การสกัดคุณลักษณะข้อความที่มีประสิทธิภาพเพื่อการจำแนกขั้วความคิดเห็น

Efficient Text Feature Extraction for Opinion Polarity Classification

ณิชาภัทร ปิ่นโพธิ์

Nichapat Pinpo

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Science
Prince of Songkla University

2563

การสกัดคุณลักษณะข้อความที่มีประสิทธิภาพเพื่อการจำแนกขั้วความคิดเห็น

Efficient Text Feature Extraction for Opinion Polarity Classification

ณิชาภัทร ปิ่นโพธิ์

Nichapat Pinpo

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Science
Prince of Songkla University
2563

ชื่อวิทยานิพนธ์ การสกัดคุณลักษณะข้อความที่มีประสิทธิภาพเพื่อการจำแนกขั้วความคิดเห็น
ผู้เขียน นางสาวณิชาภัทร ปิ่นโพธิ์
สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

.....

(ดร.นิวรรณ วัฒนกิจรุ่งโรจน์)

คณะกรรมการสอบ

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สิรภัทร เชี่ยวชาญวัฒนา)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ลัดดา ปรีชาวีรกุล)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.จารุณี ดวงสุวรรณ)

.....กรรมการ

(ดร.นิวรรณ วัฒนกิจรุ่งโรจน์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

.....

(ศาสตราจารย์ ดร.ดำรงศักดิ์ ฟ้ารุ่งสาง)

คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ.....
(ดร.นิวรรณ วัฒนกิจรุ่งโรจน์)
อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ.....
(นางสาวณิชาภัทร ปิ่นโพธิ์)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....
(นางสาวณิชาภัทร ปิ่นโพธิ์)
นักศึกษา

ชื่อวิทยานิพนธ์ การสกัดคุณลักษณะข้อความที่มีประสิทธิภาพเพื่อการจำแนกขั้วความคิดเห็น

ผู้เขียน นางสาวณิชาภัทร ปิ่นโพธิ์

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2562

บทคัดย่อ

ในปัจจุบัน ผู้ใช้สื่อสังคมออนไลน์สามารถที่จะแสดงความคิดเห็นผ่านการพิมพ์ข้อความในเรื่องที่สนใจได้อย่างอิสระ ข้อความเหล่านั้นสามารถนำมาวิเคราะห์เพื่อจำแนกหาทิศทางการแสดงความคิดเห็นในเชิงบวกและเชิงลบ โดยการวิเคราะห์หาทิศทางความคิดเห็นจะต้องสร้างเวกเตอร์เพื่อใช้เป็นตัวแทนของข้อความก่อน วิธีทั่วไป คือ การแทนข้อความด้วยเวกเตอร์แสดงค่าน้ำหนักหรือค่าความถี่ของคำที่มีจำนวนมิติเท่ากับจำนวนคำศัพท์ที่มีอยู่ในพจนานุกรมที่ประกอบด้วยคำศัพท์ทั้งหมดที่สามารถมีได้ในข้อความทั้งหมดที่พิจารณา ถ้าคำศัพท์มีปริมาณมาก จำนวนคำที่มีอยู่ในพจนานุกรมจะเพิ่มขึ้น ทำให้เวกเตอร์แทนข้อความที่ได้นั้นจะมีขนาดใหญ่ตามไปด้วย ซึ่งจะทำให้การสร้างและใช้โมเดลในการจำแนกขั้วความคิดเห็นต้องใช้เวลาในการประมวลผลที่นาน

วิทยานิพนธ์นี้ ได้นำเสนอการสกัดคุณลักษณะแทนข้อความในรูปของเวกเตอร์ 2 รูปแบบ คือ เวกเตอร์ V4D และเวกเตอร์ V8D ซึ่งเป็นเวกเตอร์ที่มีมิติน้อย โดยมีการพิจารณาคุณลักษณะที่ได้มาจาก ค่าน้ำหนักคำเชิงบวกและคำเชิงลบที่ปรากฏในข้อความ นอกจากนี้ยังได้มีการพิจารณาคุณลักษณะที่ได้จากคำศัพท์บอกการปฏิเสธซึ่งมีความสำคัญต่อความหมายของข้อความและการจำแนกขั้วความคิดเห็น เวกเตอร์แทนข้อความที่ได้นำเสนอจะถูกใช้เป็นข้อมูลนำเข้าเพื่อสร้างโมเดลในการจำแนก ซึ่งในงานวิทยานิพนธ์นี้ทำการศึกษาการสร้างโมเดล 4 วิธี ได้แก่ วิธี k-Nearest Neighbors วิธี Naive Bayes วิธี Artificial Neural Networks และวิธี Support Vector Machine จากการทดลองบนชุดข้อมูลข้อความแสดงความคิดเห็นที่มาจากหลากหลายของโดเมนจำนวน 8 ชุดข้อมูล เพื่อเปรียบเทียบประสิทธิภาพของการสกัดคุณลักษณะในรูปแบบของเวกเตอร์แทนข้อความที่เสนอ ได้แก่ เวกเตอร์ V4D และเวกเตอร์ V8D กับการสกัดคุณลักษณะในรูปแบบของเวกเตอร์แบบดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF ซึ่งได้ถูกนำมาเป็นข้อมูลนำเข้าในการสร้างโมเดลสำหรับจำแนกขั้วความคิดเห็น พบว่า เวกเตอร์แทนข้อความที่เสนอช่วยเพิ่มความถูกต้องในการจำแนกขั้วความคิดเห็นและให้ประสิทธิภาพในแง่ของพื้นที่ในการจัดเก็บข้อมูลและเวลาที่ใช้ในการประมวลผลได้ดีที่สุด

Thesis Title Efficient Text Feature Extraction for Opinion Polarity Classification
Author Miss Nichapat Pinpo
Major Program Computer Science
Academic Year 2019

ABSTRACT

Recently, social media users can comment with texts to describe their opinions. These texts can be analyzed to classify them into positive and negative directions. Before creating classifier, the feature vectors for representing the texts must be prepared firstly. Generally, texts are represented by vectors of weights or frequencies of terms that appear in the text. The number of dimensions of vector is equal to the number of terms in the dictionary derived from the possible words in all texts. The large amount of words in dictionary leads to the high dimensional vector for representing text and bring about the long processing time to training and testing the text classification models.

This thesis proposed two methods for representing texts including V4D and V8D which are the low-dimensional vectors. The set of positive and negative words were considered to create the vectors. In addition, the feature vectors were derived by using the words of negation which have the significant meanings in a classification of text opinions. In this thesis, four classification techniques including k - Nearest Neighbors, Naive Bayes, Artificial Neural Networks and Support Vector Machine were studies to classify the opinion texts. By experimenting on eight data sets with various domains, the proposed vectors, including V4D and V8D, were compared with the traditional vectors, including TF and TF-IDF in the view of the performances when they were applied to the classification problem. The experimental results show that the proposed vectors for representing text can improve the performance of opinion text classification and provide the best efficiency in the terms of used space and processing time.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความช่วยเหลือและการสนับสนุนจากบุคคลหลายฝ่าย ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง คือ

ดร.นิวรรณ วัฒนกิจรุ่งโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่กรุณาให้คำปรึกษาแนะนำและช่วยเหลือในการแก้ปัญหาต่างๆ ให้แก่ผู้วิจัยเสมอมา พร้อมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ผู้ช่วยศาสตราจารย์ ดร.สิรภัทร เชี่ยวชาญวัฒนา ประธานกรรมการในการสอบวิทยานิพนธ์ ที่กรุณาช่วยให้คำแนะนำที่มีคุณค่า ทำให้วิทยานิพนธ์ให้มีความสมบูรณ์

ผู้ช่วยศาสตราจารย์ ดร.ลัดดา ปรีชาวีรกุล กรรมการในการสอบวิทยานิพนธ์ที่กรุณาช่วยให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

ผู้ช่วยศาสตราจารย์ ดร.จารุณี ดวงสุวรรณ กรรมการในการสอบวิทยานิพนธ์ที่กรุณาช่วยให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

อาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ทุกท่าน ที่ให้ความรู้ด้านวิชาการ ซึ่งสามารถนำมาใช้ในการทำวิทยานิพนธ์ได้เป็นอย่างดียิ่ง

เจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่ให้ความช่วยเหลือและอำนวยความสะดวกเกี่ยวกับเอกสารต่างๆ

เพื่อนๆ พี่ๆ และน้องๆ ภาควิชาวิทยาการคอมพิวเตอร์ ที่คอยให้กำลังใจและช่วยเหลือให้คำปรึกษาในการทำวิทยานิพนธ์

คุณพ่อ คุณแม่ และน้องสาว รวมถึงญาติๆ ทุกคน ที่ให้การสนับสนุนคอยเป็นห่วงสุขภาพและให้กำลังใจแก่ผู้วิจัยมาโดยตลอด

ผู้วิจัยขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

ณิชาภัทร ปิ่นโพธิ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

รายการตาราง

ตาราง **หน้า**

รายการตาราง (ต่อ)

ตาราง **หน้า**

รายการภาพประกอบ

รูป **หน้า**

รายการภาพประกอบ (ต่อ)

รูป **หน้า**

รายการภาพประกอบ (ต่อ)

รูป **หน้า**

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

ในปัจจุบันสื่อสังคมออนไลน์ได้รับความนิยมอย่างมาก เนื่องจากผู้คนสามารถที่จะติดต่อสื่อสารกันได้อย่างสะดวกรวดเร็วและประหยัดค่าใช้จ่าย อีกทั้งยังสามารถแลกเปลี่ยนข้อมูลและแสดงความคิดเห็นในเรื่องที่สนใจทั้งในทิศทางเชิงบวกและทิศทางเชิงลบได้อย่างอิสระ ส่วนใหญ่การแสดงความคิดเห็นของผู้ใช้จะอยู่ในรูปแบบข้อความ ซึ่งข้อความแสดงความคิดเห็นเหล่านั้นจะมีอยู่จำนวนมากมหาศาลบนสื่อสังคมออนไลน์ โดยที่สามารถนำมาใช้ในการวิเคราะห์หาทิศทางความคิดเห็นได้ และข้อความแสดงความคิดเห็นมีประโยชน์อย่างมากในด้านต่างๆ เช่น ทางด้านธุรกิจ (Cho & Kang, 2012) การแสดงความคิดเห็นที่มีต่อสินค้า จะช่วยให้ทำให้รู้ถึงข้อบกพร่องแล้วนำไปพัฒนาเพื่อให้ได้สินค้าที่มีคุณภาพและตรงตามความต้องการของผู้บริโภคมากยิ่งขึ้น ทางด้านบริการ (Isah et al., 2014) ผู้ประกอบการสามารถนำความคิดเห็นไปปรับปรุงและพัฒนาคุณภาพการบริการให้ดียิ่งขึ้น และทางด้านสังคมและการเมืองการปกครอง (Ramteke et al., 2016) การทราบถึงความคิดเห็นและปัญหาของประชาชน จะช่วยให้หน่วยงานรัฐสามารถกำหนดนโยบายและพัฒนาเศรษฐกิจและสังคมให้ตรงตามความต้องการของประชาชนได้อย่างมีประสิทธิภาพ

โดยทั่วไป ข้อความแสดงความคิดเห็นอยู่ในรูปของภาษาธรรมชาติซึ่งเป็นข้อมูลที่ไม่มีโครงสร้าง (Gharehchopogh & Khalifelu, 2011) การที่จะนำข้อความแสดงความคิดเห็นไปวิเคราะห์หาทิศทางการแสดงความคิดเห็นได้นั้น จะต้องผ่านการสกัดคุณลักษณะที่เป็นตัวแทนของข้อความเพื่อสามารถคำนวณด้วยคอมพิวเตอร์ได้ โดยการสกัดคุณลักษณะส่วนใหญ่จะอยู่ในรูปแบบของเวกเตอร์หลายมิติ (Manning et al., 2014) เช่น เวกเตอร์ TF เป็นการแทนข้อความด้วยเวกเตอร์แสดงค่าน้ำหนักหรือค่าความถี่ของคำที่มีจำนวนมิติเท่ากับจำนวนคำศัพท์ที่มีอยู่ในพจนานุกรมที่ประกอบด้วยคำศัพท์ทั้งหมดที่สามารถมีได้ในข้อความทั้งหมดที่พิจารณา และอีกเทคนิคหนึ่งที่นิยมใช้เช่นกัน คือ เวกเตอร์ TF-IDF เป็นเวกเตอร์แทนข้อความที่มีการพิจารณาทั้งเวกเตอร์ TF และ IDF ร่วมกัน โดย IDF มีหลักการดังนี้ ทำการพิจารณาจำนวนเอกสารที่มีคำศัพท์นั้นๆ ปรากฏอยู่ด้วย ถ้า

หากคำศัพท์นั้นปรากฏอยู่ในเอกสารจำนวนมาก แสดงว่าคำนั้นมีความสำคัญน้อย ในทางกลับกันถ้าคำศัพท์นั้นปรากฏแค่เพียงไม่กี่เอกสาร จะถือว่าคำนั้นมีความสำคัญมากเพราะเป็นคำที่หายาก เมื่อได้เวกเตอร์ที่เป็นตัวแทนข้อความแล้ว หลังจากนั้นนำไปวิเคราะห์เพื่อสร้างโมเดลทำนายขั้วทิศทางความคิดเห็น

การจำแนกขั้วความคิดเห็นจะต้องแบ่งข้อมูลออกเป็น 2 ชุด ได้แก่ ชุดข้อมูลที่หนึ่งใช้สำหรับสอนเพื่อสร้างโมเดลในการจำแนก โดยจะมีข้อมูลที่ใช้นำเข้าสำหรับการสร้าง 2 ส่วน คือ เวกเตอร์แทนข้อความและประเภทของความคิดเห็น (label) ซึ่งเป็นขั้วความคิดเห็นเชิงบวกและเชิงลบของแต่ละข้อความ และอีกชุดข้อมูลจะใช้สำหรับการประเมินผลความถูกต้องของโมเดลที่ได้ การจำแนกที่กล่าวมานั้น เป็นวิธีการสร้างโมเดลแบบมีผู้สอน (Supervised Machine Learning) โดยวิธีที่นิยมใช้กัน ได้แก่ วิธี k-Nearest Neighbors วิธี Naive Bayes วิธี Artificial Neural Networks และวิธี Support Vector Machine

สิ่งที่ยังคงท้าทาย นั้นคือ เทคนิคในการสกัดคุณลักษณะแทนข้อความ ซึ่งการแทนข้อความด้วยเวกเตอร์ที่แสดงค่าน้ำหนักหรือค่าความถี่ของคำที่มีจำนวนมิติเท่ากับจำนวนคำศัพท์ที่มีอยู่ในพจนานุกรมที่ประกอบด้วยคำที่ได้มาจากคำศัพท์ในข้อความทั้งหมดที่นำมาพิจารณา ถ้าหากคำศัพท์มีปริมาณมาก จำนวนคำที่มีอยู่ในพจนานุกรมจะเพิ่มขึ้น ทำให้เวกเตอร์แทนข้อความที่ได้นั้นจะมีขนาดใหญ่ตามไปด้วย ซึ่งจะทำให้การสร้างและใช้โมเดลในการจำแนกขั้วความคิดเห็นต้องใช้เวลาในการประมวลผลที่นานอีกด้วย ในการทำวิทยานิพนธ์นี้ได้มีแนวคิดที่จะนำเสนอวิธีการสร้างเวกเตอร์แทนข้อความที่มีมิติน้อย ซึ่งจะประหยัดเนื้อที่และใช้เวลาในการเรียนรู้เพื่อสร้างโมเดลได้เร็ว แต่ยังคงต้องสามารถนำไปใช้ในการสร้างโมเดลที่มีประสิทธิภาพในการจำแนกข้อความแสดงความคิดเห็นได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์

1) ศึกษาการเตรียมข้อมูลที่เป็นข้อความ
2) เสนอกระบวนการสกัดคุณลักษณะแทนข้อความที่มีประสิทธิภาพเพื่อจำแนกขั้วความคิดเห็น
3) เปรียบเทียบกระบวนการสกัดคุณลักษณะแทนข้อความที่มีประสิทธิภาพเพื่อจำแนกขั้วความคิดเห็นกับงานวิจัยที่เกี่ยวข้อง

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รูปแบบเวกเตอร์คุณลักษณะแทนข้อความที่มีประสิทธิภาพเพื่อจำแนกขั้วความคิดเห็น
2) ได้แบบจำลองสำหรับจำแนกข้อความแสดงความคิดเห็น

1.4 งานวิจัยที่เกี่ยวข้อง

การจำแนกขั้วความคิดเห็นมีการศึกษาและวิจัยกันอย่างแพร่หลายในหลากหลายโดเมน เช่น ความคิดเห็นทางด้านธุรกิจ ความคิดเห็นทางด้านการเมือง ความคิดเห็นทางด้านการแพทย์ ผู้วิจัยได้นำเสนอการสกัดคุณลักษณะเพื่อแทนข้อความและใช้การเรียนรู้ของเครื่องเพื่อจำแนกความคิดเห็น

Text Sentiment Classification for SNS-based marketing using Domain Sentiment Dictionary (Cho & Kang, 2012)

ในปี ค.ศ. 2012 Cho และ Kang ได้นำเสนอวิธีการจำแนกข้อความที่ประกอบด้วยคำศัพท์ทางการและคำศัพท์ที่ไม่เป็นทางการซึ่งเป็นคำศัพท์ใหม่หรือ emoticons ที่ผู้ใช้อินเตอร์เน็ตได้สร้างคำเหล่านั้นขึ้นมา ผู้วิจัยได้ทำการทดลองเปรียบเทียบกับวิธีการจำแนกข้อความที่ประกอบด้วยคำศัพท์ทางการอย่างเดียว โดยผู้วิจัยได้กำหนดกลุ่มโดเมนของแต่ละคำศัพท์มี 5 กลุ่ม ได้แก่ consumer product, person, travel, food และ movie โดยใช้ TF-IDF เป็นคุณลักษณะแทนข้อความจาก webpage, personal blog, twitter, facebook และ me2Day ที่มีมิติของเวกเตอร์ขึ้นอยู่กับจำนวนชุดคำศัพท์ของแต่ละโดเมน จำแนกประเภทข้อความโดยใช้ Support Vector Machine ผลการทดลองปรากฏว่าวิธีที่ผู้วิจัยนำเสนอมีประสิทธิภาพค่าความถูกต้องที่ดีสุดในทุกโดเมน

Tweet Sentiment Analysis with Classifier Ensemble (Da Silva et al., 2014)

ในปี ค.ศ. 2014 Silva, Hruschka และ Hruschka ได้นำเสนอการใช้ความหลากหลายขององค์ประกอบในการวิเคราะห์ข้อความแสดงความคิดเห็นจากทวิตเตอร์ที่มีความหลากหลายโดเมน ได้แก่ ทางด้านการเมือง ทางด้านสุขภาพ เป็นต้น และผู้วิจัยได้ทำการจำแนกประเภทของข้อความโดยการใช้ตัวจำแนกจากหลากหลายวิธีเพื่อหาผลลัพธ์ มีการเปรียบเทียบประสิทธิภาพระหว่างตัวจำแนกเดี่ยว ได้แก่ Naive Bayes, Support Vector Machine, Logistic Regression และ Random Forest กับตัวจำแนกแบบร่วมกัน (Ensemble Classifier) ในงานวิจัยนี้ผู้วิจัยได้ทำการทดลองโดยใช้คุณลักษณะ 2 รูปแบบ คือ Bag of Words และ feature hashing ที่มีมิติของเวกเตอร์ขึ้นอยู่กับ

จำนวนคำที่ปรากฏในชุดคำศัพท์ และมีการเปรียบเทียบเพื่อหาประสิทธิภาพของแต่ละคุณลักษณะ ซึ่งผลการทดลองปรากฏว่าตัวจำแนกแบบร่วมกันจะมีประสิทธิภาพที่ดีสุด และในส่วนของคุณลักษณะ Bag of Words มีประสิทธิภาพในแง่ของค่าความถูกต้องและ feature hashing มีประสิทธิภาพในแง่ของการประหยัดเวลาในการคำนวณ

Comparison of Text Sentiment Analysis based on Machine Learning (Zhang & Zheng, 2016)

ในปี ค.ศ. 2016 Zhang และ Zheng ได้นำเสนอการวิเคราะห์ขั้วความคิดเห็นที่เป็นข้อความภาษาจีน โดยใช้ประเภทของคำกริยา คำคุณศัพท์ และคำกริยาวิเศษณ์ในการวิเคราะห์ และทำการคำนวณค่าน้ำหนักของคำโดยใช้ TF-IDF เป็นคุณลักษณะแทนข้อความที่มีมิติของเวกเตอร์ขึ้นอยู่กับจำนวนคำศัพท์ที่ปรากฏในข้อความทั้งหมด ผู้วิจัยได้ทำการจำแนกประเภทข้อความโดยการเปรียบเทียบประสิทธิภาพของตัวจำแนก Support Vector Machine กับ Extreme Learning Machine with Kernels ผลการทดลองปรากฏว่าสำหรับชุดข้อมูลที่เป็นข้อความภาษาจีนในงานวิจัยนี้ ตัวจำแนก Extreme Learning Machine with Kernels ให้มีประสิทธิภาพค่าความถูกต้องที่ดีสุด

Classification of Sentiment Reviews using n-gram Machine Learning Approach (Tripathy et al., 2016)

ในปี ค.ศ. 2016 Tripathy, Agrawal และ KumarRath ได้นำเสนอการจำแนกข้อความแสดงความคิดเห็นบนชุดข้อมูลของการรีวิวภาพยนตร์จาก IMDb ผู้วิจัยได้ทำการทดลองเปรียบเทียบรูปแบบคุณลักษณะโดยใช้เทคนิค n-gram ที่มีหลากหลายรูปแบบ ได้แก่ unigram, bigram, trigram, การใช้ unigram และ bigram ร่วมกัน, การใช้ bigram และ trigram ร่วมกัน และ การใช้ unigram, bigram และ trigram ร่วมกัน และทำการคำนวณค่าน้ำหนักของคำโดยใช้ TF-IDF ในการสร้างเวกเตอร์แทนข้อความที่มีมิติของเวกเตอร์ขึ้นอยู่กับจำนวนคำศัพท์ที่ปรากฏในข้อความทั้งหมด และผู้วิจัยได้ทำการทดลองเปรียบเทียบประสิทธิภาพค่าความถูกต้องของตัวจำแนก 4 เทคนิค ได้แก่ Naive Bayes, Maximum Entropy, Support Vector Machine และ Stochastic Gradient Descent ผลการทดลองปรากฏว่าคุณลักษณะที่ใช้เทคนิค n-gram แบบ bigram ที่ใช้ตัวจำแนก Stochastic Gradient Descent ได้ให้ประสิทธิภาพดีที่สุด ซึ่งผลการทดลองในงานวิจัยนี้ ผู้วิจัยได้กล่าวไว้ว่า เมื่อค่า n ในเทคนิค n-gram มีค่าที่น้อยจะส่งผลให้มีประสิทธิภาพค่าความถูกต้องที่ดีกว่าค่า n ที่มาก

Negation Handling in Sentiment Analysis at Sentence (Farooq et al., 2017)

ในปี ค.ศ. 2017 Farooq, Mansoor, Nongaillard, Ouzrout และ Abdul Qadir ได้ศึกษาปัญหาของการระบุขอบเขตของคำศัพท์บอกถึงการปฏิเสธ (negation) และเสนอวิธีการจัดการคำศัพท์บอกถึงการปฏิเสธตามหลักภาษาศาสตร์ โดยกำหนดรายการคำศัพท์บอกถึงการปฏิเสธซึ่งทำหน้าที่เป็นตัวบ่งชี้ในการปรากฏของคำปฏิเสธ แบ่งออกเป็น 3 ประเภทดังนี้ ประเภทที่ 1 คือคำปฏิเสธทางไวยากรณ์ (syntactic negations) ประกอบด้วยคำปฏิเสธที่ทำการกลับขั้วความคิดเห็นของคำศัพท์ เช่น no, not และ never เป็นต้น และคำปฏิเสธประเภทที่ 2 คือคำปฏิเสธทาง diminisher (diminisher negations) ประกอบด้วยคำปฏิเสธที่ทำการลดค่าคะแนน strength ขั้วความคิดเห็นของคำศัพท์ เช่น hardly, little และ rarely เป็นต้น และประเภทสุดท้ายคือคำปฏิเสธทางสัณฐานวิทยา (morphological negations) ประกอบด้วยคำที่ระบุถึงการปรากฏตัวของคำปฏิเสธ ซึ่งเป็นคำปฏิเสธที่เกิดขึ้นได้โดยใช้คำนำหน้า (prefix) หรือคำต่อท้าย (suffix) ด้วยคำที่เป็นรากศัพท์ (root) เช่น คำนำหน้า ได้แก่ de-, dis-, mis- และคำต่อท้าย ได้แก่ -less โดยงานวิจัยนี้ใช้ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นจากเว็บไซต์ Amazon, Ebay และ Cnet และทำการจำแนกขั้วความคิดเห็นจากการคำนวณผลลัพธ์ค่าคะแนน strength ของแต่ละคำที่ได้มาจาก POS tagging ซึ่งผลการทดลองจากการเปรียบเทียบวิธีการจัดการคำปฏิเสธที่ผู้วิจัยนำเสนอกับวิธีที่มีอยู่แล้วปรากฏว่าผลการทดลองวิธีที่ผู้วิจัยนำเสนอจะให้ประสิทธิภาพดีที่สุดที่ค่า accuracy เท่ากับ 83.3

Sentiment Classification of Tweets with Non-Language Features (Akilandeswari & Jothi, 2018)

ในปี ค.ศ. 2018 Jeyapal และ Ganesan ได้นำเสนอการวิเคราะห์ข้อความจากทวิตเตอร์ โดยมีการเปรียบเทียบคุณลักษณะของข้อความที่นำมาวิเคราะห์ซึ่งมี 2 รูปแบบดังนี้ รูปแบบแรกคือ Language Features ประกอบด้วยคำศัพท์ภาษาอังกฤษซึ่งนำประเภทของคำคุณศัพท์ คำกริยา และคำกริยาวิเศษณ์มาใช้ในการวิเคราะห์ และอีกรูปแบบคือ Non Language Features ได้แก่ ส่วนของคำสั้น (shortened words) ที่นิยมใช้กัน เช่น GUD แทนคำว่า good และ CRZ แทนคำว่า crazy เป็นต้น และสัญลักษณ์อิโมติคอนแสดงอารมณ์ ในงานวิจัยนี้ผู้วิจัยได้สร้างโมเดลเพื่อใช้ในการคำนวณหาคะแนน สำหรับการสกัดคุณลักษณะของข้อความและทำให้สามารถจำแนกประเภทของข้อความได้ ซึ่งผลการทดลองปรากฏว่าการที่นำคำสั้นและสัญลักษณ์อิโมติคอนแสดงอารมณ์มาใช้ในการวิเคราะห์ข้อความด้วย หรือรูปแบบ Non Language Features จะให้ประสิทธิภาพความถูกต้องได้ดีที่สุด

Sentiment Classification of Online Consumer Reviews using Word Vector Representation (Bansal & Srivastava, 2018)

ในปี ค.ศ. 2018 Bansal และ Srivastava ได้นำเสนอการจำแนกข้อความแสดงความคิดเห็นบนชุดข้อมูลของการรีวิวสินค้าจาก Amazon เกี่ยวกับโทรศัพท์มือถือ ผู้วิจัยได้ทำการทดลองการสกัดคุณลักษณะโดยการใช้เทคนิค word2vec เพื่อหาความสัมพันธ์เชิงความหมายของแต่ละคุณลักษณะโดยใช้โมเดล CBOW และโมเดล Skip-gram ในการสร้างเวกเตอร์แทนข้อความซึ่งจะมีจำนวนมิติของเวกเตอร์ขึ้นอยู่กับการเลือกจำนวนมิติของผู้วิจัย ถ้ามิติสูงจะให้ผลที่ดีกว่า ผู้วิจัยใช้ 400 มิติในการสร้างเวกเตอร์แทนข้อมูลได้ทำการเปรียบเทียบประสิทธิภาพโดยใช้ค่า Accuracy เป็นตัวประเมินผลความถูกต้องในการจำแนกของแต่ละคุณลักษณะและใช้ Support Vector Machine, Naive Bayes, Logistic Regression และ Random Forest สำหรับตัวจำแนกข้อความ ซึ่งผลการทดลองปรากฏว่าคุณลักษณะโดยใช้โมเดล CBOW ที่ใช้ตัวจำแนก Random Forest ได้ให้ประสิทธิภาพดีที่สุด

An Ensemble Classification System for Twitter Sentiment Analysis (Ankit & Saleena, 2018)

ในปี ค.ศ. 2018 Ankit และ Saleena ได้นำเสนอเทคนิควิธีเพื่อเพิ่มประสิทธิภาพค่าความถูกต้องในการจำแนกประเภทของข้อความจากทวิตเตอร์ คุณลักษณะที่ใช้แทนข้อความคือ Bag of Words ซึ่งจำนวนมิติของเวกเตอร์ขึ้นอยู่กับจำนวนของคำที่ปรากฏในชุดคำศัพท์ โดยใช้การจำแนกแบบร่วมกัน (Ensemble Classifier) ซึ่งผู้วิจัยได้คิดค้นวิธีการรวมผลการจำแนกจากวิธีการจำแนกแบบเดี่ยว (Single Classifier) 4 เทคนิค ได้แก่ Naive Bayes, Support Vector Machine, Logistic Regression และ Random Forest วิธีที่นำเสนอมีการคำนวณคะแนน ขั้วความคิดเห็นโดยให้ค่าน้ำหนักตามหลักความน่าจะเป็นซึ่งพิจารณาจากผลลัพธ์ของวิธีการจำแนกทั้ง 4 เทคนิค โดยได้ทำการทดลองจำแนกข้อความเปรียบเทียบกับวิธีการจำแนกแบบเดี่ยวและการจำแนกแบบร่วมกันที่ใช้การรวมผลด้วย Majority Voting ผลการทดลองปรากฏว่าวิธีการจำแนกประเภทข้อความโดยใช้ตัวจำแนกแบบร่วมกันที่มีการรวมผลตามที่ผู้วิจัยได้นำเสนอมีประสิทธิภาพค่าความถูกต้องดีที่สุด

งานวิจัยที่กล่าวมาข้างต้นนี้สามารถสรุปได้ดังตารางที่ 1.1

ตารางที่ 1.1 สรุปงานวิจัยที่เกี่ยวข้อง

ชื่อเรื่อง	ปี ค.ศ.	คุณลักษณะที่ให้ผลดีที่สุด	ตัวจำแนกที่ดีที่สุด	ประสิทธิภาพที่ดีที่สุด	ชุดข้อมูล
Text Sentiment Classification for SNS-based marketing using Domain Sentiment Dictionary	2012	คุณลักษณะแบบ TF-IDF	Support Vector Machine	ค่า F1 ของขั้วความคิดเห็นที่เป็น: Positive เท่ากับ 85.00, Negative เท่ากับ 63.00 และ Neutral เท่ากับ 66.00	ข้อมูลที่เป็นโดเมนเกี่ยวกับ person จากแหล่งข้อมูล ได้แก่ webpage, personal blog, twitter, facebook และ me2day
Tweet Sentiment Analysis with Classifier Ensemble	2014	คุณลักษณะแบบ Bag of Words ที่ใช้ opinion lexicon ร่วมด้วย	ตัวจำแนกแบบร่วมกัน ของ Naive Bayes, Support Vector Machine และ Random Forest	ค่า Accuracy เท่ากับ 84.89	ข้อมูลชุด Sander เป็นข้อมูลเกี่ยวกับคำที่ค้นหาจาก twitter ได้แก่ @apple, #google, #microsoft และ #twitter
Comparison of Text Sentiment Analysis based on Machine Learning	2016	คุณลักษณะแบบ TF-IDF	Extreme Learning Machine with Kernels	ค่า Accuracy เท่ากับ 88.74	ข้อมูลที่เป็นข้อความภาษาจีน
Classification of Sentiment Reviews using n-gram Machine Learning Approach	2016	คุณลักษณะแบบ TF-IDF ที่สร้างโดยใช้เทคนิค bi-gram	Stochastic Gradient Descent	ค่า Accuracy เท่ากับ 95.00	ข้อมูลการรีวิวภาพยนตร์จาก IMDb

ตารางที่ 1.1 สรุปงานวิจัยที่เกี่ยวข้อง (ต่อ)

ชื่อเรื่อง	ปี ค.ศ.	คุณลักษณะที่ให้ผลดีที่สุด	ตัวจำแนกที่ดีที่สุด	ประสิทธิภาพที่ดีที่สุด	ชุดข้อมูล
Negation Handling in Sentiment Analysis at Sentence	2017	คุณลักษณะแบบ POS tagging	โมเดลที่ใช้การคำนวณผลลัพธ์จากค่าคะแนน strength	ค่า Accuracy เท่ากับ 83.3	ข้อมูลการรีวิวจาก Amazon, Ebay และ Cnet
Sentiment Classification of Tweets with Non-Language Features	2018	คุณลักษณะแบบ POS tagger ที่ใช้คุณลักษณะ Non-Language ร่วมด้วย	โมเดล Sentiment Scoring (SS)	ค่า Overall Accuracy เท่ากับ 84.00	ชุดข้อมูล Shapdeal เป็นข้อมูลจาก twitter
Sentiment Classification of Online Consumer Reviews using Word Vector Representation	2018	คุณลักษณะแบบ Continuous bag of Words	Random Forest	ค่า Accuracy เท่ากับ 90.66	ข้อมูลการรีวิวสินค้าจาก Amazon เกี่ยวกับโทรศัพท์มือถือ
An Ensemble Classification System for Twitter Sentiment Analysis	2018	คุณลักษณะแบบ Bag of Words	ตัวจำแนกแบบร่วมกัน ของ Naive Bayes, Support Vector Machine, Logistic Regression และ Random Forest	ค่าเฉลี่ยของ F1 เท่ากับ 76.85	ชุดข้อมูล First Gop debate เป็นข้อมูลจาก Crowflower

1.5 ขอบเขตงานวิจัย

1) เสนอการสกัดคุณลักษณะข้อความที่มีประสิทธิภาพเพื่อจำแนกขั้วความคิดเห็น
2) เปรียบเทียบประสิทธิภาพการสกัดคุณลักษณะที่คิดค้นขึ้นกับการสกัดคุณลักษณะแบบดั้งเดิม เช่น TF-IDF บนชุดคำศัพท์จากข้อมูลสอน

1.6 ขั้นตอนการดำเนินการ

1) ศึกษางานวิจัยและทฤษฎีที่เกี่ยวข้องสำหรับการสกัดคุณลักษณะข้อความและการจำแนกขั้วความคิดเห็น
2) วิเคราะห์และเสนอแนวคิดการสกัดคุณลักษณะข้อความที่มีประสิทธิภาพเพื่อจำแนกขั้วความคิดเห็น
3) ศึกษาเทคโนโลยีและเครื่องมือสนับสนุนสำหรับงานวิจัยการสกัดคุณลักษณะข้อความ
4) คิดค้นการสกัดคุณลักษณะข้อความที่มีประสิทธิภาพเพื่อจำแนกขั้วความคิดเห็น
5) เตรียมชุดข้อมูลและพจนานุกรมสำหรับงานวิจัยการสกัดคุณลักษณะข้อความ
6) เขียนโปรแกรมสกัดคุณลักษณะข้อความ
7) เขียนโปรแกรมสร้างเวกเตอร์แทนข้อความ
8) เขียนโปรแกรมเพื่อทำการทดลองสร้างโมเดลจำแนกขั้วความคิดเห็นโดยใช้คุณลักษณะที่คิดขึ้นและคุณลักษณะที่เคยมีมา
9) ทดสอบประสิทธิภาพความถูกต้องของโมเดลแต่ละแบบ
10) สรุปผลการทดลอง
11) จัดทำเอกสารและเขียนบทความวิจัยเพื่อเผยแพร่

1.7 ระยะเวลาการดำเนินการ

มกราคม 2561 – เมษายน 2563

1.8 แผนการดำเนินการ

แผนการดำเนินการวิจัยสามารถแสดงได้ดังตารางที่ 1.2

ตารางที่ 1.2 ระยะเวลาการดำเนินงานวิจัย

กิจกรรมดำเนินการ	เดือน																											
	2561												2562												2563			
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
1																												
2																												
3																												
4																												
5																												
6																												
7																												
8																												
9																												
10																												
11																												

หมายเหตุ รายละเอียดกิจกรรมดำเนินการอยู่ในหัวข้อ 1.6

1.9 สถานที่ดำเนินงาน

ห้องปฏิบัติการ CS 207 ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

1.10 เครื่องมือที่ใช้ในการดำเนินงานวิจัย

1) **ด้านซอฟต์แวร์**
   - ระบบปฏิบัติการ Microsoft Windows 10 Pro
   - โปรแกรม R Studio
2) **ด้านฮาร์ดแวร์**
   - เครื่องคอมพิวเตอร์จำนวน 1 เครื่อง มีคุณสมบัติดังนี้
     - System Manufacturer: Dell
     - Processor: Intel(R) Core(TM) i5-4590 CPU @3.30GHz 3.30 GHz
     - Memory (RAM): 4.00 GB

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 การประมวลผลข้อความ

ข้อความโดยทั่วไปนั้น จะประกอบไปด้วย ตัวเลข สัญลักษณ์พิเศษ และคำศัพท์ ซึ่งตัวเลข สัญลักษณ์พิเศษ และบางคำศัพท์นั้นอาจจะไม่บ่งบอกความหมายของข้อความ จึงสามารถกำจัดออกจากข้อความไปได้ โดยกระบวนการจัดเตรียมข้อความ ประกอบด้วย การทำความสะอาดข้อความ การกำจัดคำหยุด การลดรูปคำ มีรายละเอียดดังต่อไปนี้

2.1.1 การทำความสะอาดข้อความและการตัดแบ่งคำ

การทำความสะอาดข้อความและการตัดแบ่ง (Manning et al., 2014) ประกอบด้วยกระบวนการดังต่อไปนี้

1) แปลงตัวอักษรให้ในข้อความเป็นพิมพ์เล็ก
2) กำจัดตัวเลข
3) กำจัดสัญลักษณ์พิเศษ
4) ตัดแบ่งคำโดยใช้ช่องว่าง

ตัวอย่างการทำความสะอาดข้อความ แสดงดังรูปที่ 2.1

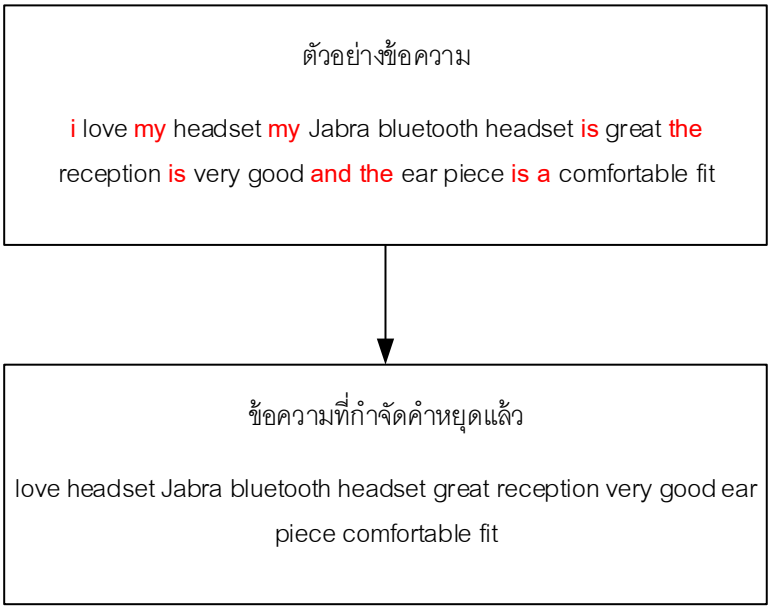
รูปที่ 2.1 ตัวอย่างการทำความสะอาดข้อความ

