

กระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุล
A Method of Imbalanced Sentiment Classification

โครงการปริญญานิพนธ์

ของ

นายพีระวัฒน์ บุญบ้านจั่ว

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2563

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

กระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุล
A Method of Imbalanced Sentiment Classification

โครงการปริญญานิพนธ์

ของ

นายพีระวัฒน์ บุญบ้านจั่ว

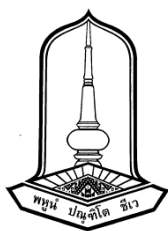
เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2563

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม



คณะกรรมการสอบโครงการปริญญานิพนธ์ ได้พิจารณาปริญญานิพนธ์ของ นายพีระวัฒน์
บุญบ้านจิว แล้วเห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบโครงการปริญญานิพนธ์

ประธานสอบ

.....
(ผู้ช่วยศาสตราจารย์ ดร.ฉัตรเกล้า เจริญผล)

กรรมการ

.....
(อาจารย์ ดร.นัฐธริยา เหล่าประชา)

ที่ปรึกษาโครงการปริญญานิพนธ์หลัก

.....
(ผู้ช่วยศาสตราจารย์ ดร.จันทิมา พลพินิจ)

หลักสูตรวิทยาการคอมพิวเตอร์อนุมัติให้รับโครงการปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการ
การศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการ
สารสนเทศ มหาวิทยาลัยมหาสารคาม

.....
(อาจารย์พระ พฤกษ์ศรี)

.....
(ผู้ช่วยศาสตราจารย์พิมลรัตน์ อ้วนศรีเมือง)

อาจารย์ผู้ประสานงานวิชาโครงการปริญญานิพนธ์

วันที่ 11 เดือน มิถุนายน พ.ศ. 2564

บทคัดย่อ

ชื่อโครงการ	กระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุล
ผู้จัดทำ	61011212107 นายพีระวัฒน์ บุญบ้านจิว
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.จันทิมา พลพินิจ
หลักสูตร	วิทยาศาสตร์บัณฑิต (สาขาวิชาวิทยาการคอมพิวเตอร์)
คณะ	วิทยาการสารสนเทศ
มหาวิทยาลัย	มหาวิทยาลัยมหาสารคาม
ปีที่พิมพ์	2563

การจำแนกความรู้สึก (Sentiment Classification) คือการจำแนกเอกสารตามชั้นความรู้สึกซึ่งโดยทั่วไปอาจจะจำแนกเป็นความรู้สึกที่เป็นบวก (Positive) ความรู้สึกที่เป็นลบ (Negative) และความรู้สึกที่เป็นกลาง (Neutral) โดยการจำแนกความรู้สึกนั้น ได้รับการศึกษามาอย่างต่อเนื่อง เพราะการประยุกต์ใช้ในหลายลักษณะ แต่โดยทั่วไปมักจะนิยมใช้ในการจำแนกความรู้สึกที่มีการแสดงไว้ในรูปแบบข้อความ (Text) เช่น ประยุกต์ใช้ในการจัดอันดับความรู้สึกจากข้อความแสดงความคิดเห็นของผู้คนที่ติดต่อสินค้าและบริการ การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้เรียน การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้คนในเรื่องการเมือง เป็นต้น ซึ่งปัญหาความไม่สมดุลของข้อมูลในคลาสนั้น เกิดจากกลุ่มตัวอย่างที่ใช้ในการเรียนรู้มีข้อมูลไม่สมดุลกัน โดยกลุ่มที่มีข้อมูลมากกว่าจะเรียกว่า “ข้อมูลกลุ่มหลัก (Majority Class)” ขณะที่กลุ่มตัวอย่างที่มีข้อมูลจำนวนน้อยกว่าจะเรียกว่า “ข้อมูลกลุ่มรอง (Minority Class)” เมื่อนำเอาชุดข้อมูลในลักษณะนี้ไปเรียนรู้เพื่อสร้างตัวจำแนกความรู้สึก (Sentiment Classifier) ข้อมูลใหม่ๆ ที่อ่านเข้ามาเพื่อวิเคราะห์เพื่อจำแนกกลุ่มด้วยตัวจำแนกความรู้สึกดังกล่าว ก็มีแนวโน้มที่จะทำนายกลุ่มของข้อมูลนั้นไปยังทิศทางของข้อมูลกลุ่มหลักที่ใช้ในการเรียนรู้ตัวจำแนกความรู้สึก ดังนั้น ในโครงงานปริญญานิพนธ์ฉบับนี้ จึงได้นำเสนอการศึกษาการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึกด้วยเทคนิคการให้น้ำหนักค่า 5 เทคนิค คือ TF-IDF, Delta TF-IDF, TF-IDF-ICF, TF-RF และ TF-IGM ร่วมกับแมชชีนเลิร์นนิง 3 ตัว คือ Naïve Bayes, K-Nearest Neighbor และสุดท้าย Convolution Neural Network

คำสำคัญ: การจำแนกเอกสาร, การให้น้ำหนักค่า, ข้อมูลไม่สมดุล, ซัพพอร์ตเวกเตอร์แมชชีน

กิตติกรรมประกาศ

โครงการปริญญานิพนธ์ฉบับนี้สำเร็จสมบูรณ์ได้ด้วยความกรุณาและความช่วยเหลืออย่างสูงยิ่งจากผู้ช่วยศาสตราจารย์ ดร.ฉัตรเกล้า เจริญผล ประธานกรรมการสอบ และอาจารย์ ดร.นัฐธริยา เหล่า-ประชา กรรมการสอบ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.จันทิมา พลพินิจ ที่ปรึกษาโครงการปริญญานิพนธ์หลัก ที่คอยสั่งสอนให้คำแนะนำ ตรวจสอบแก้ไขข้อบกพร่อง รวมถึงช่วยชี้แนะแนวทางในการค้นคว้าหาความรู้เพื่อนำมาใช้ในโครงการปริญญานิพนธ์นี้ ทางผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์ จากท่านอาจารย์และขอกราบขอบพระคุณไว้เป็นอย่างสูง

ขอขอบพระคุณ บิดา มารดา ตลอดจนผู้ที่เกี่ยวข้องกับทุกท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ ที่ได้ให้กำลังใจและมีส่วนช่วยเหลือให้โครงการปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี และขอขอบคุณอาจารย์ทุกท่านในภาควิชาวิทยาการคอมพิวเตอร์ที่ ได้อบรม สั่งสอน และให้ความรู้ จนผู้จัดทำสามารถนำความรู้ความสามารถในหลายๆ ด้านมาประกอบกันจนเกิดโครงการปริญญานิพนธ์นี้ขึ้น

ท้ายที่สุด คณะผู้จัดทำหวังว่าโครงการปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์กับผู้สนใจไม่มากนัก

พีระวัฒน์ บุญบ้านจิว

สารบัญ

หน้า

บทคัดย่อ	ก
กิตติกรรมประกาศ.....	ข
สารบัญ.....	ค
สารบัญตาราง.....	ฉ
สารบัญภาพประกอบ.....	ช
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน.....	3
1.6 แผนการดำเนินงาน.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 ข้อมูลที่ไม่สมดุล (Imbalanced Data).....	5
2.2 การจำแนกความรู้สึก (Sentiment Classification).....	6
2.3 เทคนิคและอัลกอริทึมที่เกี่ยวข้อง.....	6
2.3.1 การจำแนกหมวดหมู่เอกสาร (Text Classification).....	6
2.3.2 ขั้นตอนการเตรียมเอกสาร (Document Pre-processing).....	7
2.3.3 การสร้างตัวแทนเอกสาร (Document Representation).....	9
2.3.4 การเลือกคุณลักษณะ (Feature Selection).....	9
2.3.5 การให้น้ำหนักคำ (Term Weighting).....	10
2.3.6 นาอ์ฟเบย์ (Naïve Bayes)	14
2.3.7 วิธีการค้นหาเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbor: KNN).....	15
2.3.8 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN)	15
2.3.9 การประเมิน (Evaluation).....	17
2.4 งานวิจัยที่เกี่ยวข้อง (Related work).....	18

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีดำเนินงานวิจัย.....	21
3.1 กรอบการดำเนินงาน.....	21
3.2 ชุดข้อมูล (Data set).....	21
3.3 การสร้างโมเดลเพื่อการจำแนกความรู้สึกของบทวิจารณ์ (Classifier Modeling).....	23
3.3.1 การเตรียมข้อมูลก่อนการประมวลผล.....	23
3.3.2 การสร้างโมเดลการจำแนกความรู้สึกของบทวิจารณ์.....	34
3.4 การวัดประสิทธิภาพของตัวจัดกลุ่มเอกสาร (Evaluation).....	45
3.4.1 การนำโมเดลเพื่อการจำแนกกลุ่มของบทวิจารณ์ไปใช้.....	45
3.4.2 การวัดประสิทธิภาพของตัวจัดกลุ่มเอกสาร (Evaluation).....	56
3.5 การปรับปรุงประสิทธิภาพโมเดลเพื่อการจำแนก.....	57
3.5.1 ปัญหาจากการทำ Lemmatization.....	57
3.5.2 ปัญหาด้านการใช้ภาษา.....	58
3.6 ตัวอย่างหน้าจอโปรแกรม.....	58
บทที่ 4 ผลการทดลอง.....	59
4.1 ข้อมูลที่ใช้ในการทดสอบ.....	59
4.2 Algorithm Setup.....	59
4.2.1 KNN Setup.....	59
4.2.2 Naïve Bayes.....	60
4.2.3 CNN Setup.....	60
4.3 ผลการทดลอง (Results).....	61
4.3.1 การทดสอบโมเดลในการจำแนกบทวิจารณ์โดยอัลกอริทึม KNN.....	61
4.3.2 การทดสอบโมเดลในการจำแนกบทวิจารณ์โดยอัลกอริทึม Naïve Bayes.....	64
4.3.3 การทดสอบโมเดลในการจำแนกบทวิจารณ์โดยอัลกอริทึม CNN.....	67
4.3.4 ภาพรวมผลการทดลอง.....	70
4.4 การทดสอบการจำแนกบทวิจารณ์ที่มีข้อมูลที่ต่างกัน 3 ชุดข้อมูลในทุกสัดส่วน.....	70
4.4.1 ทดสอบโมเดลกับ 3 สัดส่วนด้วยข้อมูล 3 ชุดที่ต่างกับอัลกอริทึม KNN.....	71

สารบัญ (ต่อ)

	หน้า
4.4.2 ทดสอบโมเดลกับ 3 สัตว์ด้วยข้อมูล 3 ชุดที่ต่างกับอัลกอริทึมนาอูฟเบย์	74
4.4.3 ทดสอบโมเดลกับ 3 สัตว์ด้วยข้อมูล 3 ชุดที่ต่างกับ CNN	77
4.5 การวิเคราะห์ผล	80
บทที่ 5 สรุปและอภิปรายผลการทดลอง	84
5.1 สรุปผลและอภิปรายผล	84
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	85
5.2.1 ปัญหาเกี่ยวกับอัลกอริทึมในการสร้างโมเดล	85
5.2.2 ปัญหาเกี่ยวกับชุดข้อมูลที่ใช้ในการสร้างโมเดล	85
5.3 ข้อเสนอแนะ	86
เอกสารอ้างอิง	87
ภาคผนวก	90
ภาคผนวก ก คู่มือการติดตั้ง	91
ภาคผนวก ข คู่มือการใช้งาน	101
บทความวิจัย	105
โปสเตอร์โครงงาน	121
ประวัติผู้จัดทำโครงงาน	123

สารบัญตาราง

	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน	4
ตารางที่ 2.1 แสดงการตัดคำหยุด	7
ตารางที่ 2.2 สัญลักษณ์สำหรับ Supervised Term Weighting (STW).....	11
ตารางที่ 2.3 ตัวอย่างการแจกแจงเอกสารสองเทอม	13
ตารางที่ 2.4 ผลลัพธ์การคำนวณการกระจายน้ำหนัก	14
ตารางที่ 3.1 แสดงการนำเสนอความสัมพันธ์ระหว่างคำและเอกสาร	25
ตารางที่ 3.2 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ <i>tf-idf</i>	29
ตารางที่ 3.3 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ <i>Delta TF-IDF</i>	30
ตารางที่ 3.4 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ <i>TF-IDF-ICF</i>	32
ตารางที่ 3.5 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ <i>TF-RF</i>	33
ตารางที่ 3.6 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ <i>TF-IGM</i>	34
ตารางที่ 3.7 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้ การให้น้ำหนักค่าแบบ <i>tf-idf</i>	37
ตารางที่ 3.8 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้ การให้น้ำหนักค่าแบบ <i>Delta TF-IDF</i>	38
ตารางที่ 3.9 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้ การให้น้ำหนักค่าแบบ <i>TF-IDF-ICF</i>	39
ตารางที่ 3.10 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้ การให้น้ำหนักค่าแบบ <i>TF-RF</i>	40
ตารางที่ 3.11 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้ การให้น้ำหนักค่าแบบ <i>TF-IGM</i>	41
ตารางที่ 3.12 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักค่าด้วย <i>tf-idf</i>	42
ตารางที่ 3.13 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN การให้น้ำหนักค่าด้วย <i>Delta TF-IDF</i>	42
ตารางที่ 3.14 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักค่าด้วย <i>TF-IDF-ICF</i>	43
ตารางที่ 3.15 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักค่าด้วย <i>TF-RF</i>	43
ตารางที่ 3.16 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักค่าด้วย <i>TF-IGM</i> ..	43
ตารางที่ 3.17 แสดงค่าสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ NV.....	46

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 3.18 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-IDF.....	49
ตารางที่ 3.19 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ Delta TF-IDF	50
ตารางที่ 3.20 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-IDF-ICF ..	52
ตารางที่ 3.21 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-RF	53
ตารางที่ 3.22 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-IGM.....	55
ตารางที่ 3.23 ตัวอย่าง Confusion Matrix.....	56
ตารางที่ 4.1 ตารางการทดสอบประสิทธิภาพของค่า k	60
ตารางที่ 4.2 ค่าเฉลี่ยในการทดลองค่า input ในการทดสอบกับอัลกอริทึม <i>CNN</i>	60
ตารางที่ 4.3 ผลการทดสอบด้วยอัลกอริทึม <i>KNN</i>	62
ตารางที่ 4.4 ผลการทดสอบด้วยอัลกอริทึม <i>Naïve Bayes</i>	65
ตารางที่ 4.5 ผลการทดสอบด้วยอัลกอริทึม <i>CNN</i>	68
ตารางที่ 4.6 ตารางค่าเฉลี่ย <i>F-measure</i> การให้น้ำหนักร่วมกับอัลกอริทึม	70
ตารางที่ 4.7 ทดสอบโมเดลที่มีสัดส่วน 100 : 10 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>KNN</i>	72
ตารางที่ 4.8 ทดสอบโมเดลที่มีสัดส่วน 100 : 20 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>KNN</i>	72
ตารางที่ 4.9 ทดสอบโมเดลที่มีสัดส่วน 100 : 30 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>KNN</i>	73
ตารางที่ 4.10 ทดสอบโมเดลที่มีสัดส่วน 100 : 10 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>นาอิวเบย์</i>	75
ตารางที่ 4.11 ทดสอบโมเดลที่มีสัดส่วน 100 : 20 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>นาอิวเบย์</i>	75
ตารางที่ 4.12 ทดสอบโมเดลที่มีสัดส่วน 100 : 30 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>นาอิวเบย์</i>	76
ตารางที่ 4.13 ทดสอบโมเดลที่มีสัดส่วน 100 : 10 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>CNN</i>	78
ตารางที่ 4.14 ทดสอบโมเดลที่มีสัดส่วน 100 : 20 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>CNN</i>	78
ตารางที่ 4.15 ทดสอบโมเดลที่มีสัดส่วน 100 : 30 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม <i>CNN</i>	79

สารบัญภาพประกอบ

	หน้า
ภาพประกอบที่ 2.1 Bag of words.....	9
ภาพประกอบที่ 2.2 ตัวอย่างการคำนวณ Convolution	16
ภาพประกอบที่ 2.3 ตัวอย่างการทำ Pooling layer.....	17
ภาพประกอบที่ 2.4 ตาราง Confusion Matrix	17
ภาพประกอบที่ 3.1 กรอบการดำเนินงานของระบบ	21
ภาพประกอบที่ 3.2 Data Collection.....	21
ภาพประกอบที่ 3.3 ตัวอย่างเอกสารข้อความแสดงความคิดเห็น	22
ภาพประกอบที่ 3.4 ตัวอย่างเอกสารที่อยู่ในรูปแบบ XML.....	22
ภาพประกอบที่ 3.5 Classifier Modeling	23
ภาพประกอบที่ 3.6 ตัวอย่าง Unknown word	587
ภาพประกอบที่ 3.7 ตัวอย่างหน้าจอโปรแกรม.....	588
ภาพประกอบที่ 4.1 ตัวอย่างบทวิจารณ์สินค้าอิเล็กทรอนิกส์ที่ใช้ในการทดสอบ.....	599
ภาพประกอบที่ 4.2 กราฟค่าเฉลี่ย F-measure การให้น้ำหนักร่วมกับอัลกอริทึม	700
ภาพประกอบที่ 4.3 Curse of Dimensionality	811
ภาพประกอบที่ 4.4 คุณลักษณะที่ไม่ส่งผลต่อการจัดกลุ่ม	822
ภาพประกอบที่ ก-1 ไฟล์ Eclipse สำหรับติดตั้ง.....	98
ภาพประกอบที่ ก-2 เลือกตัวเลือกการติดตั้งโปรแกรม.....	98
ภาพประกอบที่ ก-3 ขั้นตอนการติดตั้งไฟล์	99
ภาพประกอบที่ ก-4 ไอคอนโปรแกรม Eclipse.....	99
ภาพประกอบที่ ก-5 แสดงข้อความ error ของโปรแกรม eclipse.....	100
ภาพประกอบที่ ก-6 แสดงไฟล์ JDK.exe	100
ภาพประกอบที่ ก-7 แสดงการติดตั้ง JDK ขั้นตอนที่ 1	101
ภาพประกอบที่ ก-8 แสดงการติดตั้ง JDK ขั้นตอนที่ 2	101
ภาพประกอบที่ ก-9 แสดงการติดตั้ง JDK ขั้นตอนที่ 3	102
ภาพประกอบที่ ก-10 แสดงการติดตั้ง JDK ขั้นตอนที่ 4	102
ภาพประกอบที่ ก-11 แสดงการติดตั้ง JDK ขั้นตอนที่ 5	103
ภาพประกอบที่ ก-12 แสดงการติดตั้ง JDK เสร็จสิ้นสมบูรณ์	103
ภาพประกอบที่ ก-13 ไฟล์ Python ที่ดาวน์โหลดมา.....	104
ภาพประกอบที่ ก-14 แสดงการติดตั้ง Python ขั้นที่ 1	104

สารบัญภาพประกอบ (ต่อ)

	หน้า
ภาพประกอบที่ ก-15 แสดงการติดตั้ง Python ขั้นที่ 2	105
ภาพประกอบที่ ก-16 ทำการแตกไฟล์ ClassifierImbalanced	105
ภาพประกอบที่ ก-17 โปรแกรม ClassifierImbalanced	105
ภาพประกอบที่ ก-18 สร้าง shotcut	106
ภาพประกอบที่ ก-19 การติดตั้งโปรแกรมเสร็จสมบูรณ์	106
ภาพประกอบที่ ข-1 ตัวอย่างโปรแกรมหน้าการสร้างโมเดล	108
ภาพประกอบที่ ข-2 ตัวอย่างโปรแกรมหน้าการสร้างโมเดล	119
ภาพประกอบที่ ข-3 ตัวอย่างโปรแกรมหน้าการนำโมเดลการจำแนกข้อมูลที่มีความไม่สมดุลไปใช้งาน	110

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

การจำแนกความรู้สึก (Sentiment Classification) [1] คือการจำแนกเอกสารตามชั้นความรู้สึกซึ่งโดยทั่วไปอาจจะจำแนกเป็นความรู้สึกที่เป็นบวก (Positive) ความรู้สึกที่เป็นลบ (Negative) และความรู้สึกที่เป็นกลาง (Neutral) โดยการจำแนกความรู้สึกนั้น ได้รับการศึกษามาอย่างต่อเนื่อง เพราะการประยุกต์ใช้ในหลายลักษณะ แต่โดยทั่วไปมักจะนิยมใช้ในการจำแนกความรู้สึกที่มีการแสดงไว้ในรูปแบบข้อความ (Text) [1] เช่น ประยุกต์ใช้ในการจัดอันดับความรู้สึกจากข้อความแสดงความคิดเห็นของผู้คนที่ติดต่อสินค้าและบริการ การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้เรียน การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้คนในเรื่องการเมือง เป็นต้น

อย่างไรก็ตาม แม้ว่าการจำแนกความรู้สึกจะได้รับการศึกษาและความสนใจอย่างต่อเนื่อง แต่ยังมีปัญหาที่พบในการจำแนกความรู้สึกหลายประเด็น ประเด็นที่น่าสนใจและยังคงได้รับการศึกษาเพื่อการแก้ปัญหาอยู่คือ ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึก (Imbalanced Sentiment Classification) โดยทั่วไปที่พบบ่อยคือปัญหาความไม่สมดุลของข้อมูลในคลาส (Class Imbalance Data) [2–5]

ซึ่งปัญหาความไม่สมดุลของข้อมูลในคลาสนั้น เกิดจากกลุ่มตัวอย่างที่ใช้ในการเรียนรู้ข้อมูลไม่สมดุลกัน โดยกลุ่มที่มีข้อมูลมากกว่าจะเรียกว่า “ข้อมูลกลุ่มหลัก (Majority Class)” ขณะที่กลุ่มตัวอย่างที่มีข้อมูลจำนวนน้อยกว่าจะเรียกว่า “ข้อมูลกลุ่มรอง (Minority Class)” เมื่อนำเอาชุดข้อมูลในลักษณะนี้ไปเรียนรู้เพื่อสร้างตัวจำแนกความรู้สึก (Sentiment Classifier) ข้อมูลใหม่ๆ ที่อ่านเข้ามาเพื่อวิเคราะห์เพื่อจำแนกกลุ่มด้วยตัวจำแนกความรู้สึกดังกล่าว ก็มีแนวโน้มที่จะทำนายกลุ่มของข้อมูลนั้นไปยังทิศทางของข้อมูลกลุ่มหลักที่ใช้ในการเรียนรู้ตัวจำแนกความรู้สึก

เทคนิคหลายๆ เทคนิคได้ถูกนำเสนอเพื่อใช้ในการควบคุมปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึก เช่น Resampling Methods [4] สำหรับวิธีการนี้จะเป็นการประยุกต์เอาวิธีสุ่มตัวอย่างซึ่งเป็นวิธีการทางสถิติ เพื่อสร้างข้อมูลสำหรับการสอน โดยมีจุดประสงค์เพื่อให้จำนวนสมาชิกในข้อมูลทั้งสองกลุ่มมีความสมดุลกัน ซึ่งประกอบด้วย 2 วิธีการใหญ่ๆ คือ Oversampling [6] และ Undersampling [6] โดยวิธีการ Oversampling จะทำการสุ่มข้อมูลในกลุ่มรองเพื่อสร้างข้อมูลใหม่ของกลุ่มรองให้มีจำนวนเพิ่มมากขึ้นให้ใกล้เคียงหรือเท่ากับจำนวนข้อมูลในกลุ่มหลัก และในทางตรงข้ามวิธีการ Undersampling จะทำการสุ่มเลือกข้อมูลสำหรับการสอนจากข้อมูลในกลุ่มหลัก ให้ได้จำนวนที่

ใกล้เคียงกับจำนวนข้อมูลในกลุ่มรอง โดยทั่วไปมักประยุกต์วิธีการแบบ Undersampling แต่ก็เกิดปัญหาข้อมูลไม่เพียงพอต่อการเรียนรู้

โดยทั่วไปแล้ว เทคนิคด้าน Resampling Methods มักจะประยุกต์ใช้การคัดเลือกคุณลักษณะ (Feature Selection) [7] เข้ามาช่วยเพื่อควบคุมปัญหาความไม่สมดุลของข้อมูลในคลาส โดยเป็นการคัดเลือกคุณลักษณะเด่นๆ ของข้อมูลในแต่ละคลาสเพื่อเป็นตัวแทนเอกสาร และใช้ในการสร้างตัวจำแนกจำแนกความรู้สึก แต่ก็พบปัญหาคือ บ่อยครั้งพบว่าคุณลักษณะในแต่ละคลาสคือคุณลักษณะเดียวกัน ดังนั้นอาจจะเป็นการยากในการนำมาใช้เพื่อการจำแนกความรู้สึก

อย่างไรก็ตาม เมื่อไม่นานมานี้ หลายงานวิจัยที่นำเสนอเทคนิคการให้น้ำหนักคำ (Term Weighting) เข้ามาช่วยในการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึก [8], [9] และพบว่าเทคนิคการให้น้ำหนักคำแบบมีผู้สอน (Supervised Term Weighting: STW) มีแนวโน้มที่จะทำให้เกิดประสิทธิภาพในการจำแนกความรู้สึกที่ดีขึ้น

ดังนั้นในโครงงานปริญญานิพนธ์ฉบับนี้ จึงได้นำเสนอการศึกษาการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึกด้วยเทคนิคการให้น้ำหนักคำแบบมีผู้สอนอย่างน้อย 3 เทคนิค พร้อมทั้งทำการเปรียบเทียบการเทคนิคการให้น้ำหนักคำแบบไม่มีผู้สอน (Unsupervised Term Weighting) ที่นิยมใช้ในการจำแนกเอกสารความรู้สึกนั้นคือ *tf-idf* (Term Frequency-Inverse Document Frequency) (Salton, Wong, & Yang, 1975) ภายใต้ตัวจำแนกความรู้สึกอย่างน้อย 3 ตัว

1.2 วัตถุประสงค์ของโครงงาน

นำเสนอกระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุลโดยมีเครื่องมือหลักคือเทคนิคการให้น้ำหนักคำแบบมีผู้สอน

1.3 ขอบเขตของโครงงาน

- 1) นำเสนอกระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุลโดยมีเครื่องมือหลักคือเทคนิคการให้น้ำหนักคำแบบมีผู้สอน (Supervised Term Weighting)
- 2) เป็นการศึกษาการจำแนกแบบ 2 กลุ่ม โดยการศึกษาความไม่สมดุลระหว่าง Majority Class และ Minority Class ใน 3 ระดับของการพิจารณา คือ
 - (1) 100:10
 - (2) 100:20
 - (3) 100:30

- 3) ข้อมูลที่ใช้ในการทดสอบในการจำแนกความรู้สึกที่ไม่สมดุล ในโครงการปริญญาโทฉบับนี้ เป็นข้อความรีวิวสินค้าอิเล็กทรอนิกส์ที่รวบรวมมาจากเว็บไซต์ Amazon เอกสารจะอยู่ในรูปแบบ XML
- 4) หนึ่งเอกสารมีคำมากกว่า 30 คำ และไม่เกิน 300 คำ ใช้ทั้งหมด 50,000 เอกสาร
- 5) ศึกษาเชิงเปรียบเทียบการให้น้ำหนักคำด้วยรูปแบบเทคนิคการให้น้ำหนักคำแบบมีผู้สอนอย่างน้อย 3 ตัว โดยเปรียบเทียบกับเทคนิค *tf-idf* ซึ่งเป็นเทคนิคการให้น้ำหนักแบบไม่มีผู้สอนที่นิยมใช้
- 6) ศึกษาเชิงเปรียบเทียบอัลกอริทึมการเรียนรู้แบบมีผู้สอนที่ใช้ในการสร้างตัวจำแนกความรู้สึกอย่างน้อย 3 ตัว
- 7) การวัดประสิทธิภาพการจำแนกความรู้สึกที่ไม่สมดุลจะประเมินด้วยค่าความระลึก (Recall) ค่าความแม่นยำ (Precision) และค่าเอฟ (F-measure: F1)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ได้กระบวนการในการจำแนกข้อความแสดงความรู้สึกที่มีข้อมูลแบบไม่สมดุล

1.5 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

Hardware: คอมพิวเตอร์รุ่น Intel® Core™ i5-9400F CPU @ 2.90 GHz ,
RAM 16 GB BUS 2666 MHz, SSD SATA 240 GB

Operating System: Windows 10 Pro

Programming Language: Java, Xml

Application Tools: Eclipse IDE for java

1.6 แผนการดำเนินงาน

โครงการปริญญาโทฉบับนี้ ดำเนินงาน ณ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม ระหว่างเดือน พฤษภาคม 2563 ถึง เมษายน 2563 ดังที่แสดงในตารางที่ 1.1

ตารางที่ 1.1 แผนการดำเนินงาน

[illegible]

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้ จะเป็นการอธิบายถึงแนวคิด ทฤษฎี และเทคนิคที่เกี่ยวข้อง ต่อการวิจัยและพัฒนากระบวนการของการจำแนกข้อความแสดงความคิดเห็น โดยที่เอกสารเหล่านั้นมีลักษณะที่ขาดความสมดุลของเอกสาร และขาดความสมดุลคุณลักษณะในคลาส โดยแนวคิดและเทคนิคที่เกี่ยวข้องต่อไปนี้

2.1 ข้อมูลที่ไม่สมดุล (Imbalanced Data)

ข้อมูลที่ไม่สมดุล [2-5] หมายถึง ข้อมูลที่มีการกระจายตัวไม่เท่าเทียมกันในแต่ละกลุ่ม หรือข้อมูลซึ่งมีอัตราของสมาชิกกลุ่มหลัก (Majority) และกลุ่มรอง (Minority) มีจำนวนไม่เท่ากัน เช่น 1000:1 หรือ 10000:1 เป็นต้น ตัวอย่างเช่น สมมติว่ามีข้อมูลเกี่ยวกับผู้ป่วยโรคมะเร็งชนิดหนึ่ง โดยที่ข้อมูลผู้ป่วยที่ไม่เป็นมะเร็งเป็นข้อมูลกลุ่มหลัก อาจจะมีข้อมูลหลายหมื่นคน ในขณะที่ข้อมูลผู้ป่วยที่เป็นโรคมะเร็งเป็นกลุ่มข้อมูลกลุ่มรองอาจจะมีข้อมูลเพียงหลักร้อยคน

ในการจำแนกข้อมูล (Data Classification) หากนำข้อมูลทั้งสองมาสร้างโมเดลเพื่อการจำแนกเอกสาร จะมีความเป็นไปได้สูงว่าเมื่อสร้างโมเดลสำหรับการจำแนกข้อมูลแล้ว ข้อมูลที่นำมาทดสอบมีโอกาสถูกทำนายเข้าเป็นกลุ่มไม่เป็นมะเร็ง เพราะจำนวนข้อมูลที่ใช้ในการสร้างโมเดลแล้ว ข้อมูลที่นำมาทดสอบมีโอกาสถูกทำนายเข้าเป็นกลุ่มไม่เป็นมะเร็ง เพราะจำนวนข้อมูลที่ใช้ในการสร้างโมเดลนั้นไม่สมดุล ดังนั้นในการทำนายกลุ่มจึงมีทิศทางถูกจำแนกไปยังกลุ่มที่มีข้อมูลมากกว่า

อย่างไรก็ตาม ความไม่สมดุลของข้อมูลในคลาส ไม่ได้หมายความว่าความแตกต่างของจำนวนข้อความ แต่รวมถึงขนาดของคลาส (Class Size) คลาสย่อย (Sub-Class) และคลาสที่มีการทับซ้อนของข้อมูล (Class Overlap) ซึ่งหมายถึงข้อมูลหนึ่งตัวสามารถปรากฏในหลายๆ คลาส เป็นต้น รายละเอียดแต่ละปัญหาสามารถอธิบายได้ดังนี้

(1) ปัญหาความไม่สมดุลอันเนื่องมาจากการกระจายข้อมูล (Data Distributed Imbalanced) [2] คือ จำนวนเอกสารข้อความในแต่ละกลุ่มมีจำนวนที่แตกต่างกันมาก กลายเป็นปัญหาของการจำแนกเอกสาร เพราะการกระจายตัวของเอกสารในแต่ละกลุ่มไม่เท่ากัน ดังนั้นในการจำแนกเอกสารเอกสารที่ถูกจำแนกจะมีโอกาสที่จะถูกทำนายไปอยู่ในกลุ่มที่มีเอกสารจำนวนมาก

(2) ปัญหาความไม่สมดุลอันเนื่องมาจากจำนวนเอกสารในแต่ละคลาสไม่เท่ากัน (Class Size Imbalanced) [2] นั่นคือ ขนาดของเอกสารในแต่ละกลุ่มไม่มีความสมดุลกัน

(3) ปัญหาการทับซ้อนของข้อมูล (Class Overlap) [2] คือ ปัญหาที่เกิดจากการที่เอกสารหนึ่งๆ มีโอกาสที่จะถูกจำแนกไปอยู่ได้ในหลายๆ กลุ่ม

(4) ปัญหากลุ่มย่อย (Sub-Class Problem) [2] คือ หลายๆ ปัญหาด้านการจำแนกเอกสาร อาจจะพบว่า ในกลุ่มๆ หนึ่งอาจจะมีหลายกลุ่มย่อย ซึ่งปัญหาดังกล่าวจะนำไปสู่ปัญหาความไม่สมดุลอันเนื่องมาจากจำนวนเอกสารในแต่ละคลาสไม่เท่ากันนั่นเอง

2.2 การจำแนกความรู้สึก (Sentiment Classification)

การจำแนกความรู้สึก [1] นั้น มีจุดประสงค์เพื่อวิเคราะห์เอกสารที่แสดงความรู้สึกออกเป็นความรู้สึกที่เป็นบวก (Positive) ความรู้สึกที่เป็นกลาง (Neutral) หรือความรู้สึกที่เป็นลบ (Negative) โดยทั่วไปเทคนิคที่ใช้ในการจำแนกความรู้สึกจะเป็นการผสมผสานระหว่างเทคนิคของการใช้การประมวลผลภาษาธรรมชาติ (Natural Language Processing : NLP) และเหมืองข้อความ (Text Mining)

ปัญหาอย่างหนึ่งที่พบในงานด้านการจำแนกเอกสารข้อความ รวมถึงการจำแนกความรู้สึกก็คือ การที่จำนวนข้อมูลในแต่ละคลาสมีขนาดไม่เท่ากัน และเรียกปัญหานี้ที่พบในการจำแนกความรู้สึกว่า “การจำแนกความรู้สึกที่ไม่สมดุล (Imbalanced Sentiment Classification) [2]–[5]” โดยกลุ่มที่มีข้อมูลมากกว่าจะเรียกว่า “ข้อมูลกลุ่มหลัก (Majority Class)” และกลุ่มที่มีข้อมูลน้อยกว่าจะเรียกว่า “ข้อมูลกลุ่มรอง (Minority Class)” ในระหว่างการทำนายกลุ่ม ก็มักมีความเอนเอียงไปในทิศทางของข้อมูลกลุ่มหลัก เพราะมีข้อมูลที่มากกว่า

จากการศึกษาที่ผ่านมา พบว่ามีวิธีการในการจัดการกับปัญหาข้อมูลที่ไม่สมดุลหลายวิธี เช่น Re-Sampling [4], One-class Classification [10] และ Cost-Sensitive Learning [10] อย่างไรก็ตามวิธีการที่นำเสนอที่ผ่านมาก็ยังไม่สามารถจัดการปัญหาข้อมูลที่ไม่สมดุลได้ทั้งหมด เพราะบริบทของข้อมูลในการศึกษามีความหลากหลาย วิธีการที่ได้ดีกับชุดข้อมูลหนึ่ง ก็ไม่ได้หมายความว่า จะใช้จัดการปัญหาข้อมูลที่ไม่สมดุลที่เกิดในข้อมูลชุดอื่นๆ ได้ดี

2.3 เทคนิคและอัลกอริทึมที่เกี่ยวข้อง

2.3.1 การจำแนกหมวดหมู่เอกสาร (Text Classification)

การจำแนกหมวดหมู่เอกสาร [10–12] เป็นการนำอัลกอริทึมการเรียนรู้ของเครื่องแบบมีผู้สอน (Supervised Machine Learning) มาประยุกต์รวมกับการประมวลผลภาษาธรรมชาติ เพื่อการจำแนกกลุ่มเอกสารแบบอัตโนมัติ โดยอาศัยการวิเคราะห์เนื้อหาของเอกสาร

โดยในการจำแนกเอกสารข้อความแบบอัตโนมัติ นั้น จะใช้อัลกอริทึมการเรียนรู้ของเครื่องแบบมีผู้สอนในการสร้างตัวจำแนกเอกสาร (Text Classifier) จากเอกสารชุดสอน (Training Set) ที่เอกสารแต่ละฉบับต้องมีลาเบล (Label) ของคลาสกำกับ

กำหนดให้ D เป็นเซตของเอกสาร ขณะที่ C เป็นเซตของคลาสที่เป็นไปได้ นั่นคือ $\{c_1, c_2, \dots, c_{|C|}\}$ และกำหนดให้ T เป็นคู่ลำดับ (d_j, c_i) ที่จะบ่งบอกว่าเอกสาร d_j อยู่ภายใต้กลุ่มหรือหมวดหมู่ c_i โดยให้ F เป็นฟังก์ชันที่กำหนดให้คู่ลำดับ (d_j, c_i) เพื่อบอกว่าเอกสาร d_j ควรอยู่ภายใต้กลุ่มหรือหมวดหมู่ c_i หรือไม่ ดังนั้น การประมวลค่าของฟังก์ชันเป้าหมายสามารถแสดงได้คือ $F : D \times C \rightarrow \{T, F\}$ ซึ่งฟังก์ชันเป้าหมายจะแทนตัวจำแนกเอกสารนั่นเอง

2.3.2 ขั้นตอนการเตรียมเอกสาร (Document Pre-processing)

ในขั้นตอนนี้ จะเป็นการการเตรียมเอกสารหรือบทความให้อยู่ในรูปแบบที่พร้อมก่อนที่จะนำเข้าไปประมวลผลในขั้นตอนถัดไป ซึ่งจะมีขั้นตอนดังต่อไปนี้

2.3.2.1 การตัดคำ (Word Segmentation)

การตัดคำเป็นขั้นตอนแรกที่จะถูกดำเนินการในการประมวลผลภาษาธรรมชาติ ซึ่งเป็นการแบ่งข้อความ (String) ออกเป็นหน่วยย่อยที่มีความหมายทางภาษา โดยทั่วไปมักนิยมแบ่งข้อความออกมาเป็น “คำ (Word)” โดยในภาษาอังกฤษ การแบ่งข้อความออกเป็น “คำ” จะใช้ช่องว่าง (White Space) หรือเครื่องหมายวรรคตอน

2.3.2.2 การตัดคำหยุด (Stop-word Removal)

การตัดคำหยุด [15] คือ กระบวนการ การตัดคำหรือสัญลักษณ์ที่พบบ่อยในเอกสาร แต่คำเหล่านั้นไม่มีนัยสำคัญ ในที่นี้หมายถึงคำที่ใช้กันโดยทั่วไปไม่มีความสำคัญต่อเอกสารเมื่อตัดออกจากเอกสารแล้วไม่ทำให้ใจความสำคัญของเอกสารเปลี่ยนแปลง

ดังนั้น การตัดคำหยุด จึงมีความจำเป็นอย่างมากในการจัดกลุ่มเอกสารแบบอัตโนมัติ เพราะจะช่วยลดระยะเวลาในการประมวลผลได้เป็นอย่างมาก เนื่องจากระบบไม่ต้องนำคำเหล่านั้นไปประมวลผล เช่น คำว่า “is”, “the”, “are”, “and” แต่จะมีการนำคำที่มีผลต่องานออกจากพจนานุกรมคำหยุด เช่นคำว่า “not”, “very”, “much” เป็นต้น

ตารางที่ 2.1 แสดงการตัดคำหยุด

เอกสารที่ผ่านการทำ Lemmatization	เอกสารที่ผ่านการตัดคำหยุด
Black / space / is / great / song	Black / space / great / song
Possible / the / worst / music / of / the / year	Worst / music / year

2.3.2.3 การเปลี่ยนรูปแบบคำให้อยู่ในรูปแบบดั้งเดิม (Lemmatization)

การทำ Lemmatization คือ การเปลี่ยนคำให้มาอยู่ในรูปแบบดั้งเดิม (Lemma) เนื่องจากคำในภาษาอังกฤษนั้น มีการใช้งานคำที่มีความหมายเหมือนกันในลักษณะ [14] เช่น คำว่า “is”, “am”, “are”, “was” จะถูกเปลี่ยนเป็นคำว่า “be” หรือ “saw”, “seen” จะถูกเปลี่ยนเป็นคำว่า “see” ดังนั้นจึงจำเป็นที่ต้องมีการเปลี่ยนรูปแบบคำเหล่านั้นให้อยู่ในรูปแบบดั้งเดิม ในโครงงานนปริณญาณพนธ์นี้จะเลือกใช้วิธีการ Lemmatization Tagging โดยใช้ API จาก Stanford ซึ่งมีขั้นตอนการทำ Lemmatization ดังนี้

1. TokenizerAnnotator เป็นการตัดคำโดยใช้หลักการเดียวกับ *Penn Treebank* เช่น isn't จะได้เป็น is, n't ตัวอย่างเอกสาร

Yummy's great song จะได้ song / Yummy / 's / great

2. ssplit เป็นการนำคำที่ผ่านกระบวนการตัดคำแล้ว มาเรียงลำดับตามประโยคเดิม

Yummy's great song จะได้ Yummy / 's / great / song

3. POS (Part-Of-Speech Tagging) เป็นการติด tag ให้แต่ละคำเพื่อบอกว่าคำๆ อยู่ในบริบทใด เช่น bigger จะถูกกำหนด tag เป็น JJR (adj., comparative) เพื่อนำไปใช้ในการหาคำที่อยู่ในรูปแบบดั้งเดิม จากตัวอย่างเอกสารข้างต้นในขั้นตอนการติด tag จะได้

