โครงการ

การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ การวิเคราะห์ความรู้สึกของความคิดเห็นที่มีต่อร้านอาหารบนเฟสบุ๊ค

Sentiment Analysis of Restaurant Reviews Posted on Facebook

ชื่อนิสิต นายธนกฤต งามสุนทรวงศ์ 583 36322 23

นายภูริวัฒน์ มีชัยเจริญยิ่ง 583 36511 23

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2561

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของโครงงานทางวิชาการที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของโครงงานทางวิชาการที่ส่งผ่านทางคณะที่สังกัด

The abstract and full text of senior projects in Chulalongkorn University Intellectual Repository(CUIR)

are the senior project authors' files submitted through the faculty.

การวิเคราะห์ความรู้สึกของความคิดเห็นที่มีต่อร้านอาหารบนเฟสบุ๊ค

นาย ธนกฤต งามสุนทรวงศ์
นาย ภูริวัฒน์ มีชัยเจริญยิ่ง

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

SENTIMENT ANALYSIS OF RESTAURANT REVIEWS POSTED ON FACEBOOK

Thanakrit Ngamsoonthornwong

Pooriwat Meechaicharoenying

A Project Submitted in Partial Fulfillment of the Requirements

for the Degree of Bachelor of Science Program in Computer Science

Department of Mathematics and Computer Science

Faculty of Science Chulalongkorn University

Academic Year 2018

Copyright of Chulalongkorn University

หัวข้อโครงงาน	การวิเคราะห์ความรู้สึกของความคิดเห็นร้านอาหารบนเฟสบุ๊ค
จัดทำโดยโดย	1. นาย ธนกฤต งามสุนทรวงศ์
	2. นาย ภูริวัฒน์ มีชัยเจริญยิ่ง
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงงานหลัก	อาจารย์ ดร. อาธร เหลืองสดใส

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

..... หัวหน้าภาควิชาคณิตศาสตร์

(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงงาน

..... อาจารย์ที่ปรึกษาโครงงานหลัก

(อาจารย์ ดร.อาธร เหลืองสดใส)

..... กรรมการ

(รศ.ดร.พีระพนธ์ โสพัศสถิตย์)

..... กรรมการ

(ผศ.ดร.เอื้อมพร ฟักสุวรรณ)

ธนกฤต งามสุนทรวงศ์, ภูริวัฒน์ มีชัยเจริญยิ่ง: การวิเคราะห์ความรู้สึกของความคิดเห็นร้านอาหารบนเฟสบุ๊ค (Sentiment analysis of restaurant reviews posted on Facebook) อ.ที่ปรึกษาโครงงานหลัก: อาจารย์ ดร.อาธร เหลืองสดใส, 60 หน้า.

ในปัจจุบันเราจะเห็นว่าประชากรส่วนหนึ่งในประเทศไทยเริ่มใช้โซเชียลมีเดีย เช่น เฟสบุ๊ค, ทวิตเตอร์ เพื่อแสดงความคิดเห็นเกี่ยวกับการให้บริการของธุรกิจ ทำให้ธุรกิจเกิดความจำเป็นที่จะต้องจ้างพนักงานมาเพื่ออ่านข้อความที่ลูกค้าแสดงความคิดเห็น เพื่อนำความคิดเห็นต่าง ๆ ไปปรับปรุงการบริการให้ดีขึ้น ในจุดนี้ผู้พัฒนาเล็งเห็นว่าหากเราต้องอ่านความคิดเห็นทั้งหมดด้วยตัวเองนั้น อาจต้องใช้เวลาในการอ่านความคิดเห็นแต่ละครั้งเป็นจำนวนมาก ประกอบกับในปัจจุบันมีการนำเทคนิคการวิเคราะห์ข้อมูลมาประยุกต์ใช้ในการจัดกลุ่มข้อความออกเป็นหมวดหมู่ อย่างเช่นการจัดข้อความเป็นหมวดหมู่ตามความหมายความรู้สึกเชิงบวกหรือเชิงลบ ดังนั้น ผู้พัฒนาจึงพัฒนาโปรแกรมเพื่อช่วยจัดกลุ่มความคิดเห็นที่มีต่อร้านอาหารบนเฟสบุ๊คออกเป็นหมวดหมู่เพื่ออำนวยความสะดวกให้กับธุรกิจในการรับความคิดเห็นเกี่ยวกับบริการที่ลูกค้าได้รับจากธุรกิจ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต.....ธนกฤต.....

ลายมือชื่อนิสิต.....ภูริวัฒน์.....

สาขาวิชาวิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงาน.....

ปีการศึกษา 2561

5833632223, 5833651123: MAJOR COMPUTER SCIENCE
KEYWORD: SENTIMENT ANALYSIS/ FACEBOOK/ NEURAL NETWORK/ SUPPORT VECTOR MACHINE/ NAIVE BAYES/ LOGISTIC REGRESSION/ SKLEARN/ NLTK/ DEEPCUT

THANAKRIT NGAMSOONTHORNWONG, POORIWAT MEECHAICHAROENYING: SENTIMENT ANALYSIS OF RESTAURANT REVIEWS POSTED ON FACEBOOK: ADVISER: ARTHORN LUANSODSAI, Ph.D., 60 pp.

At present, there are numbers of Thai people using social media, such as Facebook or Twitter, to give feedback about services of businesses. Businesses, therefore, need several employees to read or monitor these comments in order to improve their services. This manual process wastes employee time and increase costs to businesses. Due to the gaining popularity of Data Analytic research, we can apply data mining techniques to classify automatically data into meaningful classes. For our work, we classify user comments from Facebook into 'Positive' and 'Negative' classes. As a result, businesses can apply our work to receive the positive or negative feedback from their customers and to improve their services accordingly.

Department: Mathematics and Computer science Student's Signature ธนกฤต

Student's Signature

Field of study: Computer science Advisor's Signature

Academic Year: 2018

กิตติกรรมประกาศ

โครงงานเรื่องการวิเคราะห์ความรู้สึกของความคิดเห็นที่มีต่อร้านอาหารบนเฟสบุ๊ค สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องด้วยการสนับสนุนและคำชี้แนะจาก อ.ดร.อาธร เหลืองสดใส อาจารย์ที่ปรึกษาโครงงานผู้คอยให้คำปรึกษาชี้แนะและช่วยแก้ไขตลอดโครงงาน รศ.ดร.พีระพนธ์ โสพัศสถิตย์ และ ผศ.ดร.เอื้อมพร ฟักสุวรรณ กรรมการสอบทั้งสองท่านผู้ให้ข้อเสนอแนะในการพัฒนาโครงงานนี้ให้ดียิ่งขึ้น อาจารย์ผู้สอนทุกท่านที่ได้มอบความรู้ให้นำมาประยุกต์ใช้กับโครงงานนี้ และเพื่อนร่วมสาขาวิชาวิทยาการคอมพิวเตอร์ผู้ให้ความช่วยเหลือในหลาย ๆ ด้าน พวกข้าพเจ้าขอขอบพระคุณทุกท่านที่กล่าวถึงและท่านที่ไม่ได้กล่าวถึงเป็นอย่างยิ่งสำหรับความช่วยเหลือที่ได้รับ

สารบัญ

หน้า

กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ซ
สารบัญรูปภาพ ..... ฎ
สารบัญตาราง ..... ฐ
บทที่ 1 1
1.1 ความเป็นมาและมูลเหตุจูงใจของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 1
1.4 วิธีการดำเนินงาน 2
บทที่ 2 5
2.1 การวิเคราะห์อารมณ์และความรู้สึก (Sentiment Analysis) 5
2.2 เครื่องมือการจำแนกข้อความนาอีฟเบย์ (Naive Bayes classifiers) 6
2.3 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) 9
2.4 โครงข่ายประสาทเทียม (Neural Network) 10
2.5 การถดถอยโลจิสติค (Logistic Regression) 14
2.6 Scikit-learn (Sklearn) 16
2.7 Natural Language Toolkit (NLTK) 17
2.8 Deepcut 18
2.9 Facebook SDK for Python 19
2.10 Tensorflow 19

บทที่ 3 21
3.1 การรวบรวมข้อมูล 21
3.1.1 ข้อความแสดงความคิดเห็นต่อร้านอาหารจากช่องทางแสดงความคิดเห็นของเพจบนเฟสบุ๊ค 21
3.1.2 ข้อความแสดงความคิดเห็นต่อร้านอาหารจากเว็บไซต์ Wongnai.com 21
3.2 การวิเคราะห์ข้อมูล 21
3.3 ขั้นตอนการวิเคราะห์ความคิดเห็น 22
3.4 การประเมิณผลเพื่อเลือกวิธีที่ใช้ในการจำแนกความคิดเห็น 22
3.4.1 คอนฟิวชันเมทริกซ์ (Confusion matrix) 23
3.4.2 ค่าความถูกต้อง (Accuracy) 23
3.4.3 ค่าความแม่นยำ (Precision) 23
3.4.4 ค่าการเรียกคืน (Recall) 24
3.4.5 ค่าวัดประสิทธิภาพโดยรวม (F-measure) 24
บทที่ 4 25
4.1 การออกแบบการทำงานของระบบ 25
4.2 แผนภาพแสดงภาพรวมการทำงานในการวิเคราะห์อารมณ์ความรู้สึกของความคิดเห็นบนเฟสบุ๊ค 50
4.3 แผนภาพยูสเคส 52
4.4 ภาษาและโปรแกรมที่ใช้ในการพัฒนาระบบ 53
บทที่ 5 55
5.1.1 การทดสอบการจำแนกความคิดเห็นด้วย NLTK Naive Bayes 55
5.1.2 การทดสอบการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes 56
5.2 การทดสอบการจำแนกความคิดเห็นด้วย Support Vector Machine 57

5.3 การทดสอบการจำแนกความคิดเห็นด้วย Neural Network 58
5.4 การทดสอบการจำแนกความคิดเห็นด้วย Logistic Regression 59
5.5 สรุปผลการทดสอบ 60
บทที่ 6 61
6.1 สรุปผล 61
6.2 ผลที่ได้รับ 61
6.2.1 ส่วนของผู้พัฒนา 61
6.2.2 ส่วนของผู้ใช้งาน 62
6.3 ปัญหาและอุปสรรค 62
6.4 วิธีการแก้ปัญหา 62
ภาคผนวก 65
ภาคผนวก ก 66
ภาคผนวก ข 71
ประวัติผู้เขียน 76

สารบัญรูปภาพ

หน้า

ภาพที่ 2.1 สมการความน่าจะเป็นของเบย์ 7
ภาพที่ 2.2 สมการความน่าจะเป็นสำหรับเครื่องมือการจำแนกข้อความนาอีฟเบย์ 7
ภาพที่ 2.3 ตัวอย่างข้อมูลที่ทำการ training แล้ว 9
ภาพที่ 2.4 ตัวอย่างการจำแนกด้วยซัพพอร์ตเวกเตอร์แมชชีน 10
ภาพที่ 2.5 Neural Network 12
ภาพที่ 2.6 Logistic Regression 15
ภาพที่ 2.7 ภาพแสดงค่า Regression ที่ทายออกมาได้แกน Y เป็น 0.5 15
ภาพที่ 2.8 การใช้ไลบรารี่ sklearn เพื่อเรียกใช้ SVM 17
ภาพที่ 2.9 ใช้ไลบรารี่ sklearn เพื่อเรียกใช้ naive bayes 17
ภาพที่ 2.10 ตัดคำและติดป้ายกำกับชนิดของคำ 18
ภาพที่ 2.11 ระบุประเภทของเอนติตี้ 18
ภาพที่ 2.12 การเรียกใช้ deepcut และคำสั่งตัดคำ 19
ภาพที่ 2.13 การเรียกใช้ไลบรารี่ Facebook SDK และคำสั่งในการเข้าถึงข้อมูลบนเฟสบุ๊ค 19
ภาพที่ 2.14 การเรียกใช้ไลบรารี่ Tensorflow 20
ภาพที่ 4.1 หน้าหลักของ Facebook Graph Api 25
ภาพที่ 4.2 หน้าเข้าสู่ระบบ Facebook 26
ภาพที่ 4.3 ตำแหน่งของปุ่ม Add New App 26
ภาพที่ 4.4 หน้าสร้าง New App ID 27
ภาพที่ 4.5 วิธีเข้าใช้งาน Graph Api Explorer 27
ภาพที่ 4.6 วิธีการได้รับ Token 28
ภาพที่ 4.7 เข้าสู่ระบบด้วยบัญชีผู้ใช้ 29
ภาพที่ 4.8 หน้าเลือกเพจที่ต้องการดึงข้อความ 30
ภาพที่ 4.9 หน้าต่างก่อนกดยืนยัน 31
ภาพที่ 4.10 หน้า Graph API ที่แสดง Token จากเพจที่เลือก 32

ภาพที่ 4.11 ผลลัพธ์การดึงข้อมูลแสดงความคิดเห็นของร้านค้าบนเฟสบุ๊ค 33
ภาพที่ 4.12 ความคิดเห็นที่มีต่อร้านอาหารจาก wongnai 35
ภาพที่ 4.13 ไฟล์ความคิดเห็นแง่บวก 35
ภาพที่ 4.14 ไฟล์ความคิดเห็นแง่ลบ 36
ภาพที่ 4.15 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกนาอีฟเบย์ NLTK 41
ภาพที่ 4.16 ตัวอย่างผลลัพธ์จากการจำแนกความคิดเห็นด้วยตัวจำแนกนาอีฟเบย์ NLTK..... 42
ภาพที่ 4.17 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกซัพพอร์ตเวกเตอร์แมชชีน 43
ภาพที่ 4.18 ตัวอย่างผลลัพธ์การจำแนกความคิดเห็นด้วยตัวจำแนกซัพพอร์ตเวกเตอร์แมชชีน..... 43
ภาพที่ 4.19 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกนาอีฟเบย์ 44
ภาพที่ 4.20 ตัวอย่างผลลัพธ์ของการจำแนกความคิดเห็นด้วยตัวจำแนกนาอีฟเบย์ 44
ภาพที่ 4.21 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกโครงข่ายประสาทเทียม..... 47
ภาพที่ 4.22 ตัวอย่างผลลัพธ์การจำแนกความคิดเห็นด้วยตัวจำแนกโครงข่ายประสาทเทียม 48
ภาพที่ 4.23 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกการถดถอยเชิงโลจิสติก..... 49
ภาพที่ 4.24 ตัวอย่างผลลัพธ์การจำแนกความคิดเห็นด้วยตัวจำแนกการถดถอยเชิงโลจิสติก 49
ภาพที่ 4.25 ภาพรวมการวิเคราะห์อารมณ์ความรู้สึกของความคิดเห็นจากรีวิวบนเฟสบุ๊ค 51

สารบัญตาราง

หน้า

ตารางที่ 1.1 ตารางเวลาดำเนินการ 3
ตารางที่ 3.1 คอนฟิวชันเมทริกซ์ 23
ตารางที่ 4.1 ตารางอธิบายแผนภาพ Use case 53
ตารางที่ 5.1 confusion matrix ของการจำแนกความคิดเห็นด้วย NLTK Naive Bayes 56
ตารางที่ 5.2 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย NLTK Naive Bayes 56
ตารางที่ 5.3 confusion matrix ของการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes 57
ตารางที่ 5.4 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes 57
ตารางที่ 5.5 confusion matrix ของการจำแนกความคิดเห็นด้วย Support Vector Machine 58
ตารางที่ 5.6 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Support Vector Machine 58
ตารางที่ 5.7 confusion matrix ของการจำแนกความคิดเห็นด้วย Neural Network 59
ตารางที่ 5.8 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Neural Network 59
ตารางที่ 5.9 confusion matrix ของการจำแนกความคิดเห็นด้วย Logistic Regression 60
ตารางที่ 5.10 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Logistic Regression 60

บทที่ 1

บทนำ

1.1 ความเป็นมาและมูลเหตุจูงใจของโครงงาน

ในปัจจุบันเราจะเห็นว่าประชากรส่วนหนึ่งในประเทศไทยได้เริ่มใช้โซเชียลมีเดียเพื่อแสดงความคิดเห็นเกี่ยวกับการให้บริการของร้านอาหารหรือห้างร้านเป็นจำนวนมากทำให้ทางร้านอาหารหรือห้างร้านจำเป็นที่จะต้องจ้างพนักงานมาเพื่ออ่านข้อความที่ลูกค้าแสดงความคิดเห็นเพื่อนำไปใช้ปรับปรุงการให้บริการ ในจุดนี้ผู้พัฒนาเล็งเห็นว่าหากเราต้องอ่านความคิดเห็นทั้งหมดด้วยตัวเองนั้นอาจจะต้องใช้เวลาในการอ่านความคิดเห็นแต่ละครั้งเป็นจำนวนมาก ประกอบกับการที่ในปัจจุบันได้มีการนำเทคนิคการวิเคราะห์ข้อมูล (Data Analytic) เข้ามาใช้ในการจัดการจัดกลุ่มข้อความให้เป็นหมวดหมู่ซึ่งหากเราสามารถนำวิธีการนี้มาจัดกลุ่มความคิดเห็นออกเป็นหมวดหมู่ได้ จะสามารถทำให้การพัฒนาการให้บริการของ ทางร้านอาหารเป็นไปได้สะดวกและรวดเร็วยิ่งขึ้น

ผู้พัฒนานำวิธีการวิเคราะห์ข้อมูลมาจัดการแยกแยะข้อความแต่ละข้อความออกเป็นหมวดหมู่ตามความเหมาะสม เช่น Positive และ Negative เพื่อง่ายต่อการนำข้อมูลไปใช้ สำหรับปรับปรุงการให้บริการต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1.เพื่อศึกษาการจำแนกอารมณ์และความรู้สึกของความคิดเห็นต่อร้านอาหาร

2.พัฒนาระบบจำแนกอารมณ์และความรู้สึกของความคิดเห็นบนเฟสบุ๊ค

3.เพื่อให้ได้ข้อมูลความต้องการของผู้บริโภคที่สามารถนำไปใช้ทางการตลาด

4.อำนวยความสะดวกของร้านอาหารในการตัดสินใจปรับปรุงแก้ไขการให้บริการ

1.3 ขอบเขตของโครงงาน

1. โครงการนี้ศึกษาความคิดเห็นของผู้ใช้บริการบนเฟสบุ๊คที่เป็นภาษาไทยมาตรฐาน

2. ความคิดเห็นที่นำมาวิเคราะห์การให้บริการจากหน้าเพจ

3. โปรแกรมสามารถใช้งานได้เมื่อเชื่อมต่ออินเทอร์เน็ตเท่านั้น

4. โปรแกรมสามารถใช้งานบนเบราเซอร์ Google Chorme
5. ผลลัพธ์ของการวิเคราะห์จะถูกแยกแยะตามความคิดเห็นด้านบวกและด้านลบ

1.4 วิธีการดำเนินงาน

1. ศึกษาวิธีและค้นคว้าหาข้อมูลที่เกี่ยวกับโครงงานได้แก่การดึงข้อมูลความคิดเห็นจากเพจบนเฟสบุ๊คการทำเหมืองข้อมูล (Data Mining) และการวิเคราะห์ข้อมูลความพึงพอใจโดยแบ่งเป็น ด้านบวกและด้านลบ
2. ศึกษาข้อมูลของซอฟต์แวร์และเทคนิคการดำเนินงานเทคนิคการใช้ Facebook Graph API เทคนิคที่ใช้ในการวิเคราะห์ข้อมูล ได้แก่ Naive Bayes ผ่านโมดูล NLTK
3. กำหนดขอบเขต ขั้นตอนการดำเนินงาน และออกแบบโปรแกรม
4. พัฒนาโปรแกรมในส่วนการดึงข้อมูลความคิดเห็นจากเพจบนเฟสบุ๊ค
5. เก็บรวบรวมข้อมูลความคิดเห็นจากหน้าเพจบนเฟสบุ๊ค
6. พัฒนาโปรแกรมในส่วนวิเคราะห์ข้อมูล
7. ตรวจสอบความถูกต้องของข้อมูลที่ได้จากการวิเคราะห์และแก้ไขข้อผิดพลาดของโปรแกรม
8. สรุปผลการดำเนินการ และจัดทำเอกสารประกอบโครงการ