2.1.2 การกำจัดคำหยุด

ในบางครั้ง คำศัพท์ที่ปรากฏขึ้นบ่อยแทบจะไม่มีผลกับการค้นหาเอกสารที่เกี่ยวข้องซึ่งคำเหล่านี้เรียกว่า คำหยุด (stopwords) ตัวอย่างรายการคำหยุดแสดงดังรูปที่ 2.2 โดยหลักการทั่วไปสำหรับการสร้างรายการคำหยุด ทำได้โดยการเรียงคำศัพท์ของคำตามจำนวนครั้งของคำศัพท์ที่ปรากฏอยู่ในเอกสารทั้งหมด แล้วเลือกคำศัพท์ที่มีความถี่สูงอันดับแรกๆ มาอยู่ในรายการคำหยุด หรืออาจทำโดยการสร้างขึ้นเองตามโดเมนของเอกสารที่พิจารณา (Manning et al., 2014) ซึ่งในปัจจุบันมีการสร้างรายการคำหยุดมาตรฐานให้สามารถนำมาใช้ได้ ตัวอย่างเช่น ชุดรายการคำหยุดจากเว็บไซต์ https://www.ranks.nl/stopwords ตัวอย่างการกำจัดคำหยุดในข้อความโดยใช้รายการคำหยุดมาตรฐาน แสดงดังรูปที่ 2.3

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

รูปที่ 2.2 ตัวอย่างรายการคำหยุด



รูปที่ 2.3 ตัวอย่างการกำจัดคำหยุด

2.1.3 การลดรูปคำ

ด้วยเหตุผลทางไวยากรณ์ที่แต่ละเอกสารจะมีการใช้รูปแบบคำศัพท์ที่แตกต่างกัน เช่น organize organizes และ organizing นอกจากนี้ยังมีคำศัพท์ที่มีความหมายคล้ายกัน เช่น democracy democratic และ democratization ในหลายๆ ครั้งการค้นหาเอกสารด้วยคำศัพท์คำหนึ่ง เราอาจได้เอกสารที่มีคำศัพท์สมนัยกับคำศัพท์ที่เราค้นหาแต่ไม่มีคำศัพท์ที่เราค้นหาโดยตรงก็เป็นได้ ดังนั้นจึงต้องมีการลดรูปคำที่สมนัยกันให้อยู่ในรูปแบบเดียวกัน เรียกว่า การลดรูปคำ (stemming) จุดประสงค์ของการลดรูปคำคือการลดรูปแบบของการผันคำศัพท์และรูปแบบที่เกี่ยวข้องกับรากศัพท์ของคำศัพท์เพื่อแปลงเป็นรูปแบบพื้นฐานที่ยังไม่ไม่มีการผันรูป

สำหรับในทางการเขียนโปรแกรมคอมพิวเตอร์ส่วนใหญ่การทำการลดรูปคำจะเป็นกระบวนการที่ตัดส่วนท้ายของคำศัพท์ที่มีการผันรูปอยู่ในรูปแบบต่างๆ จนได้เป็นคำศัพท์ที่มีการลดรูปเป็นรากศัพท์เดียวกัน ตัวอย่างการลดรูปคำศัพท์แสดงดังรูปที่ 2.4 และตัวอย่างการลดรูปคำศัพท์ในข้อความแสดงดังรูปที่ 2.5

car, cars, car's, cars' ⇒ car

รูปที่ 2.4 ตัวอย่างการลดรูปคำศัพท์

the boy's cars are different colors ⇒ the boy car are differ color

รูปที่ 2.5 ตัวอย่างการลดรูปคำศัพท์ในข้อความ

อัลกอริทึมที่มีการใช้บ่อยที่สุดสำหรับการทำการลดรูปคำในภาษาอังกฤษ และเป็นหนึ่งในอัลกอริทึมที่มีการยอมรับว่ามีประสิทธิภาพ คือ Porter's algorithm (Van Rijsbergen et al., 1980) ซึ่ง Martin Porter ได้คิดค้นเมื่อ ค.ศ. 1980 ซึ่งกฎการลดรูปที่อยู่ใน Porter's algorithm เรียกว่า Potter stemmer

สำหรับการเลือกใช้กฎในการลดรูปคำหากมีหลายกฎที่ตรงกับเงื่อนไขจะเลือกใช้กฎที่มีความยาวส่วนท้ายมากที่สุดก่อน ตัวอย่างกฎดังรูปที่ 2.6 สมมติว่าพิจารณาการลดรูปคำศัพท์ caresses จะเห็นว่าความยาวของตัวอักษรในส่วนท้ายของคำศัพท์สอดคล้องกับกฎข้อที่ 1 คือลงท้ายด้วย SSES

และ สอดคล้องกับกฎข้อที่ 4 คือลงท้ายด้วย S นั่นแสดงว่าจะต้องใช้กฎข้อที่ 1 เพราะมีความยาวของตัวอักษรในส่วนท้ายมากกว่า จึงได้ว่า caresses ถูกลดรูปเป็น caress เป็นต้น

Rule			**Example**		
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
SS	→	SS	caress	→	caress
S	→		cats	→	cat

รูปที่ 2.6 ตัวอย่างกลุ่มของกฎขั้นตอนที่ 1

กฎอีกหลายข้อจะใช้แนวคิดการตรวจสอบรูปแบบและความยาวตัวอักษรของคำศัพท์ทั้งส่วนท้าย (suffixes) และส่วนหน้า (prefixes) ที่ตรงกับเงื่อนไข โดยในส่วนหน้าจะเป็นการตรวจสอบรูปแบบและความยาวของตัวอักษรประกอบการใช้กฎด้วย นอกจากนี้หลังจากการลดรูปคำศัพท์แล้วคำศัพท์ผลลัพธ์ที่ได้จะต้องมีความยาวมากกว่าหนึ่งตัวอักษรเสมอ ตัวอย่างกฎแสดงในรูปที่ 2.7 เมื่อนำมาใช้ลดรูปคำศัพท์ replacement จะได้เป็น replac แต่สำหรับคำศัพท์ cement ไม่สามารถลดรูปได้เป็น c เพราะเมื่อลดรูปแล้วเหลือเพียงตัวอักษรเดียว เป็นต้น

(m > 1) EMEMT →

รูปที่ 2.7 ตัวอย่างกฎที่มีการตรวจสอบความยาวส่วนหน้าของคำศัพท์

นอกจากการลดรูปคำโดยใช้ Porter stemmer ยังมีได้หลายวิธี เช่น Lovins stemmer (Lovins, 1968) และ Paice/Husk stemmer (Paice, 1990) โดยแต่ละวิธีจะมีขั้นตอนการลดรูปคำศัพท์ที่แตกต่างกัน ตัวอย่างผลลัพธ์แต่ละวิธีแสดงดังรูปที่ 2.8

Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

รูปที่ 2.8 ตัวอย่างผลการลดรูปศัพท์ในข้อความ เมื่อใช้วิธีที่ต่างกัน (Manning et al., 2014)

2.2 การสกัดคุณลักษณะแทนข้อความ

หลังจากที่ผ่านกระบวนการทำความสะอาดข้อความแล้ว จะต้องมีการสกัดคุณลักษณะที่เป็นตัวแทนของข้อความก่อนที่จะนำไปวิเคราะห์ โดยส่วนใหญ่แล้วจะแทนด้วยเวกเตอร์ที่มีจำนวนมิติเท่ากับจำนวนคำศัพท์ที่อยู่ในพจนานุกรม

พจนานุกรมเป็นเซตของคำที่จะนำมาใช้ในการสร้างเวกเตอร์แทนข้อความ โดยพจนานุกรมจะเป็นสิ่งกำหนดจำนวนมิติของเวกเตอร์แทนข้อความ การสร้างพจนานุกรมมีได้หลายวิธี เช่น

- การใช้คำศัพท์จากข้อความที่จะนำมาวิเคราะห์ซึ่งผ่านกระบวนการทำความสะอาด การกำจัดคำหยุด และการลดรูปคำข้างต้น
- การกำหนดคำศัพท์ในพจนานุกรมขึ้นมาเองให้เหมาะกับลักษณะของข้อความที่กำลังวิเคราะห์
- การใช้ชุดคำศัพท์จากชุดข้อมูลมาตรฐาน เช่น คำศัพท์ที่เกี่ยวกับทางการแพทย์ คำศัพท์ที่เกี่ยวกับทิศทางการแสดงความคิดเห็น เป็นต้น

อย่างไรก็ตามการเลือกใช้พจนานุกรมมีความสำคัญในการวิเคราะห์ หากว่าพจนานุกรมประกอบด้วยคำศัพท์มากเกินไปจะทำให้จำนวนมิติของเวกเตอร์มากและส่งผลให้ประมวลผลช้า แต่หากพจนานุกรมประกอบด้วยคำศัพท์น้อยเกินไปอาจส่งผลให้ผลการวิเคราะห์ข้อมูลคลาดเคลื่อน

เวกเตอร์แทนข้อความประกอบด้วยสมาชิกในแต่ละมิติที่แทนด้วยค่าน้ำหนักของความเกี่ยวข้องของเอกสารและเซตของคำศัพท์ที่อยู่ในพจนานุกรม การแทนข้อความในลักษณะเวกเตอร์นี้เรียกว่า Bag of Words Model (BOW) หรือ Vector Space Model (VSM) วิธีพื้นฐานที่เป็นที่นิยม ได้แก่ Binary vector (Bv), Term Frequency (TF) และ Term Frequency-Inverse Document Frequency (TF-IDF) โดยจะกล่าวรายละเอียดของแต่ละวิธีดังต่อไปนี้

2.2.1 การแทนเอกสารแบบ Binary vector

การสร้างเวกเตอร์แทนข้อความแบบ Binary vector โดยแต่ละมิติของเวกเตอร์จะสัมพันธ์กันกับคำศัพท์แต่ละคำ พิจารณาการให้ค่าน้ำหนักกับค่าศัพท์แต่ละคำตามการปรากฏคำศัพท์นั้นในข้อความ ถ้าคำศัพท์ไม่ปรากฏในข้อความจะมีค่าน้ำหนักคำศัพท์นั้นเป็น 0 ในทางกลับกันคำศัพท์ที่ปรากฏในข้อความจะมีค่าน้ำหนักสำหรับคำศัพท์นั้นเป็น 1 กำหนดให้ DocX แทนข้อความฉบับที่ x สมมติให้มีข้อความทั้งหมด 6 ฉบับและพจนานุกรมประกอบด้วยคำศัพท์ 5 คำ ได้แก่ ตัวอย่างแสดงการแทนข้อความแบบเวกเตอร์ไบนารีแสดงดังตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างการแทนเอกสารแบบเวกเตอร์ Binary vector

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6
angry	1	1	0	0	0	1
crazy	1	1	0	1	0	0
dislike	1	1	0	1	1	0
happy	0	1	0	0	0	0
joy	1	0	1	1	1	1

จากตารางที่ 2.1 จะอธิบายได้ว่า ตัวอย่างการแทนเอกสาร $\mathrm{Doc\,1}$ มีคำศัพท์ angry crazy dislike และ joy ปรากฏในเอกสาร แต่คำศัพท์ happy ไม่ปรากฏในเอกสาร ดังนั้นเวกเตอร์แทนเอกสาร $\mathrm{Doc\,1}$ ที่เป็นเวกเตอร์ 5 มิติคือ $[\,1\,1\,1\,0\,1\,]^T$

2.2.2 การแทนเอกสารแบบ Term Frequency

การสร้างเวกเตอร์แทนข้อความแบบ Term Frequency (TF) (Manning et al., 2014) จะพิจารณาค่าน้ำหนักของแต่ละคำศัพท์ที่ค้นหาขึ้นอยู่กับว่าคำศัพท์นั้นปรากฏในข้อความมากน้อยเพียงใด ถ้าคำศัพท์ปรากฏบ่อยครั้งในข้อความ ค่าของเวกเตอร์ในมิติที่ตรงคำศัพท์นั้นควรจะได้ค่าน้ำหนักที่สูงกว่าคำศัพท์ที่ปรากฏน้อย ซึ่งจะสามารถทำการสร้างเวกเตอร์โดยการกำหนดค่าน้ำหนักให้เท่ากับจำนวนครั้งการปรากฏคำศัพท์ในข้อความ ซึ่งรูปแบบการให้ค่าน้ำหนักนี้จะเรียกว่า ความถี่ของคำศัพท์ แทนด้วย $\mathrm{TF}_{t,d}$ ตัวห้อยแสดงถึงคำศัพท์และข้อความตามลำดับ ตัวอย่างแสดงการแทนข้อความแบบ Term Frequency แสดงดังตารางที่ 2.2

ตารางที่ 2.2 ตัวอย่างการแทนเอกสารแบบ Term Frequency

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6
angry	157	73	0	0	0	1
crazy	4	157	0	2	0	0
dislike	232	227	0	2	1	0
happy	0	10	0	0	0	0
joy	2	0	3	8	5	8

จากตารางที่ 2.2 จะอธิบายได้ว่าพจนานุกรมประกอบด้วย คำศัพท์ 5 คำ จึงสร้างคุณลักษณะแทนข้อความด้วยเวกเตอร์ 5 มิติ ตัวอย่างการแทนข้อความ Doc1 มีคำศัพท์ทั้งหมด 5 คำ ซึ่งคำศัพท์ angry ปรากฏ 157 ครั้ง คำศัพท์ crazy ปรากฏ 4 ครั้ง คำศัพท์ dislike ปรากฏ 232 ครั้ง คำศัพท์ happy ปรากฏ 0 ครั้ง และคำศัพท์ joy ปรากฏ 2 ครั้ง ดังนั้นเวกเตอร์แทนข้อความ Doc1 ที่เป็นเวกเตอร์ 5 มิติคือ $\left[157\ 4\ 232\ 0\ 2\right]^T$

2.2.3 การแทนเอกสารแบบ Term Frequency-Inverse Document Frequency

การแทนข้อความแบบ Term Frequency ที่ได้กล่าวมาข้างต้นนั้นยังคงมีปัญหาสำคัญเนื่องจากทุกคำศัพท์ที่ค้นหาจะมีความสำคัญของความเกี่ยวข้องสำหรับการสอบถามเท่ากันหมด แต่ในความเป็นจริงแล้วคำศัพท์บางคำอาจจะมีความเกี่ยวข้องเพียงเล็กน้อยเท่านั้น ตัวอย่างเช่น ชุดข้อมูลข้อความทางอุตสาหกรรมยานยนต์ซึ่งจะมีคำศัพท์ auto ปรากฏเกือบทุกข้อความ ด้วยเหตุนี้จึงต้องมีวิธีในการลดผลกระทบของความเกี่ยวข้องของคำศัพท์ที่ปรากฏขึ้นบ่อยในชุดข้อความที่มีอยู่ทั้งหมด แนวทางหนึ่งในการลดความเกี่ยวข้องของคำศัพท์คือการลดค่าน้ำหนักของคำศัพท์ที่มี Collection Frequency (CF) สูง ซึ่ง Collection Frequency คือความถี่ของคำศัพท์ที่ปรากฏในชุด

ข้อความที่มีอยู่ทั้งหมด โดยแนวคิดนี้เป็นการลดค่าน้ำหนัก Term Frequency ของคำศัพท์ตามการเพิ่มขึ้นใน Collection Frequency

อีกแนวทางหนึ่งคือการใช้วิธี Document Frequency (DF) แทนด้วย DF_t ซึ่งคือจำนวนเอกสารที่คำศัพท์ t ปรากฏอยู่จากชุดเอกสารที่มีอยู่ทั้งหมด ในการให้ค่าน้ำหนักเพื่อระบุความแตกต่างของเอกสารการใช้จำนวนเอกสารที่ปรากฏคำศัพท์ที่ค้นหาจะดีกว่าการใช้จำนวนคำศัพท์ที่ค้นหาในชุดเอกสารทั้งหมด ตัวอย่างรูปที่ 2.9 แสดงความแตกต่างของ Collection Frequency และ Document Frequency จะเห็นว่าค่า Collection Frequency สำหรับคำศัพท์ try และ insurance จะมีค่าที่พอๆ กัน แต่ Document Frequency จะมีค่าที่แตกต่างกันอย่างเห็นได้ชัด นั่นหมายความว่ามีแค่บางเอกสารที่มีคำศัพท์ insurance ปรากฏอยู่เมื่อเทียบกับคำศัพท์ try แม้ว่าความถี่ในเอกสารทั้งหมดจะพอๆ กัน แต่คำศัพท์ insurance จัดเป็นคำศัพท์หายาก จึงต้องมีค่าน้ำหนักที่สูงกว่า try

Word	CF	DF
try	10422	8760
insurance	10440	3997

รูปที่ 2.9 ตัวอย่าง Collection Frequency และ Document Frequency ในชุดข้อมูลเอกสาร Reuters (Manning et al., 2014)

จากที่กล่าวมาข้างต้นค่าน้ำหนัก Document Frequency หรือ DF แสดงให้เห็นจำนวนเอกสารที่ปรากฏคำศัพท์ที่พิจารณา สมมติให้เป็นคำศัพท์ t ในชุดเอกสารที่มีอยู่ทั้งหมด N ฉบับ ถ้า Document Frequency ของคำศัพท์มีค่าสูงมากหมายความว่ามีความสำคัญน้อยแต่ถ้า Document Frequency ของคำศัพท์มีค่าน้อยแสดงว่ามีความสำคัญมากเพราะเป็นคำศัพท์หายาก จึงต้องมีการกำหนดส่วนกลับของ Document Frequency เรียกว่า Inverse Document Frequency (IDF) ของคำศัพท์ t ดังนี้

$$\mathrm{IDF}_t = \log_{10}\left(\frac{N}{\mathrm{DF}_t}\right) \tag{2.1}$$

ดังนั้น Inverse Document Frequency ของคำศัพท์ที่หายากจะมีค่าสูง ในขณะที่ Inverse Document Frequency ของคำศัพท์ที่พบบ่อยจะมีค่าน้อย รูปที่ 2.10 แสดงตัวอย่างของ Inverse Document Frequency ในชุดข้อมูลเอกสาร Reuters ที่มีเอกสารทั้งหมด 806,791 ฉบับ

คำศัพท์(term)	DF_t	IDF_t
car	18,165	1.65
auto	6,723	2.08
insurance	19,241	1.62
best	25,235	1.5

รูปที่ 2.10 ตัวอย่างของค่า Inverse Document Frequency ในชุดข้อมูลเอกสาร Reuters

อธิบายความเกี่ยวข้องของเอกสารจากการคำนวณด้วยวิธี Inverse Document Frequency ดังนี้

1) Inverse Document Frequency มีค่าสูงสุด เมื่อคำศัพท์ t ปรากฏในเอกสารจำนวนเล็กน้อย ดังนั้นเอกสารเหล่านั้นจึงมีค่าน้ำหนักความเกี่ยวข้องที่สูงกับคำศัพท์ t
2) Inverse Document Frequency มีค่าน้อย เมื่อคำศัพท์นั้นปรากฏในเอกสารจำนวนมาก ดังนั้นเอกสารเหล่านั้นจะมีค่าน้ำหนักความเกี่ยวข้องน้อยลงกับคำศัพท์ t
3) Inverse Document Frequency มีค่าต่ำสุด เมื่อคำศัพท์นั้นปรากฏในเอกสารแทบทั้งหมด ดังนั้นเอกสารเหล่านั้นแทบจะไม่มีค่าน้ำหนักความเกี่ยวข้องเลย

ค่า Term Frequency (TF) และ Inverse Document Frequency (IDF) บ่งบอกถึงค่าน้ำหนักของคำศัพท์และนำมาใช้เพื่อสร้างค่าน้ำหนักสำหรับแต่ละคำศัพท์ในแต่ละเอกสาร เรียกว่าค่าน้ำหนัก TF-IDF แสดงดังสมการ

$$\text{TF-IDF}_{t,d} = \text{TF}_{t,d} \times \text{IDF}_t \qquad (2.2)$$

2.3 การจำแนกประเภทข้อมูล

การจำแนก (Classification) เป็นเทคนิคที่สำคัญในศาสตร์การวิเคราะห์ข้อมูล (Data Analytics) การรู้จำรูปแบบ (Pattern recognition) และการเรียนรู้ของเครื่อง (Machine Learning) (Singh et al., 2016) การจำแนกจัดเป็นเทคนิคการเรียนรู้แบบมีผู้สอน เพราะว่าเป็นการจำแนกข้อมูลโดยอาศัยข้อมูลที่ทราบมาก่อนหน้า การทำนายประเภทหรือคลาสของข้อมูลทดสอบแต่ละตัวจะประกอบด้วยการสกัดคุณลักษณะและการตรวจสอบว่าตรงกับรูปแบบของข้อมูลสอนในคลาสใด การจำแนกประกอบด้วย 2 กระบวนการหลัก คือ กระบวนการซึ่งดำเนินการกับข้อมูลสอนจนได้โมเดล และกระบวนการนำโมเดลที่ได้มาทำการตรวจสอบโดยใช้ข้อมูลทดสอบเพื่อวัดประสิทธิภาพ

ของโมเดล การจำแนกได้ถูกนำไปประยุกต์หลายด้าน เช่น การจำแนกเอกสาร (Mishu & Rafiuddin, 2016) การกรองสแปม (Chae et al., 2017) การจำแนกภาพ (Guo et al., 2017) การตรวจจับการฉ้อโกง (Jurgovsky et al., 2018) การวิเคราะห์ความเสี่ยง (Shaw & Gentry, 1990) เป็นต้น เทคนิคการจำแนกที่นิยมใช้กันอย่างแพร่หลายได้แก่ Naive Bayes, k-Nearest Neighbors, Support Vector Machine และ Artificial Neural Networks เป็นต้น มีรายละเอียดดังที่จะกล่าวต่อไปนี้

2.3.1 การจำแนกประเภทข้อมูลด้วย Naive Bayes

ก่อนที่จะอธิบายว่าทฤษฎีบทของเบย์ใช้สำหรับการจำแนกประเภท กำหนดให้ $\mathbf{X}$ แทนชุดคุณลักษณะของข้อมูลและ Y แทนประเภทหรือคลาสของข้อมูล สำหรับ $\mathbf{X}$ และ Y ใดๆ จะหาความสัมพันธ์ในลักษณะของความน่าจะเป็นที่ $\mathbf{X}$ จะมีประเภทเป็น Y ซึ่งเขียนแทนด้วย $P(Y|\mathbf{X})$ ความน่าจะเป็นแบบมีเงื่อนไขนี้เรียกอีกอย่างว่า posterior probability ในขณะที่ $P(Y)$ แทน prior probability ซึ่งคือความน่าจะเป็นของที่ข้อมูลทั้งหมดจะเป็นคลาส Y ก่อนที่จะมีการกำหนดเงื่อนไขข้อมูล

ในกระบวนการสอนต้องการที่จะได้ posterior probability $P(Y|\mathbf{X})$ สำหรับทุก $\mathbf{X}$ และ Y ที่เป็นไปได้บนพื้นฐานของชุดข้อมูลสอนที่มีอยู่ โดยให้ข้อมูลทดสอบเขียนแทนด้วย $\mathbf{X}'$ สามารถถูกจำแนกได้โดยการหาคลาส Y' ที่ให้ค่า $P(Y'|\mathbf{X}')$ มากที่สุด พิจารณาได้ดังตัวอย่างการทำนายการผิดนัดชำระเงินของผู้กู้สินเชื่อ แสดงดังรูปที่ 2.11 จากตัวอย่างจะเห็นว่าคุณลักษณะของชุดข้อมูลสอนประกอบด้วย การเป็นเจ้าของบ้าน (Home Owner), สถานภาพการสมรส (Marital Status) และรายได้ประจำปี (Annual Income) เมื่อผู้กู้ยืมสินเชื่อที่ผิดนัดชำระเงินจะถูกจัดอยู่ในประเภทคลาส $\mathbf{Yes}$ ในขณะที่ผู้กู้ยืมสินเชื่อที่ชำระเงินภายในเวลาที่กำหนดจะถูกจัดอยู่ในประเภท $\mathbf{No}$

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

รูปที่ 2.11 ตัวอย่างชุดข้อมูลสอนสำหรับการทำนายปัญหาการผิดนัดชำระหนี้ (Tan et al., 2019)

สมมติข้อมูลทดสอบคือ

$$\mathbf{X} = (\text{Home Owner} = \text{No}, \text{Marital Status} = \text{Married}, \text{Annual Income} = 120\text{K})$$

ในการที่จะจำแนกประเภทของข้อมูลทดสอบ $\mathbf{X}$ ต้องหาค่าความน่าจะเป็น $P(\text{Yes}|\mathbf{X})$ และ $P(\text{No}|\mathbf{X})$ โดยพิจารณาจากข้อมูลสอน ถ้าหาก $P(\text{Yes}|\mathbf{X})$ มีค่ามากกว่า $P(\text{No}|\mathbf{X})$ แล้ว ข้อมูลนั้นจะถูกจำแนกประเภทให้เป็นคลาส Yes หรือมิฉะนั้นข้อมูลจะถูกจำแนกประเภทให้เป็นคลาส No

การหาค่า posterior probability ให้มีความถูกต้องสำหรับทุกคลาสและค่าคุณลักษณะที่เป็นไปได้เป็นสิ่งที่ยากเนื่องจากจะต้องมีชุดข้อมูลการสอนที่มีขนาดใหญ่ ทฤษฎีของเบย์ จึงถูกนำมาใช้ในการหาค่าความน่าจะเป็นแบบมีเงื่อนไขดังกล่าวเพราะว่าทฤษฎีของเบย์ จะแสดงค่า posterior probability $P(Y|\mathbf{X})$ ในรูปของ $P(Y)$, $P(\mathbf{X}|Y)$ และ $P(\mathbf{X})$ ดังสมการ (2.3)

$$P(Y|\mathbf{X}) = \frac{P(\mathbf{X}|Y) \times P(Y)}{P(\mathbf{X})} \tag{2.3}$$

ในการคำนวณ $P(Y|\mathbf{X})$ สำหรับ Y ที่แตกต่างกัน จะเห็นได้ว่า $P(\mathbf{X})$ จะใช้ค่าเดียวกันเสมอ ดังนั้นในกรณีที่เราต้องการเปรียบเทียบค่า $P(Y|\mathbf{X})$ จึงสามารถละเว้นการหาค่า $P(\mathbf{X})$ ได้ ส่วน $P(Y)$ สามารถคำนวณได้โดยตรงจากชุดข้อมูลสอนด้วยการคำนวณจากสัดส่วนของชุดข้อมูลสอนที่อยู่ในคลาส Y

การจำแนกประเภทโดยอาศัยการเรียนรู้แบบ Naive Bayes เป็นการประมาณความน่าจะ-เป็นแบบมีเงื่อนไขโดยสมมติให้ชุดคุณลักษณะเป็นอิสระต่อกันตามเงื่อนไขเมื่อมีประเภทคลาสเป็น $Y = y$ จากสมมติฐานดังกล่าวนี้ จะได้ว่า

$$P\left(\mathbf{X}|Y = y\right) = \prod_{i=1}^{d} P\left(X_i|Y = y\right) \tag{2.4}$$

โดยที่แต่ละค่าคุณลักษณะในเซต $\mathbf{X} = \left\{X_1, X_2, \ldots, X_d\right\}$ ประกอบด้วย d คุณลักษณะ

ด้วยสมมติฐานความอิสระตามเงื่อนไข การคำนวณความน่าจะเป็นแบบมีเงื่อนไข สามารถคำนวณหาค่าความน่าจะเป็นสำหรับแต่ละคลาส Y เมื่อกำหนดเงื่อนไขคุณลักษณะ $\mathbf{X}$ ได้ โดยแทนสมการ (2.4) ใน (2.3) จะได้

$$P\left(Y|\mathbf{X}\right) = \frac{P(Y)\prod_{i=1}^{d} P\left(X_i|Y\right)}{P\left(\mathbf{X}\right)} \tag{2.5}$$

เนื่องจาก $P\left(\mathbf{X}\right)$ เป็นค่าเดียวกันสำหรับทุก Y จึงพิจารณาให้เป็น $P(\mathbf{X})$ เป็นค่าคงที่ที่ไม่เท่ากับศูนย์ได้ และจะเลือกคลาสที่มีค่ามากที่สุดจากการคำนวณของ $P(Y)\prod_{i=1}^{d} P\left(X_i|Y\right)$ หรือกล่าวได้ว่า $P\left(Y|\mathbf{X}\right)$ แปรผันตรงกับ $P(Y)\prod_{i=1}^{d} P\left(X_i|Y\right)$

$$P\left(Y|\mathbf{X}\right) \propto P(Y)\prod_{i=1}^{d} P\left(X_i|Y\right) \tag{2.6}$$

จากชุดข้อมูลสอนในรูปที่ 2.11 นำมาสร้างแบบจำลองด้วยการเรียนรู้แบบ Naive Bayes ได้ดังรูปที่ 2.12

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(ก)

P(Yes) = 3/10
P(No) = 7/10

P(Home Owner=Yes|No) = 3/7
P(Home Owner=No|No) = 4/7
P(Home Owner=Yes|Yes) = 0
P(Home Owner=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No) = 1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/3
P(Marital Status=Divorced|Yes) = 1/3
P(Marital Status=Married|Yes) = 0

For Annual Income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

(ข)

รูปที่ 2.12 ตัวจำแนก Naive Bayes สำหรับปัญหาการจำแนกประเภทการชำระเงินของผู้กู้ยืมสินเชื่อ

จากรูปที่ 2.12 จะได้ว่าคุณลักษณะ $\mathbf{X}$ ประกอบด้วย 3 แอทริบิวต์ ได่แก่ Home Owner, Marital Status และ Annual Income ส่วนคลาสหรือ Y ประเภท คือ แอทริบิวต์ Defaulted Borrower พิจารณาการคำนวณเพื่อการจำแนกได้ดังตัวอย่างต่อไปนี้

- การหาค่า $P(\text{Home Owner} = \text{Yes} | \text{No})$ ทำได้โดยพิจารณาว่ามี 7 คนเป็นผู้กู้ยืมสินเชื่อที่ชำระเงินภายในเวลาที่กำหนด $(\text{Defaulted Borrower} = \text{No}$ หรือ $Y = \text{No})$ และใน 7 คนนี้เป็นผู้มีบ้านเป็นของตนเอง $(\text{Home Owner} = \text{Yes})$ อยู่ 3 คน จึงได้ว่า $P(\text{Home Owner} = \text{Yes} | \text{No}) = 3/7$
- การหาค่า $P(\text{Marital Status} = \text{Single} | \text{Yes})$ ทำได้โดยพิจารณาว่ามี 3 คนเป็นผู้กู้ยืมสินเชื่อที่ผิดนัดชำระเงิน $(\text{Defaulted Borrower} = \text{Yes}$ หรือ $Y = \text{Yes})$ และใน 3 คนนี้เป็นคนโสด (Marital Status) อยู่ 2 คน จึงได้ว่า $P(\text{Marital Status} = \text{Single} | \text{Yes}) = 2/3$
- การหาค่า $P(\text{Annual Income} = \mathbf{X} | \text{Yes})$ เนื่องจาก Annual Income เป็นค่าจำนวนจริง การหาค่าความน่าจะเป็นจึงต้องพิจารณาจากค่าเฉลี่ยตัวอย่าง (sample mean, μ) และความแปรปรวนตัวอย่าง (sample variance, σ^2) ดังสมการ (2.7)

$$P(X; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(X-\mu)^2}{2\sigma^2} \right) \qquad (2.7)$$

เมื่อ X คือ ค่า Annual Income ของข้อมูลทดสอบ

μ คือ ค่าเฉลี่ยตัวอย่าง

σ คือ ส่วนเบี่ยงเบนมาตรฐานหรือรากที่สองของค่าความแปรปรวนตัวอย่าง

การจำแนกประเภทโดยอาศัยการเรียนรู้แบบ Naive Bayes โดยใช้ชุดข้อมูลสอน ดังรูปที่ 2.12 (ก) เราจะได้ตัวจำแนกจากการคำนวณความน่าจะเป็นแบบมีเงื่อนไขตามคลาสสำหรับแต่ละคุณลักษณะที่เป็นหมวดหมู่ (category และ binary) พร้อมกับค่าเฉลี่ยตัวอย่าง (sample mean) และความแปรปรวนตัวอย่าง (sample variance) สำหรับคุณลักษณะแบบต่อเนื่อง (continuous) ดังรูปที่ 2.12 (ข)

ในการจำแนกประเภทคลาสของข้อมูลทดสอบพิจารณาดังตัวอย่างต่อไปนี้ สมมติให้ข้อมูลทดสอบคือ $\mathbf{X} = (\text{Home Owner} = \text{No}, \text{Marital Status} = \text{Married}, \text{Annual Income} = 120\text{K})$ ซึ่งเราต้องการหาค่า $P(\text{No}|\mathbf{X})$ และ $P(\text{Yes}|\mathbf{X})$ เพื่อตัดสินใจจำแนกว่า $\mathbf{X}$ จะอยู่ในคลาส Yes หรือ **No** สามารถประมาณได้โดยการคำนวณความน่าจะเป็น prior probability $P(Y)$ และความน่าจะเป็นของคุณลักษณะ X_i แบบมีเงื่อนไขเมื่อกำหนดคลาสดังความสัมพันธ์ (2.6)

ความน่าจะเป็น prior probability ของแต่ละคลาสสามารถประมาณได้โดยการคำนวณสัดส่วนของข้อมูลการสอนที่เป็นของแต่ละคลาส เนื่องจากมีข้อมูลสอน 3 รายการที่เป็นของคลาส Yes และข้อมูลสอน 7 รายการที่เป็นของคลาส **No** ซึ่งจะได้ $P(\text{Yes}) = 0.3$ และ $P(\text{No}) = 0.7$ จากค่าความน่าจะเป็นบนชุดข้อมูลสอนที่ได้คำนวณไว้ ดังรูปที่ 2.12 (ข) จะนำมาใช้หาค่า $P(\mathbf{X}|\text{No})$ และ $P(\mathbf{X}|\text{Yes})$ ดังนี้

$$\begin{aligned} P(\mathbf{X}|\text{No}) &= P(\text{Home Owner} = \text{No}|\text{No}) \times P(\text{Status} = \text{Married}|\text{No}) \times P(\text{Annual Income} = \$120\text{K}|\text{No}) \\ &= \frac{4}{7} \times \frac{4}{7} \times 0.0072 \\ &= 0.0024 \end{aligned}$$

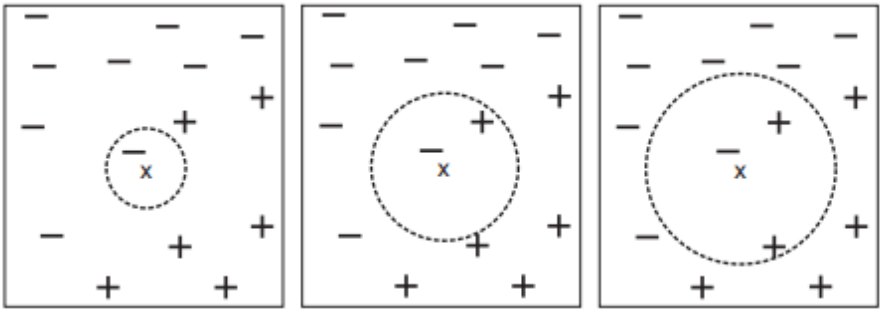
$$\begin{aligned} P(\mathbf{X}|\text{Yes}) &= P(\text{Home Owner} = \text{No}|\text{Yes}) \times P(\text{Status} = \text{Married}|\text{Yes}) \times P(\text{Annual Income} = \$120\text{K}|\text{Yes}) \\ &= 1 \times 0 \times 1.2 \times 10^{-9} \\ &= 0 \end{aligned}$$

ความน่าจะเป็นสำหรับแต่ละคลาส Y เมื่อกำหนดเงื่อนไขคุณลักษณะ $\mathbf{X}$ ดังสมการที่ (2.5) จะเห็นว่า $P(X)$ มีค่าเท่ากันทั้งในกรณีของคลาส Yes และ **No** จึงสามารถกำหนดให้ $1/P(X)$ มีค่าเท่ากับ α นั่นคือ ความน่าจะเป็นของคลาส **No** คือ $P(\text{No}|\mathbf{X}) = \alpha \times 0.7 \times 0.0024 = 0.0016$ และความน่าจะเป็นของคลาส Yes คือ $P(\text{Yes}|\mathbf{X}) = \alpha \times 0.3 \times 0 = 0$ เนื่องจาก $P(\text{No}|\mathbf{X}) > P(\text{Yes}|\mathbf{X})$ ดังนั้น ข้อมูลจึงถูกจัดประเภทให้เป็น **No**

2.3.2 การจำแนกประเภทข้อมูลด้วยวิธี k-Nearest Neighbors

เทคนิคการจำแนกประเภทข้อมูลด้วยวิธี k-Nearest Neighbors (k-NN) หรือเพื่อนบ้านที่ใกล้ที่สุด เป็นวิธีการเรียนรู้จากข้อมูลเพื่อนบ้านใกล้เคียงที่เป็นชุดข้อมูลการสอนเพื่อทำนายประเภทคลาสให้กับข้อมูลใหม่โดยไม่ต้องสร้างโมเดลในการทำนาย ซึ่งการทำนายประเภทคลาสให้กับข้อมูลใหม่จะต้องทำการวัดระยะห่างระหว่างข้อมูลสอนกับข้อมูลใหม่เพื่อระบุเพื่อนบ้านจำนวน k ตัว ซึ่งคือข้อมูลสอน k ตัวที่ใกล้กับข้อมูลใหม่มากที่สุด นำไปใช้ทำนายประเภทคลาสให้กับข้อมูลใหม่ได้

เทคนิคการจำแนกประเภทข้อมูลด้วยวิธีเพื่อนบ้านที่ดีที่สุดสามารถกำหนดรูปแบบในการสร้างขอบเขตตัดสินใจ โดยการกำหนดค่า k ซึ่งเป็นจำนวนข้อมูลสอนที่เป็นเพื่อนบ้านใกล้เคียงกับข้อมูลใหม่หรือข้อมูลทดสอบ แสดงดังรูปที่ 2.13

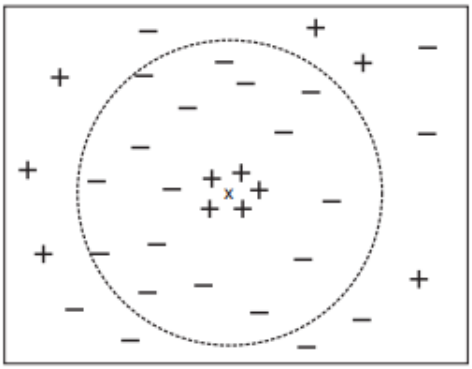


(ก) 1-nearest neighbor (ข) 2-nearest neighbor (ค) 3-nearest neighbor

รูปที่ 2.13 ตัวอย่างของเพื่อนบ้านที่ใกล้ที่สุดที่มีค่า k เป็น 1 2 และ 3 (Tan et al., 2019)

จากรูปที่ 2.13 เป็นตัวอย่างการสร้างขอบเขตตัดสินใจโดยกำหนดเพื่อนบ้านที่ใกล้ที่สุดมีค่า k เป็น 1 2 และ 3 เพื่อใช้ทำนายประเภทคลาสของข้อมูลทดสอบตามคลาสของเพื่อนบ้านที่ใกล้ที่สุด ตัวอย่างดังรูปที่ 2.13 (ก) กำหนด $k=1$ จึงพิจารณาข้อมูลสอนที่เป็นเพื่อนบ้านที่ใกล้ที่สุด 1 ตัวซึ่งเป็นคลาสลบ (-) ดังนั้นข้อมูลใหม่จะถูกกำหนดให้เป็นคลาสลบตามคลาสของข้อมูลสอนที่เป็นเพื่อนบ้าน สำหรับกรณีที่ $k>1$ และข้อมูลสอนที่เป็นเพื่อนบ้านมีมากกว่าหนึ่งคลาสแล้วการทำนาย