Yummy (NNP) | 's (POS) | great (NNP) | Song (NN)

4. Lemma จะเป็นการนำเอาคำที่ได้ภายหลังจากการติด tag มาทำ lemma โดยใช้ Wordnet ซึ่งจะมีการจัดกลุ่ม tag เป็น 5 กลุ่มคือ verbs (v), nouns (n), adjectives (a), satellite adjectives (s), adverbs (r)

Yummy	จะเป็นคำว่า	Yummy
's	จะเป็นคำว่า	is
great	จะเป็นคำว่า	great
song	จะเป็นคำว่า	song

2.3.3 การสร้างตัวแทนเอกสาร (Document Representation)

เนื่องจากคอมพิวเตอร์ไม่สามารถเรียนรู้ และจำแนกหมวดหมู่เอกสารที่เป็นภาษาธรรมชาติได้โดยตรง จึงจำเป็นต้องแปลงเอกสารให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถใช้ในการเรียนรู้ได้ โดยขั้นตอนนี้เรียกว่า การทำดัชนี (Indexing) [16] เพื่อสร้างตัวแทนเนื้อหาเอกสาร (Document Representation) สำหรับใช้ในกระบวนการเรียนรู้ วัตถุประสงค์ของการสร้างดัชนี คือ การคำนวณค่าที่จะนำมาใช้เป็นค่าคุณลักษณะของเอกสาร หรืออาจจะเรียกได้ว่าการหาน้ำหนัก (Term Weighting) การสร้างดัชนีโดยทั่วไปที่นิยมใช้กัน จะเริ่มจากการสร้างเวกเตอร์ตัวแทนเอกสารจากนั้นจะสร้างเมตริกซ์ของกลุ่มเอกสารขึ้นจากเวกเตอร์เอกสารทั้งหมดในกลุ่ม ซึ่งวิธีหาความถี่ของคำที่ปรากฏในเอกสารที่ผ่านการตัดคำมาเป็นค่าน้ำหนัก ถ้าคำใดผ่านการตัดคำมีปริมาณมาก ก็จะมีค่าความถี่มาก ซึ่งจะส่งผลให้ได้ค่าน้ำหนักที่มีค่าสูงมากตาม เมื่อถึงขั้นตอนนี้จะได้รูปแบบที่มีลักษณะของการแสดงความสัมพันธ์ระหว่างคำ (Words : w) และเอกสารทั้งหมด (Documents : d) ด้วย เวกเตอร์ 2 มิติ ซึ่งคำที่ได้นั้นต้องผ่านการทำดัชนีและการตัดคำหยุด (Stop-words) ออกไป และเอกสารทั้งหมดอยู่ในรูปแบบ Vector Space Model หากพีเจอร์ (Feature) ที่ใช้เป็น “คำ” บางครั้งจึงเรียกรูปแบบนี้ว่า “ถุงคำ (Bag of Words: BOW)” [16] โดยสามารถแสดงได้ดังภาพประกอบที่ 2.1

	w_1	w_2	...	w_k	...	w_v
d_1	w_{11}	w_{12}	...	w_{1k}	...	w_{1v}
d_2	w_{21}	w_{22}	...	w_{2k}	...	w_{2v}
...
d_N	w_{N1}	w_{N2}	...	w_{Nk}	...	w_{Nv}

ภาพประกอบที่ 2.1 Bag of words

2.3.4 การเลือกคุณลักษณะ (Feature Selection)

ภายหลังจากการตัดคำ การตัดคำหยุด และการคัดเลือกคำด้วยพจนานุกรม คลังคำที่ได้จะถูกนำเข้าสู่ขั้นตอนของการคัดเลือกคุณลักษณะด้วย *Information Gain* สำหรับการเลือกคุณลักษณะจะเป็นวิธีเบื้องต้นในการลดขนาดเอกสาร [18, 19] เพราะการนำคำที่ไม่มีนัยสำคัญออกแล้วยังไม่เพียงพอ ซึ่งจำนวนคุณลักษณะมีผลต่อประสิทธิภาพของการจำแนกหมวดหมู่เอกสาร เนื่องจากอัลกอริทึมที่ใช้ในการเรียนรู้เพื่อสร้างตัวจำแนกหมวดหมู่ โดยทั่วไปไม่สามารถรองรับการทำงานกับจำนวนคุณลักษณะของเอกสารที่สูงมากได้ดี การลดขนาดของเอกสารจึงเป็นขั้นตอนหนึ่งที่จะต้อง

กระทำการก่อน ในโครงงานปริญาานิพนธ์นี้จะใช้ค่าเกนสารสนเทศ (IG: Information Gain) เป็นตัววัดคุณลักษณะของเอกสาร ซึ่งค่า IG จะคำนวณจากจำนวนบิตที่ได้รับสำหรับการทำนายกลุ่ม โดยการดูจากการมีอยู่หรือไม่มีอยู่ของคำในเอกสาร ให้ C_1, \dots, C_K แทนเซตที่เป็นไปได้ของกลุ่ม คำ IG ของคำ w นิยามโดย

$$IG(w) = - \sum P(C_j) \log P(C_j) + P(w) \sum P(C_j|w) \log P(C_j|w) + P(w) \sum P(C_j|w) \log P(C_j|w)$$

$P(C_j)$ คือความน่าจะเป็นของคลาสแต่ละคลาส

$P(w)$ คือความน่าจะเป็นของ “คำ” แต่ละคำที่พบ

$P(C_j|w)$ คือความน่าจะเป็นของ “คลาส” เพื่อพิจารณาจาก “คำ”

เมื่อทำการคำนวณค่า IG ของแต่ละคุณลักษณะที่ได้ จากนั้นทำการจัดเรียงคุณลักษณะที่มีค่า IG มากไปหาน้อยและทำการตัดคุณลักษณะที่มีค่าต่ำกว่าเกณฑ์ทิ้งไป ซึ่งจะช่วยลดระยะเวลาในการประมวลผล และยังคงความแม่นยำในการจัดกลุ่มเอกสาร

2.3.5 การให้น้ำหนักคำ (Term Weighting)

การให้น้ำหนักคำ [17] ถือว่าเป็นส่วนหนึ่งของการจัดการเอกสาร โดยรูปแบบการให้น้ำหนักสามารถแบ่งออกเป็นสองประเภทหลักตามการใช้งานข้อมูลชั้นเรียนในเอกสารการฝึกอบรม ดังนี้

1. การให้น้ำหนักคำแบบไม่มีผู้สอน (Unsupervised Term Weighting: UTW) [18] คือรูปแบบการให้น้ำหนักคำที่ไม่ใช้ข้อมูลชั้นเรียนเพื่อสร้างน้ำหนัก รูปแบบที่ได้รับความนิยมมากที่สุดคือ Term Frequency - Inverse Document Frequency (TF-IDF) ซึ่งถูกใช้อย่างมีประสิทธิภาพในการศึกษาการดึงข้อมูล แต่อย่างไรก็ตามมันไม่เหมาะสำหรับงานการจัดหมวดหมู่ข้อความ เนื่องจากการให้น้ำหนักคำแบบ UTW เป็นการให้น้ำหนักคำกับเอกสารทั้งหมดโดยไม่แบ่งหมวดหมู่เอกสาร โดยหากใช้รูปแบบนี้จะทำให้ประสิทธิภาพในการจำแนกหมวดหมู่ข้อความลดลง

2. การให้น้ำหนักคำแบบมีผู้สอน (Supervised Term Weighting: STW) [11] ซึ่งได้รับการเสนอครั้งแรกโดย Debolc และ Sebastiani [11] การให้น้ำหนักคำแบบ STW จะใช้ชุดข้อมูลการฝึกอบรมของข้อมูลระดับชั้นเรียนเพื่อคำนวณน้ำหนักของคำศัพท์ โดยการให้น้ำหนักในแบบนี้จะใช้ประโยชน์จากข้อมูลระดับที่รู้จักในคลังข้อมูลการฝึกอบรม โดยจะทำให้การให้น้ำหนักมีประสิทธิภาพที่ดียิ่งขึ้น สำหรับการจำแนกหมวดหมู่ความรู้สึกของข้อความ การวิเคราะห์ความรู้สึก การจำแนกความไม่สมดุลของชุดเอกสาร และอื่นๆ โดยองค์ประกอบพื้นฐานของการกำหนดน้ำหนักมีดังตารางที่ 2.2

ตารางที่ 2.2 สัญลักษณ์สำหรับ Supervised Term Weighting (STW)

	c_k	\bar{c}_k
t_i	A	C
\bar{t}_i	B	D

โดยตัวแปรพื้นฐานมีดังต่อไปนี้

t_i คือ คำที่มีในเอกสาร

\bar{t}_i คือ คำที่ไม่มีในเอกสาร

c_k คือ กลุ่มเอกสารกลุ่มหลัก

\bar{c}_k คือ กลุ่มเอกสารกลุ่มรอง

A คือ จำนวนเอกสารใน c_k ที่คำว่า t_i เกิดขึ้นอย่างน้อยหนึ่งครั้ง

C คือ จำนวนเอกสารที่ไม่ได้เป็นของ c_k ที่คำว่า t_i เกิดขึ้นอย่างน้อยหนึ่งครั้ง

B คือ จำนวนเอกสารที่เป็นของ c_k โดยที่คำว่า t_i ไม่ได้เกิดขึ้น

D คือ จำนวนเอกสารที่ไม่ได้เป็นของ c_k โดยที่คำว่า t_i ไม่ได้เกิดขึ้น

N คือ จำนวนเอกสารทั้งหมดในคลังข้อมูล $N = A + B + C + D$

N_p คือ จำนวนเอกสารในชั้นบวก $N_p = A + B$

N_n คือ จำนวนเอกสารในชั้นเรียนที่เป็นลบ $N_n = C + D$

และตัวแปรพื้นฐานข้างต้นนำไปใช้ในอัลกอริทึมดังนี้

(1) Delta Term Frequency - Inverse Document Frequency (Delta TF-IDF)

Delta TF-IDF ถูกเสนอโดย Martineau และ Finin [19] มันคำนวณความแตกต่างของคะแนน TF-IDF ในคลาสที่เป็นบวกและลบเพื่อปรับปรุงความแม่นยำ ในฐานะที่เป็น STW จะพิจารณาการกระจายของคุณสมบัติระหว่างสองคลาสก่อนการจำแนกประเภทการรับรู้และการเพิ่มความสูงของผลค่าที่แตกต่างกัน Delta TF-IDF ช่วยเพิ่มความสำคัญของคำที่กระจายอย่างไม่สม่ำเสมอระหว่างคลาสบวกและคลาสลบ โดยที่ N_p และ N_n คือจำนวนของเอกสารในคลาสบวกและลบตามลำดับ ส่วน A และ C แสดงความถี่เอกสารของคำว่า t_i ในคลาสบวกและลบตามลำดับ ดัง (1)

$$w_{\&TF.IDF}(t_i) = TF(t_i, d_j) \times \log_2\left(\frac{N_p \times C + 1.5}{A \times N_n + 1.5}\right) \quad (1)$$

(2) Term Frequency - Inverse Document Frequency -Inverse Class Frequency (TF-IDF-ICF)

TF-IDF-ICF เป็นรูปแบบการควบคุมน้ำหนักตามแบบ TF-IDF แบบดั้งเดิม อย่างไรก็ตามมันเพิ่มปัจจัยความถี่ผกผันในคลาส (Inverse Class Frequency : ICF) [8] เพื่อให้ค่าน้ำหนักที่สูงขึ้นไปยังคำที่หายากที่เกิดขึ้นน้อยกว่าในเอกสาร (IDF) และ Class (ICF) และใน (2) M คือจำนวนคลาสในคอลเลกชันและ $CF(t_i)$ สอดคล้องกับความถี่ของคลาสที่คำ t_i ปรากฏในคอลเลกชัน TF-IDF-ICF แสดงใน (2)

$$ICF(t_i) = (1 + \log(\frac{M}{CF(t_i)})) \quad (2)$$

$$w_{TF.ICF}(t_i) = TF(t_i, d_j) \times IDF(t_i) \times ICF(t_i) \quad (3)$$

(3) Term Frequency - Relevance Frequency (TF-RF)

TF-RF [18] ได้รับการเสนอเช่นเดียวกับ Delta TF-IDF และ TF-RF คำนึงถึงการกระจายคำศัพท์ในชั้นเรียนทั้งบวกและลบ อย่างไรก็ตามมีการพิจารณาเฉพาะเอกสารที่มีคำดังกล่าว นั่นคือ ความเกี่ยวข้องของความถี่ (RF) ของข้อกำหนด TF-RF ถูกระบุใน (3) โดยที่ตัวหารน้อยที่สุดคือ 1 เพื่อหลีกเลี่ยงการหารด้วยศูนย์

$$w_{TF.RF}(t_i) = TF(t_i, d_j) \times \log_2(2 + \frac{A}{\max(1, C)}) \quad (4)$$

(4) Term Frequency - Inverse Gravity Moment (TF-IGM)

TF-IGM [20] ถูกนำเสนอให้วัดความไม่สม่ำเสมอหรือความเข้มข้นของการแจกแจงคำศัพท์ระหว่างคลาสซึ่งสะท้อนให้เห็นถึงอำนาจการจำแนกชั้นข้อตกลง

สมการ IGM มาตรฐานกำหนดอันดับ (r) ตามความเข้มข้นของการแจกแจงระหว่างคลาสของคำซึ่งคล้ายกับแนวคิดของ “แรงโน้มถ่วงโมเมนต์ (Gravity Moment: GM)” จากฟิสิกส์ IGM ถูกระบุใน (5) โดยที่ f_{ir} ($r = 1, 2, \dots, M$) ระบุจำนวนเอกสารที่มีคำว่า t_i ในคลาส $r - th$ ซึ่งส่วนโค้งเรียงตามลำดับจากมากไปน้อย ดังนั้น f_{i1} จึงแสดงความถี่ของ t_i ในคลาสที่ปรากฏบ่อยที่สุด

$$IGM(t_i) = (\frac{f_{i1}}{\sum_{r=1}^M f_{ir} \times r}) \quad (5)$$

โดยน้ำหนักเทอม TF-IGM นั้นกำหนดตาม $IGM(t_i)$ ดังที่แสดงใน (6) ค่า λ คือสัมประสิทธิ์แบบปรับได้ที่ใช้เพื่อรักษาสมมูลสัมพัทธ์ระหว่างปัจจัยทั่วโลก และท้องถิ่นในน้ำหนักของค่าสัมประสิทธิ์ λ มีค่าเริ่มต้นที่ 7.0 และสามารถตั้งเป็นค่าระหว่าง 5.0 ถึง 9.0 [20]

$$w_{TF,IGM}(t_i) = TF(t_i, d_j) \times (1 \times \lambda \times IGM(t_i)) \quad (6)$$

เพื่อแสดงให้เห็นถึงคุณสมบัติของการวัดน้ำหนักในระยะต่างๆ ได้ดีขึ้นให้พิจารณาองค์ประกอบพื้นฐานที่แสดงในตารางที่ 2.2 สมมติว่าชุดข้อมูลการฝึกอบรมมี 100 เอกสาร โดยพิจารณาการกระจายคำศัพท์ t_1 และ t_2 สำหรับสองคลาส c_p และ c_n ตามที่กำหนดไว้ใน

ตารางที่ 2.3 ตัวอย่างการแจกแจงเอกสารสองเทอม

	c_p	c_n			c_p	c_n
t_1	27	5		t_2	10	20
\bar{t}_1	3	65		\bar{t}_2	25	45

โดยคำนึงถึงการกระจาย t_1 ในตารางที่ 2.3 สามารถนำมาคำนวณการให้น้ำหนักได้ดังนี้

$$IDF(t_1) = \log(100/(27 + 5)) = \log(3.125) = 0.4949$$

$$IDF - ICF(t_1) = (1 + 0.4949) * (1 + \log(2/2)) = 1.4949$$

$$Delta\ IDF(t_1, c_p) = \log_2\left(\frac{30 * 5 + 0.5}{27 * 70 + 0.5}\right) = -3.6510$$

$$Delta\ IDF(t_1, c_n) = \log_2\left(\frac{70 * 27 + 0.5}{5 * 30 + 0.5}\right) = 1.8445$$

$$RF(t_1, c_p) = \log_2(2 + (27/5)) = 2.8875$$

$$RF(t_1, c_n) = \log_2(2 + (3/65)) = 1.0329$$

$$IGM(t_1) = 27/((27 * 1) + (5 * 2)) = 0.7297$$

$$IGM.imp(t_1) = 27/((27 * 1) + (5 * 2) + 0.0458) = 0.7288$$

สามารถแสดงผลลัพธ์การคำนวณการกระจายน้ำหนักของ t_1 และ t_2 ได้ดังตารางที่ 2.4

ตารางที่ 2.4 ผลลัพธ์การคำนวณการกระจายน้ำหนัก

Weighting Scheme	$t_1 c_p$	$t_1 c_n$	$t_2 c_p$	$t_2 c_n$
IDF	0.4949	0.4949	0.5229	0.5229
IDF – ICF	2.9898	2.9898	3.0458	3.0458
Delta IDF	-3.6510	1.8445	-0.3782	-0.1069
RF	2.8875	1.0329	1.2630	1.3536
IGM	0.3333	0.3333	0.5000	0.5000

2.3.6 นาอิวเบย์ (Naïve Bayes)

นาอิวเบย์ (Naïve Bayes) เป็นการนำเอาหลักความน่าจะเป็นเข้ามาใช้ในการจำแนกข้อความ เนื่องจากนาอิวเบย์นั้นเป็นอัลกอริทึมที่ง่ายไม่ซับซ้อน และมีความรวดเร็วในการใช้งาน ซึ่งในการคำนวณนาอิวเบย์จะเริ่มคำนวณจากแต่ละตัวอย่าง จากตัวอย่างแรกไปยังตัวอย่างที่ n โดยค่าเป้าหมายที่ต้องการของแต่ละตัวอย่าง เป็นค่าใดๆ ภายในเซต V เมื่อ V มีสมาชิกเป็นค่าเป้าหมายที่ต้องการ ในที่นี้หมายถึงจำนวนกลุ่มของข้อมูล

นาอิวเบย์เป็นการเรียนรู้ที่ง่าย เป็นวิธีการจำแนกประเภทของข้อมูลที่มีประสิทธิภาพวิธีหนึ่ง และเหมาะกับการนำมาใช้กับกรณีที่มีเซตตัวอย่างเป็นจำนวนมาก และแต่ละคุณสมบัติ (Attribute) ของตัวอย่างเป็นอิสระต่อกัน โดยนำการจำแนกประเภทนาอิวเบย์มาประยุกต์ใช้ในการจำแนกประเภทของเอกสาร (Document Classification) พบว่ายังสามารถใช้งานได้ดีไม่ต่างจากการจำแนกวิธีการอื่นๆ และวิธีการไม่มีความซับซ้อน

การกำหนดความน่าจะเป็นของข้อมูลที่จะเป็นกลุ่ม V_j สำหรับข้อมูลที่มีคุณสมบัติ n ตัว $X = \{a_1, a_2, \dots, a_n\}$ หรือใช้สัญลักษณ์ว่า $P(a_1, a_2, \dots, a_n)$ คือ

$$P(v_j | a_1, a_2, \dots, a_n) = \prod_{i=1}^n P(a_i | v_j) \quad (7)$$

โดยที่ \prod หมายถึงผลคูณของค่า $P(a_i | v_j)$ เมื่อ i และ j มีค่าเท่ากับ $1, 2, 3, \dots, n$

วิธีการเรียนรู้ที่ง่ายไปใช้มีวิธีดังต่อไปนี้คือ

(1) หาค่าความน่าจะเป็นของค่าที่พบในแต่ละกลุ่มโดยนำค่า $P(a_1, a_2, \dots, a_n | v_j)$ จากสมการมาคูณกับค่าความน่าจะเป็นของกลุ่มนั้นๆ คือ $P(v_j)$ ได้เท่ากับ V_{NB}

(2) นำค่าที่ได้มาเปรียบเทียบกับกลุ่มที่มีความน่าจะเป็นสูงสุดคือกลุ่มที่ข้อมูลนั้นอยู่ และจะถูกจัดเข้าไป เขียนเป็นสมการได้คือ

$$v_{NB} = \operatorname{argmax} P(v_j) \times \prod_{i=1}^n P(a_i | v_j) \quad : v_j \in V \quad (8)$$

2.3.7 วิธีการค้นหาเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbor: KNN)

วิธีการ KNN จะเป็นการจำแนกประเภทข้อมูลโดยขึ้นกับข้อมูลที่มีคุณสมบัติใกล้เคียงที่สุด K ตัวจากชุดข้อมูลตัวอย่าง แล้วเลือกคลาสที่สมาชิกส่วนใหญ่ที่อยู่ในกลุ่ม K ดังกล่าวสังกัดอยู่มากที่สุดให้กับ สมาชิกใหม่ การจำแนกประเภทข้อมูลโดยใช้ข้อมูลข้างเคียง K ตัวจะประกอบด้วยแอตทริบิวต์หลายตัวแปร X_i ซึ่งจะนำมาใช้ในการแบ่งกลุ่ม Y_i โดยระบุค่าตัวเลขจำนวนเต็มบวกให้กับ K ซึ่งค่านี้จะเป็นตัวบอกจำนวนของกรณี (Case) ที่จะต้องค้นหาในการทำนายกรณีใหม่ โดยในที่นี้จะกำหนด 1-KNN หมายถึง อัลกอริทึมนี้จะค้นหา 1 กรณีที่มีลักษณะใกล้เคียงกับกรณีใหม่ (1 Nearest Cases) การนำระยะทางที่หาได้จากสมาชิกในข้อมูลตัวอย่างฝึกฝน มาเรียงลำดับจากน้อยไปหามากแล้วเลือกสมาชิกที่มีระยะทาง (Distance) ใกล้เคียงที่สุดออกมา K ตัว โดยใช้การวัดระยะทางแบบ Euclidean distance มีหลักการ คือ การวัดระยะทางระหว่างสองวัตถุ ถ้าวัตถุห่างกันมากแสดงว่าวัตถุนั้นมีความคล้ายคลึงกันน้อย ถ้ามีค่าน้อยก็แสดงว่ามีความคล้ายคลึงกันมาก โดยที่ ค่า p_i แทน คุณสมบัติจากฐานข้อมูล q_i แทนคุณสมบัติที่ผู้ใช้ระบุ

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (9)$$

2.3.8 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN)

CNN ได้รับการการนำเสนอ เพื่อให้ได้ผลลัพธ์ที่น่าประทับใจในภารกิจที่สำคัญในทางปฏิบัติของการจัดหมวดหมู่ประโยค ซึ่ง CNN สามารถใช้ประโยชน์จากการแทนคำแบบกระจายโดยการแปลงโทเค็น (Tokens) ที่ประกอบด้วยแต่ละประโยคเป็นเวกเตอร์ก่อนแล้วสร้างเมทริกซ์เพื่อใช้เป็นอินพุต

Convolutional Neural Network หรือ CNN ซึ่งเป็นโครงสร้าง Neural network แบบพิเศษ ที่มีความสามารถในการจำแนกข้อมูลได้ดีกว่า Neural network ทั่วไปมาก โดย CNN คือการที่

ใช้ Layer ชนิดพิเศษ ที่เรียกว่า Convolution layer ซึ่งทำหน้าที่สกัดเอาส่วนต่างๆ ของข้อมูลออกมา CNN จะใช้ Convolution layer มาประกอบกับ Layer ชนิดอื่น เช่น Pooling layer แล้วนำกลุ่ม Layer ดังกล่าวมาซ้อนต่อกัน โดยอาจเปลี่ยน Hyperparameter บางอย่าง เช่นขนาดของ Filter layer (ซึ่งเป็นส่วนหนึ่งของ Convolution layer) และจำนวน Channel ของ layer วิธีการนำเอาส่วนต่างๆ มาประกอบกันนี้ เรียกว่าเป็นโครงสร้าง (Architecture) ของ CNN ซึ่งมีหลายแบบ เช่น LeNet, AlexNet, VGG, ResNet, Inception Network เป็นต้น ส่วนประกอบต่างๆ ของ CNN ซึ่งเป็นพื้นฐานที่เป็นส่วนสำคัญในการทำงานของ CNN ดังนี้

1) Convolution layer

$$2*1 + 4*0 + 1*1 + 1*1 + 1*0 + 6*1 + 7*1 + 6*0 + 4*1 = -1$$

2	4	1	0	5	3
1	1	6	4	2	3
7	6	4	2	1	0
6	9	2	1	8	9
4	1	1	4	5	7
0	5	3	2	1	7

 \star

1	0	-1
1	0	-1
1	0	-1

 $=$

-1			

$W^{[l]}$

$W^{[l]} a^{[l-1]}$

$a^{[l-1]}$

ภาพประกอบที่ 2.2 ตัวอย่างการคำนวณ Convolution

จากภาพประกอบที่ 2.2 สมมติเรามี Matrix ข่ายมือ ขนาด 6x6 และมี Matrix ตรงกลาง ซึ่งเรียกว่า Filter หรือ Kernel ขนาด 3x3 เราจะนำเฉพาะ 3x3 ช่องแรกของ Matrix แรก มาคูณแบบ Element-wise กับ Filter matrix แล้วนำผลที่ได้แต่ละค่า (ซึ่งมีทั้งสิ้น 9 ค่า) มาบวกกัน แล้วนำไปใส่ในแถวแรกคอลัมน์แรกของ Matrix ที่สามซึ่งเป็นผลลัพธ์ โดยในภาพ ผลลัพธ์ที่วางเท่ากับ -1

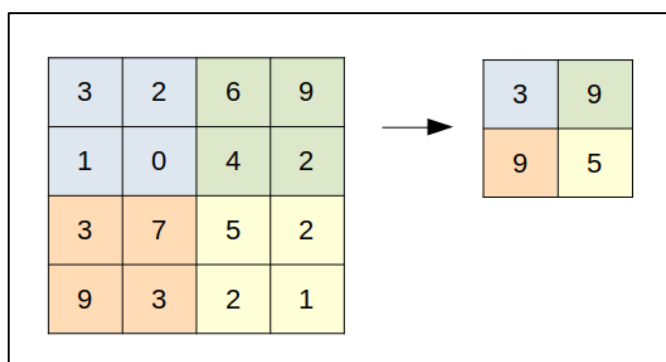
ถัดมา เราจะเลื่อนกรอบขนาด 3x3 ใน Matrix แรกไปทางขวา 1 ช่อง แล้วทำแบบเดิม ผลลัพธ์ที่ได้ นำไปใส่ในแถว 1 ช่อง 2 ของ Matrix ผลลัพธ์ ทำไปเรื่อยๆ จนสุดทาง แล้วเลื่อนกรอบ 3x3 ลงมาด้านล่าง 1 ช่อง (ชิดขอบด้านซ้ายมือ) แล้วทำแบบเดิม จนกระทั่งเติมค่าใน Matrix ผลลัพธ์จนเต็ม

กระบวนการนี้ เรียกว่า Convolution ซึ่งแสดงสัญลักษณ์ด้วย \star ส่วน Neural network ที่มี Layer ที่ใช้กระบวนการ Convolution นี้อย่างน้อย 1 Layer เราก็เรียกว่า Convolutional neural network

2) Pooling layer

หลังจากที่ข้อมูลผ่าน Convolution layer แล้ว บ่อยครั้งที่จะถูกส่งเข้า Layer อีกแบบหนึ่งที่เรียกว่า Pooling layer

หน้าที่ของ Pooling layer คือการสกัดเอาส่วนที่สำคัญที่สุดของข้อมูล และเพิ่มประสิทธิภาพการประมวลผลให้รวดเร็วยิ่งขึ้น กลไกของ Pooling layer นั้นเรียบง่ายมาก คือการสกัดเอาเฉพาะค่าสูงสุดของ Grid เก็บไว้ใน Output เช่นจากภาพประกอบที่ 2.3 แสดง Pooling layer ขนาด 2x2 โดยมีค่า Stride s=2:

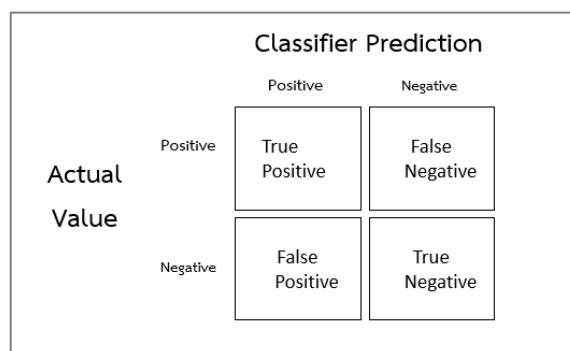


ภาพประกอบที่ 2.3 ตัวอย่างการทำ Pooling layer

Pooling layer ที่สกัดเอาเฉพาะค่าสูงสุดของ Grid เก็บไว้ เรียกว่า Max pooling ซึ่งเป็นรูปแบบที่ใช้บ่อยที่สุด นอกจากนั้นยังมี Average pooling ซึ่งหาค่าเฉลี่ยของ Grid เก็บไว้ แต่ใช้น้อยกว่า Max pooling มาก หลังจากที่ทำ Pooling layer เสร็จ ก็จะได้ feature map หรือ feature vector ที่จะนำไปทำเป็น model สำหรับทดสอบกับชุดข้อมูลอื่นๆ

2.3.9 การประเมิน (Evaluation)

ขั้นตอนการประเมินโมเดลเพื่อใช้ในการจัดการกลุ่มเอกสารก่อนนำไปใช้งานจริงที่โดยทั่วไปจะใช้เทคนิคมาตรฐาน [22] ที่เรียกว่า การวัดค่าความระลึก (Recall) การวัดค่าความแม่นยำ (Precision) และการวัดค่า F-measure



ภาพประกอบที่ 2.4 ตาราง Confusion Matrix

- True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่าจริง และคนบอกว่าจริง
- True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และคนบอกว่าไม่จริง
- False Positive (FP) คือ สิ่งที่โปรแกรมบอกว่าจริง แต่คนบอกว่าไม่จริง
- False Negative (FN) คือ สิ่งที่โปรแกรมบอกว่าไม่จริง แต่คนบอกว่าจริง

โดยนำค่าตาราง Confusion matrix มาใช้ในการคำนวณหาค่าความระลึก ค่าความแม่นยำ และค่า F-measure ได้ดังสมการต่อไปนี้

การวัดค่าความระลึก (Recall) [22] คือ เป็นอัตราส่วนของเอกสารที่จัดกลุ่มได้ จากเอกสารทั้งหมดที่มีอยู่ โดยจะนำค่าจากตาราง Confusion matrix มาใช้ในการคำนวณหาค่าความระลึก ได้ดังนี้

$$Recall = \frac{tp}{tp + fn} \quad (10)$$

การวัดค่าความแม่นยำ (Precision) [22] คือ เป็นอัตราส่วนของเอกสารที่จัดกลุ่มได้และถูกต้อง ส่วนด้วยจำนวนเอกสารที่จัดกลุ่มได้

$$Precision = \frac{tp}{tp + fp} \quad (11)$$

การวัดค่า F-measure [22] เป็นการพิจารณาค่าความสัมพันธ์ระหว่างค่าความระลึกและค่าความแม่นยำ

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

โดยที่ค่า F จะมีค่าระหว่าง 0 ถึง 1 ซึ่งถ้าหากค่า F มีค่าเข้าใกล้ 1 มากเท่าไรก็จำหมายถึงการจัดกลุ่มเอกสารมีประสิทธิภาพและมีความถูกต้องมากขึ้นเท่านั้น

2.4 งานวิจัยที่เกี่ยวข้อง (Related work)

ในการจำแนกความรู้สึกของเอกสารข้อความก็พบปัญหาของข้อมูลที่ไม่สมดุล ซึ่ง Li และคณะ [2] ได้ศึกษาเกี่ยวกับข้อมูลที่ไม่สมดุลหลายรูปแบบ เช่น จำนวนเอกสารที่ไม่สมดุล ขนาดของคลาสที่ไม่สมดุล รวมถึงความไม่สมดุลในคลาสย่อย จากการศึกษาที่ต่อเนื่องพบว่า ประเด็นที่หนึ่ง จำนวนเอกสารข้อความในสองคลาสจะเท่ากัน ความแตกต่างของจำนวนคำในเอกสารกลายเป็นปัจจัยสำคัญที่มีผลต่อความถูกต้องของการจำแนกเอกสาร ประเด็นที่สอง เพื่อปรับปรุงความถูกต้องของการจำแนกเอกสารด้วยการเพิ่มจำนวนของกลุ่มข้อมูลที่มีจำนวนน้อย และประเด็นที่สาม ในกรณีของข้อมูลที่ไม่สมดุล ค่าเดียวกันที่ปรากฏในสองคลาสมักจะเป็นสารสนเทศสำคัญของคลาส นั่นคือ คลาสทับซ้อนกันจะไม่ส่งผลกระทบต่อความถูกต้องของการจัดประเภท

Flavio Carvalho และ Gustavo Pai Guedes ได้นำเสนอการให้น้ำหนักคำแบบ Supervised Term Weighting ที่เหมาะสมต่อการจำแนกความรู้สึกที่ไม่สมดุล โดยได้นำเสนอการให้น้ำหนักคำที่ได้รับการควบคุมดูแลเจ็ดชุดและแผนการกำหนดน้ำหนัก ซึ่งวิธีนี้เป็นวิธีที่มีประสิทธิภาพมากกว่าการให้น้ำหนักคำในแบบ Unsupervised Term Weighting เนื่องจากการให้น้ำหนักคำในรูปแบบนี้เป็นใช้ประโยชน์จากข้อมูลที่อยู่ในคลังข้อมูลการฝึกอบรม

ในปี ค.ศ. 2011 Shoushan Li และคณะได้ทำงานวิจัย Imbalance Sentiment Classification [23] เพราะเล็งเห็นปัญหาในการจำแนกความรู้สึกที่ไม่สมดุลของข้อมูล เนื่องจากวิธีก่อนหน้านี้มีปัญหาในการทำงานค่อนข้างมาก จึงได้นำเสนอ วิธีการจำแนกความรู้สึกที่ไม่สมดุล โดยเสนอโครงร่างการจัดกลุ่มแบบ under-sampling ด้วยการแบ่งเป็นกลุ่มเพื่อเอาชนะปัญหาการกระจายระดับความไม่สมดุลในการจำแนกความรู้สึกที่ไม่สมดุล ภายใต้กรอบงานนี้ กลุ่มตัวอย่างในกลุ่มเสียงส่วนใหญ่จะถูกจัดกลุ่มเป็นกลุ่มแรก จากนั้นเลือกกลุ่มตัวอย่างจำนวนที่เหมาะสมจากแต่ละกลุ่มจากตัวอย่างการฝึกอบรมของข้อมูลส่วนใหญ่

ในงานวิจัยของ Ah-Pine และ Pavel Soriano Morales [6] ศึกษาแก้ปัญหาความไม่สมดุลของข้อมูลในการวิเคราะห์ความรู้สึก (Sentiment Classification) ที่ใช้ข้อมูลจาก twitter ที่พบว่าการกระจายกลุ่มของข้อมูลมีความเอนเอียงไปกลุ่มใดกลุ่มหนึ่ง นั่นคือจำนวนข้อมูลในแต่ละกลุ่มขาดความสมดุล ดังนั้นนักวิจัยจึงนำเสนอการทำเทคนิคการสุ่มตัวอย่างแบบสังเคราะห์ (Synthetic Oversampling Techniques) สำหรับการจำแนกกลุ่มข้อความ Twitter

อย่างไรก็ตาม งานวิจัยส่วนใหญ่ที่ใช้ในการแก้ปัญหาข้อมูลไม่สมดุลในการจำแนกเอกสารมันทำผ่านการคัดเลือกคุณลักษณะที่เหมาะสม (Feature Selection) เช่น งาน Zheng และคณะ นำเสนอการศึกษาเรื่องการคัดเลือกเอกสารที่เหมาะสม เพื่อเพิ่มประสิทธิภาพในการจำแนกเอกสารข้อความที่มีประสิทธิภาพ โดยทั่วไป information gain (IG), chi-square (CHI), correlation coefficient (CC) และ odds ratios (OR) ล้วนเป็นเทคนิคในการคัดเลือกคุณลักษณะที่มีประสิทธิภาพ CC และ OR เป็นตัวชี้วัดด้านเดียว (one-sided metrics) ในขณะที่ IG และ CHI เป็นแบบสองด้าน (two-sided metrics) การเลือกคุณสมบัติโดยใช้การวัดด้านเดียวเลือกคุณลักษณะที่บ่งบอกถึงการเป็นสมาชิก (membership) มากที่สุดเท่านั้น ในขณะที่การเลือกคุณลักษณะโดยใช้การวัดสองด้านโดยนิยรมคุณลักษณะที่บ่งบอกถึงการเป็นสมาชิกมากที่สุด (เช่น คุณสมบัติเชิงบวก) ด้วยการไม่สนใจร่องรอยหรือเครื่องหมายของคุณลักษณะ

ซึ่งในการศึกษาที่ผ่านมาจะไม่ให้ความสำคัญกับคุณลักษณะเชิงลบ (negative features) ที่ค่อนข้างมีความสำคัญ ในขณะที่ต่อมา พบว่าการผสมผสานคุณสมบัติทั้งเชิงบวกและเชิงลบจะสามารถเพิ่มประสิทธิภาพในการจำแนกเอกสาร โดยเฉพาะอย่างยิ่งกับข้อมูลที่ไม่สมดุล ในงานวิจัยนี้ นักวิจัยได้ศึกษาเกี่ยวกับกระบวนการในการคัดเลือกเอกสารที่มีการควบคุมคุณสมบัติทั้งเชิงบวกและเชิงลบอย่างเหมาะสม ขณะที่มีการใช้ multinomial naïve Bayes และ regularized logistic regression ในการ

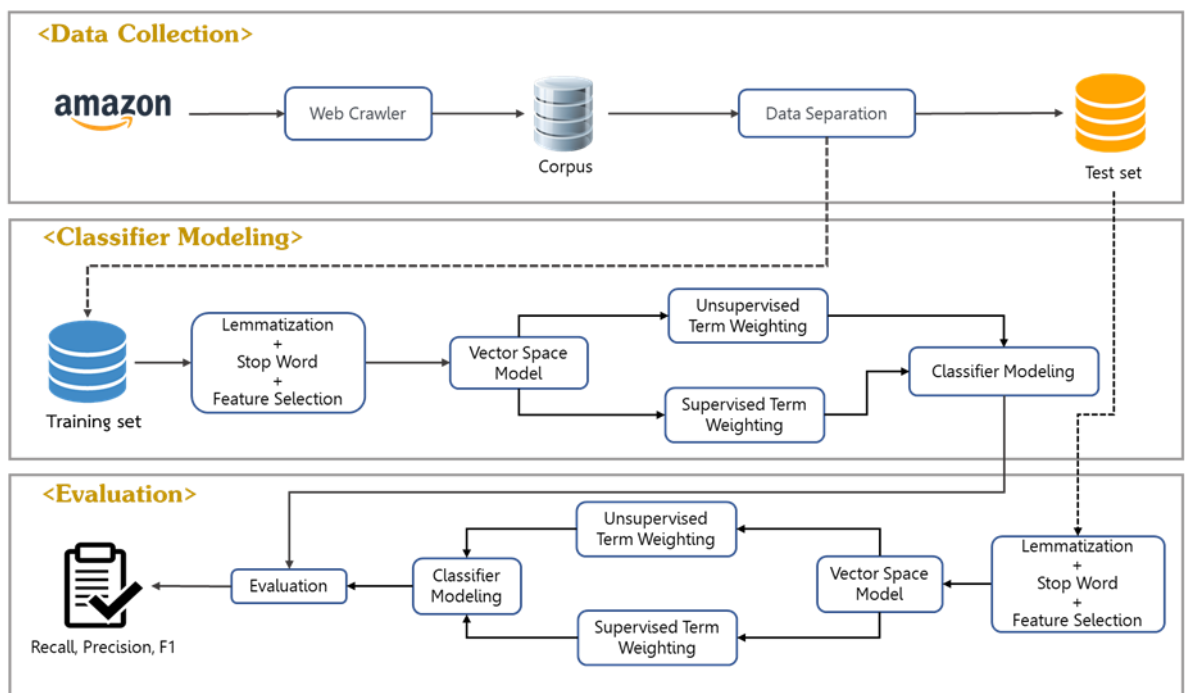
สร้างตัวจำแนกเอกสาร ผลลัพธ์ที่ได้จากการทดสอบแสดงให้เห็นกระบวนการคัดเลือกคุณลักษณะในการรวมคุณสมบัติบวกและลบในการแก้ปัญหาข้อมูลที่ไม่สมดุลได้ให้ประสิทธิภาพที่ดี

บทที่ 3

วิธีดำเนินงานวิจัย

ในบทนี้จะอธิบายถึงชุดข้อมูลข้อความแสดงความคิดเห็นที่เกี่ยวกับอุปกรณ์อิเล็กทรอนิกส์ ซึ่งรวบรวมมาจากเว็บไซต์ Amazon ที่ใช้ในโครงงานนี้ และวิธีการดำเนินงาน ดังนี้

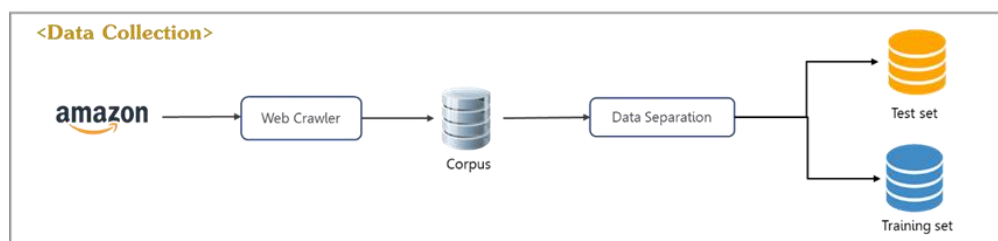
3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 3.1 กรอบการดำเนินงานของระบบ