ตารางเวลาดำเนินการ

ขั้นตอนการดำเนินการ	ปี พ.ศ. 2561					ปี พ.ศ. 2562			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย
1. ศึกษาวิธีและค้นคว้าข้อมูลที่เกี่ยวกับโครงงาน และรวบรวมข้อมูลความคิดเห็นจากเฟสบุ๊ค									
2. ศึกษาข้อมูลเกี่ยวกับซอฟต์แวร์และเทคนิคการดำเนินงาน									
3. กำหนดขอบเขต ขั้นตอนการดำเนินงาน ออกแบบโปรแกรม									
4. พัฒนาส่วนการดึงข้อมูลความคิดเห็นจากเพจบนเฟสบุ๊ค									
5. รวบรวมข้อมูลความคิดเห็นจากหน้าเพจบนเฟสบุ๊ค									
6. พัฒนาส่วนวิเคราะห์ข้อมูล									
7. ตรวจสอบความถูกต้องของผลวิเคราะห์และแก้ไขข้อผิดพลาด									
8. สรุปผลการดำเนินการและจัดทำเอกสารประกอบ									

ตารางที่ 1.1 ตารางเวลาดำเนินการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ต่อนิสิต

1. ได้เรียนรู้วิธีเขียนโปรแกรมโดยใช้ภาษา Python

2. ฝึกฝนการพัฒนาโปรแกรม

3. ได้เรียนรู้เกี่ยวกับวิธีการจัดการการดำเนินงาน

4. ได้เรียนรู้วิธีการใช้งาน Facebook Graph Api

4. ได้ความรู้เกี่ยวกับการนำความรู้เรื่อง NLP, Data Mining และ Sentiment Analysis มาใช้ร่วมกัน

ประโยชน์ที่ได้ต่อโครงงาน

1. นำข้อมูลที่ได้จากการเก็บไปใช้ในการวางแผนงานในอนาคต

2. สามารถวิเคราะห์เสียงตอบรับของผู้ใช้จากการให้บริการ

3.ข้อมูลที่ได้สามารถนำไปใช้ประโยชน์ในการพัฒนาการให้บริการของร้านอาหารได้

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบจำแนกความคิดเห็นของการรีวิวร้านอาหารบนเฟสบุ๊ค ซึ่งเป็นความรู้พื้นฐานที่ใช้ในการพัฒนาระบบนี้ทั้งหมด

2.1 การวิเคราะห์อารมณ์และความรู้สึก (Sentiment Analysis)

การวิเคราะห์อารมณ์และความรู้สึก [1][2] เป็นการวิเคราะห์จากการทำเหมืองความคิดเห็นโดยอ้างถึง การใช้ระบบประมวลภาษาธรรมชาติ หรือการวิเคราะห์ข้อความเพื่อบ่งบอกถึงความรู้สึกที่มีต่อบางสิ่ง เช่น ความรู้สึกดี (positive) หรือความรู้สึกไม่ดี (negative) ซึ่งการวิเคราะห์อารมณ์และความรู้สึกต้องอาศัยกระบวนการประมวลผลภาษาธรรมชาติ (Natural language processing - NLP) การวิเคราะห์ข้อความ (text analysis)

งานวิจัย [3] เป็นงานวิจัยที่พัฒนาระบบการวิเคราะห์ความรู้สึกจากข้อความบนทวิตเตอร์ โดยเก็บข้อมูลจากผู้ใช้หลักที่เป็นคนไทย (seed) และผู้ติดตามของผู้ใช้หลักที่เป็นคนไทย (followers) โดยจะเก็บเฉพาะข้อความที่เป็นภาษาไทยเท่านั้น ใช้เทคนิค word vector โดยใช้ word2vec ในการแปลงข้อความเป็น word vector และใช้ Dynamic Convolutional Neural Network กับ Long Short Term Memmories ในการจำแนกข้อความเป็น 2 แบบ คือข้อความที่มีความคิดเห็นเชิงบวกและข้อความที่มีความคิดเห็นเชิงลบ อีกทั้งยังมีการแสดงให้เห็นอีกว่าลำดับของคำในประโยคสามารถส่งผลต่อการวิเคราะห์ความคิดเห็นได้

งานวิจัย [4] เป็นงานวิจัยที่พัฒนาระบบการวิเคราะห์อารมณ์ที่ผู้แสดงความคิดเห็นมีต่อวิดีโอบนเว็บไซต์ Youtube โดยเก็บข้อมูลจากวิดีโอภาษาไทยทั้งหมด 50 วิดีโอ แบ่งเป็น วิดีโอโฆษณา 30 วิดีโอ และวิดีโอที่เป็นวิดีโอเพลง (MV) อีก 20 วิดีโอ โดยจะจำแนกข้อความตามอารมณ์ 6 รูปแบบ ได้แก่ อารมณ์โกรธ, กลัว, รังเกียจ, ตกใจ, เศร้า, และมีความสุข โดยใช้ตัวจำแนกนาอีฟเบย์ (Naive Bayes Classifier), ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine), และต้นไม้การตัดสินใจแบบ J48 (J48 decision tree)

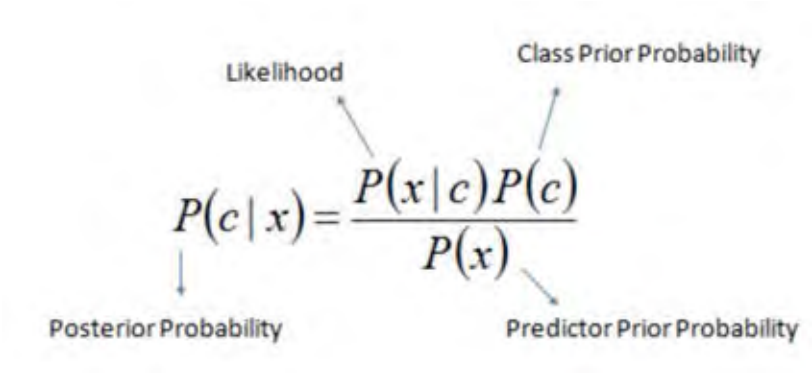
โครงงาน [5] เป็นโครงงานที่พัฒนาระบบการวิเคราะห์ความรู้สึกของความคิดเห็นต่อสายการบินเว็บไซต์ โดยใช้เทคนิคคลังคำศัพท์ (SentiWordNet), ตัวจำแนกนาอีฟเบย์ (Naive Bayes Classifier), และซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) โดยแบ่งความรู้สึกของความคิดเห็นเป็น 2 แบบ คือความคิดเห็นเชิงบวกและความคิดเห็นเชิงลบ โดยทำการวิเคราะห์ความคิดเห็นที่มีต่อสายการบินยอดนิยม 10 สายการบิน และแสดงผลแยกเป็นความคิดเห็นที่มีต่อแต่ละสายการบิน เพื่อช่วยให้ผู้ใช้สามารถเลือกสายการบินที่ตรงกับความต้องการ

โครงงาน [6] เป็นโครงงานที่พัฒนาระบบการวิเคราะห์อารมณ์ความรู้สึกของข้อความบน Twitter เกี่ยวกับสถานที่ท่องเที่ยวในประเทศไทย โดยใช้ตัวจำแนกนาอีพเบย์ (Naive Bayes Classifier), ขั้นตอนวิธีการถดถอยเชิงโลจิสติก (Logistic Regression Algorithm), ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine), และเทคนิค 1vsAll (One-against-all) โดยสามารถวิเคราะห์ความรู้สึกของความคิดเห็นเป็น 2 แบบ คือความคิดเห็นเชิงบวกและความคิดเห็นเชิงลบ สามารถเลือกแสดงผลเป็นความคิดเห็นที่มีต่อสถานที่ท่องเที่ยวจาก 1 ใน 18 แห่งที่ได้รวบรวมข้อมูลไว้ เพื่อช่วยให้ผู้ใช้นำมาประกอบการตัดสินใจในการเลือกสถานที่ท่องเที่ยวได้

โครงงาน [7] เป็นโครงงานที่พัฒนาระบบการวิเคราะห์ความคิดเห็นของลูกค้าของร้านอาหารนานาชาติในกรุงเทพ โดยใช้เทคนิคคลังคำศัพท์ (SentiWordNet) ตัวจำแนกนาอีฟเบย์ (Naive Bayes Classifier) และซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) โดยสามารถแบ่งความรู้สึกชองความคิดเห็นเป็น 2 แบบ คือความคิดเห็นเชิงบวกและความคิดเห็นเชิงลบ โดยมีการใช้ Natural Language Toolkit ในการกำกับหน้าที่ของคำ (part of speech tagging) มีการแบ่งความคิดเห็นออกเป็น 4 ด้าน คือด้านอาหาร, ด้านบรรยากาศ, ด้านบริการ, และด้านความคุ้มค่า เพื่อช่วยให้ผู้สนใจสามารถเลือกร้านอาหารที่ตรงกับความต้องการได้

2.2 เครื่องมือการจำแนกข้อความนาอีฟเบย์ (Naive Bayes classifiers)

นาอีฟเบย์ เป็นการใช้แบบจำลอง (Model) ในการจำแนกข้อมูลออกเป็นกลุ่ม คือแบบจำลองการจำแนกข้อมูลวิธีหนึ่งโดยที่หลักการของวิธีการนี้จะใช้การคำนวณความน่าจะเป็น ซึ่งอยู่บนพื้นฐานจากทฤษฎีของเบย์ (Bayes' Theorem) เพื่อหาว่าสมมติฐานใดน่าจะถูกต้องที่สุด และมีสมมติฐานให้โอกาสเกิดเหตุการณ์ต่าง ๆ เป็นอิสระต่อกัน (Independence) โดยสมการคำนวณแสดงดังภาพ 2.1



ภาพที่ 2.1 สมการความน่าจะเป็นของเบย์

ที่มา http://uc-r.github.io/naive_bayes

$P(c|x)$ คือค่าความน่าจะเป็นที่ข้อมูลที่มีคุณลักษณะ x จะอยู่ในกลุ่ม c

$P(x|c)$ คือค่าความน่าจะเป็นที่ข้อมูลที่อยู่ในกลุ่ม c จะมีคุณลักษณะ x

โดยที่ $x = x1 \cap x2 \ldots \cap xn$ โดยที่ n คือจำนวนคุณลักษณะในข้อมูลที่สนใจ

$P(c)$ คือค่าความน่าจะเป็นที่เหตุการณ์ c เกิดขึ้น

$P(x)$ คือค่าความน่าจะเป็นของการเกิดเหตุการณ์ x

แต่การที่คุณลักษณะ $x = x1 \cap x2 \ldots \cap xn$ ที่เกิดขึ้นในข้อมูลที่สนใจอาจจะมีจำนวนน้อยมากหรือไม่มีรูปแบบของคุณลักษณะแบบนี้เกิดขึ้นเลย ดังนั้นจึงได้ใช้หลักการที่ว่าแต่ละคุณลักษณะเป็นอิสระต่อกันทำให้สามารถเปลี่ยนสมการ $P(x|c)$ ได้เป็นดังภาพ 2.2

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

ภาพที่ 2.2 สมการความน่าจะเป็นสำหรับเครื่องมือการจำแนกข้อความนาอีฟเบย์

ที่มา http://uc-r.github.io/naive_bayes

$P(x1|c)$ คือความน่าจะเป็นของคุณลักษณะหรือเหตุการณ์ที่ 1 เมื่อรู้ว่า x อยู่ในคลาส c

$P(x2|c)$ คือความน่าจะเป็นของคุณลักษณะหรือเหตุการณ์ที่ 2 เมื่อรู้ว่า x อยู่ในคลาส c

$P(xn|c)$ คือความน่าจะเป็นของคุณลักษณะหรือเหตุการณ์ที่ n เมื่อรู้ว่า x อยู่ในคลาส c

ตัวอย่าง Naive bayes classifier

สมมติว่ามีข้อมูล “ไปลอง ห้ามพลาด เชื่อสิ หมดไปเท่าไหร่กับบุฟ” จะแบ่งได้เป็น ['ไป', 'ลอง', 'ห้าม', 'พลาด', 'เชื่อ', 'สิ', 'หมด', 'ไป', 'เท่า', 'ไหร่', 'กับ', 'บุฟ'] แบ่งได้ทั้งหมด 12 คำ แบ่งเป็นเชิงบวก 9 คำ เชิงลบ 3 คำ คำเชิงบวกที่ระบบเคยเห็นจากข้อมูลที่ใช้ฝึก 7 คำ คำเชิงลบที่ระบบเคยเห็นจากข้อมูลที่ใช้ฝึก 2 คำ คำเชิงบวกที่ระบบไม่เคยเห็น 2 คำ คำเชิงลบที่ระบบไม่เคยเห็น 1 คำ

โดย c คือ class (เป็นข้อความเชิงบวกหรือลบ)

x คือ attribute (มีในระบบจากข้อมูลที่ใช้ฝึก)

$P()$ = ความน่าจะเป็น

ต้องการที่หาความน่าจะเป็นของอารมณ์ความรู้สึกที่เป็น Positive จากคำที่มีในระบบจากข้อมูลที่มี $\mathrm{P(x|c)} = 7/9$ (คือจำนวนข้อความเชิงบวกที่ตรงกับข้อความในระบบส่วนด้วยข้อความเชิงบวกทั้งหมด)

$\mathrm{P(c)} = 9/12$ (คือ จำนวนข้อความเชิงบวกทั้งหมดจากข้อความทั้งหมด)

$\mathrm{P(x)} = 9/12$ (คือ จำนวนข้อความที่เคยเจอจากข้อความทั้งหมด)

จากสูตร เราจะได้ว่า $\mathrm{P(c|x)}$ จะได้ค่าเท่ากับ 0.78

ดังนั้น เราจะได้ว่าข้อมูลที่มีในระบบ จะมีโอกาสเป็นเชิงบวก 0.78 และจะเป็นเชิงลบ 0.22 สรุปผลเมื่อเราวิเคราะห์อารมณ์ความรู้สึกออกมาแล้วจะได้ว่าประโยคนี้ให้ค่าน้ำหนักเป็นเชิงบวก

สำหรับส่วนของโปรแกรมที่ใช้คือ

```
import nltk
classifier = nltk.NaiveBayesClassifier.train(train_features)
```

ทำการเรียกใช้ไลบรารี่ NLTK เพื่อสร้างตัวจำแนกแบบนาอีฟเบย์

แล้วทำการฝึกตัวจำแนกด้วยข้อมูลที่เราเตรียมไว้

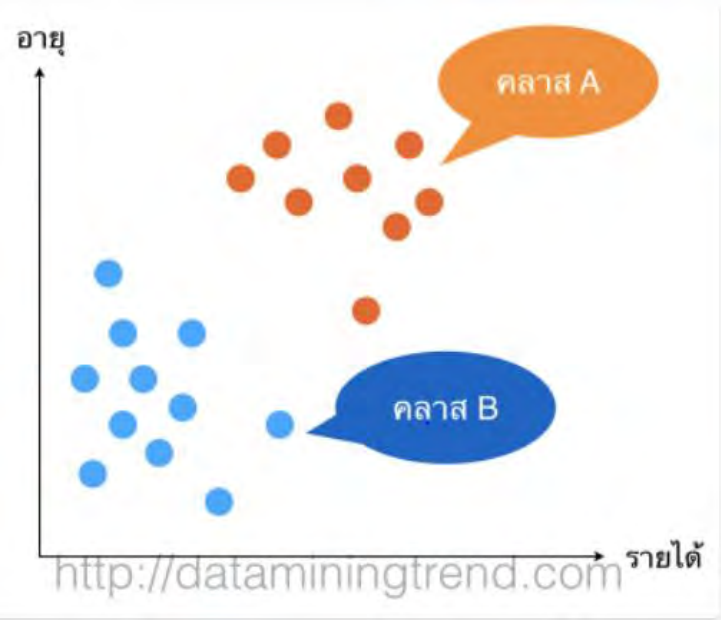
```
observed = classifier.classify(feats)
```

เป็นการให้ตัวจำแนกทำนายว่าความคิดเห็นเป็นความคิดเห็นไปในแง่ใด

2.3 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine)

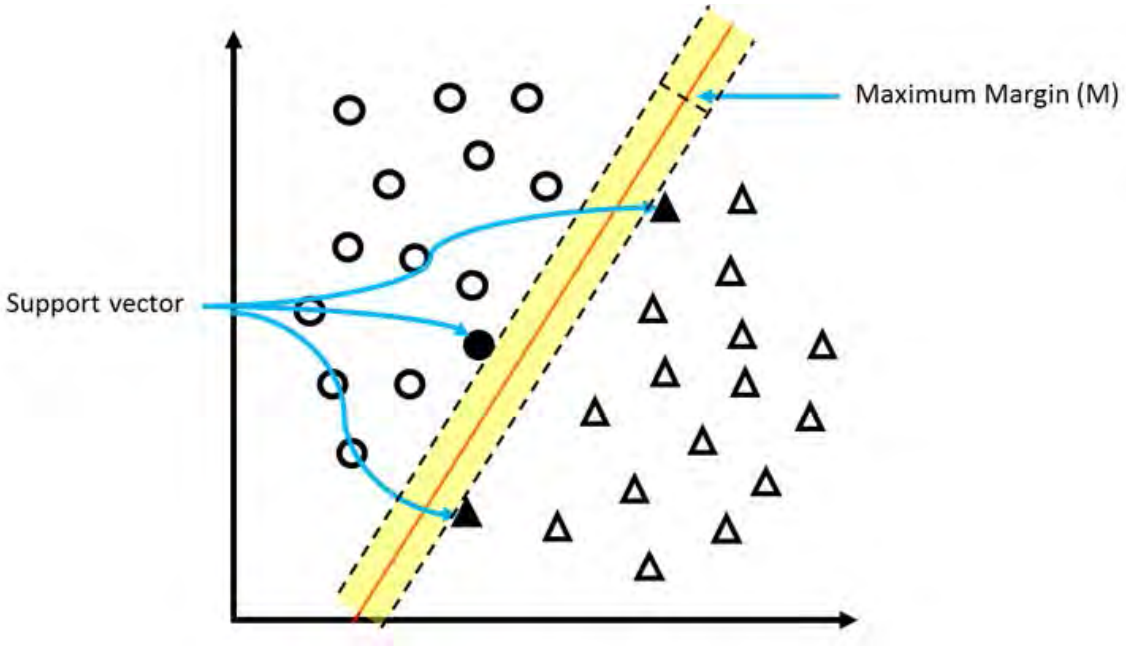
ซัพพอร์ตเวกเตอร์แมชชีน คือเทคนิครูปแบบหนึ่งในการจำแนกข้อมูลบนปริภูมิคุณลักษณะ (Features space) เป็นสองกลุ่มด้วยระนาบ โดยระนาบนี้จะมีระนาบคู่ขนานอยู่ทั้งสองข้าง โดยที่ระนาบคู่ขนานนี้จะสัมผัสกับข้อมูลที่อยู่ใกล้ระนาบหลักมากที่สุด (support vector) และมีระยะห่าง (margin) จากระนาบหลักมากที่สุด (maximum margin)