ประเภทคลาสให้กับข้อมูลทดสอบจะถูกกำหนดโดยการหาเสียงคลาสส่วนใหญ่ (majority voting) ของเพื่อนบ้านที่ใกล้ที่สุด ตัวอย่างดังรูปที่ 2.13 (ค) กำหนด $k=3$ จึงพิจารณาข้อมูลสอนที่เป็นเพื่อนบ้านที่ใกล้ที่สุด 3 ตัว ซึ่งเป็นคลาสบวก (+) 2 ตัวและเป็นคลาสลบ (-) 1 ตัว เมื่อใช้รูปแบบการจัดประเภทคลาสตามการหาเสียงส่วนใหญ่จะได้ว่าคลาสของข้อมูลทดสอบเป็นคลาสบวก ในกรณีที่จำนวนคลาสของข้อมูลเท่ากันไม่สามารถหาคลาสโดยใช้รูปแบบการหาเสียงส่วนใหญ่ ดังรูปที่ 2.13 (ข) แนวทางที่ทำได้คือการคำนวณหาค่าเฉลี่ยหรือผลรวมของระยะทางของเพื่อนบ้านแต่ละคลาสแล้วกำหนดข้อมูลตัวทดสอบให้เป็นไปตามคลาสของประเภทที่มีค่าเฉลี่ยหรือผลรวมของระยะทางน้อยที่สุด โดยรูปแบบของการสร้างขอบเขตตัดสินใจดังกล่าวทำให้แบบจำลองมีความยืดหยุ่นมากขึ้น ขอบเขตการจำแนกโดยใช้ข้อมูลสอนที่เป็นเพื่อนบ้านที่ดีที่สุดมีความผันผวนสูงเพราะขึ้นอยู่กับจำนวนเพื่อนบ้านใกล้เคียงจำนวน k ตัว ซึ่งการเพิ่มค่า k อาจจะลดความผันผวนดังกล่าวได้ การที่เลือกค่า k ให้เหมาะสมมีความสำคัญมาก หาก k มีขนาดเล็กเกินไปจะทำให้ตัวจำแนกเพื่อนบ้านที่ใกล้ที่สุดอาจจะไม่เหมาะสม หรือเกิด overfitting ได้ ในทางกลับกันหาก k มีขนาดใหญ่เกินไปจะทำให้ตัวจำแนกเพื่อนบ้านที่ใกล้ที่สุดอาจจะจำแนกคลาสของข้อมูลทดสอบผิดพลาดเนื่องจากข้อมูลสอนที่เป็นเพื่อนบ้านอาจจะอยู่ห่างกับข้อมูลสอนมากเกินไป ดังตัวอย่างรูปที่ 2.14



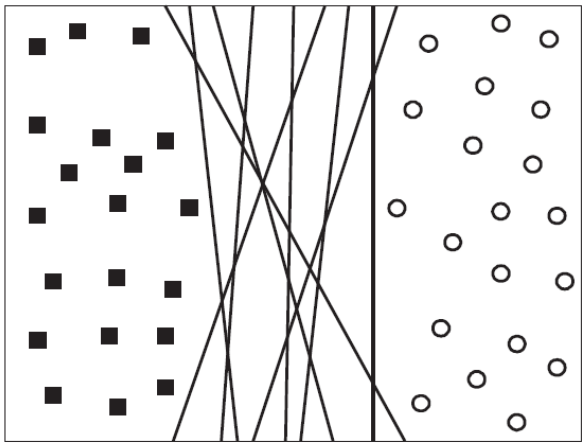
รูปที่ 2.14 ตัวอย่างของเพื่อนบ้านที่ใกล้ที่สุดที่ค่า k มีขนาดใหญ่ (Tan et al., 2019)

อย่างไรก็ตามการทำนายคลาสให้กับข้อมูลทดสอบด้วยเทคนิคการจำแนกประเภทข้อมูลด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุดมีค่าใช้จ่ายเพิ่มมากขึ้นตามจำนวนของข้อมูลสอนเนื่องจากจำเป็นต้องคำนวณหาค่าความใกล้เคียงระหว่างชุดข้อมูลสอนทุกตัวกับข้อมูลทดสอบที่เข้ามาเพื่อใช้ในการหาข้อมูลสอนที่ใกล้สุด k ตัว สำหรับเป็นเพื่อนบ้านในการตัดสินใจในการทำนายคลาสให้กับข้อมูลทดสอบ

2.3.3 การจำแนกประเภทข้อมูลด้วยวิธี Support Vector Machine

เทคนิค Support Vector Machine หรือ SVM นี้มีรากฐานมาจากทฤษฎีการเรียนรู้เชิงสถิติ และแสดงผลลัพธ์ที่มีประโยชน์ในการใช้งานจริงได้หลายอย่าง เช่น การรู้จำลายมือและการจัดหมวดหมู่ข้อความ ซึ่ง SVM ยังสามารถทำงานได้ดีมากกับข้อมูลที่มิติสูง โดย SVM จะแสดงถึงขอบเขตการตัดสินใจโดยใช้ชุดข้อมูลย่อยของตัวอย่างข้อมูลการสอนหรือเรียกว่า เวกเตอร์สนับสนุน (Support Vector)

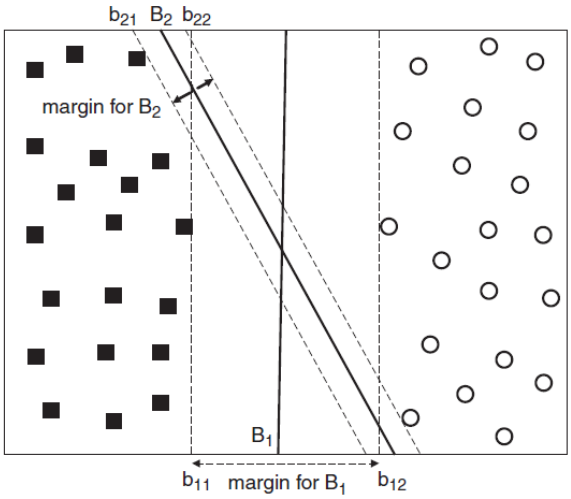
พิจารณาขอบเขตการตัดสินใจที่เป็นไปได้ ดังรูปที่ 2.15 จะเห็นว่าจุดของตัวอย่างข้อมูลของสี่เหลี่ยมและวงกลมที่เป็นข้อมูลสองคลาสที่แตกต่างกัน โดยชุดข้อมูลสามารถแบ่งแยกแต่ละประเภทได้โดยเส้นตรง กล่าวคือเราสามารถหาไฮเปอร์เพลน (hyperplane) ได้โดยการให้จุดสี่เหลี่ยมทั้งหมดอยู่ด้านหนึ่งของไฮเปอร์เพลนและจุดวงกลมทั้งหมดอยู่อีกด้านหนึ่งของไฮเปอร์เพลน อย่างไรก็ตามในรูปที่ 2.15 มีหลายไฮเปอร์เพลนที่เป็นไปได้ แม้ว่าข้อผิดพลาดในการเรียนรู้จะเป็นศูนย์ก็ตามแต่ไม่ได้มีการรับประกันว่าไฮเปอร์เพลนจะทำงานได้ดีพอๆ กันเมื่อต้องใช้จำแนกตัวอย่างข้อมูลที่ไม่เคยเห็นมาก่อน (unseen data) ซึ่งการสร้างตัวจำแนกทำได้โดยเลือกหนึ่งในไฮเปอร์เพลนเหล่านี้เพื่อแสดงขอบเขตการตัดสินใจโดยพิจารณาไฮเปอร์เพลนที่ดีที่สุดกับตัวอย่างข้อมูลทดสอบ



รูปที่ 2.15 ขอบเขตการตัดสินใจที่เป็นไปได้สำหรับชุดข้อมูลที่แยกโดยใช้การแบ่งเชิงเส้น (Tan et al., 2019)

เพื่อให้ได้ภาพที่ชัดเจนยิ่งขึ้นว่าตัวเลือกของไฮเปอร์เพลนที่แตกต่างกันจะมีผลต่อข้อผิดพลาดอย่างไร ให้พิจารณาขอบเขตการตัดสินใจสองขอบเขตคือ B_1 และ B_2 ดังแสดงในรูปที่ 2.16 ขอบเขตการตัดสินใจทั้งสองนั้นสามารถที่จะแยกตัวอย่างข้อมูลการสอนออกเป็นแต่ละคลาสได้โดยไม่

มีการจำแนกประเภทข้อมูลที่ผิดพลาด ในแต่ละขอบเขตการตัดสินใจ B_i จะเกี่ยวข้องกับคู่ของไฮเปอร์เพลนที่แทนด้วย b_{i1} และ b_{i2} ตามลำดับ ซึ่ง b_{i1} นั้นได้มาจากการเคลื่อนไฮเปอเพลนในแนวขนานออกจากขอบเขตการตัดสินใจจนกระทั่งสัมผัสกับจุดสี่เหลี่ยมที่ใกล้ที่สุด ในขณะที่ b_{i2} นั้นได้มาจากการเคลื่อนไฮเปอร์เพลนจนกระทั่งสัมผัสกับจุดวงกลมที่ใกล้ที่สุดโดยที่ระยะห่างระหว่างไฮเปอร์เพลนทั้งสองนี้เรียกว่า ระยะขอบ (margin) ของตัวจำแนก จากรูปที่ 2.16 จะสังเกตว่าระยะขอบของ B_1 นั้นใหญ่กว่าระยะขอบขอบของ B_2 นั่นหมายความว่า B_1 เป็นไฮเปอร์เพลนที่มีระยะขอบมากที่สุดของข้อมูลการสอนนี้



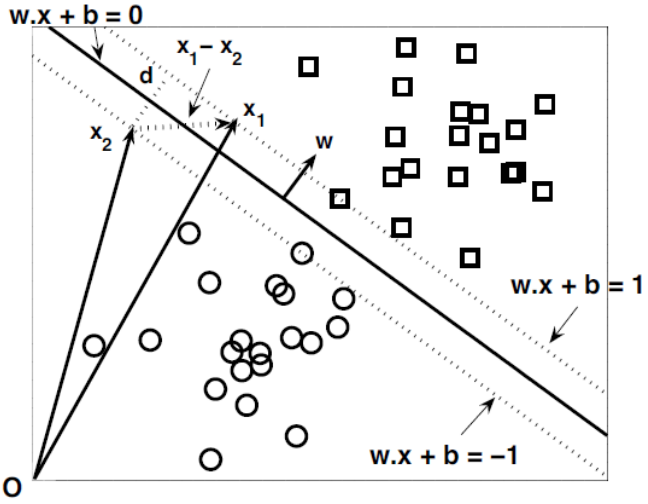
รูปที่ 2.16 ขอบเขตการตัดสินใจที่แบ่งแยกชุดข้อมูล
โดยใช้เส้นแบ่ง B_1 และ B_2 (Tan et al., 2019)

SVM แบบเชิงเส้นเป็นตัวจำแนกที่มีไฮเปอร์เพลนที่มีระยะห่างมากที่สุดหรือเรียกว่า ตัวจำแนกระยะขอบมากที่สุด เพื่อให้เข้าใจว่า SVM เรียนรู้ขอบเขตดังกล่าวได้อย่างไร เราจะต้องพิจารณาเบื้องต้นเกี่ยวกับขอบเขตตัดสินใจและระยะขอบของตัวจำแนกเชิงเส้น

กำหนดให้พิจารณาปัญหาในการจำแนกข้อมูลออกเป็นสองคลาสซึ่งประกอบด้วยข้อมูลสอน N ตัวอย่าง แต่ละข้อมูลจะแสดงโดยรายการคู่อันดับ $(\mathbf{x}_i, y_i)$ โดยที่ $(i = 1, 2, \ldots, N)$ เมื่อ $\mathbf{x} = (x_{i1}, x_{i2}, \ldots, x_{id})^T$ แทนข้อมูลสอนตัวที่ i ซึ่งมี d มิติหรือ d คุณลักษณะ และ $y_i \in \{-1, 1\}$ แทนคลาสของข้อมูลสอน $\mathbf{x}_i$ เราสามารถเขียนสมการขอบเขตการตัดสินใจของตัวจำแนกเชิงเส้นได้ดังรูปแบบต่อไปนี้

$$\mathbf{w} \cdot \mathbf{x}_i + b = 0 \tag{2.8}$$

โดยที่ $\mathbf{w}$ และ b คือพารามิเตอร์ของโมเดล



รูปที่ 2.17 ขอบเขตการตัดสินใจและระยะขอบของ SVM (Tan et al., 2019)

รูปที่ 2.17 แสดงชุดข้อมูลการสอนสองมิติที่ประกอบด้วยจุดสี่เหลี่ยมและจุดวงกลม ซึ่งขอบเขตการตัดสินใจที่ทำการแบ่งตัวอย่างข้อมูลการสอนออกเป็นแต่ละคลาสจะแสดงด้วยเส้นทึบ และข้อมูลใดที่อยู่บนเส้นขอบเขตการตัดสินใจจะต้องเป็นไปตามสมการที่ 2.8 ตัวอย่างเช่น สมมติให้ $\mathbf{x}_a$ และ $\mathbf{x}_b$ เป็นสองจุดที่อยู่บนขอบเขตการตัดสินใจจะได้ว่า

$$\mathbf{w} \cdot \mathbf{x}_a + b = 0 \tag{2.9}$$

$$\mathbf{w} \cdot \mathbf{x}_b + b = 0 \tag{2.10}$$

เมื่อนำสมการ (2.10) – (2.9) จะให้ผลดังนี้

$$\mathbf{w} \cdot \left(\mathbf{x}_b - \mathbf{x}_a \right) = 0 \tag{2.11}$$

โดยที่ $\mathbf{x}_b - \mathbf{x}_a$ เป็นเวกเตอร์ขนานกับขอบการตัดสินใจและพุ่งตรงจาก $\mathbf{x}_a$ ไปถึง $\mathbf{x}_b$ เนื่องจากผลลัพธ์เป็นศูนย์ทำให้ทิศทางของ $\mathbf{w}$ จะต้องตั้งฉากกับขอบการตัดสินใจดังแสดงในรูปที่ 2.17

สำหรับจุดสี่เหลี่ยม $\mathbf{x}_s$ ใดๆ ที่อยู่เหนือขอบเขตการตัดสินใจ เราสามารถแสดงได้ดังนี้

$$\mathbf{w} \cdot \mathbf{x}_s + b > 0 \tag{2.12}$$

ในทำนองเดียวกันสำหรับจุดวงกลม $\mathbf{x}_c$ ใดๆ ที่อยู่ใต้ขอบเขตการตัดสินใจ เราสามารถแสดงได้ดังนี้

$$\mathbf{w}\cdot\mathbf{x}_c + b < 0 \tag{2.13}$$

ถ้าหากเรากำหนดเครื่องหมายให้จุดสี่เหลี่ยมทั้งหมดเป็นคลาส $+1$ และจุดวงกลมทั้งหมดเป็นคลาส -1 จากนั้นเราทำนายเครื่องหมายคลาส y สำหรับตัวอย่างข้อมูลทดสอบ $\mathbf{z}$ ดังวิธีต่อไปนี้

$$y = \begin{cases} 1, & \text{ถ้า } \mathbf{w}\cdot\mathbf{z}+b>0 \\ -1, & \text{ถ้า } \mathbf{w}\cdot\mathbf{z}+b<0 \end{cases} \tag{2.14}$$

กำหนดให้พิจารณาจุดสี่เหลี่ยมและจุดวงกลมที่ใกล้เคียงกับขอบเขตตัดสินใจ เนื่องจากจุดสี่เหลี่ยมตั้งอยู่เหนือขอบเขตการตัดสินใจซึ่งจะต้องเป็นไปตามสมการ (2.12) ในขณะที่จุดวงกลมจะเป็นไปตามสมการ (2.13) เราสามารถใช้พารามิเตอร์ $\mathbf{w}$ และ b กำหนดขยายขอบเขตการตัดสินใจเพื่อให้ได้ไฮเปอร์เพลนที่ขนานกับเส้นขอบเขตการตัดสินใจทั้งในฝั่งคลาส $+1$ และคลาส -1 ได้แก่ ไฮเปอร์เพลน b_{i1} และ b_{i2} ตามลำดับสามารถแสดงได้ดังนี้

$$b_{i1}: \quad \mathbf{w}\cdot\mathbf{x}+b=1 \tag{2.15}$$

$$b_{i2}: \quad \mathbf{w}\cdot\mathbf{x}+b=-1 \tag{2.16}$$

ขั้นตอนในการเรียนรู้ของ SVM เกี่ยวข้องกับการประมาณพารามิเตอร์ $\mathbf{w}$ และ b ของขอบเขตการตัดสินใจจากข้อมูลการสอนซึ่งพารามิเตอร์ต้องถูกเลือกในตัวจำแนกที่ตรงตามเงื่อนไขสองข้อต่อไปนี้

$$\begin{aligned} \mathbf{w}\cdot\mathbf{x}_i + b \geq 1 \quad & \text{ถ้า } y_i = 1 \\ \mathbf{w}\cdot\mathbf{x}_i + b \leq -1 \quad & \text{ถ้า } y_i = -1 \end{aligned} \tag{2.17}$$

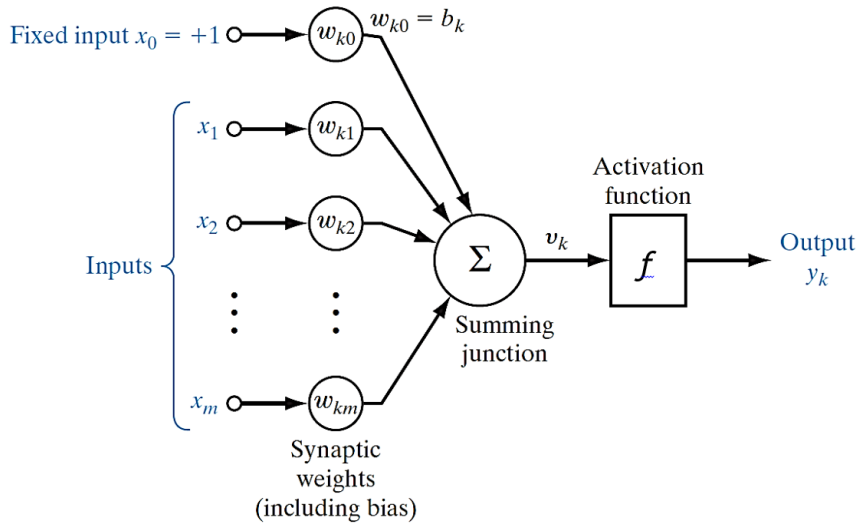
เงื่อนไขเหล่านี้จะกำหนดข้อกำหนดที่ข้อมูลการสอนทั้งหมดจากคลาส $y=1$ (เช่นจุดสี่เหลี่ยม) จะต้องอยู่บนหรือเหนือไฮเปอร์เพลน $\mathbf{w}\cdot\mathbf{x}+b=1$ ในขณะที่ข้อมูลเหล่านั้นจากคลาส $y=-1$ (เช่นจุดวงกลม) จะต้องอยู่บนหรือด้านล่างไฮเปอร์เพลน $\mathbf{w}\cdot\mathbf{x}+b=-1$ ทั้งสองกรณีของคลาสที่ต่างกันสามารถสรุปได้ในรูปแบบที่กะทัดรัดมากขึ้นดังนี้

$$y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 \qquad \text{โดยที่ } i = 1, 2, \ldots, N \tag{2.18}$$

2.3.4 เทคนิคการจำแนกประเภทข้อมูลด้วยวิธี Artificial Neural Networks

โครงข่ายประสาทเทียม หรือ Artificial Neural Networks (ANN) ประกอบด้วย นิวรอน (neural) เป็นจำนวนมาก แสดงโครงสร้างของแต่ละนิวรอน แสดงดังรูปที่ 2.18 สมมติให้เป็นนิวรอนที่ k มีองค์ประกอบพื้นฐาน 3 ประการ (Haykin, 2009) ดังนี้

1) เซตของ synaptic หรือ connection link ซึ่งแต่ละเส้นจะมีค่าน้ำหนัก (weight) กำกับอยู่ ถ้าข้อมูลที่เข้ามายังนิวรอนที่ k มีจำนวนมิติเท่ากับ m แล้วจะมีเส้นเชื่อมทั้งหมด $m+1$ เส้น และมีค่าน้ำหนักกำกับแต่ละเส้น คือ w_{ij} เมื่อ $j=0,1,2,\ldots,m$ เมื่อมีข้อมูล $\mathbf{x}=[x_1,x_2,\ldots.x_m]^T$ เข้ามายังนิวรอนที่ k จะต้องนำข้อมูลมิติที่ j หรือ x_j คูณกันกับค่าน้ำหนัก w_{ij} ของแต่ละ synaptic ที่ j โดยจะอธิบายตัวห้อยของค่าน้ำหนัก w_{ij} ได้ดังนี้ ตัวห้อยที่หนึ่ง หมายถึง นิวรอนที่รับข้อมูลเข้ามา ตัวห้อยที่สอง หมายถึง synaptic ที่สัมพันธ์กับข้อมูลในแต่ละมิติ
2) ผลรวมของผลคูณข้อมูลที่รับเข้ามาและค่าน้ำหนักตามลำดับ synaptic ของแต่ละนิวรอน ซึ่งการดำเนินการนี้จะเป็นผลรวมเชิงเส้น (linear combiner) และผลรวมที่ได้ก็จะนำมารวมกับค่า bias ซึ่ง bias เป็นค่าที่ปรับความยืดหยุ่นของค่าข้อมูลที่รับเข้ามา
3) ฟังก์ชันกระตุ้น (activation function) จะเป็นฟังก์ชันที่ใช้สำหรับในการหาผลลัพธ์ข้อมูลของนิวรอนนั้น



รูปที่ 2.18 โครงสร้างของนิวรอน (Haykin, 2009)

จากรูปที่ 2.18 จะสามารถเขียนสมการได้ดังนี้

$$v_k = \left(\sum_{j=1}^{m} w_{kj} x_j \right) + b_k \tag{2.19}$$

และ

$$y_k = f\left(v_k\right) \tag{2.20}$$

เมื่อ $x_1, x_2, \ldots, x_m$ คือ ค่าข้อมูลนำเข้า

$w_{k1}, w_{k2}, \ldots, w_{km}$ คือ ค่าน้ำหนัก synaptic ของแต่ละนิวรอน k

v_k คือ ผลรวมของข้อมูลและ bias

b_k คือ ค่า bias

f คือ ฟังก์ชัน f ที่ใช้ในการคำนวณ

y_k คือ ผลลัพธ์ของนิวรอน

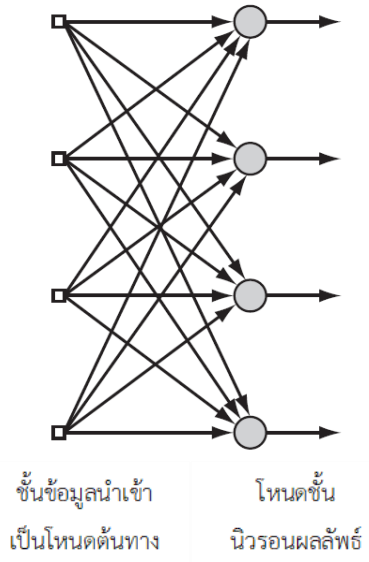
ผลลัพธ์ที่ได้จากนิวรอนนั้นอาจจะไปเป็นข้อมูลนำเข้าของนิวรอนอื่นหรือจะเป็นผลลัพธ์ของโครงข่ายก็ได้ โดยทั่วไปโครงสร้างของโครงข่ายประสาทเทียมจะเป็นแบบป้อนไปหน้า (feed forward) การเชื่อมต่อดังกล่าวนี้ ทำให้เกิดเป็นโครงข่ายที่มีหลายนิวรอนและมีได้หลายชั้น ซึ่งมีการส่งข้อมูลจากชั้นหนึ่งไปยังชั้นถัดไปจนถึงชั้นผลลัพธ์

โครงสร้างของโครงข่ายประสาทเทียมจะการเชื่อมโยงกันของนิวรอนด้วยอัลกอริทึมการเรียนรู้ที่ใช้สอนโครงข่ายเพื่อที่จะออกแบบลักษณะโครงสร้างของโครงข่ายประสาทเทียม โดยทั่วไปโครงสร้างของโครงข่ายประสาทเทียมจะมีลักษณะของชั้นที่แตกต่างกันออกไปต่อไปนี้ (Haykin, 2009)

1) โครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าที่มีชั้นเดียว

โครงข่ายประสาทเทียมโดยทั่วไป นิวรอนจะถูกจัดเรียงในรูปแบบเป็นชั้น โดยรูปแบบอย่างง่ายที่สุดจะประกอบด้วย ชั้นข้อมูลนำเข้า (input layer) ของโหนดต้นทางที่จะทำการส่งข้อมูลไหลไปยังชั้นผลลัพธ์ (output layer) ซึ่งเป็นโหนดคำนวณ โดยตรง และจะไม่ทำการส่งข้อมูลไหลย้อนกลับไปยังชั้นข้อมูลนำเข้าของโหนดต้นทางอีก กล่าวได้ว่า โครงข่ายแบบนี้จะเป็นแบบป้อนไปข้างหน้า

แสดงตัวอย่างดังรูปที่ 2.19 เป็นกรณีที่มี 4 โหนดในชั้นข้อมูลนำเข้าและชั้นผลลัพธ์ ตัวอย่างของโครงข่ายดังกล่าวนี้จะเรียกว่าโครงข่ายแบบชั้นเดียว (single layer) ซึ่งหมายถึงชั้นผลลัพธ์ของโครงข่าย ทั้งนี้เราจะไม่นับชั้นของโหนดต้นทางเนื่องจากโหนดนั้นไม่มีการคำนวณ



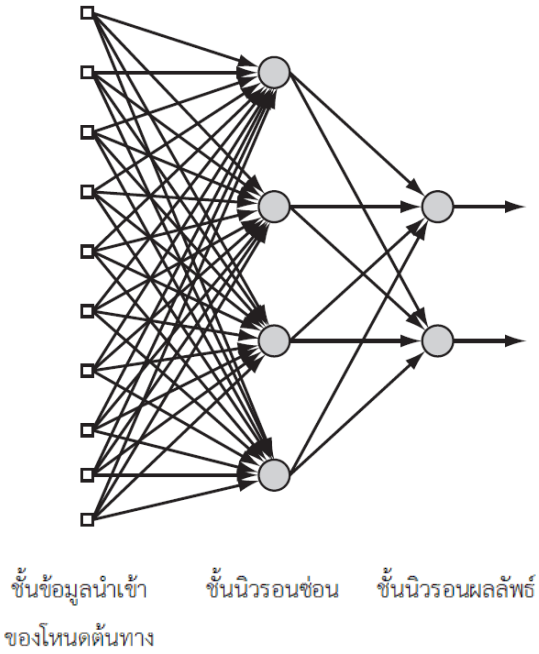
รูปที่ 2.19 ตัวอย่างของโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าที่มีนิวรอนชั้นเดียว (Haykin, 2009)

2) โครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าที่มีหลายชั้น

ในโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าที่มีหลายชั้นจะมีชั้นที่แยกออกมาเรียกว่า ชั้นซ่อน (hidden layer) โดยจะมีชั้นซ่อนหนึ่งชั้นหรืออาจจะมีมากกว่านั้น โหนดคำนวณจะเรียกว่านิวรอนซ่อน (hidden neurons) หรือหน่วยซ่อน (hidden units) ซึ่งคำว่า “ซ่อน” จะหมายถึงในส่วนนี้ของโครงข่ายประสาทเทียมจะไม่เห็นโดยตรงจากทั้งข้อมูลนำเข้า หรือผลลัพธ์ (output) ของโครงข่าย หน้าที่ของนิวรอนซ่อนคือการแทรกทำงานเพื่อให้เกิดผลดีขึ้นระหว่างข้อมูลนำเข้าจากภายนอก (external input) และผลลัพธ์ของโครงข่าย ด้วยการที่เพิ่มชั้นซ่อนหนึ่งชั้นหรือมากกว่านั้นทำให้โครงข่ายมีความสามารถในการคำนวณทางสถิติขั้นสูงที่จะสามารถแบ่งข้อมูลนำเข้าได้

โหนดต้นทางในชั้นข้อมูลนำเข้าของโครงข่ายจะจัดเตรียมรูปแบบการไหลผ่านหรือเวกเตอร์นำเข้าข้อมูล ซึ่งเป็นสัญญาณนำเข้าที่ใช้กับนิวรอนหรือโหนดคำนวณในชั้นที่สอง จากนั้นสัญญาณส่งออกของชั้นที่สองจะถูกใช้เป็นข้อมูลนำเข้าไปยังชั้นที่สามและชั้นอื่นๆ ต่อไปที่เหลืออยู่ในโครงข่าย โดยปกตินิวรอนในแต่ละชั้นของโครงข่ายจะมีข้อมูลนำเข้าที่เป็นสัญญาณส่งออกมาจากชั้นก่อนหน้านี้

เท่านั้น ชุดสัญญาณที่ส่งออกของนิวรอนในชั้นผลลัพธ์หรือชั้นสุดท้ายของโครงข่ายถือเป็นการตอบสนองโดยรวมของโครงข่ายกับรูปแบบการไหลผ่านของข้อมูลจากโหนดต้นทางในชั้นข้อมูลนำเข้าหรือชั้นแรกนั่นเอง ในรูปที่ 2.20 เป็นการแสดงตัวอย่างโครงร่างของโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าที่มีหลายชั้นสำหรับกรณีที่มีชั้นซ่อนเดียว เพื่อความกระชับของโครงข่ายประสาทเทียมในรูปที่ 2.20 ซึ่งเราจะเรียกว่า โครงข่าย 10-4-2 เพราะมี 10 โหนดต้นทาง 4 นิวรอนที่ซ่อนอยู่ และ 2 นิวรอนผลลัพธ์ และอีกหนึ่งตัวอย่างของโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าที่มี m โหนดต้นทาง h1 นิวรอนในชั้นซ่อนที่หนึ่ง h2 นิวรอนในชั้นซ่อนที่สองและ q นิวรอนในชั้นผลลัพธ์ จะเรียกว่าโครงข่าย m-h1-h2-q



รูปที่ 2.20 ตัวอย่างของโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าที่มีนิวรอนหลายชั้น (Haykin, 2009)

2.4 ตัววัดระยะทาง

การวัดระยะทางได้กลายมาเป็นเครื่องมือสำคัญในการใช้งานหลายๆ ด้านของคณิตศาสตร์ ได้แก่ เรขาคณิต ความน่าจะเป็นสถิติ ทฤษฎีกราฟ การจัดกลุ่ม การวิเคราะห์ข้อมูล การจดจำรูปแบบและอื่นๆ อีกมากมาย ซึ่งในทางคณิตศาสตร์ระยะทางเป็นการประยุกต์ที่มีหลักการความคิดเกี่ยวกับระยะทางซึ่งเป็นความยาวระหว่างจุดข้อมูลสองจุด และระยะทางยังช่วยให้สามารถจัดกลุ่มข้อมูลที่มี

ความคล้ายคลึงกันและแยกพวกข้อมูลที่ไม่คล้ายกันได้ ส่วนใหญ่การจัดกลุ่มจะเป็นการแบ่งส่วนของข้อมูลออกเป็นกลุ่มๆ โดยที่กลุ่มของข้อมูลที่มีระยะที่ใกล้กันจะอยู่กลุ่มเดียวกัน ในทางกลับกันถ้าข้อมูลที่มีระยะที่ไกลออกไปจะอยู่กลุ่มที่แตกต่างกันไป

การวัดระยะทางเป็นส่วนสำคัญของการเรียนรู้ของเครื่อง โดยทั่วไปทำการคำนวณหาระยะห่างระหว่างจุดข้อมูลแล้วกำหนดความคล้ายคลึงกันระหว่างจุดข้อมูล ซึ่งตัววัดระยะทางมีได้หลายวิธี ในที่นี้จะกล่าวถึง 4 วิธีดังต่อไปนี้

2.4.1 Euclidean Distance

Euclidean Distance เป็นตัววัดระยะทางที่นิยมใช้กันอย่างมาก โดยการหาระยะทางของเส้นตรงที่เชื่อมระหว่างจุดข้อมูลสองจุด ถ้าหากระยะห่างระหว่างจุดข้อมูลมีค่าน้อยแสดงว่าจุดข้อมูลทั้งสองจุดมีความใกล้เคียงหรือความคล้ายคลึงกันมาก แต่ถ้าหากระยะห่างระหว่างจุดข้อมูลมีค่ามากแสดงว่าจุดข้อมูลทั้งสองจุดนี้ห่างกันหรือมีความแตกต่างกันมาก ซึ่งสามารถคำนวณหาระยะทางด้วยตัววัดระยะทางแบบ Euclidean Distance ได้ดังสมการ 2.21

$$D_{euclidean} = \sqrt{\sum_{i=1}^{n} \left(p_i - q_i \right)^2} \tag{2.21}$$

โดยที่

n แทน จำนวนมิติของข้อมูล

$\mathbf{p}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{p} = \left[p_1, p_2, \ldots, p_n \right]^T$

$\mathbf{q}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{q} = \left[q_1, q_2, \ldots, q_n \right]^T$

2.4.2 Manhattan Distance

Manhattan Distance เป็นตัววัดระยะทางแบบทางเดินรถโดยการคำนวณหาผลรวมความยาวที่แตกต่างกันของระยะทางระหว่างจุดข้อมูลสองจุดที่มีลักษณะคล้ายกริด (grid) ตามแนวแกนแนวตั้งและแนวนอน ซึ่งสามารถคำนวณหาระยะทางด้วยตัววัดระยะทางแบบ Manhattan Distance ได้ดังสมการ 2.22

$$D_{manhattan} = \sum_{i=1}^{n} \left| p_i - q_i \right| \tag{2.22}$$

โดยที่

n แทน จำนวนมิติของข้อมูล

$\mathbf{p}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{p} = [p_1, p_2, \ldots, p_n]^T$

$\mathbf{q}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$

2.4.3 Minkowski Distance

Minkowski Distance เป็นการวัดระยะทางหรือความคล้ายคลึงกันระหว่างจุดข้อมูลสองจุด โดยมีลักษณะทั่วไปที่ได้มาจากรูปแบบของ Euclidean Distance และ Manhattan Distance ซึ่งสามารถคำนวณหาระยะทางด้วยตัววัดระยะทางแบบ Minkowski Distance ได้ดังสมการ 2.23

$$D_{minkowski} = \sqrt[h]{\sum_{i=1}^{n} |p_i - q_i|^h} \tag{2.23}$$

โดยที่

n แทน จำนวนมิติของข้อมูล

$\mathbf{p}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{p} = [p_1, p_2, \ldots, p_n]^T$

$\mathbf{q}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$

h แทน ค่าจำนวนจริง เมื่อ h เท่ากับ 1 หรือ 2 ซึ่งสอดคล้องกับ Manhattan Distance และ Euclidean Distance ตามลำดับ และ เมื่อ h มีค่าเท่ากับอนันต์ (infinity) ซึ่งสอดคล้องกับ Chebychev Distance

2.4.4 Cosine Similarity

Cosine Similarity เป็นการหาความคล้ายคลึงกันโดยการวัดมุมโคไซน์ระหว่างจุดข้อมูลสองจุด เมื่อระยะห่างระหว่างจุดข้อมูลเพิ่มขึ้นความคล้ายคลึงกันของโคไซน์หรือปริมาณความคล้ายคลึงจะลดลง ในทางกลับกันเมื่อระยะห่างระหว่างจุดข้อมูลน้อยลงความคล้ายคลึงกันของโคไซน์หรือปริมาณความคล้ายคลึงจะมากขึ้น ดังนั้นคะแนนที่อยู่ใกล้กันจะคล้ายกันมากกว่าคะแนนที่อยู่ไกลกัน ซึ่งสามารถคำนวณหาระยะทางด้วยตัววัดระยะทางแบบ Cosine Similarity ได้ดังสมการ 2.24

$$D_{similarity} = \cos\theta = \frac{p_i \cdot q_i}{\|p_i\| \times \|q_i\|} = \frac{\sum_{i=1}^{n} p_i \times q_i}{\sqrt{\sum_{i=1}^{n} (p_i)^2} \times \sqrt{\sum_{i=1}^{n} (q_i)^2}} \quad (2.24)$$

โดยที่

n แทน จำนวนมิติของข้อมูล

$\mathbf{p}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{p} = [p_1, p_2, \ldots, p_n]^T$

$\mathbf{q}$ แทน จุดข้อมูลใดๆ เมื่อ $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$

2.5 การวัดประสิทธิภาพของโมเดลการจำแนกประเภทข้อมูล

ข้อมูลที่นำเข้ามานั้น จะต้องมีการแบ่งข้อมูลออกเป็น 2 ส่วน ประกอบด้วย ข้อมูลสอน (training data) เป็นข้อมูลส่วนที่ใช้สำหรับการเรียนรู้เพื่อสร้างโมเดล และข้อมูลทดสอบ (testing data) เป็นข้อมูลส่วนที่ใช้สำหรับประเมินความถูกต้องของโมเดล ซึ่งจะประเมินว่าโมเดลสามารถจำแนกคลาสของข้อมูลสอนซึ่งเป็นข้อมูลที่โมเดลไม่เคยพบมาก่อน (unseen data) สามารถประเมินได้โดยการเปรียบเทียบผลการจำแนกคลาสของข้อมูลทดสอบที่ได้จากโมเดลกับคลาสจริงของข้อมูลทดสอบนั้นๆ ผลการจำแนกตัวที่ถูกต้องและไม่ถูกต้องจะถูกนำมาคำนวณตัวชี้วัดประสิทธิภาพ ควรต้องมีความครอบคลุมความเป็นไปได้ของข้อความที่จะเกิดขึ้น จะต้องไม่เฉพาะเจาะจงเลือกการแบ่งข้อมูลสอนและทดสอบเพียงแค่ชุดใดชุดหนึ่งแล้วสรุปผล การวัดประสิทธิภาพของโมเดลการจำแนกมีหลักการแบ่งข้อมูลสอนและข้อมูลทดสอบได้หลายวิธี ในที่นี้จะกล่าวถึง 2 วิธีดังต่อไปนี้

2.5.1 การแบ่งข้อมูลสอนและข้อมูลทดสอบ

ข้อมูลที่นำเข้ามานั้น จะต้องมีการแบ่งข้อมูลออกเป็น 2 ส่วน ประกอบด้วย ข้อมูลสอน (train data) เป็นข้อมูลส่วนที่ใช้สำหรับการเรียนรู้เพื่อสร้างโมเดล และข้อมูลทดสอบ (test data) เป็นข้อมูลส่วนที่ใช้สำหรับประเมินความถูกต้องของโมเดล ซึ่งจะประเมินว่าโมเดลสามารถจำแนกคลาสของข้อมูลสอนซึ่งเป็นข้อมูลที่โมเดลไม่เคยพบมาก่อน (unseen data) สามารถประเมินได้โดยการเปรียบเทียบผลการจำแนกคลาสของข้อมูลทดสอบที่ได้จากโมเดลกับคลาสจริงของข้อมูลทดสอบนั้นๆ ผลการจำแนกตัวที่ถูกต้องและไม่ถูกต้องจะถูกนำมาคำนวณเพื่อหาประสิทธิภาพในการจำแนก

1) การแบ่งข้อมูลแบบ Holdout Method

การแบ่งข้อมูลแบบ holdout method (Tan et al., 2019) วิธีนี้จะเป็นการแบ่งข้อมูลแบบการสุ่มเลือกจากข้อมูลทั้งหมดออกเป็น 2 ส่วน คือ ข้อมูลสอนและข้อมูลทดสอบ ซึ่งสัดส่วนของการแบ่งข้อมูลสอนและข้อมูลทดสอบขึ้นอยู่กับการตัดสินใจของผู้ทำการทดลอง เช่น การแบ่งข้อมูลออกเป็นสองในสามสำหรับข้อมูลสอนและหนึ่งในสามสำหรับข้อมูลทดสอบ

สำหรับขนาดของข้อมูลแต่ละส่วนถ้ามีสัดส่วนที่ไม่เหมาะสมจะทำให้การแบ่งข้อมูลชุดนั้นไม่มีประสิทธิภาพต่อการวิเคราะห์ข้อมูลและอัตราค่าความคาดเคลื่อนอาจจะมีความไม่เสถียร เช่น ถ้าข้อมูลสอนมีขนาดเล็ก ตัวอย่างของรูปแบบการจำแนกที่ใช้ในการเรียนรู้ก็จะไม่เพียงพอ และอาจจะทำให้การประเมินประสิทธิภาพของข้อมูลทดสอบมีค่าความคลาดเคลื่อนที่สูงเกินไป ในทางกลับกันถ้าข้อมูลทดสอบมีขนาดเล็ก การประเมินประสิทธิภาพก็อาจจะมีความน่าเชื่อถือน้อย เนื่องจากจำนวนข้อมูลที่ใช้ในการทดสอบน้อย ทำให้ได้ผลที่ไม่แน่นอน

วิธีการแบ่งข้อมูลแบบ holdout method นี้สามารถทำซ้ำได้หลายครั้ง เพื่อให้มีการกระจายตัวของอัตราค่าความผิดพลาดได้ การทดสอบแบบนี้ เรียกว่า การสุ่มตัวอย่างย่อย