ภาพรวมของระบบการควบคุมข้อมูลไม่สมดุลในการจำแนกความรู้สึก จะแบ่งการทำงานออกเป็น 3 ส่วนหลัก คือ

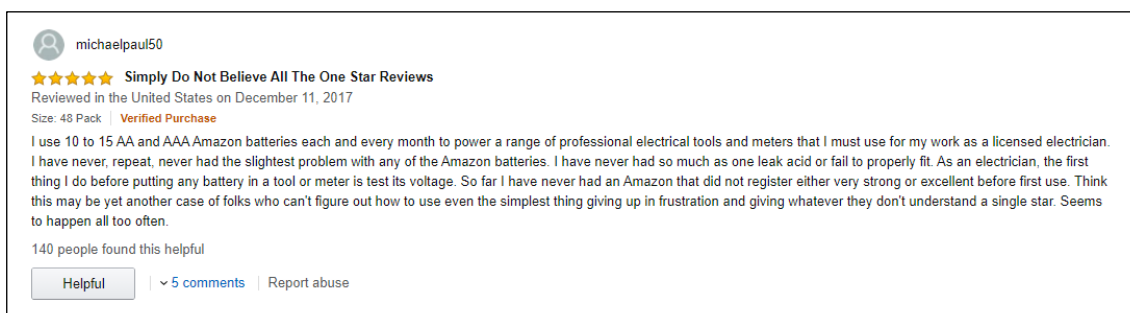
3.2 ชุดข้อมูล (Data set)



ภาพประกอบที่ 3.2 Data Collection

ในส่วนนี้ เป็นส่วนของการเก็บรวบรวมข้อมูล ในโครงการปัญญาประดิษฐ์นี้เป็นข้อความแสดงความคิดเห็นที่เกี่ยวกับข้อความแสดงความคิดเห็นที่เกี่ยวกับอุปกรณ์อิเล็กทรอนิกส์ ซึ่งรวบรวมจากเว็บไซต์ Amazon โดยแบ่งเป็น 2 รูปแบบ คือ ข้อความแสดงความคิดเห็นที่เป็นเชิงบวก (Positive) และข้อความแสดงความคิดเห็นที่เป็นเชิงลบ (Negative) และในการเตรียมข้อมูล จะใช้ข้อมูล Train อย่างน้อย 1000 บทวิจารณ์ต่อกลุ่มความคิดเห็น และใช้ข้อมูลชุดทดสอบ (Test) อย่างน้อย 200 บทวิจารณ์ต่อกลุ่มความคิดเห็น

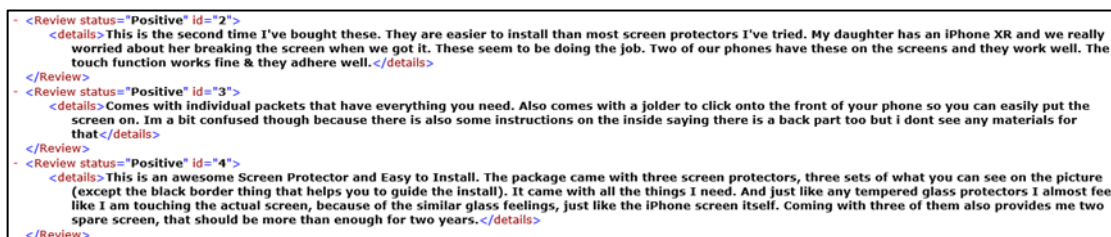
โดยในโครงการปัญญาประดิษฐ์นี้ ได้ใช้ชุดข้อความแสดงความคิดเห็นที่เกี่ยวกับอุปกรณ์อิเล็กทรอนิกส์ ซึ่งรวบรวมมาจากเว็บไซต์ Amazon โดยจะมีการแบ่งเองสารออกเป็น 2 ชุด คือ ชุดข้อมูลสอน (Training set) และ ชุดข้อมูลทดสอบ (Test set) ซึ่งเอกสารจะอยู่ในรูปแบบ XML ข้อมูลที่ใช้ทั้งหมด 50,000 ความคิดเห็นและมีค่าระหว่าง 30 ถึง 300 คำต่อหนึ่งเอกสารข้อความแสดงความคิดเห็น



ภาพประกอบที่ 3.3 ตัวอย่างเอกสารข้อความแสดงความคิดเห็น

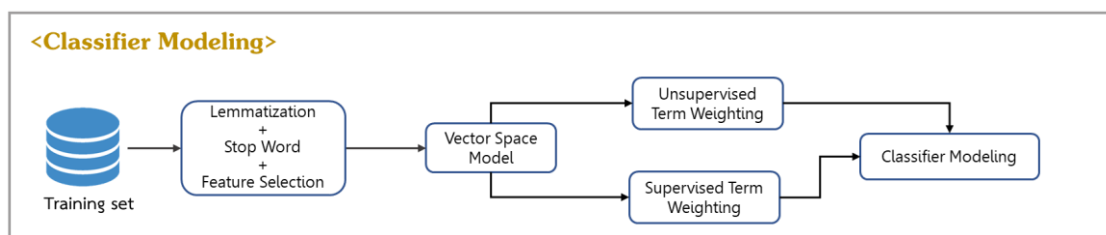
ที่มา : <https://www.amazon.com/AmazonBasics-Performance-Alkaline-Batteries-Count/product-reviews/B00MNV8E0C/>

จากภาพประกอบที่ 3.3 เป็นตัวอย่างเอกสารข้อความแสดงความคิดเห็นจากเว็บ Amazon ที่ใช้ในงานวิจัยนี้โดยจะทำการดาวน์โหลดออกมาในรูปแบบ XML ซึ่งจะประกอบไปด้วย รหัส (ID), สถานะ (Status) และเนื้อหาของเอกสาร (details) ดังภาพประกอบที่ 3.4



ภาพประกอบที่ 3.4 ตัวอย่างเอกสารที่อยู่ในรูปแบบ XML

3.3 การสร้างโมเดลเพื่อการจำแนกความรู้สึกของบทวิจารณ์ (Classifier Modeling)



ภาพประกอบที่ 3.5 Classifier Modeling

ในการสร้างโมเดลเพื่อจำแนกความรู้สึกของบทวิจารณ์ จะมีขั้นตอนหลักในการประมวลผลดังนี้

3.3.1 การเตรียมข้อมูลก่อนการประมวลผล

ในขั้นตอนก่อนการประมวลผล จะเป็นการเตรียมเอกสารหรือบทความให้อยู่ในรูปแบบที่พร้อมจะนำไปประมวลผลในขั้นตอนถัดไปได้ ซึ่งจะมีขั้นตอนดังนี้

สมมติให้มีเอกสารบทวิจารณ์เกี่ยวกับอุปกรณ์อิเล็กทรอนิกส์ 6 เอกสาร ได้แก่

D_1 : One of worst electronic items.

D_2 : That's the worst electronic device ever used.

D_3 : Bad HDMI.

D_4 : So good

D_5 : This's a Good electric device!

D_6 : Best device

ขั้นตอนที่ 1 : การตัดคำและการตัดคำหยุด (Stop-word Removal) เป็นกระบวนการตัดคำหรือสัญลักษณ์ที่พบบ่อยมากในเอกสาร แต่คำหรือสัญลักษณ์เหล่านั้นไม่ได้ส่งผลต่อการจัดกลุ่มเอกสาร

ตัวอย่างเอกสารหลังจากทำการตัดคำ

D_1 : one / of / worst / electrical / items

D_2 : that / 's / the / worst / electric / device / ever / used

D_3 : bad / electric

D_4 : so / good

D_5 : good / this / 's / a / electronic / device

D_6 : best / device

ตัวอย่างเอกสารหลังจากทำการตัดคำหยุด

D_1 : worst / electrical / items

D_2 : worst / electric / device

D_3 : bad / electric

D_4 : so / good

D_5 : good / electronic / device

D_6 : best / device

ขั้นตอนที่ 2 : การทำ Lemmatization Tagging จะเป็นการเปลี่ยนคำให้อยู่ในรูปแบบดั้งเดิม โดยมีขั้นตอนดังนี้

1. TokenizerAnnotator เป็นกระบวนการตัดคำโดยใช้หลักการเดียวกันกับ Penn Treebank

D_1 : worst / electrical / items

D_2 : worst / electric / device

D_3 : bad / electric

D_4 : so / good

D_5 : good / electronic / device

D_6 : best / device

2. ssplit เป็นการนำคำที่ผ่านกระบวนการตัดคำมาเรียงลำดับตามประโยคเดิม

D_1 : worst / electrical / items

D_2 : worst / electric / device

D_3 : bad / electric

D_4 : so / good

D_5 : good / electronic / device

D_6 : best / device

3. POS (Part-Of-Speech Tagging) เป็นการติด tag ให้กับคำแต่ละคำ โดยใช้ Penn Treebank Tagset

D_1 : worst (JJS) | electronic (JJ) | items (NNS)
 D_2 : worst (JJS) | electronic (JJ) | device (NN)
 D_3 : bad (JJ) | electric (JJ)
 D_4 : good (JJ)
 D_5 : electronic (JJ) | device (NN)
 D_6 : Best (RB) | device (NN)

4. Lemma เป็นการนำคำที่ได้ภายหลังการติด tag มาทำ lemma โดยใช้ Wordnet

D_1 : worst / electrical / items
 D_2 : worst / electric / device
 D_3 : bad / electric
 D_4 : good
 D_5 : good / electronic / device
 D_6 : best / device

ขั้นตอนที่ 3 : การนำคำที่ได้จากขั้นตอนที่ 2 ไปเปรียบเทียบกับคำใน Dictionary หาก คำนั้น ไม่มีใน Dictionary จะทำการตัดคำนั้นทิ้ง เช่น คำว่า “hdmi” ซึ่งเป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์ และไม่ใช่คำหยุด เป็นต้น

ขั้นตอนที่ 4 : การสร้างตัวแทนเอกสาร จะเป็นการนำเสนอความสัมพันธ์ระหว่างคำและเอกสาร ในรูปแบบเวกเตอร์ จากขั้นตอนที่ 3 สามารถแสดงในรูปของ BOW ได้ดังนี้

ตารางที่ 3.1 แสดงการนำเสนอความสัมพันธ์ระหว่างคำและเอกสาร

W_i	worst	electric	bad	good	best	items	device
D_1	1	1	0	0	0	1	0
D_2	1	1	0	0	0	0	1
D_3	0	1	1	0	0	0	0
D_4	0	0	0	1	0	0	0
D_5	0	1	0	1	0	0	1
D_6	0	0	0	0	1	0	1

จากตารางที่ 3.1 จะเห็นว่า BOW นอกจากจะแสดงความสัมพันธ์ระหว่างคำและเอกสารแล้ว ยังสามารถแสดงให้เห็นความถี่ของคำที่ปรากฏในเอกสารนั้นๆ อีกด้วย

ขั้นตอนที่ 5 : การเลือกคุณลักษณะด้วย Information Gain เพื่อตัดคำที่ไม่มีนัยสำคัญออกเพื่อให้โมเดลมีประสิทธิภาพและลดระยะเวลาในการประมวลผลลง

คำนวณค่า $Info(D)$ หรือค่าเอนโทรปี (entropy) ของชุดข้อมูล (dataset: D) ที่กำลังศึกษาตามสมการที่ 13

$$Info(D) = - \sum_{i=1}^n P(c_i) * \log_2 P(c_i) \quad (13)$$

โดย D คือ ชุดข้อมูล

$P(c_i)$ คือ ความน่าจะเป็นของแต่ละคลาสในชุดข้อมูลนั้นๆ

\log คือ \log ฐาน 2

$$\begin{aligned} Info(D) &= - [(0.5) \log_2(0.5)] - [(0.5) \log_2(0.5)] \\ &= 1 \end{aligned}$$

จากนั้นคำนวณค่า $Info$ ของแต่ละ *sub-class* ในแต่ละ *Attribute* นั้นๆ ด้วย $info(attribute, a_i)$ ซึ่งเป็นฟังก์ชันที่ระบุปริมาณข้อมูลที่ต้องการเพื่อการจำแนก *class* ของข้อมูลโดยใช้ *attribute A* เป็นตัวตรวจสอบเพื่อแยกข้อมูลตามสมการที่ 14

$$Info(attribute, a_i) = \sum_{i=1}^n \frac{|a_i|}{|A|} * Info(a_i) \quad (14)$$

โดย A คือ จำนวนข้อมูลทั้งหมดใน *Attribute* ที่กำลังพิจารณา

a_i คือ *sub-class* ใน *Attribute* ที่กำลังพิจารณา

$|a_i|$ คือ จำนวนข้อมูลใน *sub-class* a_i

$$\begin{aligned} Info(Allword, worst) &= 2/7 \times [-(2/7 \times \log_2(2/7)) - (0/7 \times \log_2(0/7))] \\ &= 0.1475 \end{aligned}$$

$$\begin{aligned} Info(Allword, electric) &= 5/7 \times [-(3/7 \times \log_2(3/7)) - (1/7 \times \log_2(1/7))] \\ &= 0.7004 \end{aligned}$$

$$Info(Allword, bad) = 1/7 \times [-(1/7 \times \log_2(1/7)) - (0/7 \times \log_2(0/7))]$$

$$\begin{aligned}
&= 0.0572 \\
Info(Allword, good) &= 2/7 \times [-(0/7 \times \log_2(0/7)) - (2/7 \times \log_2(2/7))] \\
&= 0.1475 \\
Info(Allword, best) &= 1/7 \times [-(0/7 \times \log_2(0/7)) - (1/7 \times \log_2(1/7))] \\
&= 0.0572 \\
Info(Allword, items) &= 1/7 \times [-(1/7 \times \log_2(1/7)) - (0/7 \times \log_2(0/7))] \\
&= 0.0572 \\
Info(Allword, device) &= 3/7 \times [-(1/7 \times \log_2(1/7)) - (2/7 \times \log_2(2/7))] \\
&= 0.3931
\end{aligned}$$

เมื่อได้ค่า *Info* ของแต่ละคำหรือแอตทริบิวต์เรียบร้อยแล้ว ต่อไปจะเป็นการหาค่า Information Gain (*IG*) ของแต่ละคำนั้นๆ ด้วยสมการที่

$$Gain(A) = Info(D) - Info(Attribute, a_i) \quad (15)$$

โดย *Gain(A)* คือ ค่าความน่าเชื่อถือของคำนั้นๆ

$$\begin{aligned}
Gain_{worst} &= 1.0 - 0.1475 = 0.8525 \\
Gain_{electric} &= 1.0 - 0.7004 = 0.2996 \\
Gain_{bad} &= 1.0 - 0.0572 = 0.9428 \\
Gain_{good} &= 1.0 - 0.1475 = 0.8525 \\
Gain_{best} &= 1.0 - 0.0572 = 0.9428 \\
Gain_{items} &= 1.0 - 0.0572 = 0.9428 \\
Gain_{device} &= 1.0 - 0.0572 = 0.9428
\end{aligned}$$

เมื่อคำนวณค่า *Gain* ของแต่ละคำเสร็จเรียบร้อยแล้ว จะทำการเรียงค่า *Gain* จากมากไปหาน้อยเพื่อลดจำนวนคำลง โดยจะตัดคำที่ไม่มีนัยสำคัญต่อเอกสารออกด้วยการวัดค่า *Gain* หากค่าใดมีค่า *Gain* เป็น 0 จะถูกตัดทิ้งทั้งหมด จากตัวอย่างข้างต้นจะเห็นว่าไม่มีคำที่มีค่า *Gain* เป็น 0 นั้นหมายความว่า คำทุกคำในตัวอย่างมีความสำคัญต่อเอกสารทั้งหมด

ขั้นตอนที่ 6 : การให้น้ำหนักคำ (Term weighting) จะมี 2 รูปแบบหลัก ดังนี้

รูปแบบที่ 1 : Unsupervised Term Weighting (UTW) โดยในปริภูมิพจน์ฉบับนี้ จะใช้รูปแบบที่ได้รับความนิยมมากที่สุดของ UTW คือ *tf-idf* การให้น้ำหนักแบบ *tf-idf* เป็นวิธีการสร้างตัวแทนเอกสารในรูปแบบของเวกเตอร์เพื่อใช้ในการจัดกลุ่มของเอกสารให้ตรงกับหมวดหมู่ที่ถูกกำหนดไว้ โดย *tf* เป็นการหาความถี่ของคำหนึ่งๆ ที่พบในแต่ละเอกสาร และ *idf* ก็คือ global weight ที่เป็นการหาส่วนกลับของความถี่ของคำในเอกสาร หรือที่เรียกว่าระบบน้ำหนักความถี่เอกสารผกผัน โดยจะสามารถแสดงขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 : หาค่า *tf* ที่มีความถี่ของคำแต่ละคำที่อยู่ในเอกสารนั้นๆ ว่าพบกี่ครั้ง

ขั้นตอนที่ 2 : หาค่า *idf* คือการหาค่าส่วนกลับของแต่ละคำในเอกสารนั้นๆ

การคำนวณหา *idf* ทำได้โดยใช้สมการ $idf = \log(N/df)$ โดย N คือจำนวนเอกสารทั้งหมดในคลังเอกสาร และ df คือจำนวนเอกสารที่มีคำนั้นๆ ปรากฏอยู่ และสามารถคำนวณหาค่า *idf* ได้ดังนี้ในที่นี้จะให้ $N = 5$

$$idf_{\text{worst}} = \log(5/2) = 0.477$$

$$idf_{\text{electric}} = \log(5/4) = 0.176$$

$$idf_{\text{bad}} = \log(5/1) = 0.778$$

$$idf_{\text{good}} = \log(5/2) = 0.477$$

$$idf_{\text{best}} = \log(5/1) = 0.778$$

$$idf_{\text{items}} = \log(5/1) = 0.778$$

$$idf_{\text{device}} = \log(5/3) = 0.301$$

ขั้นตอนที่ 3 : การคำนวณหาค่า *tf-idf*

ในขั้นตอนนี้จะเป็นการนำเอาค่า *tf* ที่ได้คูณเข้ากับค่า *idf* เช่น ในเอกสารที่ 1 จะพบคำ 3 คำ คือ “worst” , “electric” และ “items” โดยค่าเหล่านี้ ที่ปรากฏในเอกสารที่ 1 มีค่า *tf* เป็น 1, 1 และ 1 ตามลำดับ เมื่อนำมาหาค่า *tf-idf* จะได้ผลลัพธ์ ดังต่อไปนี้

$$tf-idf_{\text{worst}} \text{ ในเอกสารที่ 1} = 1 \times 0.477 = 0.477$$

$$tf-idf_{\text{worst}} \text{ ในเอกสารที่ 2} = 1 \times 0.477 = 0.477$$

$$tf-idf_{\text{electric}} \text{ ในเอกสารที่ 1} = 1 \times 0.176 = 0.176$$

$$tf-idf_{electric} \text{ ในเอกสารที่ } 2 = 1 \times 0.176 = 0.176$$

$$tf-idf_{electric} \text{ ในเอกสารที่ } 3 = 1 \times 0.176 = 0.176$$

$$tf-idf_{electric} \text{ ในเอกสารที่ } 5 = 1 \times 0.176 = 0.176$$

$$tf-idf_{bad} \text{ ในเอกสารที่ } 3 = 1 \times 0.778 = 0.778$$

$$tf-idf_{best} \text{ ในเอกสารที่ } 6 = 1 \times 0.778 = 0.778$$

$$tf-idf_{good} \text{ ในเอกสารที่ } 4 = 1 \times 0.477 = 0.477$$

$$tf-idf_{good} \text{ ในเอกสารที่ } 5 = 1 \times 0.477 = 0.477$$

$$tf-idf_{items} \text{ ในเอกสารที่ } 1 = 1 \times 0.778 = 0.778$$

$$tf-idf_{device} \text{ ในเอกสารที่ } 2 = 1 \times 0.301 = 0.301$$

$$tf-idf_{device} \text{ ในเอกสารที่ } 5 = 1 \times 0.301 = 0.301$$

$$tf-idf_{device} \text{ ในเอกสารที่ } 6 = 1 \times 0.301 = 0.301$$

ตารางที่ 3.2 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ $tf-idf$

W_i	worst	electric	bad	good	best	items	device
D_1	0.477	0.176	0	0	0	0.788	0
D_2	0.477	0.176	0	0	0	0	0.301
D_3	0	0.176	0.778	0	0	0	0
D_4	0	0	0	0.477	0	0	0
D_5	0	0.176	0	0.477	0	0	0.301
D_6	0	0	0	0	0.788	0	0.301

รูปแบบที่ 2 : Supervised Term Weighting (STW) ในรูปแบบนี้จะมีทั้งหมด 4 รูปแบบ

1) Delta TF-IDF

Delta TF-IDF ช่วยเพิ่มสำคัญของคำที่กระจายอย่างไม่สม่ำเสมอระหว่างคลาสบวกและคลาสลบ โดยที่ N_p และ N_n คือจำนวนของเอกสารในคลาสบวกและลบ ในตัวอย่างของเรามีจำนวนเอกสารที่อยู่ในคลาสบวก 1 เอกสารและคลาสลบ 4 เอกสาร ส่วน A และ C แสดงความถี่เอกสารของคำว่า t_i ในคลาสบวกและลบตามลำดับ จากตารางที่ 3.1 สามารถนำมาคำนวณน้ำหนักค่าของ Delta TF-IDF ดังนี้

ขั้นตอนที่ 1 : หาค่า tf ที่เป็นความถี่ของคำแต่ละคำที่อยู่ในเอกสารนั้นๆ ว่าพบกี่ครั้ง

ขั้นตอนที่ 2 : หาค่า $\Delta TF-IDF$ ของแต่ละคำในเอกสารนั้นๆ ซึ่งสามารถคำนวณหาค่า $\Delta TF-IDF$ ของแต่ละเอกสาร ได้ดังนี้

$$W_{\&TF.IDF}(worst) \text{ ในเอกสารที่ 1} = 1 * \log_2 \left(\frac{3 * 0 + 1.5}{2 * 3 + 1.5} \right) = -2.321$$

$$W_{\&TF.IDF}(electric) \text{ ในเอกสารที่ 1} = 1 * \log_2 \left(\frac{3 * 1 + 1.5}{3 * 3 + 1.5} \right) = -1.222$$

$$W_{\&TF.IDF}(items) \text{ ในเอกสารที่ 1} = 1 * \log_2 \left(\frac{3 * 0 + 1.5}{1 * 3 + 1.5} \right) = -1.584$$

$$W_{\&TF.IDF}(worst) \text{ ในเอกสารที่ 2} = 1 * \log_2 \left(\frac{3 * 0 + 1.5}{2 * 3 + 1.5} \right) = -2.321$$

$$W_{\&TF.IDF}(electric) \text{ ในเอกสารที่ 2} = 1 * \log_2 \left(\frac{3 * 1 + 1.5}{3 * 3 + 1.5} \right) = -1.222$$

$$W_{\&TF.IDF}(device) \text{ ในเอกสารที่ 2} = 1 * \log_2 \left(\frac{3 * 2 + 1.5}{1 * 3 + 1.5} \right) = 0.736$$

$$W_{\&TF.IDF}(electric) \text{ ในเอกสารที่ 3} = 1 * \log_2 \left(\frac{3 * 1 + 1.5}{3 * 3 + 1.5} \right) = -1.222$$

$$W_{\&TF.IDF}(bad) \text{ ในเอกสารที่ 3} = 1 * \log_2 \left(\frac{3 * 0 + 1.5}{1 * 3 + 1.5} \right) = -1.584$$

$$W_{\&TF.IDF}(good) \text{ ในเอกสารที่ 4} = 1 * \log_2 \left(\frac{3 * 0 + 1.5}{2 * 3 + 1.5} \right) = -2.321$$

$$W_{\&TF.IDF}(electric) \text{ ในเอกสารที่ 5} = 1 * \log_2 \left(\frac{3 * 3 + 1.5}{1 * 3 + 1.5} \right) = 1.222$$

$$W_{\&TF.IDF}(good) \text{ ในเอกสารที่ 5} = 1 * \log_2 \left(\frac{3 * 0 + 1.5}{2 * 3 + 1.5} \right) = -2.321$$

$$W_{\&TF.IDF}(device) \text{ ในเอกสารที่ 5} = 1 * \log_2 \left(\frac{3 * 1 + 1.5}{2 * 3 + 1.5} \right) = -0.893$$

$$W_{\&TF.IDF}(best) \text{ ในเอกสารที่ 6} = 1 * \log_2 \left(\frac{3 * 0 + 1.5}{1 * 3 + 1.5} \right) = -1.584$$

$$W_{\&TF.IDF}(device) \text{ ในเอกสารที่ 6} = 1 * \log_2 \left(\frac{3 * 1 + 1.5}{2 * 3 + 1.5} \right) = -0.736$$

ตารางที่ 3.3 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ $\Delta TF-IDF$

W_i	worst	electric	bad	good	best	items	device
D_1	-2.321	-1.222	0	0	0	-1.584	0
D_2	-2.321	-1.222	0	0	0	0	0.736
D_3	0	-1.222	-1.584	0	0	0	0
D_4	0	0	0	-2.321	0	0	0
D_5	0	1.222	0	-2.321	0	0	-0.736
D_6	0	0	0	0	-1.584	0	-0.736

2) TF-IDF-ICF

TF-IDF-ICF เป็นรูปแบบการควบคุมน้ำหนักตามแบบ TF-IDF แบบดั้งเดิม โดยเพิ่มปัจจัยความถี่ผกผันในคลาส (Inverse Class Frequency: ICF) เพื่อให้ค่าน้ำหนักคำสูงขึ้นสำหรับคำหายากที่เกิดขึ้นน้อยในเอกสารและคลาส โดย M คือจำนวนคลาสในที่นี่เท่ากับ 2 จากตารางที่ 3.1 สามารถนำมาคำนวณน้ำหนักคำของ TF-IDF-ICF ดังนี้

ขั้นตอนที่ 1 : หาค่า tf ที่เป็นความถี่ของคำแต่ละคำที่อยู่ในเอกสารนั้นๆ

ขั้นตอนที่ 2 : หาค่า idf คือการหาค่าส่วนกลับของแต่ละคำในเอกสารนั้นๆ

ขั้นตอนที่ 3 : หาค่า icf คือปัจจัยความถี่ผกผันในคลาสของแต่ละคำในเอกสารนั้นๆ

$$ICF(worst) = 1 + \log(2/1) = 1.301$$

$$ICF(electric) = 1 + \log(2/2) = 1.000$$

$$ICF(bad) = 1 + \log(2/1) = 1.301$$

$$ICF(good) = 1 + \log(2/1) = 1.301$$

$$ICF(best) = 1 + \log(2/1) = 1.301$$

$$ICF(items) = 1 + \log(2/1) = 1.301$$

$$ICF(device) = 1 + \log(2/2) = 1.000$$

ขั้นตอนที่ 4 : หาค่า TF-IDF-ICF ของแต่ละคำในเอกสารนั้นๆ

$$W_{TF,ICF}(worst) \text{ ในเอกสารที่ 1} = 1 \times 0.477 \times 1.301 = 0.620$$

$$W_{TF,ICF}(electric) \text{ ในเอกสารที่ 1} = 1 \times 0.176 \times 1.000 = 0.176$$

$$W_{TF,ICF}(items) \text{ ในเอกสารที่ 1} = 1 \times 0.778 \times 1.301 = 1.012$$

$$W_{TF,ICF}(worst) \text{ ในเอกสารที่ 2} = 1 \times 0.477 \times 1.301 = 0.620$$

$$W_{TF,ICF}(electric) \text{ ในเอกสารที่ 2} = 1 \times 0.176 \times 1.000 = 0.176$$

$$W_{TF,ICF}(device) \text{ ในเอกสารที่ 2} = 1 \times 0.301 \times 1.000 = 0.301$$

$$W_{TF,ICF}(electric) \text{ ในเอกสารที่ 3} = 1 \times 0.176 \times 1.000 = 0.176$$

$$W_{TF,ICF}(bad) \text{ ในเอกสารที่ 3} = 1 \times 0.778 \times 1.301 = 1.012$$

$$W_{TF,ICF}(good) \text{ ในเอกสารที่ 4} = 1 \times 0.477 \times 1.301 = 0.620$$

$$W_{TF,ICF}(electric) \text{ ในเอกสารที่ 5} = 1 \times 0.176 \times 1.000 = 0.176$$

$$W_{TF,ICF}(good) \text{ ในเอกสารที่ 5} = 1 \times 0.477 \times 1.301 = 0.620$$

$$W_{TF,ICF}(device) \text{ ในเอกสารที่ 5} = 1 \times 0.301 \times 1.000 = 0.301$$

$$W_{TF,ICF}(best) \text{ ในเอกสารที่ 6} = 1 \times 0.778 \times 1.301 = 1.012$$

$$W_{TF,ICF}(device) \text{ ในเอกสารที่ 6} = 1 \times 0.301 \times 1.000 = 0.301$$

ตารางที่ 3.4 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ TF-IDF-ICF

W_i	worst	electric	bad	good	best	items	device
D_1	0.620	0.176	0	0	0	1.012	0
D_2	0.620	0.176	0	0	0	0	0.301
D_3	0	0.176	1.012	0	0	0	0
D_4	0	0	0	0.620	0	0	0
D_5	0	0.176	0	0.620	0	0	0.301
D_6	0	0	0	0	1.012	0	0.301

3) TF-RF

TF-RF มีความเกี่ยวข้องของควมถี่ (RF) ของข้อกำหนด TF-RF โดยที่ตัวหารน้อยที่สุดคือ 1 เพื่อหลีกเลี่ยงการหารด้วยศูนย์ จากตารางที่ 3.1 สามารถนำมาคำนวณน้ำหนักค่าของ TF-IDF-ICF ดังนี้

ขั้นตอนที่ 1 : หาค่า tf ที่เป็นความถี่ของคำแต่ละคำที่อยู่ในเอกสารนั้นๆ

ขั้นตอนที่ 2 : หาค่า $TF-RF$ คือการหาค่าส่วนกลับของแต่ละคำในเอกสารนั้นๆ

$$W_{TF,RF}(worst) \quad \text{ในเอกสารที่ 1} = 1 * \log_2 \left(2 + \frac{2}{\max(1,0)} \right) = 2.000$$

$$W_{TF,RF}(electric) \quad \text{ในเอกสารที่ 1} = 1 * \log_2 \left(2 + \frac{3}{\max(1,1)} \right) = 2.321$$

$$W_{TF,RF}(items) \quad \text{ในเอกสารที่ 1} = 1 * \log_2 \left(2 + \frac{1}{\max(1,0)} \right) = 1.584$$

$$W_{TF,RF}(worst) \quad \text{ในเอกสารที่ 2} = 1 * \log_2 \left(2 + \frac{2}{\max(1,0)} \right) = 2.000$$

$$W_{TF,RF}(electric) \quad \text{ในเอกสารที่ 2} = 1 * \log_2 \left(2 + \frac{3}{\max(1,1)} \right) = 2.321$$

$$W_{TF,RF}(device) \quad \text{ในเอกสารที่ 2} = 1 * \log_2 \left(2 + \frac{1}{\max(1,2)} \right) = 1.584$$

$$W_{TF,RF}(electric) \quad \text{ในเอกสารที่ 3} = 1 * \log_2 \left(2 + \frac{3}{\max(1,1)} \right) = 2.321$$

$$W_{TF,RF}(bad) \quad \text{ในเอกสารที่ 3} = 1 * \log_2 \left(2 + \frac{1}{\max(1,0)} \right) = 1.584$$

$$W_{TF,RF}(good) \quad \text{ในเอกสารที่ 4} = 1 * \log_2 \left(2 + \frac{2}{\max(1,0)} \right) = 2.000$$

$$W_{TF,RF}(electric) \quad \text{ในเอกสารที่ 5} = 1 * \log_2 \left(2 + \frac{1}{\max(1,3)} \right) = 1.736$$

$$W_{TF,RF}(good) \quad \text{ในเอกสารที่ 5} = 1 * \log_2 \left(2 + \frac{2}{\max(1,0)} \right) = 2.000$$

$$W_{TF,RF}(device) \quad \text{ในเอกสารที่ 5} = 1 * \log_2 \left(2 + \frac{2}{\max(1,1)} \right) = 2.000$$

$$W_{TF,RF}(best) \quad \text{ในเอกสารที่ 6} = 1 * \log_2 \left(2 + \frac{1}{\max(1,0)} \right) = 1.584$$

$$W_{TF,RF}(device) \text{ ในเอกสารที่ 6} = 1 * \log_2 \left(2 + \frac{2}{\max(1,1)} \right) = 2.000$$

ตารางที่ 3.5 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ TF-RF

W_i	worst	electric	bad	good	best	items	device
D_1	2.000	2.321	0	0	0	1.584	0
D_2	2.000	2.321	0	0	0	0	1.584
D_3	0	2.321	1.584	0	0	0	0
D_4	0	0	0	2.000	0	0	0
D_5	0	1.736	0	2.000	0	0	2.000
D_6	0	0	0	0	1.584	0	2.000

4) TF-IGM

ระยะความถี่-ช่วงเวลาแรงโน้มถ่วงผกผัน (Term Frequency - Inverse Gravity Moment : TF-IGM) [20] ถูกนำเสนอให้วัดความไม่สม่ำเสมอหรือความเข้มข้นของการแจกแจงคำศัพท์ระหว่างคลาส โดยสามารถนำมาคำนวณได้ดังนี้

ขั้นตอนที่ 1 : หาค่า tf ที่ เป็นความถี่ของคำแต่ละคำที่อยู่ในเอกสารนั้นๆ

ขั้นตอนที่ 2 : หาค่า igm ที่ เป็นความถี่ของคำแต่ละคำที่กระจายอยู่ในแต่ละคลาส

$$igm(worst) = \frac{2}{(1 \times 2) + (2 \times 0)} = 1.0$$

$$igm(electric) = \frac{3}{(1 \times 3) + (2 \times 1)} = 0.6$$

$$igm(bad) = \frac{1}{(1 \times 1) + (2 \times 0)} = 1.0$$

$$igm(good) = \frac{2}{(1 \times 2) + (2 \times 0)} = 1.0$$

$$igm(best) = \frac{1}{(1 \times 1) + (2 \times 0)} = 1.0$$

$$igm(items) = \frac{1}{(1 \times 1) + (2 \times 0)} = 1.0$$

$$igm(device) = \frac{2}{(1 \times 2) + (2 \times 1)} = 0.5$$

ขั้นตอนที่ 4 : หาค่า TF-IGM ของแต่ละคำในเอกสารนั้นๆ

$$IGM(worst) \text{ ในเอกสารที่ 1} = 1 \times (1 + 7.0 \times 1.0) = 8.0$$

$$IGM(electric) \text{ ในเอกสารที่ 1} = 1 \times (1 + 7.0 \times 0.6) = 5.2$$

$$IGM(items) \text{ ในเอกสารที่ 1} = 1 \times (1 + 7.0 \times 1.0) = 8.0$$

$$IGM(worst) \text{ ในเอกสารที่ 2} = 1 \times (1 + 7.0 \times 1.0) = 8.0$$

$$IGM(electric) \text{ ในเอกสารที่ 2} = 1 \times (1 + 7.0 \times 0.6) = 5.2$$

$$IGM(device) \text{ ในเอกสารที่ 2} = 1 \times (1 + 7.0 \times 0.5) = 3.5$$

$$IGM(electric) \text{ ในเอกสารที่ 3} = 1 \times (1 + 7.0 \times 0.6) = 5.2$$

$$IGM(bad) \text{ ในเอกสารที่ 3} = 1 \times (1 + 7.0 \times 1.0) = 8.0$$

$$IGM(good) \text{ ในเอกสารที่ 4} = 1 \times (1 + 7.0 \times 1.0) = 8.0$$

$$IGM(electric) \text{ ในเอกสารที่ 5} = 1 \times (1 + 7.0 \times 0.6) = 5.2$$

$$IGM(good) \text{ ในเอกสารที่ 5} = 1 \times (1 + 7.0 \times 1.0) = 8.0$$

$$IGM(device) \text{ ในเอกสารที่ 5} = 1 \times (1 + 7.0 \times 0.5) = 3.5$$

$$IGM(best) \text{ ในเอกสารที่ 6} = 1 \times (1 + 7.0 \times 1.0) = 8.0$$

$$IGM(device) \text{ ในเอกสารที่ 6} = 1 \times (1 + 7.0 \times 0.5) = 3.5$$

ตารางที่ 3.6 BOW แสดงค่าและน้ำหนักค่าในแต่ละเอกสารด้วยการให้น้ำหนักแบบ TF-IGM

W_i	worst	electric	bad	good	best	items	device
D_1	8.0	5.2	0	0	0	8.0	0
D_2	8.0	5.2	0	0	0	0	3.5
D_3	0	5.2	8.0	0	0	0	0
D_4	0	0	0	8.0	0	0	0
D_5	0	5.2	0	8.0	0	0	3.5
D_6	0	0	0	0	8.0	0	3.5

3.3.2 การสร้างโมเดลการจำแนกความรู้สึกของบทวิจารณ์

โครงการปริญญาโทฉบับนี้ ได้นำเสนออัลกอริทึมสำหรับการจำแนกความรู้สึกของบทวิจารณ์ทั้งหมด 2 อัลกอริทึม นั่นคือ อัลกอริทึมนาอ์เบย์ (Naïve Bayes) และอัลกอริทึมเพื่อนบ้านใกล้ที่สุด (K-nearest Neighbor) เนื่องจากอัลกอริทึมที่เลือกใช้เป็นอัลกอริทึมที่มีประสิทธิภาพดีในการจัดกลุ่มเอกสารข้อความ โดยในการสร้างโมเดลด้วย Naïve Bayes และ K-nearest Neighbor จะมีการนำเอา BOW ที่ได้จากการคำนวณน้ำหนักค่าของแต่ละรูปแบบ มาทำการสร้างโมเดล ดังตัวอย่างต่อไปนี้

1) การจำแนกระดับบทวิจารณ์ด้วยอัลกอริทึม Naïve Bayes

การจำแนกบทวิจารณ์ด้วยอัลกอริทึม *Naïve Bayes* เป็นการนำเทคนิคการวิเคราะห์ความรู้สึกจากข้อความ และการจำแนกหมวดหมู่เอกสารมาประยุกต์ใช้ในการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม

$$P(v_j | a_1, a_2, \dots, a_n) = \prod_{i=1}^n P(a_i | v_j) \quad (16)$$

จากเอกสารทั้งหมด 6 เอกสาร จะมีเอกสารที่เป็นเอกสารที่มีความรู้สึกเป็นบวก (Positive) จำนวน 3 เอกสาร และเอกสารที่มีความรู้สึกเป็นลบ (Negative) จำนวน 3 เอกสาร ดังนั้นจะหาความน่าจะเป็นของคำสำคัญที่อยู่ในแต่ละเอกสารที่แยกคลาสออกจากกันจะได้ความน่าจะเป็น ดังสมการที่ 17

$$P(a_i | v_j) = \frac{\text{count}(a_i, v_j)}{\text{count}(v_j)} \quad (17)$$

โดยที่ $\text{count}(a_i, v_j)$ คือค่าความถี่ของคำที่ i ที่อยู่ในกลุ่มที่ j
 $\text{count}(v_j)$ คือค่าความถี่รวมในกลุ่มที่ j

แต่ในบางครั้งที่หาความน่าจะเป็นโดยใช้ *Naïve Bayes* นั้นอาจจะมีกรณีที่ค่าความถี่ของคำที่เกิดขึ้นเป็น 0 หรือก็คือคำที่อยู่ในถ้อยคำ ไม่ปรากฏอยู่ในเอกสารทำให้ค่าความน่าจะเป็นของคำนั้นเป็น 0 ตามไปด้วย ซึ่งไม่เป็นที่ยอมรับในทางสถิติที่โอกาสในการพยากรณ์จะมีค่าเป็นศูนย์ และเพื่อหลีกเลี่ยงการเกิดกรณีนี้จึงมีการปรับสมการด้วย *Laplace Smoothing* ที่มีการเพิ่มค่าความถี่ของข้อมูลเข้าไปอีกครั้งละ 1 และบวกเพิ่มค่าความถี่รวมด้วยค่าคงที่ k (อาจใช้ค่าขนาดของ *BOW*) จากคำทั้งหมด n คำ และกลุ่มทั้งหมด m กลุ่ม ดังนั้นจึงได้สมการ *Naïve Bayes* ที่ปรับแล้วดังนี้

$$P(a_i | v_j) = \frac{1 + \text{count}(a_i, v_j)}{k + \text{count}(v_j)} \quad (18)$$

โดยที่ $\text{count}(a_i, v_j)$ คือ ค่าความถี่ของคำที่ i ที่อยู่ในกลุ่มที่ j
 $\text{count}(v_j)$ คือ ค่าความถี่รวมในกลุ่มที่ j
 k คือ ค่าคงที่ที่มีการนำมาบวกเข้า

i มีค่าเท่ากับ $1, 2, 3..., n$

j มีค่าเท่ากับ $1, 2, 3..., m$

ในโครงการปัญญาประดิษฐ์นี้จะใช้สมการ *Naïve Bayes* ที่มีการปรับสมการ มาใช้ในการประมาณค่าความน่าจะเป็น โดยจะใช้ในการประมาณค่าความน่าจะเป็นของกลุ่ม และประมาณค่าความน่าจะเป็นของคำที่อยู่ในกลุ่ม โดยใช้ค่าน้ำหนักของคำดังที่ได้แสดงไว้ในขั้นตอนการนำเสนอเอกสาร ดังตัวอย่างต่อไปนี้

$$P(Class_i) = \frac{1 + \text{count}(\text{doc}, Class_j)}{NumClass + \text{count}(Class_i)} \quad (19)$$