ตัวอย่างเช่นถ้าเรามีข้อมูลที่เราได้ทำการ training มาแล้วแบ่งเป็น 2 กลุ่มตามภาพที่ 2.3 เราจะใช้เลือกเส้น linear model ที่ไม่จำรูปแบบของข้อมูลที่ใช้ฝึกมากเกินไป (Over fitting) โดยที่ linear model ที่เลือกจะมีระยะห่างระหว่าง 2 คลาสห่างกันมากที่สุด ตามภาพ 2.4



ภาพที่ 2.3 ตัวอย่างข้อมูลที่ทำการ training แล้ว

ที่มา http://dataminingtrend.com/2014/support-vector-machine-svm/?fbclid=IwAR28s37oo4LPlaAt1wo3hWIXjEh7aInJJW9cm_d5FWc7UbSbYCyS_rd9VMM



ภาพที่ 2.4 ตัวอย่างการจำแนกด้วยซัพพอร์ตเวกเตอร์แมชชีน

ที่มา: https://knowledge.snru.ac.th/ซัพพอร์ตเวกเตอร์แมชชีน

สำหรับส่วนของโปรแกรมที่ใช้คือ

```
from sklearn import svm


svm = svm.LinearSVC()
svm.fit(onehot_.transform(X_train), y_train)
```

เป็นการเรียกใช้ไลบรารี่ Scikit-learn เพื่อสร้างตัวจำแนกแบบซัพพอร์ตเวกเตอร์แมชชีนแล้วทำการปรับตัวจำแนกให้เข้ากับข้อมูลที่เราเตรียมไว้

```
svm.predict(onehot_.transform(testdata_tokens))
```

เป็นการให้ตัวจำแนกทำนายว่าความคิดเห็นเป็นความคิดเห็นไปในแง่ใด

2.4 โครงข่ายประสาทเทียม (Neural Network)

Neural Network หรือ โครงข่ายประสาทเทียม คือการที่เราเอาเทคนิคการพยากรณ์ด้วยสมการเชิงเส้น เช่น linear regression มาใช้ต่อกับหลายๆ ตัว และหลายๆชั้น ทำให้การ Prediction

ที่จากเดิมอ้างอิงจากเส้นตรงเส้นเดียว กลายร่างเป็นเส้นตรงหลายๆ เส้นที่ช่วยให้ Prediction ทำได้อย่างละเอียดมากขึ้นกว่าเดิม และแม่นยำกว่าเดิม โดยที่ระบบจะทำการเปลี่ยนค่า weight กับ bias จนให้ได้ผลลัพธ์ที่ดีที่สุด ตามภาพ 2.5 ซึ่ง Neural Network มีส่วนประกอบคือ Input layer, Hidden layer และ Output layer

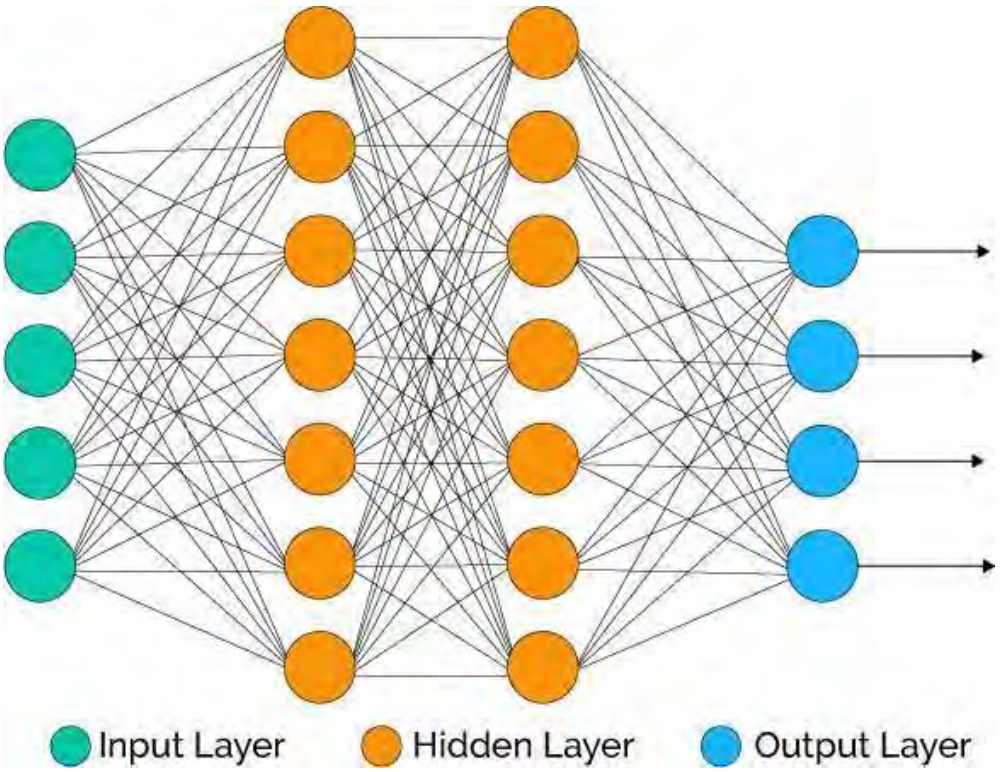
สมการ linear regression: predicted_Y = w * X + bias
เมื่อ X คือข้อมูลเข้า, w คือน้ำหนักของข้อมูลประเภทนั้น, bias คือ ค่าคงที่ค่าหนึ่ง

หน้าที่ของส่วนประกอบแต่ละส่วนของ Neural Network

1. Input Layer เป็นข้อมูล input จำนวนของโหนดจะขึ้นอยู่กับจำนวนของ input ว่ามีข้อมูลอะไรบ้างที่นำเข้ามาคิดในโมเดล เช่น ถ้าข้อมูลของลูกค้าเป็น input ซึ่งประกอบด้วย อายุ เพศ จังหวัดที่อาศัย รวมทั้งสิ้น 4 อย่าง input layer ก็จะมี 4 โหนด

2. Hidden Layer มีกี่ชั้นก็ได้ แล้วแต่ผู้สร้าง แล้วแต่ละชั้นจะมีจำนวนของ Neuron เท่าไหร่ก็ได้เช่นกัน ซึ่งการเพิ่มชั้นและจำนวน neuron ก็จะส่งผลต่อการทำงานของโมเดล ในส่วนของ hidden layer มีการทำงานเปรียบเสมือนส่วนที่เรียนรู้ข้อมูลเชิงลึก หรือ deep learning โดยสิ่งสำคัญใน hidden layer อีกอย่างหนึ่งคือ ทุกๆ โหนดต้องประกอบด้วย nonlinearity function

3. Output Layer จะนำเอาข้อมูลจากการคำนวณไปใช้ จำนวนของโหนดในชั้นนี้ ขึ้นอยู่กับรูปแบบของ output ที่เราจะเอาไปใช้ ซึ่งถ้ายกตัวอย่างง่ายๆ เช่น ถ้างานที่ทำเป็น Regression ก็กำหนดให้ output layer เป็นแบบ 1 โหนด เพราะต้องการคำตอบค่าเดียว ถ้าเป็นหลายค่าก็เพิ่มไปตามที่ต้องการ เช่น ในบางงานเราอาจจะ predict หาตำแหน่งของภาพในแกน x และ y พร้อมๆ กัน ในกรณีนี้เราจะใส่ output layer เป็น 2 โหนด เป็นต้น



ภาพที่ 2.5 Neural Network

ที่มา https://towardsdatascience.com/machine-learning-fundamentals-ii-neural-networks-f1e7b2cb3eef

สำหรับส่วนของโปรแกรมที่ใช้คือ

```
import tensorflow as tf

tf.reset_default_graph()

vocab_len = len(onehot_nn.classes_)
inputs_ = tf.placeholder(dtype=tf.float32, shape=[None, vocab_len],
name="inputs")
targets_ = tf.placeholder(dtype=tf.float32, shape=[None, 2], name="targets")

h1 = tf.layers.dense(inputs_, 500, activation=tf.nn.relu)
logits = tf.layers.dense(h1, 2, activation=tf.nn.relu)
```

```
output = tf.nn.sigmoid(logits)

loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=logits,
labels=targets_))

optimizer = tf.train.AdamOptimizer(0.001).minimize(loss)

correct_pred = tf.equal(tf.argmax(logits, 1), tf.argmax(targets_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32), name='accuracy')
sess = tf.Session()

# Initializing the variables
sess.run(tf.global_variables_initializer())
```

เป็นการเรียกใช้ไลบรารี่ Scikit-learn เพื่อสร้างตัวจำแนกแบบโครงข่ายประสาทเทียม และทำการปรับค่าของตัวจำแนก

```
epochs = 10
batch_size = 3000

for epoch in range(epochs):
    for X_batch, y_batch in get_batch(onehot_nn.transform(nnx_train),
label2bool(nny_train), batch_size):
        loss_value, _ = sess.run([loss, optimizer], feed_dict={
            inputs_: X_batch,
            targets_: y_batch
        })
        print("Epoch: {} \t Training loss: {}".format(epoch, loss_value))

    acc = sess.run(accuracy, feed_dict={
```

```
        inputs_: onehot_nn.transform(nnx_valid),
        targets_: label2bool(nny_valid)
    })

    print("Epoch: {} \t Validation Accuracy: {}".format(epoch, acc))
```

เป็นการฝึกตัวจำแนกด้วยข้อมูลที่เตรียมไว้

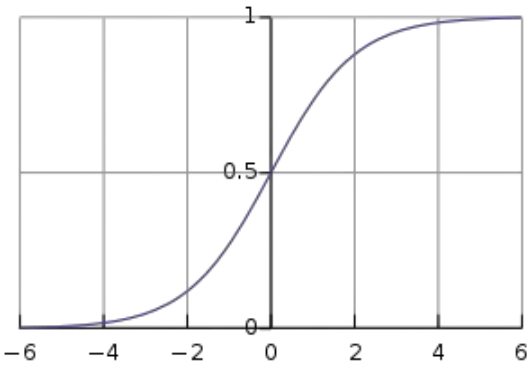
```
test_result = sess.run(output,feed_dict={
    inputs_:onehot_nn.transform(X_test)
})
i=0
nn_pred = []
for stuff in test_result:
    if test_result[i][0]>test_result[i][1]:
        nn_pred.append('pos')
    else:
        nn_pred.append('neg')
    i=i+1
```

เป็นการให้ตัวจำแนกทำนายว่าความคิดเห็นเป็นความคิดเห็นไปในแง่ใด

2.5 การถดถอยโลจิสติค (Logistic Regression)

คือการทำ Classification โดย output แล้วออกมาเป็นจำนวนตัวเลข(Regression) จากนั้นนำไป mapping กับ Sigmoid function ก็จะได้ว่าข้อมูลอันนี้เป็นคลาส หรือ ไม่ได้เป็นคลาส

Output ของ Logistic Regression เป็นจำนวนตัวเลขแต่เมื่อนำมาใช้งานร่วมกับ Sigmoid หน้าตาตามภาพ 2.6

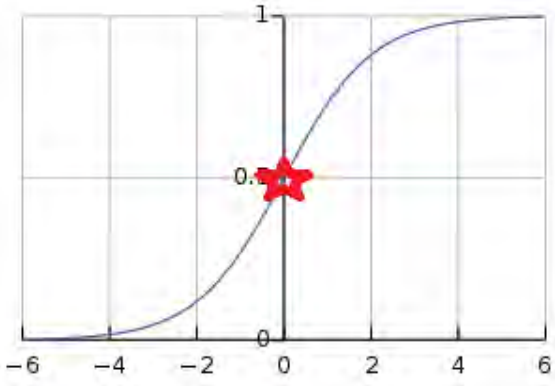


ภาพที่ 2.6 Logistic Regression

แกน x เป็น output ของ Regression แกน y เป็นความน่าจะเป็น

ที่มา https://medium.com/mmp-li/logistic-regression-ไม่มีอะไรเป็นไปตามอย่างที่คิดเสมอ-machine-learning-101-bba2f666234d

สมมุติผลลัพธ์จริง ๆ ของเราทายออกมาได้ 0 บนแกน Y คือ ความน่าจะเป็นที่จะเป็น Class นั้นๆ อยู่ในช่วง 0–1 บนแกน X คือ ค่า Regression ที่ทายออกมาได้ เราเอาผลลัพธ์มาเทียบกับกราฟในรูปก็จะได้แกน Y เป็น 0.5 ตามภาพ 2.7



ภาพที่ 2.7 ภาพแสดงค่า Regression ที่ทายออกมาได้แกน Y เป็น 0.5

ที่มา https://medium.com/mmp-li/logistic-regression-ไม่มีอะไรเป็นไปตามอย่างที่คิดเสมอ-machine-learning-101-bba2f666234d

ถ้าความน่าจะเป็นอยู่ในช่วง 0.0–0.49 ก็จะได้ว่าไม่เป็นคลาสนั้นๆ

ถ้าความน่าจะเป็นอยู่ในช่วง 0.49–1.00 ก็จะได้ว่าเป็นคลาสนั้นๆ เป็นต้น

โดยความน่าจะเป็นค่าจะต้องไม่เกิน 1 โดย Logistic Regression สามารถให้ output ออกมาได้ว่าเป็นคลาสนี้ หรือ ไม่เป็นคลาสนี้โดยการใช้ Sigmoid function และสามารถให้ output ออกเป็น “ความน่าจะเป็น” ของคลาสนั้นๆได้อีกด้วย

สมการ logistic regression: $\rho(X) = \frac{1}{1+e^{-(a+\Sigma_i^n B_i X_i)}}$

เมื่อ $\rho(X)$ คือ ค่าความน่าจะเป็นที่จะเกิดเหตุการณ์ที่เราสนใจ, a คือ ค่าคงที่ค่าหนึ่ง, X คือ ค่า Input ของข้อมูลที่มีผลต่อ $\rho(X)$, B คือ น้ำหนักของแต่ละปัจจัยที่มีผลต่อ $\rho(X)$, i คือ ตัวเลขระบุว่าค่าของ B และ X เป็นค่า B ของตัวที่ i และเป็นค่า X ของตัวที่ i

โดยส่วนของโปรแกรมที่นำมาใช้มี

```
from sklearn.linear_model import LogisticRegression
log_model = LogisticRegression()
log_model = log_model.fit(X=lrx_train, y=lry_train)
```

เป็นการเรียกใช้ไลบรารี่ Scikit-learn เพื่อสร้างตัวจำแนกแบบการถดถอยโลจิสติคแล้วทำการปรับตัวจำแนกให้เข้ากับข้อมูลที่เราเตรียมไว้

```
lry_pred = log_model.predict(lrx_test)
```

เป็นการให้ตัวจำแนกทำนายว่าความคิดเห็นเป็นความคิดเห็นไปในแง่ใด

2.6 Scikit-learn (Sklearn)

Scikit-learn เป็นไลบรารี่ในภาษา Python ที่ใช้สำหรับการทำ ฟmachine learning โดยมีขั้นตอนวิธีต่างๆ สำหรับการทำ machine learning ให้เลือกใช้อย่างหลากหลาย เช่น Naive Bayes Classifier, Support Vector Machine, Neural Network, Logistic Regression

ตัวอย่างคำสั่งใน Sklearn

1. การใช้ไลบรารี่ sklearn เพื่อเรียกใช้ SVM

```
from sklearn.svm import SVC

svm = SVC(kernel='rbf', random_state=0, gamma=.10, C=1.0)
svm.fit(X_train_std, y_train)
```

ภาพที่ 2.8 การใช้ไลบรารี่ sklearn เพื่อเรียกใช้ SVM

ที่มา https://medium.com/@pichitchai.pim/รวมให้ครบแล้วเริ่มเขียน-classification-model-ยอดนิยม-k-mean-clustering-ด้วย-python-6d10bb58d486

2. การใช้ไลบรารี่ sklearn เพื่อเรียกใช้ naive bayes

```
from sklearn.naive_bayes import GaussianNB

validation_size = 0.33

seed = 7
X_train, X_validation, Y_train, Y_validation = train_test_split(XX,
yy, test size=validation size, random state=seed)
```

ภาพที่ 2.9 ใช้ไลบรารี่ sklearn เพื่อเรียกใช้ naive bayes

ที่มา https://medium.com/@pichitchai.pim/รวมให้ครบแล้วเริ่มเขียน-classification-model-ยอดนิยม-k-mean-clustering-ด้วย-python-6d10bb58d486

2.7 Natural Language Toolkit (NLTK)

Natural Language Toolkit เป็นไลบรารี่ในภาษา Python ที่ใช้สำหรับการประมวลผลภาษาธรรมชาติ โดยมีคลังข้อมูลภาษามากกว่า 50 คลัง และสามารถจัดแบ่งประเภท, ตัดแบ่งคำ, การกำกับชนิดของคำ, การจัดรูปประโยค, และ การให้เหตุผลทางความหมาย

ตัวอย่างคำสั่งใน NLTK

1. การตัดคำและติดป้ายกำกับชนิดขอคำ

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

ภาพที่ 2.10 ตัดคำและติดป้ายกำกับชนิดของคำ

ที่มา https://www.nltk.org/

2. การระบุประเภทของเอนติตี้

```
>>> entities = nltk.chunk.ne_chunk(tagged)
>>> entities
Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'),
        ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'),
      Tree('PERSON', [('Arthur', 'NNP')]),
        ('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'),
        ('very', 'RB'), ('good', 'JJ'), ('.', '.')])
```

ภาพที่ 2.11 ระบุประเภทของเอนติตี้

ที่มา https://www.nltk.org/

2.8 Deepcut

Deepcut เป็นไลบรารี่ในภาษา Python ที่ใช้สำหรับการตัดแบ่งคำโดยใช้ Deep Neural Network

การเรียกใช้ไลบรารี่ deepcut และคำสั่งตัดคำ

```
import deepcut
list_word = deepcut.tokenize('ตัดคำได้ดีมาก')
print(list_word)
```

ภาพที่ 2.12 การเรียกใช้ deepcut และคำสั่งตัดคำ

ที่มา https://www.patanasongsivilai.com/blog/tudkumthai/

2.9 Facebook SDK for Python

Facebook SDK for Python เป็นไลบรารี่ในภาษา Python ที่ใช้สำหรับเข้าถึงข้อมูลบนเฟสบุ๊ค โดยเราจะต้องติดตั้งโมดูล facebook-sdk จากเว็บแล้วนำไปติดตั้งบนเครื่องโดยใช้คำสั่ง python setup.py install

ตัวอย่างคำสั่งที่ใช้ใน Facebook SDK for Python

การเรียกใช้ไลบรารี่ Facebook SDK และคำสั่งในการเข้าถึงข้อมูลบนเฟสบุ๊ค

```
import facebook
graph = facebook.GraphAPI(oauth_access_token) #เอาได้ที่ https://developers.facebook.com/tools/access_token/
profile = graph.get_object("me")
friends = graph.get_connections("me", "friends")
graph.put_object("me", "feed", message="I am writing on my wall!")
```

ภาพที่ 2.13 การเรียกใช้ไลบรารี่ Facebook SDK และคำสั่งในการเข้าถึงข้อมูลบนเฟสบุ๊ค

ที่มา https://python3.wannaphong.com/2015/01/facebook-api-python-facebook-sdk.html

2.10 Tensorflow

Tensorflow คือไลบรารี่สำหรับการเขียนโปรแกรม Dataflow ที่ใช้ในการพัฒนา Machine learning ได้รับการพัฒนาโดยบริษัท Google รองรับการเขียนบน Python ไม่ว่าจะเป็น Python2 หรือ Python3