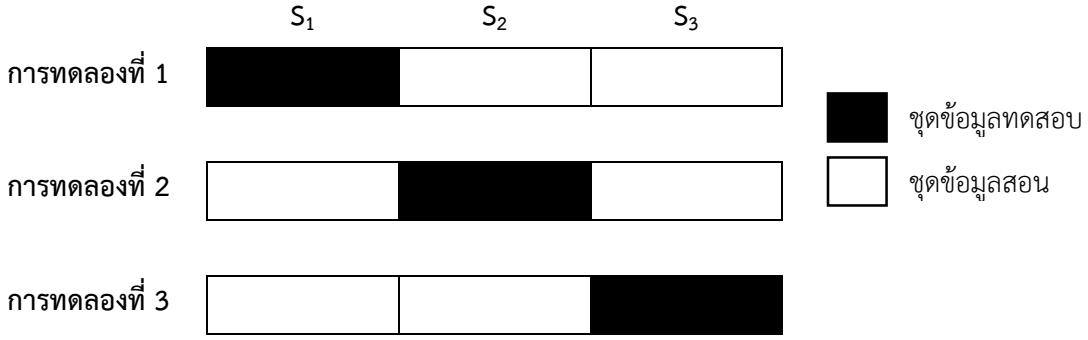
2) การแบ่งข้อมูลแบบ k-fold Cross-Validation

การแบ่งข้อมูลแบบ k-fold Cross-Validation (Tan et al., 2019) เป็นวิธีที่มีการใช้กันอย่างแพร่หลาย ซึ่งมีจุดประสงค์เพื่อให้สามารถใช้ข้อมูลทั้งหมดได้อย่างมีประสิทธิภาพทั้งข้อมูลสอนสำหรับการเรียนรู้และข้อมูลทดสอบสำหรับทดสอบเพื่อประเมินประสิทธิภาพของโมเดลที่ได้

วิธีทั่วไปในการแบ่งข้อมูลออกเป็น k ส่วนโดยที่แต่ละส่วนที่มีขนาดเท่ากัน ส่วนหนึ่งของข้อมูลจะถูกเลือกให้เป็นข้อมูลทดสอบสำหรับการทดสอบเพื่อประเมินประสิทธิภาพของโมเดล และส่วนที่เหลือถูกใช้เป็นข้อมูลสอนสำหรับการเรียนรู้ข้อมูลเพื่อสร้างโมเดล โดยที่ขั้นตอนนี้จะทำซ้ำเป็นจำนวน k ครั้งที่มีการใช้ข้อมูลทดสอบที่แตกต่างกัน แล้วประเมินประสิทธิภาพของโมเดลที่ได้ในแต่ละครั้งโดยตัวชี้วัดประสิทธิภาพจะกล่าวในหัวข้อถัดไป

ทุกครั้งในการแบ่งสัดส่วนของข้อมูลจะถูกใช้สำหรับข้อมูลทดสอบเพียงหนึ่งส่วน และ (k-1) ส่วนสำหรับข้อมูลสอน ยกตัวอย่างวิธีการแบ่งข้อมูลนี้อย่างง่าย สมมติว่าแบ่งชุดข้อมูลทั้งหมดออกเป็นชุดย่อยเท่าๆ กันสามส่วน คือ S1, S2 และ S3 แสดงดังรูปที่ 2.21 ในที่นี้จะมีการทดลองทั้งหมด 3 ครั้งสำหรับการทดลองครั้งแรกโมเดลจะเรียนรู้ข้อมูลโดยใช้ชุดย่อย S2 และ S3 และทดสอบ

ประสิทธิภาพของโมเดลโดยใช้ชุด S1 แล้วนำผลการทดสอบที่ได้ไปคำนวณค่าตัวชี้วัดประสิทธิภาพของโมเดลที่ได้ในการทดลองครั้งนี้ ในทำนองเดียวกันสำหรับการทดสอบครั้งที่สอง เราใช้ S1 และ S3 เป็นชุดข้อมูลสอนสำหรับสร้างโมเดล และ S2 เป็นชุดข้อมูลทดสอบ เพื่อคำนวณหาค่าตัวชี้วัดประสิทธิภาพในการทดสอบครั้งที่สองได้จากชุดย่อย S2 สุดท้ายเราใช้ S1 และ S2 สำหรับเป็นชุดข้อมูลสอนในการทดลองครั้งที่สาม และ S3 สำหรับชุดข้อมูลทดสอบ แล้วคำนวณค่าตัวชี้วัดประสิทธิภาพสำหรับการทดลองครั้งที่สาม จากนั้นประสิทธิภาพโดยรวมนั้นได้มาจากการรวมค่าตัวชี้วัดประสิทธิภาพที่ได้จากทุกการทดลองและหารด้วยขนาดของสัดส่วนที่ใช้ในการแบ่งข้อมูล วิธีนี้เรียกว่า 3-fold Cross-Validation



รูปที่ 2.21 ตัวอย่างการแบ่งข้อมูลแบบ 3-fold Cross-Validation

ในการเลือกค่า k ใน k-fold Cross-Validation จะขึ้นอยู่กับลักษณะของปัญหา ถ้า k มีค่าน้อยจะส่งผลให้ชุดข้อมูลสอนมีขนาดเล็กลงซึ่งจะทำให้อาจเกิดความคลาดเคลื่อนมากขึ้น ในทางกลับกันถ้า k มีค่าที่สูงจะส่งผลให้ชุดสอนมีขนาดที่ใหญ่ขึ้นซึ่งจะช่วยลดความคลาดเคลื่อนที่เกิดขึ้นได้ อย่างไรก็ตามสำหรับการวิเคราะห์ในงานวิจัยส่วนใหญ่จะเลือก k อยู่ระหว่าง 5 ถึง 10 (Tan et al., 2019) ซึ่งเป็นค่าที่เหมาะสมสำหรับการประเมินประสิทธิภาพ กล่าวคือในแต่ละครั้งก็จะสามารถใช้ข้อมูลสำหรับการสอนได้ถึง 80% ถึง 90% นั่นเอง

บทที่ 3

การสกัดคุณลักษณะแทนข้อความเพื่อจำแนกขั้วความคิดเห็น

ในงานวิจัยนี้ สนใจศึกษาการวิเคราะห์และสกัดคุณลักษณะแทนข้อความการแสดงความคิดเห็นที่ปรากฏบนสื่อออนไลน์ เช่น การแสดงความคิดเห็นต่อสินค้าและบริการ ซึ่งข้อความเหล่านั้นสามารถจำแนกออกเป็นขั้วความคิดเห็นเชิงบวก (Positive) หรือขั้วความคิดเห็นเชิงลบ (Negative) ได้ ดังตัวอย่างต่อไปนี้

ตารางที่ 3.1 ตัวอย่างข้อความแสดงความคิดเห็นและขั้วความคิดเห็น

ข้อความแสดงความคิดเห็น	ขั้วความคิดเห็น
The case is great and works fine with the 680.	Positive
If you are Razr owner...you must have this!	Positive
He was very impressed when going from the original battery to the extended battery.	Positive
I didn't think that the instructions provided were helpful to me.	Negative
The design is very odd, as the ear "clip" is not very comfortable at all.	Negative
I bought it for my mother and she had a problem with the battery.	Negative

การสกัดคุณลักษณะแทนข้อความแสดงความคิดเห็นเหล่านี้ สามารถใช้วิธีแบบดั้งเดิมที่ได้กล่าวไว้ในบทที่ 2 กล่าวคือ แทนข้อความด้วยเวกเตอร์ที่มีจำนวนมิติเท่ากับจำนวนคำศัพท์ที่จะมีได้ในข้อความทั้งหมด ซึ่งจะทำให้เวกเตอร์แทนข้อความมีมิติที่สูงมากนำไปสู่การสิ้นเปลืองเนื้อที่ในการเก็บข้อมูลและใช้เวลาในการประมวลผลที่นาน ในงานวิจัยจึงได้คิดค้นและพัฒนาการสกัดคุณลักษณะแทนข้อความเพื่อเพิ่มประสิทธิภาพให้ได้ผลการจำแนกขั้วความคิดเห็น ทั้งประสิทธิภาพในแง่ของความถูกต้องในการจำแนก และประสิทธิภาพในแง่ของการใช้พื้นที่และเวลาในการประมวลผล โดยเสนอการสกัดคุณลักษณะแทนข้อความ 2 วิธี ดังรายละเอียดที่จะกล่าวต่อไป

3.1 การสกัดคุณลักษณะแทนข้อความ

ในวิทยานิพนธ์นี้ เสนอการแทนข้อความด้วยเวกเตอร์มิติน้อย ที่จะต้องประกอบด้วยคุณลักษณะที่สื่อถึงความคิดเห็นของข้อความที่นำมาวิเคราะห์ 2 รูปแบบ คือ เวกเตอร์ 4 มิติและเวกเตอร์ 8 มิติ ดังหลักการและวิธีการสร้างที่จะกล่าวต่อไปนี้

3.1.1 คุณลักษณะแทนข้อความแบบเวกเตอร์ 4 มิติ

ข้อความแสดงความคิดเห็น มีความแตกต่างจากข้อความในหนังสือ หรือบทความทั่วๆ ไป กล่าวคือ ข้อความแสดงความคิดเห็นจะสื่อถึงทิศทางของความคิดเห็นว่าเป็นขั้วความคิดเห็นเชิงบวกหรือขั้วความคิดเห็นเชิงลบ ข้อความแสดงความคิดเห็นเหล่านี้มักจะประกอบด้วยคำศัพท์เชิงบวก เช่น good love และ like เป็นต้น และคำศัพท์เชิงลบ เช่น bad poor และ boring เป็นต้น ซึ่งมีนักวิจัยได้วิเคราะห์รวมรวมคลังคำเชิงบวกและคำเชิงลบ (Pang & Lee, 2004) ที่เป็นภาษาอังกฤษไว้ให้ผู้สนใจได้นำไปใช้ในการวิเคราะห์ข้อความต่อไป ในงานวิจัยนี้ จึงได้สร้างเวกเตอร์แทนข้อความแสดงความคิดเห็น โดยพิจารณาจากคำเชิงบวกและคำเชิงลบที่ปรากฏในข้อความแสดงความคิดเห็น โดยมีขั้นตอนวิธีดัง **Algorithm 1** ในรูปที่ 3.1

Algorithm 1 : Feature Extraction -V4D

กำหนดให้ $\mathbf{D}_{positive}$ แทนคลังของคำศัพท์เชิงบวก

$\mathbf{D}_{negative}$ แทนคลังของคำศัพท์เชิงลบ

ข้อมูลเข้า: $\mathbf{T} = \{T_1, T_2, ..., T_n\}$ แทน เซตของข้อความแสดงความคิดเห็น

ข้อมูลผลลัพธ์: $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ แทน เซตของเวกเตอร์ 4 มิติ

1: สำหรับแต่ละ $T_i \in \mathbf{T}$ โดยที่ $i = 1, 2, ..., n$

2: $word\_list_i = \text{TextPrePocessing}(T_i)$

3: $u_1 = 0,\ u_2 = 0$

4: สำหรับแต่ละ $word \in word\_list_i$

5: ถ้า $word \in \mathbf{D}_{positive}$

6: $u_1 = u_1 + 1$

7: หรือ มิฉะนั้น ถ้า $word \in \mathbf{D}_{negative}$

8: $u_2 = u_2 + 1$

9: $u_3 = u_1 + u_2$

10: $u_4 = u_1 - u_2$

11: $\mathbf{v}_i = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T$

12. $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$

13: Return $\mathbf{V}$

14: **Procedure:** $\text{TextPrePocessing1}(T)$

15: $T1$ = ข้อความ T ที่ได้มีการแทนที่อักขระตัวพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก

16: $T2$ = ข้อความ $T1$ ที่ได้มีกำจัดอักขระพิเศษและตัวเลข

17: $T3$ = ข้อความ $T2$ ที่ได้มีกำจัดคำหยุด

18: $word\_list$ = รายการคำที่ได้แบ่งข้อความ $T3$ ด้วยช่องว่าง

19: Return $word\_list$

รูปที่ 3.1 ขั้นตอนวิธีการสกัดคุณลักษณะแบบเวกเตอร์ V4D

การสร้างเวกเตอร์แทนข้อความดัง **Algorithm 1** มีการนำเข้าข้อมูล เซตของข้อความ คลังคำศัพท์เชิงบวก และคลังคำศัพท์เชิงลบ และได้ผลลัพธ์เป็นเวกเตอร์ 4 มิติ แต่ละข้อความจะถูกนำมาสร้างเป็นเวกเตอร์ ตามกระบวนการในบรรทัดที่ 2-11 แต่ละข้อความจะถูกทำความสะอาดเพื่อกำจัดตัวเลข อักขระพิเศษ และคำหยุด ดังขั้นตอนในกระบวนการ TextPrePocessing1 ดังบรรทัดที่ 14-19 ซึ่งจะได้ผลลัพธ์เป็นรายการคำที่มีอยู่ในข้อความ จากนั้นหาค่าใน 2 มิติแรกของเวกเตอร์ ได้แก่ ค่าความถี่ของคำศัพท์เชิงบวกและค่าความถี่ของคำศัพท์เชิงบวกที่ปรากฏในข้อความที่พิจารณา ดังขั้นตอนในบรรทัดที่ 3-8 สำหรับค่าในมิติที่ 3 ของเวกเตอร์ คือค่าผลรวมของค่าความถี่ของคำศัพท์เชิงบวกและค่าความถี่ของคำศัพท์เชิงลบ สำหรับค่าในมิติที่ 4 ของเวกเตอร์ คือค่าผลต่างของค่าความถี่ของคำศัพท์เชิงบวกและค่าความถี่ของคำศัพท์เชิงลบ ดังขั้นตอนในบรรทัดที่ 9 และบรรทัดที่ 10 ตามลำดับ ทั้งนี้ขั้นตอนวิธีใน **Algorithm 1** ได้ถูกนำไปใช้ในการสร้างเวกเตอร์แทนข้อความสำหรับจำแนกขั้วความคิดเห็น และผลการทดลองเบื้องต้นมีการนำเสนอในบทความวิจัย (ณิชาภัทร และนิวรรณ, 2561)

3.1.2 คุณลักษณะแทนข้อความแบบเวกเตอร์ 8 มิติ

เมื่อพิจารณาข้อความต่างๆ บนสื่อออนไลน์มักจะมีคำว่า “no” หรือ “not” ซึ่งในการประมวลผลข้อความโดยทั่วไปจะมีการกำจัดคำหยุดออกไปก่อนและเซตของคำหยุดมาตรฐานจะประกอบด้วยคำศัพท์ “no” และ “not” แต่คำหยุดเหล่านี้มักจะมีนัยสำคัญในการบ่งบอกว่าข้อความนั้นๆ เป็นการแสดงความคิดเห็นในทิศทางเชิงลบหรือเชิงบวก เช่น I do not like the product จะเห็นว่า ข้อความดังกล่าวเป็นการแสดงความคิดเห็นเชิงลบ หากกำจัดคำหยุด “not” ออกไป ข้อความดังกล่าวจะกลายเป็นการแสดงความคิดเห็นเชิงบวก หรือแม้กระทั่งหากยังคงคำว่า “not” ไว้ แต่ละคำจะถูกพิจารณาแยกจากกันโดยช่องว่าง จะเห็นว่าคำว่า “like” ในข้อความยังแสดงถึงคุณลักษณะความคิดเห็นเชิงบวก ผู้วิจัยมีข้อสังเกตว่า คำว่า “not” และคำว่า “like” ต้องพิจารณาในกรณีที่มีการคำได้ถูกว่าต่อเนื่องกันเป็น “not like” จึงจะเป็นคุณลักษณะความคิดเห็นเชิงลบ นอกจากนี้ ในข้อความแสดงความคิดเห็นเขียนคำว่า “n’t” แทนการเขียน “not” ดังนั้นในการประมวลผลข้อความก่อนนำมาสร้างเวกเตอร์ จะต้องมีการเปลี่ยนรูป “n’t” ให้เป็น “not” จากแนวคิดดังที่ได้กล่าวมา จึงได้เสนอขั้นตอนวิธีในการสกัดคุณลักษณะแทนข้อความดัง **Algorithm 2** ในรูปที่ 3.2

Algorithm 2 : Feature Extraction –V8D

กำหนดให้ $\mathbf{D}_{positive}$ แทนเซตของคำศัพท์เชิงบวก และ $\mathbf{D}_{negative}$ แทนเซตของคำศัพท์เชิงลบ

ข้อมูลเข้า: $\mathbf{T} = \{T_1, T_2, ..., T_n\}$ แทน เซตของข้อความแสดงความคิดเห็น

ข้อมูลผลลัพธ์: $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ แทน เซตของเวกเตอร์ 8 มิติ

1: สำหรับแต่ละ $T_i \in \mathbf{T}$ โดยที่ $i = 1, 2, ..., n$

2: $word\_list_i = \text{TextPrePocessing2}(T_i)$

3: $u_1 = 0,\ u_2 = 0,\ u_5 = 0,\ u_6 = 0,\ u_7 = 0,\ u_8 = 0$

4: สำหรับแต่ละ $word_{i,j} \in word\_list_i$ โดยที่ $j = 1, 2, ..., p$

5: ถ้า $word_{i,j} \in \mathbf{D}_{positive}$

6: $u_1 = u_1 + 1$

7: หรือ มิฉะนั้น ถ้า $word_{i,j} \in \mathbf{D}_{negative}$

8: $u_2 = u_2 + 1$

9: $u_3 = u_1 + u_2$

10: $u_4 = u_1 - u_2$

11: สำหรับแต่ละ $word_{i,j} \in word\_list_i$ โดยที่ $j = 2, 3, ..., p$

12. ถ้า $word_{i,j-1} == \text{'no'}$ และ $word_{i,j} \in \mathbf{D}_{positive}$

13: $u_5 = u_5 + 1$

14: หรือ มิฉะนั้น ถ้า $word_{i,j-1} == \text{'not'}$ และ $word_{i,j} \in \mathbf{D}_{positive}$

15: $u_6 = u_6 + 1$

16: หรือ มิฉะนั้น ถ้า $word_{i,j-1} == \text{'no'}$ และ $word_{i,j} \in \mathbf{D}_{negative}$

17: $u_7 = u_7 + 1$

18: หรือ มิฉะนั้น ถ้า $word_{i,j-1} == \text{'not'}$ และ $word_{i,j} \in \mathbf{D}_{negative}$

19: $u_8 = u_8 + 1$

20: $\mathbf{v}_i = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \end{bmatrix}^T$

21. $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$

22: Return $\mathbf{V}$

23: **Procedure:** $\text{TextPrePocessing2}(T)$

24: $T1$ = ข้อความ T ที่ได้มีการแทนที่อักขระตัวพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก

25: $T2$ = ข้อความ $T1$ ที่ได้มีการแทนที่ n’t ด้วย not

26: $T3$ = ข้อความ $T2$ ที่ได้มีการกำจัดอักขระพิเศษ และตัวเลข

27: $T4$ = ข้อความ $T3$ ที่ได้มีกำจัดคำหยุด ยกเว้น no และ not

28: $word\_list$ = รายการคำที่ได้แบ่งข้อความ $T4$ ด้วยช่องว่าง

29: Return $word\_list$

รูปที่ 3.2 ขั้นตอนวิธีการสกัดคุณลักษณะแบบเวกเตอร์ V8D

การสร้างเวกเตอร์แทนข้อความดัง **Algorithm 2** มีการนำเข้าข้อมูล เซตของข้อความ คลังคำศัพท์เชิงบวก และเซตของคำศัพท์เชิงลบโดยที่มีการเพิ่มคำว่า “no” และ “not” ในคลังคำศัพท์เชิงลบ ผลลัพธ์ที่ได้จาก **Algorithm 2** เป็นคุณลักษณะแทนแต่ละข้อความซึ่งอยู่ในรูปของเวกเตอร์ 8 มิติ แต่ละข้อความจะถูกนำมาสร้างเป็นเวกเตอร์ ตามกระบวนการในบรรทัดที่ 2-20 แต่ละข้อความจะถูกทำความสะอาดเพื่อกำจัดตัวเลข อักขระพิเศษ และคำหยุด ดังขั้นตอนในกระบวนการ TextPrePocessing 2 ดังบรรทัดที่ 23-29 ซึ่งจะได้ผลลัพธ์เป็นรายการคำที่มีอยู่ในข้อความ จากนั้นหาค่าใน 4 มิติแรกของเวกเตอร์เช่นเดียวกับวิธีใน **Algorithm 1** สำหรับมิติที่ 5 เป็นการหาค่าความถี่ของรูปแบบ คำว่า “no” แล้วตามด้วย คำศัพท์เชิงบวก ดังบรรทัดที่ 12-13 สำหรับมิติที่ 6 เป็นการหาค่าความถี่ของรูปแบบ คำว่า “not” แล้วตามด้วย คำศัพท์เชิงบวก ดังบรรทัดที่ 14-15 สำหรับมิติที่ 7 เป็นการหาค่าความถี่ของรูปแบบ คำว่า “no” แล้วตามด้วย คำศัพท์เชิงบวก ดังบรรทัดที่ 16-17 สำหรับมิติที่ 6 เป็นการหาค่าความถี่ของรูปแบบ คำว่า “not” แล้วตามด้วย คำศัพท์เชิงบวก ดังบรรทัดที่ 18-19

เมื่อได้เวกเตอร์แทนข้อความโดยใช้วิธีใน **Algorithm 1** หรือ **Algorithm 2** แล้ว เวกเตอร์เหล่านั้นจะถูกนำมาสร้างโมเดลในการจำแนกหรือทำนายขั้วความคิดเห็นต่อไป

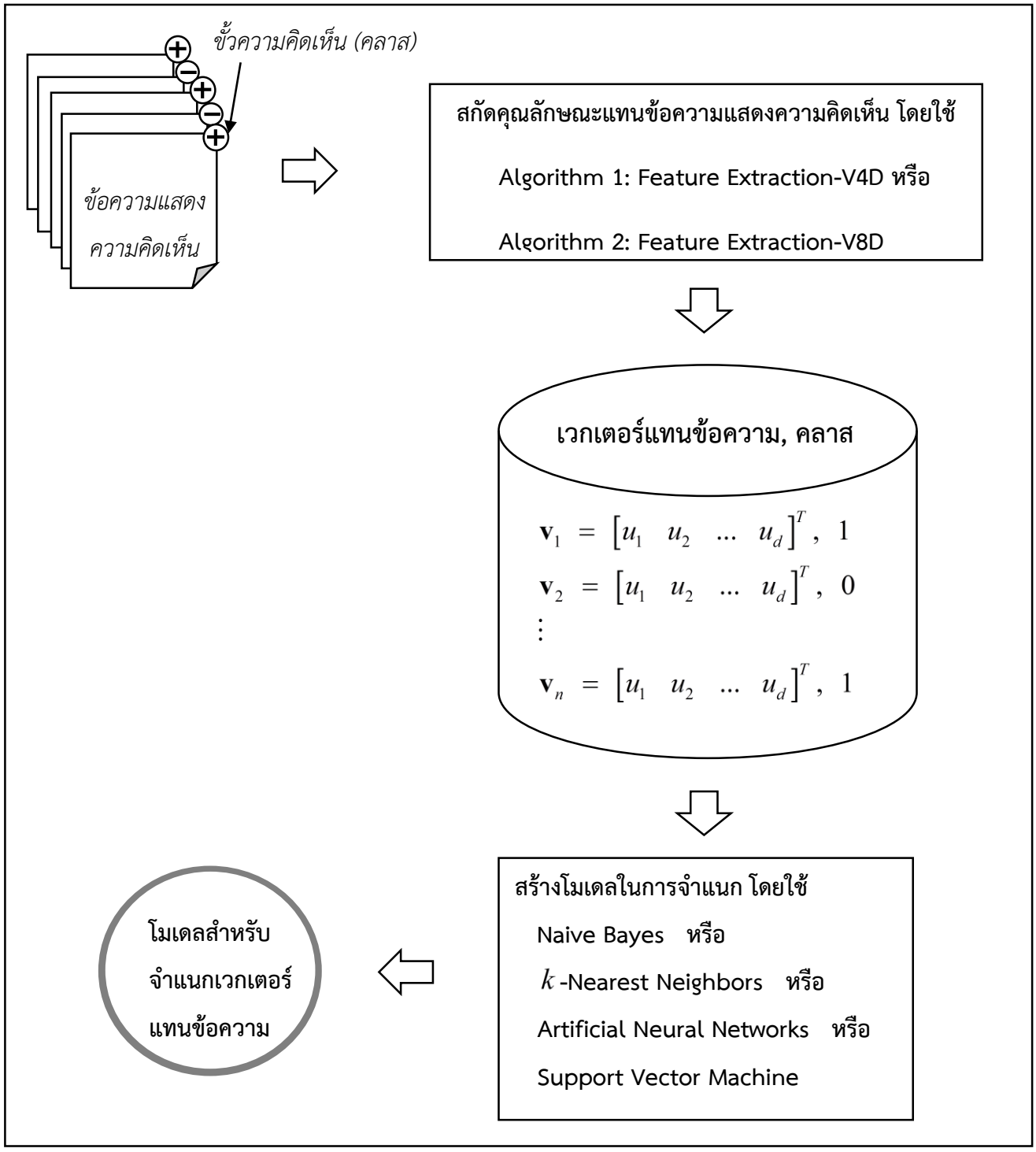
3.2 การสร้างและใช้โมเดลจำแนกข้อความแสดงความคิดเห็น

ในหัวข้อนี้จะกล่าวถึง การนำวิธีการสกัดคุณลักษณะแทนข้อความตามขั้นตอนวิธีที่ได้กล่าวในหัวข้อ 3.1 มาใช้ในการเตรียมข้อมูลนำเข้าเพื่อสร้างโมเดลสำหรับจำแนกข้อความแสดงความคิดเห็น และการนำโมเดลที่ได้ไปใช้ในการทำนายหรือจำแนกข้อความแสดงความคิดเห็น ดังรายละเอียดที่จะกล่าวต่อไปนี้

3.2.1 การสร้างโมเดลจำแนกข้อความแสดงความคิดเห็น

ข้อมูลที่นำมาใช้ในการสร้างโมเดล ประกอบด้วย 2 ส่วน คือ เวกเตอร์แทนข้อความ และขั้วความคิดเห็นของข้อความ เรียกว่า คลาส โดยเวกเตอร์แทนข้อความได้มากจากการนำข้อความแสดงความคิดเห็นมาผ่านขั้นตอนวิธีการเตรียมข้อความหรือทำความสะอาดข้อความและการสกัดคุณลักษณะเพื่อให้ได้เวกเตอร์แทนข้อความ โดยในงานวิจัยมีได้เสนอการสกัดคุณลักษณะ 2 วิธี ดังที่ได้แสดงไว้ใน Algorithm 1 และ Algorithm 2 จากนั้นเซตของเวกเตอร์แทนข้อความที่ได้จะเป็นข้อมูลนำเข้าเพื่อสร้างโมเดลสำหรับจำแนก โดยใช้วิธี Naive Bayes หรือ k -Nearest Neighbors

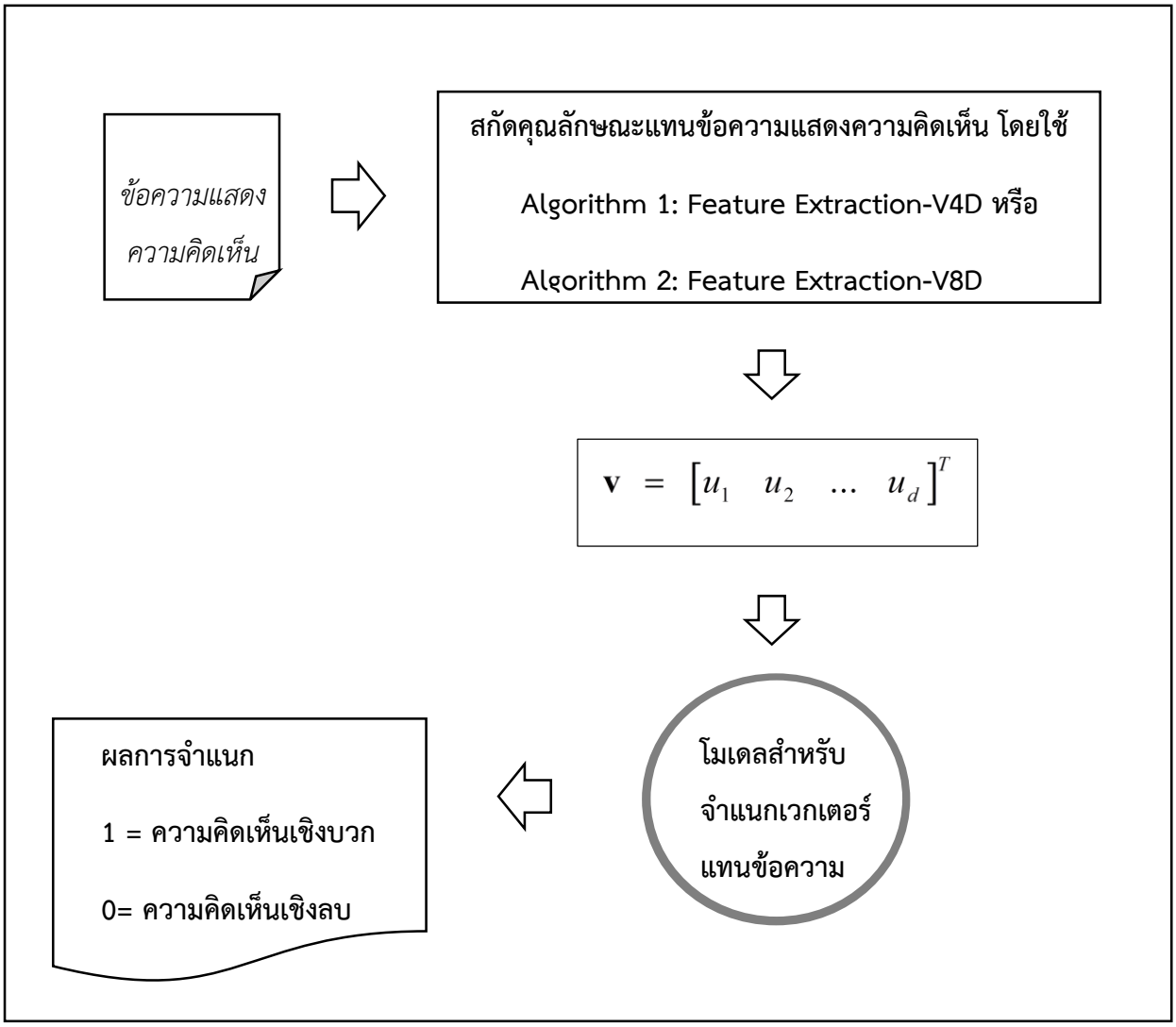
หรือ Artificial Neural Networks หรือ Support Vector Machine ภาพรวมของการสร้างโมเดล มีกระบวนการทำงานดังรูปที่ 3.3



รูปที่ 3.3 ภาพรวมของกระบวนการสร้างโมเดล

3.2.2 การใช้โมเดลจำแนกข้อความแสดงความคิดเห็น

ในการจำแนกข้อความแสดงความคิดเห็น จะมีการนำเข้าข้อความที่ต้องการจำแนก แล้วทำการสกัดคุณลักษณะแทนข้อความเช่นเดียวกันกับวิธีที่ใช้ในขั้นตอนการสร้างโมเดลเพื่อให้ได้เวกเตอร์แทนข้อความ จากนั้นนำเข้าสู่โมเดลที่สร้างไว้เพื่อจำแนกได้ผลลัพธ์ว่าข้อความที่นำเข้าเป็นขั้วความคิดเห็นเชิงบวกหรือขั้วความคิดเห็นเชิงลบ ภาพรวมของการใช้โมเดลมีกระบวนการทำงานดังรูปที่ 3.4



รูปที่ 3.4 ภาพรวมของกระบวนการใช้โมเดลที่สร้างขึ้น

บทที่ 4

การทดลองและผลการทดลอง

บทนี้กล่าวถึงการออกแบบการทดลองและได้นำเสนอผลลัพธ์ที่ได้จากการทดลองการสกัดคุณลักษณะแทนข้อความที่มีประสิทธิภาพเพื่อการจำแนกขั้วความคิดเห็น โดยทำการทดลองบนชุดข้อมูลที่เป็นข้อความเกี่ยวกับการแสดงความคิดเห็นบนสื่อออนไลน์และประเมินผลของการสกัดคุณลักษณะแทนข้อความเพื่อการจำแนกขั้วความคิดเห็น โดยใช้โปรแกรม R Studio

4.1 การทดลอง

ในหัวข้อนี้จะกล่าวถึง รายละเอียดชุดข้อมูลที่นำมาใช้ในการทดลอง วิธีการสกัดคุณลักษณะที่ทำการทดลองเปรียบเทียบ วิธีการจำแนกที่ใช้เป็นเครื่องมือในการวัดประสิทธิภาพการสกัดคุณลักษณะ และตัวชี้วัดประสิทธิภาพที่ใช้ มีรายละเอียดดังต่อไปนี้

4.1.1 ชุดข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลองมี 2 ส่วน ประกอบด้วย ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นและชุดข้อมูลที่เป็นคลังคำเชิงบวกและคลังคำเชิงลบ

1) ชุดข้อมูลข้อความแสดงความคิดเห็น

ชุดข้อมูลที่นำมาใช้ในการทดลองได้มาจากข้อความแสดงความคิดเห็นเกี่ยวกับผลิตภัณฑ์ การบริการ และการสนทนาของผู้ที่ใช้สื่อสังคมออนไลน์ เพื่อนำมาใช้สำหรับการวิจัย ประกอบด้วย ชุดข้อมูล 8 ชุดข้อมูล แต่ละชุดข้อมูลประกอบด้วยข้อความแสดงความคิดเห็นที่เป็นข้อความเชิงบวกและข้อความเชิงลบ โดยแต่ละชุดข้อมูลมาจากความหลากหลายของโดเมน มีรายละเอียดดังนี้

- **ชุดข้อมูล DS1 - Amazon**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับสินค้าประเภทโทรศัพท์มือถือที่มาจากเว็บไซต์ amazon.com โดยมีข้อความแสดงความคิดเห็นทั้งหมด 1,000 ข้อความ ประกอบด้วยข้อความเชิงบวก 500 ข้อความ และข้อความเชิงลบ 500 ข้อความ (Pang & Lee, 2004)

- **ชุดข้อมูล DS2 - IMDb**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับภาพยนตร์ที่มาจากเว็บไซต์ imdb.com โดยมีข้อความแสดงความคิดเห็นทั้งหมด 1,000 ข้อความ ประกอบด้วย ข้อความเชิงบวก 500 ข้อความ และข้อความเชิงลบ 500 ข้อความ (Pang & Lee, 2004)

- **ชุดข้อมูล DS3 - Yelp**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับร้านอาหารที่มาจากเว็บไซต์ yelp.com โดยมีข้อความแสดงความคิดเห็นทั้งหมด 1,000 ข้อความ ประกอบด้วย ข้อความเชิงบวก 500 ข้อความ และข้อความเชิงลบ 500 ข้อความ (Pang & Lee, 2004)

- **ชุดข้อมูล DS4 - Apparel**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับสินค้าประเภทเครื่องนุ่งห่มที่มาจากเว็บไซต์ amazon.com โดยมีข้อความแสดงความคิดเห็นทั้งหมด 2,000 ข้อความ ประกอบด้วย ข้อความเชิงบวก 1,000 ข้อความ และข้อความเชิงลบ 1,000 ข้อความ (Blitzer et al., 2007)

- **ชุดข้อมูล DS5 – Health**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับสินค้าประเภทอุปกรณ์สุขภาพและของใช้ส่วนตัวที่มาจากเว็บไซต์ amazon.com โดยมีข้อความแสดงความคิดเห็นทั้งหมด 2,000 ข้อความ ประกอบด้วย ข้อความเชิงบวก 1,000 ข้อความ และข้อความเชิงลบ 1,000 ข้อความ (Blitzer et al., 2007)

- **ชุดข้อมูล DS6 - Music**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับสินค้าประเภทอุปกรณ์เพลงที่มาจากเว็บไซต์ amazon.com โดยมีข้อความแสดงความคิดเห็นทั้งหมด 2,000 ข้อความ ประกอบด้วย ข้อความเชิงบวก 1,000 ข้อความ และข้อความเชิงลบ 1,000 ข้อความ (Blitzer et al., 2007)

- **ชุดข้อมูล DS7 - Sports**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับสินค้าประเภทอุปกรณ์กีฬาและกิจกรรมกลางแจ้งที่มาจากเว็บไซต์ amazon.com โดยมีข้อความแสดงความคิดเห็นทั้งหมด 2,000 ข้อความ ประกอบด้วย ข้อความเชิงบวก 1,000 ข้อความ และข้อความเชิงลบ 1,000 ข้อความ (Blitzer et al., 2007)

- **ชุดข้อมูล DS8 - US Airline**

ชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นเกี่ยวกับปัญหาของแต่ละสายการบินที่สำคัญในสหรัฐฯ มีทั้งหมด 6 สายการบิน ได้แก่ American, Delta, Southwest, United, US Airways และ

Virgin American ซึ่งชุดข้อมูลนี้เป็นข้อความของเดือนกุมภาพันธ์ พ.ศ. 2558 (ค.ศ. 2015) โดยมีข้อความแสดงความคิดเห็นทั้งหมด 11,541 ข้อความ ประกอบด้วยข้อความเชิงบวก 2,363 ข้อความ และข้อความเชิงลบ 9,178 ข้อความ (Kaggle, 2019)

ข้อมูลข้อความแสดงความคิดเห็นที่ใช้ในการทดลองทั้งหมด สรุปได้ดังตารางที่ 4.1

ตารางที่ 4.1 ชุดข้อมูลที่ใช้ในการทดลอง

ข้อมูล	จำนวนข้อความเชิงบวก	จำนวนข้อความเชิงลบ
DS1 - Amazon	500	500
DS2 - IMDb	500	500
DS3 - Yelp	500	500
DS4 - Apparel	1,000	1,000
DS5 – Health	1,000	1,000
DS6 - Music	1,000	1,000
DS7 - Sports	1,000	1,000
DS8 - US Airline	2,363	9,178

2) คลังคำเชิงบวกและคลังคำเชิงลบ

ชุดข้อมูลที่เป็นคลังคำศัพท์ที่มีความหมายในทิศทางเชิงบวกและทิศทางเชิงลบ ถูกสร้างโดย Hu และ Liu (Hu & Liu, 2004) โดยคลังคำศัพท์ประกอบด้วยคำศัพท์เชิงบวกจำนวน 2,006 คำ และคำศัพท์เชิงลบจำนวน 4,783 คำ ซึ่งในงานวิจัยได้นำคลังคำเชิงบวกและคลังคำเชิงลบมาใช้ในการวิเคราะห์เพื่อการสกัดคุณลักษณะแทนข้อความเพื่อการจำแนกขั้วความคิดเห็น

4.1.2 วิธีการสกัดคุณลักษณะที่ทำการทดลองเปรียบเทียบ

การวัดประสิทธิภาพของการสกัดคุณลักษณะแทนข้อความทำการทดลองโดยการนำคุณลักษณะแทนข้อความแบบดั้งเดิม นั่นคือเวกเตอร์ TF และเวกเตอร์ TF-IDF โดยที่แต่ละเวกเตอร์มีขนาดเท่ากับจำนวนมิติของจำนวนคำศัพท์ในชุดข้อมูลสอน มาทำการเปรียบเทียบกับคุณลักษณะแทนข้อความที่นำเสนอ นั่นคือเวกเตอร์ 4 มิติ ให้ชื่อว่า V4D และเวกเตอร์ 8 มิติ ให้ชื่อว่า V8D โดยที่แต่ละเวกเตอร์มีขนาดเท่ากับ 4 มิติและ 8 มิติตามลำดับ

4.1.3 วิธีการจำแนกที่ใช้เป็นเครื่องมือในการวัดประสิทธิภาพการสกัดคุณลักษณะ

การวัดประสิทธิภาพประสิทธิภาพการจำแนกขั้วความคิดเห็นของคุณลักษณะแทนข้อความจะใช้วิธีการจำแนก 4 วิธี ดังนี้

1) วิธีการจำแนก k-Nearest Neighbor (k-NN) เป็นวิธีการที่จำแนกประเภทขั้วความคิดเห็นโดยการวัดระยะห่างระหว่างข้อมูลทดสอบกับข้อมูลสอนด้วยใช้การวัดแบบยูคลิด (Euclidean distance) จากนั้นเลือกข้อมูลสอนที่ใกล้เคียงที่สุด k ตัว โดยกำหนดค่าพารามิเตอร์ k ในการทดสอบทั้งหมด 3 ค่า คือ k เท่ากับ 3 5 และ 7