โดยที่ $\text{count}(\text{doc}, Class_j)$ คือ จำนวนของเอกสารที่อยู่ในคลาส j
 $\text{count}(Class_j)$ คือ จำนวนของเอกสารทั้งหมด
 $NumClass$ คือ จำนวน $class$ ที่ใช้ในการสร้างโมเดล

$$P(w_i | Class_j) = \frac{1 + \text{count}(w_i | Class_j)}{TotalWord + \text{count}(Class_j)} \quad (20)$$

โดยที่ $\text{count}(w_i | Class_j)$ คือ ความถี่ของคำ i ที่อยู่ในกลุ่มที่ j
 $\text{count}(Class_j)$ คือ ความถี่รวมของคำทุกคำที่อยู่ในกลุ่มที่ j
 $TotalWord$ คือ จำนวนคำทั้งหมด

โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย *Naïve Bayes* โดยใช้การให้น้ำหนักคำแบบ *tf-idf*

Class = “Positive”

$$\begin{aligned} P(\text{Positive}) &= (1+3)/(2+6) &= 0.5 \\ P(\text{worst} | \text{Positive}) &= (1+0.0)/(7+2.52) &= 0.1050 \\ P(\text{electric} | \text{Positive}) &= (1+0.176)/(7+2.52) &= 0.1235 \\ P(\text{bad} | \text{Positive}) &= (1+0.0)/(7+2.52) &= 0.1050 \\ P(\text{good} | \text{Positive}) &= (1+0.954)/(7+2.52) &= 0.2052 \\ P(\text{best} | \text{Positive}) &= (1+0.788)/(7+2.52) &= 0.1878 \end{aligned}$$

$$P(\text{items} \mid \text{Positive}) = (1+0.0)/(7+2.52) = 0.1050$$

$$P(\text{device} \mid \text{Positive}) = (1+0.602)/(7+2.52) = 0.1682$$

Class = “Negative”

$$P(\text{Negative}) = (1+3)/(2+6) = 0.5$$

$$P(\text{worst} \mid \text{Negative}) = (1+0.954)/(7+2.395) = 0.2079$$

$$P(\text{electric} \mid \text{Negative}) = (1+0.352)/(7+2.395) = 0.1439$$

$$P(\text{bad} \mid \text{Negative}) = (1+0.788)/(7+2.395) = 0.1903$$

$$P(\text{best} \mid \text{Negative}) = (1+0.0)/(7+2.395) = 0.1064$$

$$P(\text{good} \mid \text{Negative}) = (1+0.0)/(7+2.395) = 0.1064$$

$$P(\text{items} \mid \text{Negative}) = (1+0.788)/(7+2.395) = 0.1903$$

$$P(\text{device} \mid \text{Negative}) = (1+0.301)/(7+2.395) = 0.1384$$

ตารางที่ 3.7 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักค่าแบบ *tf-idf*

W_i	worst	electric	bad	good	best	items	device	
D_1	0.2079	0.1439	0.1903	0.1064	0.1064	0.1903	0.1384	Negative
D_2	0.2079	0.1439	0.1903	0.1064	0.1064	0.1903	0.1384	
D_3	0.2079	0.1439	0.1903	0.1064	0.1064	0.1903	0.1384	
D_4	0.1050	0.1235	0.1050	0.2052	0.1878	0.1050	0.1682	Positive
D_5	0.1050	0.1235	0.1050	0.2052	0.1878	0.1050	0.1682	
D_6	0.1050	0.1235	0.1050	0.2052	0.1878	0.1050	0.1682	

โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักค่าแบบ *Delta TF-IDF*

Class = “Positive”

$$P(\text{worst} \mid \text{Positive}) = (1+0.0)/(7+-10.553) = -0.2814$$

$$P(\text{electric} \mid \text{Positive}) = (1+1.440)/(7+-10.553) = -0.6867$$

$$P(\text{bad} \mid \text{Positive}) = (1+0.0)/(7+-10.553) = -0.2814$$

$$P(\text{good} \mid \text{Positive}) = (1+-7.4)/(7+-10.553) = 1.8012$$

$$P(best | Positive) = (1+2.807)/(7+10.553) = 0.5085$$

$$P(items | Positive) = (1+0.0)/(7+10.553) = -0.2814$$

$$P(device | Positive) = (1+1.786)/(7+10.553) = 0.2212$$

Class = “Negative”

$$P(worst | Negative) = (1+7.4)/(7+16.441) = 0.6778$$

$$P(electric | Negative) = (1+4.32)/(7+16.441) = 0.3516$$

$$P(bad | Negative) = (1+2.807)/(7+16.441) = 0.1913$$

$$P(best | Negative) = (1+0.0)/(7+16.441) = -0.1059$$

$$P(good | Negative) = (1+0.0)/(7+16.441) = -0.1059$$

$$P(items | Negative) = (1+2.807)/(7+16.441) = 0.1913$$

$$P(device | Negative) = (1+0.893)/(7+16.441) = -0.2005$$

ตารางที่ 3.8 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักค่าแบบ *Delta TF-IDF*

W_i	worst	electric	bad	good	best	items	device	
D_1	0.6778	0.3516	0.1913	-0.1059	-0.1059	0.1913	-0.2005	Negative
D_2	0.6778	0.3516	0.1913	-0.1059	-0.1059	0.1913	-0.2005	
D_3	0.6778	0.3516	0.1913	-0.1059	-0.1059	0.1913	-0.2005	
D_4	-0.2814	-0.6867	-0.2814	1.8012	0.5085	-0.2814	0.2212	Positive
D_5	-0.2814	-0.6867	-0.2814	1.8012	0.5085	-0.2814	0.2212	
D_6	-0.2814	-0.6867	-0.2814	1.8012	0.5085	-0.2814	0.2212	

โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักค่าแบบ *TF-IDF-ICF*

Class = “Positive”

$$P(worst | Positive) = (1+0.0) / (7+ 3.029) = 0.0997$$

$$P(electric | Positive) = (1+0.176) / (7+ 3.029) = 0.1172$$

$$P(bad | Positive) = (1+0.0) / (7+ 3.029) = 0.0997$$

$$P(good | Positive) = (1+1.240) / (7+ 3.029) = 0.2233$$

$$P(best | Positive) = (1+1.012) / (7+ 3.029) = 0.2006$$

$$P(items | Positive) = (1+0.0) / (7+ 3.029) = 0.0997$$

$$P(device | Positive) = (1+0.602) / (7+ 3.029) = 0.1597$$

Class = “Negative”

$$P(worst | Negative) = (1+1.240) / (7+4.093) = 0.2019$$

$$P(electric | Negative) = (1+0.528) / (7+4.093) = 0.1377$$

$$P(bad | Negative) = (1+1.012) / (7+4.093) = 0.1813$$

$$P(best | Negative) = (1+0.0) / (7+4.093) = 0.0901$$

$$P(good | Negative) = (1+0.0) / (7+4.093) = 0.0901$$

$$P(items | Negative) = (1+1.012) / (7+4.093) = 0.1813$$

$$P(device | Negative) = (1+0.301) / (7+4.093) = 0.1172$$

ตารางที่ 3.9 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักคำแบบ TF-IDF-ICF

W_i	worst	electric	bad	good	best	items	device	
D_1	0.2019	0.1377	0.1813	0.0901	0.0901	0.1813	0.1172	Negative
D_2	0.2019	0.1377	0.1813	0.0901	0.0901	0.1813	0.1172	
D_3	0.2019	0.1377	0.1813	0.0901	0.0901	0.1813	0.1172	
D_4	0.0997	0.1172	0.0997	0.2233	0.2006	0.0997	0.1597	Positive
D_5	0.0997	0.1172	0.0997	0.2233	0.2006	0.0997	0.1597	
D_6	0.0997	0.1172	0.0997	0.2233	0.2006	0.0997	0.1597	

โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักคำแบบ TF-RF

Class = “Positive”

$$P(worst | Positive) = (1+0.0) / (7+11.32) = 0.0545$$

$$P(electric | Positive) = (1+1.736) / (7+11.32) = 0.1493$$

$$P(bad | Positive) = (1+0.0) / (7+11.32) = 0.0545$$

$$P(good | Positive) = (1+4.0) / (7+11.32) = 0.2729$$

$$P(best | Positive) = (1+1.584) / (7+11.32) = 0.1410$$

$$P(items | Positive) = (1+0.0) / (7+11.32) = 0.0545$$

$$P(device | Positive) = (1+4.0) / (7+11.32) = 0.2729$$

Class = “Negative”

$$P(worst | Negative) = (1+4.0) / (7+15.715) = 0.2201$$

$$P(electric | Negative) = (1+6.963) / (7+15.715) = 0.3505$$

$$P(bad | Negative) = (1+1.584) / (7+15.715) = 0.1137$$

$$P(best | Negative) = (1+0.0) / (7+15.715) = 0.0440$$

$$P(good | Negative) = (1+0.0) / (7+15.715) = 0.0440$$

$$P(items | Negative) = (1+1.584) / (7+15.715) = 0.1137$$

$$P(device | Negative) = (1+1.584) / (7+15.715) = 0.1137$$

ตารางที่ 3.10 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักค่าแบบ TF-RF

W_i	worst	electric	bad	good	best	items	device	
D_1	0.2201	0.3505	0.1137	0.0440	0.0440	0.1137	0.1137	Negative
D_2	0.2201	0.3505	0.1137	0.0440	0.0440	0.1137	0.1137	
D_3	0.2201	0.3505	0.1137	0.0440	0.0440	0.1137	0.1137	
D_4	0.0545	0.1493	0.0545	0.2729	0.1410	0.0545	0.2729	Positive
D_5	0.0545	0.1493	0.0545	0.2729	0.1410	0.0545	0.2729	
D_6	0.0545	0.1493	0.0545	0.2729	0.1410	0.0545	0.2729	

โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักค่าแบบ TF-IGM

Class = “Positive”

$$P(worst | Positive) = (1+0.0)/(7+36.2) = 0.0231$$

$$P(electric | Positive) = (1+5.2)/(7+36.2) = 0.1435$$

$$P(bad | Positive) = (1+0.0)/(7+36.2) = 0.0231$$

$$P(good | Positive) = (1+16)/(7+36.2) = 0.3935$$

$$P(best \mid Positive) = (1+8.0)/(7+36.2) = 0.2083$$

$$P(items \mid Positive) = (1+0.0)/(7+36.2) = 0.0231$$

$$P(device \mid Positive) = (1+7.0)/(7+36.2) = 0.1851$$

Class = “Negative”

$$P(worst \mid Negative) = (1+16.0)/(7+51.1) = 0.2925$$

$$P(electric \mid Negative) = (1+15.6)/(7+51.1) = 0.2857$$

$$P(bad \mid Negative) = (1+8.0)/(7+51.1) = 0.1549$$

$$P(best \mid Negative) = (1+0.0)/(7+51.1) = 0.0172$$

$$P(good \mid Negative) = (1+0.0)/(7+51.1) = 0.0172$$

$$P(items \mid Negative) = (1+8.0)/(7+51.1) = 0.1549$$

$$P(device \mid Negative) = (1+3.5)/(7+51.1) = 0.0774$$

ตารางที่ 3.11 โมเดลการจำแนกความรู้สึกของบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย Naïve Bayes โดยใช้การให้น้ำหนักค่าแบบ TF-IDF

W_i	worst	electric	bad	good	best	items	device	
D_1	0.2925	0.2857	0.1549	0.0172	0.0172	0.1549	0.0774	Negative
D_2	0.2925	0.2857	0.1549	0.0172	0.0172	0.1549	0.0774	
D_3	0.2925	0.2857	0.1549	0.0172	0.0172	0.1549	0.0774	
D_4	0.0231	0.1435	0.0231	0.3935	0.2083	0.0231	0.1851	Positive
D_5	0.0231	0.1435	0.0231	0.3935	0.2083	0.0231	0.1851	
D_6	0.0231	0.1435	0.0231	0.3935	0.2083	0.0231	0.1851	

2) การจำแนกบทวิจารณ์ด้วย K-nearest Neighbor (KNN)

KNN เป็นอัลกอริทึมที่ใช้ในการจัดกลุ่มข้อมูลที่ไม่ซับซ้อนเข้าใจง่าย ซึ่งวิธีนี้จะสามารถสร้างโมเดลที่มีประสิทธิภาพได้แม้เงื่อนไขที่ใช้ในการตัดสินใจจะมีความซับซ้อนก็ตาม โดยจะใช้หลักการเปรียบเทียบข้อมูลที่สนใจ (x) กับข้อมูลที่ถูกจัดกลุ่มไว้ก่อนล่วงหน้าในคลังข้อมูล เพื่อตรวจสอบว่าข้อมูล x นั้นคล้ายคลึงกับกลุ่มใด และถ้าหากข้อมูล x คล้ายคลึงกับกลุ่มใดมากที่สุด ระบบก็จะจัดข้อมูลให้ข้อมูล x เข้าไปอยู่ในกลุ่มนั้น แต่ในการตัดสินใจว่า x จะคล้ายกับข้อมูลในกลุ่มใดในคลังข้อมูล

จะขึ้นอยู่กับข้อกำหนดค่า k (ค่า k คือการเอาข้อมูลจำนวน k ตัวที่อยู่ใกล้ x มากที่สุดมาพิจารณา) เช่น ในการจำแนกระดับคะแนนบทวิจารณ์มีข้อมูลอยู่ 5 กลุ่ม และกำหนด $k=5$ ภายหลังจากการประมวลผลพบว่า ข้อมูล 5 อันดับแรกที่อยู่ใกล้ x มากที่สุดนั้น มาจากกลุ่มที่ 2 จำนวน 3 ตัว และมาจากกลุ่มที่ 1 จำนวน 2 ตัว ระบบก็จะพิจารณาข้อมูล x ให้อยู่กลุ่มที่ 2

สมมติให้มีเอกสารบทวิจารณ์เกี่ยวกับสินค้าอิเล็กทรอนิกส์ทั้งหมด 5 เอกสาร คือ

D_1 : One of worst electrical items.

D_2 : That's the worst electric device ever used.

D_3 : Bad HDMI.

D_4 : So Bad.

D_5 : This's a Good electronic device!

ตารางที่ 3.12 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักค่าด้วย $tf-idf$

W_i	worst	electric	bad	good	best	items	device
D_1	0.477	0.176	0	0	0	0.788	0
D_2	0.477	0.176	0	0	0	0	0.301
D_3	0	0.176	0.778	0	0	0	0
D_4	0	0	0	0.477	0	0	0
D_5	0	0.176	0	0.477	0	0	0.301
D_6	0	0	0	0	0.788	0	0.301

ตารางที่ 3.13 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN การให้น้ำหนักค่าด้วย $\Delta TF-IDF$

W_i	worst	electric	bad	good	best	items	device
D_1	-2.321	-1.222	0	0	0	-1.584	0
D_2	-2.321	-1.222	0	0	0	0	0.736
D_3	0	-1.222	-1.584	0	0	0	0
D_4	0	0	0	-2.321	0	0	0
D_5	0	1.222	0	-2.321	0	0	-0.736
D_6	0	0	0	0	-1.584	0	-0.736

ตารางที่ 3.14 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักคำด้วย TF-IDF-ICF

W_i	worst	electric	bad	good	best	items	device
D_1	0.620	0.176	0	0	0	1.012	0
D_2	0.620	0.176	0	0	0	0	0.301
D_3	0	0.176	1.012	0	0	0	0
D_4	0	0	0	0.620	0	0	0
D_5	0	0.176	0	0.620	0	0	0.301
D_6	0	0	0	0	1.012	0	0.301

ตารางที่ 3.15 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักคำด้วย TF-RF

W_i	worst	electric	bad	good	best	items	device
D_1	2.000	2.321	0	0	0	1.584	0
D_2	2.000	2.321	0	0	0	0	1.584
D_3	0	2.321	1.584	0	0	0	0
D_4	0	0	0	2.000	0	0	0
D_5	0	1.736	0	2.000	0	0	2.000
D_6	0	0	0	0	1.584	0	2.000

ตารางที่ 3.16 โมเดลวิเคราะห์ระดับคะแนนบทวิจารณ์ด้วย KNN โดยการให้น้ำหนักคำด้วย TF-IGM

W_i	worst	electric	bad	good	best	items	device
D_1	8.0	5.2	0	0	0	8.0	0
D_2	8.0	5.2	0	0	0	0	3.5
D_3	0	5.2	8.0	0	0	0	0
D_4	0	0	0	8.0	0	0	0
D_5	0	5.2	0	8.0	0	0	3.5
D_6	0	0	0	0	8.0	0	3.5

จากตารางที่ 3.12 ถึง ตารางที่ 3.16 จะเห็นว่าเอกสารตัวอย่าง เมื่อผ่านกระบวนการ pre-processing ที่ได้นำเสนอไปนั้น ก็จะได้เอกสารซึ่งเป็นข้อมูลที่ถูกต้องแล้วให้

ตัวแทนของแต่ละกลุ่ม และจะถูกนำไปใช้เปรียบเทียบกับข้อมูลที่เข้ามาใหม่ต่อไป โดยขั้นตอนของ KNN มีดังนี้

ขั้นตอนที่ 1 : การกำหนดค่า k

ขั้นตอนการกำหนดค่า k เป็นการกำหนดค่าเพื่อใช้เป็นเป้าหมายในการเลือกค่าที่ใกล้เคียงกับข้อมูลที่สนใจ โดยค่า k ที่กำหนดต้องเป็นเลขคี่ เพื่อให้โปรแกรมสามารถใช้ตัดสินใจได้ว่า x ควรจะถูกจัดอยู่ในกลุ่มใด ในโครงงานปริญญาานิพนธ์นี้กำหนดให้ค่า $k=3$, $k=5$ และ $k=7$

ขั้นตอนที่ 2 : คำนวณหาระยะทางระหว่าง x กับข้อมูลทุกตัวในคลังข้อมูล

การคำนวณค่าระยะทางระหว่างข้อมูลที่สนใจ กับข้อมูลทุกตัวในคลังข้อมูลจะทำการคำนวณระยะทางด้วย Euclidian distance เนื่องจากง่ายต่อความเข้าใจ และลักษณะการคำนวณที่คล้ายกับทฤษฎีบทพีทาโกรัส ซึ่งคำนวณได้ตามสมการดังต่อไปนี้

$$\sqrt{\sum_{i=0}^r [x_i - y_i]^2} \quad (21)$$

โดยที่ E คือ ระยะทางระหว่างข้อมูลที่สนใจ x กับข้อมูลในคลัง y

x_i คือ คุณลักษณะที่ i ของข้อมูลที่สนใจ x

y_i คือ คุณลักษณะที่ i ของข้อมูลที่ถูกเลือกไว้ในคลังข้อมูล y

ซึ่งข้อมูลที่สนใจ x จะถูกเปรียบเทียบกับข้อมูลในคลังข้อมูล y ทั้งหมด

ขั้นตอนที่ 3 : จัดเรียงลำดับของระยะทาง

เมื่อวัดระยะทางระหว่างข้อมูลที่สนใจ x กับข้อมูลในคลังข้อมูลเสร็จเรียบร้อยแล้ว จะมีการนำระยะทางที่วัดได้มาเรียงลำดับจากระยะทางที่น้อยที่สุดไปหามากที่สุด

ขั้นตอนที่ 4 : พิจารณาข้อมูลที่ใกล้ที่สุด k ตัว

เมื่อทำการจัดเรียงลำดับของระยะทางแล้วจะเลือกค่าระยะทางที่น้อยที่สุดจำนวน k ตัวมาพิจารณาหาคำตอบ เช่น ถ้าหากค่า $k=5$ ก็จะเลือกข้อมูลจากลำดับที่ 1 ถึง 5 มาพิจารณา

ขั้นตอนที่ 5 : กำหนด Class ให้กับข้อมูล x

การกำหนด Class ให้กับข้อมูล x จะทำโดยการพิจารณาว่าข้อมูลจำนวน 5 ตัวที่อยู่ใกล้ x มากที่สุดอยู่กลุ่มใดบ้าง เช่น ถ้าข้อมูลในกลุ่มที่ 4 มีจำนวน 3 ตัว อยู่ในกลุ่ม 5 จำนวน 1 ตัว และกลุ่มที่ 2 จำนวน 1 ตัว ระบบจึงตัดสินใจให้ข้อมูล x อยู่ในกลุ่มที่ 4

อย่างไรก็ตาม มีข้อสังเกตว่าถ้าเลือกค่า k น้อยเกินไปอาจจะทำให้ไวต่อสัญญาณรบกวนได้ และถ้าหากเลือกค่า k มากเกินไปอาจจะทำให้มีกลุ่มข้อมูลอื่นๆ มาปะปนกับข้อมูลที่กำลังสนใจได้เช่นกัน ดังนั้นวิธีการนี้จึงมีทั้งข้อดีและข้อเสียซึ่ง ข้อดีคือเป็นวิธีการที่ง่ายและให้ประสิทธิภาพความถูกต้องสูง แต่ข้อเสียคือเวลาที่ใช้ในการประมวลผลค่อนข้างนาน เพราะการทำนายข้อมูลที่เข้ามาใหม่จะอาศัยการเปรียบเทียบข้อมูลใหม่กับข้อมูลเรียนรู้จำนวน k ตัวที่อยู่ใกล้ที่สุด

3.4 การวัดประสิทธิภาพของตัวจัดกลุ่มเอกสาร (Evaluation)

เป็นส่วนของการประเมินประสิทธิภาพแบบจำลองเพื่อการจำแนกรู้สึกที่สร้างขึ้นภายใต้การให้น้ำหนักที่แตกต่างกัน โดยจะประเมินผลลัพธ์ของการจำแนกรู้สึกด้วยเทคนิคการวัดค่าความระลึก (Recall), การวัดความแม่นยำ (Precision) และ การวัดค่าเอฟ (F-measure หรือ F1) โดยจะมีขั้นตอนดังนี้

3.4.1 การนำโมเดลเพื่อการจำแนกกลุ่มของบทวิจารณ์ไปใช้

เป็นขั้นตอนของการนำเอาโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์มาใช้ในการวิเคราะห์ว่าบทวิจารณ์ที่ผู้ซื้อได้เขียนเกี่ยวกับสินค้าอิเล็กทรอนิกส์นั้นๆ ควรจัดอยู่ในกลุ่มใด โดยจะมีการรับ “ข้อความ” เข้ามา แล้วโมเดลจะวิเคราะห์ว่าข้อความที่เข้ามาถูกจัดอยู่ในกลุ่มใด

เมื่อได้โมเดลเพื่อการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แล้ว สามารถนำมาใช้จัดกลุ่มข้อความแสดงความคิดเห็นที่ผู้ซื้อได้ไปแสดงความคิดเห็นไว้ในเว็บไซต์ Amazon ที่มีการแสดงความคิดเห็นเกี่ยวกับสินค้าอิเล็กทรอนิกส์ เพื่อจำแนกระดับตามที่ต้องการ สำหรับขั้นตอนในการจำแนกระดับคะแนนข้อความบทวิจารณ์ มีดังนี้

ตัวอย่างข้อความแสดงความคิดเห็น

D_{new} : impressive and good device.

ขั้นตอนแรกจะเป็นการตัดคำและการตัดคำหยุด เพื่อกำจัดคำที่ไม่มีนัยสำคัญกับเอกสารออก

ตารางที่ 3.17 แสดงคำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ NV

Document	ข้อความที่ผ่านกระบวนการ pre-processing
New	impressive / good / device

เมื่อได้คำสำคัญจากข้อความแสดงความคิดเห็นแล้ว เราจะใช้โมเดลที่สร้างขึ้นด้วยอัลกอริทึมข้างต้นในการวิเคราะห์ข้อความแสดงความคิดเห็น โดยจะมี 2 โมเดล ดังนี้

(1) การนำโมเดลไปใช้ในจำแนกบทวิจารณ์ด้วย Naïve Bayes

ในการจัดกลุ่มเอกสารข้อความที่เข้ามาใหม่ จะประเมินจากผลรวมความน่าจะเป็นของแต่ละคำในเอกสาร โดยใช้ความน่าจะเป็นของแต่ละคำที่ถูกคำนวณไว้ก่อนหน้านี้ ในที่นี้จะประเมินจากทุกคลาส ถ้าหากค่าประเมินในคลาสใดสูงสุด จะสรุปได้ว่าเอกสารที่นำมาประเมินอยู่ในกลุ่มนั้น

$$v_{NB} = \operatorname{argmax} P(v_j) \times \prod_{i=1}^n P(a_i | v_j) \quad : v_j \in V \quad (22)$$

จากสมการที่ 3.7 ซึ่งกำหนดให้ V_{NB} คือเอกสารที่ผ่านการจัดกลุ่ม ซึ่งสามารถคำนวณความน่าจะเป็นเพื่อประเมินคลาส ที่ละคลาสตามลำดับ โดย $P(Class) = 0.5$ ในทุกคลาส จะได้ว่า

โมเดลการจำแนกบทวิจารณ์ที่มีการให้น้ำหนักค่าแบบ *tf-idf*

พิจารณาใน Class = “Positive”

$$\begin{aligned} V_{NEW} &= P(Positive) \times P(good|Positive) \times P(device|Positive) \\ &= (0.5) \times (0.2052) \times (0.1682) \\ &= 0.01725732 \end{aligned}$$

พิจารณาใน Class = “Negative”

$$\begin{aligned} V_{NEW} &= P(Negative) \times P(good|Negative) \times P(device|Negative) \\ &= (0.5) \times (0.1064) \times (0.1384) \\ &= 0.00736288 \end{aligned}$$

จากผลลัพธ์ข้างต้นจะเห็นได้ว่า D_{NEW} นั้นมีค่าความน่าจะเป็นอยู่ที่ 0.01725732 ใน Class= “Positive” มากกว่า Class = “Negative” ดังนั้นจึงสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่มของ Positive

โมเดลการจำแนกบทวิจารณ์ที่มีการให้น้ำหนักค่าแบบ *Delta TF-IDF*

พิจารณาใน *Class* = “*Positive*”

$$\begin{aligned} V_{\text{NEW}} &= P(\text{Positive}) \times P(\text{good}|\text{Positive}) \times P(\text{device}|\text{Positive}) \\ &= (0.5) \times (1.8012) \times (0.2212) \\ &= 0.19921272 \end{aligned}$$

พิจารณาใน *Class* = “*Negative*”

$$\begin{aligned} V_{\text{NEW}} &= P(\text{Negative}) \times P(\text{good}|\text{Negative}) \times P(\text{device}|\text{Negative}) \\ &= (0.5) \times (-0.1059) \times (-0.2005) \\ &= 0.010616475 \end{aligned}$$

จากผลลัพธ์ข้างต้นจะเห็นได้ว่า D_{NEW} นั้นมีค่าความน่าจะเป็นอยู่ที่ 0.19921272 ใน *Class*= “*Positive*” มากกว่า *Class* = “*Negative*” ดังนั้นจึงสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่มของ *Positive*

โมเดลการจำแนกบทวิจารณ์ที่มีการให้น้ำหนักค่าแบบ *TF-IDF-ICF*

พิจารณาใน *Class* = “*Positive*”

$$\begin{aligned} V_{\text{NEW}} &= P(\text{Positive}) \times P(\text{good}|\text{Positive}) \times P(\text{device}|\text{Positive}) \\ &= (0.5) \times (0.2233) \times (0.1597) \\ &= 0.017830505 \end{aligned}$$

พิจารณาใน *Class* = “*Negative*”

$$\begin{aligned} V_{\text{NEW}} &= P(\text{Negative}) \times P(\text{good}|\text{Negative}) \times P(\text{device}|\text{Negative}) \\ &= (0.5) \times (0.0901) \times (0.1172) \\ &= 0.00527986 \end{aligned}$$

จากผลลัพธ์ข้างต้นจะเห็นได้ว่า D_{NEW} นั้นมีค่าความน่าจะเป็นอยู่ที่ 0.017830505 ใน *Class*= “*Positive*” มากกว่า *Class* = “*Negative*” ดังนั้นจึงสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่มของ *Positive*

โมเดลการจำแนกบทวิจารณ์ที่มีการให้น้ำหนักค่าแบบ *TF-RF*

พิจารณาใน *Class* = “*Positive*”

$$\begin{aligned}
 V_{\text{NEW}} &= P(\text{Positive}) \times P(\text{good}|\text{Positive}) \times P(\text{device}|\text{Positive}) \\
 &= (0.5) \times (0.2729) \times (0.2729) \\
 &= 0.037237205
 \end{aligned}$$

พิจารณาใน Class = “Negative”

$$\begin{aligned}
 V_{\text{NEW}} &= P(\text{Negative}) \times P(\text{good}|\text{Negative}) \times P(\text{device}|\text{Negative}) \\
 &= (0.5) \times (0.0440) \times (0.1137) \\
 &= 0.0025014
 \end{aligned}$$

จากผลลัพธ์ข้างต้นจะเห็นได้ว่า D_{NEW} นั้นมีค่าความน่าจะเป็นอยู่ที่ 0.037237205 ใน Class= “Positive” มากกว่า Class = “Negative” ดังนั้นจึงสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่มของ Positive

โมเดลการจำแนกบทวิจารณ์ที่มีการให้น้ำหนักค่าแบบ TF-IGM

พิจารณาใน Class = “Positive”

$$\begin{aligned}
 V_{\text{NEW}} &= P(\text{Positive}) \times P(\text{good}|\text{Positive}) \times P(\text{device}|\text{Positive}) \\
 &= (0.5) \times (0.3935) \times (0.1851) \\
 &= 0.036418425
 \end{aligned}$$

พิจารณาใน Class = “Negative”

$$\begin{aligned}
 V_{\text{NEW}} &= P(\text{Negative}) \times P(\text{good}|\text{Negative}) \times P(\text{device}|\text{Negative}) \\
 &= (0.5) \times (0.0172) \times (0.0774) \\
 &= 0.00066564
 \end{aligned}$$

จากผลลัพธ์ข้างต้นจะเห็นได้ว่า D_{NEW} นั้นมีค่าความน่าจะเป็นอยู่ที่ 0.036418425 ใน Class= “Positive” มากกว่า Class = “Negative” ดังนั้นจึงสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่มของ Positive

(2) การนำโมเดลไปใช้ในจำแนกบทวิจารณ์ด้วย KNN

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (23)$$

จากสมการที่ 20 เป็นการหาค่าระยะทางระหว่างเอกสารที่เข้ามาใหม่ กับทุกเอกสารที่อยู่
ในโมเดลว่าเอกสารใดมีความใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุด โดยค่าระยะทางยิ่งน้อยแสดงว่า
เอกสารที่เข้ามาใหม่ใกล้เคียงกับเอกสารนั้นๆ มาก ซึ่งในโครงการงานปริญญาโทฉบับนี้จะพิจารณาเอกสารที่
ใกล้เคียงมากที่สุด 3 และ 5 เอกสาร โดยเรียงจากน้อยไปมาก

การใช้โมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย KNN โดยใช้การให้น้ำหนัก
ค่าแบบ *tf-idf* อ้างอิงค่าที่ใช้พิจารณาจากตารางที่ 3.2

ตัวอย่างเอกสารที่เข้ามาใหม่

D_{new} : impressive and good device.

ตารางที่ 3.18 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-IDF

Word	impressive	good	device
D_{New}	1	1	1

ให้น้ำหนักคำในเอกสารตัวอย่างด้วย *tf-idf*

$$W_{good} = 1 * 0.477 = 0.477$$

$$W_{device} = 1 * 0.301 = 0.301$$

พิจารณาใน Class = “Negative”

$$\begin{aligned} D_1 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0.477)^2 + (0 - 0.301)^2} \\ &= 0.31813 \end{aligned}$$

$$\begin{aligned} D_2 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0.477)^2 + (0.301 - 0.301)^2} \\ &= 0.227529 \end{aligned}$$

$$\begin{aligned} D_3 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0.477)^2 + (0 - 0.301)^2} \\ &= 0.31813 \end{aligned}$$

พิจารณาใน Class = “Positive”

$$\begin{aligned}
 D_4 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
 &= \sqrt{(0.477 - 0.477)^2 + (0 - 0.301)^2} \\
 &= 0.090601
 \end{aligned}$$

$$\begin{aligned}
 D_5 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
 &= \sqrt{(0.477 - 0.477)^2 + (0.301 - 0.301)^2} \\
 &= 0.0
 \end{aligned}$$

$$\begin{aligned}
 D_6 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
 &= \sqrt{(0 - 0.477)^2 + (0.301 - 0.301)^2} \\
 &= 0.227529
 \end{aligned}$$

พิจารณาโดยใช้ $K = 3$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_4 และ D_2 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม *Positive* จำนวน 2 เอกสาร และอยู่ในกลุ่ม *Negative* จำนวน 1 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม *Positive*

พิจารณาโดยใช้ $K = 5$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_4 , D_2 , D_1 และ D_1 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม *Positive* จำนวน 3 เอกสาร อยู่ในกลุ่ม *Negative* จำนวน 2 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม *Positive* การใช้โมเดลการจำแนกระดับคะแนนบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย KNN โดยใช้การให้น้ำหนักค่าแบบ *Delta TF-IDF* อ้างอิงค่าที่ใช้พิจารณาจากตารางที่ 3.3

ตัวอย่างเอกสารที่เข้ามาใหม่

D_{new} : impressive and good device.

ตารางที่ 3.19 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ Delta TF-IDF

Word	impressive	good	device
D_{New}	1	1	1

ให้น้ำหนักคำในเอกสารตัวอย่างด้วย *Delta TF-IDF*

$$W_{good} = 1 * \log_2 \left(\frac{1 * 0 + 0.5}{1 * 0 + 0.5} \right) = 0$$

$$W_{device} = 1 * \log_2 \left(\frac{1 * 0 + 0.5}{1 * 0 + 0.5} \right) = 0$$

พิจารณาใน Class = “Negative”

$$\begin{aligned} D_1 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0)^2 + (0 - 0)^2} \\ &= 0.0 \end{aligned}$$

$$\begin{aligned} D_2 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0)^2 + (0.893 - 0)^2} \\ &= 0.893 \end{aligned}$$

$$\begin{aligned} D_3 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0)^2 + (0 - 0)^2} \\ &= 0.0 \end{aligned}$$

พิจารณาใน Class = “Positive”

$$\begin{aligned} D_4 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(-3.700 - 0)^2 + (0 - 0)^2} \\ &= 3.7 \end{aligned}$$

$$\begin{aligned} D_5 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(-3.700 - 0)^2 + (-0.893 - 0)^2} \\ &= 4.593 \end{aligned}$$

$$\begin{aligned} D_6 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0)^2 + (-0.893 - 0)^2} \\ &= 0.893 \end{aligned}$$

พิจารณาโดยใช้ $K = 3$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_1 , D_2 และ D_6 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม Negative จำนวน 2 เอกสาร และอยู่ในกลุ่ม Positive จำนวน 1 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม Negative

พิจารณาโดยใช้ $K = 5$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_1 , D_2 , D_6 , D_3 และ D_4 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม Negative

จำนวน 3 เอกสาร อยู่ในกลุ่ม *Positive* จำนวน 2 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม *Negative*

การใช้โมเดลการจำแนกระดับคะแนนบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย KNN โดยใช้การให้น้ำหนักค่าแบบ *TF-IDF-ICF* อ้างอิงค่าที่ใช้พิจารณาจากตารางที่ 3.4

ตัวอย่างเอกสารที่เข้ามาใหม่

D_{new} : impressive and good device.

ตารางที่ 3.20 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-IDF-ICF

Word	impressive	good	device
D_{New}	1	1	1

ให้น้ำหนักค่าในเอกสารตัวอย่างด้วย *TF-IDF-ICF*

$$W_{good} = 1 * 0.477 * 1.301 = 0.620$$

$$W_{device} = 1 * 0.301 * 1 = 0.301$$

พิจารณาใน Class = “*Negative*”

$$\begin{aligned} D_1 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0.620)^2 + (0 - 0.301)^2} \\ &= 0.921 \end{aligned}$$

$$\begin{aligned} D_2 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0.620)^2 + (0.301 - 0.301)^2} \\ &= 0.620 \end{aligned}$$

$$\begin{aligned} D_3 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 0.620)^2 + (0 - 0.301)^2} \\ &= 0.921 \end{aligned}$$

พิจารณาใน Class = “*Positive*”

$$\begin{aligned} D_4 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0.620 - 0.620)^2 + (0 - 0.301)^2} \\ &= 0.301 \end{aligned}$$

$$\begin{aligned}
 D_5 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
 &= \sqrt{(0.620 - 0.620)^2 + (0.301 - 0.301)^2} \\
 &= 0 \\
 D_6 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
 &= \sqrt{(0 - 0.620)^2 + (0.301 - 0.301)^2} \\
 &= 0.620
 \end{aligned}$$

พิจารณาโดยใช้ $K = 3$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_4 และ D_2 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม *Positive* จำนวน 2 เอกสาร และอยู่ในกลุ่ม *Negative* จำนวน 1 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม *Positive*

พิจารณาโดยใช้ $K = 5$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_4 , D_2 , D_6 และ D_1 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม *Positive* จำนวน 3 เอกสาร อยู่ในกลุ่ม *Negative* จำนวน 2 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม *Positive*

การใช้โมเดลการจำแนกระดับคะแนนบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย KNN โดยใช้การให้น้ำหนักค่าแบบ TF-RF อ้างอิงค่าที่ใช้พิจารณาจากตารางที่ 3.5

ตัวอย่างเอกสารที่เข้ามาใหม่

D_{new} : impressive and good device.

ตารางที่ 3.21 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-RF

Word	impressive	good	device
D_{New}	1	1	1

ให้น้ำหนักคำในเอกสารตัวอย่างด้วย TF-RF

$$W_{good} = 1 * \log_2 \left(2 + \frac{1}{\max(1,0)} \right) = 1.584$$

$$W_{device} = 1 * \log_2 \left(2 + \frac{1}{\max(1,0)} \right) = 1.584$$

พิจารณาใน Class = “Negative”

$$D_1 = \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2}$$

$$\begin{aligned}
&= \sqrt{(0 - 1.584)^2 + (0 - 1.584)^2} \\
&= 2.2401 \\
D_2 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
&= \sqrt{(0 - 1.584)^2 + (1.584 - 1.584)^2} \\
&= 1.584 \\
D_3 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
&= \sqrt{(0 - 1.584)^2 + (0 - 1.584)^2} \\
&= 2.2401
\end{aligned}$$

พิจารณาใน Class = “Positive”

$$\begin{aligned}
D_4 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
&= \sqrt{(0 - 1.584)^2 + (2.000 - 1.584)^2} \\
&= 1.6377 \\
D_5 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
&= \sqrt{(2.000 - 1.584)^2 + (2.000 - 1.584)^2} \\
&= 0 \\
D_6 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
&= \sqrt{(2.000 - 1.584)^2 + (0.0 - 1.584)^2} \\
&= 1.6377
\end{aligned}$$

พิจารณาโดยใช้ $K = 3$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_4 และ D_2 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม Positive จำนวน 2 เอกสาร และอยู่ในกลุ่ม Negative จำนวน 1 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม Positive

พิจารณาโดยใช้ $K = 5$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_2 , D_4 , D_6 และ D_1 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม Positive จำนวน 3 เอกสาร อยู่ในกลุ่ม Negative จำนวน 2 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม Positive

การใช้โมเดลการจำแนกระดับคะแนนบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วย KNN โดยใช้การให้น้ำหนักคำแบบ TF-IGM อ้างอิงค่าที่ใช้พิจารณาจากตารางที่ 3.6

ตัวอย่างเอกสารที่เข้ามาใหม่

D_{new} : impressive and good device.