การเรียกใช้ไลบรารี่ Tensorflow

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
```

ภาพที่ 2.14 การเรียกใช้ไลบรารี่ Tensorflow

ที่มา https://medium.com/@ariyakluankloi/tensorflow-part-1-ลองมาเล่น-tensorflow-กันสักหน่อย-6f8d9b3a577d

บทที่ 3

การรวบรวมข้อมูลและการวิเคราะห์ข้อมูล

ในบทนี้จะกล่าวถึงวิธีการรวบรวมข้อความแสดงความคิดเห็นและการวิเคราะห์ข้อมูลเพื่อจำแนกข้อความแสดงความคิดเห็น ดังรายละเอียดต่อไปนี้

3.1 การรวบรวมข้อมูล

ข้อมูลต่างๆ ที่ผู้พัฒนาใช้ในการวิเคราะห์ ออกแบบ และพัฒนาระบบ มีดังต่อไปนี้

3.1.1 ข้อความแสดงความคิดเห็นต่อร้านอาหารจากช่องทางแสดงความคิดเห็นของเพจบนเฟสบุ๊ค

ผู้พัฒนาได้ทำการรวบรวมข้อความแสดงความคิดเห็นจากช่องทางแสดงความคิดเห็นของเพจบนเฟสบุ๊คด้วยไลบรารี่ Facebook SDK for Python แต่เนื่องจากการรวบรวมความคิดเห็นจากเฟสบุ๊คนั้น ในขณะนี้ทางผู้พัฒนาทำได้เฉพาะเพจที่ตนเองเป็นเจ้าของและการที่มีข้อความแสดงความคิดเห็นเฉพาะแค่บางเพจทำให้มีข้อมูลไม่เพียงพอที่จะนำมาใช้พัฒนาระบบได้ ทางผู้พัฒนาจึงได้จัดหาข้อมูลความคิดเห็นจากแหล่งอื่น

3.1.2 ข้อความแสดงความคิดเห็นต่อร้านอาหารจากเว็บไซต์ Wongnai.com

ผู้พัฒนาได้สืบค้นหาความคิดเห็นเกี่ยวกับร้านอาหารที่เป็นภาษาไทยบนเว็บไซต์ต่าง ๆ และได้พบกับข้อมูลการแสดงความคิดเห็นเกี่ยวกับร้านอาหารที่มาจากเว็บไซต์ www.wongnai.com ที่ทาง Wongnai ได้ทำการเผยแพร่ [18] ในรูปแบบของเอกสารที่มีชนิดเป็น csv โดยผู้พัฒนาได้สุ่มเลือกความคิดเห็นมาทั้งหมด 300 ข้อความ ซึ่งหลังจากคัดแยกข้อมูล แล้วนับได้ 999 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 555 ประโยค และความคิดเห็นเชิงลบ 444 ประโยค

3.2 การวิเคราะห์ข้อมูล

หลังจากได้ข้อมูลมาจาก wongnai แล้ว ทางผู้พัฒนาได้จัดระเบียบข้อมูลโดยแบ่งข้อมูลในแต่ละความคิดเห็นออกเป็นประโยค โดยแต่ละประโยคอยู่ในบรรทัดเดียวกันและได้ทำการลบส่วนที่เป็น

ลิงค์, สัญลักษณ์, เครื่องหมายวรรคตอน หลังจากนั้นได้ทำการจำแนกข้อมูลโดยแยกว่าในแต่ละประโยคนั้นแสดงความคิดเห็นเชิงบวกหรือความคิดเห็นเชิงลบ สำหรับประโยคที่ไม่แสดงความคิดเห็นจะถูกคัดออก

ตัวอย่างความคิดเห็น เช่น “ก๋วยเตี๋ยวไก่อร่อยดีแต่ถ้วยเล็กไปหน่อย ส้มตำอร่อยครบรส มีมะนาวฝานแบบโบราณ น่าประทับใจ ไก่ทอดรสชาตพอใช้ออกไปทางเค็ม สเตะเนื้อนุ่มแต่ขนมปังปิ้งที่เสริฟมาด้วยกันไม่กรอบ หอยทอดไม่สดเหม็นจนทานไม่ได้ ทำลายความประทับใจในการกินไปอย่างน่าเสียดาย”

ตัวอย่างประโยคที่แสดงความคิดเห็นเชิงบวก เช่น “ส้มตำอร่อยครบรส มีมะนาวฝานแบบโบราณ น่าประทับใจ”

ตัวอย่างประโยคที่แสดงความคิดเห็นเชิงลบ เช่น “หอยทอดไม่สดเหม็นจนทานไม่ได้ ทำลายความประทับใจในการกินไปอย่างน่าเสียดาย”

3.3 ขั้นตอนการวิเคราะห์ความคิดเห็น

1. นำข้อมูลเข้าจากไฟล์รูปแบบ csv ไปเป็นไฟล์รูปแบบ txt
2. แบ่งความคิดเห็นออกเป็นประโยค แก้คำที่สะกดผิด ลบเครื่องหมายและสัญลักษณ์ต่าง ๆ
3. แยกประโยคที่แสดงความคิดเห็นในแง่บวกและแง่ลบไว้คนละไฟล์
4. นำไฟล์ที่รวบรวมความคิดเห็นแง่บวก และไฟล์ที่รวบรวมความคิดเห็นแง่ลบเข้าสู่ระบบ

3.4 การประเมิณผลเพื่อเลือกวิธีที่ใช้ในการจำแนกความคิดเห็น

ในการทดลองนี้ ได้ใช้วิธีการในการคัดแยกข้อมูลทั้งหมด 5 วิธีการ โดยจะประเมิณผลการทดลองการจำแนกความคิดเห็นด้วยเกณฑ์ต่างๆ ดังนี้

3.4.1 คอนฟิวชันเมทริกซ์ (Confusion matrix)

คอนฟิวชันเมทริกซ์ คือ ตารางแสดงผลว่าข้อมูลที่ระบบทำนายนั้นจริงๆ แล้วเป็นข้อมูลประเภทใด (actual class) และระบบทำนายว่าเป็นประเภทใด (predicted class)

ตารางที่ 3.1 คอนฟิวชันเมทริกซ์

		predicted	
		positive	negative
actual	positive	True Positive	False Negative
	negative	False Positive	True Negative

โดยค่าในตารางมีความหมายดังนี้

1) True Positive (TP) คือ ข้อมูลที่เป็น positive และระบบทำนายว่าเป็น positive