2) วิธีการจำแนก Naive Bayes (NB) เป็นวิธีการที่จำแนกประเภทขั้วความคิดเห็นโดยการคำนวณหลักความน่าจะเป็นของทฤษฎีของเบย์ (Bayes theorem)

3) วิธีการจำแนก Artificial Neural Networks (ANN) เป็นวิธีการที่จำแนกประเภทขั้วความคิดเห็นโดยโครงข่ายประสามเทียมที่ใช้เทคนิคการเรียนรู้ด้วยสมการเชิงเส้นที่เชื่อมต่อกับจำนวนนิวรอนหลายๆ โหนดและจำนวนชั้นของโครงข่ายประสาทเทียมหลายๆ ชั้น และแต่ละนิวรอนจะมีการคำนวณโดยฟังก์ชันต่างๆ เพื่อปรับค่าผลลัพธ์ข้อมูลให้ได้ดีที่สุด ในการทดลองจะกำหนดจำนวนชั้นซ่อนเท่ากับจำนวน 1 ชั้นและ 2 ชั้น และกำหนดจำนวนโหนดของนิวรอนเท่ากับจำนวนมิติของข้อมูลนำเข้าและครึ่งหนึ่งของจำนวนมิติของข้อมูลนำเข้า โดยใช้ฟังก์ชัน logistic สำหรับใช้คำนวณในการปรับค่าผลลัพธ์ข้อมูลในนิวรอน โดยวิธีการจำแนก ANN มีการทดลอง 4 รูปแบบ ดังนี้

เมื่อกำหนดให้ m แทนจำนวนโหนดของนิวรอนที่เท่ากับจำนวนของมิติของข้อมูล

- [m] คือ มีจำนวนชั้นซ่อน 1 ชั้น โดยที่จำนวนโหนดของนิวรอนเท่ากับจำนวนมิติของข้อมูลนำเข้า
- [m/2] คือ มีจำนวนชั้นซ่อน 1 ชั้น โดยที่จำนวนโหนดของนิวรอนเท่ากับครึ่งหนึ่งของจำนวนมิติทั้งหมดของข้อมูลนำเข้า
- [m,m] คือ มีจำนวนชั้นซ่อน 2 ชั้น โดยชั้นที่แต่ละชั้น มีจำนวนโหนดของนิวรอนเท่ากับจำนวนมิติทั้งหมดของข้อมูลนำเข้า
- [m,m/2] คือ มี hidden layer 2 ชั้น โดยชั้นที่ 1 มีจำนวนโหนดของนิวรอนเท่ากับจำนวนมิติทั้งหมดของข้อมูลนำเข้าและชั้นที่ 2 มีจำนวนโหนดของนิวรอนเท่ากับครึ่งหนึ่งของจำนวนมิติทั้งหมดของข้อมูลนำเข้า

4) วิธีการจำแนก Support Vector Machine (SVM) เป็นวิธีการที่จำแนกประเภทขั้วความคิดเห็นโดยใช้เทคนิคการคำนวณสมการเพื่อสร้างเส้นที่สามารถแบ่งกลุ่มชุดข้อมูลสอนได้ดีที่สุด โดยในการทดลองจะเลือกสมการ linear เพื่อสร้างเส้นแบ่งชุดข้อมูลและกำหนดค่าพารามิเตอร์ c ซึ่งค่ากำหนดความยืดหยุ่นของระยะขอบ (soft margin) ในการทดสอบทั้งหมด 5 ค่า คือ c เท่ากับ 0.01 0.1 1 10 และ 100

4.1.4 ตัวชี้วัดประสิทธิภาพที่ใช้

การวัดประสิทธิภาพของการสกัดคุณลักษณะแทนข้อความ จะพิจารณาจากผลการจำแนกขั้วความคิดเห็นบนข้อมูลทดสอบที่ได้จากโมเดลที่สร้างขึ้น โดยทำการทดลองโดยแบ่งชุดข้อมูลออกเป็น 2 ส่วน คือชุดข้อมูลสอนและชุดข้อมูลทดสอบแบบ 5-fold Cross-Validation แล้วทำการประเมินผลการจำแนกซึ่งประกอบด้วย 2 คลาสที่เป็นไปได้ ได้แก่ คลาส 0 หมายถึง ความคิดเห็นเชิงลบ และคลาส 1 หมายถึง ความคิดเห็นเชิงบวก โดยจะพิจารณาผลการจำแนกจากโมเดลบนชุดข้อมูลทดสอบว่าเป็นคลาสใดเปรียบเทียบกับคลาสจริงที่ระบุไว้ในแต่ละข้อมูลทดสอบ การเปรียบเทียบคลาสดังกล่าวสามารถนำมาเขียนอยู่ในรูปของความสัมพันธ์ระหว่างคลาสจริง (Actual class) และคลาสจากการจำแนกหรือการทำนายของโมเดล (Predicted class) เรียกว่า Confusion Matrix ข้อมูลที่นำมาทดลองมี 2 คลาส ได้แก่ คลาส 0 และ คลาส 1 กำหนดสัญลักษณ์ดังนี้

- T_{00} คือ จำนวนข้อมูลทดสอบที่เป็นคลาส 0 และโมเดลจำแนกว่าเป็นคลาส 0
- T_{01} คือ จำนวนข้อมูลทดสอบที่เป็นคลาส 0 แต่โมเดลจำแนกว่าเป็นคลาส 1
- T_{10} คือ จำนวนข้อมูลทดสอบที่เป็นคลาส 1 แต่โมเดลจำแนกว่าเป็นคลาส 0
- T_{11} คือ จำนวนข้อมูลทดสอบที่เป็นคลาส 1 และโมเดลจำแนกว่าเป็นคลาส 1

จะได้ Confusion Matrix แสดงดังตารางที่ 4.2

ตารางที่ 4.2 ความสัมพันธ์ระหว่างคลาสจริงและคลาสจากการทำนาย (Confusion Matrix)

		คลาสจากข้อมูลทำนาย (Predicted)	
		คลาส = 0	**คลาส = 1**
คลาสจากข้อมูลทดสอบ (Actual)	**คลาส = 0**	T_{00}	T_{01}
	คลาส = 1	T_{10}	T_{11}

โมเดลที่มีประสิทธิภาพควรจะให้ค่า T_{00} และ T_{11} ที่สูง เพราะเป็นจำนวนของข้อมูลทดสอบที่โมเดลจำแนกได้ถูกต้อง เนื่องจากมีการวัดประสิทธิภาพของโมเดลอีกมากมาย ทั้งในเรื่องของความแม่นยำในการจำแนก ความถูกต้องในการจำแนก และความครบถ้วนในการจำแนก จึงมีการกำหนดตัววัดประสิทธิภาพของโมเดล ดังนี้

1) ค่า Accuracy

Accuracy หรือ ค่าความถูกต้อง เป็นตัววัดความสามารถในการจำแนกโดยรวม มีสูตรคำนวณดังนี้

$$\text{Accuracy} = \frac{T_{00}+T_{11}}{T_{00}+T_{01}+T_{10}+T_{11}} = \frac{\text{จำนวนข้อมูลทดสอบที่จำแนกได้ถูกต้อง}}{\text{จำนวนข้อมูลทดสอบทั้งหมด}} \qquad (4.1)$$

2) ค่า Precision

Precision หรือ ค่าความแม่นยำ เป็นตัววัดความแม่นยำของการจำแนกโดยพิจารณาจากผลการจำแนกของแต่ละคลาสที่สนใจ กำหนดให้ Precision_0 และ Precision_1 แทน Precision สำหรับคลาส 0 และ คลาส 1 ตามลำดับ มีสูตรคำนวณดังนี้

$$\text{Precision}_0 = \frac{T_{00}}{T_{00}+T_{10}} = \frac{\text{จำนวนข้อมูลทดสอบที่เป็นคลาส } 0 \text{ ที่โมเดลจำแนกได้ถูกต้อง}}{\text{จำนวนข้อมูลทดสอบที่โมเดลจำแนกว่าเป็นคลาส } 0} \qquad (4.2)$$

$$\text{Precision}_1 = \frac{T_{11}}{T_{11}+T_{01}} = \frac{\text{จำนวนข้อมูลทดสอบที่เป็นคลาส } 1 \text{ ที่โมเดลจำแนกได้ถูกต้อง}}{\text{จำนวนข้อมูลทดสอบที่โมเดลจำแนกว่าเป็นคลาส } 1} \qquad (4.3)$$

ค่าเฉลี่ย Precision ของทั้ง 2 คลาสคำนวณได้ดังนี้

$$\text{Precision} = \left(\frac{\text{Precision}_0 + \text{Precision}_1}{2} \right) \qquad (4.4)$$

3) ค่า Recall

Recall หรือ ค่าความครบถ้วนl เป็นตัววัดความครบถ้วนของการจำแนกโดยพิจารณาว่าข้อมูลแต่ละคลาสที่มีอยู่จริงในชุดข้อมูลทดสอบเมื่อนำมาทดสอบกับโมเดลแล้วโมเดลสามารถจำแนกได้ถูกต้องมากน้อยเพียงใด กำหนดให้ Recall_0 และ Recall_1 แทน Precision สำหรับคลาส 0 และคลาส 1 ตามลำดับ มีสูตรคำนวณดังนี้

$$\text{Recall}_0 = \frac{\text{T}_{00}}{\text{T}_{00} + \text{T}_{01}} = \frac{\text{จำนวนข้อมูลทดสอบที่เป็นคลาส 0 ที่โมเดลจำแนกได้ถูกต้อง}}{\text{จำนวนข้อมูลคลาส 0 ที่มีอยู่จริงในชุดข้อมูลทดสอบ}} \tag{4.5}$$

$$\text{Recall}_1 = \frac{\text{T}_{11}}{\text{T}_{11} + \text{T}_{10}} = \frac{\text{จำนวนข้อมูลทดสอบที่เป็นคลาส 1 ที่โมเดลจำแนกได้ถูกต้อง}}{\text{จำนวนข้อมูลคลาส 1 ที่มีอยู่จริงในชุดข้อมูลทดสอบ}} \tag{4.6}$$

ค่าเฉลี่ย Recall ของทั้ง 2 คลาสคำนวณได้ดังนี้

$$\text{Recall} = \left(\frac{\text{Recall}_0 + \text{Recall}_1}{2} \right) \tag{4.7}$$

4) ค่า F1

F1 หรือ ค่าเอฟ เป็นตัววัดประสิทธิภาพที่พิจารณาทั้ง Precision และ Recall ร่วมกัน โดยพิจารณาจากการจำแนกของแต่ละคลาสที่สนใจ กำหนดให้ F1_0 และ F1_1 แทน Precision สำหรับคลาส 0 และ คลาส 1 ตามลำดับ มีสูตรคำนวณดังนี้

$$\text{F1}_0 = \frac{2(\text{Precision}_0)(\text{Recall}_0)}{(\text{Precision}_0 + \text{Recall}_0)} \tag{4.8}$$

$$\text{F1}_1 = \frac{2(\text{Precision}_1)(\text{Recall}_1)}{(\text{Precision}_1 + \text{Recall}_1)} \tag{4.9}$$

ค่าเฉลี่ย F1 ของทั้ง 2 คลาสคำนวณได้ดังนี้

$$\text{F1} = \left(\frac{\text{F1}_0 + \text{F1}_1}{2} \right) \tag{4.10}$$

4.2 ผลการทดลอง

การทดลองการสกัดคุณลักษณะแทนข้อความที่มีประสิทธิภาพเพื่อการจำแนกขั้วความคิดเห็นด้วยชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็น 8 ชุดข้อมูล ที่ใช้ในการสร้างคุณลักษณะแทนข้อความแบบที่นำเสนอและคุณลักษณะแทนข้อความแบบดั้งเดิม โดยทำการทดลองเปรียบเทียบประสิทธิภาพของคุณลักษณะแทนข้อความด้วยเวกเตอร์ 4 รูปแบบ ได้แก่ เวกเตอร์ Term Frequency (TF) เวกเตอร์ Term Frequency – Inverse Document Frequency (TF-IDF) เวกเตอร์ 4 มิติที่นำเสนอ (V4D) และเวกเตอร์ 8 มิติที่นำเสนอ (V8D) เมื่อใช้วิธีการจำแนก 4 วิธี ได้แก่ วิธี k-Nearest Neighbors (k-NN) วิธี Naive Bayes (NB) วิธี Artificial Neural Networks (ANN) และ วิธี Support Vector Machine (SVM) ได้ผลการทดลองดังรายละเอียดที่จะกล่าวต่อไปนี้

4.2.1 การทดลองประสิทธิภาพของการจำแนกด้วยวิธีการจำแนก k-Nearest Neighbors

การทดลองเปรียบเทียบประสิทธิภาพของวิธีการสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอกับคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF โดยใช้วิธีการจำแนก k-Nearest Neighbors (k-NN) ซึ่งในการทดลองมีการกำหนดค่าพารามิเตอร์ k ที่ต่างกัน 3 ค่า ได้แก่ $k=3$ $k=5$ และ $k=7$ ผลการทดลองสำหรับแต่ละชุดข้อมูลแสดงดังตารางที่ 4.3

ตารางที่ 4.3 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี k-NN

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ k-NN		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS1 - Amazon					
TF	k = 3	0.7200	0.7517	0.7200	0.7103
	k = 5	0.7140	0.7441	0.7140	0.7031
	k = 7	0.7030	0.7401	0.7030	0.6895
TF-IDF	k = 3	0.5500	0.6396	0.5500	0.4677
	k = 5	0.5580	0.7010	0.5580	0.4663
	k = 7	0.5560	0.6741	0.5560	0.4672
V4D	k = 3	0.8610	0.8691	0.8610	0.8602
	k = 5	0.8610	0.8688	0.8610	0.8602
	k = 7	0.8620	0.8699	0.8620	0.8612
V8D	k = 3	**0.8670***	**0.8735***	**0.8670***	0.8664
	k = 5	**0.8670***	0.8725	**0.8670***	**0.8665***
	k = 7	0.8650	0.8707	0.8650	0.8645
ชุดข้อมูล DS2 - IMDb					
TF	k = 3	0.6480	0.6883	0.6480	0.6270
	k = 5	0.6450	0.6800	0.6450	0.6241
	k = 7	0.6320	0.6628	0.6320	0.6113
TF-IDF	k = 3	0.5110	0.6362	0.5110	0.3646
	k = 5	0.5130	0.5532	0.5130	0.3813
	k = 7	0.5110	0.4694	0.5110	0.3762
V4D	k = 3	0.7940	0.7956	0.7940	0.7937
	k = 5	0.7980	0.7995	0.7980	0.7978
	k = 7	**0.7990***	**0.8006***	**0.7990***	**0.7987***
V8D	k = 3	0.7970	0.7980	0.7970	0.7968
	k = 5	0.7970	0.7985	0.7970	0.7967
	k = 7	**0.7990***	**0.8006***	**0.7990***	**0.7987***

ตารางที่ 4.3 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี k-NN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ k-NN		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS3 - Yelp					
TF	k = 3	0.6830	0.7007	0.6830	0.6754
	k = 5	0.6780	0.6958	0.6780	0.6700
	k = 7	0.6830	0.7020	0.6830	0.6747
TF-IDF	k = 3	0.5410	0.6814	0.5410	0.4289
	k = 5	0.5430	0.6542	0.5430	0.4470
	k = 7	0.5370	0.6922	0.5370	0.4217
V4D	k = 3	0.8230	0.8309	0.8230	0.8220
	k = 5	0.8180	0.8256	0.8180	0.8170
	k = 7	0.8190	0.8263	0.8190	0.8180
V8D	k = 3	**0.8260***	**0.8331***	**0.8260***	**0.8252***
	k = 5	0.8250	0.8315	0.8250	0.8242
	k = 7	0.8240	0.8302	0.8240	0.8232
ชุดข้อมูล DS4 - Apparel					
TF	k = 3	0.6025	0.6244	0.6025	0.5844
	k = 5	0.6160	0.6290	0.6160	0.6061
	k = 7	0.6305	0.6374	0.6305	0.6261
TF-IDF	k = 3	0.5195	0.5551	0.5195	0.4194
	k = 5	0.5130	0.5757	0.5130	0.4088
	k = 7	0.5315	0.5364	0.5315	0.4305
V4D	k = 3	0.7675	0.7713	0.7675	0.7667
	k = 5	0.7635	0.7670	0.7635	0.7627
	k = 7	0.7655	0.7690	0.7655	0.7647
V8D	k = 3	0.7735	0.7755	0.7735	0.7731
	k = 5	**0.7775***	**0.7797***	**0.7775***	**0.7771***
	k = 7	0.7770	0.7792	0.7770	0.7765

ตารางที่ 4.3 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี k-NN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ k-NN		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS5 - Health					
TF	k = 3	0.5880	0.6350	0.5880	0.5444
	k = 5	0.5785	0.6355	0.5785	0.5257
	k = 7	0.5885	0.6483	0.5885	0.5366
TF-IDF	k = 3	0.5045	0.5798	0.5045	0.3887
	k = 5	0.5150	0.5968	0.5150	0.3971
	k = 7	0.5075	0.6410	0.5075	0.3889
V4D	k = 3	0.6860	0.6872	0.6860	0.6854
	k = 5	0.6855	0.6866	0.6855	0.6850
	k = 7	0.6890	0.6901	0.6890	0.6885
V8D	k = 3	0.7080	0.7091	0.7080	0.7076
	k = 5	**0.7085***	**0.7093***	**0.7085***	**0.7082***
	k = 7	0.7070	0.7081	0.7070	0.7066
ชุดข้อมูล DS6 - Music					
TF	k = 3	0.5405	0.5599	0.5405	0.5006
	k = 5	0.5405	0.5676	0.5405	0.4900
	k = 7	0.5370	0.5702	0.5370	0.4767
TF-IDF	k = 3	0.5150	0.5719	0.5150	0.3933
	k = 5	0.5155	0.6275	0.5155	0.3771
	k = 7	0.5040	0.5238	0.5040	0.3537
V4D	k = 3	0.7080	0.7086	0.7080	0.7078
	k = 5	0.7165	0.7168	0.7165	0.7164
	k = 7	**0.7190***	**0.7191***	**0.7190***	**0.7190***
V8D	k = 3	0.7030	0.7035	0.7030	0.7028
	k = 5	0.7130	0.7136	0.7130	0.7128
	k = 7	0.7125	0.7129	0.7125	0.7123

ตารางที่ 4.3 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี k-NN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ k-NN		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS7 - Sports					
TF	k = 3	0.5920	0.6214	0.5920	0.5611
	k = 5	0.5895	0.6233	0.5895	0.5588
	k = 7	0.5850	0.6057	0.5850	0.5627
TF-IDF	k = 3	0.5185	0.5719	0.5185	0.4121
	k = 5	0.5150	0.5476	0.5150	0.4079
	k = 7	0.5155	0.5000	0.5155	0.4113
V4D	k = 3	0.7225	0.7250	0.7225	0.7217
	k = 5	0.7245	0.7265	0.7245	0.7239
	k = 7	**0.7280***	**0.7304***	**0.7280***	**0.7273***
V8D	k = 3	0.7180	0.7189	0.7180	0.7177
	k = 5	0.7245	0.7252	0.7245	0.7243
	k = 7	0.7215	0.7224	0.7215	0.7212
ชุดข้อมูล DS8 - US Airline					
TF	k = 3	0.5841	0.6484	0.6744	0.5573
	k = 5	0.5270	0.6320	0.6426	0.5075
	k = 7	0.4927	0.6238	0.6243	0.4755
TF-IDF	k = 3	0.7725	0.6413	0.5620	0.5676
	k = 5	0.7836	0.6624	0.5578	0.5614
	k = 7	0.7905	0.6809	0.5551	0.5568
V4D	k = 3	0.8542	0.7814	0.7492	0.7629
	k = 5	0.8542	0.7814	0.7492	0.7629
	k = 7	0.8542	0.7814	0.7492	0.7629
V8D	k = 3	**0.8560**	**0.7831**	**0.7569**	**0.7684**
	k = 5	**0.8560**	**0.7831**	**0.7569**	**0.7684**
	k = 7	**0.8560**	**0.7831**	**0.7569**	**0.7684**

จากผลการทดลองในตารางที่ 4.3 พบว่าการสกัดคุณลักษณะแทนข้อความที่นำเสนอ ได้แก่ เวกเตอร์ V4D และเวกเตอร์ V8D มีประสิทธิภาพมากกว่าการสกัดคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ TF และ TF-IDF เมื่อจำแนกด้วยวิธี k-NN สรุปผลที่ดีที่สุดได้ดังตาราง 4.4

ตารางที่ 4.4 สรุปคุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดเมื่อจำแนกด้วย k-NN

ชุดข้อมูล	คุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดในแต่ละตัวชี้วัด			
	Accuracy	Precision	Recall	F1
DS1	V8D, k = 3 V8D, k = 5	V8D, k = 3	V8D, k = 3 V8D, k = 5	V8D, k = 3
DS2	V4D, k = 7 V8D, k = 7	V4D, k = 7 V8D, k = 7	V4D, k = 7 V8D, k = 7	V4D, k = 7 V8D, k = 7
DS3	V8D, k = 3	V8D, k = 3	V8D, k = 3	V8D, k = 3
DS4	V8D, k = 5	V8D, k = 5	V8D, k = 5	V8D, k = 5
DS5	V8D, k = 5	V8D, k = 5	V8D, k = 5	V8D, k = 5
DS6	V4D, k = 7	V4D, k = 7	V4D, k = 7	V4D, k = 7
DS7	V4D, k = 7	V4D, k = 7	V4D, k = 7	V4D, k = 7
DS8	V8D, k = 3 V8D, k = 5 V8D, k = 7	V8D, k = 3 V8D, k = 5 V8D, k = 7	V8D, k = 3 V8D, k = 5 V8D, k = 7	V8D, k = 3 V8D, k = 5 V8D, k = 7

จากตาราง 4.4 จะได้ว่าผลการทดลองเมื่อใช้เวกเตอร์ V8D ให้ผลดีที่สุด ในแง่ตัวชี้วัด Accuracy Precision Recall และ F1 ทั้งนี้การทดลองบนชุดข้อมูล DS2 พบว่าเวกเตอร์ V4D ให้ผลที่ดีที่สุดเช่นเดียวกับเวกเตอร์ V8D ดังนั้นจะเห็นว่าคุณลักษณะแทนข้อความทั้งแบบ V4D และ V8D ที่นำเสนอ จะให้ประสิทธิภาพดีที่สุดสำหรับทุกชุดข้อมูล

4.2.2 การทดลองประสิทธิภาพของการจำแนกด้วยวิธีการจำแนก Naive Bayes

การทดลองเปรียบเทียบประสิทธิภาพของวิธีการสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอกับคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF โดยใช้วิธีการจำแนก Naive Bayes (NB) ผลการทดลองสำหรับแต่ละชุดข้อมูลแสดงดังตารางที่ 4.5

ตารางที่ 4.5 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี NB

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ	Accuracy	Precision	Recall	F1
ชุดข้อมูล DS1 - Amazon				
TF	0.5000	0.4000	0.5000	0.3402
TF-IDF	0.5000	0.2500	0.5000	0.3333
V4D	**0.8670***	**0.8720***	**0.8670***	**0.8665***
V8D	0.6520	0.7560	0.6520	0.6126
ชุดข้อมูล DS2 - IMDb				
TF	0.5000	0.2500	0.5000	0.3333
TF-IDF	0.5000	0.2500	0.5000	0.3333
V4D	0.5460	0.6745	0.5460	0.4444
V8D	**0.7850***	**0.7940***	**0.7850***	**0.7832***
ชุดข้อมูล DS3 - Yelp				
TF	0.4950	0.4184	0.4950	0.3408
TF-IDF	0.4990	0.2497	0.4990	0.3329
V4D	0.7780	0.8174	0.7780	0.7651
V8D	**0.8150***	**0.8221***	**0.8150***	**0.8140***
ชุดข้อมูล DS4 - Apparel				
TF	0.5545	0.6342	0.5545	0.4711
TF-IDF	0.5080	0.5865	0.5080	0.3584
V4D	0.7570	0.7661	0.7570	0.7545
V8D	**0.7810***	**0.7821***	**0.7810***	**0.7808***

ตารางที่ 4.5 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี NB (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ	Accuracy	Precision	Recall	F1
ชุดข้อมูล DS5 - Health				
TF	0.5190	0.5255	0.5190	0.4621
TF-IDF	0.5035	0.4367	0.5035	0.3689
V4D	0.6815	**0.6965***	0.6815	0.6744
V8D	**0.6895***	0.6937	**0.6895***	**0.6877***
ชุดข้อมูล DS6 - Music				
TF	0.5025	0.4629	0.5025	0.3558
TF-IDF	0.5005	0.3501	0.5005	0.3344
V4D	0.5865	0.6273	0.5865	0.5487
V8D	**0.7020***	**0.7035***	**0.7020***	**0.7014***
ชุดข้อมูล DS7 - Sports				
TF	0.5090	0.5461	0.5090	0.3702
TF-IDF	0.5020	0.6172	0.5020	0.3386
V4D	0.7130	0.7239	0.7130	0.7089
V8D	**0.7315***	**0.7331***	**0.7315***	**0.7310***
ชุดข้อมูล DS8 - US Airline				
TF	0.2047	0.1024	0.5000	0.1700
TF-IDF	0.2047	0.1024	0.5000	0.1700
V4D	0.4715	0.6230	0.6531	0.4682
V8D	**0.8448***	**0.7620***	**0.7624***	**0.7620***

จากผลการทดลองในตารางที่ 4.5 พบว่าการสกัดคุณลักษณะแทนข้อความที่นำเสนอ ได้แก่ เวกเตอร์ V4D และเวกเตอร์ V8D มีประสิทธิภาพมากกว่าการสกัดคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ TF และ TF-IDF เมื่อจำแนกด้วยวิธี NB สรุปผลที่ดีที่สุดได้ดังตาราง 4.6

ตารางที่ 4.6 สรุปคุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดเมื่อจำแนกด้วย NB

ชุดข้อมูล	คุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดในแต่ละตัวชี้วัด			
	Accuracy	Precision	Recall	F1
DS1	V4D	V4D	V4D	V4D
DS2	V8D	V8D	V8D	V8D
DS3	V8D	V8D	V8D	V8D
DS4	V8D	V8D	V8D	V8D
DS5	V8D	V4D	V8D	V8D
DS6	V8D	V8D	V8D	V8D
DS7	V8D	V8D	V8D	V8D
DS8	V8D	V8D	V8D	V8D

จากตาราง 4.6 จะได้ว่าผลการทดลองดังนี้

- สำหรับชุดข้อมูล DS2 DS3 DS4 DS6 DS7 และ DS8 เมื่อใช้เวกเตอร์ V8D ให้ผลดีที่สุด ในแง่ตัวชี้วัด Accuracy Precision Recall และ F1
- สำหรับชุดข้อมูล DS1 พบว่าเวกเตอร์ V4D ให้ผลดีที่สุด ในแง่ตัวชี้วัด Accuracy Precision Recall และ F1
- สำหรับชุดข้อมูล DS5 พบว่าเวกเตอร์ V8D ให้ผลดีที่สุด ในแง่ตัวชี้วัด Accuracy Recall และ F1 และเวกเตอร์ V4D ในแง่ตัวชี้วัด Precision

จะเห็นได้ว่าคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอ จะให้ประสิทธิภาพดีที่สุดเมื่อใช้กับการจำแนกด้วยวิธี Naive Bayes สำหรับทุกชุดข้อมูลที่ทำการทดลอง

4.2.3 การทดลองประสิทธิภาพของการจำแนกด้วยวิธีการจำแนก Artificial Neural Networks

การทดลองเปรียบเทียบประสิทธิภาพของวิธีการสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอกับคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF โดยใช้วิธีการจำแนก Artificial Neural Networks (ANN) ซึ่งในการทำการทดลองมีการกำหนดจำนวนชั้นซ่อนและจำนวนโหนดในแต่ละชั้น 4 รูปแบบ ดังนี้

แบบที่ 1 มีจำนวนชั้นซ่อน 1 ชั้น โดยที่มีจำนวนโหนดเท่ากับจำนวนมิติ ($\boldsymbol{m}$)

แบบที่ 2 มีจำนวนชั้นซ่อน 1 ชั้น โดยที่มีจำนวนโหนดเท่ากับครึ่งหนึ่งของจำนวนมิติ ($\frac{m}{2}$)

แบบที่ 3 มีจำนวนชั้นซ่อน 2 ชั้น โดยที่แต่ละชั้นมีจำนวนโหนดเท่ากับจำนวนมิติ (m, m)

แบบที่ 4 มีจำนวนชั้นซ่อน 2 ชั้น โดยที่ชั้นที่ 1 มีจำนวนโหนดเท่ากับจำนวนมิติและชั้นที่ 2 มีจำนวนโหนดเท่ากับครึ่งหนึ่งของจำนวนมิติ ($m, \frac{m}{2}$)

โดยที่ m แทนจำนวนมิติของเวกเตอร์ข้อมูลนำเข้า ซึ่งในกรณีเวกเตอร์ V4D จะได้ m มีค่าเท่ากับ 4 และเวกเตอร์ V8D จะได้ว่า m มีค่าเท่ากับ 8 แต่สำหรับกรณีเวกเตอร์ TF และเวกเตอร์ TF-IDF ซึ่งมีจำนวนมิติของเวกเตอร์ข้อมูลนำเข้ามาก จากที่ได้ทำการทดลองหากกำหนดจำนวนโหนดให้มีค่าเท่ากับจำนวนมิติจะทำให้ใช้เวลาในการประมวลผลนานและมีปัญหาในเรื่องการจองหน่วยความจำ นอกจากนี้เมื่อเทียบกับการกำหนดจำนวนโหนดให้เท่ากับจำนวนโหนดที่ใช้กับวิธีที่ทำเสนอ พบว่าประสิทธิภาพใกล้เคียงกันหรือในบางครั้งการกำหนดจำนวนโหนดให้เท่ากับจำนวนมิติจะให้ผลที่แย่กว่า ดังนั้นสำหรับกรณีเวกเตอร์เวกเตอร์ TF และเวกเตอร์ TF-IDF จะกำหนดให้ m m มีค่าเท่ากับ 8 ผลการทดลองสำหรับแต่ละชุดข้อมูลแสดงดังตารางที่ 4.7

ตารางที่ 4.7 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี ANN

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / จำนวนโหนดในชั้นซ่อน		Accuracy	Precision	Recall	F1
		ชุดข้อมูล DS1 - Amazon			
TF	จำนวนโหนด : 8	0.7900	0.7916	0.7900	0.7897
	จำนวนโหนด : 4	0.7950	0.8006	0.7950	0.7941
	จำนวนโหนด : 8,8	0.8030	0.8051	0.8030	0.8026
	จำนวนโหนด : 8,4	0.7920	0.7945	0.7920	0.7916
TF-IDF	จำนวนโหนด : 8	0.5720	0.6859	0.5720	0.4948
	จำนวนโหนด : 4	0.5770	0.6751	0.5770	0.5077
	จำนวนโหนด : 8, 8	0.5830	0.6830	0.5830	0.5182
	จำนวนโหนด : 8, 4	0.5820	0.6811	0.5820	0.5177
V4D	จำนวนโหนด : 4	0.8630	0.8708	0.8630	0.8622
	จำนวนโหนด : 2	0.8640	0.8713	0.8640	0.8633
	จำนวนโหนด : 4,4	0.8620	0.8705	0.8620	0.8612
	จำนวนโหนด : 4,2	0.8650	0.8737	0.8650	0.8642
V8D	จำนวนโหนด : 8	0.8760	0.8805	0.8760	0.8756
	จำนวนโหนด : 4	0.8740	0.8789	0.8740	0.8736
	จำนวนโหนด : 8,8	0.8760	0.8809	0.8760	0.8756
	จำนวนโหนด : 8,4	**0.8770***	**0.8816***	**0.8770***	**0.8766***
		ชุดข้อมูล DS2 - IMDb			
TF	จำนวนโหนด : 8	0.7310	0.7358	0.7310	0.7292
	จำนวนโหนด : 4	0.7310	0.7336	0.7310	0.7303
	จำนวนโหนด : 8,8	0.7420	0.7444	0.7420	0.7415
	จำนวนโหนด : 8,4	0.7310	0.7348	0.7310	0.7299

ตารางที่ 4.7 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี ANN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / จำนวนโหนดในชั้นซ่อน		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS2 - IMDb (ต่อ)					
TF-IDF	จำนวนโหนด : 8	0.5590	0.6251	0.5590	0.4917
	จำนวนโหนด : 4	0.5570	0.6345	0.5570	0.4802
	จำนวนโหนด : 8,8	0.5600	0.6465	0.5600	0.4827
	จำนวนโหนด : 8,4	0.5540	0.6314	0.5540	0.4750
V4D	จำนวนโหนด : 4	0.7900	0.7930	0.7900	0.7895
	จำนวนโหนด : 2	0.7950	0.7972	0.7950	0.7946
	จำนวนโหนด : 4,4	0.7900	0.7916	0.7900	0.7897
	จำนวนโหนด : 4,2	0.7890	0.7907	0.7890	0.7887
V8D	จำนวนโหนด : 8	0.8010	0.8020	0.8010	0.8008
	จำนวนโหนด : 4	**0.8040***	0.8050	**0.8040***	**0.8038***
	จำนวนโหนด : 8,8	**0.8040***	**0.8052***	**0.8040***	**0.8038***
	จำนวนโหนด : 8,4	0.8020	0.8031	0.8020	0.8018
ชุดข้อมูล DS3 - Yelp					
TF	จำนวนโหนด : 8	0.7310	0.7358	0.7310	0.7292
	จำนวนโหนด : 4	0.7300	0.7350	0.7300	0.7283
	จำนวนโหนด : 8,8	0.7460	0.7494	0.7460	0.7447
	จำนวนโหนด : 8,4	0.7390	0.7423	0.7390	0.7380
TF-IDF	จำนวนโหนด : 8	0.5590	0.6251	0.5590	0.4917
	จำนวนโหนด : 4	0.5600	0.6345	0.5600	0.4923
	จำนวนโหนด : 8,8	0.5630	0.6410	0.5630	0.4936
	จำนวนโหนด : 8,4	0.5570	0.6362	0.5570	0.4809
V4D	จำนวนโหนด : 4	0.8130	0.8216	0.8130	0.8118
	จำนวนโหนด : 2	0.8130	0.8233	0.8130	0.8116
	จำนวนโหนด : 4,4	0.8140	0.8251	0.8140	0.8125
	จำนวนโหนด : 4,2	0.8140	0.8256	0.8140	0.8124

ตารางที่ 4.7 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี ANN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / จำนวนโหนดในชั้นซ่อน		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS3 - Yelp (ต่อ)					
V8D	จำนวนโหนด : 8	0.8230	0.8313	0.8230	0.8219
	จำนวนโหนด : 4	0.8240	0.8324	0.8240	0.8229
	จำนวนโหนด : 8,8	0.8240	0.8324	0.8240	0.8229
	จำนวนโหนด : 8,4	**0.8260***	**0.8336***	**0.8260***	**0.8250***
ชุดข้อมูล DS4 - Apparel					
TF	จำนวนโหนด : 8	0.7450	0.7492	0.7450	0.7439
	จำนวนโหนด : 4	0.7300	0.7310	0.7300	0.7297
	จำนวนโหนด : 8,8	0.7425	0.7503	0.7425	0.7403
	จำนวนโหนด : 8,4	0.7410	0.7439	0.7410	0.7402
TF-IDF	จำนวนโหนด : 8	0.5845	0.6612	0.5845	0.5293
	จำนวนโหนด : 4	0.5840	0.6480	0.5840	0.5351
	จำนวนโหนด : 8,8	0.5830	0.6533	0.5830	0.5282
	จำนวนโหนด : 8,4	0.5800	0.6536	0.5800	0.5238
V4D	จำนวนโหนด : 4	0.7705	0.7746	0.7705	0.7697
	จำนวนโหนด : 2	0.7805	0.7822	0.7805	0.7802
	จำนวนโหนด : 4,4	0.7715	0.7766	0.7715	0.7704
	จำนวนโหนด : 4,2	0.7670	0.7723	0.7670	0.7659
V8D	จำนวนโหนด : 8	0.7765	0.7792	0.7765	0.7759
	จำนวนโหนด : 4	**0.7925***	**0.7936***	**0.7925***	**0.7923***
	จำนวนโหนด : 8,8	0.7700	0.7723	0.7700	0.7695
	จำนวนโหนด : 8,4	0.7800	0.7817	0.7800	0.7797

ตารางที่ 4.7 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี ANN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / จำนวนโหนดในชั้นซ่อน		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS5 - Health					
TF	จำนวนโหนด : 8	**0.7495***	**0.7510***	**0.7495***	**0.7491***
	จำนวนโหนด : 4	0.6945	0.6949	0.6945	0.6943
	จำนวนโหนด : 8,8	0.7350	0.7366	0.7350	0.7346
	จำนวนโหนด : 8,4	0.7295	0.7308	0.7295	0.7291
TF-IDF	จำนวนโหนด : 8	0.5800	0.6515	0.5800	0.5247
	จำนวนโหนด : 4	0.5610	0.6297	0.5610	0.4949
	จำนวนโหนด : 8,8	0.5885	0.6633	0.5885	0.5349
	จำนวนโหนด : 8,4	0.5910	0.6658	0.5910	0.5385
V4D	จำนวนโหนด : 4	0.6995	0.7021	0.6995	0.6984
	จำนวนโหนด : 2	0.6950	0.6982	0.6950	0.6937
	จำนวนโหนด : 4,4	0.6980	0.7015	0.6980	0.6966
	จำนวนโหนด : 4,2	0.6975	0.7012	0.6975	0.6960
V8D	จำนวนโหนด : 8	0.7320	0.7331	0.7320	0.7317
	จำนวนโหนด : 4	0.7325	0.7333	0.7325	0.7323
	จำนวนโหนด : 8,8	0.7265	0.7277	0.7265	0.7262
	จำนวนโหนด : 8,4	0.7245	0.7264	0.7245	0.7238
ชุดข้อมูล DS6 - Music					
TF	จำนวนโหนด : 8	**0.7545***	**0.7552***	**0.7545***	**0.7543***
	จำนวนโหนด : 4	0.6965	0.7008	0.6965	0.6947
	จำนวนโหนด : 8,8	0.6910	0.6933	0.6910	0.6900
	จำนวนโหนด : 8,4	0.6885	0.6891	0.6885	0.6882
TF-IDF	จำนวนโหนด : 8	0.5780	0.6499	0.5780	0.5214
	จำนวนโหนด : 4	0.5590	0.6268	0.5590	0.4913
	จำนวนโหนด : 8,8	0.5625	0.6260	0.5625	0.4990
	จำนวนโหนด : 8,4	0.5640	0.6138	0.5640	0.5098

ตารางที่ 4.7 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี ANN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / จำนวนโหนดในชั้นซ่อน		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS6 - Music (ต่อ)					
V4D	จำนวนโหนด : 4	0.7250	0.7256	0.7250	0.7248
	จำนวนโหนด : 2	0.7195	0.7224	0.7195	0.7186
	จำนวนโหนด : 4,4	0.7190	0.7202	0.7190	0.7186
	จำนวนโหนด : 4,2	0.7185	0.7192	0.7185	0.7182
V8D	จำนวนโหนด : 8	0.7320	0.7325	0.7320	0.7319
	จำนวนโหนด : 4	0.7260	0.7263	0.7260	0.7259
	จำนวนโหนด : 8,8	0.7185	0.7196	0.7185	0.7181
	จำนวนโหนด : 8,4	0.7255	0.7258	0.7255	0.7254
ชุดข้อมูล DS7 - Sports					
TF	จำนวนโหนด : 8	**0.7520***	**0.7535***	**0.7520***	**0.7517***
	จำนวนโหนด : 4	0.7205	0.7213	0.7205	0.7202
	จำนวนโหนด : 8,8	0.7170	0.7173	0.7170	0.7169
	จำนวนโหนด : 8,4	0.7190	0.7197	0.7190	0.7188
TF-IDF	จำนวนโหนด : 8	0.5790	0.6413	0.5790	0.5256
	จำนวนโหนด : 4	0.5755	0.6430	0.5755	0.5203
	จำนวนโหนด : 8,8	0.5765	0.6403	0.5765	0.5221
	จำนวนโหนด : 8,4	0.5790	0.6498	0.5790	0.5221
V4D	จำนวนโหนด : 4	0.7255	0.7290	0.7255	0.7245
	จำนวนโหนด : 2	0.7310	0.7336	0.7310	0.7303
	จำนวนโหนด : 4,4	0.7250	0.7309	0.7250	0.7234
	จำนวนโหนด : 4,2	0.7240	0.7286	0.7240	0.7228
V8D	จำนวนโหนด : 8	0.7315	0.7327	0.7315	0.7311
	จำนวนโหนด : 4	0.7500	0.7532	0.7500	0.7492
	จำนวนโหนด : 8,8	0.7235	0.7238	0.7235	0.7234
	จำนวนโหนด : 8,4	0.7335	0.7352	0.7335	0.7331