ตารางที่ 3.22 คำสำคัญที่ได้หลังจากผ่านกระบวนการ pre-processing ในการทดสอบ TF-IGM

Word	impressive	good	device
D_{New}	1	1	1

ให้น้ำหนักคำในเอกสารตัวอย่างด้วย TF-IGM

$$W_{good} = 1 * (1 + 7.0 * 1) = 7.0$$

$$W_{device} = 1 * (1 + 7.0 * 0.5) = 3.5$$

พิจารณาใน Class = “Negative”

$$\begin{aligned} D_1 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 7.0)^2 + (0 - 3.5)^2} \\ &= 7.82623792125 \end{aligned}$$

$$\begin{aligned} D_2 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 7.0)^2 + (3.5 - 3.5)^2} \\ &= 7.0 \end{aligned}$$

$$\begin{aligned} D_3 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(0 - 7.0)^2 + (0 - 3.5)^2} \\ &= 7.82623792125 \end{aligned}$$

พิจารณาใน Class = “Positive”

$$\begin{aligned} D_4 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(7.0 - 7.0)^2 + (0 - 3.5)^2} \\ &= 3.5 \end{aligned}$$

$$\begin{aligned} D_5 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\ &= \sqrt{(7.0 - 7.0)^2 + (3.5 - 3.5)^2} \end{aligned}$$

$$\begin{aligned}
 &= 0 \\
 D_6 &= \sqrt{(Old_{good} - New_{good})^2 + (Old_{device} - New_{device})^2} \\
 &= \sqrt{(0 - 7.0)^2 + (3.5 - 3.5)^2} \\
 &= 7.0
 \end{aligned}$$

พิจารณาโดยใช้ $K = 3$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_4 และ D_2 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม *Positive* จำนวน 2 เอกสาร และอยู่ในกลุ่ม *Negative* จำนวน 1 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม *Positive*

พิจารณาโดยใช้ $K = 5$ จะเห็นว่า เอกสารที่มีความใกล้เคียงกับ D_{NEW} มากที่สุด คือ D_5 , D_4 , D_2 , D_6 และ D_1 ตามลำดับ ซึ่งเอกสารที่ใกล้เคียงกับเอกสารที่เข้ามาใหม่มากที่สุดอยู่ในกลุ่ม *Positive* จำนวน 3 เอกสาร อยู่ในกลุ่ม *Negative* จำนวน 2 เอกสาร ดังนั้นจึงสามารถสรุปได้ว่า D_{NEW} จัดอยู่ในกลุ่ม *Positive*

3.4.2 การวัดประสิทธิภาพของตัวจัดกลุ่มเอกสาร (Evaluation)

การวัดประสิทธิภาพของตัวจัดกลุ่มเอกสารเป็นขั้นตอนการประเมินโมเดลเพื่อใช้ในการจัดกลุ่มเอกสารก่อนการนำไปใช้งานจริงที่โดยทั่วไป จะใช้เทคนิคมาตรฐานที่นิยมใช้กันอย่างแพร่หลาย ที่เรียกว่า การวัดค่าความระลึก (Recall) การวัดค่าความแม่นยำ (Precision) และการวัดค่า F-Measure ตัวอย่าง

ตารางที่ 3.23 ตัวอย่าง Confusion Matrix

N=50		Prediction	
		Class 1	Class 2
Actual	Class 1	42	5
	Class 2	8	45

1. การวัดค่าความระลึก (Recall)

สมมติให้ แต่ละ class มีเอกสารจำนวน 50 เอกสาร ซึ่งรวมทั้งสิ้น 100 เอกสาร และในการจำแนกเอกสารอัตโนมัติทำนายได้ถูกต้องตามความจริง (TP) และทำนายผิด (FN) จะได้ค่าความระลึกดังต่อไปนี้

$$R(class\ 1) = 42/(42+8) = 0.84$$

$$R(class\ 2) = 45/(45+5) = 0.90$$

$$\text{ดังนั้น Average Recall} = (0.84+0.90)/2 = 0.87$$

2. การวัดค่าความแม่นยำ (Precision)

สมมติให้ แต่ละ class มีเอกสารจำนวน 50 เอกสาร ซึ่งรวมทั้งสิ้น 100 เอกสาร และในการจำแนกเอกสารอัตโนมัติทำนายได้ถูกต้องตามความจริง (TP) และทำนายไม่ถูกต้องตามความจริง (FP) จะได้ค่าความแม่นยำดังต่อไปนี้

$$P(class\ 1) = 42/(42+5) = 0.8936$$

$$P(class\ 2) = 45/(45+8) = 0.8490$$

$$\text{ดังนั้น Average Precision} = (0.8936+0.8490)/2 = 0.8713$$

3. การวัดค่า F-Measure

คือผลเฉลี่ยระหว่างค่าความแม่นยำและค่าความระลึกลสามารถแสดงตัวอย่างคำนวณได้ดังต่อไปนี้

$$\begin{aligned} \text{F-measure} &= 2 * (0.87*0.8713)/(0.87+0.8713) \\ &= 0.8706 \end{aligned}$$

3.5 การปรับปรุงประสิทธิภาพโมเดลเพื่อการจำแนก

3.5.1 ปัญหาจากการทำ Lemmatization

เนื่องจากการทำ Lemma เป็นการเปลี่ยนคำให้อยู่ในรูปแบบดั้งเดิม ตัวอย่างเช่น คำว่า This's จะถูกเปลี่ยนเป็น this, be และด้วยเหตุนี้เอง ทำให้คำบางคำที่มีผลต่อการแสดงความรู้สึก อาจถูกเปลี่ยนแปลงไป เช่นคำว่า don't เมื่อผ่านกระบวนการเปลี่ยนรูปคำให้อยู่ในรูปแบบดั้งเดิมแล้ว จะได้คำว่า do และคำว่า not ซึ่งจะเห็นว่า หากสองคำนี้ถูกแยกออกจากกันทำให้ความหมาย หรือค่าน้ำหนักของคำเปลี่ยนแปลงไป เช่น

I don't like this device. จะได้คำว่า I / do / ' / not / like / this / device / .

และจากตัวอย่างข้างต้นจะเห็นได้ว่า หลังจากผ่านการทำ Lemma จะได้อักขระพิเศษเข้ามาในการประมวลผลด้วยดังตัวอย่าง ทำให้มีคำมากยิ่งขึ้นซึ่งคำที่ได้ไม่ได้มีผลกับการแสดงความรู้สึก แต่ถูกนำมาคำนวณ ทำให้ใช้ระยะเวลาในการประมวลผลมากขึ้น

3.5.2 ปัญหาด้านการใช้ภาษา

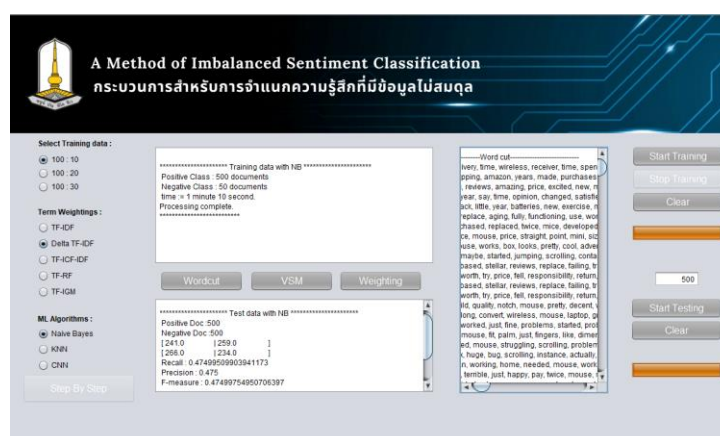
เนื่องจากข้อมูลที่ใช้ในการสร้างโมเดลเป็นเอกสารข้อความแสดงความคิดเห็นเกี่ยวกับสินค้าอิเล็กทรอนิกส์ ที่เปิดให้ทุกคนสามารถเข้ามาเขียนแสดงความคิดเห็นและให้คะแนนสินค้านั้นๆ ได้ ทำให้เกิดปัญหาด้านการใช้ภาษา คือการใช้คำที่ไม่มีความหมาย หรือไม่มีในพจนานุกรม (Unknown word) ดังนั้น จึงได้มีการนำพจนานุกรมมาใช้เพื่อคัดกรองคำเหล่านั้นออกไป เพราะคำเหล่านั้นไม่ได้มีความหมาย หรือส่งผลต่อการจัดกลุ่มเอกสาร

```
File Edit Format View Help
baddddsandra=1
soooooooooooooooooo =1
wompwomp=1
trejuo=1
hummm=1
zzzzzzzzzzzzzzzz=1
jimmy=1
ahhh=1
pwiiwhy=1
emma=4
s2=1
s3=1
jennysue=1
arghhhhhhhhhhhhhyes=1
wOwvvvvvvvvvvvvvv=1
```

ภาพประกอบที่ 3.6 ตัวอย่าง Unknown word

3.6 ตัวอย่างหน้าจอโปรแกรม

ตัวอย่างหน้าจอการทำงานของโปรแกรมที่เราจะนำเสนอระบบการควบคุมข้อมูลไม่สมดุลในการจำแนกความรู้สึก



ภาพประกอบที่ 3.7 ตัวอย่างหน้าจอโปรแกรม

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลอง ในการนำตัวจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ ที่ได้จากขั้นตอนการดำเนินงาน มาทำการทดลองเพื่อจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ที่ต้องการตรวจสอบ

4.1 ข้อมูลที่ใช้ในการทดสอบ

ข้อมูลที่ใช้การในทดสอบการสร้างโมเดลสำหรับการจำแนกบทวิจารณ์อิเล็กทรอนิกส์นั้น จะเป็นชุดข้อมูลทดสอบ (Test set) ที่ได้ทำการคัดเลือกไว้แล้วในขั้นตอนข้างต้น ที่เก็บอยู่ในรูปแบบของ XML ดังภาพประกอบที่ 4.1

```
<?xml version="1.0" encoding="UTF-8"?>
- <Reviews>
  - <Review status="Positive" id="1">
    <details>I had never used a screen protector before, but with my new phone, both the clerk in the phone store and my daughter suggested it would be a good idea. I didn't want a cheap peel off film so I went on Amazon and picked the Mkeke XR Screen protector. It comes with 3 in the package, alcohol wipes and a frame for installation. Installation was a breeze, and now I have a nice new sturdy plastic screen protector, and two as backup should this one get scratched. Very pleased with the product, the price was right too.</details>
  </Review>
  - <Review status="Positive" id="2">
    <details>This is the second time I've bought these. They are easier to install than most screen protectors I've tried. My daughter has an iPhone XR and we really worried about her breaking the screen when we got it. These seem to be doing the job. Two of our phones have these on the screens and they work well. The touch function works fine & they adhere well.</details>
  </Review>
  - <Review status="Positive" id="3">
    <details>Comes with individual packets that have everything you need. Also comes with a jolder to click onto the front of your phone so you can easily put the screen on. Im a bit confused though because there is also some instructions on the inside saying there is a back part too but i dont see any materials for that</details>
  </Review>
  - <Review status="Positive" id="4">
    <details>This is an awesome Screen Protector and Easy to Install. The package came with three screen protectors, three sets of what you can see on the picture (except the black border thing that helps you to guide the install). It came with all the things I need. And just like any tempered glass protectors I almost feel like I am touching the actual screen, because of the similar glass feelings, just like the iPhone screen itself. Coming with three of them also provides me two spare screen, that should be more than enough for two years.</details>
  </Review>
</Reviews>
```

ภาพประกอบที่ 4.1 ตัวอย่างบทวิจารณ์สินค้าอิเล็กทรอนิกส์ที่ใช้ในการทดสอบ

4.2 Algorithm Setup

ในหัวข้อนี้จะกล่าวถึงการตั้งค่าในอัลกอริทึมที่ใช้ในปริยุฏฐานิพนธ์นี้ ซึ่งประกอบไปด้วย 3 อัลกอริทึมดังนี้

4.2.1 KNN Setup

อัลกอริทึม KNN นั้นมีการกำหนดค่า k โดยค่า k ที่ใช้ในงานปริยุฏฐานิพนธ์นี้คือ 7 โดยการที่ได้ค่า k นั้นมาจากการทำการทดสอบค่า k ทั้งหมด 4 ค่า คือ 5, 7, 11 และ 15 กับทุกสัดส่วนที่ใช้ในการสร้างโมเดลแล้วนำมาเฉลี่ยหาค่าความระลึก ค่าความแม่นยำ และค่าเฉลี่ย F -measure โดยเราจะทำการนำค่า k ไปทดสอบกับทุกการให้น้ำหนักค่ากับทุกสัดส่วนในการสร้างโมเดล

ตารางที่ 4.1 ตารางการทดสอบประสิทธิภาพของค่า k

ค่า k	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
5	0.6547	0.6615	0.6580
7	0.6834	0.6713	0.6772
11	0.6458	0.6450	0.6454
15	0.6232	0.6220	0.6226

ดังนั้นจากตารางที่ 4.1 เห็นได้ว่าค่า k ที่มามีค่าเฉลี่ยมากที่สุด คือ $k=7$ เนื่องจากข้อมูลที่เราลองมาคือ $k=5$ เนื่องจาก ข้อมูลที่ใช้ในการสร้างโมเดลนั้น เป็นชุดข้อมูลที่ไม่มีความสมดุล ทำให้การที่ค่า k เยอะมีประสิทธิภาพที่ต่ำนั้นเป็นเรื่องที่เห็นได้เป็นปกติ ดังนั้น เราจึงได้ทำการเลือกใช้ค่า $k=7$ ในงานปริญาานิพนธ์นี้

4.2.2 Naïve Bayes

สำหรับอัลกอริทึม Naïve Bayes นั้นได้ทำการใช้ Multinomial Naïve Bayes (MNB) ในการสร้างและทดสอบโมเดล เนื่องจาก MNB นั้นถูกสร้างขึ้นมาเพื่อใช้ในการจำแนกเอกสาร โดยมีการคำนวณสัดส่วนเอกสาร ซึ่ง MNB คือ ตัวทำนายที่ใช้โดยลักษณะนามคือความถี่ของคำที่มีอยู่ในเอกสารมาใช้ให้เกิดประโยชน์มากที่สุด เนื่องจาก Naïve Bayes อื่น นั้นไม่เหมาะสมกับการนำมาจำแนกข้อมูลที่ไม่สมดุลในการสร้างโมเดลมากนัก

4.2.3 CNN Setup

ในส่วนของอัลกอริทึม CNN นั้นจะมีการเซตค่าในการสร้างโมเดลของอัลกอริทึมโดยในแต่ละส่วนของการตั้งค่าได้มีการทดสอบประสิทธิภาพในการตั้งค่าเสมอ จึงจะนำการตั้งค่านั้นไปใช้งานจริง โดยการทดสอบในงานปริญาานิพนธ์นี้ได้ใช้ตัว Conv1D ซึ่งเป็นตัวที่ถูกใช้สำหรับ NLP มากที่สุด โดยเราได้กำหนด `kernel_size = 4` เนื่องจากมีประสิทธิภาพที่ดีและใช้เวลาสั้นในการสร้างโมเดล

ตารางที่ 4.2 ค่าเฉลี่ยในการทดลองค่า input ในการทดสอบกับอัลกอริทึม CNN

filters	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
20	0.3104	0.3641	0.3322
30	0.4323	0.4752	0.4511
50	0.6475	0.6654	0.6534

ตารางที่ 4.2 ค่าเฉลี่ยในการทดลองค่า input ในการทดสอบกับอัลกอริทึม CNN (ต่อ)

filters	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
60	0.5497	0.5293	0.5395
70	0.3764	0.3354	0.3549

จากตารางที่ 4.2 จะเห็นได้ว่าเมื่อค่า filters อยู่ในระดับ 50 มีค่าเฉลี่ยสูงที่สุดในการทดสอบ เนื่องจากข้อมูลที่ใช้ในการสร้างโมเดลนั้น มีข้อมูลอยู่ในระดับกลาง ทำให้การที่ค่า filters น้อยหรือมากจนเกินไปจะทำให้ประสิทธิภาพของข้อมูลลดลง ดังนั้นในปัญหานี้จึงเลือกค่า filters = 50

4.3 ผลการทดลอง (Results)

ในหัวข้อนี้จะกล่าวถึงผลการทดลองทั้งหมดในระบบการจำแนกบทวิจารณ์อิเล็กทรอนิกส์ ซึ่งได้มีการสร้างโมเดลโดยใช้อัลกอริทึม Naïve Bayes อัลกอริทึมการหาเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbor: KNN) และอัลกอริทึมโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN) ดังหัวข้อต่อไปนี้

4.3.1 การทดสอบโมเดลในการจำแนกบทวิจารณ์โดยอัลกอริทึม KNN

สำหรับโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม ได้แก่ Positive class และ Negative class ซึ่งจะใช้เอกสารในการสร้างโมเดลตามสัดส่วนของเอกสารที่ไม่สมดุลกัน โดยจะให้ Positive class เป็นคลาสหลัก ที่มีเอกสาร 500 เอกสาร และให้ Negative class เป็นคลาสรองที่มีสัดส่วนเอกสารเป็นร้อยละ 10 20 และ 30 ของคลาสหลัก โดยใช้อัลกอริทึม KNN ในการทำนายเอกสารที่มีการให้คะแนนค่า

ในขั้นตอนการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม จะใช้เอกสารในการทดสอบจำนวน 1000 เอกสาร ซึ่งแบ่งออกเป็น 2 กลุ่ม จำนวนกลุ่มละ 500 เอกสาร เพื่อหาค่าความระลึก ค่าความแม่นยำ และค่า F-measure ในการประเมินความถูกต้องในการวิเคราะห์

ตารางที่ 4.3 ผลการทดสอบด้วยอัลกอริทึม KNN

การให้น้ำหนัก ค่า	สัดส่วนเอกสารที่ใช้ ในการสร้างโมเดล (ร้อยละ)	จำนวนFeature ที่ใช้ในการสร้าง โมเดล	เวลาที่ใช้ในการ สร้างโมเดล (นาท)	เวลาที่ใช้ในการ ทดสอบโมเดล (นาท)	ค่าความระลึก	ค่าความ แม่นยำ	ค่าเฉลี่ย F-measure
TF-IDF	100:10	1924	1.44	0.11	0.5162	0.5021	0.5074
	100:20	2081	1.54	0.12	0.5447	0.5246	0.5342
	100:30	2119	2.04	0.14	0.5941	0.5702	0.5801
	ค่าเฉลี่ย				0.5462	0.5346	0.5403
Delta TF-IDF	100:10	1924	1.38	0.10	0.5562	0.5544	0.5546
	100:20	2081	1.49	0.15	0.5714	0.5804	0.5766
	100:30	2119	2.14	0.14	0.5922	0.5752	0.5812
	ค่าเฉลี่ย				0.5566	0.5550	0.5574
TF-ICF-IDF	100:10	1924	1.40	0.12	0.5610	0.5532	0.5564
	100:20	2081	1.58	0.15	0.6012	0.5830	0.5912
	100:30	2119	2.10	0.14	0.6332	0.6242	0.6262
	ค่าเฉลี่ย				0.5967	0.5834	0.5866

ตารางที่ 4.3 ผลการทดสอบด้วยอัลกอริทึม KNN (ต่อ)

การให้น้ำหนัก ค่า	สัดส่วนเอกสารที่ใช้ ในการสร้างโมเดล (ร้อยละ)	จำนวนFeature ที่ใช้ในการสร้าง โมเดล	เวลาที่ใช้ในการ สร้างโมเดล (นาทื)	เวลาที่ใช้ในการ ทดสอบโมเดล (นาทื)	ค่าความระลึก	ค่าความ แม่นยำ	ค่าเฉลี่ย F-measure
TF-RF	100:10	1924	1.37	0.14	0.6401	0.6410	0.6403
	100:20	2081	1.52	0.13	0.6711	0.6862	0.6812
	100:30	2119	2.07	0.15	0.7035	0.7046	0.7062
	ค่าเฉลี่ย				0.6763	0.6734	0.6724
TF- IGM	100:10	1924	1.39	0.13	0.6456	0.6684	0.6594
	100:20	2081	1.50	0.13	0.6803	0.6614	0.6703
	100:30	2119	2.02	0.15	0.7045	0.7164	0.7021
	ค่าเฉลี่ย				0.6734	0.6794	0.6782

จากผลการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ที่มีข้อมูลไม่สมดุลแบบ 2 กลุ่ม โดยใช้อัลกอริทึม KNN ดังตารางที่ 4.3 จะเห็นว่าการให้น้ำหนักค่า TF-IGM มีค่า F-measure สูงสุดในทุกสัดส่วนเอกสารที่ใช้ในการสร้างโมเดลด้วยอัลกอริทึม KNN โดยมีค่าเฉลี่ย F-measure อยู่ที่ 0.7052

4.3.2 การทดสอบโมเดลในการจำแนกบทวิจารณ์โดยอัลกอริทึม Naïve Bayes

สำหรับโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม ได้แก่ Positive class และ Negative class ซึ่งจะใช้เอกสารในการสร้างโมเดลตามสัดส่วนของเอกสารที่ไม่สมดุลกัน โดยจะให้ Positive class เป็นคลาสหลัก ที่มีเอกสาร 500 เอกสาร และให้ Negative class เป็นคลาสรองที่มีสัดส่วนเอกสารเป็นร้อยละ 10 20 และ 30 ของคลาสหลัก โดยใช้อัลกอริทึม *Naïve Bayes* ในการทำนายเอกสารที่มีการให้น้ำหนักค่า

ในขั้นตอนการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม จะใช้เอกสารในการทดสอบจำนวน 1000 เอกสาร ซึ่งแบ่งออกเป็น 2 กลุ่ม จำนวนกลุ่มละ 500 เอกสาร เพื่อหาค่าความระลึก ค่าความแม่นยำ และค่า *F-measure* ในการประเมินความถูกต้องในการวิเคราะห์

ตารางที่ 4.4 ผลการทดสอบด้วยอัลกอริทึม *Naïve Bayes*

การให้น้ำหนักคำ	สัดส่วนเอกสารที่ใช้ในการสร้างโมเดล (ร้อยละ)	จำนวนFeature ที่ใช้ในการสร้างโมเดล	เวลาที่ใช้ในการสร้างโมเดล (นาท)	เวลาที่ใช้ในการทดสอบโมเดล (นาท)	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
TF-IDF	100:10	1924	1.08	0.08	0.5546	0.5350	0.5421
	100:20	2081	1.38	0.09	0.5747	0.5542	0.5632
	100:30	2119	1.45	0.09	0.6304	0.6346	0.6323
	ค่าเฉลี่ย				0.5767	0.5764	0.5737
Delta TF-IDF	100:10	1924	1.07	0.08	0.5431	0.5294	0.5374
	100:20	2081	1.32	0.09	0.6466	0.6546	0.6575
	100:30	2119	1.44	0.08	0.7045	0.6978	0.7068
	ค่าเฉลี่ย				0.6369	0.6268	0.6274
TF-ICF-IDF	100:10	1924	1.10	0.09	0.6143	0.6233	0.6176
	100:20	2081	1.29	0.08	0.6436	0.6312	0.6366
	100:30	2119	1.39	0.08	0.6744	0.6561	0.6674
	ค่าเฉลี่ย				0.6424	0.6337	0.6339

ตารางที่ 4.4 ผลการทดสอบด้วยอัลกอริทึม *Naïve Bayes* (ต่อ)

การให้น้ำหนัก คำ	สัดส่วนเอกสารที่ใช้ ในการสร้างโมเดล (ร้อยละ)	จำนวนFeature ที่ใช้ในการสร้าง โมเดล	เวลาที่ใช้ในการ สร้างโมเดล (นาท)	เวลาที่ใช้ในการ ทดสอบโมเดล (นาท)	ค่าความ ระลึก	ค่าความ แม่นยำ	ค่าเฉลี่ย F-measure
<i>TF-RF</i>	100:10	1924	1.07	0.08	0.6332	0.6242	0.6262
	100:20	2081	1.34	0.08	0.6811	0.6862	0.6812
	100:30	2119	1.47	0.08	0.7135	0.7146	0.7122
	ค่าเฉลี่ย				0.6763	0.6734	0.6732
<i>TF-IGM</i>	100:10	1924	1.39	0.09	0.6477	0.6345	0.6412
	100:20	2081	1.48	0.09	0.6716	0.6764	0.6735
	100:30	2119	2.05	0.10	0.7548	0.7068	0.7282
	ค่าเฉลี่ย				0.6913	0.6725	0.6809

จากผลการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ที่มีข้อมูลไม่สมดุลแบบ 2 กลุ่ม โดยใช้อัลกอริทึม *Naïve Bayes* ดังตารางที่ 4.4 จะเห็นว่า การให้น้ำหนักคำ *TF-IGM* มีค่าเฉลี่ย *F-measure* สูงที่สุดอยู่ที่ 0.6809

4.3.3 การทดสอบโมเดลในการจำแนกบทวิจารณ์โดยอัลกอริทึม CNN

สำหรับโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม ได้แก่ Positive class และ Negative class ซึ่งจะใช้เอกสารในการสร้างโมเดลตามสัดส่วนของเอกสารที่ไม่สมดุลกัน โดยจะให้ Positive class เป็นคลาสหลัก ที่มีเอกสาร 500 เอกสาร และให้ Negative class เป็นคลาสรองที่มีสัดส่วนเอกสารเป็นร้อยละ 10 20 และ 30 ของคลาสหลัก โดยใช้อัลกอริทึม CNN ในการทำนายเอกสารที่มีการให้น้ำหนักค่า

ในขั้นตอนการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม จะใช้เอกสารในการทดสอบจำนวน 1000 เอกสาร ซึ่งแบ่งออกเป็น 2 กลุ่ม จำนวนกลุ่มละ 500 เอกสาร เพื่อหาค่าความระลึก ค่าความแม่นยำ และค่า *F-measure* ในการประเมินความถูกต้องในการวิเคราะห์

ตารางที่ 4.5 ผลการทดสอบด้วยอัลกอริทึม CNN

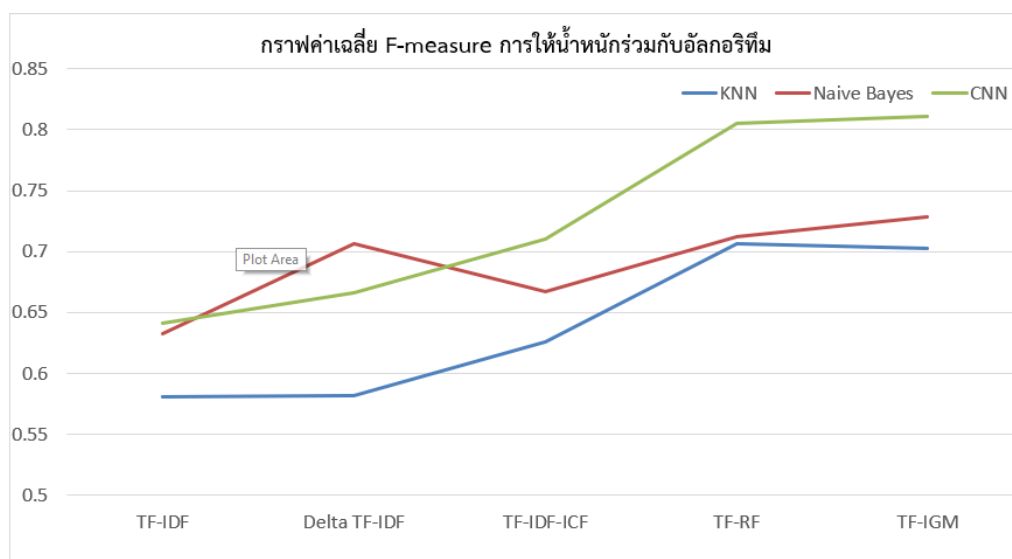
การให้น้ำหนักคำ	สัดส่วนเอกสารที่ใช้ในการสร้างโมเดล (ร้อยละ)	จำนวนFeature ที่ใช้ในการสร้างโมเดล	เวลาที่ใช้ในการสร้างโมเดล (นาท)	เวลาที่ใช้ในการทดสอบโมเดล (นาท)	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
TF-IDF	100:10	1924	2.08	1.05	0.5562	0.5644	0.5546
	100:20	2081	2.58	1.06	0.5914	0.6304	0.6066
	100:30	2119	3.45	1.05	0.6398	0.6202	0.6412
	ค่าเฉลี่ย				0.5966	0.6150	0.6074
Delta TF-IDF	100:10	1924	2.42	1.05	0.5610	0.5832	0.5764
	100:20	2081	3.44	1.05	0.6112	0.6530	0.6412
	100:30	2119	3.55	1.06	0.6632	0.6742	0.6662
	ค่าเฉลี่ย				0.6267	0.6334	0.6266
TF-ICF-IDF	100:10	1924	2.42	1.06	0.6342	0.6384	0.6362
	100:20	2081	3.48	1.07	0.6871	0.6872	0.6852
	100:30	2119	4.25	1.07	0.7135	0.7066	0.7102
	ค่าเฉลี่ย				0.6782	0.6774	0.6761

ตารางที่ 4.5 ผลการทดสอบด้วยอัลกอริทึม CNN (ต่อ)

การให้น้ำหนัก ค่า	สัดส่วนเอกสารที่ใช้ ในการสร้างโมเดล (ร้อยละ)	จำนวนFeature ที่ใช้ในการสร้าง โมเดล	เวลาที่ใช้ในการ สร้างโมเดล (นาท)	เวลาที่ใช้ในการ ทดสอบโมเดล (นาท)	ค่าความ ระลึก	ค่าความ แม่นยำ	ค่าเฉลี่ย F-measure
TF-RF	100:10	1924	2.22	1.06	0.6221	0.6512	0.6354
	100:20	2081	3.38	1.05	0.7398	0.7130	0.7212
	100:30	2119	4.05	1.05	0.8132	0.7954	0.8054
	ค่าเฉลี่ย				0.7257	0.7198	0.7282
TF- IGM	100:10	1924	2.52	1.06	0.6552	0.6752	0.6652
	100:20	2081	3.48	1.06	0.7598	0.7430	0.7512
	100:30	2119	4.35	1.06	0.8142	0.8214	0.8112
	ค่าเฉลี่ย				0.7430	0.7438	0.7431

จากผลการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ที่มีข้อมูลไม่สมดุลแบบ 2 กลุ่ม โดยใช้อัลกอริทึม CNN ดังตารางที่ 4.5 จะเห็นว่าการให้น้ำหนักค่า TF-IGM มีค่า F-measure สูงสุดในทุกสัดส่วนเอกสารที่ใช้ในการสร้างโมเดลด้วยอัลกอริทึม CNN โดยมีค่าเฉลี่ย F-measure อยู่ที่ 0.7431

4.3.4 ภาพรวมผลการทดลอง



ภาพประกอบที่ 4.2 กราฟค่าเฉลี่ย *F-measure* การให้น้ำหนักร่วมกับอัลกอริทึม

จากภาพประกอบที่ 4.2 และตารางที่ 4.6 จะเห็นได้ว่าการให้น้ำหนักในรูปแบบต่างๆ มีประสิทธิภาพที่ดีในแต่ละอัลกอริทึมที่แตกต่างกัน ยกเว้นการให้น้ำหนัก *TF-IGM* ที่มีประสิทธิภาพที่ดีในทุกอัลกอริทึม

ตารางที่ 4.6 ตารางค่าเฉลี่ย *F-measure* การให้น้ำหนักร่วมกับอัลกอริทึม

	<i>KNN</i>	<i>Naive Bayes</i>	<i>CNN</i>
<i>TF-IDF</i>	0.5801	0.6323	0.6412
<i>Delta TF-IDF</i>	0.5812	0.7068	0.6662
<i>TF-IDF-ICF</i>	0.6262	0.6674	0.7102
<i>TF-RF</i>	0.7062	0.7122	0.8054
<i>TF-IGM</i>	0.7021	0.7282	0.8112

4.4 การทดสอบการจำแนกบทวิจารณ์ที่มีข้อมูลที่ต่างกัน 3 ชุดข้อมูลในทุกสัดส่วน

เนื่องจากอัลกอริทึมที่ใช้ในงานปริญาานิพนธ์นี้นั้น มีการกล่าวถึงการเพิ่มประสิทธิภาพของข้อมูลหากมีข้อมูลในการสร้างโมเดลที่มากขึ้น ดังนั้นในหัวข้อนี้จะทำการทดสอบการสร้างโมเดลที่มีข้อมูลในแต่ละสัดส่วนต่างกั้ดังต่อไปนี้

4.4.1 ทดสอบโมเดลกับ 3 สัดส่วนด้วยข้อมูล 3 ชุดที่ต่างกันกับอัลกอริทึม KNN

สำหรับการทดสอบโมเดลที่มีสัดส่วน 100 : 10, 100 : 20 และ 100 : 30 กับชุดข้อมูล 3 ชุด โดยข้อมูลกลุ่มหลักใช้ 100, 250 และ 500 เอกสาร และข้อมูลกลุ่มรองใช้ 10, 25 และ 50 เอกสาร ซึ่งจะสร้างโมเดลและทดสอบในอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุด (KNN)

โดยในขั้นตอนการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม จะใช้เอกสารในการทดสอบจำนวน 1000 เอกสาร ซึ่งแบ่งออกเป็น 2 กลุ่ม จำนวนกลุ่มละ 500 เอกสาร เพื่อหาค่าความระลึก ค่าความแม่นยำ และค่า *F-measure* ในการประเมินความถูกต้องในการวิเคราะห์

ตารางที่ 4.7 ทดสอบโมเดลที่มีสัดส่วน 100 : 10 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม KNN

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM
100 : 10	0.30	0.33	0.40	0.44	0.32	0.50	0.48	0.44	0.38	0.50	0.40	0.41	0.42	0.41	0.38
250 : 25	0.48	0.50	0.46	0.46	0.54	0.52	0.49	0.47	0.48	0.51	0.48	0.51	0.46	0.47	0.52
500 : 50	0.51	0.55	0.56	0.64	0.64	0.52	0.55	0.55	0.64	0.66	0.51	0.55	0.55	0.64	0.65

ตารางที่ 4.8 ทดสอบโมเดลที่มีสัดส่วน 100 : 20 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม KNN

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM
100 : 10	0.45	0.52	0.47	0.49	0.52	0.47	0.50	0.48	0.48	0.48	0.46	0.51	0.47	0.48	0.50
250 : 25	0.50	0.54	0.51	0.54	0.55	0.49	0.54	0.50	0.55	0.52	0.49	0.54	0.50	0.54	0.53
500 : 50	0.54	0.57	0.60	0.67	0.68	0.52	0.58	0.58	0.68	0.66	0.53	0.57	0.59	0.68	0.67

ตารางที่ 4.9 ทดสอบโมเดลที่มีสัดส่วน 100 : 30 กับข้อมูล 3 ชุดที่ต่างกันกับทุกอัลกอริทึม KNN

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM
100 : 10	0.52	0.57	0.49	0.60	0.62	0.51	0.56	0.52	0.57	0.54	0.52	0.56	0.51	0.58	0.58
250 : 25	0.54	0.60	0.51	0.63	0.65	0.56	0.58	0.57	0.59	0.62	0.55	0.59	0.54	0.61	0.63
500 : 50	0.59	0.65	0.63	0.70	0.70	0.57	0.64	0.62	0.70	0.71	0.58	0.64	0.62	0.70	0.70

ในการทดสอบกับอัลกอริทึม KNN ดังตารางที่ 4.7 - ตารางที่ 4.9 นั้นเห็นได้ชัดว่าหากข้อมูลที่ใช้ในการสร้างโมเดลมีน้อยจะทำให้ประสิทธิภาพการทำงานของอัลกอริทึมลดลง และในขณะที่ข้อมูลในการสร้างโมเดลมีมากขึ้นประสิทธิภาพในการสร้างโมเดลก็ยิ่งเพิ่มขึ้นเช่นกัน โดยในการทดสอบกับโมเดลที่มีสัดส่วน 100:10, 100:20 และ 100:30 นั้นในชุดข้อมูลกลุ่มหลักที่มีขนาด 100 และ 250 เอกสาร นั้นการให้น้ำหนักค่าแบบ *Delta TF-IDF* สามารถถึงประสิทธิภาพออกมาได้มากขึ้นในขณะที่การให้น้ำหนักแบบ *TF-IGM* ไม่สามารถถึงประสิทธิภาพออกมาได้เท่าที่ควร แต่ในชุดข้อมูลกลุ่มหลักที่มีขนาด 500 เอกสาร การให้น้ำหนัก TF-RF และ TF-IGM ก็ให้ประสิทธิภาพที่ดีในทุกสัดส่วน

4.4.2 ทดสอบโมเดลกับ 3 สัดส่วนด้วยข้อมูล 3 ชุดที่ต่างกันกับอัลกอริทึมนาอูฟเบย์

สำหรับการทดสอบโมเดลที่มีสัดส่วน 100 : 10, 100 : 20 และ 100 : 30 กับชุดข้อมูล 3 ชุด โดยข้อมูลกลุ่มหลักใช้ 100, 250 และ 500 เอกสาร และข้อมูลกลุ่มรองใช้ 10, 25 และ 50 เอกสาร ซึ่งจะสร้างโมเดลและทดสอบในอัลกอริทึมนาอูฟเบย์

โดยในขั้นตอนการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม จะใช้เอกสารในการทดสอบจำนวน 1000 เอกสาร ซึ่งแบ่งออกเป็น 2 กลุ่ม จำนวนกลุ่มละ 500 เอกสาร เพื่อหาค่าความระลึก ค่าความแม่นยำ และค่า *F-measure* ในการประเมินความถูกต้องในการวิเคราะห์

ตารางที่ 4.10 ทดสอบโมเดลที่มีสัดส่วน 100 : 10 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึมนาอ์ฟเบย์

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF- ICF	TF-RF	TF-IGM	TF- IDF	Delta TF-IDF	TF-IDF- ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF- ICF	TF-RF	TF-IGM
100 : 10	0.43	0.45	0.48	0.44	0.46	0.43	0.42	0.48	0.47	0.46	0.43	0.42	0.48	0.45	0.46
250 : 25	0.47	0.50	0.52	0.51	0.49	0.47	0.51	0.52	0.51	0.50	0.47	0.50	0.52	0.51	0.49
500 : 50	0.55	0.54	0.61	0.63	0.64	0.53	0.55	0.62	0.62	0.63	0.54	0.54	0.61	0.62	0.64

ตารางที่ 4.11 ทดสอบโมเดลที่มีสัดส่วน 100 : 20 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึมนาอ์ฟเบย์

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF- IDF	TF-IDF- ICF	TF-RF	TF-IGM	TF- IDF	Delta TF-IDF	TF-IDF- ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF- ICF	TF-RF	TF-IGM
100 : 10	0.46	0.48	0.51	0.48	0.50	0.47	0.48	0.51	0.49	0.49	0.46	0.48	0.51	0.48	0.49
250 : 25	0.48	0.51	0.54	0.52	0.57	0.49	0.52	0.56	0.54	0.56	0.48	0.51	0.55	0.53	0.56
500 : 50	0.57	0.64	0.64	0.68	0.67	0.55	0.65	0.63	0.68	0.67	0.56	0.65	0.63	0.68	0.67

ตารางที่ 4.12 ทดสอบโมเดลที่มีสัดส่วน 100 : 30 กับข้อมูล 3 ชุดที่ต่างกันกับทุกอัลกอริทึมนาอ์ฟเบย์

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM
100 : 10	0.48	0.51	0.51	0.50	0.47	0.48	0.51	0.53	0.49	0.47	0.48	0.51	0.52	0.49	0.47
250 : 25	0.53	0.58	0.57	0.60	0.64	0.51	0.57	0.55	0.57	0.63	0.52	0.57	0.56	0.58	0.63
500 : 50	0.63	0.64	0.67	0.71	0.75	0.63	0.69	0.65	0.71	0.70	0.63	0.66	0.66	0.71	0.72

ในการทดสอบกับอัลกอริทึมนาอ์ฟเบย์ ดังตารางที่ 4.10 - ตารางที่ 4.12 นั้นเห็นได้ชัดว่าหากข้อมูลที่ใช้ในการสร้างโมเดลหากมีน้อยจะทำให้ประสิทธิภาพการทำงานของอัลกอริทึมลดลง และในขณะที่ข้อมูลในการสร้างโมเดลมีมากขึ้นประสิทธิภาพในการสร้างโมเดลก็ยิ่งเพิ่มขึ้นเช่นกัน โดยในการทดสอบกับโมเดลที่มีสัดส่วน 100:10, 100:20 และ 100:30 นั้นในชุดข้อมูลกลุ่มหลักที่มีขนาด 100 นั้นการให้น้ำหนักค่าแบบ *Delta TF-IDF* และ *TF-IDF-ICF* สามารถดึงประสิทธิภาพออกมาได้มากขึ้นในขณะที่การให้น้ำหนักแบบอื่น ไม่สามารถดึงประสิทธิภาพออกมาได้เท่าที่ควร แต่ในชุดข้อมูลกลุ่มหลักที่มีขนาด 250 และ 500 เอกสาร การให้น้ำหนัก TF-RF และ TF-IGM กับให้ประสิทธิภาพที่ดีในทุกสัดส่วน และการให้น้ำหนัก *TF-IDF* ก็ยังคงให้ประสิทธิภาพการทำงานร่วมกับอัลกอริทึมน้อยเช่นเดิม

4.4.3 ทดสอบโมเดลกับ 3 สัดส่วนด้วยข้อมูล 3 ชุดที่ต่างกับกับ CNN

สำหรับการทดสอบโมเดลที่มีสัดส่วน 100 : 10, 100 : 20 และ 100 : 30 กับชุดข้อมูล 3 ชุด โดยข้อมูลกลุ่มหลักใช้ 100, 250 และ 500 เอกสาร และข้อมูลกลุ่มรองใช้ 10, 25 และ 50 เอกสาร ซึ่งจะสร้างโมเดลและทดสอบในอัลกอริทึม CNN

โดยในขั้นตอนการทดสอบโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์แบบ 2 กลุ่ม จะใช้เอกสารในการทดสอบจำนวน 1000 เอกสาร ซึ่งแบ่งออกเป็น 2 กลุ่ม จำนวนกลุ่มละ 500 เอกสาร เพื่อหาค่าความระลึก ค่าความแม่นยำ และค่า *F-measure* ในการประเมินความถูกต้องในการวิเคราะห์

ตารางที่ 4.13 ทดสอบโมเดลที่มีสัดส่วน 100 : 10 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม CNN

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM
100 : 10	0.43	0.45	0.48	0.44	0.46	0.43	0.42	0.48	0.47	0.46	0.43	0.42	0.48	0.45	0.46
250 : 25	0.47	0.50	0.52	0.51	0.49	0.47	0.51	0.52	0.51	0.50	0.47	0.50	0.52	0.51	0.49
500 : 50	0.55	0.54	0.61	0.63	0.64	0.53	0.55	0.62	0.62	0.63	0.54	0.54	0.61	0.62	0.64

ตารางที่ 4.14 ทดสอบโมเดลที่มีสัดส่วน 100 : 20 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม CNN

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM
100 : 10	0.46	0.48	0.51	0.48	0.50	0.47	0.48	0.51	0.49	0.49	0.46	0.48	0.51	0.48	0.49
250 : 25	0.48	0.51	0.54	0.52	0.57	0.49	0.52	0.56	0.54	0.56	0.48	0.51	0.55	0.53	0.56
500 : 50	0.57	0.64	0.64	0.68	0.67	0.55	0.65	0.63	0.68	0.67	0.56	0.65	0.63	0.68	0.67

ตารางที่ 4.15 ทดสอบโมเดลที่มีสัดส่วน 100 : 30 กับข้อมูล 3 ชุดที่ต่างกับกับทุกอัลกอริทึม *CNN*