2) False Negative (FN) คือ ข้อมูลที่เป็น positive แต่ระบบทำนายว่าเป็น negative
3) False Positive (FP) คือ ข้อมูลที่เป็น negative แต่ระบบทำนายว่าเป็น positive
4) True Negative (TN) คือ ข้อมูลที่เป็น negative และระบบทำนายว่าเป็น negative

3.4.2 ค่าความถูกต้อง (Accuracy)

ค่าความถูกต้องของการจำแนกความคิดเห็น คือ เปอร์เซนต์ของจำนวนข้อความที่ระบบทายได้ถูกต้อง $(TP + TN)$ ต่อจำนวนข้อความทั้งหมด $(TP + FN + FP + TN)$ คำนวณได้จาก $\frac{TP+TN}{TP+FN+FP+TN}$

3.4.3 ค่าความแม่นยำ (Precision)

ค่าความแม่นยำ คือ เปอร์เซนต์ของจำนวนข้อความที่ระบบทายถูกต้องว่าเป็นประเภทนั้น $(TP$ หรือ $TN)$ ต่อจำนวนข้อความที่ระบบทายว่าเป็นประเภทนั้นทั้งหมด $(TP + FP$ หรือ $TN + FN)$

3.4.4 ค่าการเรียกคืน (Recall)

ค่าการเรียกคืน คือ เปอร์เซนต์ของจำนวนข้อความที่ระบบทายถูกต้องว่าเป็นประเภทนั้น (TPหรือTN) ต่อจำนวนข้อความที่เป็นประเภทนั้นทั้งหมด ($TP + FN$หรือ $TN + FP$)

3.4.5 ค่าวัดประสิทธิภาพโดยรวม (F-measure)

ค่าวัดประสิทธิภาพโดยรวม คือ

การวัดประสิทธิภาพโดยรวมของทั้งค่าความแม่นยำและค่าการเรียกคืน คำนวณได้จาก

$$\frac{2 \times Recall \times Precision}{Recall + Precision}$$

บทที่ 4

การออกแบบและพัฒนาระบบ

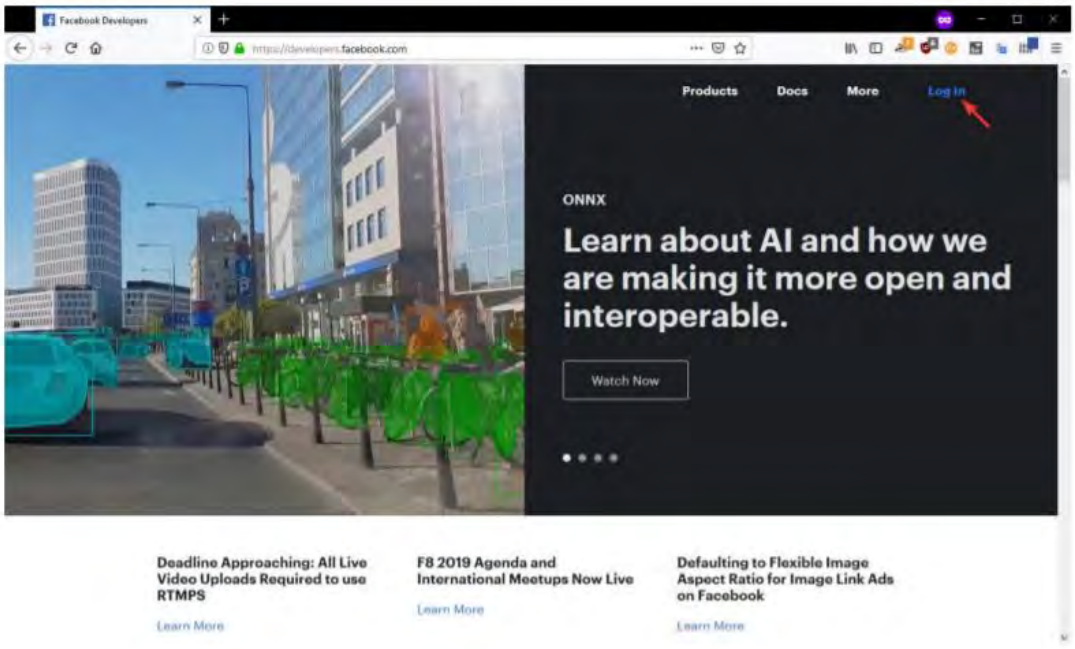
ในบทนี้จะกล่าวถึงการออกแบบและการพัฒนาระบบจำแนกความรู้สึกของความคิดเห็นที่มีต่อร้านอาหารบนเฟสบุ๊ค โดยจะกล่าวถึงการออกแบบระบบทั้งหมด 4 หัวข้อ ได้แก่ การออกแบบภาพรวมการทำงานของระบบ แผนภาพขั้นตอนการจำแนกความคิดเห็นของระบบ แผนภาพยูสเคส (Use Case diagram) ภาษาและโปรแกรมที่ใช้ในการพัฒนาระบบ ดังต่อไปนี้

4.1 การออกแบบการทำงานของระบบ

ในส่วนของการออกแบบการทำงานของระบบ จะอธิบายการทำงานของโปรแกรมตั้งแต่ส่วนของโปรแกรมที่ใช้ในการดึงข้อความแสดงความคิดเห็นจากเฟสบุ๊คไปจนถึงส่วนของโปรแกรมที่ใช้ในการวิเคราะห์หาความรู้สึกจากข้อความแสดงความคิดเห็นที่ดึงมา

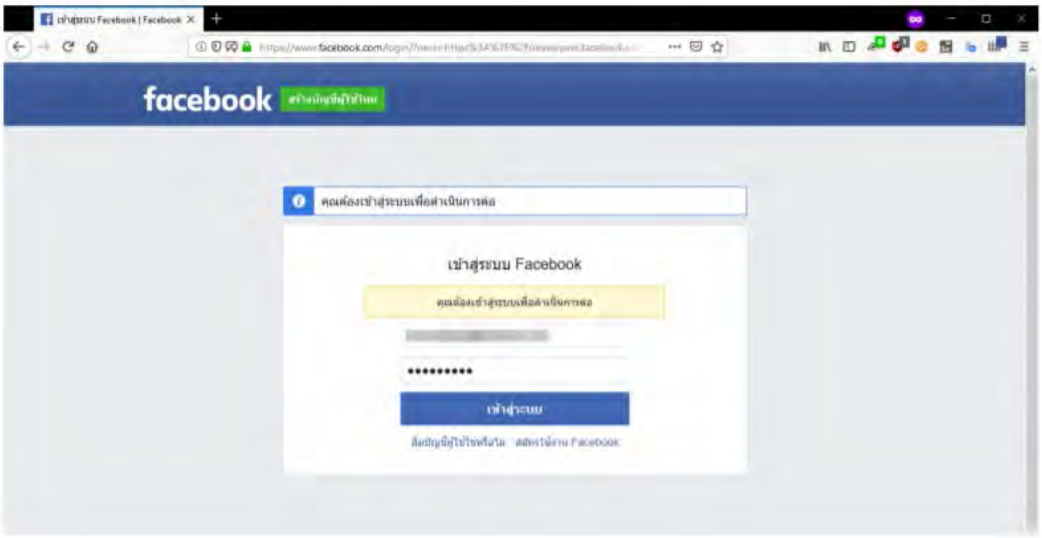
การจะดึงข้อมูลจากเพจบนเฟสบุ๊คนั้น เราจำเป็นที่จะต้องใช้ token โดยเราสามารถได้รับ token ตามขั้นตอนต่อไปนี้

1. ไปที่ developer.facebook.com แล้วกดปุ่ม Log in ด้านขวาบน



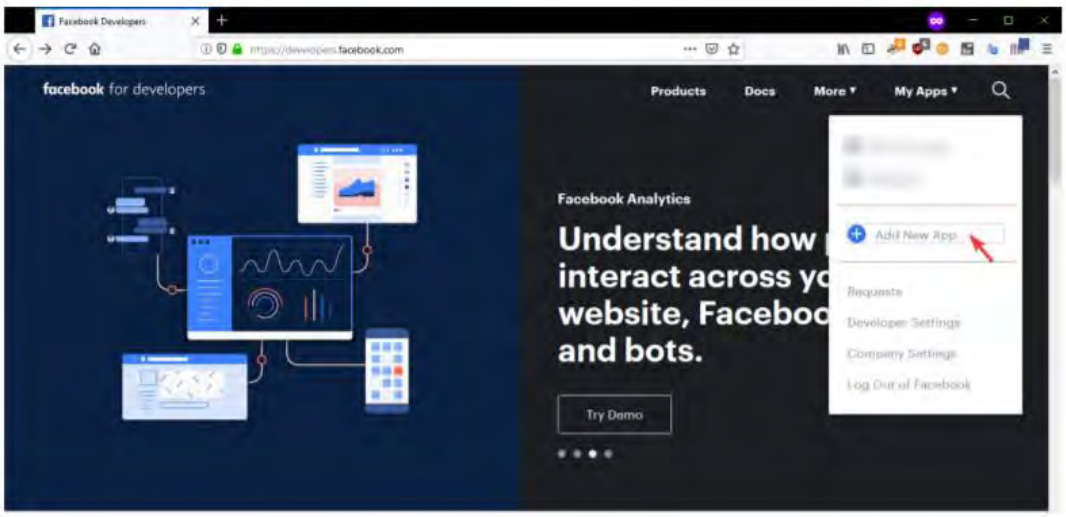
ภาพที่ 4.1 หน้าหลักของ Facebook Graph Api

2. เข้าสู่ระบบด้วยบัญชีผู้ใช้ที่เป็นเจ้าของเพจ



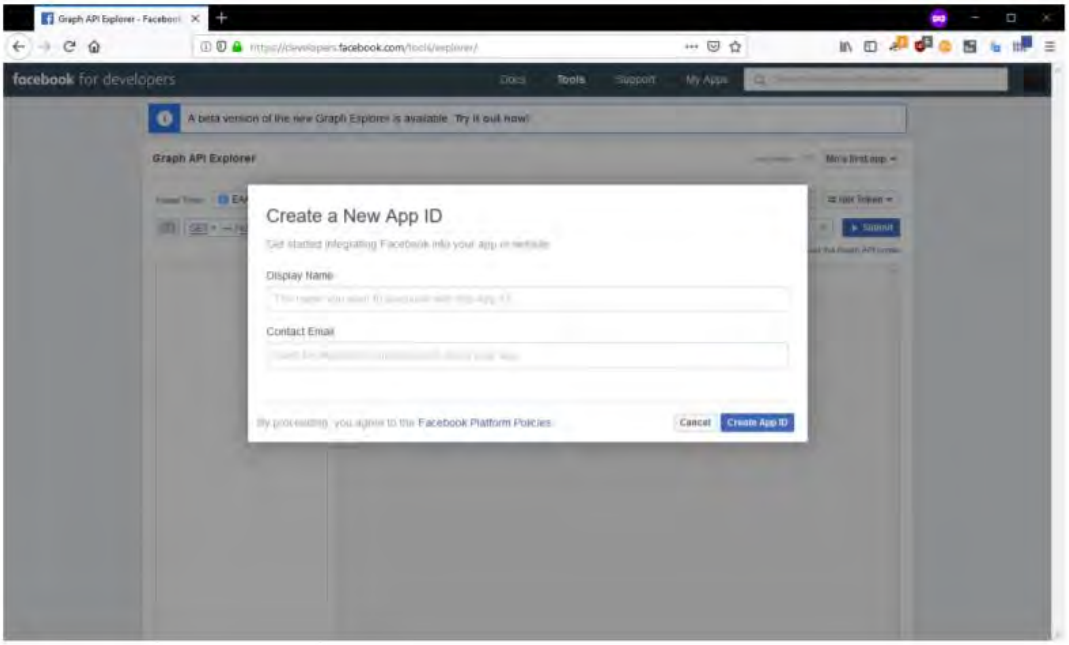
ภาพที่ 4.2 หน้าเข้าสู่ระบบ Facebook

3. เลือก My Apps > Add New App



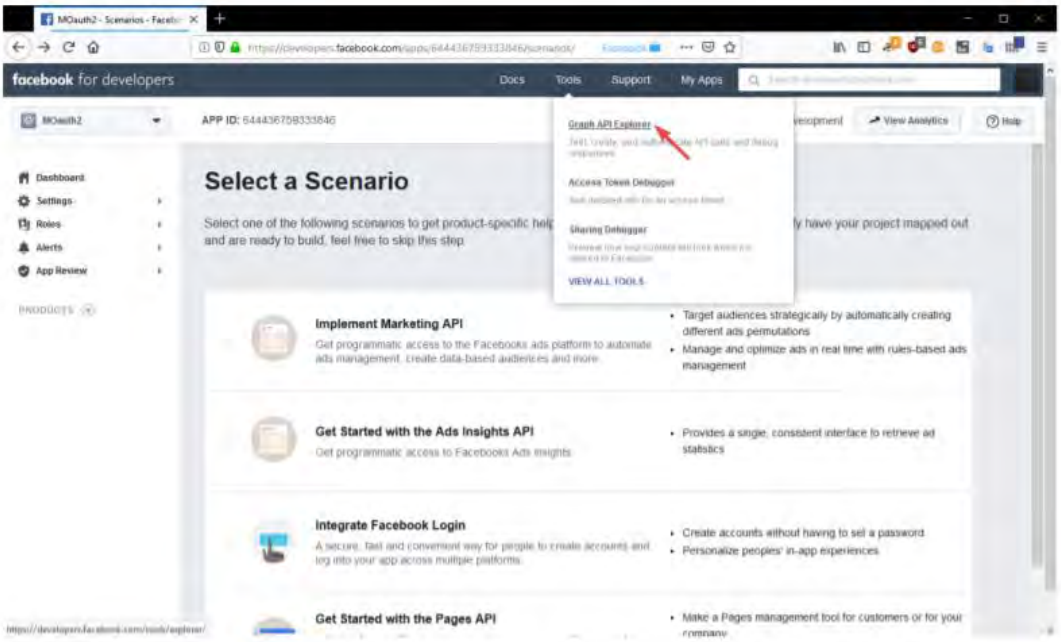
ภาพที่ 4.3 ตำแหน่งของปุ่ม Add New App

4. กรอกข้อมูลสำหรับแอพพลิเคชั่นของเรา



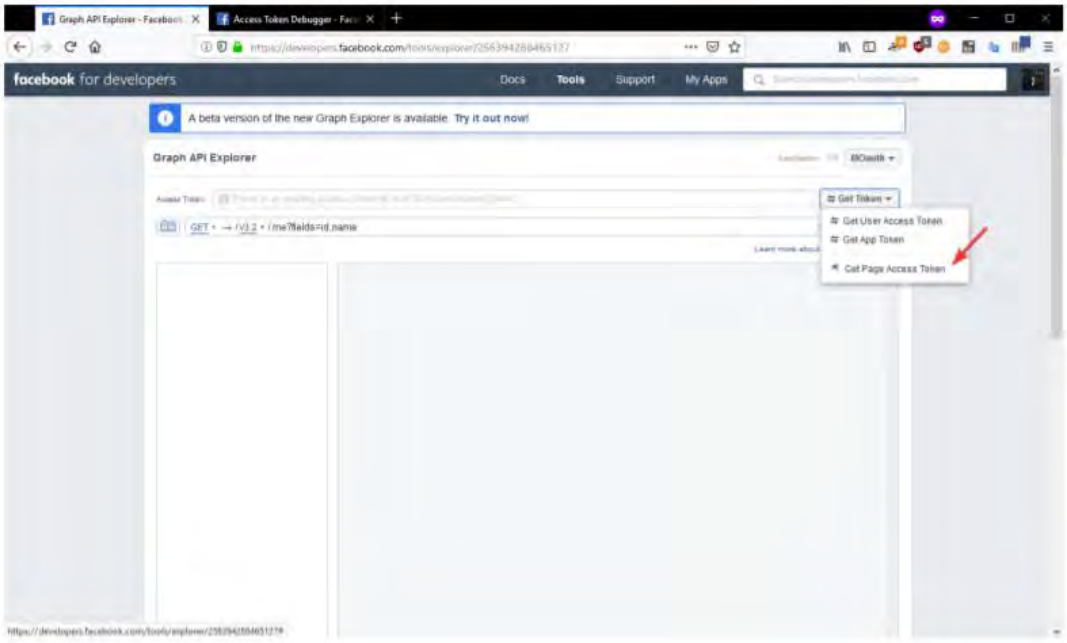
ภาพที่ 4.4 หน้าสร้าง New App ID

5. ไปที่ Tool > Graph API Explorer



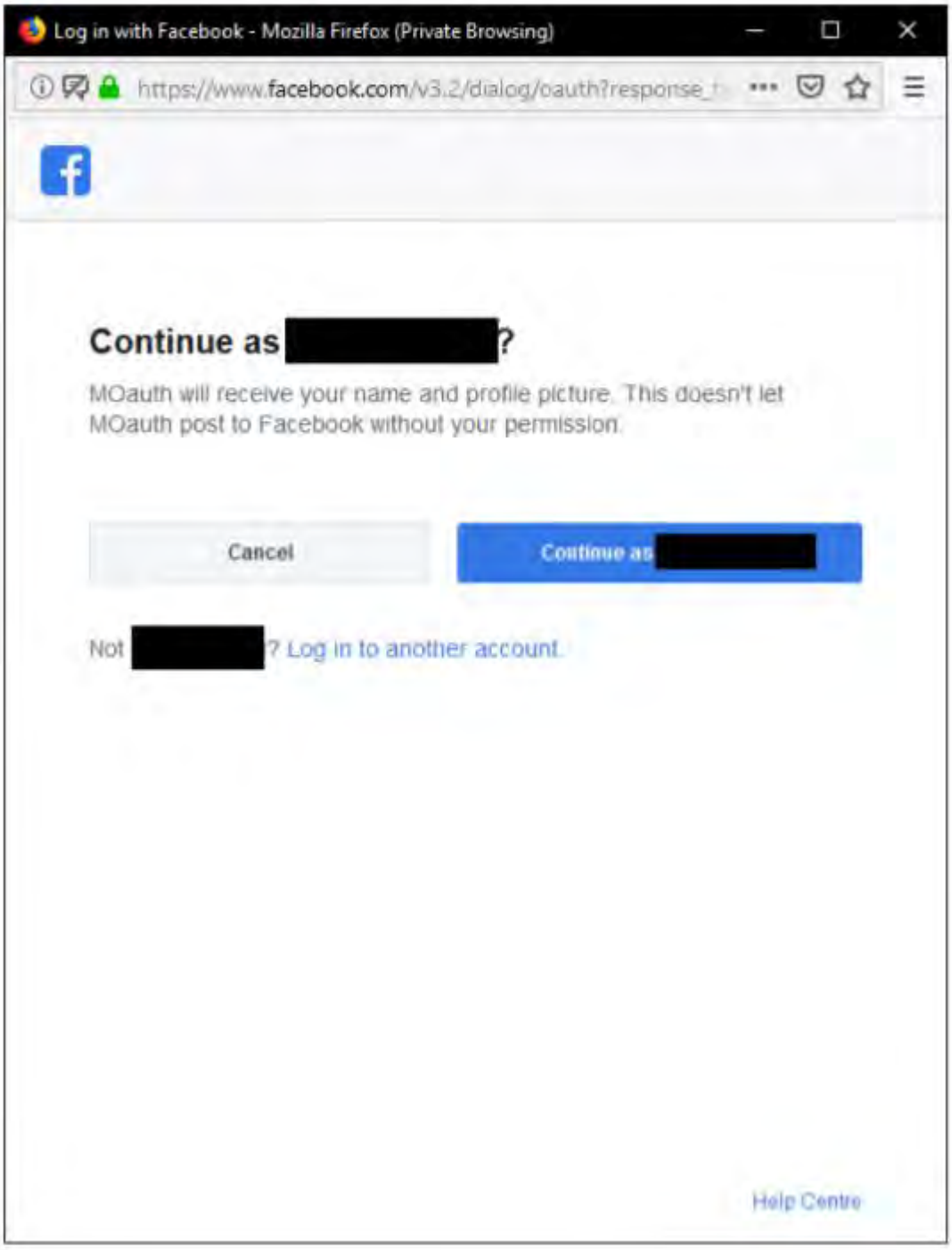
ภาพที่ 4.5 วิธีเข้าใช้งาน Graph Api Explorer

6. เลือก Get Token > Get Page Access Token



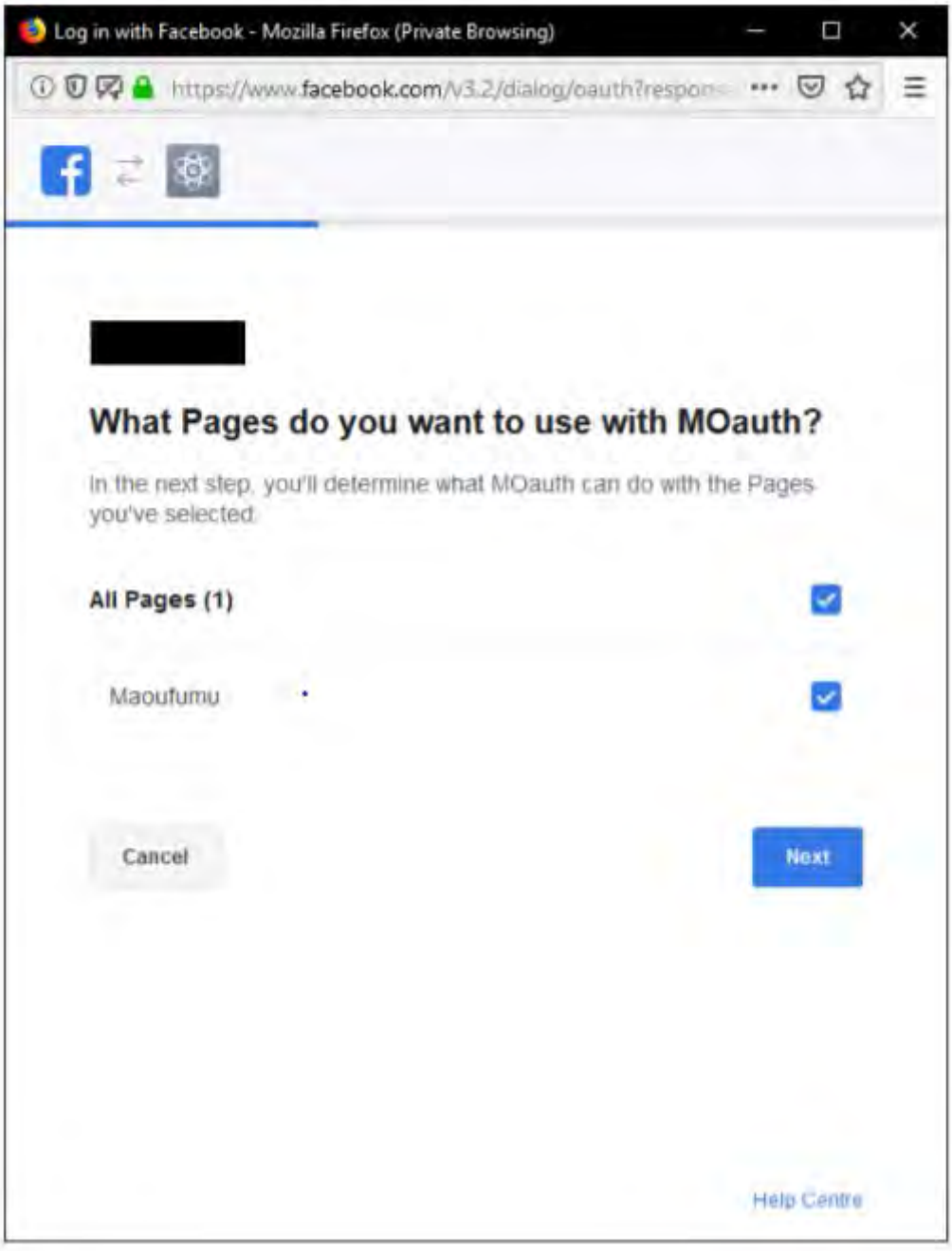
ภาพที่ 4.6 วิธีการได้รับ Token

7. เลือกเข้าสู่ระบบด้วยบัญชีผู้ใช้ที่เป็นเจ้าของเพจ



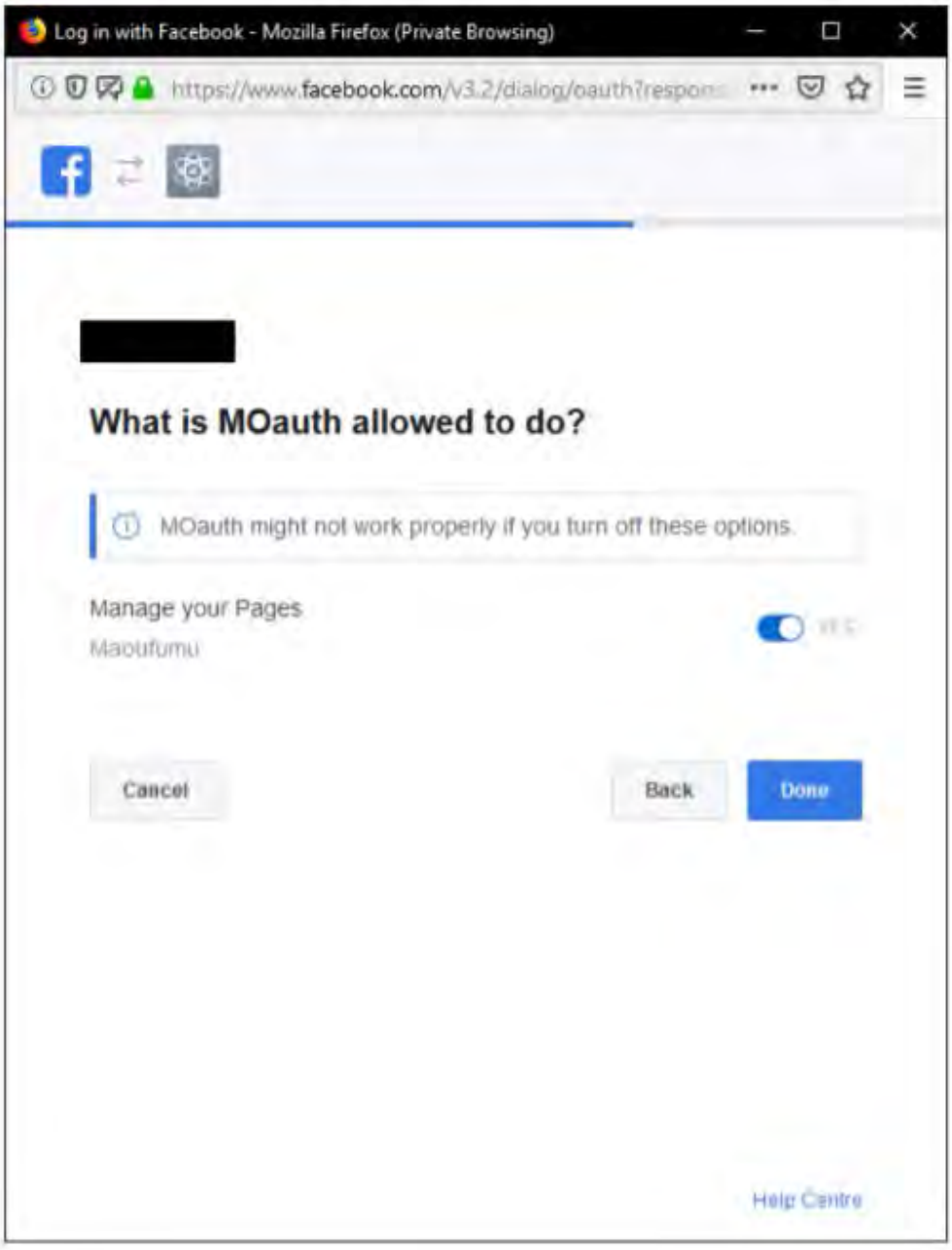
ภาพที่ 4.7 เข้าสู่ระบบด้วยบัญชีผู้ใช้

8. เลือกเพจที่ต้องการดึงความคิดเห็น



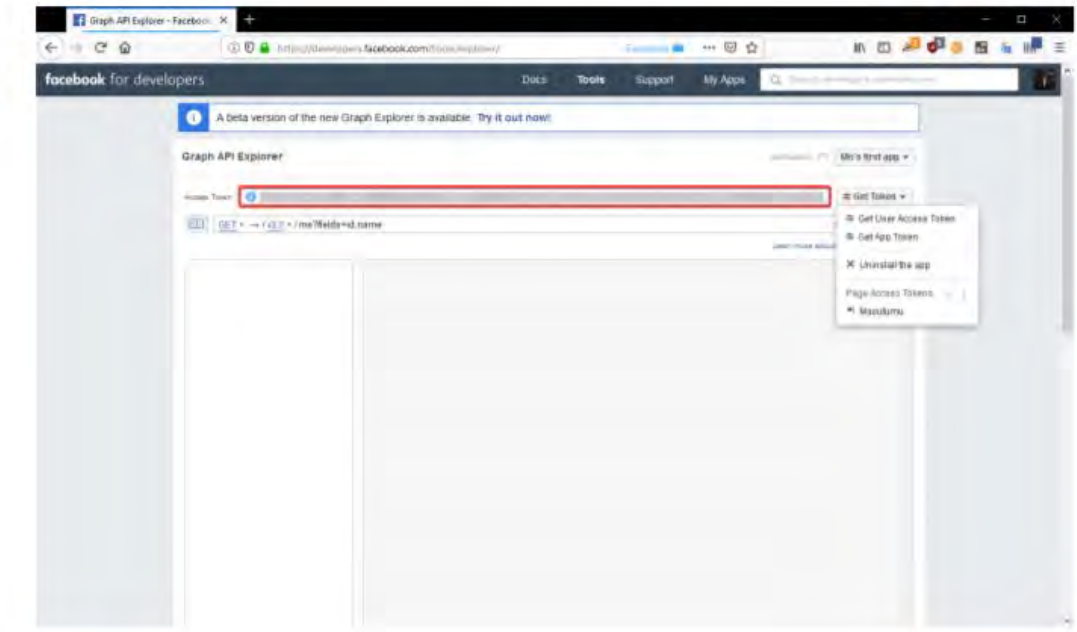
ภาพที่ 4.8 หน้าเลือกเพจที่ต้องการดึงข้อความ

9.กดยืนยัน



ภาพที่ 4.9 หน้าต่างก่อนกดยืนยัน

10. เฟสบุ๊คจะกลับมาที่หน้า Graph API explorer โดยช่อง access token จะมี token ของเพจที่เราเลือกไว้เรียบร้อย



ภาพที่ 4.10 หน้า Graph API ที่แสดง Token จากเพจที่เลือก

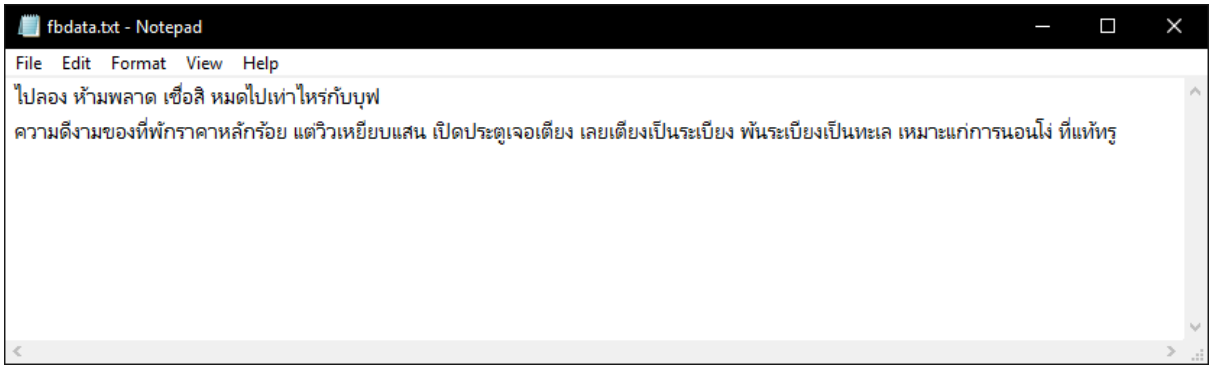
12.คัดลอกโทเคนไปใช้

หลังจากได้ token แล้ว เราจะทำการดึงข้อความแสดงความคิดเห็นจากเฟสบุ๊คโดยจะมีการเตรียมข้อมูลเบื้องต้นดังส่วนของโปรแกรมต่อไปนี้ โดยในส่วนของ token คือ token ที่เราได้มาจากขั้นตอนในด้านบน