ตารางที่ 4.7 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี ANN (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / จำนวนโหนดในชั้นซ่อน		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS8 - US Airline					
TF	จำนวนโหนด : 8	0.8684	**0.8223**	0.7543	0.7767
	จำนวนโหนด : 4	**0.8692**	0.8202	0.7512	0.7756
	จำนวนโหนด : 8,8	0.8687	0.8179	**0.7592**	**0.7789**
	จำนวนโหนด : 8,4	0.8688	0.8202	0.7549	0.7771
TF-IDF	จำนวนโหนด : 8	0.8220	0.7883	0.5923	0.6079
	จำนวนโหนด : 4	0.8201	0.7652	0.5966	0.6140
	จำนวนโหนด : 8,8	0.8187	0.7611	0.5944	0.6105
	จำนวนโหนด : 8,4	0.8197	0.7666	0.5964	0.6136
V4D	จำนวนโหนด : 4	0.8549	0.7828	0.7490	0.7634
	จำนวนโหนด : 2	0.8549	0.7828	0.7490	0.7634
	จำนวนโหนด : 4,4	0.8549	0.7828	0.7490	0.7634
	จำนวนโหนด : 4,2	0.8543	0.7816	0.7492	0.7630
V8D	จำนวนโหนด : 8	0.8562	0.7835	0.7568	0.7686
	จำนวนโหนด : 4	0.8564	0.7839	0.7573	0.7690
	จำนวนโหนด : 8,8	0.8560	0.7832	0.7567	0.7684
	จำนวนโหนด : 8,4	0.8559	0.7832	0.7557	0.7678

จากผลการทดลองในตารางที่ 4.7 สามารถสรุปผลที่ให้ประสิทธิภาพดีที่สุดสำหรับแต่ละชุดข้อมูล สรุปได้ดังตาราง 4.8

ตารางที่ 4.8 สรุปคุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดเมื่อจำแนกด้วย ANN

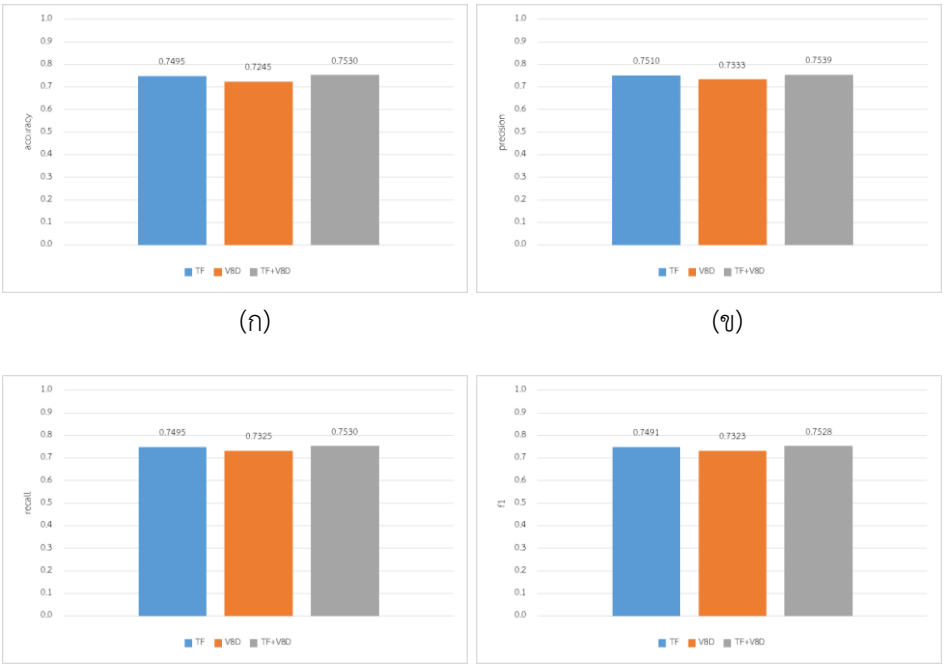
ชุดข้อมูล	คุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดในแต่ละตัวชี้วัด			
	Accuracy	Precision	Recall	F1
DS1	V8D, จำนวนโหนด : 4	V8D, จำนวนโหนด : 4	V8D, จำนวนโหนด : 4	V8D, จำนวนโหนด : 4
DS2	V8D, จำนวนโหนด : 4 V8D, จำนวนโหนด : 8,8	V8D, จำนวนโหนด : 8,8	V8D, จำนวนโหนด : 4 V8D, จำนวนโหนด : 8,8	V8D, จำนวนโหนด : 4 V8D, จำนวนโหนด : 8,8
DS3	V8D, จำนวนโหนด : 8,4	V8D, จำนวนโหนด : 8,4	V8D, จำนวนโหนด : 8,4	V8D, จำนวนโหนด : 8,4
DS4	V8D, จำนวนโหนด : 4	V8D, จำนวนโหนด : 4	V8D, จำนวนโหนด : 4	V8D, จำนวนโหนด : 4
DS5	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8
DS6	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8
DS7	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8
DS8	TF, จำนวนโหนด : 4	TF, จำนวนโหนด : 8	TF, จำนวนโหนด : 8,8	TF, จำนวนโหนด : 8,8

จากตาราง 4.8 จะได้ว่าผลการทดลองเมื่อใช้วิธีจำแนก ANN ประสิทธิภาพในแง่ตัวชี้วัด Accuracy Precision Recall และ F1 จะสรุปได้ดังนี้

- สำหรับชุดข้อมูล DS1 DS2 DS3 และ DS4 เมื่อใช้เวกเตอร์ V8D ให้ผลดีที่สุด
- สำหรับชุดข้อมูล DS5 DS6 DS7 และ DS8 เมื่อใช้เวกเตอร์ TF ให้ผลดีที่สุด

จะเห็นได้ว่าคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ที่นำเสนอ จะให้ประสิทธิภาพดีที่สุดเมื่อใช้กับการจำแนกด้วยวิธี Artificial Neural Networks สำหรับชุดข้อมูล DS1 - Amazon DS2 - IMDb DS3 - Yelp และ DS4 – Apparel และสำหรับชุดข้อมูลที่เหลือนั้น เวกเตอร์ TF ให้ผลดีที่สุด และเวกเตอร์ V8D ให้ผลที่ดีเป็นลำดับรองลงมา จากนั้นผู้วิจัยจึงได้ทำการทดลองเพิ่มเติม โดยนำ

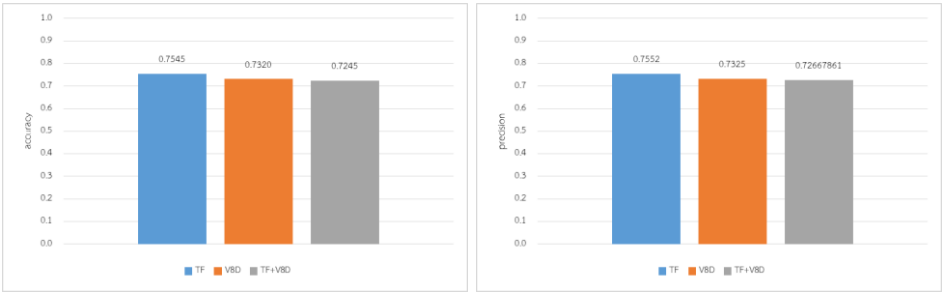
เวกเตอร์ V8D มาใช้ร่วมกันกับเวกเตอร์ TF และทำการทดลองบนชุดข้อมูล DS5 DS6 DS7 และ DS8 พบว่าผลการทดลองไม่ได้ต่างไปจากเดิมมากนัก มีเพียงผลการทดลองบนชุดข้อมูล DS5 และชุดข้อมูล DS8 ที่เวกเตอร์ V8D ช่วยเพิ่มประสิทธิภาพในการจำแนก แสดงดังรูปที่ 4.1 – 4.4 ตามลำดับ อย่างไรก็ตามเวกเตอร์ TF ต้องใช้พื้นที่ในการจัดเก็บข้อมูลที่สูงมากและใช้เวลาในการประมวลผลที่นานเมื่อเทียบกับเวกเตอร์ V8D



(ก) (ข)

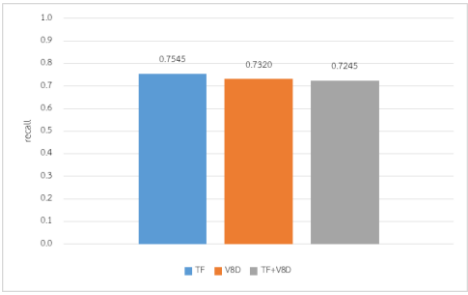
(ค) (ง)

รูปที่ 4.1 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี ANN บนชุดข้อมูล DS5 - Health

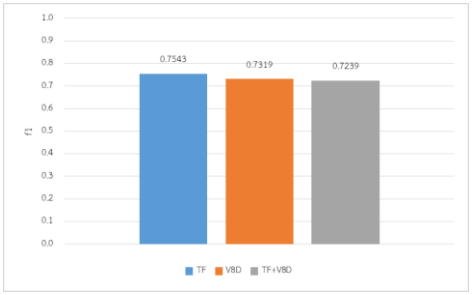


(ก)

(ข)

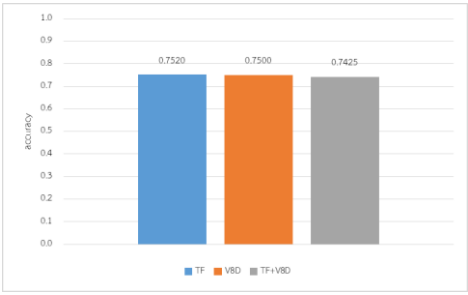


(ค)

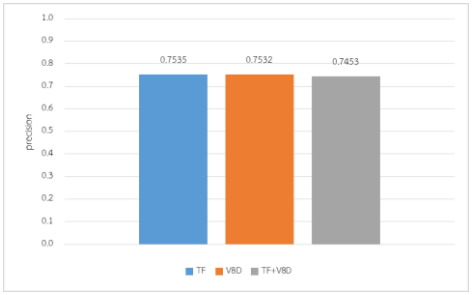


(ง)

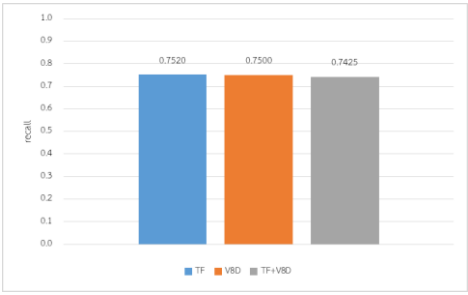
รูปที่ 4.2 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี ANN บนชุดข้อมูล DS6 – Music



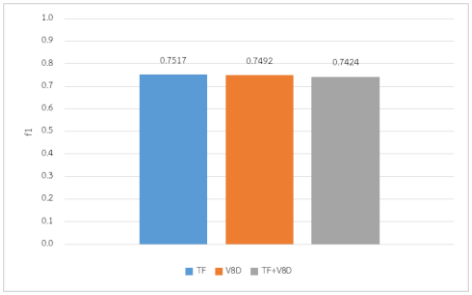
(ก)



(ข)

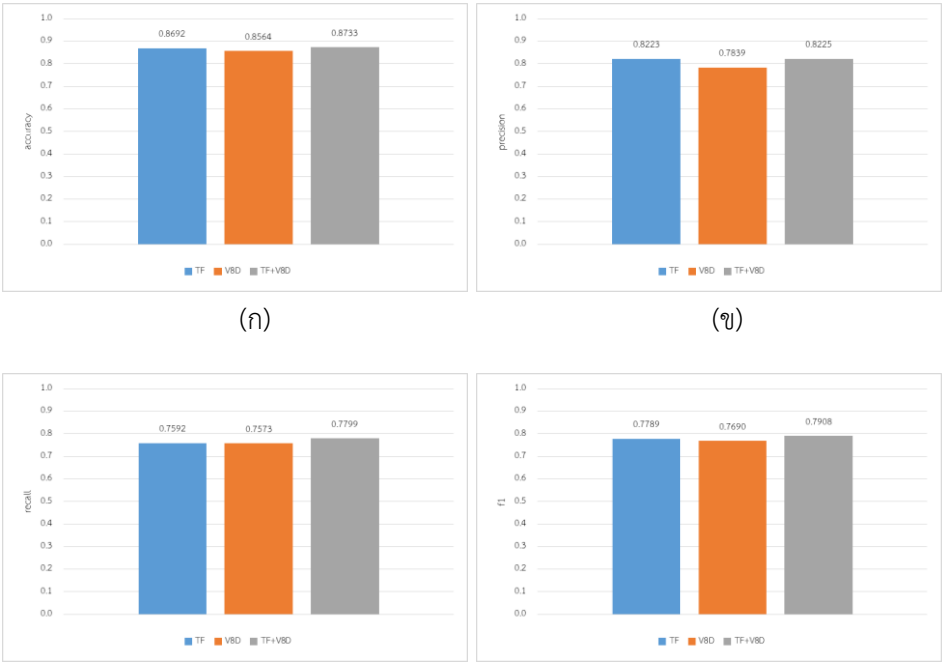


(ค)



(ง)

รูปที่ 4.3 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี ANN บนชุดข้อมูล DS7 – Sports



(ก) (ข)

(ค) (ง)

รูปที่ 4.4 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี ANN บนชุดข้อมูล DS8 – US Airline

4.2.4 การทดลองประสิทธิภาพของการจำแนกด้วยวิธีการจำแนก Support Vector Machine

การทดลองเปรียบเทียบประสิทธิภาพของวิธีการสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอกับคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF โดยใช้วิธีการจำแนก Support Vector Machine (SVM) ซึ่งในการทดลองมีการกำหนดค่าพารามิเตอร์ c ที่ต่างกัน 5 ค่า ได้แก่ c = 0.01 c = 0.1 c = 1 c = 10 และ c = 100 ผลการทดลองสำหรับแต่ละชุดข้อมูลแสดงดังตารางที่ 4.9

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี SVM

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ SVM		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS1 - Amazon					
TF	c = 0.01	0.7550	0.7871	0.7550	0.7465
	c = 0.1	0.8270	0.8295	0.8270	0.8266
	c = 1	0.7980	0.8023	0.7980	0.7971
	c = 10	0.7600	0.7656	0.7600	0.7586
	c = 100	0.7640	0.7742	0.7640	0.7617
TF-IDF	c = 0.01	0.5880	0.7136	0.5880	0.5171
	c = 0.1	0.5760	0.6914	0.5760	0.5007
	c = 1	0.5630	0.6921	0.5630	0.4750
	c = 10	0.5550	0.6720	0.5550	0.4654
	c = 100	0.5550	0.6690	0.5550	0.4665
V4D	c = 0.01	0.8600	0.8659	0.8600	0.8594
	c = 0.1	0.8560	0.8595	0.8560	0.8557
	c = 1	0.8530	0.8549	0.8530	0.8528
	c = 10	0.8530	0.8549	0.8530	0.8528
	c = 100	0.8540	0.8558	0.8540	0.8538
V8D	c = 0.01	**0.8700***	**0.8759***	**0.8700***	**0.8695***
	c = 0.1	0.8640	0.8653	0.8640	0.8639
	c = 1	0.8650	0.8675	0.8650	0.8648
	c = 10	0.8650	0.8675	0.8650	0.8648
	c = 100	0.8650	0.8675	0.8650	0.8648
ชุดข้อมูล DS2 - IMDb					
TF	c = 0.01	0.7540	0.7559	0.7540	0.7536
	c = 0.1	0.7730	0.7747	0.7730	0.7727
	c = 1	0.7250	0.7256	0.7250	0.7248
	c = 10	0.7030	0.7042	0.7030	0.7025
	c = 100	0.6690	0.6734	0.6690	0.6668

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี SVM (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ SVM		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS2 - IMDb (ต่อ)					
TF-IDF	c = 0.01	0.5610	0.6721	0.5610	0.4761
	c = 0.1	0.5540	0.6338	0.5540	0.4740
	c = 1	0.5500	0.6333	0.5500	0.4655
	c = 10	0.5470	0.6332	0.5470	0.4592
	c = 100	0.5480	0.6364	0.5480	0.4599
V4D	c = 0.01	0.7790	0.7907	0.7790	0.7767
	c = 0.1	0.7810	0.7902	0.7810	0.7792
	c = 1	0.7700	0.7793	0.7700	0.7683
	c = 10	0.7640	0.7717	0.7640	0.7623
	c = 100	0.7710	0.7827	0.7710	0.7686
V8D	c = 0.01	**0.7900***	**0.7980***	**0.7900***	**0.7886***
	c = 0.1	0.7860	0.7957	0.7860	0.7842
	c = 1	0.7870	0.7961	0.7870	0.7853
	c = 10	0.7880	0.7963	0.7880	0.7865
	c = 100	0.7880	0.7968	0.7880	0.7864
ชุดข้อมูล DS3 - Yelp					
TF	c = 0.01	0.7660	0.7795	0.7660	0.7631
	c = 0.1	0.8090	0.8103	0.8090	0.8088
	c = 1	0.7810	0.7829	0.7810	0.7806
	c = 10	0.7650	0.7675	0.7650	0.7645
	c = 100	0.7550	0.7609	0.7550	0.7536
TF-IDF	c = 0.01	0.5740	0.6666	0.5740	0.5032
	c = 0.1	0.5760	0.6754	0.5760	0.5066
	c = 1	0.5690	0.6577	0.5690	0.4974
	c = 10	0.5540	0.6314	0.5540	0.4751
	c = 100	0.5550	0.6466	0.5550	0.4712

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี SVM (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ SVM		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS3 - Yelp (ต่อ)					
V4D	c = 0.01	0.8190	0.8252	**0.8190***	0.8181
	c = 0.1	0.8190	0.8252	**0.8190***	0.8181
	c = 1	0.7840	0.8104	0.7840	0.7789
	c = 10	0.8110	0.8227	0.8110	0.8085
	c = 100	0.8130	0.8256	0.8130	0.8109
V8D	c = 0.01	0.8120	0.8271	0.8120	0.8097
	c = 0.1	0.8190	0.8321	**0.8190***	0.8172
	c = 1	0.8160	**0.8339***	0.8160	0.8132
	c = 10	0.8160	0.8298	0.8160	0.8139
	c = 100	**0.8210***	0.8303	0.8210	**0.8197***
ชุดข้อมูล DS4 - Apparel					
TF	c = 0.01	**0.8055***	**0.8078***	**0.8055***	**0.8052***
	c = 0.1	0.7855	0.7873	0.7855	0.7852
	c = 1	0.7855	0.7873	0.7855	0.7852
	c = 10	0.7175	0.7185	0.7175	0.7172
	c = 100	0.7115	0.7129	0.7115	0.7111
TF-IDF	c = 0.01	0.5875	0.6655	0.5875	0.5314
	c = 0.1	0.5830	0.6517	0.5830	0.5283
	c = 1	0.5830	0.6517	0.5830	0.5283
	c = 10	0.5675	0.6385	0.5675	0.5030
	c = 100	0.5655	0.6424	0.5655	0.4980
V4D	c = 0.01	0.7805	0.7816	0.7805	0.7803
	c = 0.1	0.7805	0.7816	0.7805	0.7803
	c = 1	0.7805	0.7816	0.7805	0.7803
	c = 10	0.7805	0.7816	0.7805	0.7803
	c = 100	0.7775	0.7815	0.7775	0.7766

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี SVM (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ SVM		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS4 - Apparel (ต่อ)					
V8D	c = 0.01	0.7915	0.7930	0.7915	0.7913
	c = 0.1	0.7930	0.7945	0.7930	0.7928
	c = 1	0.7930	0.7945	0.7930	0.7928
	c = 10	0.7950	0.7965	0.7950	0.7948
	c = 100	0.7910	0.7928	0.7910	0.7907
ชุดข้อมูล DS5 - Health					
TF	c = 0.01	**0.7845***	**0.7855***	**0.7845***	**0.7843***
	c = 0.1	0.7635	0.7649	0.7635	0.7631
	c = 1	0.7300	0.7309	0.7300	0.7297
	c = 10	0.7010	0.7037	0.7010	0.6998
	c = 100	0.6845	0.6898	0.6845	0.6820
TF-IDF	c = 0.01	0.5780	0.6455	0.5780	0.5229
	c = 0.1	0.5675	0.6276	0.5675	0.5099
	c = 1	0.5795	0.6528	0.5795	0.5211
	c = 10	0.5735	0.6550	0.5735	0.5078
	c = 100	0.5730	0.6542	0.5730	0.5071
V4D	c = 0.01	0.6910	0.6931	0.6910	0.6901
	c = 0.1	0.6910	0.6931	0.6910	0.6901
	c = 1	0.6910	0.6931	0.6910	0.6901
	c = 10	0.6910	0.6931	0.6910	0.6901
	c = 100	0.6995	0.7006	0.6995	0.6991
V8D	c = 0.01	0.7175	0.7215	0.7175	0.7162
	c = 0.1	0.7205	0.7235	0.7205	0.7196
	c = 1	0.7330	0.7348	0.7330	0.7325
	c = 10	0.7315	0.7339	0.7315	0.7308
	c = 100	0.7325	0.7342	0.7325	0.7320

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี SVM (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ SVM		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS6 - Music					
TF	c = 0.01	**0.7495***	**0.7522***	**0.7495***	**0.7489***
	c = 0.1	0.7085	0.7096	0.7085	0.7081
	c = 1	0.6790	0.6806	0.6790	0.6783
	c = 10	0.6375	0.6410	0.6375	0.6350
	c = 100	0.6025	0.6055	0.6025	0.5998
TF-IDF	c = 0.01	0.5635	0.6302	0.5635	0.4988
	c = 0.1	0.5580	0.6218	0.5580	0.4905
	c = 1	0.5405	0.6007	0.5405	0.4585
	c = 10	0.5395	0.5998	0.5395	0.4578
	c = 100	0.5395	0.5998	0.5395	0.4578
V4D	c = 0.01	0.7110	0.7116	0.7110	0.7108
	c = 0.1	0.7110	0.7116	0.7110	0.7108
	c = 1	0.7110	0.7116	0.7110	0.7108
	c = 10	0.7110	0.7116	0.7110	0.7108
	c = 100	0.7075	0.7121	0.7075	0.7058
V8D	c = 0.01	0.7155	0.7157	0.7155	0.7154
	c = 0.1	0.7150	0.7152	0.7150	0.7150
	c = 1	0.7145	0.7146	0.7145	0.7145
	c = 10	0.7145	0.7146	0.7145	0.7145
	c = 100	0.6845	0.6991	0.6845	0.6777
ชุดข้อมูล DS7 - Sports					
TF	c = 0.01	**0.7900***	**0.7907***	**0.7900***	**0.7899***
	c = 0.1	0.7525	0.7531	0.7525	0.7524
	c = 1	0.7235	0.7242	0.7235	0.7233
	c = 10	0.7025	0.7050	0.7025	0.7017
	c = 100	0.6620	0.6666	0.6620	0.6596

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี SVM (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ SVM		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS7 - Sports (ต่อ)					
TF-IDF	c = 0.01	0.5845	0.6630	0.5845	0.5276
	c = 0.1	0.5660	0.6226	0.5660	0.5109
	c = 1	0.5645	0.6259	0.5645	0.5053
	c = 10	0.5605	0.6250	0.5605	0.4980
	c = 100	0.5615	0.6269	0.5615	0.4991
V4D	c = 0.01	0.7155	0.7231	0.7155	0.7132
	c = 0.1	0.7125	0.7219	0.7125	0.7095
	c = 1	0.7155	0.7225	0.7155	0.7133
	c = 10	0.7160	0.7235	0.7160	0.7137
	c = 100	0.6990	0.7099	0.6990	0.6948
V8D	c = 0.01	0.7335	0.7408	0.7335	0.7315
	c = 0.1	0.7415	0.7448	0.7415	0.7406
	c = 1	0.7410	0.7440	0.7410	0.7402
	c = 10	0.7415	0.7447	0.7415	0.7406
	c = 100	0.7285	0.7322	0.7285	0.7274
ชุดข้อมูล DS8 - US Airline					
TF	c = 0.01	0.8193	0.7738	0.7099	0.7206
	c = 0.1	**0.8609***	0.8271	0.7754	0.7858
	c = 1	**0.8609***	0.7996	**0.7881***	**0.7879***
	c = 10	0.8357	0.7534	0.7710	0.7589
	c = 100	0.7796	0.6891	0.7304	0.7003
TF-IDF	c = 0.01	0.8180	0.8245	0.5768	0.5843
	c = 0.1	0.8199	0.7969	0.5841	0.5959
	c = 1	0.8125	0.7400	0.5789	0.5888
	c = 10	0.8004	0.6829	0.5694	0.5758
	c = 100	0.7893	0.6426	0.5641	0.5696

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพเมื่อใช้การจำแนกด้วยวิธี SVM (ต่อ)

โดยที่เครื่องหมาย * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

เวกเตอร์แทนข้อความ / พารามิเตอร์สำหรับ SVM		Accuracy	Precision	Recall	F1
ชุดข้อมูล DS8 - US Airline (ต่อ)					
V4D	c = 0.01	0.7999	**0.8603***	0.5130	0.4698
	c = 0.1	0.8005	0.8528	0.5146	0.4732
	c = 1	0.8000	0.8226	0.5132	0.4702
	c = 10	0.7993	0.8223	0.5115	0.4668
	c = 100	0.8005	0.8528	0.5146	0.4732
V8D	c = 0.01	0.8072	0.8291	0.5336	0.5107
	c = 0.1	0.8072	0.8291	0.5336	0.5107
	c = 1	0.8049	0.8222	0.5277	0.4996
	c = 10	0.8034	0.8277	0.5233	0.4908
	c = 100	0.8072	0.8291	0.5336	0.5107

จากผลการทดลองในตารางที่ 4.9 สามารถสรุปผลที่ให้ประสิทธิภาพดีที่สุดสำหรับแต่ละชุดข้อมูล สรุปได้ดังตาราง 4.10

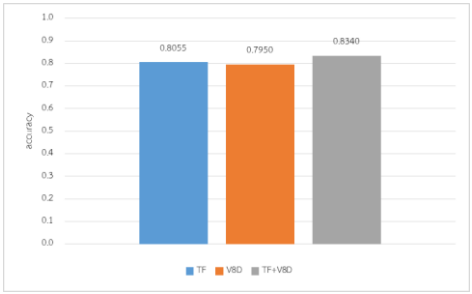
ตารางที่ 4.10 สรุปคุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดเมื่อจำแนกด้วย SVM

ชุดข้อมูล	คุณลักษณะและพารามิเตอร์ที่ให้ผลดีที่สุดในแต่ละตัวชี้วัด			
	Accuracy	Precision	Recall	F1
DS1	V8D, c = 0.01	V8D, c = 0.01	V8D, c = 0.01	V8D, c = 0.01
DS2	V8D, c = 0.01	V8D, c = 0.01	V8D, c = 0.01	V8D, c = 0.01
DS3	V8D, c = 100	V8D, c = 1	V4D, = 0.01 V8D, c = 0.1	V8D, c = 100
DS4	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01
DS5	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01
DS6	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01
DS7	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01	TF, c = 0.01
DS8	TF, c = 0.1	V8D, c = 0.01	TF, c = 1	TF, c = 1

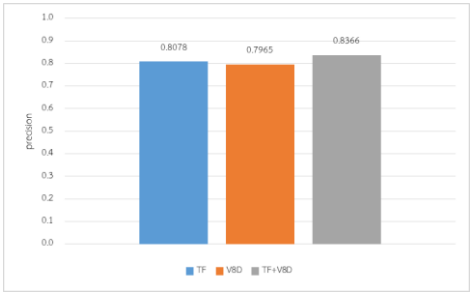
จากตาราง 4.10 จะได้ว่าผลการทดลองเมื่อใช้วิธีจำแนก SVM ประสิทธิภาพในแง่ตัวชี้วัด Accuracy Precision Recall และ F1 จะสรุปได้ดังนี้

- สำหรับชุดข้อมูล DS1 DS2 และ DS3 เมื่อใช้เวกเตอร์ V8D ให้ผลดีที่สุด
- สำหรับชุดข้อมูล DS4 DS5 DS6 และ DS7 เมื่อใช้เวกเตอร์ TF ให้ผลดีที่สุด
- สำหรับชุดข้อมูล DS8 เมื่อใช้เวกเตอร์ V8D ให้ผลที่ดีที่สุด ในแง่ตัวชี้วัด Precision และเวกเตอร์ TF ให้ผลดีที่สุด ในแง่ตัวชี้วัด Accuracy Recall และ F1

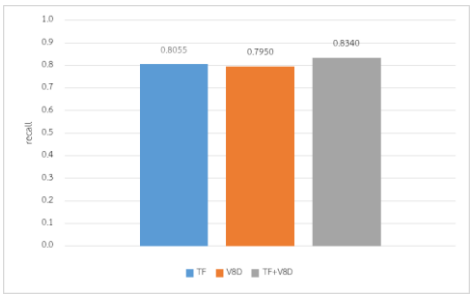
จะเห็นได้ว่าคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ที่นำเสนอ จะให้ประสิทธิภาพดีที่สุดเมื่อใช้กับการจำแนกด้วยวิธี Support Vector Machine สำหรับชุดข้อมูล DS1 - Amazon DS2 - IMDb และ DS3 - Yelp และสำหรับชุดข้อมูลที่เหลือนั้น เวกเตอร์ TF ให้ผลดีที่สุดและเวกเตอร์ V8D ให้ผลที่ดีเป็นลำดับรองลงมา จึงได้ทำการทดลองเพิ่มเติม โดยนำเวกเตอร์ V8D มาใช้ร่วมกันกับเวกเตอร์ TF และทำการทดลองบนชุดข้อมูล DS4 DS5 DS6 DS7 และ DS8 พบว่าเวกเตอร์ V8D ช่วยเพิ่มประสิทธิภาพในการจำแนกได้ แสดงดังรูปที่ 4.5 – 4.9 ตามลำดับ



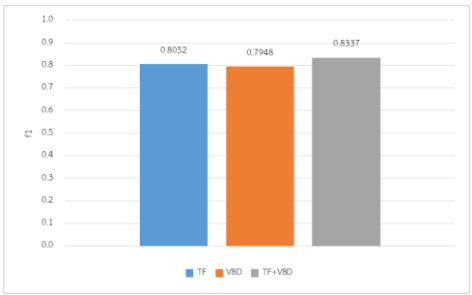
(ก)



(ข)

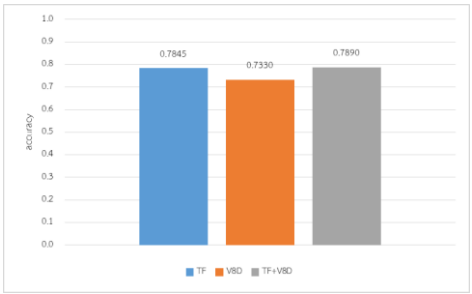


(ค)

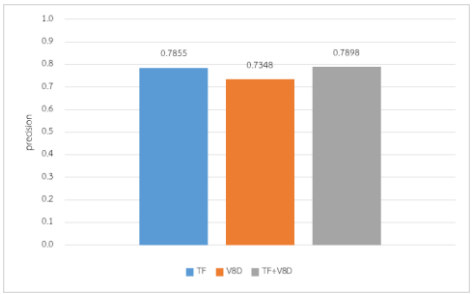


(ง)

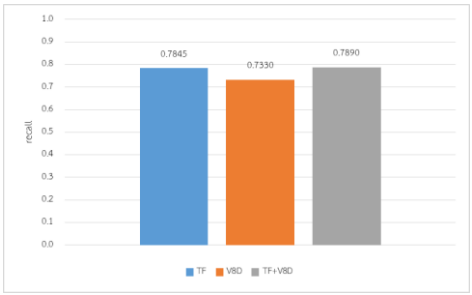
รูปที่ 4.5 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี SVM บนชุดข้อมูล DS4 – Apparel



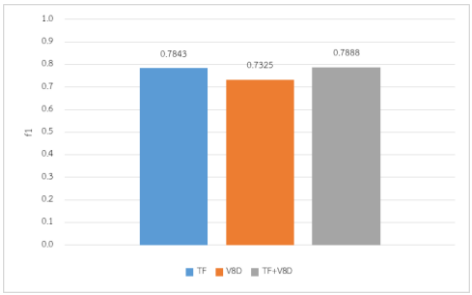
(ก)



(ข)

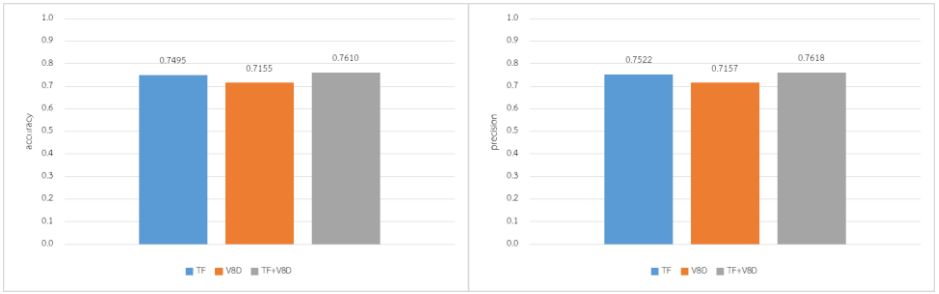


(ค)

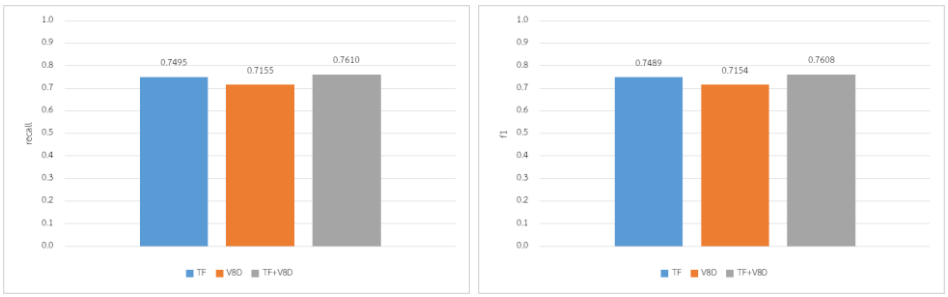


(ง)

รูปที่ 4.6 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี SVM บนชุดข้อมูล DS5 – Health

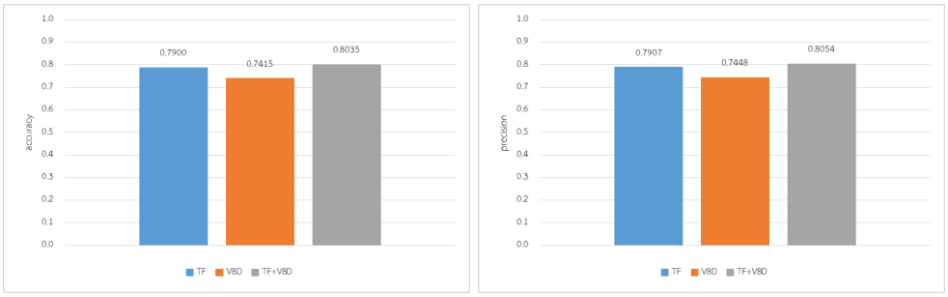


(ก) (ข)

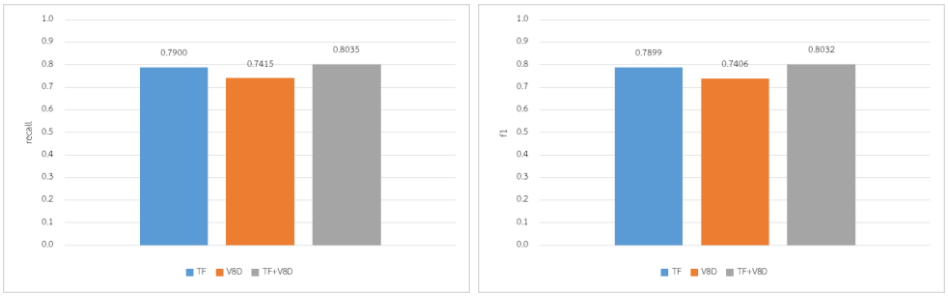


(ค) (ง)

รูปที่ 4.7 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี SVM บนชุดข้อมูล DS6 – Music

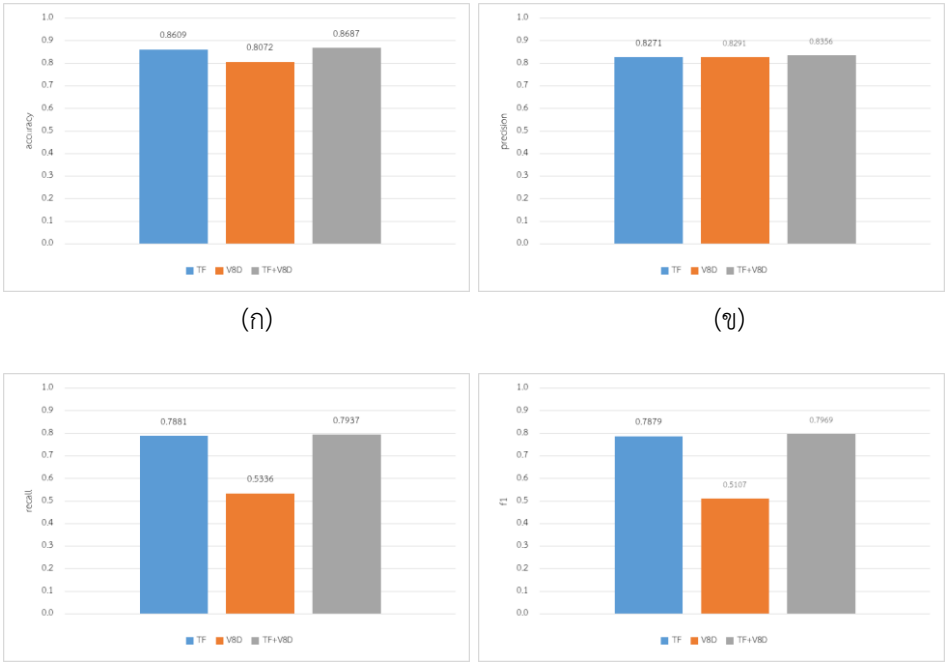


(ก) (ข)



(ค) (ง)

รูปที่ 4.8 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี SVM บนชุดข้อมูล DS7 – Sports



(ก) (ข)

(ค) (ง)