P : N	Recall					Precision					F-measure				
	TF-IDF	Delta TF-IDF	TF-IDF- ICF	TF-RF	TF-IGM	TF- IDF	Delta TF-IDF	TF-IDF-ICF	TF-RF	TF-IGM	TF-IDF	Delta TF-IDF	TF-IDF- ICF	TF-RF	TF-IGM
100 : 10	0.48	0.51	0.51	0.50	0.47	0.48	0.51	0.53	0.49	0.47	0.48	0.51	0.52	0.49	0.47
250 : 25	0.53	0.58	0.57	0.60	0.64	0.51	0.57	0.55	0.57	0.63	0.52	0.57	0.56	0.58	0.63
500 : 50	0.63	0.64	0.67	0.71	0.75	0.63	0.69	0.65	0.71	0.70	0.63	0.66	0.66	0.71	0.72

ในการทดสอบกับอัลกอริทึม *CNN* ดังตารางที่ 4.13 - ตารางที่ 4.15 นั้นเห็นได้ชัดว่าหากข้อมูลที่ใช้ในการสร้างโมเดลมีน้อยจะทำให้ประสิทธิภาพการทำงานของอัลกอริทึมลดลง และในขณะที่ข้อมูลในการสร้างโมเดลมีมากขึ้นประสิทธิภาพในการสร้างโมเดลก็ยิ่งเพิ่มขึ้นเช่นกัน โดยในการทดสอบกับโมเดลที่มีสัดส่วน 100:10, 100:20 และ 100:30 นั้นในชุดข้อมูลกลุ่มหลักที่มีขนาด 100, 250 และ 500 เอกสาร นั้นการให้น้ำหนักค่าแบบ *Delta TF-IDF*, *TF-IDF-ICF*, *TF-RF* และ *TF-IGM* ให้ประสิทธิภาพที่ดีในทุกสัดส่วน

4.5 การวิเคราะห์ผล

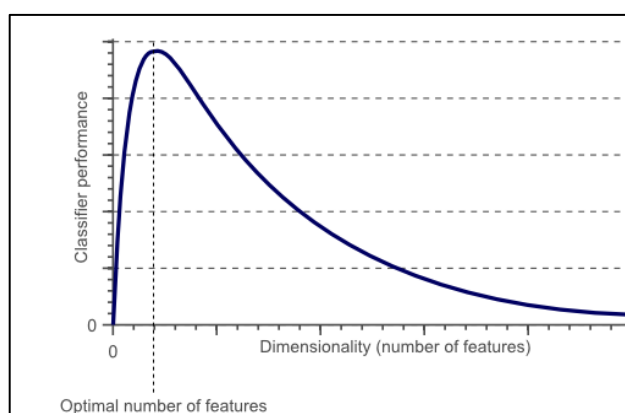
จากผลการทดสอบจะเห็นได้ว่ากรณีที่อัตราส่วนของข้อมูลที่สูงขึ้นนั้นส่งผลให้ประสิทธิภาพของอัลกอริทึมดีขึ้น และถ้าหากข้อมูลที่ใช้ในการสร้างมีมากขึ้นก็ส่งผลให้มีประสิทธิภาพที่ดีขึ้นเช่นกันต่อให้มีการไม่สมดุลของข้อมูลมากก็ตาม ซึ่งในการวิเคราะห์ผลนั้นประกอบไปด้วย

1) วิเคราะห์เกี่ยวกับวิธีการให้น้ำหนักคำ

- สำหรับรูปแบบการให้น้ำหนักคำทั้ง 5 รูปแบบจะเห็นได้ชัดว่ารูปแบบการให้น้ำหนักคำ *TF-IGM* มีค่าเฉลี่ยสูงสุดในทุกอัลกอริทึมเนื่องจากรูปแบบการให้น้ำหนักคำแบบ *TF-IGM* นั้น ถูกนำเสนอให้วัดความไม่สม่ำเสมอหรือความเข้มข้นของการแจกแจงคำศัพท์ระหว่างคลาสซึ่งสะท้อนให้เห็นถึงอำนาจการจำแนกชั้นข้อตกลง จึงทำให้เห็นความชัดเจนของการแยกข้อมูลในแต่ละคลาสเป็นอย่างดี ซึ่งเมื่อนำรูปแบบการให้น้ำหนักคำไปใช้กับอัลกอริทึม *CNN* แล้วทำให้เห็นว่าหากเอกสารมีข้อมูลไม่สมดุลทำให้การให้น้ำหนักคำแบบ *TF-IGM* ที่ใช้ร่วมกับอัลกอริทึม *CNN* สามารถแก้ปัญหาได้ดีที่สุด เมื่อเอกสารมีสัดส่วนที่ 100: 10 โดยมีค่าเฉลี่ยอยู่ที่ 0.6652 เมื่อเทียบกับรูปแบบอื่นๆ
- รองลงมาคือรูปแบบการให้น้ำหนักคำแบบ *TF-ICF-IDF* ที่มีค่าเฉลี่ยอยู่ที่ 0.6362 และรูปแบบที่มีค่าเฉลี่ยต่ำสุดคือ *TF-IDF* ที่มีค่าเฉลี่ยอยู่ที่ 0.5546
- สำหรับรูปแบบการให้น้ำหนักที่มีค่าเฉลี่ยมากที่สุดที่ทดสอบกับชุดข้อมูลมีสัดส่วน 100:20 และ 100:30 คือการให้น้ำหนักคำ *TF-IGM* ที่ทดสอบร่วมกับอัลกอริทึม *CNN* เช่นเดียวกับสัดส่วน 100:10 โดยมีค่า *F-measure* อยู่ที่ 0.7512 และ 0.8112 ตามลำดับ ซึ่งสัดส่วน 100:30 เป็นค่าที่สูงที่สุดในการทดสอบรูปแบบการให้ทั้งหมด และเห็นได้ชัดว่าหากข้อมูลมีความไม่สมดุลต่างกันน้อยจะให้การจำแนกข้อมูลมีประสิทธิภาพมาก
- ส่วนการให้น้ำหนัก *TF-IDF* นั้นมีประสิทธิภาพต่ำที่สุดในทุกอัลกอริทึม เนื่องจากการให้น้ำหนัก *TF-IDF* นั้นเป็นการให้น้ำหนักคำที่คิดจากความถี่ของคำทั้งหมดของคลังเอกสาร นั้นทำให้การให้น้ำหนักในรูปแบบนี้มีประสิทธิภาพที่ต่ำกว่ารูปแบบอื่นที่คิดจากความถี่ของคำในแต่ละคลาส
- ดังนั้นสรุปได้ว่ารูปแบบการให้น้ำหนัก *UTW (TF-IDF)* นั้นให้ประสิทธิภาพต่ำกว่ารูปแบบการให้น้ำหนัก *STW (Delta TF-IDF, TF-IDF-ICF, TF-RF และ TF-IGM)* เนื่องจากการให้น้ำหนักคำรูปแบบ *UTW* เป็นการให้น้ำหนักที่คิดจากข้อมูลทั้งหมดในคลังข้อมูลแต่ในส่วนของ *STW* เป็นการให้น้ำหนักคำที่คิดจากกลุ่มของเอกสารเป็นหลัก ซึ่งเหมาะสมกับการใช้ในการแก้ปัญหาในงานปริญาานิพนธ์นี้

2) วิเคราะห์เกี่ยวกับอัลกอริทึม

- สำหรับอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุด จะมีประสิทธิภาพต่ำเมื่อมีคุณลักษณะจำนวนมาก เพราะถูกรบกวนจากคุณลักษณะที่ไม่เกี่ยวข้องได้ง่าย แต่เมื่อนำมาใช้ร่วมกับการให้น้ำหนักค่าแบบ STW จะมีประสิทธิภาพดีขึ้น ตามที่ได้กล่าวไว้ในข้างต้น เพราะว่าอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุดได้รับผลกระทบจากคุณลักษณะที่ไม่เกี่ยวข้อง (Irrelevant feature) ต่อการวัดระยะทาง หรือการเกิดปัญหาของมิติข้อมูล (Curse of Dimensionality) อีกทั้งอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุดเหมาะกับชุดข้อมูลสอนที่มีปริมาณมาก แต่มีตัวอย่างคุณลักษณะจำนวนน้อย ดังภาพประกอบที่ 4.3



ภาพประกอบที่ 4.3 Curse of Dimensionality

- และอีกข้อที่สำคัญคือ การเลือกค่า k เพราะถ้าหากใช้ค่า k น้อยเกินไปอาจทำให้ไวต่อสัญญาณรบกวนได้ และถ้าหากเลือกค่า k มากเกินไปก็อาจจะทำให้มีกลุ่มข้อมูลอื่นๆ มาปะปนกับข้อมูลที่กำลังสนใจได้เช่นกัน

- ต่อมาสำหรับอัลกอริทึมนาอีฟเบย์ จะใช้งานได้ดีเมื่อมีคุณลักษณะจำนวนมาก และคุณลักษณะเป็นอิสระต่อกัน สังเกตได้จากตารางที่ 4.4 จะเห็นว่าอัลกอริทึมนาอีฟเบย์จะมีประสิทธิภาพดีที่สุด เมื่อใช้ร่วมกับการให้น้ำหนักค่าแบบ STW ที่เป็นความถี่ของค่าที่เกิดในแต่ละกลุ่มเท่านั้น แต่ถ้าหากใช้ร่วมกับการให้น้ำหนักค่าที่มีการนำ *global weight* มาคำนวณรวมด้วย อาจจะทำให้ประสิทธิภาพโมเดลลดลง

- สำหรับอัลกอริทึมโครงข่ายประสาทแบบคอนโวลูชัน เมื่อทำการทดสอบกับการให้น้ำหนักค่าแบบ STW แล้วทำให้การจำแนกข้อมูลที่ไม่สมดุลมีประสิทธิภาพที่มากขึ้นเนื่องจาก CNN นั้นมีประสิทธิภาพในการจำแนกข้อมูลที่ไม่สมดุลอยู่แล้ว และหากต้องการให้ CNN มีประสิทธิภาพมากขึ้น

ควรใช้ชุดข้อมูลมีขนาดใหญ่ เนื่องจาก CNN นั้นถูกสร้างมาเพื่อทดสอบกับชุดข้อมูลชุดสอนที่มีขนาดใหญ่

3) วิเคราะห์การใช้งานการให้น้ำหนักร่วมกับอัลกอริทึม

- สำหรับการใช้งานการให้น้ำหนักร่วมกับอัลกอริทึมที่มีค่าเฉลี่ยมากที่สุดคือ การให้น้ำหนัก TF-IGM ร่วมกับอัลกอริทึม CNN ที่มีค่าเฉลี่ย *F-measure* สูงที่สุดอยู่ที่ 0.8112 ซึ่งมีประสิทธิภาพที่สุดที่ได้อธิบายไปข้างต้นแล้ว
- รองลงมาคือ การให้น้ำหนัก TF-RF ร่วมกับอัลกอริทึม CNN เช่นกันกับการให้น้ำหนัก TF-IGM เนื่องจากการให้น้ำหนักของทั้งสองรูปแบบนั้นมีการคำนวณการให้น้ำหนักค่าที่คล้ายคลึงกันจึงทำให้มีประสิทธิภาพที่ใกล้เคียงกันมากที่สุด

4) วิเคราะห์เกี่ยวกับเวลาที่ใช้ในการสร้างและทดสอบ

สำหรับเวลาที่ใช้ในการประมวลผลจะขึ้นอยู่กับปัจจัย ดังนี้

1. จำนวนคุณลักษณะ (Feature)

ถ้าหากมีคุณลักษณะจำนวนมากเวลาที่ใช้ในการประมวลผลก็จะมากขึ้นตามไปด้วย เนื่องจากระบบต้องนำคุณลักษณะเหล่านั้นมาประมวลผล ดังนั้นโครงงานนี้จึงได้มีการลดจำนวนคุณลักษณะด้วยการใช้ *information gain* และการคัดเลือกค่าด้วยพจนานุกรม ซึ่งการลดคุณลักษณะเหล่านี้ ไม่ส่งผลต่อความถูกต้องของการจัดกลุ่มเอกสาร เนื่องจากคุณลักษณะที่ถูกคัดออกไปไม่มีความสำคัญต่อการจัดกลุ่มเอกสาร แต่เป็นข้อมูลจริง ตัวอย่างค่าที่ถูกคัดออก

```
File Edit Format View Help
baddddsandra=1
soooooooooooooooooo =1
wompwomp=1
trejuo=1
hummm=1
zzzzzzzzzzzzzzzzzz=1
jimmy=1
ahhhh=1
wwiiwhy=1
emma=4
s2=1
s3=1
jennysue=1
arghhhhhhhhhhhhhyes=1
wowwwwwwwwwwwww=1
```

ภาพประกอบที่ 4.4 คุณลักษณะที่ไม่ส่งผลต่อการจัดกลุ่ม

2. อัลกอริทึมที่ใช้ในการจัดกลุ่มเอกสาร (Algorithm)

สำหรับอัลกอริทึมนาอีฟเบย์นั้น ในการสร้างและทดสอบโมเดลจะใช้เวลาในการประมวลผลค่อนข้างเร็วเนื่องจากการคำนวณไม่ซับซ้อน ซึ่งแตกต่างจากอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุดที่ใช้ในการสร้างโมเดลจะมีความรวดเร็ว แต่จะใช้เวลาค่อนข้างนานในการทดสอบโมเดล เนื่องจากอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุดจะเป็นการนำเอกสารที่เข้ามาใหม่ไปคำนวณน้ำหนัก แล้ววัดระยะทางระหว่างเอกสารที่เข้ามาใหม่กับทุกเอกสารในโมเดล แล้วนำมาเรียงลำดับทำให้ตอนทดสอบโมเดลใช้เวลามากกว่าตอนสร้างโมเดลนั่นเอง สุดท้ายอันกอริทึมโครงข่ายประสาทคอนโวลูชันใช้เวลาในการสร้างโมเดลค่อนข้างนานเนื่องจากมีความซับซ้อนในการคำนวณอัลกอริทึม แต่เวลาที่ใช้ในการทดสอบโมเดลมีความเร็วใกล้เคียงกับทั้งสองอัลกอริทึม

บทที่ 5

สรุปและอภิปรายผลการทดลอง

ในบทนี้จะเป็นการสรุปภาพรวมของการสร้างโมเดลการจำแนกเอกสารข้อความที่ไม่สมดุล จากข้อความแสดงความคิดเห็นของลูกค้าที่ซื้อสินค้าอิเล็กทรอนิกส์ต่างๆ ที่ได้ทำการรวบรวมไว้ดังนี้

5.1 สรุปผลและอภิปรายผล

โครงงานฉบับนี้ เป็นงานวิจัยทางด้านการแก้ปัญหการจำแนกข้อมูลที่ไม่สมดุล โดยใช้ชุดข้อมูลที่เป็นบทวิจารณ์สินค้าอิเล็กทรอนิกส์ ซึ่งเป็นการสร้างโมเดลที่ไม่มีความสมดุลของข้อมูล ที่มีสัดส่วน 100 : 10, 100 : 20 และ 100 : 30 เพื่อคัดแยกกลุ่มข้อความ โดยใช้อัลกอริทึมอิมพีเบย์ (Naive Bayes) อัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุด (KNN) และอัลกอริทึมโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN) ส่วนการให้น้ำหนักคำจะมีอยู่ 5 รูปแบบหลักคือ TF-IDF, Delta TF-IDF, TF-ICF-IDF, TF-RF และ TF-IGM

ขั้นตอนในการสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุลนั้น ในขั้นตอนแรกจะเป็นการรวบรวมข้อมูลที่เป็นบทวิจารณ์สินค้าอิเล็กทรอนิกส์มาจากเว็บไซต์ Amazon จากนั้นคัดแยกออกเป็นสองกลุ่ม โดยจะมีการแบ่งเอกสารออกเป็น 2 ชุด คือ ชุดข้อมูลสอน (Training) และ ชุดข้อมูลทดสอบ (Test set) โดยชุดข้อมูลสอนจะแบ่งเป็นชุดย่อยสามชุดที่มีขนาดข้อมูลที่ไม่สมดุลกันโดยสัดส่วนข้อมูลชุดหลักที่เป็น Positive class มากกว่าข้อมูลที่เป็นชุดรอง Negative class คือ 100 : 10, 100 : 20 และ 100 : 30 คัดแยกข้อมูลเสร็จเรียบร้อยแล้ว ก็จะนำข้อมูลเข้าสู่ขั้นตอนก่อนการประมวลผลต่อไป

ขั้นตอนก่อนการประมวลผล (Text pre-processing) เป็นการนำเอาเอกสารที่ได้จากขั้นตอนก่อนหน้านี้มาทำการตัดคำ การตัดคำหยุด การคัดเลือกคำด้วยพจนานุกรม และการเลือกคุณลักษณะเพื่อกรองคำที่มีความเกี่ยวข้องกับการจัดกลุ่มเอกสารน้อยที่สุดออกด้วย IG และเพื่อหาจำนวนคำทั้งหมดในเอกสาร โดยเอกสารที่ผ่านกระบวนการนี้จะอยู่ในรูปแบบ *Vector Space Model* เพื่อแสดงให้เห็นถึงความสัมพันธ์ระหว่างเอกสารและคำที่ปรากฏในเอกสาร พร้อมทั้งการให้น้ำหนักของคำเพื่อแสดงว่าคำๆ นั้นมีความสำคัญกับเอกสารมากน้อยเพียงใด โดยการให้น้ำหนักคำจะมีอยู่ 5 รูปแบบคือ TF-IDF, Delta TF-IDF, TF-ICF-IDF, TF-RF และ TF-IGM ซึ่งการให้น้ำหนักคำในเอกสารจะให้น้ำหนักแยกตาม class สำหรับการสร้างโมเดลจะมีอัลกอริทึมที่ใช้ในการสร้างโมเดล 3 อัลกอริทึม คืออัลกอริทึมอิมพีเบย์ (Naive Bayes) อัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุด (KNN) และอัลกอริทึมโครงข่ายประสาทแบบคอนโวลูชัน

ชั้น (Convolutional Neural Network: CNN) ต่อไปจะเข้าสู่ขั้นตอนการจัดเก็บโมเดลเพื่อใช้ในการประมวลผลถัดไป

ขั้นตอนการทดสอบโมเดลเมื่อได้โมเดลการจำแนกข้อมูลที่ไม่สมดุลเกี่ยวกับสินค้าอิเล็กทรอนิกส์เป็นที่เรียบร้อยแล้วจากขั้นตอนข้างต้น สามารถนำมาใช้จัดระดับคะแนนข้อความบทวิจารณ์สินค้าอิเล็กทรอนิกส์ของผู้ซื้อสินค้าอื่นๆ เพื่อให้ทราบว่าบทวิจารณ์นั้นๆ จัดควรอยู่ในกลุ่ม Positive class หรือ Negative class

สำหรับขั้นตอนการวัดประสิทธิภาพในโครงงานนี้จะใช้ การวัดค่าความระลึก ค่าความแม่นยำ และการวัดค่า *F-measure* โดยค่าความระลึกจะเป็นอัตราส่วนของเอกสารที่จัดกลุ่มได้จากเอกสารทั้งหมดที่มีอยู่ ส่วนค่าความแม่นยำเป็นอัตราส่วนของเอกสารที่จัดกลุ่มได้ถูกต้อง จากจำนวนของเอกสารทั้งหมดที่จัดกลุ่มได้ ค่า *F-measure* เป็นการพิจารณาค่าความสัมพันธ์ระหว่างค่าความระลึกและค่าความแม่นยำ


5.2 ปัญหาและอุปสรรคในการดำเนินงาน

5.2.1 ปัญหาเกี่ยวกับอัลกอริทึมในการสร้างโมเดล

เนื่องจากอัลกอริทึม *CNN* และ *KNN* นั้นเหมาะกับการทดลองกับชุดข้อมูลชุดสอนที่มีขนาดใหญ่ แต่ในโครงงานปริญญาโทนี้ทำได้ ทำการทดลองกับชุดข้อมูลชุดสอนที่มีขนาดเล็ก จึงทำให้ไม่สามารถถึงประสิทธิภาพสูงสุดของอัลกอริทึม *CNN* และ *KNN* ออกมาได้ ทั้งนี้เวลาในการสร้างโมเดลนั้นค่อนข้างนาน ดังนั้นควรจะทำการ save model ไว้หากได้ค่าเฉลี่ยที่พึงพอใจแล้ว

5.2.2 ปัญหาเกี่ยวกับชุดข้อมูลที่ใช้ในการสร้างโมเดล

เนื่องจากเอกสารข้อความแสดงความคิดเห็นเกี่ยวกับสินค้าอิเล็กทรอนิกส์ที่รวบรวมมานั้น เป็นข้อความที่ทุกคนที่ซื้อสินค้า สามารถเข้ามาเขียนแสดงความรู้สึกต่อสินค้านั้นๆ ได้ ทำให้เกิดการใช้คำที่ไม่มีความหมาย (Unknown word) และไม่พบในพจนานุกรม ทำให้การสร้างและการทดสอบโมเดลมีความไม่เสถียร ถึงแม้ในโครงงานนี้จะใช้พจนานุกรมในการคัดกรองคำเหล่านั้นแล้วก็ตาม แต่ในอนาคตก็อาจจะมีคำเหล่านี้หลุดเข้าในขั้นตอนการสร้างโมเดลได้



```

File Edit Format View Help
baddddsandra=1
soooooooooooooooooo =1
wompwomp=1
trejuo=1
hummm=1
zzzzzzzzzzzzzzzz=1
jimmy=1
ahhh=1
wwiiwhy=1
emma=4
s2=1
s3=1
jennysue=1
arghhhhhhhhhhhyes=1
wowwwwwwwwwww=1

```

ภาพประกอบที่ 5.1 ตัวอย่างคำที่ไม่มีความหมาย (Unknown word)

5.3 ข้อเสนอแนะ

1. การให้น้ำหนักคำแต่ละรูปแบบ STW ควรมีชุดข้อมูล 2 กลุ่มเป็นต้นไปและมีขนาดข้อมูลที่มีขนาดใหญ่
2. ประสิทธิภาพของโมเดลจะขึ้นอยู่กับจำนวนเอกสารที่ใช้ในการสร้างโมเดล และความถูกต้องของเอกสารที่ใช้สร้างโมเดลด้วย

ดังนั้น การสร้างโมเดลหรือตัวจัดกลุ่มเอกสาร ควรมีจำนวนคำศัพท์ที่จำเป็นสำหรับการจัดกลุ่มปริมาณไม่น้อยจนเกินไป และถ้าหากคำศัพท์ที่รวบรวมมาตรงกับเอกสารข้อความที่ต้องการนำมาวิเคราะห์เพื่อจัดกลุ่ม จะทำให้ประสิทธิภาพในการวิเคราะห์ของโปรแกรมมีมากขึ้น รวมไปถึงจำนวนเอกสารที่ใช้ในขั้นตอนการสร้างโมเดล เพราะโครงงานฉบับนี้นำเสนออัลกอริทึมการเรียนรู้แบบมีผู้สอน

เอกสารอ้างอิง

1. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," May 2002, Accessed: Aug. 06, 2020. [Online]. Available: <http://arxiv.org/abs/cs/0205070>.
2. Y. Li, G. Sun, and Y. Zhu, "Data imbalance problem in text classification," *Proc. - 3rd Int. Symp. Inf. Process. ISIP 2010*, pp. 301–305, 2010, doi: 10.1109/ISIP.2010.47.
3. Y. Liu, H. T. Loh, and A. Sun, "Imbalanced text classification: A term weighting approach," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 690–701, 2009, doi: <https://doi.org/10.1016/j.eswa.2007.10.042>.
4. N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," 2002.
5. R. Longadge and S. Dongre, "Class Imbalance Problem in Data Mining Review," May 2013, Accessed: Aug. 06, 2020. [Online]. Available: <http://arxiv.org/abs/1305.1707>.
6. J. Ah-Pine and E. P. S. Morales, "A study of synthetic oversampling for twitter imbalanced sentiment analysis," *CEUR Workshop Proc.*, vol. 1646, pp. 17–24, 2016.
7. C. Zhang, J. Bi, and P. Soda, *Feature selection and resampling in class imbalance learning: Which comes first? An empirical study in the biological domain*. 2017.
8. F. Ren and M. G. Sohrab, "Class-indexing-based term weighting for automatic text classification," *Inf. Sci. (Ny)*, vol. 236, pp. 109–125, 2013, doi: <https://doi.org/10.1016/j.ins.2013.02.029>.
9. Y. Gu and X. Gu, "A Supervised Term Weighting Scheme for Multi-class Text Categorization BT - Intelligent Computing Methodologies," 2017, pp. 436–447.
10. P. Juszczak and R. P. W. Duin, "Uncertainty sampling methods for one-class classifiers."
11. F. Debole and F. Sebastiani, "Supervised Term Weighting for Automated Text Categorization BT - Text Mining and its Applications," 2004, pp. 81–97.
12. A. C. E. S. Lima and L. N. de Castro, "Automatic sentiment analysis of Twitter

เอกสารอ้างอิง (ต่อ)

- messages,” in *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, 2012, pp. 52–57, doi: 10.1109/CASoN.2012.6412377.
13. M. Ibrahim and M. Carman, “Undersampling Techniques to Re-balance Training Data for Large Scale Learning-to-Rank BT - Information Retrieval Technology,” 2014, pp. 444–457.
14. V. Balakrishnan and L.-Y. Ethel, “Stemming and Lemmatization: A Comparison of Retrieval Performances,” *Lect. Notes Softw. Eng.*, vol. 2, no. 3, pp. 262–267, 2014, doi: 10.7763/lmse.2014.v2.134.
15. F. Sebastiani, “Machine Learning in Automated Text Categorization.” [Online]. Available: www.ira.uka.de/bibliography/Ai/automated.text.
16. G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988, doi: [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
17. G. Domeniconi, G. Moro, R. Pasolini, and C. Sartori, “A Comparison of Term Weighting Schemes for Text Classification and Sentiment Analysis with a Supervised Variant of tf.idf BT - Data Management Technologies and Applications,” 2016, pp. 39–58.
18. M. Lan, C. L. Tan, J. Su, and Y. Lu, “Supervised and Traditional Term Weighting Methods for Automatic Text Categorization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 721–735, 2009, doi: 10.1109/TPAMI.2008.110.
19. J. Martineau, T. Finin, C. Fink, C. Piatko, J. Mayfield, and Z. Syed, “Delta TFIDF: An Improved Feature Space for Sentiment Analysis,” *Proc. Second Int. Conf. Weblogs Soc. Media (ICWSM)*, vol. 29, no. May, pp. 490–497, 2008, [Online]. Available: <http://ebiquity.umbc.edu/papers/select/person/Tim/Finin/>.
20. K. Chen, Z. Zhang, J. Long, and H. Zhang, “Turning from TF-IDF to TF-IGM for term weighting in text classification,” *Expert Syst. Appl.*, vol. 66, pp. 245–260, 2016, doi: <https://doi.org/10.1016/j.eswa.2016.09.009>.
21. T. Dogan and A. K. Uysal, “Improved inverse gravity moment term weighting for text classification,” *Expert Syst. Appl.*, vol. 130, pp. 45–59, 2019, doi: <https://doi.org/10.1016/j.eswa.2019.04.015>.

เอกสารอ้างอิง (ต่อ)

22. D. M. W, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION,” *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011, [Online]. Available: <http://dspace.flinders.edu.au/dspace/http://www.bioinfo.in/contents.php?id=51>.
23. S. Li, G. Zhou, Z. Wang, S. Y. M. Lee, and R. Wang, “Imbalanced sentiment classification,” *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 2469–2472, 2011, doi: 10.1145/2063576.2063994.

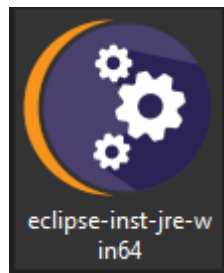
ภาคผนวก

ภาคผนวก ก
คู่มือการติดตั้ง

คู่มือการติดตั้ง

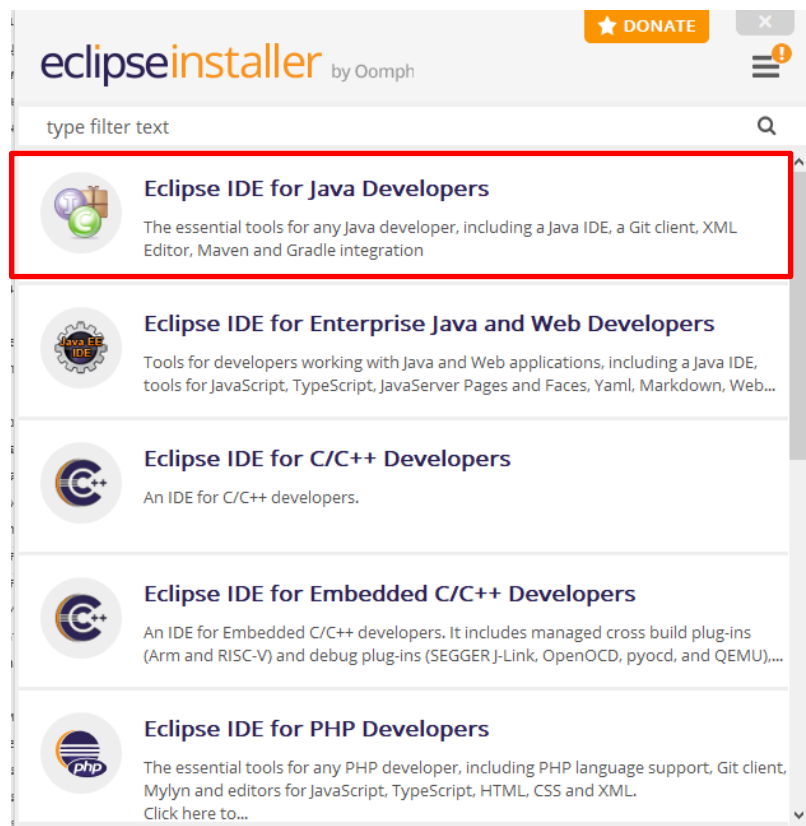
1. ขั้นตอนการติดตั้งโปรแกรม Eclipse

- 1) ทำการติดตั้งไฟล์ โดยการคลิกขวา run as administrator ที่ชื่อไฟล์ eclipse-inst-jre-win64



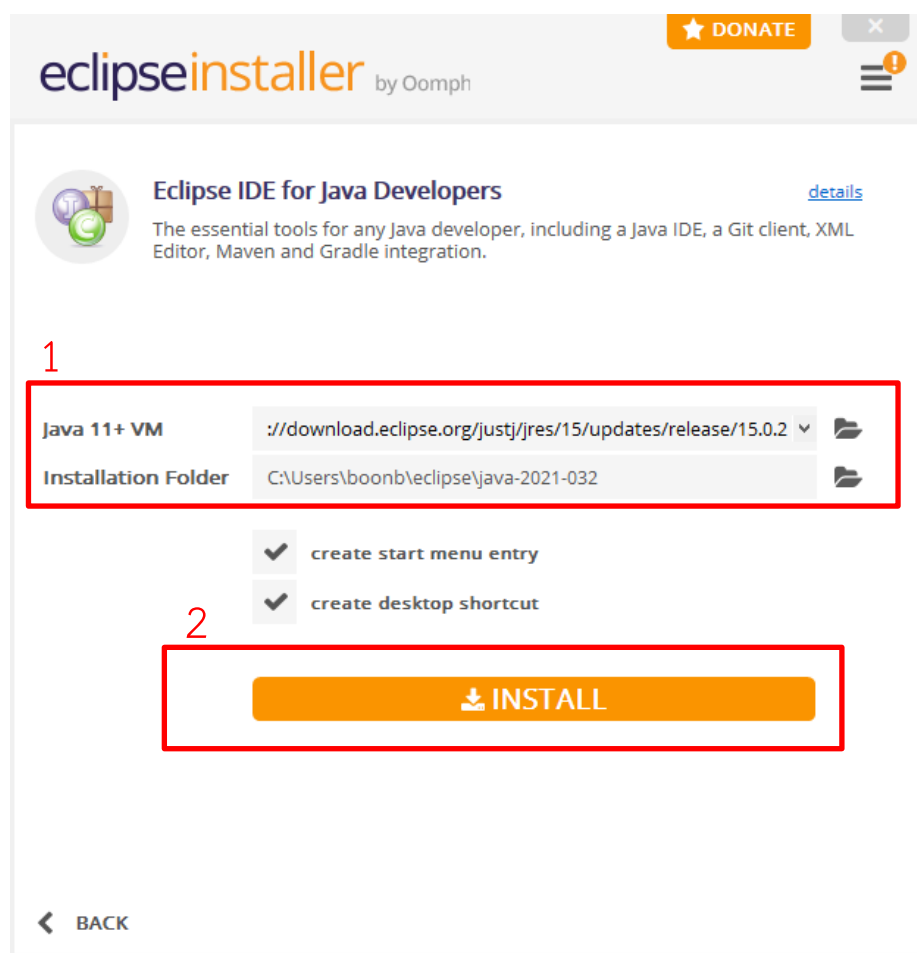
ภาพประกอบที่ ก-1 ไฟล์ Eclipse สำหรับติดตั้ง

- 2) จากนั้นกดเลือก Eclipse IDE for Java Developers ดังภาพ



ภาพประกอบที่ ก-2 เลือกตัวเลือกการติดตั้งโปรแกรม

- 3) เลือกพื้นที่จัดเก็บตามต้องการดังตัวอย่างจัดเก็บไว้ที่ C:\Users\boonb\eclipse\java-2021-032 และเลือก java version 15.0.0 ขึ้นไป แล้วกดปุ่ม Install จากนั้นจะมีไอคอนโปรแกรมขึ้นที่หน้า Desktop ดังภาพประกอบที่ ก-4



ภาพประกอบที่ ก-3 ขั้นตอนการติดตั้งไฟล์

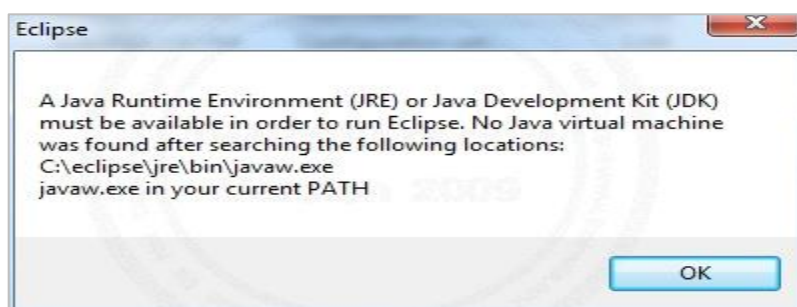


ภาพประกอบที่ ก-4 ไอคอนโปรแกรม Eclipse

กรณีที่หลังจากการติดตั้งโปรแกรม eclipse แล้วเกิด error

- 1) หากเกิด Error ดังภาพประกอบที่ ก-3 คือ A Java Runtime Environment (JRE) or Java Development Kit (JDK)... แสดงว่ายังไม่ได้ทำการติดตั้ง Java Development JDK โดยสามารถดาวน์โหลด JDK ได้ที่

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



ภาพประกอบที่ ก-5 แสดงข้อความ error ของโปรแกรม eclipse

- 2) หลังจากดาวน์โหลด JDK เรียบร้อยแล้ว ให้ทำการคลิกขวาเปิดไฟล์ JDK ที่ดาวน์โหลดมา เพื่อทำการติดตั้ง ดังภาพประกอบที่ ก-4



ภาพประกอบที่ ก-6 แสดงไฟล์ JDK.exe

- 3) คลิกเลือกที่ Next ดังภาพประกอบที่ ก-6



ภาพประกอบที่ ก-7 แสดงการติดตั้ง JDK ขั้นตอนที่ 1

- 4) คลิกเลือกที่ Next ดังภาพประกอบที่ ก-8



ภาพประกอบที่ ก-8 แสดงการติดตั้ง JDK ขั้นตอนที่ 2

- 5) รอให้แถบ Status เต็มดังภาพประกอบที่ ก-9



ภาพประกอบที่ ก-9 แสดงการติดตั้ง JDK ขั้นตอนที่ 3

- 6) คลิกเลือกที่ Next ดังภาพประกอบที่ ก-10



ภาพประกอบที่ ก-10 แสดงการติดตั้ง JDK ขั้นตอนที่ 4

- 7) กำลังติดตั้ง JDK ให้รอจนเสร็จสิ้น ดังภาพประกอบที่ ก-11



ภาพประกอบที่ ก-11 แสดงการติดตั้ง JDK ขั้นตอนที่ 5

- 8) การติดตั้งเสร็จสิ้นคลิกเลือก Continue ดังภาพประกอบที่ ก-12



ภาพประกอบที่ ก-12 แสดงการติดตั้ง JDK เสร็จสิ้นสมบูรณ์

2. ขั้นตอนการติดตั้งโปรแกรม Python

- ดาวน์โหลด Python จากเว็บไซต์ <https://www.python.org/downloads/>
- หลังจากดาวน์โหลด Python เรียบร้อยแล้ว ให้ทำการดับเบิลคลิกไฟล์ที่ได้ทำการดาวน์โหลดมาเพื่อทำการติดตั้ง ดังภาพ



ภาพประกอบที่ ก-13 ไฟล์ Python ที่ดาวน์โหลดมา

- คลิก Add Python 3.8 to PATH ดังภาพประกอบที่ ก-14



ภาพประกอบที่ ก-14 แสดงการติดตั้ง Python ขั้นที่ 1

- คลิก Install Now ดังภาพประกอบที่ ก-14
- คลิก Close ดังภาพประกอบที่ ก-15 เป็นการเสร็จสิ้นการติดตั้ง



ภาพประกอบที่ ก-15 แสดงการติดตั้ง Python ขั้นที่ 2

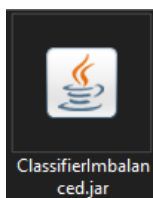
3. ขั้นตอนการติดตั้งโปรแกรม ClassifierImbalanced

- a. ทำการแยกไฟล์โดยคลิกขวาที่ชื่อไฟล์ (ClassifierImbalanced.zip) แล้วเลือกตำแหน่งที่จะเก็บไฟล์ตัวอย่างเช่น C:\Users\Window Name\Desktop ดังภาพประกอบที่ ก-16



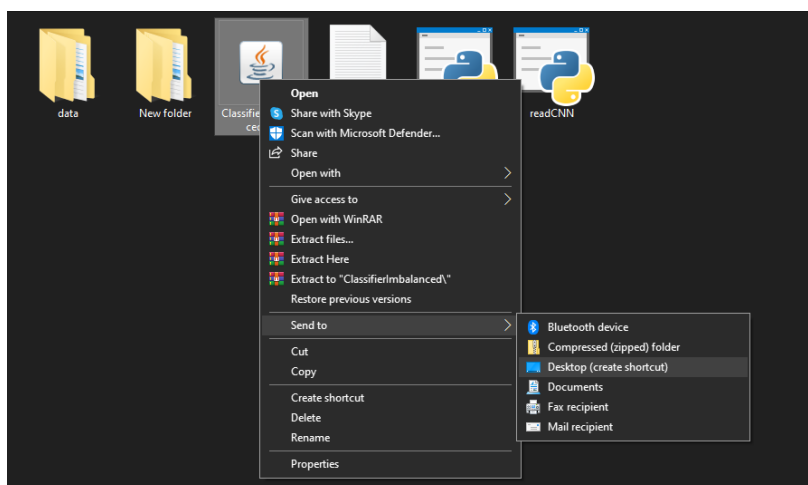
ภาพประกอบที่ ก-16 ทำการแตกไฟล์ ClassifierImbalanced

- b. แยกไฟล์ไว้ที่ C:\Users\Window Name\Desktop\ ClassifierImbalanced จากนั้นให้เปิดโปรแกรม ClassifierImbalanced โดยการดับเบิลคลิกที่ ClassifierImbalanced.jar ดังภาพประกอบที่ ก-17



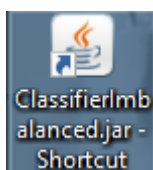
ภาพประกอบที่ ก-17 โปรแกรม ClassifierImbalanced

- c. สร้าง short cut แล้วนำไปเก็บไว้ที่ Desktop โดยคลิกขวาที่ ClassifierImbalanced.jar เลือกที่ Send to เลือกที่ Desktop (create shortcut) ดังภาพประกอบที่ ก-18



ภาพประกอบที่ ก-18 สร้าง shortcut

- d. เมื่อสร้าง shortcut ไว้บนหน้าเดสก์ท็อปเรียบร้อยแล้ว จะถือว่าการติดตั้งโปรแกรมเสร็จสมบูรณ์ โดยจะได้โปรแกรกดังภาพประกอบที่ ก-19



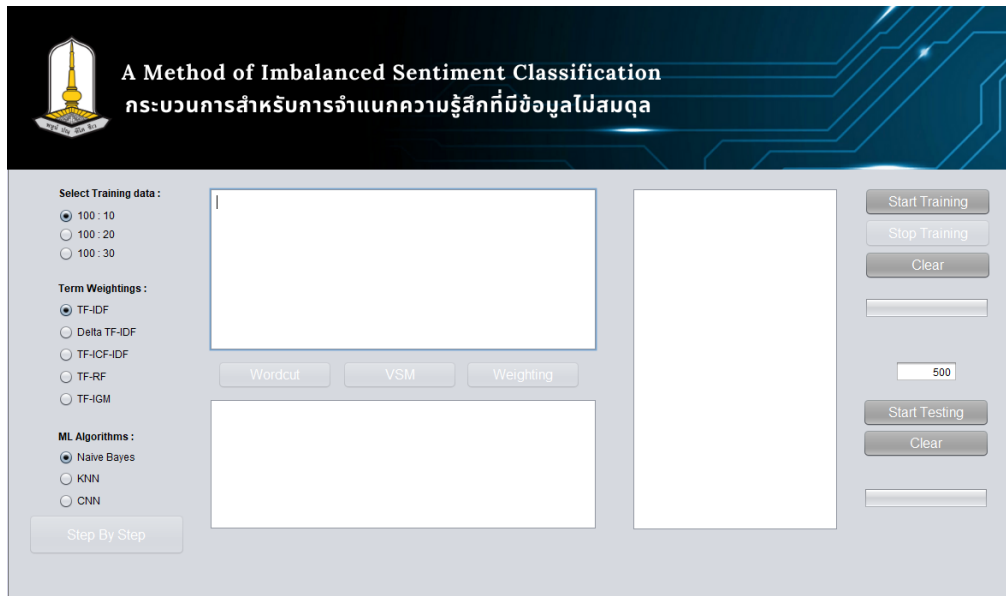
ภาพประกอบที่ ก-19 การติดตั้งโปรแกรมเสร็จสมบูรณ์