```
import facebook

import codecs

import re



token = "token"

graph = facebook.GraphAPI(access_token=token, version="3.1")
```

```
info = graph.get_object(id='me',fields='ratings{review_text}')

rating = info['ratings']

data = rating['data']

print(data)

print('==============')

with codecs.open("data\\fbdata.txt", "w", "utf-8") as outputFile:

  for review in data:

    me = re.sub(r'[^ก-๙0-9A-z]',' ', review['review_text'])

    me = re.sub(r'[ๆ]',' ',me)

    me = re.sub(r'([ก-ฮ])\1{2,}', r'\1',me)

    me = re.sub(r'(\s)+',' ',me)

    outputFile.write(me+'\r\n')
```

โดยมีตัวอย่างผลลัพธ์ที่ได้คือ



ภาพที่ 4.11 ผลลัพธ์การดึงข้อมูลแสดงความคิดเห็นของร้านค้าบนเฟสบุ๊ค

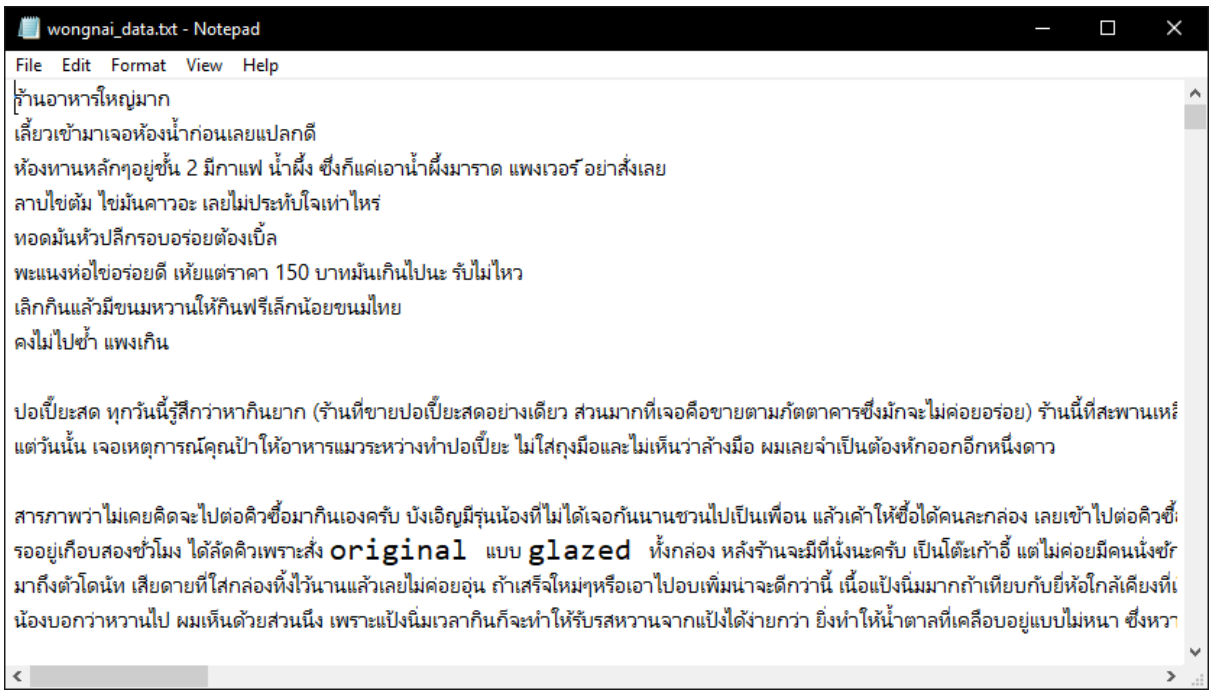
ก่อนหน้าที่จะสามารถวิเคราะห์ข้อมูลได้จำเป็นต้องทำการจัดการข้อมูล เพื่อให้โปรแกรมสามารถแยกแยะอารมณ์ความรู้สึกออกมาได้เป็นเชิงบวกและเชิงลบ เราจะใช้ความคิดเห็นจาก

Wongnai เป็นจำนวนทั้งหมด 999 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 555 ประโยค และความคิดเห็นเชิงลบ 444 ประโยค โดยแบ่งข้อมูลออกเป็น 2 ส่วน ส่วนแรก 749 ประโยค ใช้เป็นข้อมูลเข้าสำหรับ Training ส่วนอีก 250 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 145 ประโยค และความคิดเห็นเชิงลบ 105 ประโยค ใช้สำหรับทดสอบระบบ โดยจะทำการเก็บประโยคหรือคำไว้เป็นไฟล์ นามสกุล .txt แล้วให้โปรแกรมนำไฟล์ที่เราได้ แยกคำหรือประโยค เป็นเชิงบวกและเชิงลบไปฝึกระบบตามส่วนของโปรแกรม 2 ,3 ,4 และ 5

1. อ่านข้อมูลของวงในจากไฟล์ ก่อนจะนำไปแยกเป็นความคิดเห็นเชิงบวกและเชิงลบด้วยมือ

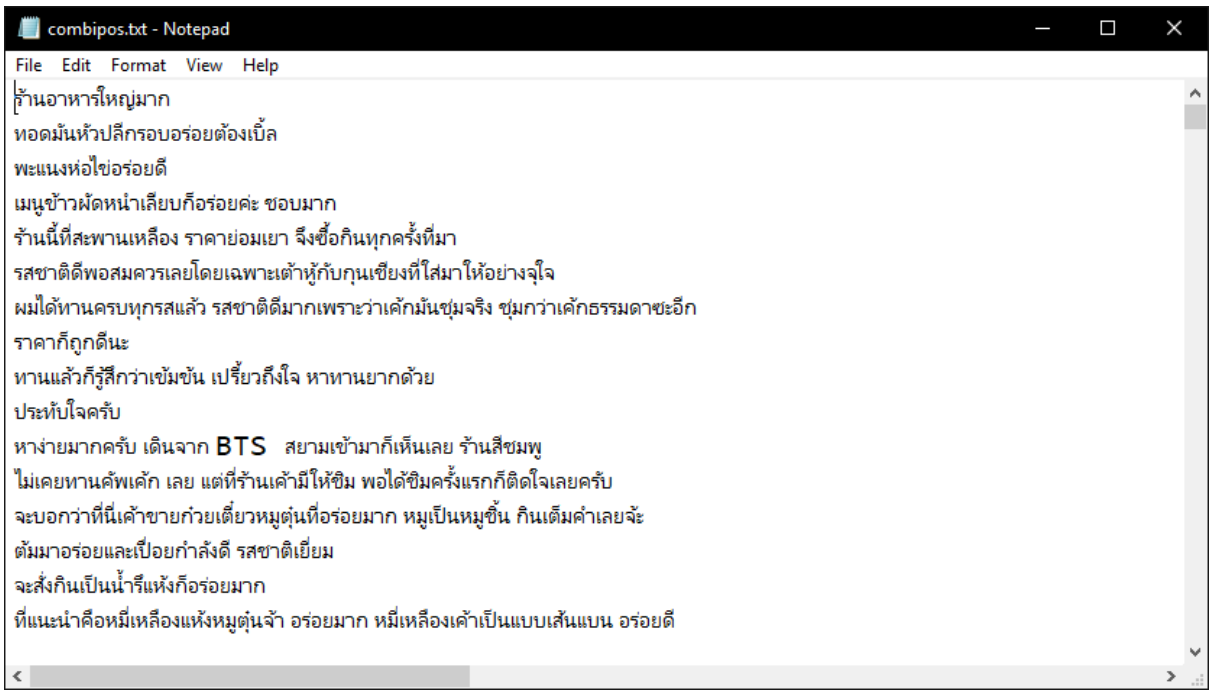
```
import pandas
import deepcut
import re
pandas.set_option('max_colwidth', 9999)

wongnai = pandas.read_csv('data\\wongnai-corpus-
master\\review\\w_review_train.csv',error_bad_lines=False, sep=';', names = ["text",
"rating"], skiprows=0, nrows=300)
reviews = []
for index, row in wongnai.iterrows():
    if row['rating'] <= 3:
        temp = re.sub(r'(\r\n)+','\r\n',row['text'])
        temp = re.sub(r'\r\n( )+\r\n','\r\n',temp)
        temp2 = temp.strip()
        print(temp2.split('\r\n'))
        reviews.append(temp2.split('\r\n'))

with open('data\\wongnai_data2.txt','w',encoding='UTF-8')as wongnai_data:
    for review in reviews:
        for sentence in review:
            print(sentence.strip(), file=wongnai_data)
        print('', file=wongnai_data)
```

โดยมีผลลัพธ์ที่ได้คือ



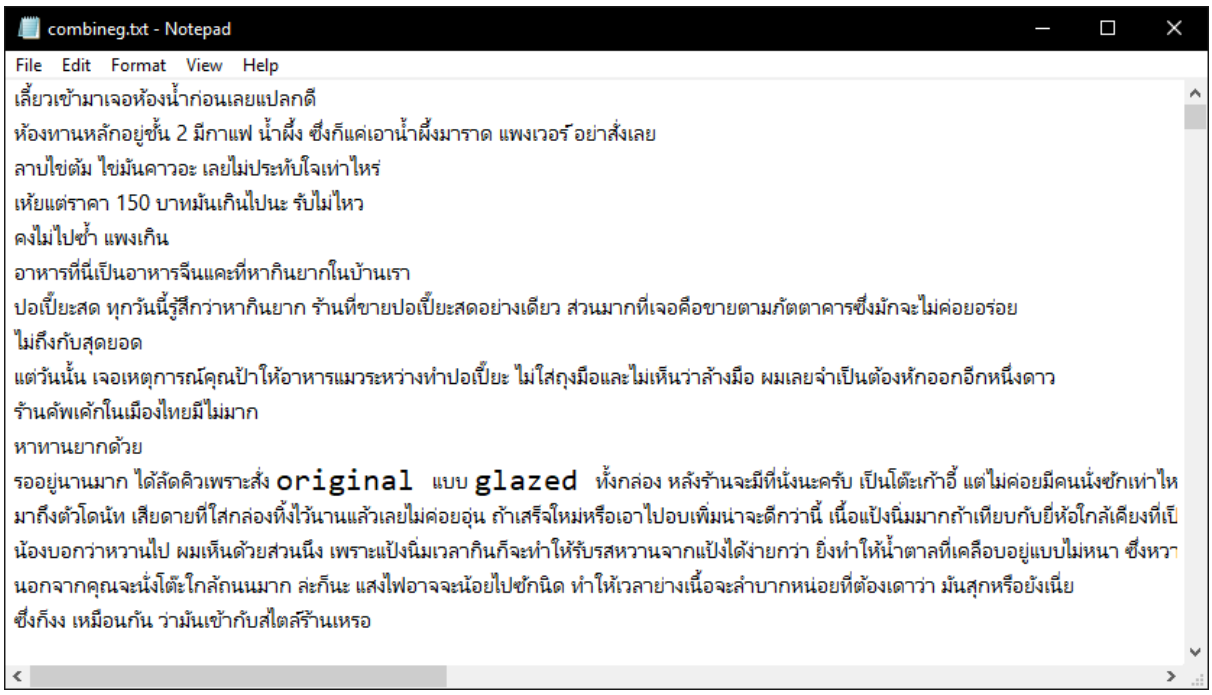
ภาพที่ 4.12 ความคิดเห็นที่มีต่อร้านอาหารจาก wongnai

เมื่อแยกแล้วจะได้เป็นไฟล์ความคิดเห็นแง่บวก



ภาพที่ 4.13 ไฟล์ความคิดเห็นแง่บวก

และไฟล์ความคิดเห็นแง่ลบ



ภาพที่ 4.14 ไฟล์ความคิดเห็นแง่ลบ

2. เรียกใช้ไลบรารี่ และอ่านข้อมูลความคิดเห็นที่แยกเตรียมไว้

```
import deepcut
import re
import numpy
import nltk
import pandas
import collections, itertools
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
pandas.set_option('max_colwidth', 999)


data_pos = []
with open('combipos.txt', 'r', encoding='utf-8') as omega:
    for line in omega:
        data_pos.append([line.strip(), 'pos'])
```

```
data_neg = []
with open('combineg.txt', 'r', encoding='utf-8') as omega:
    for line in omega:
        data_neg.append([line.strip(), 'neg'])
```

3. ทำการแบ่งประโยคออกเป็นคำ และทำการกำกับเชิงความคิดเห็นของประโยค

```
def split_words (sentence):
    return deepcut.tokenize(''.join(sentence.split()))

def get_words_in_reviews(sentences):
    all_words = []
    for (words, sentiment) in sentences:
        all_words.extend(words)
    return all_words

def get_word_features(wordlist):
    wordlist = nltk.FreqDist(wordlist)
    word_features = [word[0] for word in wordlist.most_common()]
    return word_features

def extract_features(document):
    document_words = set(document)
    features = {}
    for word in word_features:
        features['contains(%s)' % word] = (word in document_words)
    return features

data = []
```

```
data_reviews = []
data_tokens = []
data_labels = []
for sentence, sentiment in data_pos+data_neg:
    data.append([split_words(sentence), sentiment])
    data_reviews.append(sentence)
    data_tokens.append(split_words(sentence))
    data_labels.append(sentiment)

features_data = numpy.array(data)
```

4. แบ่งข้อมูลสำหรับฝึกตัวจำแนกและสำหรับทดสอบตัวจำแนก

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    data_tokens,
    data_labels,
    test_size=0.25,
    random_state=1412)

from sklearn.preprocessing import MultiLabelBinarizer

onehot_ = MultiLabelBinarizer()
onehot_.fit(data_tokens)



from sklearn.model_selection import KFold

k_fold = KFold(n_splits=10, random_state=1997, shuffle=True)
```

```
word_features = None
accuracy_scores = []
```

หลังจากเราได้ทำการฝึกตัวจำแนกแล้วเราจะได้โปรแกรมที่สามารถวิเคราะห์ความคิดเห็น

ในส่วนของการวิเคราะห์ข้อมูลเราจะใช้ตัวจำแนกนาอีฟเบย์ แบบ NLTK เป็นตัวจำแนกหลัก ส่วนตัวจำแนกอื่นๆจะเป็นตัวจำแนกที่ใช้ในการหาค่าความแม่นยำว่าตัวจำแนกใดมีความแม่นยำสูงสุด โดยจะแสดงไลบรารี่ ที่ใช้ในแต่ละตัวจำแนกและผลลัพธ์ของตัวจำแนกนั้นตามส่วนของโปรแกรม 5. ถึง 20.

5. ทำการฝึกตัวจำแนกนาอีฟเบย์ NLTK

```
word_features = get_word_features(get_words_in_reviews( features_data.tolist()))
train_features = nltk.classify.apply_features( extract_features, features_data.tolist())
classifier = nltk.NaiveBayesClassifier.train( train_features)
```

```
for train_set, test_set in k_fold.split(features_data):
    word_features = get_word_features(get_words_in_reviews(
features_data[train_set].tolist()))
    train_features = nltk.classify.apply_features(
extract_features,features_data[train_set].tolist())
    test_features = nltk.classify.apply_features(extract_features,
features_data[test_set].tolist())
    classifier = nltk.NaiveBayesClassifier.train (train_features)
    refsets = collections.defaultdict(set)
    testsets = collections.defaultdict(set)
    refset = []
    testset = []

    for i, (feats, label) in enumerate(test_features):
```

```
        observed = classifier.classify(feats)
        refset.append(label)
        testset.append(observed)
```

6. ทดสอบความแม่นยำของตัวจำแนกนาอีฟเบย์ NLTK

```
Xy_test = []
for z in range(len(y_test)):
    Xy_test.append([X_test[z],y_test[z]])


features_data2 = numpy.array(Xy_test)
refset2 = []
testset2 = []
out_features = nltk.classify.apply_features(extract_features, features_data2.tolist())
for i, (feats, label) in enumerate(out_features):
    observed = classifier.classify(feats)
    #print(observed)
    refset2.append(label)
    testset2.append(observed)
print('Accuracy = ',accuracy_score(refset2,testset2))
print(classification_report(refset2,testset2))
print(confusion_matrix(refset2,testset2,labels=['pos','neg']))
```

```
Accuracy =  0.804
              precision    recall  f1-score   support

         neg       0.74      0.82      0.78       105
         pos       0.86      0.79      0.82       145

   micro avg       0.80      0.80      0.80       250
   macro avg       0.80      0.81      0.80       250
weighted avg       0.81      0.80      0.81       250

[[115  30]
 [ 19  86]]
```

ภาพที่ 4.15 ค่าการประเมิณผลต่าง ๆ ของตัวจำแนกนาอีฟเบย์ NLTK

7. จำแนกข้อมูลด้วยตัวจำแนกนาอีฟเบย์ NLTK

```
features_data3 = numpy.array(testdata)
refset3 = []
testset3 = []
out_features = nltk.classify.apply_features(extract_features, features_data3.tolist())
for i, (feats, label) in enumerate(out_features):
    observed = classifier.classify(feats)
    print(observed)
    refset3.append(label)
    testset3.append(observed)
```

```
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
pos
```

ภาพที่ 4.16 ตัวอย่างผลลัพธ์จากการจำแนกความคิดเห็นด้วยตัวจำแนกนาอีฟเบย์ NLTK

8. สร้างตัวจำแนกซัพพอร์ตเวกเตอร์แมชชีน และปรับตัวแบบให้เข้ากับข้อมูล

```
from sklearn import svm
svm = svm.LinearSVC()
svm.fit(onehot_.transform(X_train), y_train)
```

9. ทดสอบความแม่นยำของตัวจำแนกซัพพอร์ตเวกเตอร์แมชชีน

```
print('Accuracy = ',accuracy_score(y_test,svm.predict(onehot_.transform(X_test))))
print(classification_report(y_test,svm.predict(onehot_.transform(X_test))))
print(confusion_matrix(y_test,svm.predict(onehot_.transform(X_test)),labels=['pos','neg']))
```

```
Accuracy =  0.804
              precision    recall  f1-score   support

         neg       0.74      0.82      0.78       105
         pos       0.86      0.79      0.82       145

   micro avg       0.80      0.80      0.80       250
   macro avg       0.80      0.81      0.80       250
weighted avg       0.81      0.80      0.81       250

[[115  30]
 [ 19  86]]
```

ภาพที่ 4.17 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกซัพพอร์ตเวกเตอร์แมชชีน

10. จำแนกข้อมูลด้วยตัวจำแนกซัพพอร์ตเวกเตอร์แมชชีน

```
svm.predict(onehot_.transform(testdata_tokens))
```

```
array(['pos', 'neg', 'pos', 'pos', 'pos', 'pos', 'pos', 'neg', 'pos',
       'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'neg', 'pos', 'pos',
       'pos', 'neg', 'pos', 'pos', 'neg', 'pos', 'pos', 'neg', 'neg',
       'neg', 'neg', 'neg', 'pos', 'neg', 'pos', 'pos', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'pos', 'pos', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg'], dtype='<U3')
```

ภาพที่ 4.18 ตัวอย่างผลลัพธ์การจำแนกความคิดเห็นด้วยตัวจำแนกซัพพอร์ตเวกเตอร์แมชชีน

11. สร้างตัวจำแนกนาอีฟเบย์ และปรับตัวแบบให้เข้ากับข้อมูล

```
from sklearn.naive_bayes import BernoulliNB
bnbc = BernoulliNB(binarize=None)
bnbc.fit(onehot_.transform(X_train), y_train)
```

12. ทดสอบความแม่นยำของตัวจำแนกนาอีฟเบย์

```
print('Accuracy = ',accuracy_score(y_test,bnbc.predict(onehot_.transform(X_test))))
```

```
print(classification_report(y_test,bnbc.predict(onehot_.transform(X_test))))
print(confusion_matrix(y_test,bnbc.predict(onehot_.transform(X_test)),labels=['pos','neg']))
```

```
Accuracy =  0.872
              precision    recall  f1-score   support

         neg       0.85      0.84      0.85       105
         pos       0.88      0.90      0.89       145

   micro avg       0.87      0.87      0.87       250
   macro avg       0.87      0.87      0.87       250
weighted avg       0.87      0.87      0.87       250

[[130  15]
 [ 17  88]]
```

