โดยการเขียนของผู้ใช้งานนั้นไม่ต้องการความถูกต้องแต่เป็นการเขียนแบบสั้นและโดยความเข้าใจของเขาเอง เพื่อที่จะแจ้งให้ผู้สนับสนุนด้านไอทีรีบดำเนินการแก้ไขปัญหาของผู้ใช้ได้เร็วที่สุด สิ่งเหล่านี้จึงเป็นผลให้ข้อความของตั๋วงานมีลักษณะเฉพาะตัวมาก ทำให้เป็นสิ่งที่ท้าทายในการนำเอามาทำงานวิจัยนี้

เครื่องมือการประมวลผลภาษาธรรมชาติ จากที่ได้กล่าวข้างต้นในเรื่องของชุดข้อมูลที่มีลักษณะเฉพาะตัว ผู้ทำวิจัยเล็งเห็นถึงความสำคัญของเทคโนโลยีทางด้านไอทีที่นำมาใช้เพื่อการแก้ปัญหาต่าง ๆ ในโลกปัจจุบัน การพัฒนาเครื่องมือสำหรับการใช้แก้ปัญหาต่างๆเป็นสิ่งสำคัญอย่างมาก PyThaiNLP คือเครื่องมือที่ผู้ทำวิจัยนำมาใช้งานและเห็นว่ามีประโยชน์อย่างมากกับการใช้เป็นเครื่องมือการประมวลผลสำหรับภาษาไทย ด้วยที่ภาษาไทยขอเราคนไทยมีล้ำลึกและมีเสน่ห์ การพัฒนาให้เครื่องมือนี้มีประสิทธิภาพมากยิ่งขึ้นจะทำให้ผู้ที่ทำงานวิจัยเกี่ยวกับเรื่องประมวลผลภาษาไทยในอนาคตสามารถทำให้ประสิทธิภาพการทำนายยอดเยี่ยมมากขึ้นและผลลัพธ์ที่ดีขึ้นอย่างมากมาย

งานวิจัยในอนาคต การวิเคราะห์ชุดตั๋วงานในเชิงลึกและเครื่องมือในการประมวลผลภาษาธรรมชาติที่ถูกพัฒนาขึ้นมาอย่างต่อเนื่อง ผู้ทำวิจัยคาดว่าจะพัฒนาต่อยอดงานวิจัยนี้ให้มีประสิทธิภาพที่ดียิ่งขึ้นอีกในอนาคต

บรรณานุกรม

Al-Hawari, F., & Barham, H. (2019). A machine learning based help desk system for IT service management. *Journal of King Saud University - Computer and Information Sciences*.

Han, J., & Akbari, M. (2018). Vertical Domain Text Classification: Towards Understanding IT Tickets Using Deep Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence,* 32(1).

Li, L., Li, W., & Gong, D. (2019). Naive Bayesian Automatic Classification of Railway Service Complaint Text Based on Eigenvalue Extraction. *Tehnicki Vjesnik - Technical Gazette,* 26, 778+.

Paramesh, S. P., Ramya, C., & Shreedhara, K. S. (2018). Classifying the Unstructured IT Service Desk Tickets Using Ensemble of Classifiers. 2018 3*rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, 221-227.

Paramesh, S. P., & Shreedhara, K. S. (2019). Automated IT Service Desk Systems Using Machine Learning Techniques. *Data Analytics and Learning*, 331-346.

Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications,* 181.

Revina, A., Buza, K., & Meister, V. G. (2020). IT Ticket Classification: The Simpler, the Better. *IEEE Access,* 8, 193380-193395.

S.P, P., & K.S, S. (2019). Building Intelligent Service Desk Systems using AI. *International Journal of Engineering Sciences and Management -A Multidisciplinary Publicationof VTU,* 1, 01-08.

ประวัติผู้เขียน

ชื่อ-สกุล	เฉลิมชัย พิเดช
วัน เดือน ปี เกิด	21 ตุลาคม 2522
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	พ.ศ. 2548 คอมพิวเตอร์ธุรกิจ มหาวิทยาลัยนอร์ทกรุงเทพ
ที่อยู่ปัจจุบัน	100/168 หมู่บ้านเพอร์เฟคเพลส3 ถนนรามคำแหง 174 แขวงมีนบุรี เขตมีนบุรี กรุงเทพมหานคร 10510