ภาคผนวก ข
คู่มือการใช้งาน

คู่มือการใช้งาน

การออกแบบระบบการจำแนกเอกสารที่มีความไม่สมดุลสำหรับบทวิจารณ์อิเล็กทรอนิกส์ มีเพียงการใช้งานโปรแกรมภายใต้ระบบปฏิบัติการ Windows เท่านั้น โดยจะแบบเป็น 2 ส่วนดังนี้

1. ส่วนของการสร้างโมเดลการจำแนกข้อมูลที่มีความไม่สมดุล



ภาพประกอบที่ ข-1 ตัวอย่างโปรแกรมหน้าการสร้างโมเดล

ในส่วนนี้จะเป็นการนำเอกสารที่เตรียมไว้มาใช้ในการสร้าง และทดสอบการจำแนกข้อมูลที่ไม่สมดุล โดยแบ่งตามอัลกอริทึมที่ใช้ในการสร้างโมเดลได้ 3 กลุ่ม และแต่ละกลุ่มสามารถสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุลได้ 5 รูปแบบ ดังนี้

1) สำหรับอัลกอริทึมนาอิวเบย์ (Naïve Bayes)

2)

แบบที่ 1 : โมเดลที่สร้างด้วยอัลกอริทึมนาอิวเบย์ และมีการให้น้ำหนักค่าแบบ *TF-IDF*

แบบที่ 2 : โมเดลที่สร้างด้วยอัลกอริทึมนาอิวเบย์ และมีการให้น้ำหนักค่าแบบ *Delta TF-IDF*

แบบที่ 3 : โมเดลที่สร้างด้วยอัลกอริทึมนาอิวเบย์ และมีการให้น้ำหนักค่าแบบ *TF-ICF-IDF*

แบบที่ 4 : โมเดลที่สร้างด้วยอัลกอริทึมนาอิวเบย์ และมีการให้น้ำหนักค่าแบบ *TF-RF*

แบบที่ 5 : โมเดลที่สร้างด้วยอัลกอริทึมนาอิวเบย์ และมีการให้น้ำหนักค่าแบบ *TF-IGM*

3) สำหรับอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุด (K-Nearest Neighbor : KNN)

แบบที่ 1 : โมเดลที่สร้างด้วยอัลกอริทึม *KNN* และมีการให้น้ำหนักค่าแบบ *TF-IDF*

แบบที่ 2 : โมเดลที่สร้างด้วยอัลกอริทึม *KNN* และมีการให้น้ำหนักค่าแบบ *Delta TF-IDF*

แบบที่ 3 : โมเดลที่สร้างด้วยอัลกอริทึม *KNN* และมีการให้น้ำหนักค่าแบบ *TF-ICF-IDF*

แบบที่ 4 : โมเดลที่สร้างด้วยอัลกอริทึม *KNN* และมีการให้น้ำหนักค่าแบบ *TF-RF*

แบบที่ 5 : โมเดลที่สร้างด้วยอัลกอริทึม *KNN* และมีการให้น้ำหนักค่าแบบ *TF-IGM*

4) สำหรับอัลกอริทึมโครงข่ายประสาทคอนโวลูชัน (Convolution Neural Network: CNN)

แบบที่ 1 : โมเดลที่สร้างด้วยอัลกอริทึม *CNN* และมีการให้น้ำหนักค่าแบบ *TF-IDF*

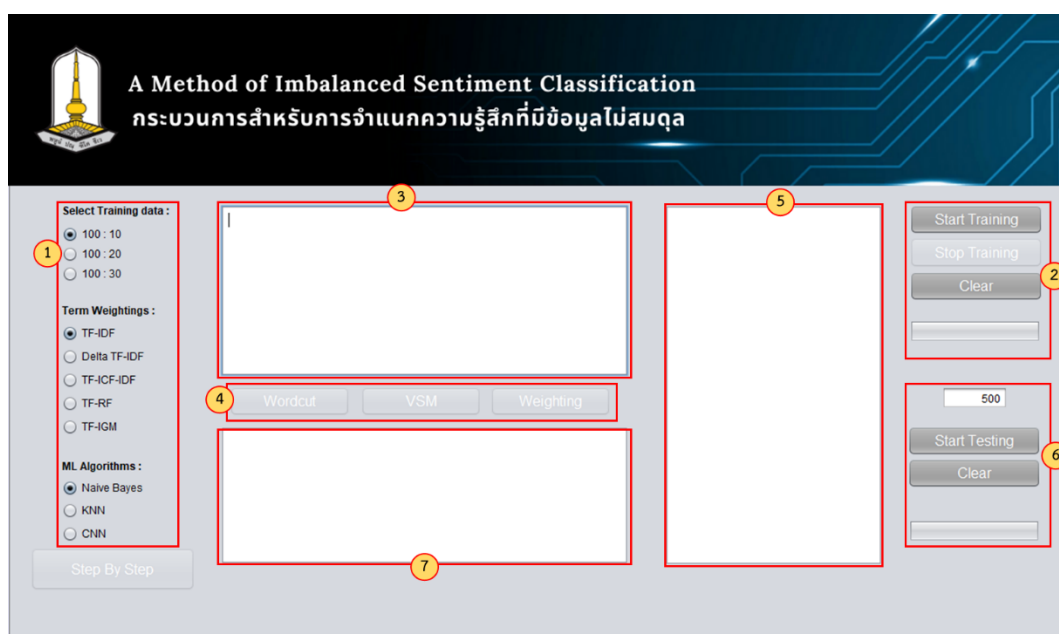
แบบที่ 2 : โมเดลที่สร้างด้วยอัลกอริทึม *CNN* และมีการให้น้ำหนักค่าแบบ *Delta TF-IDF*

แบบที่ 3 : โมเดลที่สร้างด้วยอัลกอริทึม *CNN* และมีการให้น้ำหนักค่าแบบ *TF-ICF-IDF*

แบบที่ 4 : โมเดลที่สร้างด้วยอัลกอริทึม *CNN* และมีการให้น้ำหนักค่าแบบ *TF-RF*

แบบที่ 5 : โมเดลที่สร้างด้วยอัลกอริทึม *CNN* และมีการให้น้ำหนักค่าแบบ *TF-IGM*

การทำงานของโปรแกรมในหน้าการสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุลทำได้ดังภาพประกอบที่ ข-2 ซึ่งมีขั้นตอนดังนี้



ภาพประกอบที่ ข-2 ตัวอย่างโปรแกรมหน้าการสร้างโมเดล

ส่วนที่ 1 : คือ ส่วนของการเลือกสัดส่วนเอกสาร อัลกอริทึม และการให้น้ำหนักค่า ที่จะใช้ในการสร้างและทดสอบโมเดล

ส่วนที่ 2 : คือ ส่วนของการสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุล ภายหลังจากทำส่วนที่ 1 ครบแล้ว

ส่วนที่ 3 : คือ ส่วนของการแสดงผลหลังจากการสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุล

ส่วนที่ 4 : คือ ส่วนของกระบวนการหลักๆ ในการสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุล

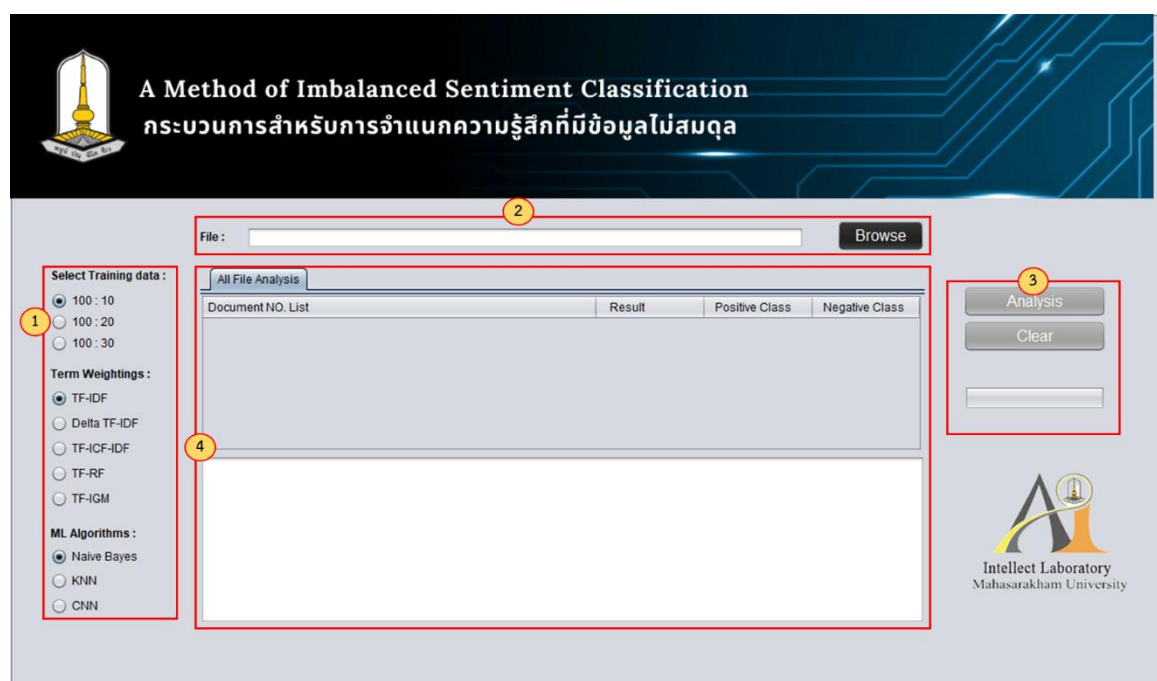
ส่วนที่ 5 : คือ ส่วนของการแสดงผลที่ได้หลังการทำแต่ละขั้นตอนในการสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุล

ส่วนที่ 6 : คือ ส่วนของการทดสอบโมเดลการจำแนกข้อมูลที่ไม่สมดุล ภายหลังจากที่ทำการสร้างโมเดลการจำแนกข้อมูลที่ไม่สมดุลจากส่วนที่ 2

ส่วนที่ 7 : คือ ส่วนของการแสดงผลการทดสอบโมเดลการจำแนกข้อมูลที่ไม่สมดุล

2. ส่วนของการนำโมเดลการจำแนกข้อมูลที่มีความไม่สมดุลไปใช้งาน

การทำงานของโปรแกรมในหน้าการนำโมเดลการจำแนกข้อมูลที่มีความไม่สมดุลไปใช้งาน ทำได้ดังภาพประกอบที่ ข-3 ซึ่งมีขั้นตอนดังนี้



ภาพประกอบที่ ข-3 ตัวอย่างโปรแกรมหน้าการนำโมเดลการจำแนกข้อมูลที่มีความไม่สมดุลไปใช้งาน

ส่วนที่ 1 : คือ ส่วนของการเลือกสัดส่วนเอกสาร อัลกอริทึม และการให้น้ำหนักคำ ที่จะใช้ในการสร้างและทดสอบโมเดล

ส่วนที่ 2 : คือ ส่วนของการโหลดเอกสารที่ต้องการจะใช้วิเคราะห์เข้ามา โดยเอกสารต้องอยู่ในรูปแบบ XML

ส่วนที่ 3 : คือส่วนที่จะทำการสั่งทำการทดสอบโมเดลที่ป้อน Analysis

ส่วนที่ 4 : คือ ส่วนของการแสดงผลการวิเคราะห์ระดับคะแนน จากเอกสารที่ทำการโหลดเข้ามาจากส่วนที่ 2 โดยจะแสดงผลได้จาก จำนวนเอกสารทั้งหมดที่โหลดเข้ามา

บทความวิจัย

กระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุล

A Method of Imbalanced Sentiment Classification

พีระวัฒน์ บุญบ้านจัว¹ และจันทิมา พลพินิจ²

Pheerawat Bunbanngio¹ and Jantima Polpinij²

บทคัดย่อ

การจำแนกความรู้สึก (Sentiment Classification) คือการจำแนกเอกสารตามชั้นความรู้สึกซึ่งโดยทั่วไปอาจจะจำแนกเป็นความรู้สึกที่เป็นบวก (Positive) ความรู้สึกที่เป็นลบ (Negative) และความรู้สึกที่เป็นกลาง (Neutral) โดยการจำแนกความรู้สึกนั้น ได้รับการศึกษามากมายต่อเนื่อง เพราะการประยุกต์ใช้ในหลายลักษณะ แต่โดยทั่วไปมักจะนิยมใช้ในการจำแนกความรู้สึกที่มีการแสดงไว้ในรูปแบบข้อความ (Text) เช่น ประยุกต์ใช้ในการจัดอันดับความรู้สึกจากข้อความแสดงความคิดเห็นของผู้คนที่ต่อสินค้าและบริการ การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้เรียน การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของคนในเรื่องการเมือง เป็นต้น ซึ่งปัญหาความไม่สมดุลของข้อมูลในคลาสนั้น เกิดจากกลุ่มตัวอย่างที่ใช้ในการเรียนรู้มีข้อมูลไม่สมดุลกัน โดยกลุ่มที่มีข้อมูลมากกว่าจะเรียกว่า “ข้อมูลกลุ่มหลัก (Majority Class)” ขณะที่กลุ่มตัวอย่างที่มีข้อมูลจำนวนน้อยกว่าจะเรียกว่า “ข้อมูลกลุ่มรอง (Minority Class)” เมื่อนำเอาชุดข้อมูลในลักษณะนี้ไปเรียนรู้เพื่อสร้างตัวจำแนกความรู้สึก (Sentiment Classifier) ข้อมูลใหม่ๆ ที่อ่านเข้ามาเพื่อวิเคราะห์เพื่อจำแนกกลุ่มด้วยตัวจำแนกความรู้สึกดังกล่าวก็มีแนวโน้มที่จะทำนายกลุ่มของข้อมูลนั้นไปยังทิศทางของข้อมูลกลุ่มหลักที่ใช้ในการเรียนรู้ตัวจำแนกความรู้สึก ดังนั้น ในโครงงานปริญญาโทฉบับนี้ จึงได้นำเสนอการศึกษาการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึกด้วยเทคนิคการให้น้ำหนักค่า 5 เทคนิค คือ TF-IDF, Delta TF-IDF, TF-IDF-ICF, TF-RF และ TF-IGM ร่วมกับแมชชีนเลิร์นนิง 3 ตัว คือ Naïve Bayes, K-Nearest Neighbor และสุดท้าย Convolution Neural Network

คำสำคัญ: การจำแนกเอกสาร, การให้น้ำหนักค่า, ข้อมูลไม่สมดุล, ซัพพอร์ตเวกเตอร์แมชชีน

บทนำ

การจำแนกความรู้สึก (Sentiment Classification) [1] คือการจำแนกเอกสารตามชั้นความรู้สึกซึ่งโดยทั่วไปอาจจะจำแนกเป็นความรู้สึกที่เป็นบวก (Positive) ความรู้สึกที่เป็นลบ (Negative) และความรู้สึกที่เป็นกลาง (Neutral) โดยการจำแนกความรู้สึกนั้น ได้รับการศึกษามาอย่างต่อเนื่อง เพราะการประยุกต์ใช้หลายลักษณะ แต่โดยทั่วไปมักจะนิยมใช้ในการจำแนกความรู้สึกที่มีการแสดงไว้ในรูปแบบข้อความ (Text) [1] เช่น ประยุกต์ใช้ในการจัดอันดับความรู้สึกจากข้อความแสดงความคิดเห็นของผู้คนที่ติดต่อสินค้าและบริการ การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้เรียน การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้คนในเรื่องการเมือง เป็นต้น

อย่างไรก็ตาม แม้ว่าการจำแนกความรู้สึก จะได้รับการศึกษาและความสนใจอย่างต่อเนื่อง แต่ยังมีปัญหาที่พบในการจำแนกความรู้สึกหลายประเด็น ประเด็นที่น่าสนใจและยังคงได้รับการศึกษาเพื่อการแก้ปัญหาอยู่คือ ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึก (Imbalanced Sentiment Classification) โดยทั่วไปที่พบบ่อยคือปัญหาความไม่สมดุลของข้อมูลในคลาส (Class Imbalance Data) [2-5]

ซึ่งปัญหาความไม่สมดุลของข้อมูลในคลาสนั้น เกิดจากกลุ่มตัวอย่างที่ใช้ในการเรียนรู้มีข้อมูลไม่สมดุลกัน โดยกลุ่มที่มีข้อมูลมากกว่าจะเรียกว่า “ข้อมูลกลุ่มหลัก (Majority Class)” ขณะที่กลุ่มตัวอย่างที่มีข้อมูลจำนวนน้อยกว่าจะเรียกว่า “ข้อมูลกลุ่มรอง (Minority Class)” เมื่อนำเอาชุดข้อมูลในลักษณะนี้ไปเรียนรู้เพื่อสร้างตัวจำแนกความรู้สึก (Sentiment Classifier) ข้อมูลใหม่ๆ ที่อ่านเข้ามาเพื่อวิเคราะห์เพื่อจำแนกกลุ่มด้วยตัวจำแนกความรู้สึกดังกล่าว ก็มีแนวโน้มที่จะทำนายกลุ่มของข้อมูลนั้นไปยังทิศทางของข้อมูลกลุ่มหลักที่ใช้ในการเรียนรู้ตัวจำแนกความรู้สึก

เทคนิคหลายๆ เทคนิคได้ถูกนำเสนอเพื่อใช้ในการควบคุมปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึก เช่น Resampling Methods [4] สำหรับวิธีการนี้จะเป็นการประยุกต์เอาวิธีสุ่มตัวอย่างซึ่งเป็นวิธีการทางสถิติ เพื่อสร้างข้อมูลสำหรับการสอน โดยมีจุดประสงค์เพื่อให้จำนวนสมาชิกในข้อมูลทั้งสองกลุ่มมีความสมดุลกัน ซึ่งประกอบด้วย 2 วิธีการใหญ่ๆ คือ Oversampling [6] และ Undersampling [6] โดยวิธีการทำแบบ Oversampling จะทำการสุ่มข้อมูลในกลุ่มรองเพื่อสร้างข้อมูลใหม่ของกลุ่มรองให้มีจำนวนเพิ่มมากขึ้น ให้ใกล้เคียงหรือเท่ากับจำนวนข้อมูลในกลุ่มหลัก และในทางตรงข้ามวิธีการ Undersampling จะทำการสุ่มเลือกข้อมูลสำหรับการสอนจากข้อมูลในกลุ่มหลัก ให้ได้จำนวนที่ใกล้เคียงกับจำนวนข้อมูลในกลุ่มรอง โดยทั่วไปมักประยุกต์วิธีการแบบ Undersampling แต่ก็จะเกิดปัญหาข้อมูลไม่เพียงพอต่อการเรียนรู้

อย่างไรก็ตาม เมื่อไม่นานมานี้ หลายงานวิจัยที่นำเสนอเทคนิคการให้น้ำหนักคำ (Term Weighting) เข้ามาช่วยในการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึก [8], [9] และพบว่าเทคนิคการให้น้ำหนักคำแบบมีผู้สอน (Supervised Term Weighting: STW) มีแนวโน้มที่จะทำให้เกิดประสิทธิภาพในการจำแนกความรู้สึกที่ดีขึ้น

ดังนั้นในโครงงานปริญญาณิพนธ์ฉบับนี้ จึงได้นำเสนอการศึกษาการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึกด้วยเทคนิคการให้น้ำหนักคำแบบมีผู้สอนอย่างน้อย 3 เทคนิค พร้อมทั้งทำการเปรียบเทียบการเทคนิคการให้น้ำหนักคำแบบไม่มีผู้สอน (Unsupervised Term Weighting) ที่นิยมใช้ในการจำแนกเอกสารความรู้สึกนั้นคือ *tf-idf* (Term Frequency-Inverse Document Frequency) (Salton, Wong, & Yang, 1975) ภายใต้วัดตัวจำแนกความรู้สึกอย่างน้อย 3 ตัว

บททวนวรรณกรรม

1. การประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP)

การประมวลผลภาษาธรรมชาติ [4, 5] คือ สาขาย่อยของปัญญาประดิษฐ์และภาษาศาสตร์ที่ศึกษาปัญหาในการประมวลผลและใช้งานภาษาธรรมชาติ รวมทั้งการทำความเข้าใจภาษาธรรมชาติ

ทั้งนี้เพื่อให้คอมพิวเตอร์สามารถเข้าใจภาษามนุษย์ได้ โดยแบ่งเป็นภาษาพูดและภาษาเขียน ซึ่งในที่นี้จะกล่าวถึงภาษาเขียนเท่านั้น

ระดับของการประมวลผลภาษาธรรมชาติ มีทั้งหมด 5 ระดับ คือ

1) Morphological Analysis เป็นการวิเคราะห์หน่วยคำที่สามารถแยกย่อยได้เป็นอะไรบ้าง และคำๆ นั้นมีหน้าที่อะไร เช่น “friendly” แยกได้เป็น “friend” และ “ly” เป็นต้น

2) Syntactic Analysis เป็นการวิเคราะห์ทางไวยากรณ์ เพื่อให้รู้ว่าประโยคหนึ่งๆ มีโครงสร้างเชิงวากยสัมพันธ์อย่างไร

3) Semantic Analysis จะเป็นการวิเคราะห์ความหมายของประโยคนั้นๆ

4) Discourse Integration เป็นการพิจารณาความหมายของประโยค โดยพิจารณาจากประโยคข้างเคียง เนื่องจากบางคำจะเข้าใจความหมายได้ต้องดูความหมายของประโยคก่อนหน้า

5) Pragmatic Analysis คือการแปลความหมายของประโยค เพื่อดูความตั้งใจในการสื่อสารของผู้สื่อสารว่าจุดประสงค์กล่าวถึงอะไร

2. การวิเคราะห์ความรู้สึก (Sentiment Analysis)

การวิเคราะห์ความรู้สึก [6-8] คืองานวิจัยที่อยู่ในกลุ่มของการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) ที่ มีกระบวนการมุ่งเน้นการวิเคราะห์และตรวจสอบความรู้สึก (Opinion) ของผู้คนจากข้อความ (Text)

ที่คนเหล่านั้นเขียนหรือโพสต์เอาไว้ เพื่อบ่งบอกความรู้สึกของตนเองที่มีต่อบางสิ่งบางอย่างที่ตนเองสนใจ เช่น ความรู้สึกดี (Positive หรือ Good) หรือความรู้สึกที่ไม่ดีหรือไม่ชอบ (Negative หรือ Bad) เช่น เมื่อลูกค้าซื้อคอมพิวเตอร์ไป 1 เครื่อง ลูกค้าอาจจะให้คะแนนเฉลี่ยเกี่ยวกับคอมพิวเตอร์รุ่นนั้นๆ ไว้ที่ 3 จากคะแนนเต็ม 5 แต่หัวข้อต่างๆ ที่สอบถามไปยังลูกค้าอาจยังไม่ครอบคลุมในทุกๆ กรณี ที่เป็นความต้องการหรือความคาดหวังต่อสินค้าและบริการของลูกค้า ทำให้ลูกค้าอาจจะไปเขียนแสดงความรู้สึกเกี่ยวกับคอมพิวเตอร์รุ่นนั้นๆ ไว้ใน Blog, Twitter หรือ Facebook ของตนเอง [10, 37]

การวิเคราะห์ความรู้สึกสามารถแบ่งได้ 3 ระดับ ดังนี้ [5]

1) การวิเคราะห์ความรู้สึกระดับเอกสาร (Document Level Analysis) เป็นการวิเคราะห์ข้อความแสดงความคิดเห็นในแบบหยาบ เนื่องจากเป็นการนำข้อความแสดงความคิดเห็นทั้งหมดจากเอกสารมาสรุป แยกข้อความความคิดเห็นเป็นขั้วบวก ขั้วลบ หรือเป็นกลาง

2) การวิเคราะห์ความรู้สึกระดับประโยค (Sentence Level Analysis) เป็นการวิเคราะห์ข้อความแสดงความคิดเห็น โดยแยกข้อความที่เป็นข้อความแสดงความคิดเห็น ออกมาจากข้อความที่เป็นข้อเท็จจริงในระดับที่เป็นประโยค แล้วนำมาแยกข้อความความคิดเห็นเป็นขั้วบวก ขั้วลบ หรือเป็นกลาง

3) การวิเคราะห์ความรู้สึกระดับคุณลักษณะ (Feature Level Analysis) เป็นการวิเคราะห์ข้อความแสดงความคิดเห็น โดยแยกคุณลักษณะที่สนใจหรือหัวข้อที่ถูกแสดงความคิดเห็นออกมาก่อน แล้วจึงนำมาแบ่งข้อความความคิดเห็นเป็นขั้วบวก ขั้วลบ หรือเป็นกลาง และนำมาจัดกลุ่มเข้ากับคำที่มีความหมายเหมือนกันในแต่ละคุณลักษณะ ซึ่งระบบจะวิเคราะห์ข้อความแสดงความคิดเห็นในระดับคุณลักษณะ แล้วนำผลลัพธ์ที่

ได้มาแสดงให้อยู่ในรูปแบบที่ผู้ใช้งานสามารถเข้าใจได้ง่ายขึ้น

3. การจำแนกหมวดหมู่เอกสาร (Text Classification)

การจำแนกหมวดหมู่เอกสาร [13, 25] เป็นการนำวิธีการเรียนรู้ด้วยคอมพิวเตอร์ (Machine Learning) ประยุกต์รวมกับการประมวลผลภาษาธรรมชาติ การจัดแบ่งกลุ่มเอกสารแบบอัตโนมัติเป็นการแบ่งกลุ่มตามเนื้อหาของเอกสาร โดยที่มีการกำหนดกลุ่มหรือหมวดหมู่ของเอกสารไว้ก่อนหน้า เป็นลักษณะการวิเคราะห์เอกสารที่เข้ามากับเอกสารในแต่ละหมวดหมู่ เพื่อดูว่าเอกสารนั้นๆ ให้มีลักษณะคล้ายกับหมวดหมู่ใดมากที่สุด

โดยสามารถให้นิยามการจำแนกหมวดหมู่เอกสาร ดังนี้ กำหนดให้คู่ลำดับ $(d_j, c_i) \in D \times C$ โดยที่ D เป็นโดเมนของเอกสาร ขณะที่ C เป็นกลุ่มเอกสารที่เป็นไปได้ $\{c_1, c_2, \dots, c|C|\}$ และกำหนดให้ T เป็นคู่ลำดับ (d_j, c_i) ที่จะบ่งบอกว่าเอกสาร d_j อยู่ภายใต้กลุ่มหรือหมวดหมู่ c_i โดยให้ F เป็นฟังก์ชันที่กำหนดให้กับคู่ลำดับ (d_j, c_i) เพื่อบอกว่าเอกสาร d_j ควรอยู่ภายใต้กลุ่มหรือหมวดหมู่ c_i หรือไม่ ดังนั้นการประมาณค่าของฟังก์ชันเป้าหมายสามารถแสดงได้คือ $F: D \times C \rightarrow \{T, F\}$ ซึ่งเป็นฟังก์ชันเป้าหมายที่จะแทนตัวจัดกลุ่มเอกสาร หรือ *Classifier*

4. การให้น้ำหนักคำ (Term Weighting)

การให้น้ำหนักคำ [17] ถือว่าเป็นส่วนหนึ่งของการจัดการเอกสาร โดยรูปแบบการให้น้ำหนักสามารถแบ่งออกเป็นสองประเภทหลักตามการใช้งานข้อมูลชั้นเรียนในเอกสารการฝึกอบรม ดังนี้ รูปแบบแรกคือ Unsupervised Term Weighting (UTW) [18] คือรูปแบบการให้น้ำหนักคำที่ซึ่งไม่ใช้ข้อมูลชั้นเรียนเพื่อสร้างน้ำหนัก รูปแบบที่ได้รับความนิยมมากที่สุดคือ TF-IDF (Term Frequency - Inverse Document, Frequency) ซึ่งถูกใช้อย่างมีประสิทธิภาพในการศึกษาการดึงข้อมูล แต่อย่างไร

ก็ตามมันไม่เหมาะสำหรับงานการจัดหมวดหมู่ข้อความ เนื่องจากการให้น้ำหนักค่าแบบ Unsupervised Term Weighting เป็นการให้น้ำหนักคำกับเอกสารทั้งหมดโดยไม่แบ่งหมวดหมู่เอกสาร โดยหากใช้รูปแบบนี้จะทำให้ประสิทธิภาพในการจำแนกหมวดหมู่ข้อความลดลง

ส่วนรูปแบบที่สองเป็นรูปแบบที่นักวิจัยใช้ในผลงานนี้ คือ Supervised Term Weighting (STW) [11] ซึ่งได้รับการเสนอครั้งแรกโดย Debolc และ Sebastiani [11] โครงสร้าง Supervised Term Weighting ใช้ชุดข้อมูลการฝึกอบรมของข้อมูลระดับชั้นเรียนเพื่อคำนวณน้ำหนักของคำศัพท์ โดยการให้น้ำหนักในแบบนี้จะใช้ประโยชน์จากข้อมูลระดับที่รู้จักในคลังข้อมูลการฝึกอบรม โดยจะทำให้การให้น้ำหนักมีประสิทธิภาพที่ดียิ่งขึ้น สำหรับการจำแนกหมวดหมู่ความรู้สึกของข้อความ การวิเคราะห์ความรู้สึก การจำแนกความไม่สมดุลของชุดเอกสาร และอื่นๆ

5. งานวิจัยที่เกี่ยวข้อง

ในการจำแนกความรู้สึกของเอกสารข้อความก็พบปัญหาของข้อมูลที่ไม่สมดุล ซึ่ง Li และคณะ [2] ได้ศึกษาเกี่ยวกับข้อมูลที่ไม่สมดุลหลายรูปแบบ เช่น จำนวนเอกสารที่ไม่สมดุล ขนาดของคลาสที่ไม่สมดุล รวมถึงความไม่สมดุลในคลาสน้อย จากการศึกษาที่ต่อเนื่องพบว่า ประเด็นที่หนึ่งจำนวนเอกสารข้อความในสองคลาสจะเท่ากัน ความแตกต่างของจำนวนคำในเอกสารกลายเป็นปัจจัยสำคัญที่มีผลต่อความถูกต้องของการจำแนกเอกสาร ประเด็นที่สอง เพื่อปรับปรุงความถูกต้องของการจำแนกเอกสารด้วยการเพิ่มจำนวนของกลุ่มข้อมูลที่มีจำนวนน้อย และประเด็นที่สาม ในกรณีของข้อมูลที่ไม่สมดุล คำเดียวกันที่ปรากฏในสองคลาสมักจะเป็นสารสนเทศสำคัญของคลาส นั่นคือ คลาสที่บั่นทอนกันจะไม่ส่งผลกระทบต่อความถูกต้องของการจัดประเภท

Flavio Carvalho และ Gustavo Pai Guedes ได้นำเสนอการให้น้ำหนักค่าแบบ Supervised Term Weighting ที่เหมาะสมต่อการจำแนกความรู้สึกที่ไม่สมดุล โดยได้นำเสนอการให้น้ำหนักค่าที่ได้รับการควบคุมดูแลเจ็ดชุดและแผนการกำหนดน้ำหนัก ซึ่งวิธีนี้เป็นวิธีที่มีประสิทธิภาพมากกว่าการให้น้ำหนักค่าในแบบ Unsupervised Term Weighting เนื่องจากการให้น้ำหนักค่าในรูปแบบนี้เป็นใช้ประโยชน์จากข้อมูลที่อยู่ในคลังข้อมูลการฝึกอบรม

ในปี ค.ศ. 2011 Shoushan Li และคณะได้ทำงานวิจัย Imbalance Sentiment Classification [23] เพราะเล็งเห็นปัญหาในการจำแนกความรู้สึกที่ไม่สมดุลของข้อมูล เนื่องจากวิธีก่อนหน้านี้มีปัญหาในการทำงานค่อนข้างมาก จึงได้นำเสนอ วิธีการจำแนกความรู้สึกที่ไม่สมดุล โดยเสนอโครงสร้างการจัดกลุ่มแบบ under-sampling ด้วยการแบ่งเป็นกลุ่มเพื่อเอาชนะปัญหาการกระจายระดับความไม่สมดุลในการจำแนกความรู้สึกที่ไม่สมดุล ภายใต้กรอบงานนี้ กลุ่มตัวอย่างในกลุ่มเสียงส่วนใหญ่จะถูกจัดกลุ่มเป็นกลุ่มแรก จากนั้นเลือกกลุ่มตัวอย่างจำนวนที่เหมาะสมจากแต่ละกลุ่มจากตัวอย่างการฝึกอบรมของข้อมูลส่วนใหญ่

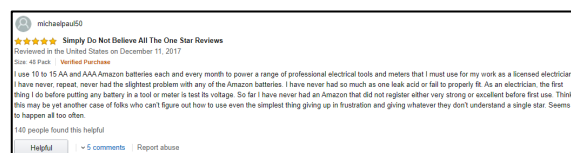
ในงานวิจัยของ Ah-Pine และ Pavel Soriano Morales [6] ศึกษาแก้ปัญหาความไม่สมดุลของข้อมูลในการวิเคราะห์ความรู้สึก (Sentiment Classification) ที่ใช้ข้อมูลจาก twitter ที่พบว่าการกระจายกลุ่มของข้อมูลมีความเอนเอียงไปกลุ่มใดกลุ่มหนึ่ง นั่นคือจำนวนข้อมูลในแต่ละกลุ่มขาดความสมดุล ดังนั้นนักวิจัยจึงนำเสนอการทำเทคนิคการสุ่มตัวอย่างแบบสังเคราะห์ (Synthetic Oversampling Techniques) สำหรับการจำแนกกลุ่มข้อความ Twitter

กระบวนการวิจัย

ในส่วนนี้จะอธิบายขั้นตอนการดำเนินงานที่นำเสนอในงานวิจัยฉบับนี้ โดยรายละเอียดสามารถแสดงได้ดังนี้

1. การรวบรวมข้อมูล

ในงานวิจัยนี้ ได้ใช้ชุดข้อความแสดงความคิดเห็นที่เกี่ยวกับอุปกรณ์อิเล็กทรอนิกส์ ซึ่งรวบรวมมาจากเว็บไซต์ Amazon โดยจะมีการแบ่งเอกสารออกเป็น 2 ชุด คือ ชุดข้อมูลสอน (Training set) และ ชุดข้อมูลทดสอบ (Test set) ซึ่งเอกสารจะอยู่ในรูปแบบ XML ข้อมูลที่ใช้ทั้งหมด 50,000 ความคิดเห็นและมีค่าระหว่าง 30 ถึง 300 คำต่อหนึ่งเอกสารข้อความแสดงความคิดเห็น



ภาพที่ 1 ตัวอย่างข้อความแสดงความคิดเห็น

จากภาพที่ 1 เป็นตัวอย่างเอกสารข้อความแสดงความคิดเห็นจากเว็บ Amazon ที่ใช้ในงานวิจัยนี้ โดยจะทำการดาวน์โหลดออกมาในรูปแบบ XML ซึ่งจะประกอบไปด้วย รหัส (ID), สถานะ (Status) และเนื้อหาของเอกสาร (details) ดังภาพที่ 2



ภาพที่ 2 ตัวอย่างเอกสารที่จัดเก็บรูปแบบ XML

2. การสร้างโมเดลเพื่อจำแนกความรู้สึกของบทวิจารณ์ (Classifier Modeling)

ส่วนที่ 1: การเตรียมข้อมูล (Text Pre-processing)

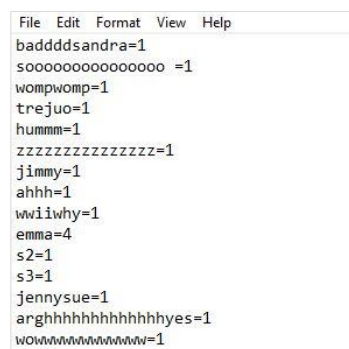
ขั้นตอนนี้เป็นการเตรียมข้อมูลเพื่อให้เหมาะสมต่อการนำไปสร้างโมเดลการจำแนกระดับคะแนนบทวิจารณ์ภาพยนตร์ โดยจะมีขั้นตอนดังนี้

ขั้นที่ 1: การตัดคำ (Tokenization) [4] การตัดคำ คือกระบวนการที่แยกข้อความออกเป็น “คำ” เนื่องจากคำเป็นหน่วยที่เล็กที่สุดในภาษาที่สื่อความหมายได้ สำหรับภาษาอังกฤษจะใช้ช่องว่าง (space) ที่คั่นระหว่างคำในการตัดคำ และจะใช้จุด “.” เพื่อบอกการจบประโยค

ขั้นที่ 2: การตัดคำหยุด (Stop-word Removal) [4] การตัดคำหยุด คือกระบวนการตัดคำหรือสัญลักษณ์ที่พบบ่อยมากในเอกสาร แต่คำหรือสัญลักษณ์เหล่านั้นไม่ได้ส่งผลต่อการจัดกลุ่มเอกสาร ดังนั้นเมื่อทำการตัดออกแล้วไม่ทำให้ใจความในเอกสารนั้นๆ เปลี่ยนไป การตัดคำหยุดมีความจำเป็นอย่างมากในการจัดกลุ่มเอกสารแบบอัตโนมัติ เพราะจะช่วยลดระยะเวลาในการประมวลผลลงได้เป็นอย่างมาก เนื่องจากระบบจะไม่เสียเวลาในการประมวลผลคำเหล่านี้ ตัวอย่างเช่น a, an, the หรือกลุ่มคำจำพวก Article

ขั้นที่ 3: การคัดเลือกคำด้วยพจนานุกรม (English-Dictionary) เนื่องจากข้อมูลที่ใช้ในการสร้างโมเดลการจำแนกระดับคะแนนบทวิจารณ์ภาพยนตร์ เป็นข้อความที่ผู้คนเข้ามาเขียนแสดงความรู้สึกต่อภาพยนตร์เรื่องนั้นๆ ทำให้เกิดการใช้ภาษาที่ผิดเพี้ยนไปจากปกติ ในงานวิจัยฉบับนี้จึงได้มีการนำพจนานุกรมมาใช้ในการคัดเลือกคำ

ขั้นที่ 4: การเลือกคุณลักษณะ (Feature Selection) ภายหลังจากขั้นตอนข้างต้นแล้ว คลังคำที่ได้จะถูกนำเข้าสู่กระบวนการคัดเลือกคุณลักษณะด้วย Information Gain [18, 19, 35] สำหรับวิธีการคัดเลือกคุณลักษณะจะเป็นวิธีเบื้องต้นในการลดขนาดเอกสาร เนื่องจากจำนวนคุณลักษณะมีผลต่อประสิทธิภาพของการจำแนกหมวดหมู่เอกสาร เพราะอัลกอริทึมที่ใช้ในการเรียนรู้เพื่อสร้างตัวจำแนกหมวดหมู่เอกสารโดยทั่วไปไม่สามารถรองรับการทำงานกับจำนวนคุณลักษณะของเอกสารที่สูงมากได้ดี



ภาพที่ 3 ตัวอย่างการใช้ภาษาที่ผิดปกติ

ดังนั้น การลดขนาดของเอกสารจึงเป็นขั้นตอนหนึ่งที่จะต้องกระทำก่อน และในงานวิจัยฉบับนี้จะใช้ค่าเกนสารสนเทศ (IG: Information Gain) เป็นตัววัดคุณลักษณะของเอกสาร ซึ่งค่า IG จะคำนวณจากจำนวนบิตที่ได้รับสำหรับการทำนายกลุ่ม โดยการดูจากการมีอยู่หรือไม่มีอยู่ของคำในเอกสาร ให้ C_1, \dots, C_K แทนเซตที่เป็นไปได้ของกลุ่ม คำ IG ของคำ w นิยามโดย

$$IG(w) = - \sum P(c_j) \log P(c_j) + P(w) \sum P(c_j|w) \log P(c_j|w) + P(w) \sum P(c_j|w) \log P(c_j|w) \quad (1)$$

โดยค่า $P(C_j)$ คือความน่าจะเป็นของกลุ่มแต่ละกลุ่มที่พบ (class) ค่า $P(w)$ คือความน่าจะเป็นของคำแต่ละคำ (word) ที่พบ และค่า $P(C_j|w)$ คือความน่าจะเป็นของกลุ่มที่ได้จากคำ

เมื่อทำการคำนวณค่า IG ของแต่ละคุณลักษณะที่ได้ จากนั้นจะทำการตัดคุณลักษณะที่มีค่า IG เท่ากับ 0 ทั้งหมด เพราะแสดงว่าค่าๆ นั้นไม่มีความสำคัญต่อการจัดกลุ่มเอกสาร อีกทั้งยังช่วยลดระยะเวลาที่ระบบใช้ในการประมวลผล

ขั้นที่ 5: การสร้างตัวแทนเอกสารและการให้น้ำหนักคำ (Document Representation and Term Weighting)

ในขั้นตอนนี้จะเป็นการนำเสนอเอกสารในรูปแบบ Vector Space Model [17] เพื่อแสดงให้เห็นถึงความสัมพันธ์ระหว่างเอกสารและคำที่

ปรากฏในเอกสาร พร้อมทั้งการให้น้ำหนักของคำ เพื่อแสดงว่าคำๆ นั้นมีความสำคัญกับเอกสารมากน้อยเพียงใด ซึ่งถ้าหากคำน้ำหนักของคำใดมีค่ามาก ก็แสดงว่ามีความสำคัญและสามารถบ่งชี้ถึงเอกสารสูง โดยการให้น้ำหนักคำจะมีอยู่ 5 รูปแบบคือ