ภาพที่ 4.19 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกนาอีฟเบย์

13. จำแนกข้อมูลด้วยตัวจำแนกนาอีฟเบย์

```
bnbc.predict(onehot_.transform(testdata_tokens))
```

```
array(['pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos',
       'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos', 'pos',
       'pos', 'pos', 'pos', 'pos', 'neg', 'pos', 'pos', 'neg', 'neg',
       'neg', 'pos', 'neg', 'pos', 'neg', 'pos', 'pos', 'neg', 'pos',
       'neg', 'neg', 'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'neg', 'neg', 'neg'], dtype='<U3')
```

ภาพที่ 4.20 ตัวอย่างผลลัพธ์ของการจำแนกความคิดเห็นด้วยตัวจำแนกนาอีฟเบย์

14. สร้างโครงข่ายประสาทเทียม พร้อมทั้งตั้งค่าเริ่มต้น

```
import tensorflow as tf
import keras
tf.reset_default_graph()
```

```
vocab_len = len(onehot_nn.classes_)
inputs_ = tf.placeholder(dtype=tf.float32, shape=[None,vocab_len], name="inputs")
targets_ = tf.placeholder(dtype=tf.float32, shape=[None,2], name="targets")

h1 = tf.layers.dense(inputs_, 500, activation=tf.nn.relu)
#h2 = tf.layers.dense(h1, 500, activation=tf.nn.relu)
#h3 = tf.layers.dense(h2, 500, activation=tf.nn.relu)
logits = tf.layers.dense(h1, 2, activation=tf.nn.relu)
output = tf.nn.sigmoid(logits)

loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=logits,
labels=targets_))

optimizer = tf.train.AdamOptimizer(0.001).minimize(loss)

correct_pred = tf.equal(tf.argmax(logits, 1), tf.argmax(targets_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32), name='accuracy')

def label2bool(labels):
    return [[1,0] if label == "pos" else [0,1] for label in labels]

def get_batch(X, y, batch_size):
    for batch_pos in range(0,len(X),batch_size):
        yield X[batch_pos:batch_pos+batch_size], y[batch_pos:batch_pos+batch_size]

sess = tf.Session()

# Initializing the variables
sess.run(tf.global_variables_initializer())
```

15. ฝึกตัวจำแนกโครงข่ายประสาทเทียม

```
epochs = 10
batch_size = 3000

for epoch in range(epochs):
    for X_batch, y_batch in get_batch(onehot_nn.transform(nnx_train),
label2bool(nny_train), batch_size):
        loss_value, _ = sess.run([loss, optimizer], feed_dict={
            inputs_: X_batch,
            targets_: y_batch
        })
        print("Epoch: {} \t Training loss: {}".format(epoch, loss_value))

    acc = sess.run(accuracy, feed_dict={
        inputs_: onehot_nn.transform(nnx_valid),
        targets_: label2bool(nny_valid)
    })

    print("Epoch: {} \t Validation Accuracy: {}".format(epoch, acc))
```

16. ทดสอบความแม่นยำของตัวจำแนกโครงข่ายประสาทเทียม

```
test_result = sess.run(output,feed_dict={
    inputs_:onehot_nn.transform(X_test)
})
i=0
nn_pred = []
for stuff in test_result:
    if test_result[i][0]>test_result[i][1]:
        nn_pred.append('pos')
```

```
    else:
        nn_pred.append('neg')
    i=i+1


print('Accuracy = ',accuracy_score(y_test, nn_pred))
print(classification_report(y_test, nn_pred))
print(confusion_matrix(y_test, nn_pred,labels=['pos','neg']))
```

```
Accuracy =  0.936
              precision    recall  f1-score   support

         neg       0.92      0.92      0.92       105
         pos       0.94      0.94      0.94       145

   micro avg       0.94      0.94      0.94       250
   macro avg       0.93      0.93      0.93       250
weighted avg       0.94      0.94      0.94       250

[[137   8]
 [  8  97]]
```

ภาพที่ 4.21 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกโครงข่ายประสาทเทียม

17. จำแนกข้อมูลด้วยตัวจำแนกโครงข่ายประสาทเทียม

```
real_result = sess.run(output,feed_dict={
    inputs_:onehot_nn.transform(testdata_tokens)
})
i=0
nn_pred_real = []
for stuff in real_result:
    #print(test_result[i][0], test_result[i][1])
    #print(test_result[i][0]-test_result[i][1])
    if real_result[i][0]>real_result[i][1]:
```

```
        #print('pos')
        nn_pred_real.append('pos')
    else:
        #print('neg')
        nn_pred_real.append('neg')
    i=i+1
```

```
pos
pos
pos
pos
pos
pos
neg
pos
pos
neg
neg
```

ภาพที่ 4.22 ตัวอย่างผลลัพธ์การจำแนกความคิดเห็นด้วยตัวจำแนกโครงข่ายประสาทเทียม

18. ทำการเตรียมข้อมูล สร้างตัวจำแนกการถดถอยโลจิสติก และปรับตัวแบบให้เข้ากับข้อมูล

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(
    analyzer = 'word',
    lowercase = False,)

features = vectorizer.fit_transform(data_reviews)
features_nd = features.toarray()

from sklearn.linear_model import LogisticRegression
log_model = LogisticRegression()
log_model = log_model.fit(X=lrx_train, y=lry_train)
```

19. ทดสอบความแม่นยำของตัวจำแนกการถดถอยโลจิสติก

```
lry_pred = log_model.predict(lrx_test)
print('Accuracy = ',accuracy_score(lry_test, lry_pred))
print(classification_report(lry_test, lry_pred))
print(confusion_matrix(lry_test, lry_pred,labels=['pos','neg']))
```

```
Accuracy =  0.776
              precision    recall  f1-score   support

         neg       0.72      0.75      0.74       105
         pos       0.82      0.79      0.80       145

   micro avg       0.78      0.78      0.78       250
   macro avg       0.77      0.77      0.77       250
weighted avg       0.78      0.78      0.78       250

[[115  30]
 [ 26  79]]
```

ภาพที่ 4.23 ค่าการประเมินผลต่าง ๆ ของตัวจำแนกการถดถอยเชิงโลจิสติก

20. จำแนกข้อมูลด้วยตัวจำแนกการถดถอยโลจิสติก

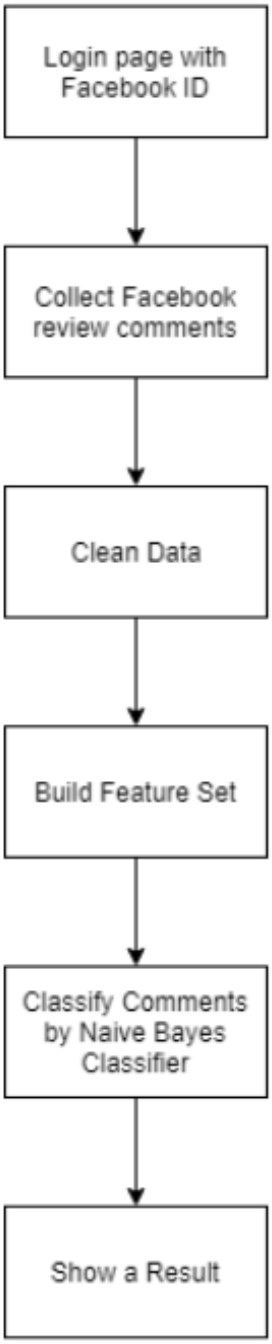
```
real_feature = vectorizer.transform(testdata_reviews)
real_feature_array = real_feature.toarray()
real_lry_pred = log_model.predict(real_feature_array)
print(real_lry_pred)
```