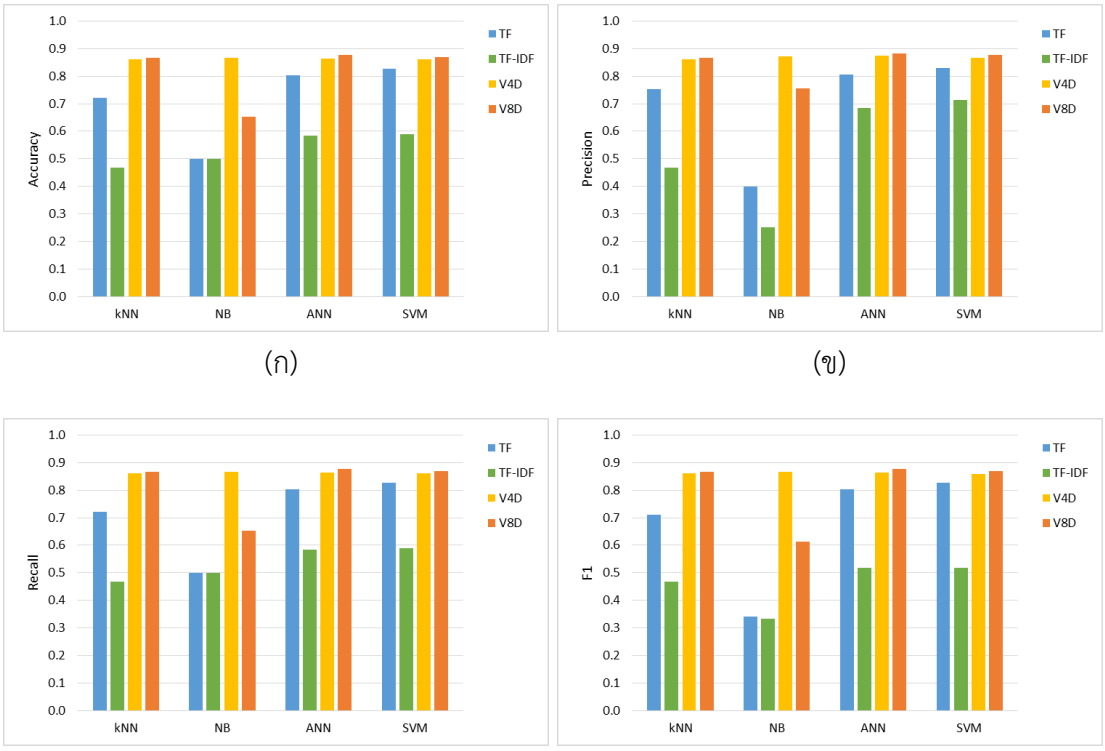
รูปที่ 4.9 ผลการทดลองเปรียบเทียบประสิทธิภาพคุณลักษณะเวกเตอร์ TF เวกเตอร์ V8D และเวกเตอร์ TF+V8D เมื่อจำแนกด้วยวิธี SVM บนชุดข้อมูล DS8 – US Airline

จากผลการทดลองเพื่อวัดประสิทธิภาพของเวกเตอร์แทนข้อความเมื่อใช้วิธีการจำแนกทั้ง 4 แบบ ดังที่แสดงในหัวข้อ 4.2.1 – 4.2.4 มีข้อแนะนำ ดังนี้

- เมื่อใช้วิธีการจำแนก k-Nearest Neighbors และ Naive Bayes ควรใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D
- เมื่อใช้วิธีการจำแนก Artificial Neural Networks และ Support Vector Machine หากต้องการประสิทธิภาพในแง่ของผลการจำแนกควรใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ TF ซึ่งต้องใช้พื้นที่จัดเก็บข้อมูลมากและเวลาในการประมวลผลที่นาน แต่หากต้องการประสิทธิภาพในแง่ของพื้นที่การจัดเก็บข้อมูลและเวลาในการประมวลผลควรใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ซึ่งมีประสิทธิภาพใกล้เคียงกัน

4.2.5 การทดลองเปรียบเทียบผลที่ดีที่สุดจากการจำแนกทั้ง 4 วิธี สำหรับแต่ละชุดข้อมูล

ในหัวข้อนี้จะพิจารณาแต่ละชุดข้อมูลว่าเหมาะสมกับวิธีการจำแนกและคุณลักษณะแทนข้อความแบบใด โดยในแต่ละวิธีการจำแนกจะทำการเลือกผลจากพารามิเตอร์ที่ให้ผลดีที่สุดสำหรับแต่ละวิธีการสกัดคุณลักษณะแทนข้อความ แล้วนำมาเปรียบเทียบประสิทธิภาพในแต่ละชุดข้อมูล ดังรูปที่ 4.10 – 4.17

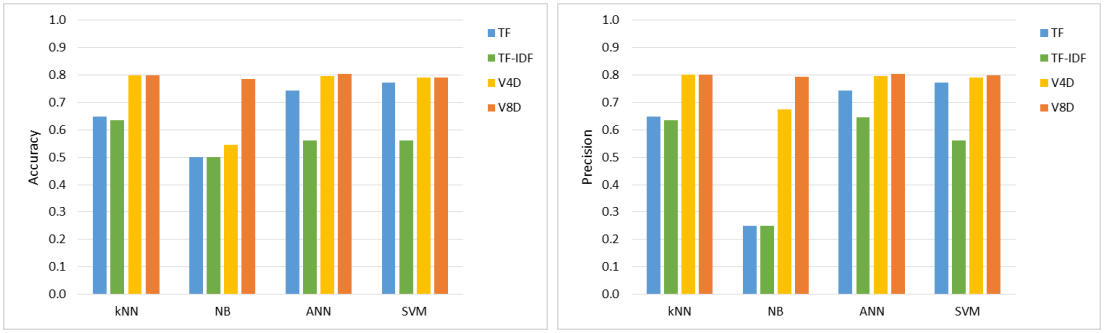


(ก) (ข)

(ค) (ง)

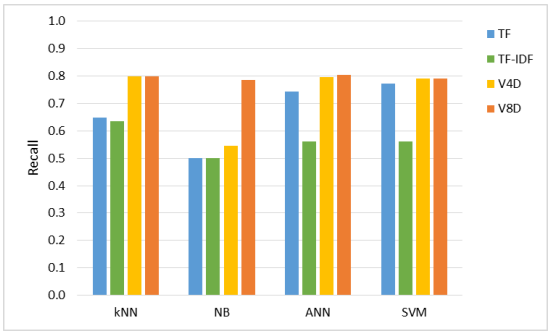
รูปที่ 4.10 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS1 - Amazon สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.10 สามารถสรุปได้ดังนี้ สำหรับชุดข้อมูล DS1 - Amazon เมื่อใช้คุณลักษณะแทนข้อความเวกเตอร์ V8D ที่นำเสนอ ด้วยวิธีการจำแนก Artificial Neural Networks จะให้ประสิทธิภาพดีที่สุด ในแง่ตัวชี้ประสิทธิภาพ Accuracy = 0.8770 Precision = 0.8816 Recall = 0.8770 และ F1 = 0.8766

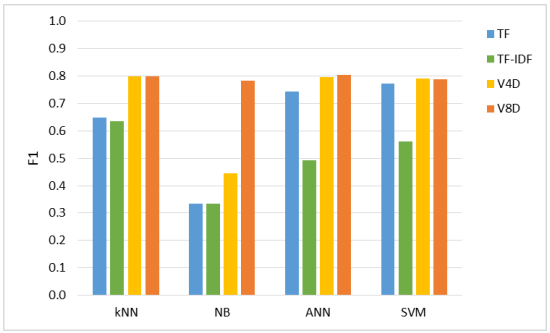


(ก)

(ข)



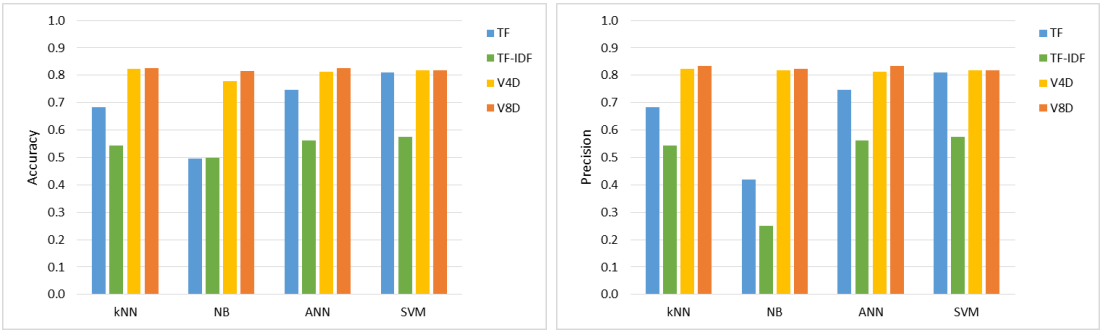
(ค)



(ง)

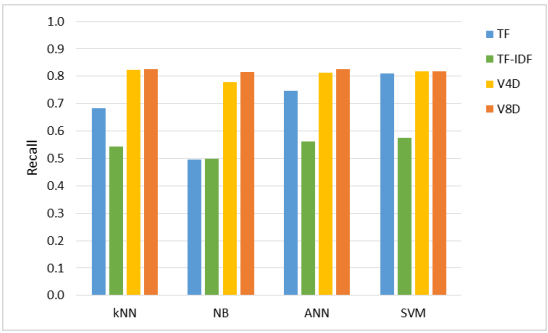
รูปที่ 4.11 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS2 - IMDb สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.11 สามารถสรุปได้ดังนี้ สำหรับชุดข้อมูล DS2 - IMDb เมื่อใช้คุณลักษณะแทนข้อความเวกเตอร์ V8D ที่นำเสนอ ด้วยวิธีการจำแนก Artificial Neural Networks จะให้ประสิทธิภาพดีที่สุด ในแง่ตัวชี้วัดประสิทธิภาพ Accuracy = 0.8040 Precision = 0.8040 Recall = 0.8040 และ F1 = 0.8040

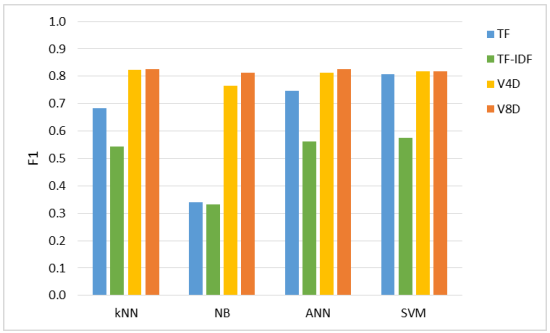


(ก)

(ข)



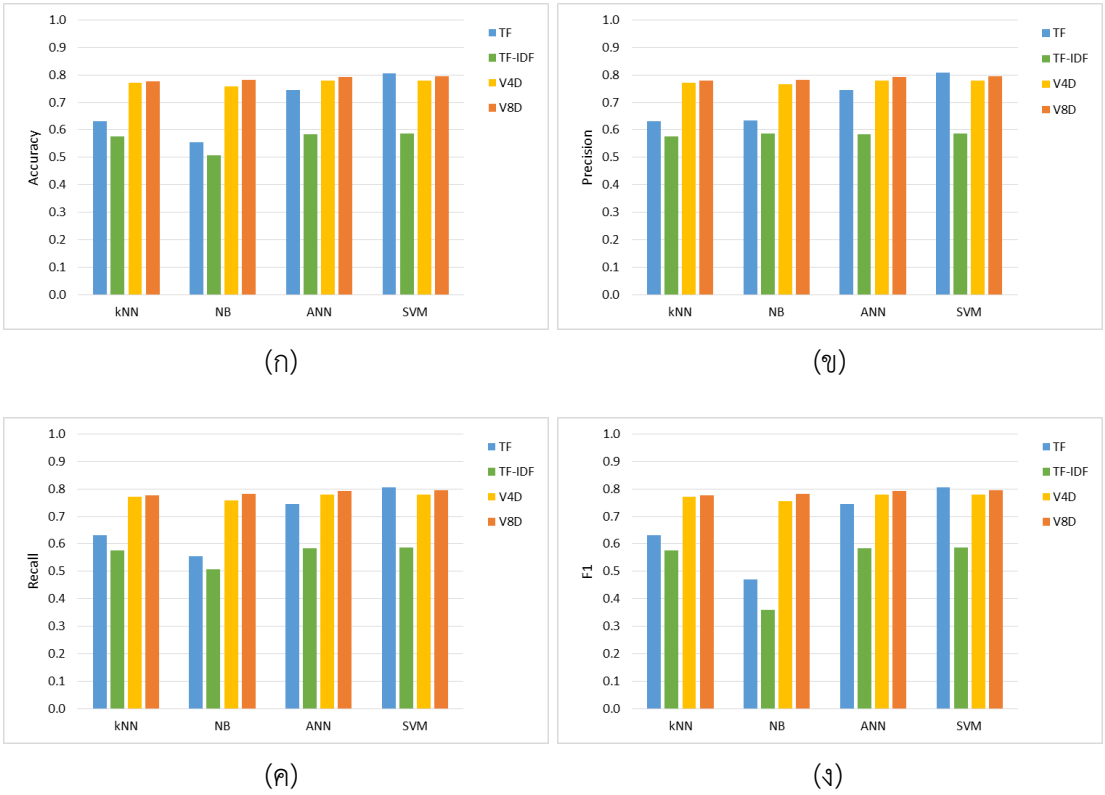
(ค)



(ง)

รูปที่ 4.12 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS3 – Yelp สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.12 จะได้ว่าชุดข้อมูล DS3 - Yelp เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับการจำแนกด้วยวิธี k-Nearest Neighbors ให้ประสิทธิภาพดีที่สุดที่ในแง่ตัวชี้วัด Accuracy = 0.8260 Recall = 0.8260 และ F1 = 0.8252 และเมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับการจำแนกด้วยวิธี Artificial Neural Networks จะให้ประสิทธิภาพดีที่สุดในแง่ตัวชี้วัด Accuracy = 0.8260 Precision = 0.8336 และ Recall = 0.8260

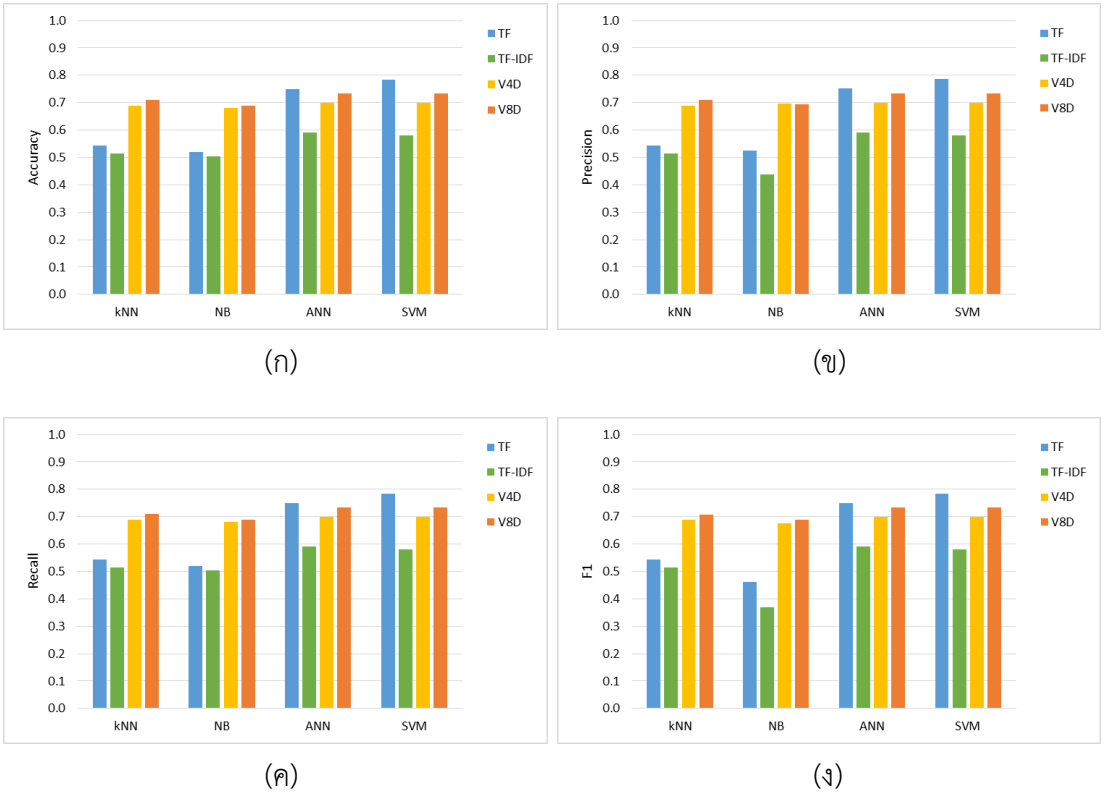


(ก) (ข)

(ค) (ง)

รูปที่ 4.13 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS4 – Apparel สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.13 จะได้ว่า สำหรับชุดข้อมูล DS4 - Apparel เมื่อใช้การจำแนกด้วยวิธี Support Vector Machine กับคุณลักษณะแทนข้อความด้วยเวกเตอร์ TF จะให้ประสิทธิภาพดีที่สุดในทุกตัวชี้วัด Accuracy Precision Recall และ F1 แต่อย่างไรก็ตาม การแทนข้อความด้วยเวกเตอร์ TF จะใช้พื้นที่ในการจัดเก็บข้อมูลมากและใช้เวลาในการประมวลผลนาน ในขณะที่การใช้วิธี Support Vector Machine ร่วมกับเวกเตอร์ V8D ให้ผลที่มีประสิทธิภาพใกล้เคียงกัน ทั้งนี้ได้ทำการทดลองใช้เวกเตอร์ TF ร่วมกับเวกเตอร์ V8D ปรากฏว่าได้ผลที่มีประสิทธิภาพที่สูงขึ้น ดังรูปที่ 4.5 ที่ได้กล่าวไว้ในหัวข้อ 4.2.4

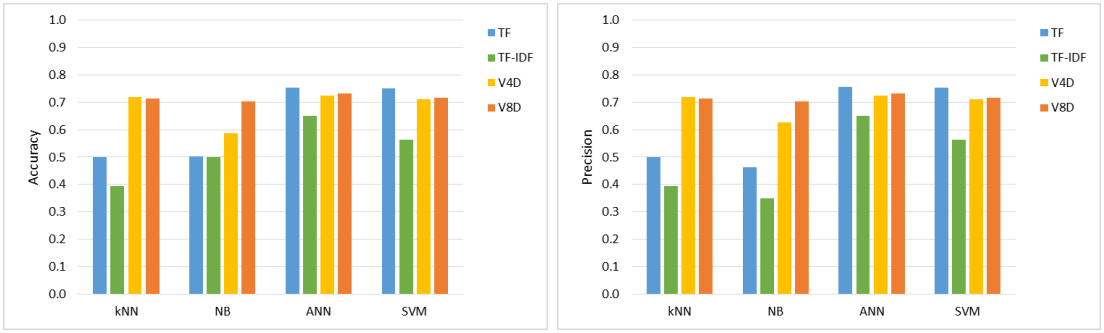


(ก) (ข)

(ค) (ง)

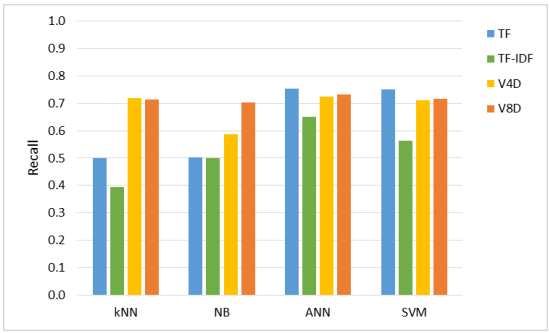
รูปที่ 4.14 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS5 – Health สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.14 จะได้ว่า สำหรับชุดข้อมูล DS5 - Health เมื่อใช้วิธีการจำแนก Support Vector Machine ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ TF จะให้ประสิทธิภาพดีที่สุด ในทุกตัวชี้วัด Accuracy Precision Recall และ F1 แต่การแทนข้อความด้วยเวกเตอร์ TF จะใช้พื้นที่ในการจัดเก็บข้อมูลมากและใช้เวลาในการประมวลผลนาน ในขณะที่การใช้วิธีการจำแนก Support Vector Machine ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ให้ผลที่มีประสิทธิภาพที่ใกล้เคียงกัน ทั้งนี้ได้ทำการทดลองใช้เวกเตอร์ TF ร่วมกับเวกเตอร์ V8D โดยใช้วิธีการจำแนก Artificial Neural Networks และวิธี Support Vector Machine ปรากฏว่าได้ผลที่มีประสิทธิภาพที่สูงขึ้น ดังรูปที่ 4.1 ที่ได้กล่าวไว้ในหัวข้อ 4.2.3 และดังรูปที่ 4.6 ที่ได้กล่าวไว้ในหัวข้อ 4.2.4 ตามลำดับ

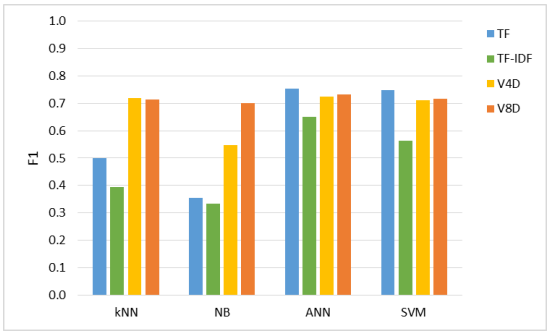


(ก)

(ข)



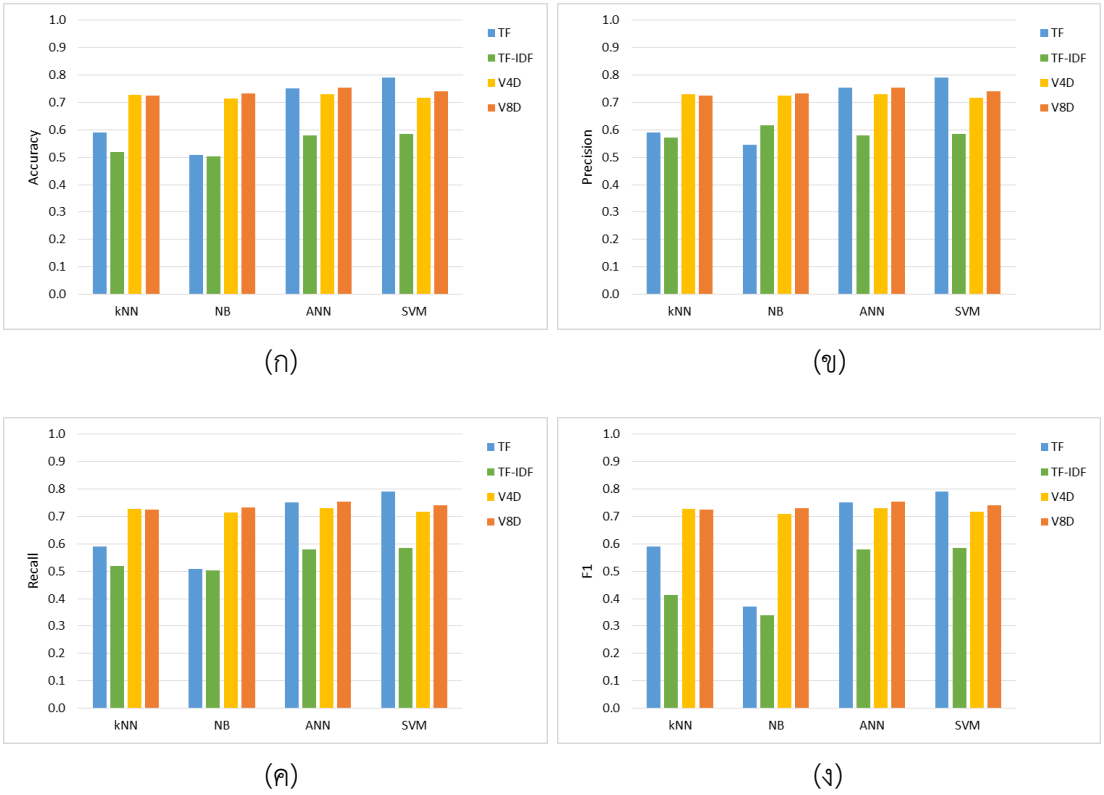
(ค)



(ง)

รูปที่ 4.15 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS6 – Music สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.15 จะได้ว่า สำหรับชุดข้อมูล DS6 - Music เมื่อใช้วิธีการจำแนก Artificial Neural Networks ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ TF จะให้ประสิทธิภาพดีที่สุด ในทุกตัวชี้วัด Accuracy Precision Recall และ F1 แต่การแทนข้อความด้วยเวกเตอร์ TF จะใช้พื้นที่ในการจัดเก็บข้อมูลมากและใช้เวลาในการประมวลผลนาน ในขณะที่การใช้วิธีการจำแนก Artificial Neural Networks ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ให้ผลที่มีประสิทธิภาพที่ใกล้เคียงกัน ทั้งนี้ได้ทำการทดลองใช้เวกเตอร์ TF ร่วมกับเวกเตอร์ V8D โดยใช้วิธีการจำแนก Support Vector Machine ปรากฏว่าได้ผลที่มีประสิทธิภาพที่สูงขึ้น ดังรูปที่ 4.7 ที่ได้กล่าวไว้ในหัวข้อ 4.2.4

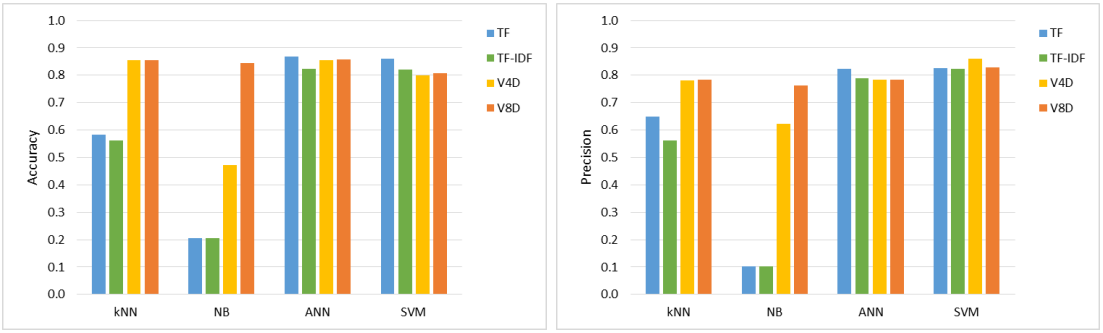


(ก) (ข)

(ค) (ง)

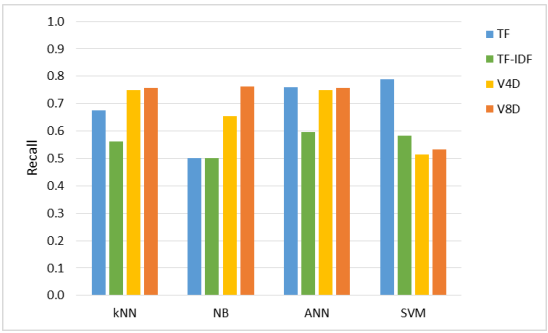
รูปที่ 4.16 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS7 – Sports สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.16 จะได้ว่า สำหรับชุดข้อมูล DS7 - Sports เมื่อใช้วิธีการจำแนก Support Vector Machine ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ TF จะให้ประสิทธิภาพดีที่สุด ในทุกตัวชี้วัด Accuracy Precision Recall และ F1 แต่การแทนข้อความด้วยเวกเตอร์ TF จะใช้พื้นที่ในการจัดเก็บข้อมูลมากและใช้เวลาในการประมวลผลนาน ในขณะที่การใช้วิธีการจำแนก Artificial Neural Networks ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ให้ผลที่มีประสิทธิภาพที่ใกล้เคียงกัน ทั้งนี้ได้ทำการทดลองใช้เวกเตอร์ TF ร่วมกับเวกเตอร์ V8D โดยใช้วิธีการจำแนก Support Vector Machine ปรากฏว่าได้ผลที่มีประสิทธิภาพที่สูงขึ้น ดังรูปที่ 4.8 ที่ได้กล่าวไว้ในหัวข้อ 4.2.4

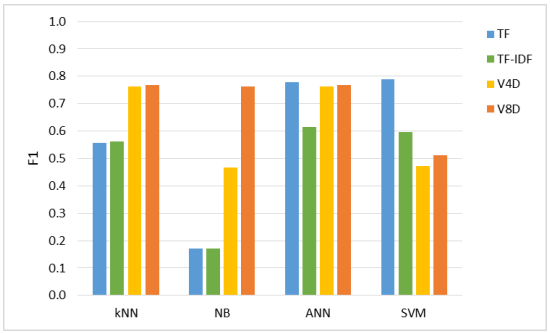


(ก)

(ข)



(ค)



(ง)

รูปที่ 4.17 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS8 – US Airline สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

จากรูปที่ 4.17 จะได้ว่า สำหรับชุดข้อมูล DS8 – US Airline เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับการจำแนกด้วยวิธี จำแนก Support Vector Machine จะให้ประสิทธิภาพดีที่สุดที่ในแง่ตัวชี้วัด Precision และเมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ TF ร่วมกับการจำแนกด้วยวิธี Artificial Neural Networks จะให้ประสิทธิภาพดีที่สุดในแง่ตัวชี้วัด Accuracy และเมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ TF ร่วมกับการจำแนกด้วยวิธี Support Vector Machine จะให้ประสิทธิภาพดีที่สุดในแง่ตัวชี้วัด Recall และ F1 ทั้งนี้ได้ทำการทดลองใช้เวกเตอร์ TF ร่วมกับเวกเตอร์ V8D โดยใช้วิธีการจำแนก Artificial Neural Networks และวิธี Support Vector Machine ปรากฏว่าได้ผลที่มีประสิทธิภาพที่สูงขึ้น ดังรูปที่ 4.4 ที่ได้กล่าวไว้ในหัวข้อ 4.2.3 และดังรูปที่ 4.9 ที่ได้กล่าวไว้ในหัวข้อ 4.2.4 ตามลำดับ

จากผลการทดลองบนทั้ง 8 ชุดข้อมูล ดังรูปที่ 4.10 – 4.17 การวัดประสิทธิภาพของเวกเตอร์แทนข้อความที่ใช้ร่วมกับวิธีการจำแนกทั้ง 4 แบบ แล้วให้ประสิทธิภาพในแง่ของตัวชี้วัดที่ดีที่สุดสำหรับแต่ละชุดข้อมูล พิจารณาได้ดังนี้

- สำหรับชุดข้อมูล DS1 – Amazon และ DS2 – IMDb เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับวิธีการจำแนก Artificial Neural Networks
- สำหรับชุดข้อมูล DS3 – Yelp เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับวิธีการจำแนก k -Nearest Neighbors และ Artificial Neural Networks
- สำหรับชุดข้อมูล DS4 – Apparel DS5 – Health และ DS7 – Sports เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ TF ร่วมกับวิธีการจำแนก Support Vector Machine
- สำหรับชุดข้อมูล DS6 - Music เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ TF ร่วมกับวิธีการจำแนก Artificial Neural Networks
- สำหรับชุดข้อมูล DS8 - US Airline เมื่อใช้วิธีการจำแนก Support Vector Machine ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D จะให้ประสิทธิภาพดีในแง่ Precision และเมื่อใช้วิธีการจำแนก Artificial Neural Networks ร่วมกับเวกเตอร์ TF จะให้ประสิทธิภาพดีในแง่ Accuracy และเมื่อใช้วิธีการจำแนก Support Vector Machine ร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ TF จะให้ประสิทธิภาพดีในแง่ Recall และ F1

จะเห็นได้ว่า สำหรับชุดข้อมูล DS1 DS2 และ DS3 เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ให้ประสิทธิภาพดีที่สุด แต่สำหรับ DS4 DS5 DS6 DS7 และ DS8 เมื่อใช้คุณลักษณะแทนด้วย TF จะให้ประสิทธิภาพที่ดีที่สุด อย่างไรก็ตาม การแทนข้อความด้วยเวกเตอร์ TF จะใช้พื้นที่ในการจัดเก็บข้อมูลมากและใช้เวลาในการประมวลผลนาน แต่หากต้องการประสิทธิภาพในแง่ของพื้นที่การจัดเก็บข้อมูลและเวลาในการประมวลผล ควรใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ซึ่งให้ผลที่มีประสิทธิภาพที่ใกล้เคียงกัน ทั้งนี้หากต้องการประสิทธิภาพในแง่ของผลการจำแนกที่มีความถูกต้องมากยิ่งขึ้นและไม่มีข้อจำกัดในเรื่องของพื้นที่และเวลาสามารถใช้คุณลักษณะแทนข้อความเวกเตอร์ TF ร่วมกับเวกเตอร์ V8D โดยใช้วิธี Artificial Neural Networks ดังผลที่ได้รายงานไว้ในหัวข้อที่ 4.2.3 สำหรับชุดข้อมูล DS5 และ DS8 ดังรูปที่ 4.1 และ 4.4 ตามลำดับ และวิธีการจำแนก Support Vector Machine ดังผลที่ได้รายงานไว้ในหัวข้อที่ 4.2.4 สำหรับชุดข้อมูล DS4 – DS8 ดังรูปที่ 4.5 – 4.9 ตามลำดับ

4.2.6 การวิเคราะห์ประสิทธิภาพของคุณลักษณะแทนข้อความในแง่ของการใช้พื้นที่และเวลาประมวลผล

จากการทดลองในหัวข้อที่ 4.2.1 – 4.2.5 ได้ทำการทดลองเปรียบเทียบประสิทธิภาพของคุณลักษณะแทนข้อความในแง่ความถูกต้องของการจำแนกขั้วความคิดเห็น และนอกจากนี้ ผู้วิจัยเล็งเห็นว่าในเรื่องของการใช้พื้นที่จัดเก็บข้อมูลของเวกเตอร์แทนข้อความและระยะเวลาที่ใช้ประมวลผลเป็นสิ่งสำคัญที่ควรพิจารณาด้วยเช่นกัน เพื่อจะได้สามารถประมวลผลได้ทันต่อการใช้งานจริง โดยเฉพาะข้อมูลที่มีปริมาณมาก

ในหัวข้อนี้จะกล่าวถึง การประเมินประสิทธิภาพของคุณลักษณะแทนข้อความ โดยการเปรียบเทียบประสิทธิภาพในแง่ของการใช้พื้นที่จัดเก็บข้อมูลและระยะเวลาที่ใช้ในการประมวลผลด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอกับคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF โดยใช้วิธีการจำแนกทั้ง 4 วิธี ซึ่งในแต่ละวิธีการจำแนกทำการทดลองโดยมีการกำหนดค่าพารามิเตอร์ ดังนี้

- วิธีการจำแนก k-Nearest Neighbors (k-NN) กำหนดค่าพารามิเตอร์ k = 1
- วิธีการจำแนก Naive Bayes (NB) กำหนดค่าพารามิเตอร์เป็นค่าเริ่มต้น
- วิธีการจำแนก Artificial Neural Networks (ANN) กำหนดจำนวนชั้นซ่อน 1 ชั้นและจำนวนโหนดเท่ากับจำนวนมิติ (m) โดยที่ m แทนจำนวนมิติของเวกเตอร์ข้อมูลนำเข้า ซึ่งในกรณีเวกเตอร์ V4D จะได้ m มีค่าเท่ากับ 4 เวกเตอร์ V8D จะได้ m มีค่าเท่ากับ 8 และเวกเตอร์เวกเตอร์ TF และเวกเตอร์ TF-IDF จะกำหนดให้ m มีค่าเท่ากับ 8 เช่นกัน
- วิธีการจำแนก Support Vector Machine (SVM) กำหนดค่าพารามิเตอร์ c = 1

และทำการทดลองโดยใช้ชุดข้อมูลทั้ง 8 ชุดมารวมกัน ซึ่งจะได้ข้อความทั้งหมด 22,541 ข้อความ หลังจากนั้นทำการกำหนดจำนวนข้อความให้กับชุดข้อมูลที่ใช้สำหรับการสอนเพื่อสร้างโมเดลจำนวน 22,541 ข้อความ และชุดข้อมูลที่ใช้สำหรับการทดสอบประสิทธิภาพโมเดลจำนวน 10 ข้อความที่ได้จากการสุ่มข้อความจากชุดข้อมูลสอน แต่เนื่องด้วยการแทนข้อความด้วยเวกเตอร์ TF และเวกเตอร์ TF-IDF ต้องใช้พื้นที่ในการจัดเก็บข้อมูลที่สูง จึงมีข้อจำกัดของเรื่องหน่วยความจำ ทำให้ไม่สามารถทำการทดลองตามการที่กำหนดได้ ผู้วิจัยจึงได้กำหนดให้จำนวนข้อความสำหรับในชุดข้อมูลสอนให้เป็น 5,000 ข้อความ ซึ่งมีข้อความเชิงบวกและข้อความเชิงลบอย่างละ 2,500 ข้อความ และสำหรับจำนวนข้อความสำหรับชุดข้อมูลทดสอบที่ได้มาจากการสุ่มจากชุดข้อมูลสอนจำนวน 10 ข้อความเช่นเดิม

จากการทดลองเปรียบเทียบคุณลักษณะแทนข้อความเพื่อประเมินประสิทธิภาพในแง่การใช้พื้นที่จัดเก็บข้อมูลของเวกเตอร์แทนข้อความ ดังตารางที่ 4.11 จะเห็นได้ว่า การแทนคุณลักษณะด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอ จะทำการสกัดคุณลักษณะที่ได้จากการพิจารณาคำศัพท์เชิงบวกและคำศัพท์เชิงลบจากชุดคลังคำศัพท์มาตรฐานที่มีคำศัพท์ทั้งหมด 4,783 คำ ซึ่งวิธีการสกัดคุณลักษณะเวกเตอร์ได้กล่าวไปในหัวข้อที่ 3.1 โดยเวกเตอร์ที่นำเสนอ ได้แก่ เวกเตอร์ V4D และเวกเตอร์ V8D จะมีขนาดข้อมูลของเวกเตอร์เป็น 4 มิติและ 8 มิติ ตามลำดับ ซึ่งมีขนาดของข้อมูลเวกเตอร์แทนข้อความที่เล็กมาก เมื่อเทียบกับเวกเตอร์ TF และเวกเตอร์ TF-IDF ที่ทำการสกัดคุณลักษณะจากการพิจารณาคำศัพท์ทั้งหมดในข้อความที่นำมาวิเคราะห์ ซึ่งส่วนใหญ่จะมีจำนวนของคำศัพท์ที่เยอะและจะทำให้ขนาดข้อมูลของเวกเตอร์มีขนาดใหญ่ตามไปด้วย โดยจำนวนของคำศัพท์และขนาดข้อมูลของเวกเตอร์จะแตกต่างกันออกไปในแต่ละครั้งการทดลองซึ่งจะขึ้นอยู่กับข้อความในแต่ละครั้งที่นำมาวิเคราะห์ และเมื่อทำการทดลองประเมินประสิทธิภาพคุณลักษณะแทนข้อความในแง่การใช้ระยะเวลาในการประมวลผล ซึ่งในงานวิจัยนี้ได้ทำการพิจารณาเวลาในการประมวลผลเป็น 2 ส่วน ประกอบด้วย ระยะเวลาที่ใช้ในการสร้างโมเดล และระยะเวลาที่ใช้ในการจำแนกขั้วความคิดเห็นเชิงบวกและเชิงลบในแต่ละข้อความ จากการทดลองเปรียบเทียบระยะเวลาในการประมวลผลจะเห็นได้ว่าเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอ ใช้ระยะเวลาในการประมวลผลน้อยที่สุด ทั้งในการสร้างโมเดลและการจำแนกขั้วความคิดเห็น ที่เป็นเช่นนี้เพราะ เวกเตอร์ V4D และ V8D มีขนาดของข้อมูลที่เล็ก จึงมีการประมวลผลที่รวดเร็ว ดังตารางที่ 4.12

ตารางที่ 4.11 การวิเคราะห์ขนาดข้อมูลของเวกเตอร์แทนข้อความ

คุณลักษณะ	พื้นที่จัดเก็บเวกเตอร์แทนข้อความ
TF	13762 × 4
TF-IDF	13762 × 4
V4D	**4** × 4
V8D	8 × 4

หมายเหตุ พื้นที่จัดเก็บจำนวนเต็ม มีค่าใช้พื้นที่ 4 ไบต์

ตารางที่ 4.12 การเปรียบเทียบระยะเวลาที่ใช้ในการสร้างโมเดลและการจำแนกข้อความสำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

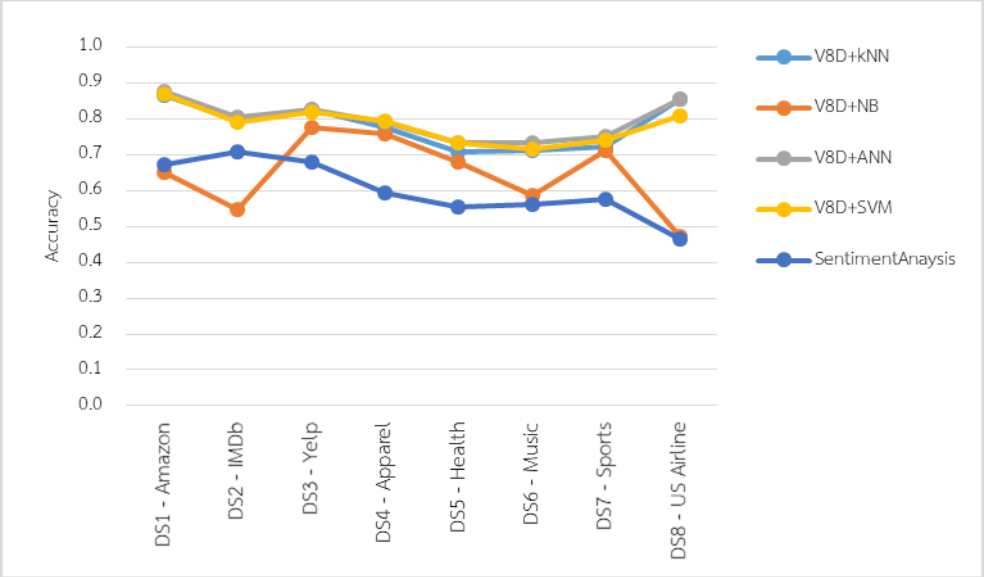
หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละการจำแนก

วิธีการจำแนก / คุณลักษณะ	เวลาที่ใช้ในการสร้างโมเดล (วินาที)	เวลาที่ใช้ในการจำแนกข้อความ (วินาที)
k-NN – TF	23.3535	50.6526
k-NN – TF-IDF	333.6311	44.1566
k-NN – V4D	1.8009	0.5527
k-NN – V8D	**1.7959***	**0.5197***
NB – TF	23.3715	3.5678
NB – TF-IDF	343.9797	3.8686
NB – V4D	**1.8668***	**0.0820***
NB – V8D	2.5954	0.1089
ANN – TF	7420.7556	0.2060
ANN – TF-IDF	8230.6944	0.5237
ANN – V4D	**2.6973***	**0.1119***
ANN – V8D	17.1514	0.1349
SVM – TF	110.7044	2.3156
SVM – TF-IDF	467.7715	3.1401
SVM – V4D	**6.9737***	0.0869
SVM – V8D	8.1380	**0.0839***

4.2.7 การทดลองเปรียบเทียบวิธีที่นำเสนอกับเครื่องมือ SentimentAnalysis

ในหัวข้อนี้จะทำการทดลองเปรียบเทียบวิธีการจำแนกที่นำเสนอโดยใช้การสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D กับวิธีการจำแนกโดยใช้เครื่องมือที่นิยมใช้ในการจำแนกที่เป็นแพ็คเกจสามารถดาวน์โหลดได้ในโปรแกรม R (Feuerriegel & Proellochs, 2019) เรียกว่า SentimentAnalysis ซึ่งใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ TF-IDF แล้วนำมาเปรียบเทียบ

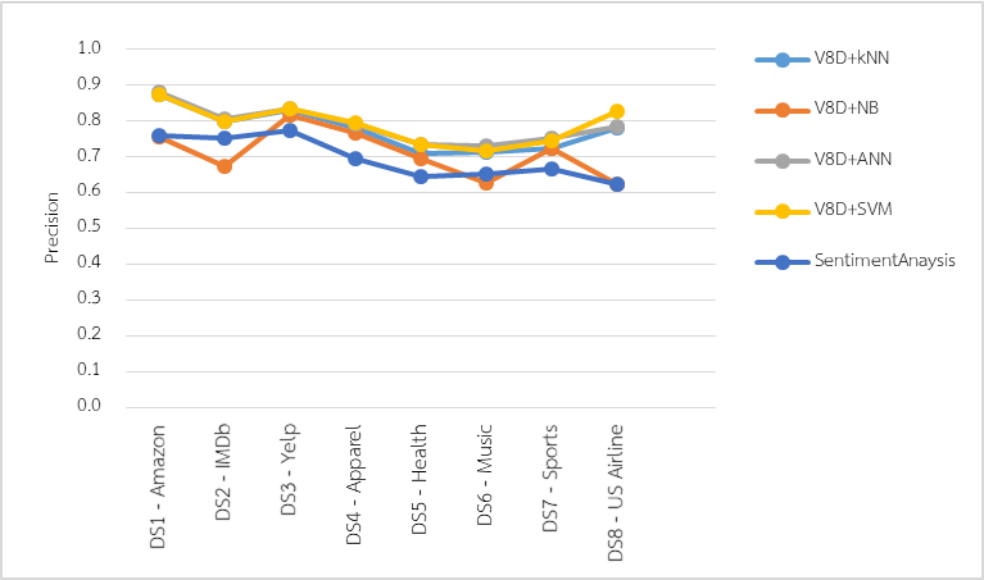
ประสิทธิภาพการจำแนกในแต่ละชุดข้อมูลในแง่ของตัวชี้วัดความถูกต้อง Accuracy Precision Recall และ F1 ดังรูปที่ 4.18 – 4.21



รูปที่ 4.18 ผลการทดลองเปรียบเทียบวิธีที่นำเสนอกับเครื่องมือ SentimentAnalysis ด้วยตัวชี้วัด Accuracy

จากรูปที่ 4.18 จะได้ว่าผลการทดลองเมื่อทำการเปรียบเทียบประสิทธิภาพของวิธีการจำแนกที่นำเสนอโดยใช้การสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D กับวิธีการจำแนกโดยใช้เครื่องมือ SentimentAnalysis ในแง่ตัวชี้วัด Accuracy จะสรุปได้ดังนี้

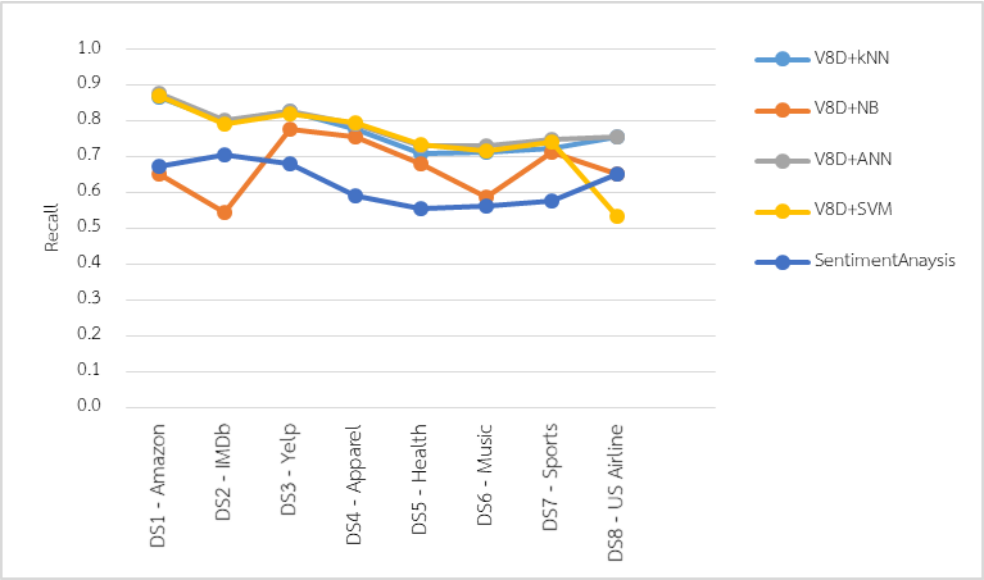
- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี k-Nearest Neighbors วิธี Artificial Neural Networks และวิธี Support Vector Machine จะให้ประสิทธิภาพดีที่สุด สำหรับทุกชุดข้อมูลการทดลอง
- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี Naive Bayes จะให้ประสิทธิภาพดีที่สุด สำหรับชุดข้อมูล DS3 DS4 DS5 DS6 DS7 และ DS8



รูปที่ 4.19 ผลการทดลองเปรียบเทียบวิธีที่นำเสนอกับเครื่องมือ SentimentAnalysis ด้วยตัวชี้วัด Precision

จากรูปที่ 4.19 จะได้ว่าผลการทดลองเมื่อทำการเปรียบเทียบประสิทธิภาพของวิธีการจำแนกที่นำเสนอโดยใช้การสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D กับวิธีการจำแนกโดยใช้เครื่องมือ SentimentAnalysis ในแง่ตัวชี้วัด Precision จะสรุปได้ดังนี้

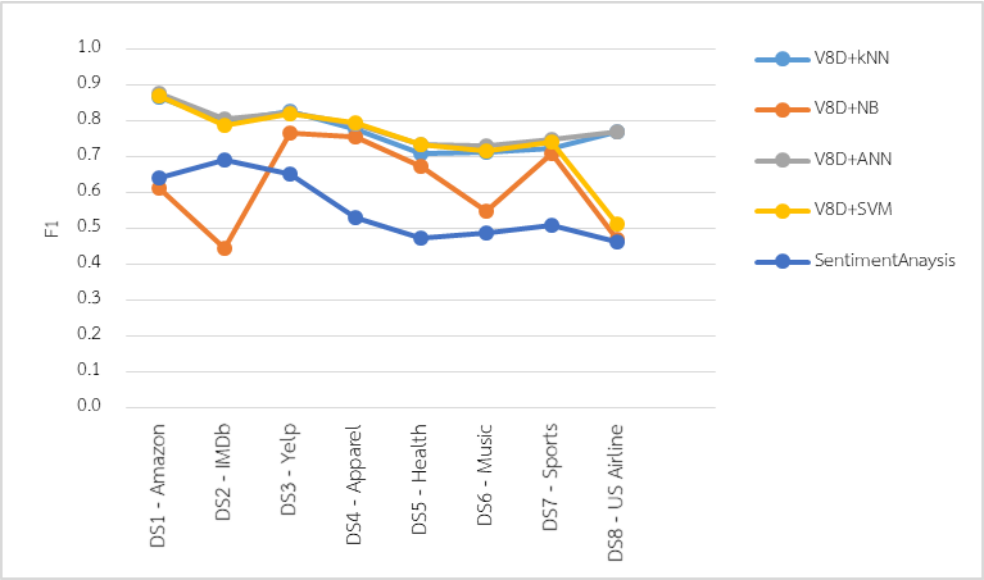
- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี k-Nearest Neighbors วิธี Artificial Neural Networks และวิธี Support Vector Machine จะให้ประสิทธิภาพดีที่สุด สำหรับทุกชุดข้อมูลการทดลอง
- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี Naive Bayes จะให้ประสิทธิภาพดีที่สุด สำหรับชุดข้อมูล DS3 DS4 DS5 และ DS7



รูปที่ 4.20 ผลการทดลองเปรียบเทียบวิธีที่นำเสนอกับเครื่องมือ SentimentAnalysis ด้วยตัวชี้วัด Recall

จากรูปที่ 4.20 จะได้ว่าผลการทดลองเมื่อทำการเปรียบเทียบประสิทธิภาพของวิธีการจำแนกที่นำเสนอโดยใช้การสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D กับวิธีการจำแนกโดยใช้เครื่องมือ SentimentAnalysis ในแง่ตัวชี้วัด Recall จะสรุปได้ดังนี้

- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี k-Nearest Neighbors วิธี Artificial Neural Networks และวิธี Support Vector Machine จะให้ประสิทธิภาพดีที่สุด สำหรับทุกชุดข้อมูลการทดลอง
- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี Naive Bayes จะให้ประสิทธิภาพดีที่สุด สำหรับชุดข้อมูล DS3 DS4 DS5 DS6 DS7 และ DS8
- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี Support Vector Machine จะให้ประสิทธิภาพดีที่สุด สำหรับชุดข้อมูล DS1 DS2 DS3 DS4 DS5 DS6 และ DS7



รูปที่ 4.21 ผลการทดลองเปรียบเทียบวิธีที่นำเสนอกับเครื่องมือ SentimentAnalysis ด้วยตัวชี้วัด F1

จากรูปที่ 4.20 จะได้ว่าผลการทดลองเมื่อทำการเปรียบเทียบประสิทธิภาพของวิธีการจำแนกที่นำเสนอโดยใช้การสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D กับวิธีการจำแนกโดยใช้เครื่องมือ SentimentAnalysis ในแง่ตัวชี้วัด F1 จะสรุปได้ดังนี้

- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี k-Nearest Neighbors วิธี Artificial Neural Networks และวิธี Support Vector Machine จะให้ประสิทธิภาพดีที่สุด สำหรับทุกชุดข้อมูลการทดลอง
- วิธีการจำแนกที่นำเสนอด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D เมื่อใช้กับการจำแนกด้วยวิธี Naive Bayes จะให้ประสิทธิภาพดีที่สุด สำหรับชุดข้อมูล DS3 DS4 DS5 DS6 DS7 และ DS8

จากผลการทดลอง ดังรูปที่ 4.18 – 4.21 เมื่อทำการเปรียบเทียบประสิทธิภาพของวิธีการจำแนกที่นำเสนอกับวิธีการจำแนกที่ใช้เครื่องมือ SentimentAnalysis ในแง่ของตัวชี้วัดความถูกต้อง Accuracy Precision Recall และ F1 สามารถสรุปได้ว่า สำหรับทุกชุดข้อมูลการทดลอง วิธีการจำแนกที่นำเสนอที่ใช้วิธี k-Nearest Neighbors และวิธี Artificial Neural Networks ร่วมกับด้วยคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D จะมีประสิทธิภาพดีที่สุดทุกตัวชี้วัดความถูกต้องเมื่อเทียบกับวิธีการจำแนกที่ใช้เครื่องมือ SentimentAnalysis

4.2.8 การทดลองเปรียบเทียบผลการจำแนกบนชุดข้อมูลที่มีจำนวนข้อความเท่ากัน

จากการทดลองในหัวข้อก่อนหน้านี้ จะเห็นว่าชุดข้อมูลที่เป็นข้อความแสดงความคิดเห็นที่ใช้ในการทำการทดลองจะมีจำนวนของข้อความในแต่ละชุดข้อมูลที่แตกต่างกันไปและบางชุดข้อมูลมีจำนวนของข้อความที่เป็นข้อความเชิงบวกและข้อความเชิงลบที่แตกต่างกันอย่างมาก (imbalanced dataset)

ในหัวข้อนี้ผู้วิจัยได้ทำการทดลองเปรียบเทียบประสิทธิภาพของวิธีการสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอกับคุณลักษณะแทนข้อความแบบดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF โดยใช้ชุดข้อมูลที่มีจำนวนของข้อความที่เท่ากันทั้ง 8 ชุดข้อมูล และในแต่ละชุดข้อมูลจะมีข้อความทั้งหมด 1,000 ข้อความ ซึ่งแบ่งออกเป็นข้อความเชิงบวกจำนวน 500 ข้อความและข้อความเชิงลบจำนวน 500 ข้อความ แล้วนำมาเปรียบเทียบประสิทธิภาพการจำแนกโดยทำการเลือกผลจากพารามิเตอร์ที่ให้ผลดีที่สุดสำหรับแต่ละวิธีการสกัดคุณลักษณะแทนข้อความในแต่ละวิธีการจำแนกบนทุกชุดข้อมูล และทำการเปรียบเทียบประสิทธิภาพผลในแง่ความถูกต้องของการจำแนกขั้วความคิดเห็นด้วยตัวชี้วัดความถูกต้อง Accuracy Precision Recall และ F1 ดังตารางที่ 4.13 – 4.20

ตาราง 4.13 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS1 - Amazonที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.7200	0.7517	0.7200	0.7103
TF-IDF	0.4677	0.4677	0.4677	0.4677
V4D	0.8620	0.8620	0.8620	0.8620
V8D	0.8670	0.8670	0.8670	0.8670
NB:				
TF	0.5000	0.4000	0.5000	0.3402
TF-IDF	0.5000	0.2500	0.5000	0.3333
V4D	0.8670	0.8720	0.8670	0.8665
V8D	0.6520	0.7560	0.6520	0.6126
ANN:				
TF	0.8030	0.8051	0.8030	0.8026
TF-IDF	0.5830	0.6830	0.5830	0.5182
V4D	0.8650	0.8737	0.8650	0.8642
V8D	**0.8770***	**0.8816***	**0.8770***	**0.8766***
SVM:				
TF	0.8270	0.8295	0.8270	0.8266
TF-IDF	0.5880	0.7136	0.5880	0.5171
V4D	0.8600	0.8659	0.8600	0.8594
V8D	0.8700	0.8759	0.8700	0.8695

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

ตาราง 4.14 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS2 - IMDb ที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.6480	0.6480	0.6480	0.6480
TF-IDF	0.6362	0.6362	0.6362	0.6362
V4D	0.7990	0.8006	0.7990	0.7987
V8D	0.7990	0.8006	0.7990	0.7987
NB:				
TF	0.5000	0.2500	0.5000	0.3333
TF-IDF	0.5000	0.2500	0.5000	0.3333
V4D	0.5460	0.6745	0.5460	0.4444
V8D	0.7850	0.7940	0.7850	0.7832
ANN:				
TF	0.7420	0.7420	0.7420	0.7420
TF-IDF	0.5600	0.6465	0.5600	0.4917
V4D	0.7950	0.7950	0.7950	0.7950
V8D	**0.8040***	**0.8040***	**0.8040***	**0.8040***
SVM:				
TF	0.7730	0.7730	0.7730	0.7730
TF-IDF	0.5610	0.5610	0.5610	0.5610
V4D	0.7907	0.7907	0.7907	0.7907
V8D	0.7900	0.7980	0.7900	0.7886

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

ตาราง 4.15 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS3 - Yelp ที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.6830	0.6830	0.6830	0.6830
TF-IDF	0.5430	0.5430	0.5430	0.5430
V4D	0.8230	0.8230	0.8230	0.8230
V8D	**0.8260***	0.8331	**0.8260***	**0.8252***
NB:				
TF	0.4950	0.4184	0.4950	0.3408
TF-IDF	0.4990	0.2497	0.4990	0.3329
V4D	0.7780	0.8174	0.7780	0.7651
V8D	0.8150	0.8221	0.8150	0.8140
ANN:				
TF	0.7460	0.7460	0.7460	0.7460
TF-IDF	0.5630	0.5630	0.5630	0.5630
V4D	0.8140	0.8140	0.8140	0.8140
V8D	**0.8260***	**0.8336***	**0.8260***	0.8250
SVM:				
TF	0.8090	0.8103	0.8090	0.8088
TF-IDF	0.5760	0.5760	0.5760	0.5760
V4D	0.8190	0.8190	0.8190	0.8190
V8D	0.8190	0.8190	0.8190	0.8190

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

ตาราง 4.16 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS4 - Apparel ที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.5330	0.6232	0.5330	0.4233
TF-IDF	0.5020	0.4005	0.5020	0.3394
V4D	0.8880	0.8984	0.8880	0.8864
V8D	0.9130	0.9210	0.9130	0.9115
NB:				
TF	0.5340	0.4949	0.5340	0.4650
TF-IDF	0.5110	0.5253	0.5110	0.3812
V4D	0.8520	0.8681	0.8520	0.8490
V8D	0.7490	0.7419	0.7490	0.7124
ANN:				
TF	0.8300	0.8459	0.8300	0.8270
TF-IDF	0.6080	0.7604	0.6060	0.5470
V4D	0.8920	0.9024	0.8920	0.8904
V8D	**0.9140***	**0.9218***	**0.9140***	**0.9126***
SVM:				
TF	0.8170	0.8399	0.8170	0.8131
TF-IDF	0.5890	0.7518	0.5890	0.5082
V4D	0.8500	0.8656	0.8500	0.8470
V8D	0.8610	0.8782	0.8610	0.8582

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

ตาราง 4.17 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS5 - Health ที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.6940	0.7033	0.6940	0.6901
TF-IDF	0.5500	0.6591	0.5500	0.4534
V4D	0.8900	0.8925	0.8900	0.8898
V8D	0.9490	0.9490	0.9490	0.9490
NB:				
TF	0.5390	0.5855	0.5390	0.4514
TF-IDF	0.5080	0.5687	0.5080	0.3557
V4D	0.8630	0.8647	0.8630	0.8629
V8D	0.7550	0.8017	0.7550	0.7427
ANN:				
TF	0.7820	0.7800	0.7790	0.7790
TF-IDF	0.5910	0.6745	0.5910	0.5339
V4D	0.8920	0.8945	0.8920	0.8918
V8D	0.9460	0.9461	0.9460	0.9460
SVM:				
TF	0.8470	0.8504	0.8470	0.8466
TF-IDF	0.6240	0.7175	0.6240	0.5785
V4D	0.8980	0.9002	0.8980	0.8979
V8D	**0.9500***	**0.9500***	**0.9500***	**0.9500***

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

ตาราง 4.18 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS6 - Music ที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.6010	0.6517	0.6010	0.5667
TF-IDF	0.5180	0.6172	0.5180	0.3865
V4D	0.8960	0.8970	0.8960	0.8959
V8D	**0.9490***	**0.9490***	**0.9490***	**0.9490***
NB:				
TF	0.5150	0.5350	0.5150	0.4436
TF-IDF	0.5020	0.4632	0.5020	0.3523
V4D	0.8880	0.8905	0.8880	0.8878
V8D	0.7470	0.7849	0.7470	0.7372
ANN:				
TF	0.7730	0.7744	0.7730	0.7727
TF-IDF	0.5710	0.6638	0.5710	0.4992
V4D	0.8990	0.8997	0.8990	0.8990
V8D	0.9420	0.9421	0.9420	0.9420
SVM:				
TF	0.8150	0.8166	0.8150	0.8148
TF-IDF	0.5710	0.6817	0.5710	0.4947
V4D	0.9010	0.9015	0.9010	0.9010
V8D	0.9480	0.9481	0.9480	0.9480

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

ตาราง 4.19 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS7 - Sports ที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.5670	0.6444	0.5670	0.5102
TF-IDF	0.5200	0.6410	0.5200	0.4075
V4D	0.9020	0.9031	0.9020	0.9019
V8D	**0.9460***	**0.9461***	**0.9460***	**0.9460***
NB:				
TF	0.5610	0.5681	0.5610	0.5157
TF-IDF	0.5060	0.5232	0.5060	0.3796
V4D	0.8780	0.8784	0.8780	0.8780
V8D	0.8220	0.8536	0.8220	0.8109
ANN:				
TF	0.8180	0.8190	0.8160	0.8160
TF-IDF	0.6000	0.7000	0.5980	0.5444
V4D	0.9080	0.9092	0.9080	0.9079
V8D	0.9450	0.9451	0.9450	0.9450
SVM:				
TF	0.8610	0.8617	0.8610	0.8609
TF-IDF	0.6140	0.7208	0.6140	0.5602
V4D	0.8920	0.8949	0.8920	0.8918
V8D	0.9440	0.9441	0.9440	0.9440

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

ตาราง 4.20 การเปรียบเทียบประสิทธิภาพของผลการจำแนกบนชุดข้อมูล DS8 - US Airline ที่มีจำนวนข้อความเท่ากัน สำหรับแต่ละวิธีการจำแนกกับการสกัดคุณลักษณะแบบต่างๆ

เวกเตอร์แทนข้อความ / พารามิเตอร์	Accuracy	Precision	Recall	F1
k -NN:				
TF	0.5310	0.7190	0.5310	0.4095
TF-IDF	0.5160	0.6004	0.5160	0.4504
V4D	0.9410	0.9413	0.9410	0.9410
V8D	**0.9500***	**0.9500***	**0.9500***	**0.9500***
NB:				
TF	0.5000	0.3833	0.5000	0.3368
TF-IDF	0.5000	0.2500	0.5000	0.3333
V4D	0.9390	0.9392	0.9390	0.9390
V8D	0.8090	0.8483	0.8090	0.8016
ANN:				
TF	0.8440	0.8463	0.8440	0.8438
TF-IDF	0.5960	0.7308	0.5960	0.5266
V4D	0.9400	0.9403	0.9400	0.9400
V8D	0.9490	0.9490	0.9490	0.9490
SVM:				
TF	0.8430	0.8520	0.8430	0.8429
TF-IDF	0.5940	0.7250	0.5940	0.5244
V4D	0.9390	0.9394	0.9390	0.9390
V8D	**0.9500***	**0.9500***	**0.9500***	**0.9500***

หมายเหตุ * แสดงค่าประสิทธิภาพที่ดีที่สุดสำหรับแต่ละชุดข้อมูล

จากผลการทดลองเมื่อเปรียบเทียบประสิทธิภาพของวิธีการสกัดคุณลักษณะแทนข้อความในแต่ละวิธีการจำแนกบนชุดข้อมูลที่มีจำนวนของข้อความที่เท่ากันทั้ง 8 ชุดข้อมูล ดังตารางที่ 4.13 – 4.20 การวัดประสิทธิภาพของเวกเตอร์แทนข้อความที่ใช้ร่วมกับวิธีการจำแนกทั้ง 4 แบบ แล้วให้ประสิทธิภาพในแง่ของตัวชี้วัดที่ดีที่สุดสำหรับแต่ละชุดข้อมูล พิจารณาได้ดังนี้

- สำหรับชุดข้อมูล DS1 – Amazon DS2 – IMDb และ DS4 – Apparel เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับวิธีการจำแนก Artificial Neural Networks
- สำหรับชุดข้อมูล DS3 – Yelp DS6 - Music และ DS7 – Sports เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับวิธีการจำแนก k -Nearest Neighbors
- สำหรับชุดข้อมูล DS5 – Health เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับวิธีการจำแนก Support Vector Machine
- สำหรับชุดข้อมูล DS8 - US Airline เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ร่วมกับวิธีการจำแนก k -Nearest Neighbors และ Support Vector Machine

จะเห็นได้ว่า เมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ให้ประสิทธิภาพดีที่สุดสำหรับทุกชุดข้อมูล

บทที่ 5

บทสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึง บทสรุป และข้อเสนอแนะในการทำวิจัย โดยมีรายละเอียดดังต่อไปนี้

5.1 บทสรุป

งานวิจัยนี้ได้นำเสนอการสกัดคุณลักษณะแทนข้อความในรูปของเวกเตอร์ 2 รูปแบบ คือ เวกเตอร์ V4D และเวกเตอร์ V8D ซึ่งเป็นเวกเตอร์ที่มีมิติน้อย โดยการพิจารณาคุณลักษณะที่ได้จากคำศัพท์บอกการปฏิเสธที่มีความสำคัญต่อความหมายของข้อความและการจำแนกขั้วความคิดเห็น และผู้วิจัยได้ทำการเปรียบเทียบประสิทธิภาพระหว่างคุณลักษณะที่นำเสนอกับคุณลักษณะดั้งเดิม ได้แก่ เวกเตอร์ TF และเวกเตอร์ TF-IDF ทำการทดลองโดยใช้ชุดข้อมูลแสดงความคิดเห็นที่มีความหลากหลายของโดเมน 8 ชุดข้อมูล และประเมินผลลัพธ์จากการจำแนกประเภทโดยวิธี k-Nearest Neighbors วิธี Naive Bayes วิธี Artificial Neural Networks และวิธี Support Vector Machine ด้วยการทดสอบแบบ 5-fold Cross Validation และประเมินประสิทธิภาพในแง่ของการใช้พื้นที่จัดเก็บข้อมูล ระยะเวลาที่ใช้ในการประมวลผล และค่าความถูกต้องของการจำแนกด้วยตัวชี้วัด ได้แก่ Accuracy Precision Recall และ F1 ในงานวิจัยนี้ยังได้ทำการเปรียบเทียบวิธีการจำแนกโดยใช้การสกัดคุณลักษณะแทนข้อความที่นำเสนอกับวิธีการจำแนกโดยใช้เครื่องมือ SentimentAnalysis และนอกจากนี้ผู้วิจัยยังได้ทำการทดลองเปรียบเทียบประสิทธิภาพของวิธีการสกัดคุณลักษณะแทนข้อความที่นำเสนอกับคุณลักษณะแทนข้อความแบบดั้งเดิม โดยใช้ชุดข้อมูลที่มีจำนวนของข้อความที่เท่ากันสำหรับทุกชุดข้อมูล

ผลการทดลองการสกัดคุณลักษณะแทนข้อความเพื่อการจำแนกขั้วความคิดเห็น สามารถสรุปได้ 5 ประเด็นดังนี้

1) ประเด็นความสามารถในการจำแนกขั้วความคิดเห็น จากการทดลองพบว่าการสกัดคุณลักษณะแทนข้อความในรูปของเวกเตอร์ 2 รูปแบบ คือ เวกเตอร์ V4D และเวกเตอร์ V8D สามารถช่วยให้ผลการจำแนกขั้วความคิดเห็นมีความถูกต้องมากยิ่งขึ้น และมีประสิทธิภาพในแง่ของพื้นที่ในการจัดเก็บข้อมูล และในแง่ของเวลาที่ใช้ในการประมวลผลได้ดีที่สุด

2) ประเด็นเปรียบเทียบประสิทธิภาพระหว่างวิธีการสกัดคุณลักษณะที่นำเสนอกับวิธีการสกัดคุณลักษณะแบบดั้งเดิม พบว่าเมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ให้ประสิทธิภาพดีที่สุดสำหรับชุดข้อมูล DS1 DS2 และ DS3 แต่สำหรับชุดข้อมูล DS4 DS5 DS6 และ DS7 เมื่อใช้คุณลักษณะแทนข้อความเวกเตอร์ TF ร่วมกับเวกเตอร์ V8D โดยใช้วิธีการจำแนก Support Vector Machine จะให้ประสิทธิภาพที่ดีที่สุด และสำหรับชุดข้อมูล DS8 เมื่อใช้คุณลักษณะแทนข้อความเวกเตอร์ TF ร่วมกับเวกเตอร์ V8D โดยใช้วิธีการจำแนก Artificial Neural Networks และวิธี Support Vector Machine จะให้ประสิทธิภาพที่ดีที่สุด

3) ประเด็นเปรียบเทียบการจำแนกประเภทด้วยวิธี วิธี k -Nearest Neighbors วิธี Naive Bayes วิธี Artificial Neural Networks และวิธี Support Vector Machine จากผลการทดลอง พบว่าเมื่อใช้การสกัดคุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ร่วมกับวิธีการจำแนกวิธี Artificial Neural Networks และวิธี Support Vector Machine จะให้ประสิทธิภาพที่ดีกว่า วิธีการจำแนกวิธี k -Nearest Neighbors และวิธี Naive Bayes แต่การสร้างโมเดลด้วยวิธี Artificial Neural Networks และวิธี Support Vector Machine จะใช้เวลาในการประมวลผลข้อมูลที่นานกว่ามาก

4) ประเด็นเปรียบเทียบวิธีการจำแนกโดยใช้การสกัดคุณลักษณะแทนข้อความที่นำเสนอกับวิธีการจำแนกโดยใช้ใช้เครื่องมือ SentimentAnalysis พบว่าวิธีการจำแนกที่นำเสนอร่วมกับคุณลักษณะแทนข้อความด้วยเวกเตอร์ V8D ที่นำเสนอ จะให้ประสิทธิภาพในแง่การจำแนกที่ดีกว่าเมื่อเทียบกับวิธีการจำแนกที่ใช้เครื่องมือ SentimentAnalysis ในทุกตัวชี้วัดความถูกต้อง

5) ประเด็นเปรียบเทียบประสิทธิภาพระหว่างวิธีการสกัดคุณลักษณะที่นำเสนอกับวิธีการสกัดคุณลักษณะแบบดั้งเดิม โดยใช้ชุดข้อมูลที่มีจำนวนของข้อความที่เท่ากันสำหรับทุกชุดข้อมูล พบว่าเมื่อใช้คุณลักษณะแทนข้อความด้วยเวกเตอร์ V4D และเวกเตอร์ V8D ที่นำเสนอ จะให้ประสิทธิภาพดีที่สุดในทุกตัวชี้วัดความถูกต้องสำหรับทุกชุดข้อมูล

5.2 ข้อเสนอแนะและงานในอนาคต

จากผลการศึกษา จะเห็นว่า ข้อมูลจากข้อความแต่ละโดเมนเหมาะกับคุณลักษณะเวกเตอร์แทนข้อความและวิธีการจำแนกที่ต่างกันไป แนวคิดที่น่าสนใจในการพัฒนาต่อยอดคือการสร้างโมเดล

ขึ้นมาหลายโมเดลที่มีการใช้คุณลักษณะและการจำแนกหลายวิธีให้มีการจำแนกร่วมกัน อาจนำไปสู่การพัฒนาการจำแนกข้อความแสดงความคิดเห็นที่มีประสิทธิภาพต่อไป

5.3 บทวิจารณ์

โดยทั่วไปข้อความแสดงความคิดเห็นมักจะประกอบด้วยคำศัพท์ต่างๆ ที่มีความหมายในทิศทางเชิงบวกและเชิงลบ และนอกจากนี้ยังมีคำศัพท์ที่บ่งบอกถึงทิศทางการปฏิเสธ ได้แก่ “no” และ “not” ซึ่งเป็นคำศัพท์ที่มีนัยสำคัญในการบ่งบอกว่าข้อความนั้นๆ เป็นการแสดงความคิดเห็นในทิศทางเชิงลบหรือเชิงบวก จึงทำให้มีความสำคัญต่อความหมายของข้อความและการจำแนกขั้วความคิดเห็น โดยเมื่อคำปฏิเสธดังกล่าวนี้ปรากฏอยู่ในข้อความแล้วจะมีผลต่อคำศัพท์ที่เกี่ยวข้องโดยตรง เนื่องจากคำปฏิเสธจะทำการกลับขั้วความคิดเห็นของคำศัพท์ที่เกี่ยวข้องให้เป็นขั้วความคิดเห็นตรงกันข้ามจากขั้วความคิดเห็นเดิมของคำศัพท์นั้น และในบางข้อความแสดงความคิดเห็นอาจจะปรากฏคำศัพท์ที่ไม่สามารถจะสื่อถึงความหมายว่าเป็นทิศทางเชิงบวกหรือเชิงลบได้ แต่เมื่อมีคำปฏิเสธปรากฏในข้อความแล้ว จะทำให้ข้อความนั้นสื่อความหมายในทิศทางเชิงลบไปในทันที จะเห็นว่าคำปฏิเสธดังที่กล่าวมามีความสำคัญต่อข้อความแสดงความคิดเห็น ดังนั้นควรทำการพิจารณาคำปฏิเสธนี้ร่วมด้วยในการวิเคราะห์ข้อความแสดงความคิดเห็นสำหรับการสกัดคุณลักษณะแทนข้อความเพื่อการจำแนกขั้วความคิดเห็น

บรรณานุกรม

ณิชาภัทร ปิ่นโพธิ์, และนิวรรณ วัฒนกิจรุ่งโรจน์. (2561). *การแทนข้อความแสดงความคิดเห็นด้วยเวกเตอร์ที่ใช้พื้นที่น้อย*. การประชุมวิชาการระดับประเทศด้านเทคโนโลยีสารสนเทศ ครั้งที่ 10. ขอนแก่น. 24-25 ตุลาคม 2561. 109-114.

Akilandeswari, J., & Jothi, G. (2018). Sentiment Classification of Tweets with Non-Language Features. *Procedia Computer Science*, *143*, 426–433. https://doi.org/10.1016/j.procs.2018.10.414

Ankit, & Saleena, N. (2018). An Ensemble Classification System for Twitter Sentiment Analysis. *Procedia Computer Science*, *132*, 937–946. https://doi.org/10.1016/j.procs.2018.05.109

Bansal, B., & Srivastava, S. (2018). Sentiment classification of online consumer reviews using word vector representations. *Procedia Computer Science*, *132*, 1147–1153. https://doi.org/10.1016/j.procs.2018.05.029

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 440–447. https://www.aclweb.org/anthology/P07-1056

Chae, M.K., Alsadoon, A., Prasad, P.W.C., & Elchouemi, A. (2017). Spam filtering email classification (SFECM) using gain and graph mining algorithm. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, 1–7. https://doi.org/10.1109/CCWC.2017.7868411

Cho, S.H., & Kang, H.B. (2012). Text sentiment classification for SNS-based marketing using domain sentiment dictionary. *2012 IEEE International Conference on Consumer Electronics (ICCE)*, 717–718. https://doi.org/10.1109/ICCE.2012.6162053

Da Silva, N.F.F., Hruschka, E.R., & Hruschka, E.R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, *66*, 170–179. https://doi.org/10.1016/j.dss.2014.07.003

Farooq, U., Mansoor, H., Nongaillard, A., Ouzrout, Y., & Qadir, M.A. (2017). Negation Handling in Sentiment Analysis at Sentence Level. *JCP*. https://doi.org/10.17706/jcp.12.5.470-478

Feuerriegel, S., & Proellochs, N. (2019). *SentimentAnalysis: Dictionary-Based Sentiment Analysis* (Version 1.3-3) [Computer software]. Retrieved October 10, 2019, from https://CRAN.R-project.org/package=SentimentAnalysis

Gharehchopogh, F.S., & Khalifelu, Z.A. (2011). Analysis and evaluation of unstructured data: Text mining versus natural language processing. *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*, 1–4. https://doi.org/10.1109/ICAICT.2011.6111017

Guo, T., Dong, J., Li, H., & Gao, Y. (2017). Simple convolutional neural network on image classification. *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 721–724. https://doi.org/10.1109/ICBDA.2017.8078730

Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd ed.). Prentice Hall: New York.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168–177. https://doi.org/10.1145/1014052.1014073

Isah, H., Trundle, P., & Neagu, D. (2014). Social media analysis for product safety using text mining and sentiment analysis. *2014 14th UK Workshop on Computational Intelligence (UKCI)*, 1–7. https://doi.org/10.1109/UKCI.2014.6930158

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, *100*, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037

Kaggle. (2019). *Twitter US Airline Sentiment* (Version 4). Retrieved September 15, 2019, from https://kaggle.com/crowdflower/twitter-airline-sentiment

Lovins, J.B. (1968). Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*.

Manning, C.D., Raghavan, P., & Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press.

Mishu, S.Z., & Rafiuddin, S.M. (2016). Performance analysis of supervised machine learning algorithms for text classification. *2016 19th International Conference on Computer and Information Technology (ICCIT)*, 409–413. https://doi.org/10.1109/ICCITECHN.2016.7860233

Paice, C.D. (1990). Another stemmer. *ACM SIGIR Forum*, *24*(3), 56–61. https://doi.org/10.1145/101306.101310

Pang, B., & Lee, L. (2004). *Movie review data* (Version 2.0). Retrieved May 7, 2019, from http://cs.cornell.edu/people/pabo/movie-review-data

Ramteke, J., Shah, S., Godhia, D., & Shaikh, A. (2016). Election result prediction using Twitter sentiment analysis. *2016 International Conference on Inventive Computation Technologies (ICICT)*, *1*, 1–5. https://doi.org/10.1109/INVENTIVE.2016.7823280

Shaw, M.J., & Gentry, J.A. (1990). Inductive learning for risk classification. *IEEE Expert*, *5*(1), 47–53. https://doi.org/10.1109/64.50856

Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 1310–1315.

Tan, P.N., Steinbach, M., Karpatne, A., & Kumar, V. (2019). *Introduction to Data Mining* (2nd ed.). New York, NY : Pearson Education.

Tripathy, A., Agrawal, A., & Rath, S.K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117–126. https://doi.org/10.1016/j.eswa.2016.03.028

Van Rijsbergen, C.J., Robertson, S.E., & Porter, M.F. (1980). *New models in probabilistic information retrieval*. London: British Library.

Zhang, X., & Zheng, X. (2016). Comparison of Text Sentiment Analysis Based on Machine Learning. *2016 15th International Symposium on Parallel and Distributed Computing (ISPDC)*, 230–233. https://doi.org/10.1109/ISPDC.2016.39

ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์ในงานประชุมวิชาการ NCIT 2018

เรื่อง การแทนข้อความแสดงความคิดเห็นด้วยเวกเตอร์ที่ใช้พื้นที่น้อย

Conference การประชุมวิชาการระดับประเทศด้านเทคโนโลยีสารสนเทศ ครั้งที่ 10 (NCIT 2018)

สถานที่ โรงแรมพูลแมน ขอนแก่น ราชา ออคิด, จังหวัดขอนแก่น

วันที่ 24-25 ตุลาคม 2561

ประวัติผู้เขียน

ชื่อ สกุล นางสาวณิชาภัทร ปิ่นโพธิ์

รหัสประจำตัวนักศึกษา 6010220087

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2559

ทุนการศึกษา

ทุนอุดหนุนการวิจัยเพื่อวิทยานิพนธ์ บัณฑิตศึกษา มหาวิทยาลัยสงขลานครินทร์ ประจำปีงบประมาณ 2562

การตีพิมพ์เผยแพร่ผลงาน

ณิชาภัทร ปิ่นโพธิ์, และนิวรรณ วัฒนกิจรุ่งโรจน์. (2561). *การแทนข้อความแสดงความคิดเห็นด้วยเวกเตอร์ที่ใช้พื้นที่น้อย*. การประชุมวิชาการระดับประเทศด้านเทคโนโลยีสารสนเทศ ครั้งที่ 10. ขอนแก่น. 24-25 ตุลาคม 2561. 109-114.