รูปแบบที่ 1: การให้น้ำหนักคำแบบ *tf-idf* [18]

เมื่อ *tf* เป็น local weight ที่ เป็นความถี่ของ

$$ICF(t_i) = (1 + \log(\frac{M}{CF(t_i)})) \quad (2.2)$$

คำหนึ่งๆ ที่พบในแต่ละเอกสาร และ *idf* ก็คือ global weight ที่เป็นการหาส่วนกลับของความถี่ของคำในเอกสาร หรือที่เรียกว่าระบบน้ำหนักความถี่เอกสารผกผัน

$$idf = \log(N/df) \quad (1)$$

โดยที่ *N* คือจำนวนเอกสารทั้งหมดในคลัง และ *df* คือจำนวนเอกสารที่มีคำนั้นๆ ปรากฏอยู่

$$tf - idf = tf \times idf \quad (2)$$

รูปแบบที่ 2 :การให้น้ำหนักระยะยาวตาม *Delta TF-IDF*

Delta TF-IDF ถูกเสนอโดย Martineau และ Finin [19] มันคำนวณความแตกต่างของคะแนน *TF-IDF* ในคลาสที่เป็นบวกและลบเพื่อปรับปรุงความแม่นยำ ในฐานะที่เป็น STW จะพิจารณาการกระจายของคุณสมบัติระหว่างสองคลาสก่อนการจำแนกประเภทการรับรู้และการเพิ่มความสูงของผลค่าที่แตกต่างกัน *Delta TF-IDF* ช่วยเพิ่มความสำคัญของคำที่กระจายอย่างไม่สม่ำเสมอระหว่างคลาสบวกและคลาสลบ โดยที่ N_p และ N_n คือจำนวนของเอกสารในคลาสบวกและลบตามลำดับ ส่วน *A* และ *C* แสดงความถี่เอกสารของคำว่า t_i ในคลาสบวกและลบตามลำดับ ดังสมการที่ 3

$$w_{\&TF-IDF}(t_i) = TF(t_i, d_j) \times \log_2(\frac{N_p \times C + 1.5}{A \times N_n + 1.5}) \quad (3)$$

รูปแบบที่ 3: การให้น้ำหนักระยะยาวตาม *TF-IDF-ICF*

TF-IDF-ICF เป็นรูปแบบการควบคุมน้ำหนักตามแบบ *TF-IDF* แบบดั้งเดิม อย่างไรก็ตามมันเพิ่มปัจจัยความถี่ผกผันในคลาส (Inverse Class Frequency : *ICF*) [8] เพื่อให้คำน้ำหนักที่สูงขึ้นไปยังคำที่หายากที่เกิดขึ้นน้อยกว่าในเอกสาร (*IDF*) และ Class (*ICF*) และใน (2.2) *M* คือจำนวนคลาสในคอลเลกชันและ $CF(t_i)$ สอดคล้องกับความถี่ของคลาสที่คำ t_i ปรากฏในคอลเลกชัน *TF-IDF-ICF* แสดงใน (4)

$$w_{TF-ICF}(t_i) = TF(t_i, d_j) \times IDF(t_i) \times ICF(t_i) \quad (4)$$

รูปแบบที่ 4: ระย่น้ำหนักตาม *TF-RF*

TF-RF (Term Frequency - Relevance Frequency) [18] ได้รับการเสนอ เช่นเดียวกับ *Delta TF-IDF*, *TF-RF* คำนึงถึงการกระจายคำศัพท์ในชั้นเรียนทั้งบวกและลบ อย่างไรก็ตามมีการพิจารณาเฉพาะเอกสารที่มีค่าดังกล่าว นั่นคือ ความเกี่ยวข้องของความถี่ (*RF*) ของข้อกำหนด *TF-RF* ถูกระบุใน (2.3) โดยที่ตัวหารน้อยที่สุดคือ 1 เพื่อหลีกเลี่ยงการหารด้วยศูนย์

$$w_{TF-RF}(t_i) = TF(t_i, d_j) \times \log_2(2 + \frac{A}{\max(1, C)}) \quad (5)$$

รูปแบบที่ 5: ระย่น้ำหนักตาม *TF-IGM*

ระยะความถี่-ช่วงเวลาแรงโน้มถ่วงผกผัน (Term Frequency - Inverse Gravity Moment : *TF-IGM*) [20] ถูกนำเสนอให้วัดความไม่สม่ำเสมอหรือความเข้มข้นของการแจกแจงคำศัพท์ระหว่างคลาสซึ่งสะท้อนให้เห็นถึงอำนาจการจำแนกชั้นข้อตกลง

สมการ *IGM* มาตรฐานกำหนดอันดับ (r) ตามความเข้มข้นของการแจกแจงระหว่างคลาสของคำซึ่งคล้ายกับแนวคิดของ “แรงโน้มถ่วง

โมเมนต์ (Gravity Moment : GM)” จากฟิสิกส์ IGM ถูกระบุใน (6) โดยที่ f_{ir} ($r = 1, 2, \dots, M$) ระบุจำนวนเอกสารที่มีคำว่า t_i ในคลาส r -th ซึ่งส่วนโค้งเรียงตามลำดับจากมากไปน้อย ดังนั้น f_{il} จึงแสดงถึงความถี่ของ t_i ในคลาสที่ปรากฏบ่อยที่สุด

$$IGM(t_i) = \left(\frac{f_{i1}}{\sum_{r=1}^M f_{ir} \times r} \right) \quad (6)$$

โดยนำหนักเทอม TF - IGM นั้นกำหนดตาม $IGM(t_i)$ ดังที่แสดงใน (7) ค่า λ คือสัมประสิทธิ์แบบปรับได้ที่ใช้เพื่อรักษาสมดุลสัมพัทธ์ระหว่างปัจจัยทั่วโลก และท้องถิ่นในน้ำหนักของคำ สัมประสิทธิ์ λ มีค่าเริ่มต้นที่ 7.0 และสามารถตั้งเป็นค่าระหว่าง 5.0 ถึง 9.0 [20] สมการ 8 นำเสนอ $SQRT$ - TF - IGM ซึ่งคำนวณสแควร์รูทของ TF ซึ่งเป็นเทคนิคในการปรับน้ำหนักในระยะที่สมเหตุสมผลมากขึ้นโดยลดผลกระทบของ TF สูง [9]

$$w_{TF,IGM}(t_i) = TF(t_i, d_j) \times (1 \times \lambda \times IGM(t_i)) \quad (7)$$

$$w_{SQRT,TF-IGM}(t_i) = \sqrt{TF(t_i, d_j)} \times (1 \times \lambda \times IGM(t_i)) \quad (8)$$

ส่วนที่ 2: การสร้างโมเดลเพื่อการจำแนกระดับคะแนนของบทวิจารณ์ภาพยนตร์

ขั้นตอนนี้เป็นขั้นตอนของการสร้างโมเดลเพื่อการจำแนกระดับคะแนนของบทวิจารณ์ด้วยอัลกอริทึมแบบมีผู้สอน (Supervised Learning) ดังนี้

1. อัลกอริทึมนาอ์ฟเบย์ (Naïve Bayes)

นาอ์ฟเบย์ (Naïve Bayes) [10, 13] เป็นการเรียนรู้รู้อย่างง่าย เป็นวิธีการจำแนกประเภทของข้อมูลที่มีประสิทธิภาพวิธีหนึ่ง และเหมาะกับการนำมาใช้กับกรณีที่มีเซตตัวอย่างเป็นจำนวนมาก และแต่ละคุณสมบัติ (Attribute) ของตัวอย่างเป็น

อิสระต่อกัน โดยนำการจำแนกประเภทนาอ์ฟเบย์มาประยุกต์ใช้ในการจำแนกประเภทของเอกสาร (Document Classification) พบว่ายังสามารถใช้งานได้ดีไม่ต่างจากการจำแนกวิธีการอื่นๆ และวิธีการไม่มีความซับซ้อน

การกำหนดความน่าจะเป็นของข้อมูลเป็นกลุ่ม V_j สำหรับข้อมูลที่มีคุณสมบัติ n ตัว ใช้สัญลักษณ์ว่า $P(a_1, a_2, \dots, a_n)$ คือ

$$P(v_j | a_1, a_2, \dots, a_n) = \prod_{i=1}^n P(a_i | v_j) \quad (9)$$

โดยที่ \prod หมายถึงผลคูณของค่า $P(a_i | v_j)$ เมื่อ i และ j มีค่าเท่ากับ $1, 2, 3, \dots, n$

วิธีการเรียนรู้แบบอย่างง่ายไปใช้วิธีดังต่อไปนี้คือ

(1) หาค่าความน่าจะเป็นของคำที่พบในแต่ละกลุ่มโดยนำค่า $P(a_1, a_2, \dots, a_n | v_j)$ จากสมการมาคูณกับค่าความน่าจะเป็นของกลุ่มนั้นๆ คือ $P(v_j)$ ได้เท่ากับ V_{NB}

(2) นำค่าที่ได้มาเปรียบเทียบกัน กลุ่มที่มีความน่าจะเป็นสูงสุดคือกลุ่มที่ข้อมูลนั้นอยู่ และจะถูกจัดเข้าไป เขียนเป็นสมการได้คือ

$$v_{NB} = \operatorname{argmax} P(v_j) \times \prod_{i=1}^n P(a_i | v_j) \quad (10)$$

ในงานวิจัยฉบับนี้ จะสร้างโมเดลการจำแนกระดับคะแนนบทวิจารณ์แบบมัลติโนเมียลนาอ์ฟเบย์ (Multinomial Naïve Bayes) ซึ่งเป็นการจำแนกระดับคะแนนบทวิจารณ์เป็น 5 กลุ่ม คือ *Very bad, Bad, Neutral, Good* และ *Very Good* โดยมีขั้นตอนดังนี้

ขั้นที่ 1: การหาความน่าจะเป็นของแต่ละกลุ่ม

$$P(v_j) = \frac{\text{count}(v_j)}{\sum_{i=1}^J \text{count}(v_i)} \quad (11)$$

ขั้นที่ 2: การหาความน่าจะเป็นของคำในแต่ละกลุ่ม

$$P(a_i | v_j) = \frac{\text{count}(a_i, v_j)}{\text{count}(v_j)} \quad (12)$$

แต่ในบางครั้งการหาความน่าจะเป็นโดยใช้ Naïve Bayes อาจจะมีกรณีที่ค่าความถี่ของคำที่เกิดขึ้นเป็น 0 หรือก็คือคำที่อยู่ในถุงคำไม่ปรากฏอยู่ในเอกสาร ทำให้ค่าความน่าจะเป็นของคำนั้นเป็น 0 ตามไปด้วย ซึ่งไม่เป็นที่ยอมรับในทางสถิติที่โอกาสในการพยากรณ์จะมีค่าเป็นศูนย์ และเพื่อหลีกเลี่ยงกรณีนี้การสร้างโมเดลการจำแนกเอกสารด้วยนาอ์ฟเบย์มักจะมีการทำ Laplace Smoothing [32] ซึ่งเป็นลักษณะการทำ Normalization โดยจะมีการเพิ่มค่าความถี่ข้อมูลเข้าไปอีกครั้งละ 1 และบวกเพิ่มค่าความถี่รวมด้วยค่าคงที่ k จากค่าทั้งหมด n คำ และกลุ่มทั้งหมด m กลุ่ม ซึ่งวิธีการนี้เป็นที่นิยมในการสร้างโมเดลเพื่อการจำแนกเอกสารด้วยนาอ์ฟเบย์ ดังนั้นจึงได้สมการนาอ์ฟเบย์ที่ปรับแล้ว ดังนี้

$$P(a_i | v_j) = \frac{1 + \text{count}(a_i, v_j)}{k + \text{count}(v_j)} \quad (13)$$

2. อัลกอริทึมเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbor)

อัลกอริทึมเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbor) [27] เป็นอัลกอริทึมที่ใช้ในการจัดกลุ่มข้อมูลที่ไม่ซับซ้อนเข้าใจง่าย ซึ่งวิธีนี้จะสามารถสร้างโมเดลที่มีประสิทธิภาพได้แม้เงื่อนไขที่ใช้ในการตัดสินใจจะมีความซับซ้อนก็ตาม ซึ่งอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจะเป็นการจำแนกประเภทข้อมูลโดยขึ้นกับข้อมูลที่มีคุณสมบัติใกล้เคียงที่สุด K ตัวจากชุดข้อมูลตัวอย่าง แล้วเลือกคลาสที่สมาชิกส่วนใหญ่ที่อยู่ในกลุ่ม K ดังกล่าว สังกัดอยู่มากที่สุดให้กับ สมาชิกใหม่ การจำแนกประเภทข้อมูลโดยใช้ข้อมูลข้างเคียง K ตัวจะประกอบด้วยเทอริบิวต์หลายตัวแปร X_i ซึ่งจะนำมาใช้ในการแบ่งกลุ่ม Y_i โดยระบุค่าตัวเลขจำนวนเต็มบวกให้กับ K ซึ่งค่านี้จะเป็นตัวบอกจำนวนของกรณี (Case) ที่จะต้องค้นหาในการทำนายกรณีใหม่ โดยในที่นี้จะกำหนด

1-KNN หมายถึง อัลกอริทึมนี้จะค้นหา 1 กรณีที่มีลักษณะใกล้เคียงกับกรณีใหม่ (1 Nearest Cases) การหาระยะทางที่หาได้จากสมาชิกในข้อมูลตัวอย่างฝึกฝน มาเรียงลำดับจากน้อยไปหามาก แล้วเลือกสมาชิกที่มีระยะทาง (Distance) ใกล้เคียงที่สุดออกมา K ตัว โดยใช้การวัดระยะทางแบบ Euclidean distance [28] ซึ่งมีหลักการ คือการวัดระยะทางระหว่างสองวัตถุ ถ้าวัตถุห่างกันมากแสดงว่าวัตถุนั้นมีความคล้ายคลึงกันน้อย ถ้าระยะทางมีค่าน้อยก็แสดงว่ามีความคล้ายคลึงกันมาก โดยที่ค่า p_i แทน คุณสมบัติจากฐานข้อมูล q_i แทน คุณสมบัติที่ผู้ใช้ระบุ

$$E(p, q) = \sqrt{\sum_{i=0}^n (p_i - q_i)^2} \quad (14)$$

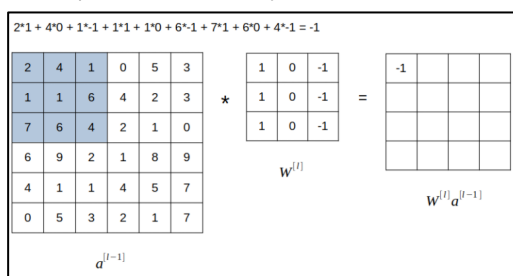
3. อัลกอริทึมโครงข่ายประสาทคอนโวลูชัน (Convolution Neural Network)

CNN ได้รับการการนำเสนอ เพื่อให้ได้ผลลัพธ์ที่น่าประทับใจในภารกิจที่สำคัญในทางปฏิบัติของการจัดหมวดหมู่ประโยค ซึ่ง CNN สามารถใช้ประโยชน์จากการแทนคำแบบกระจายโดยการแปลงโทเค็น (Tokens) ที่ประกอบด้วยแต่ละประโยคเป็นเวกเตอร์ก่อนแล้วสร้างเมทริกซ์เพื่อใช้เป็นอินพุต

Convolutional Neural Network หรือ CNN ซึ่งเป็นโครงสร้าง Neural network แบบพิเศษที่มีความสามารถในการจำแนกข้อมูลได้ดีกว่า Neural network ทั่วไปมาก โดย CNN คือการใช้ Layer ชนิดพิเศษ ที่เรียกว่า Convolution layer ซึ่งทำหน้าที่สกัดเอาส่วนต่างๆ ของข้อมูลออกมา CNN จะใช้ Convolution layer มาประกอบกับ Layer ชนิดอื่น เช่น Pooling layer แล้วนำกลุ่ม Layer ดังกล่าวมาซ้อนต่อๆ กัน โดยอาจเปลี่ยน Hyperparameter บางอย่าง เช่นขนาดของ Filter layer (ซึ่งเป็นส่วนหนึ่งของ Convolution layer) และจำนวน Channel ของ layer วิธีการนำเอาส่วน

ต่างๆ มาประกอบกันนี้ เรียกว่าเป็นโครงสร้าง (Architecture) ของ CNN ซึ่งมีหลายแบบ เช่น LeNet, AlexNet, VGG, ResNet, Inception Network เป็นต้น ส่วนประกอบต่างๆ ของ CNN ซึ่งเป็นพื้นฐานที่เป็นส่วนสำคัญในการทำงานของ CNN ดังนี้

3) Convolution layer



ภาพที่ 3 ตัวอย่างการคำนวณ Convolution

จากภาพที่ 3 สมมติเรามี Matrix ข่ายมือ ขนาด 6x6 และมี Matrix ตรงกลาง ซึ่งเรียกว่า Filter หรือ Kernel ขนาด 3x3 เราจะนำเฉพาะ 3x3 ช่องแรกของ Matrix แรก มาคูณแบบ Element-wise กับ Filter matrix แล้วนำผลที่ได้แต่ละค่า (ซึ่งมีทั้งสิ้น 9 ค่า) มาบวกกัน แล้วนำไปใส่ในแถวแรก คอลัมน์แรกของ Matrix ที่สามซึ่งเป็นผลลัพธ์ โดยในภาพ ผลลัพธ์ที่ว่า เท่ากับ -1

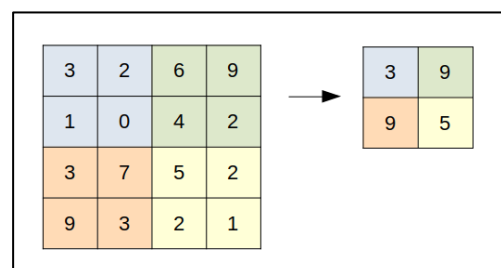
ถัดมา เราจะเลื่อนกรอบขนาด 3x3 ใน Matrix แรกไปทางขวา 1 ช่อง แล้วทำแบบเดิม ผลลัพธ์ที่ได้ นำไปใส่ในแถว 1 ช่อง 2 ของ Matrix ผลลัพธ์ ทำไปเรื่อยๆ จนสุดทาง แล้วเลื่อนกรอบ 3x3 ลงมาด้านล่าง 1 ช่อง (ขีดขอบด้านซ้ายมือ) แล้วทำแบบเดิม จนกระทั่งเติมค่าใน Matrix ผลลัพธ์จนเต็ม

กระบวนการนี้ เรียกว่า Convolution ซึ่งแสดงสัญลักษณ์ด้วย * ส่วน Neural network ที่มี Layer ที่ใช้กระบวนการ Convolution น้อยอย่างน้อย 1 Layer เราก็เรียกว่า Convolutional neural network

4) Pooling layer

หลังจากที่ข้อมูลผ่าน Convolution layer แล้ว บ่อยครั้งที่จะถูกส่งเข้า Layer อีกแบบหนึ่ง ที่เรียกว่า Pooling layer

หน้าที่ของ Pooling layer คือการสกัดเอาส่วนที่สำคัญที่สุดของข้อมูล และเพิ่มประสิทธิภาพการประมวลผลให้รวดเร็วขึ้น กลไกของ Pooling layer นั้นเรียบง่ายมาก คือการสกัดเอาเฉพาะค่าสูงสุดของ Grid เก็บไว้ใน Output เช่นจากภาพที่ 4 แสดง Pooling layer ขนาด 2x2 โดยมีค่า Stride s=2:



ภาพที่ 4 ตัวอย่างการทำ Pooling layer

Pooling layer ที่สกัดเอาเฉพาะค่าสูงสุดของ Grid เก็บไว้ เรียกว่า Max pooling ซึ่งเป็นรูปแบบที่ใช้บ่อยที่สุด นอกจากนั้นยังมี Average pooling ซึ่งหาค่าเฉลี่ยของ Grid เก็บไว้ แต่ใช้น้อยกว่า Max pooling มาก หลังจากที่ทำ Pooling layer เสร็จ ก็จะได้ feature map หรือ feature vector ที่จะนำไปทำเป็น model สำหรับทดสอบกับชุดข้อมูลอื่นๆ

ส่วนที่ 3: การวัดประสิทธิภาพโมเดลเพื่อการจำแนกระดับคะแนนของบทวิจารณ์ภาพยนตร์

เป็นขั้นตอนการประเมินโมเดลเพื่อใช้ในการจัดกลุ่มเอกสารก่อนการนำไปใช้งานจริงที่โดยทั่วไป จะใช้เทคนิคมาตรฐาน [22] คือ

การค่าความระลึก (Recall) ซึ่งจะเป็นอัตราส่วนของเอกสารที่จัดกลุ่มได้จากเอกสารทั้งหมดที่มีอยู่ โดยจะนำค่าจากตาราง Confusion-matrix มาใช้ในการคำนวณหาความระลึกได้ดังนี้

$$Recall = \frac{tp}{tp + fn} \quad (15)$$

การวัดค่าความแม่นยำ (Precision) เป็นอัตราส่วนของเอกสารที่จัดกลุ่มได้และถูกต้อง ส่วนด้วยจำนวนของเอกสารที่จัดกลุ่มได้

$$Precision = \frac{tp}{tp + fp} \quad (16)$$

การวัดค่า F-measure หรือ F1 เป็นการพิจารณาค่าความสัมพันธ์ระหว่างค่าความระลึก และค่าความแม่นยำ

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (17)$$

โดยที่ค่า F จะมีค่าระหว่าง 0 ถึง 1 ซึ่งถ้าหากค่า $F-measure$ มีค่าเข้าใกล้ 1 มากเท่าไร ก็ะหมายถึงการจัดกลุ่มเอกสารนั้นมีประสิทธิภาพ และมีความถูกต้องมากขึ้นเท่านั้น

ผลการวิจัย

ผลการประเมินประสิทธิภาพโมเดลการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์

ตารางที่ 1 ผลการประเมินที่ใช้ในการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วยอัลกอริทึม KNN

การให้น้ำหนักคำ	สัดส่วนเอกสารที่ใช้ในการสร้างโมเดล (ร้อยละ)	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
TF-IDF	100:10	0.5162	0.5021	0.5074
	100:20	0.5447	0.5246	0.5342
	100:30	0.5941	0.5702	0.5821
	ค่าเฉลี่ย	0.5462	0.5346	0.5363
Delta TF-IDF	100:10	0.5362	0.5744	0.5546
	100:20	0.5414	0.5204	0.5366
	100:30	0.5922	0.5702	0.5812
	ค่าเฉลี่ย	0.5566	0.5550	0.5574
TF-ICF-IDF	100:10	0.5610	0.5532	0.5564
	100:20	0.6012	0.5830	0.5912
	100:30	0.6332	0.6242	0.6262
	ค่าเฉลี่ย	0.5967	0.5834	0.5866
TF-RF	100:10	0.6601	0.6410	0.6532
	100:20	0.6911	0.6862	0.6812
	100:30	0.7135	0.7046	0.7062
	ค่าเฉลี่ย	0.6863	0.6734	0.6767
TF- IGM	100:10	0.6956	0.6884	0.6894
	100:20	0.7103	0.7014	0.7063
	100:30	0.7345	0.7264	0.7201
	ค่าเฉลี่ย	0.7134	0.7054	0.7052

ตารางที่ 2 ผลการประเมินที่ใช้ในการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วยอัลกอริทึม Naïve Bayes

การให้น้ำหนักคำ	สัดส่วนเอกสารที่ใช้ในการสร้างโมเดล (ร้อยละ)	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
TF-IDF	100:10	0.5546	0.5350	0.5421
	100:20	0.5747	0.5542	0.5632
	100:30	0.6304	0.6346	0.6323
	ค่าเฉลี่ย	0.5767	0.5764	0.5737
Delta TF-IDF	100:10	0.5431	0.5294	0.5374
	100:20	0.5766	0.5546	0.5675
	100:30	0.6445	0.6328	0.6368
	ค่าเฉลี่ย	0.5769	0.5568	0.5734
TF-ICF-IDF	100:10	0.6143	0.6233	0.6176
	100:20	0.6436	0.6312	0.6366
	100:30	0.6744	0.6561	0.6674
	ค่าเฉลี่ย	0.6424	0.6337	0.6339
TF-RF	100:10	0.6332	0.6242	0.6262
	100:20	0.6911	0.6862	0.6812
	100:30	0.7135	0.7046	0.7062
	ค่าเฉลี่ย	0.6863	0.6734	0.6767
TF- IGM	100:10	0.6977	0.6945	0.6912
	100:20	0.7216	0.7264	0.7235
	100:30	0.7448	0.7468	0.7482
	ค่าเฉลี่ย	0.7287	0.7266	0.7232

ตารางที่ 3 ผลการประเมินที่ใช้ในการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วยอัลกอริทึม CNN

การให้น้ำหนักคำ	สัดส่วนเอกสารที่ใช้ในการสร้างโมเดล (ร้อยละ)	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
TF-IDF	100:10	0.5562	0.5644	0.5546
	100:20	0.5914	0.6304	0.6066
	100:30	0.6398	0.6202	0.6412
	ค่าเฉลี่ย	0.5966	0.6150	0.6074

ตารางที่ 3 ผลการประเมินที่ใช้ในการจำแนกบทวิจารณ์สินค้าอิเล็กทรอนิกส์ด้วยอัลกอริทึม CNN (ต่อ)

การให้น้ำหนักคำ	สัดส่วนเอกสารที่ใช้ในการสร้างโมเดล (ร้อยละ)	ค่าความระลึก	ค่าความแม่นยำ	ค่าเฉลี่ย F-measure
<i>Delta TF-IDF</i>	100:10	0.5610	0.5832	0.5764
	100:20	0.6112	0.6530	0.6412
	100:30	0.6632	0.6742	0.6662
	ค่าเฉลี่ย	0.6267	0.6534	0.6266
<i>TF-ICF-IDF</i>	100:10	0.6342	0.6384	0.6362
	100:20	0.6871	0.6872	0.6852
	100:30	0.7135	0.7066	0.7102
	ค่าเฉลี่ย	0.6782	0.6774	0.6761
<i>TF-RF</i>	100:10	0.6221	0.6512	0.6354
	100:20	0.7398	0.7130	0.7212
	100:30	0.8132	0.7954	0.8054
	ค่าเฉลี่ย	0.7257	0.7198	0.7282
<i>TF- IGM</i>	100:10	0.6552	0.6752	0.6652
	100:20	0.7598	0.7430	0.7512
	100:30	0.8142	0.8214	0.8112
	ค่าเฉลี่ย	0.7430	0.7438	0.7441

สำหรับรูปแบบการให้น้ำหนักคำแต่ละรูปแบบจะเห็นได้ชัดว่ารูปแบบการให้น้ำหนักคำ *TF-IGM* มีค่าเฉลี่ยสูงสุดในทุกอัลกอริทึม เนื่องจากรูปแบบการให้น้ำหนักคำแบบ *TF-IGM* นั้น ถูกนำเสนอให้วัดความไม่สม่ำเสมอหรือความเข้มข้นของการแจกแจงคำศัพท์ระหว่างคลาสซึ่งสะท้อนให้เห็นถึงอำนาจการจำแนกชั้นข้อตกลง จึงทำให้เห็นความชัดเจนของการแยกข้อมูลในแต่ละคลาสเป็นอย่างดี ซึ่งเมื่อนำรูปแบบการให้น้ำหนักคำไปใช้กับอัลกอริทึม *CNN* แล้วทำให้เห็นว่าหากเอกสารมีข้อมูลไม่สมดุลมากการให้น้ำหนักคำแบบ *TF-IGM* ที่ใช้กับอัลกอริทึม *CNN* สามารถแก้ปัญหาได้ดีที่สุดเมื่อเอกสารมีสัดส่วนที่ 100: 10 โดยมีค่าเฉลี่ยอยู่ที่ 0.6652 เมื่อเทียบกับรูปแบบอื่นๆ รองลงมาคือรูปแบบการให้น้ำหนักคำแบบ *TF-ICF-IDF* ที่มี

ค่าเฉลี่ยอยู่ที่ 0.6362 และรูปแบบที่มีค่าเฉลี่ยต่ำสุดคือ *TF-IDF* ที่มีค่าเฉลี่ยอยู่ที่ 0.5546

สำหรับรูปแบบการให้น้ำหนักที่มีค่าเฉลี่ยมากที่สุดที่ทดสอบกับชุดข้อมูลมีสัดส่วน 100:20 และ 100:30 นั้น คือรูปแบบ *TF-IGM* ที่ทดสอบกับอัลกอริทึม *CNN* เช่นเดียวกับสัดส่วน 100:10 โดยมีค่าเฉลี่ย *F-measure* อยู่ที่ 0.7512 และ 0.8112 ตามลำดับ ซึ่งสัดส่วน 100:30 เป็นค่าที่สูงที่สุดในการทดสอบรูปแบบการให้ทั้งหมด และเห็นได้ชัดว่าหากข้อมูลมีค่าความไม่สมดุลต่างกันั้นก็จะให้การจำแนกข้อมูลมีประสิทธิภาพมาก

วิจารณ์และสรุปผล

เนื่องจากบ่อยครั้งที่ การจำแนกเอกสารที่ไม่สมดุลกันนั้นมีการเอนเอียงการให้คะแนนไปฝั่งที่มี

ข้อมูลมากกว่าเนื่องจากมีข้อมูลที่ครอบคลุมการทำนายที่ดีกว่า

ดังนั้นงานวิจัยฉบับนี้จึงได้นำเสนอวิธีการการจำแนกข้อมูลที่ไม่สมดุลด้วยการให้น้ำหนักคำเปรียบเทียบ 2 รูปแบบหลักคือ UTW และ STW โดย UTW ใช้รูปแบบการให้น้ำหนักคำที่ได้รับความนิยมมากที่สุดคือ TF-IDF และ STW ใช้ทั้งหมด 4 รูปแบบคือ Delta TF-IDF, TF-ICF-IDF, TF-RF และ TF-IGM โดยผลที่ได้คือการให้น้ำหนักคำแบบ STW มีประสิทธิภาพในการจำแนกข้อมูลที่ไม่สมดุลมากกว่ารูปแบบการให้น้ำหนักคำแบบ UTW ซึ่งได้แก่การให้น้ำหนักคำแบบ TF-IGM โดยใช้อัลกอริทึม CNN ในการสร้างโมเดล มีค่าเฉลี่ย F-measure สูงที่สุดอยู่ที่ 74.41%

เอกสารอ้างอิง

- [1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," May 2002, Accessed: Aug. 06, 2020. [Online]. Available: <http://arxiv.org/abs/cs/0205070>.
- [2] Y. Li, G. Sun, and Y. Zhu, "Data imbalance problem in text classification," *Proc. - 3rd Int. Symp. Inf. Process. ISIP 2010*, pp. 301–305, 2010, doi: 10.1109/ISIP.2010.47.
- [3] Y. Liu, H. T. Loh, and A. Sun, "Imbalanced text classification: A term weighting approach," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 690–701, 2009, doi: <https://doi.org/10.1016/j.eswa.2007.10.042>.
- [4] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," 2002.
- [5] R. Longadge and S. Dongre, "Class Imbalance Problem in Data Mining Review," May 2013, Accessed: Aug. 06, 2020. [Online]. Available: <http://arxiv.org/abs/1305.1707>.
- [6] J. Ah-Pine and E. P. S. Morales, "A study of synthetic oversampling for twitter imbalanced sentiment analysis," *CEUR Workshop Proc.*, vol. 1646, pp. 17–24, 2016.
- [7] C. Zhang, J. Bi, and P. Soda, *Feature selection and resampling in class imbalance learning: Which comes first? An empirical study in the biological domain*. 2017.
- [8] F. Ren and M. G. Sohrab, "Class-indexing-based term weighting for automatic text classification," *Inf. Sci. (Ny)*, vol. 236, pp. 109–125, 2013, doi: <https://doi.org/10.1016/j.ins.2013.02.029>.
- [9] Y. Gu and X. Gu, "A Supervised Term Weighting Scheme for Multi-class Text Categorization BT - Intelligent Computing Methodologies," 2017, pp. 436–447.
- [10] P. Juszczak and R. P. W. Duin, "Uncertainty sampling methods for one-class classifiers."
- [11] F. Debole and F. Sebastiani, "Supervised Term Weighting for Automated Text Categorization BT - Text Mining and its Applications," 2004, pp. 81–97.
- [12] A. C. E. S. Lima and L. N. de Castro, "Automatic sentiment analysis of Twitter messages," in *2012 Fourth International Conference on Computational Aspects*

- of Social Networks (CASoN), 2012, pp. 52–57, doi: 10.1109/CASoN.2012.6412377.
- [13] M. Ibrahim and M. Carman, “Undersampling Techniques to Re-balance Training Data for Large Scale Learning-to-Rank BT - Information Retrieval Technology,” 2014, pp. 444–457.
- [14] V. Balakrishnan and L.-Y. Ethel, “Stemming and Lemmatization: A Comparison of Retrieval Performances,” *Lect. Notes Softw. Eng.*, vol. 2, no. 3, pp. 262–267, 2014, doi: 10.7763/Inse.2014.v2.134.
- [15] F. Sebastiani, “Machine Learning in Automated Text Categorization.” [Online]. Available: www.ira.uka.de/bibliography/Ai/automated.text.
- [16] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988, doi: [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
- [17] G. Domeniconi, G. Moro, R. Pasolini, and C. Sartori, “A Comparison of Term Weighting Schemes for Text Classification and Sentiment Analysis with a Supervised Variant of tf.idf BT - Data Management Technologies and Applications,” 2016, pp. 39–58.
- [18] M. Lan, C. L. Tan, J. Su, and Y. Lu, “Supervised and Traditional Term Weighting Methods for Automatic Text Categorization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 721–735, 2009, doi: 10.1109/TPAMI.2008.110.
- [19] J. Martineau, T. Finin, C. Fink, C. Piatko, J. Mayfield, and Z. Syed, “Delta TFIDF: An Improved Feature Space for Sentiment Analysis,” *Proc. Second Int. Conf. Weblogs Soc. Media (ICWSM)*, vol. 29, no. May, pp. 490–497, 2008, [Online]. Available: <http://ebiquity.umbc.edu/papers/select/person/Tim/Finin/>.
- [20] K. Chen, Z. Zhang, J. Long, and H. Zhang, “Turning from TF-IDF to TF-IGM for term weighting in text classification,” *Expert Syst. Appl.*, vol. 66, pp. 245–260, 2016, doi: <https://doi.org/10.1016/j.eswa.2016.09.009>.
- [21] T. Dogan and A. K. Uysal, “Improved inverse gravity moment term weighting for text classification,” *Expert Syst. Appl.*, vol. 130, pp. 45–59, 2019, doi: <https://doi.org/10.1016/j.eswa.2019.04.015>.
- [22] D. M. W, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION,” *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011, [Online]. Available: <http://dspace.flinders.edu.au/dspace/http://www.bioinfo.in/contents.php?id=51>.
- [23] S. Li, G. Zhou, Z. Wang, S. Y. M. Lee, and R. Wang, “Imbalanced sentiment classification,” *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 2469–2472, 2011, doi: 10.1145/2063576.2063994.

โปสเตอร์โครงการ

A Method of Imbalanced Sentiment Classification

กระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุล



MAHASARAKHAM
UNIVERSITY

ผู้พัฒนา : พิระวัฒน์ บุญบ้านจั่ว (Pheerawat Bunbanngio)

อาจารย์ที่ปรึกษา : จันทิมา พลพิณ (Jantima Polpinij)

Intellect Laboratory สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
knarf.pheerawat@gmail.com, jantima.p@msu.ac.th



ที่มาและความสำคัญ

การจำแนกความรู้สึก (Sentiment Classification) คือการจำแนกเอกสารตามข้อความซึ่งโดยทั่วไปจะจำแนกเป็นความรู้สึกที่เป็นบวก (Positive) ความรู้สึกที่เป็นลบ (Negative) และความรู้สึกที่เป็นกลาง (Neutral) โดยการเรียนรู้จากข้อมูลนั้น ได้รับการศึกษาอย่างต่อเนื่อง เพราะการประยุกต์ใช้หลายลักษณะ แต่โดยทั่วไปมักจะนิยมใช้ในการจำแนกความรู้สึกที่มีการแสดงไว้ในรูปแบบข้อความ (Text) เช่น ประยุกต์ใช้ในการจัดอันดับความรู้สึกจากข้อความแสดงความคิดเห็นของผู้คนที่ได้รับสินค้าและบริการ การประยุกต์ใช้เพื่อวิเคราะห์ความรู้สึกของผู้คนในเหตุการณ์เมือง เป็นต้น ซึ่งปัญหาความไม่สมดุลของข้อมูลในคลาสนั้น เกิดจากกลุ่มตัวอย่างที่ใช้ในการเรียนรู้มีข้อมูลไม่สมดุลกัน โดยกลุ่มที่มีข้อมูลมากกว่าจะเรียกว่า "ข้อมูลกลุ่มหลัก (Majority Class)" ขณะที่กลุ่มตัวอย่างที่มีข้อมูลจำนวนน้อยกว่าจะเรียกว่า "ข้อมูลกลุ่มรอง (Minority Class)" เมื่อนำเอาข้อมูลในลักษณะนี้ไปเรียนรู้เพื่อสร้างตัวจำแนกความรู้สึก (Sentiment Classifier) ข้อมูลใหม่ๆ ที่อ่านเข้ามาเพื่อวิเคราะห์เพื่อจำแนกกลุ่มด้วยตัวจำแนกความรู้สึกดังกล่าว ก็มีแนวโน้มที่จะทำนายกลุ่มของข้อมูลนั้นไปยังทิศทางของข้อมูลกลุ่มหลักที่ใช้ในการเรียนรู้ตัวจำแนกความรู้สึก ดังนั้น ในโครงการปัญญาประดิษฐ์นี้ จึงได้นำเสนอการศึกษาการแก้ปัญหาความไม่สมดุลของข้อมูลในการจำแนกความรู้สึกด้วยเทคนิคการให้น้ำหนักค่า 5 เทคนิค คือ TF-IDF, Delta TF-IDF, TF-IDF-ICF, TF-RF และ TF-IGM ร่วมกับแมชชีนเลิร์นนิง 3 ตัว คือ Naive Bayes, K-Nearest Neighbor และสุดท้าย Convolution Neural Network

คำสำคัญ: การจำแนกเอกสาร, การให้น้ำหนักค่า, ข้อมูลไม่สมดุล, ซัพพอร์ตเวกเตอร์แมชชีน

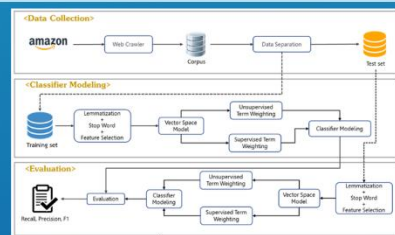
วัตถุประสงค์

นำเสนอกระบวนการสำหรับการจำแนกความรู้สึกที่มีข้อมูลไม่สมดุลโดยมีเครื่องมือหลักคือเทคนิคการให้น้ำหนักค่าแบบมีผู้สอน

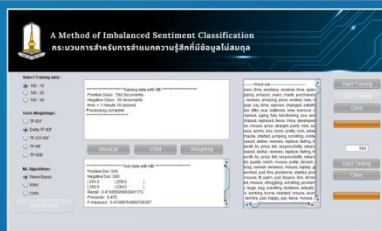
ประโยชน์ที่คาดว่าจะได้รับ

ได้กระบวนการในการจำแนกข้อความแสดงความรู้สึกที่มีข้อมูลแบบไม่สมดุล

กรอบการดำเนินงาน



ตัวอย่างหน้าจอการทำงาน



สรุป

เนื่องจากบ่อยครั้งที่ การจำแนกเอกสารที่ไม่สมดุลกันนั้นมีการเอนเอียงการให้ค่าแบบไม่เชิงที่มีข้อมูลมากกว่าเนื่องจากมีข้อมูลที่ครอบคลุมการทำนายที่ดีกว่า

ดังนั้นงานวิจัยฉบับนี้จึงได้นำเสนอวิธีการการจำแนกข้อมูลที่ไม่สมดุลด้วยการให้น้ำหนักค่าเปรียบเทียบ 2 รูปแบบหลักคือ UTW และ STW โดย UTW ใช้รูปแบบการให้น้ำหนักค่าที่ได้รับความนิยมมากที่สุดคือ TF-IDF และ STW ใช้ทั้งหมด 4 รูปแบบคือ Delta TF-IDF, TF-ICF-IDF, TF-RF และ TF-IGM โดยผลที่ได้คือการให้น้ำหนักค่าแบบ STW มีประสิทธิภาพในการจำแนกข้อมูลที่ไม่สมดุลมากกว่ารูปแบบการให้น้ำหนักค่าแบบ UTW ซึ่งได้แก่การให้น้ำหนักค่าแบบ TF-IGM โดยใช้อัลกอริทึม CNN ในการสร้างโมเดล มีค่าเฉลี่ย F-measure สูงที่สุดอยู่ที่ 74.41%

ประวัติผู้จัดทำโครงการ

ประวัติย่อผู้จัดทำโครงการ

ประวัติย่อผู้จัดทำโครงการคนที่ 1

ชื่อ ชื่อสกุล	พิระวัฒน์ บุญบ้านจั่ว
วัน เดือน ปีเกิด	วันที่ 23 ตุลาคม 2542
สถานที่เกิด	อำเภอเมือง จังหวัดขอนแก่น
ที่อยู่ที่สามารถติดต่อได้	153 ม.19 ต.สาวะถี อ.เมือง จ.ขอนแก่น 40000
โทรศัพท์มือถือ	080-941-0986
อีเมล	boonbannkiw231042@gamil.com
ประวัติการศึกษา	พ.ศ. 2560 ได้สำเร็จการศึกษาชั้นมัธยมศึกษาตอนปลาย โรงเรียนนครขอนแก่น จังหวัดขอนแก่น