```
['pos' 'pos' 'pos' 'pos' 'pos' 'pos' 'pos' 'pos' 'neg' 'pos' 'pos' 'pos'
 'pos' 'pos' 'pos' 'pos' 'pos' 'pos' 'pos' 'neg' 'pos' 'pos' 'pos' 'pos'
 'pos' 'neg' 'pos' 'pos' 'neg' 'neg' 'pos' 'pos' 'pos' 'pos' 'neg' 'neg'
 'pos' 'pos' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'pos' 'pos']
```

ภาพที่ 4.24 ตัวอย่างผลลัพธ์การจำแนกความคิดเห็นด้วยตัวจำแนกการถดถอยเชิงโลจิสติก

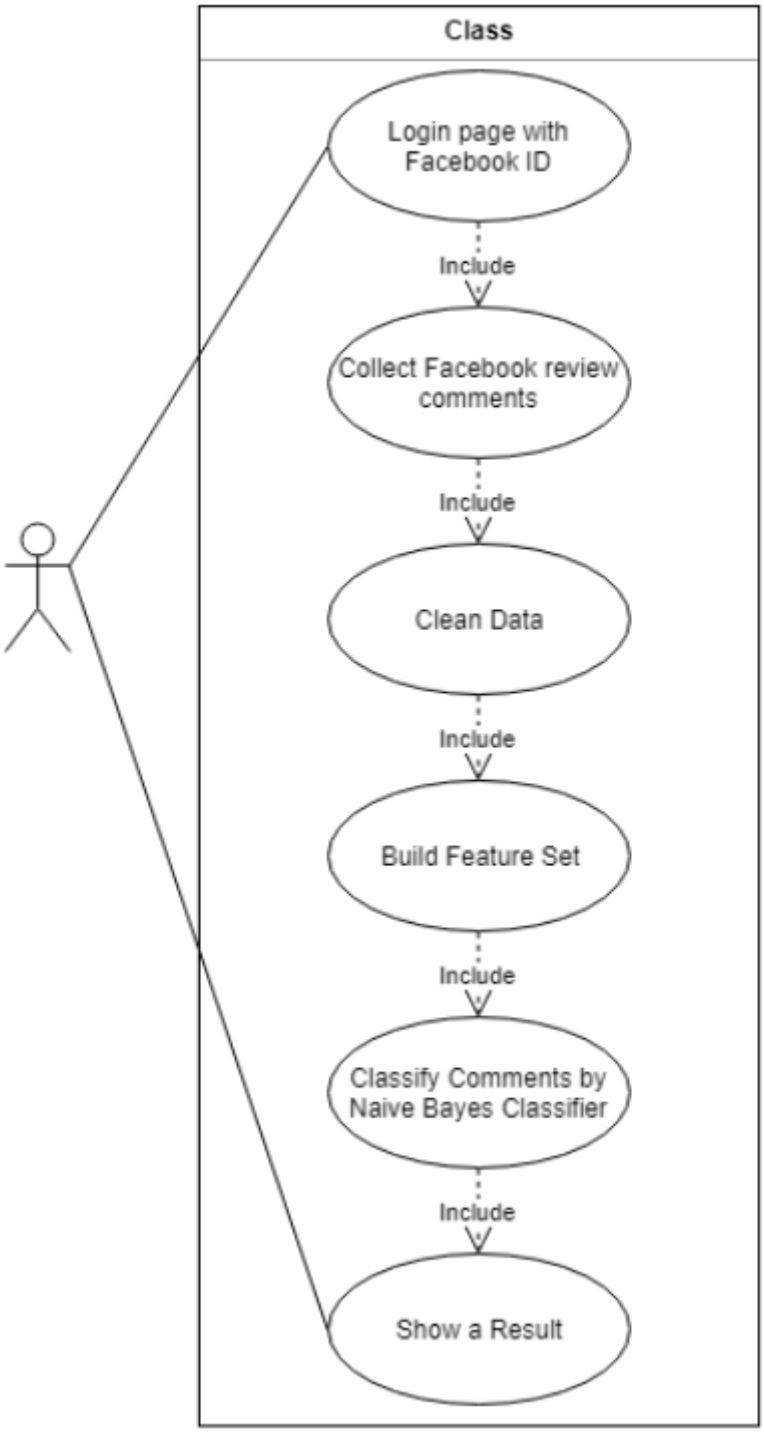
4.2 แผนภาพแสดงภาพรวมการทำงานในการวิเคราะห์อารมณ์ความรู้สึกของความคิดเห็นบนเฟสบุ๊ค

แสดงภาพรวมของการทำงานของระบบในการเคราะห์ข้อความแสดงความคิดเห็นจากรีวิวบนเฟสบุ๊คโดยที่การทำงานของระบบจะเริ่มจากผู้ใช้ล็อกอินด้วยแอคเคาท์เฟสบุ๊คแล้วระบบจะทำการดึงข้อมูลความคิดเห็นจากรีวิวบนหน้าเพจเฟสบุ๊คที่ผู้ใช้เป็นเจ้าของและนำมาข้อมูลมาทำการปรับให้อยู่ในรูปแบบที่สามารถนำไปใช้ได้ ก่อนที่จะนำข้อมูลไปทำนายเชิงความคิดเห็นด้วยตัวจำแนกที่สร้างไว้ ดังภาพที่ 4.25



ภาพที่ 4.25 ภาพรวมการวิเคราะห์อารมณ์ความรู้สึกของความคิดเห็นจากรีวิวบนเฟสบุ๊ค

4.3 แผนภาพยูสเคส



ภาพที่ 4.26 แผนภาพยูสเคสองระบบวิเคราะห์ความคิดเห็นของการรีวิวบนเฟสบุ๊ค

ตารางที่ 4.1 ตารางอธิบายแผนภาพ Use case

Use case name	ระบบวิเคราะห์ความคิดเห็นของการรีวิวบนเฟสบุ๊ค
Participation actor	ผู้ใช้งานระบบ
Entry condition	เปิดระบบ
Flow of event	1. ผู้ใช้งานล็อกอินด้วยแอคเคาท์ของเฟสบุ๊คที่ต้องการนำความคิดเห็นของการรีวิว 2. ระบบดึงความคิดเห็นจากแอคเคาท์นั้น 3. ระบบเตรียมข้อมูลเพื่อที่จะนำไปวิเคราะห์ข้อความแสดงความคิดเห็น 4. ระบบสร้างชุดข้อมูลฟีเจอร์สำหรับวิเคราะห์ประเภทของข้อความแสดงความคิดเห็น 5. ระบบวิเคราะห์ประเภทอารมณ์ของข้อความแสดงความคิดเห็น 6. ระบบแสดงผลลัพธ์ของการวิเคราะห์อารมณ์ของข้อความแสดงความคิดเห็นจากรีวิวบนเฟสบุ๊ค
Exit condition	ปิดระบบ

4.4 ภาษาและโปรแกรมที่ใช้ในการพัฒนาระบบ

ภาษาที่นำมาใช้ในการพัฒนาระบบ

1. ภาษาPython - นำมาใช้ในการพัฒนาส่วนวิเคราะห์ระบบ

โปรแกรมที่นำมาใช้ในการพัฒนาระบบ

1. Visual Studio Code – ใช้สำหรับพัฒนาโปรแกรม
2. Google colaboratory - ใช้สำหรับพัฒนาโปรแกรม
3. Notepad - ใช้สำหรับข้อมูล text file
4. Facebook SDK - ชุดฟังก์ชั่นการทำงานฝั่งไคลเอนต์สำหรับการการเรียก API และการใช้การเข้าสู่ระบบด้วย Facebook
5. Virtualenv - ช่วยในการแยก environment ในการทำงาน Python ของแต่ละโปรเจคออกจากกัน

บทที่ 5

ผลการทดสอบระบบ

ในบทนี้จะกล่าวถึงการสรุปผลการออกแบบและการพัฒนาระบบจากบทที่ 4 โดยการทดสอบระบบจำแนกความคิดเห็นบนเฟสบุ๊ค สามารถทดสอบความน่าเชื่อถือของผลลัพธ์การแยกประเภทของข้อความได้ โดยจะพิจารณาจากค่าความถูกต้องและค่าความแม่นยำในการจำแนกความคิดเห็นว่ามีเปอร์เซนต์ความถูกต้องมากน้อยเพียงใด

5.1 การทดสอบการจำแนกความคิดเห็นด้วย Naive Bayes

การทดสอบระบบด้วย Naive Bayes ได้ใช้ไลบรารี่ 2 ตัวแตกต่างกันคือ ไลบรารี่ NLTK และ Scikit-learn โดยที่การจำแนกความคิดเห็นด้วย NLTK Naive Bayes นั้น มีการทำ feature extraction ส่วน Scikit-learn Naive Bayes ไม่มีการทำ feature extraction ซึ่งการทำ feature extraction คือ การจำแนกข้อมูลที่เป็นลักษณะที่สำคัญออกมา สำหรับในที่นี้คือจำนวนครั้งที่คำคำนั้นปรากฏในข้อมูลที่ใช้ฝึกระบบ และเมื่อนำ feature ไปใช้ เราจะได้ว่าในแต่ละประโยคนั้น มีคำใดบ้างที่ระบบเคยเห็นจากข้อมูลฝึก

5.1.1 การทดสอบการจำแนกความคิดเห็นด้วย NLTK Naive Bayes

การทดสอบการจำแนกความคิดเห็นด้วย Naive Bayes ของ NLTK กับประโยคทั้งหมด 999 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 555 ประโยค และความคิดเห็นเชิงลบ 444 ประโยค โดยแบ่งข้อมูลออกเป็น 2 ส่วน ส่วนแรก 749 ประโยค ใช้เป็นข้อมูลเข้าสำหรับให้ระบบเรียนรู้ ส่วนอีก 250 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 145 ประโยค และความคิดเห็นเชิงลบ 105 ประโยค ใช้สำหรับทดสอบระบบ และพิจารณาค่าความถูกต้อง ค่าความแม่นยำ ค่าการเรียกคืน และค่าวัดประสิทธิภาพว่ามีมากน้อยเพียงใด โดยได้ผลการทดสอบดังนี้

ตารางที่ 5.1 confusion matrix ของการจำแนกความคิดเห็นด้วย NLTK Naive Bayes

		predicted	
		positive	negative
actual	positive	139	6
	negative	4	101

ตารางที่ 5.2 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย NLTK Naive Bayes

	Precision	Recall	F-measure
positive	0.97	0.96	0.97
negative	0.94	0.96	0.95

และมีค่าความถูกต้อง = 0.96

5.1.2 การทดสอบการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes

การทดสอบการจำแนกความคิดเห็นด้วย Naive Bayes ของ Scikit-learn กับประโยคทั้งหมด 999 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 555 ประโยค และความคิดเห็นเชิงลบ 444 ประโยค โดยแบ่งข้อมูลออกเป็น 2 ส่วน ส่วนแรก 749 ประโยค ใช้เป็นข้อมูลเข้าสำหรับให้ระบบเรียนรู้ ส่วนอีก 250 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 145 ประโยค และความคิดเห็นเชิงลบ 105 ประโยค ใช้สำหรับทดสอบระบบ และพิจารณาค่าความถูกต้อง ค่าความแม่นยำ ค่าการเรียกคืน และค่าวัดประสิทธิภาพ ว่ามีมากน้อยเพียงใด โดยได้ผลการทดสอบดังนี้

ตารางที่ 5.3 confusion matrix ของการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes

		predicted	
		positive	negative
actual	positive	130	15
	negative	17	88

ตารางที่ 5.4 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes

	Precision	Recall	F-measure
positive	0.88	0.90	0.89
negative	0.85	0.84	0.85

และมีค่าความถูกต้อง = 0.872

5.2 การทดสอบการจำแนกความคิดเห็นด้วย Support Vector Machine

การทดสอบการจำแนกความคิดเห็นด้วย Support Vector Machine กับประโยคทั้งหมด 999 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 555 ประโยค และความคิดเห็นเชิงลบ 444 ประโยค โดยแบ่งข้อมูลออกเป็น 2 ส่วน ส่วนแรก 749 ประโยค ใช้เป็นข้อมูลเข้าสำหรับให้ระบบเรียนรู้ ส่วนอีก 250 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 145 ประโยค และความคิดเห็นเชิงลบ 105 ประโยค ใช้สำหรับทดสอบระบบ และพิจารณาค่าความถูกต้อง ค่าความแม่นยำ ค่าการเรียกคืน และค่าวัดประสิทธิภาพ ว่ามีมากน้อยเพียงใด โดยได้ผลการทดสอบดังนี้

ตารางที่ 5.5 confusion matrix ของการจำแนกความคิดเห็นด้วย Support Vector Machine

		predicted	
		positive	negative
actual	positive	115	30
	negative	19	86

ตารางที่ 5.6 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Support Vector Machine

	Precision	Recall	F-measure
positive	0.86	0.79	0.82
negative	0.74	0.82	0.78

และมีค่าความถูกต้อง = 0.804

5.3 การทดสอบการจำแนกความคิดเห็นด้วย Neural Network

การทดสอบการจำแนกความคิดเห็นด้วย Neural Network กับประโยคทั้งหมด 999 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 555 ประโยค และความคิดเห็นเชิงลบ 444 ประโยค โดยแบ่งข้อมูลออกเป็น 2 ส่วน ส่วนแรก 600 ประโยค ใช้เป็นข้อมูลเข้าสำหรับให้ระบบเรียนรู้ ส่วนที่สอง 250 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 145 ประโยค และความคิดเห็นเชิงลบ 105 ประโยค ใช้สำหรับทดสอบระบบ และพิจารณาค่าความถูกต้อง ค่าความแม่นยำ ค่าการเรียกคืน และค่าวัดประสิทธิภาพว่ามีมากน้อยเพียงใด โดยได้ผลการทดสอบดังนี้

ตารางที่ 5.7 confusion matrix ของการจำแนกความคิดเห็นด้วย Neural Network

		predicted	
		positive	negative
actual	positive	142	3
	negative	18	87

ตารางที่ 5.8 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Neural Network

	Precision	Recall	F-measure
positive	0.89	0.98	0.93
negative	0.97	0.83	0.89

และมีค่าความถูกต้อง = 0.916

5.4 การทดสอบการจำแนกความคิดเห็นด้วย Logistic Regression

การทดสอบการจำแนกความคิดเห็นด้วย Logistic Regression กับประโยคทั้งหมด 999 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 555 ประโยค และความคิดเห็นเชิงลบ 444 ประโยค โดยแบ่งข้อมูลออกเป็น 2 ส่วน ส่วนแรก 749 ประโยค ใช้เป็นข้อมูลเข้าสำหรับให้ระบบเรียนรู้ ส่วนอีก 250 ประโยค แบ่งเป็นความคิดเห็นเชิงบวก 145 ประโยค และความคิดเห็นเชิงลบ 105 ประโยค ใช้สำหรับทดสอบระบบ และพิจารณาค่าความถูกต้อง ค่าความแม่นยำ ค่าการเรียกคืน และค่าวัดประสิทธิภาพว่ามีมากน้อยเพียงใด โดยได้ผลการทดสอบดังนี้

ตารางที่ 5.9 confusion matrix ของการจำแนกความคิดเห็นด้วย Logistic Regression

		predicted	
		positive	negative
actual	positive	115	30
	negative	26	79

ตารางที่ 5.10 ประสิทธิภาพของการจำแนกความคิดเห็นด้วย Logistic Regression

	Precision	Recall	F-measure
positive	0.82	0.79	0.80
negative	0.72	0.75	0.74

และมีค่าความถูกต้อง = 0.776

5.5 สรุปผลการทดสอบ

จากการทดสอบ พบว่า การจำแนกความคิดเห็นด้วย NLTK Naive Bayes ให้ผลลัพธ์ที่มีค่าความถูกต้องที่สูงกว่าการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes Support, Vector Machine, Neural network, Logistic Regression

ดังนั้นจึงสรุปว่า การจำแนกความคิดเห็นด้วย NLTK Naive Bayes นั้น เหมาะสมที่จะนำมาใช้จำแนกความคิดเห็นเกี่ยวกับร้านอาหารมากที่สุด

สำหรับความผิดพลาดในการจำแนกความคิดเห็นนั้น เป็นไปได้จากหลายสาเหตุ เช่น การฝึกตัวจำแนกไม่เหมือนกัน การแบ่งแยกข้อความเป็นประโยคทำได้ไม่ดีพอ การไม่เท่ากันของความคิดเห็นเชิงบวกและความคิดเห็นเชิงลบ จำนวนประโยคนำเข้าสำหรับให้ระบบเรียนรู้มีน้อยเกินไป และความกำกวมของคำ

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 สรุปผล

ระบบวิเคราะห์ความรู้สึกของความคิดเห็นที่มีต่อร้านอาหารบนเฟสบุ๊คนั้น ทำขั้นตอนการทำงานโดยใช้ภาษา Python เพื่อทำการดึงข้อมูล ตัดคำ และวิเคราะห์ความรู้สึกของความคิดเห็นที่มีต่อร้านอาหารว่าเป็นความคิดเห็นเชิงบวกหรือเชิงลบ

จากการทดสอบการจำแนกความคิดเห็น พบว่าการจำแนกความคิดเห็นด้วย NLTK Naive Bayes มีค่าความแม่นยำมากที่สุด ที่ 0.96 จาก นั้นเป็นการจำแนกความคิดเห็นด้วย Neural Network, Scikit-learn Naive Bayes, Support Vector Macchine, Logistic Regression ลดหลั่นกันลงมาตามลำดับ

6.2 ผลที่ได้รับ

ผลที่ได้รับจากการวิเคราะห์ข้อความรีวิวจากเฟสบุ๊ค แบ่งเป็น 2 ส่วน ได้แก่ ส่วนของผู้พัฒนาและส่วนของผู้ใช้งาน

6.2.1 ส่วนของผู้พัฒนา

1. ผู้พัฒนาได้เรียนรู้เกี่ยวกับรายละเอียดและขั้นตอนการพัฒนาระบบวิเคราะห์ความคิดเห็นที่มีต่อร้านอาหารบนเฟสบุ๊ค
2. ผู้พัฒนาได้เรียนรู้เกี่ยวกับการเขียนโปรแกรมด้วยภาษาต่าง ๆ ที่นำมาใช้พัฒนาระบบ
3. ผู้พัฒนาได้เรียนรู้การวางแผนการทำงาน ฝึกการตัดสินใจ และการทำงานเป็นทีม
4. ผู้พัฒนาได้นำความรู้ที่เรียนมาและศึกษาเพิ่มเติมมาใช้ในการประยุกต์ใช้กับโครงงาน

6.2.2 ส่วนของผู้ใช้งาน

1. ผู้ใช้สามารถนำผลลัพธ์ที่ได้จากการวิเคราะห์ข้อความรีวิวบนเฟสบุ๊คไปใช้ในการพัฒนาการให้บริการหรือแก้ไขได้

6.3 ปัญหาและอุปสรรค

1. ผู้พัฒนาใช้เวลาในการศึกษาภาษา โปรแกรม และวิธีการพัฒนาโปรแกรมเป็นเวลานาน
2. ข้อความที่นำมาจากเฟสบุ๊คมีจำนวนข้อความไม่ถึงตามที่ทางผู้พัฒนาต้องการ เนื่องจากข้อความความคิดเห็นที่เป็นรีวิว สามารถใส่ได้แค่คนละ 1 ข้อความจึงเกิดปัญหาในเรื่องจำนวนข้อความที่ต้องการตามขอบเขตที่วางไว้
3. ตัวระบบยังไม่สามารถจำแนกความคิดเห็นที่เป็นกลางได้
4. การจัดเตรียมข้อมูลต้องทำด้วยตนเอง อาจทำให้ใช้เวลานาน
5. ข้อความแสดงความคิดเห็นจากรีวิวบนเฟสบุ๊คต้องเป็นข้อความที่เป็นสาธารณะ
6. ทางผู้พัฒนาไม่สามารถสร้างเว็บแอพพลิเคชั่นที่ใช้งานสำหรับการวิเคราะห์ความรู้สึกที่มีต่อร้านอาหารได้

6.4 วิธีการแก้ปัญหา

1. เริ่มศึกษาภาษา โปรแกรม และวิธีการพัฒนาโปรแกรมให้เร็วขึ้น
2. มีวิธีแก้อยู่ 2 ทางคือ 1. ทางผู้พัฒนาต้องพัฒนาแอพพลิเคชันผ่านแพลตฟอร์มของเฟสบุ๊คให้สามารถขอเข้าถึงข้อมูลที่เป็นสาธารณะของเพจอื่นได้ 2. ใช้วิธีพัฒนาระบบด้วยข้อมูลจากแหล่งอื่น และนำระบบที่พัฒนามาทดสอบกันข้อมูลที่ดึงมาจากเพจบนเฟสบุ๊คของตนเอง
3. พัฒนาระบบให้สามารถจำแนกความคิดเห็นที่เป็นกลางได้
4. พัฒนาส่วนของระบบให้สามารถจัดเตรียมข้อมูลแบบอัตโนมัติ โดยอาจใช้วิธีคล้ายกับ regular expression คือแปลงตัวอักษรเป็น unicode แล้วลบตัวอักษรที่ไม่อยู่ในช่วง unicode ที่กำหนดออก
5. ต้องเรียนผู้การสร้างเว็บแอพพลิเคชั่นควบคู่ไปกับการศึกษาการทำงานของระบบวิเคราะห์ความรู้สึก หรือเริ่มศึกษาการสร้างเว็บแอพพลิเคชั่นให้เร็วขึ้น

เอกสารอ้างอิง

[1] Wikipedia. Sentiment analysis [Online]. Available at: http://en.wikipedia.org/wiki/Sentiment_analysis [2018, Oct 5]

[2] Monkeylearn. Sentiment analysis [Online]. Available at: https://monkeylearn.com/sentiment-analysis/ [2018, Oct 5]

[3] Peerapon Vateekul, and Thanabhat Koomsubha, “A Study of Sentiment Analysis Using Deep Learning Techniques on Thai Twitter Data”. 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)

[4] Phakhawat Sarakit, Thanaruk Theeramunkong, Choochart Haruechaiyasak, and Manabu Okumura. “Classifying emotion in Thai youtube comments”. 2015 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)

[5] นภัสสร วิภาบุษบากร และ ปุณยวีร์ คงไพศาลโสภณ. การวิเคราะห์ความรู้สึกของความคิดเห็นต่อสายการบินบนเว็บไซต์. โครงงานวิทยาศาสตร์, สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, ปีการศึกษา 2560.

[6] ณัฐริกา เมษินทรีย และ แพรพลอย ธโนศวรรย์. การวิเคราะห์อารมณ์ความรู้สึกของข้อความคิดเห็นบนทวิตเตอร์เกี่ยวกับสถานที่ท่องเที่ยวในประเทศไทย. โครงงานวิทยาศาสตร์, สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, ปีการศึกษา 2560.

[7] ชิษณุพงศ์ วรรณโพธิ์พร และ รัญชิดา ศักยโรจน์กุล. การวิเคราะห์ข้อความแสดงความคิดเห็นของผู้ใช้บริการต่อร้านอาหารนานาชาติในกรุงเทพมหานคร. โครงงานวิทยาศาสตร์, สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, ปีการศึกษา 2560

[8] DumpDataSci. อธิบาย Naive Bayes แบบภาษามนุษย์ (ใช้ในการทำ Classification) [Online]. Available at: https://medium.com/@dumpdatasci.th/อธิบาย-Naive-Bayes-แบบภาษามนุษย์-ใช้ในการทำ-Classification-2b8d35e1e764 [2018, Oct 18]

[9] Sitake. โมเดล Naive Bayes และการแปลความหมาย [Online]. Available at: http://dataminingtrend.com/2014/naive-bayes/ [2018, Oct 18]

[10] Chaiyanan. ซัพพอร์ตเวกเตอร์แมชชิน (Support Vector Machine: SVM) [Online]. Available at: https://knowledge.snru.ac.th/ซัพพอร์ตเวกเตอร์แมชชีน/ [2019, Feb 11]

[11] Mr.P L. SVM อดีตเคยหวานปัจจุบันแอบเซง : Machine Learning 101 [Online]. Available at: https://medium.com/mmp-li/svm-อดีตเคยหวานปัจจุบันแอบเซง-machine-learning-101-6008753c780c [2019, Feb 11]

[12] Saran Khotsatian. Machine Learning: ANN คืออะไร [Online]. Available at: https://medium.com/@sarankhotsathian/machine-learning-ann-คืออะไร-3527a9aa0c8c [2019, Mar 10]

[13] Engkarat Techapanurak. Bayesian Neural Network (ตอนที่ 3): อะไรคือ Deep Learning และอะไรคือ Neural Network-ฉบับมือใหม่ [Online]. Available at: https://medium.com/@dopplerz/bayesian-neural-network-ตอนที่-3-อะไรคือ-deep-learning-และอะไรคือ-neural-network-ฉบับมือใหม่-cce9227ab599 [2019, Mar 10]

[14] Mr.P L. Logistic Regression ทำนายผู้รอดชีวิตบนเรือไทเทนิค + Keras Neural Network: Machine Learning 101 [Online]. Available at: https://medium.com/mmp-li/logistic-regression-ไม่มีอะไรเป็นไปตามอย่างที่คิดเสมอ-machine-learning- 101-bba2f666 234d [2019, Mar 26]

[15] Wikipedia. Scikit-learn [Online]. Available at: https://en.wikipedia.org/wiki/Scikit-learn [2018, Oct 5]

[16] NLTK. Natural Language Toolkit - NLTK 3.4 documentation [Online]. Available at: https://www.nltk.org/ [2018, Oct 5]

[17] rkcosmos. Deepcut: A Thai word tokenization library using Deep Neural Network [Online]. Available at: https://github.com/rkcosmos/deepcut [2018, Oct 5]

[18] Wongnai. Wongnai corpus [Online]. Available at: https://github.com/wongnai/wongnai-corpus [Visited date: 2018, Oct 5]

ภาคผนวก

ภาคผนวก ก

แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal

ปีการศึกษา 2561

ชื่อโครงงาน (ภาษาไทย) การวิเคราะห์ข้อมูลความคิดเห็นของสินค้าหรือบริการที่แสดงบนเฟสบุ๊ค

ชื่อโครงงาน (ภาษาอังกฤษ) Sentiment analysis of a product or a service posted on Facebook

อาจารย์ที่ปรึกษา อาจารย์ ดร. อาธร เหลืองสดใส

ผู้ดำเนินการ 1. นาย ธนกฤต งามสุนทรวงศ์ เลขประจำตัวนิสิต 5833632223

2. นาย ภูริวัฒน์ มีชัยเจริญยิ่ง เลขประจำตัวนิสิต 5833651123

สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

หลักการและเหตุผล

ในปัจจุบันเราจะเห็นว่าประชากรส่วนหนึ่งในประเทศไทยได้เริ่มใช้โซเชียลมีเดียเพื่อแสดงความคิดเห็นเกี่ยวกับการให้บริการของร้านอาหารหรือห้างร้านเป็นจำนวนมาก ทำให้ทางร้านอาหารหรือห้างร้านจำเป็นที่จะต้องจ้างพนักงานมาเพื่ออ่านข้อความที่ลูกค้าแสดงความคิดเห็น เพื่อนำไปปรับปรุงและพัฒนาการให้บริการ ในจุดนี้ผู้พัฒนาเล็งเห็นว่าหากเราต้องอ่านความคิดเห็นทั้งหมดด้วยตัวเองนั้น อาจจะต้องใช้เวลาในการอ่านความคิดเห็นแต่ละครั้งเป็นจำนวนมาก ประกอบกับการที่ในปัจจุบันได้มีการนำเทคนิคการวิเคราะห์ข้อมูล (Data Analytic) เข้ามาใช้ในการจัดการจัดกลุ่มข้อความเป็นหมวหมู่ ซึ่งหากเราสามารถนำวิธีการนี้มาช่วยจัดการแยกแยะข้อความแสดงความคิดเห็นออกเป็นหมวดหมู่ได้ จะสามารถทำให้การพัฒนาการให้บริการของทางร้านอาหารเป็นไปได้สะดวกและรวดเร็วยิ่งขึ้น

ผู้พัฒนานำวิธีการวิเคราะห์ข้อมูลมาจัดการแยกแยะข้อความแต่ละข้อความออกเป็นหมวดหมู่ตามความเหมาะสม เช่น เป็นข้อความเชิงบวกหรือเป็นข้อความเชิงลบ เพื่อให้ง่ายต่อการนำข้อมูลไปใช้สำหรับปรับปรุงพัฒนาการบริการต่อไป

วัตถุประสงค์

เพื่อพัฒนาโปรแกรมสำหรับดึงข้อความที่เป็นการแสดงความเห็นจากหน้าเพจในเฟสบุ๊คและนำมาวิเคราะห์เพื่อจำแนกข้อความแสดงความคิดเห็นเหล่านั้นออกเป็นหมวดหมู่ตามความเหมาะสม เช่น แบ่งเป็นความคิดเห็นในแง่บวกและความคิดเห็นในแง่ลบ เพื่อให้เพื่อให้ง่ายต่อการนำไปใช้ในการให้บริการของบริษัทหรือ หน่วยงานที่ศึกษา และสามารถนำข้อมูลที่ได้ไปใช้งานในด้านอื่น ๆ โดยการนำข้อมูลไปประมวลผลเป็นค่าทางสถิติเป็นต้น

ขอบเขตของโครงงาน

1. โครงการนี้ศึกษาความคิดเห็นของผู้ใช้บริการบนเฟสบุ๊คที่เป็นภาษาไทยมาตรฐาน
2. ความคิดเห็นที่นำมาวิเคราะห์การให้บริการจากหน้าเพจ
   โดยนำความคิดเห็นมาจากช่องทางสำหรับรีวิวของเพจอย่างน้อย 50 ความคิดเห็นต่อเพจ
3. โปรแกรมสามารถใช้งานได้เมื่อเชื่อมต่ออินเทอร์เน็ตเท่านั้น
4. โปรแกรมสามารถใช้งานบนเบราเซอร์ Google Chorme โดยผลลัพธ์ของการวิเคราะห์จะถูกแยกเป็นความคิดเห็นด้านบวกและด้านลบ

วิธีการดำเนินงาน

1. ศึกษาวิธีและค้นคว้าหาข้อมูลที่เกี่ยวกับโครงงาน ได้แก่การดึงข้อมูลความคิดเห็นจากเพจบนเฟสบุ๊ค การทำเหมืองข้อมูล (Data Mining) และการวิเคราะห์ข้อมูลความพึงพอใจแบ่งเป็นด้านบวก, ด้านลบ และแยกแยะความต้องการของผู้ใช้บริการ
2. ศึกษาข้อมูลของซอฟต์แวร์และเทคนิคการดำเนินงาน
   - เทคนิคการใช้ Facebook Graph API
   - เทคนิคที่ใช้ในการวิเคราะห์ข้อมูล ได้แก่ Naive Bayes ผ่านโมดูล NLTK
3. กำหนดขอบเขต ขั้นตอนการดำเนินงาน และออกแบบโปรแกรม
4. พัฒนาโปรแกรมในส่วนการดึงข้อมูลความคิดเห็นจากเพจบนเฟสบุ๊ค
5. เก็บรวบรวมข้อมูลความคิดเห็นจากหน้าเพจบนเฟสบุ๊ค
6. พัฒนาโปรแกรมในส่วนวิเคราะห์ข้อมูล
7. ตรวจสอบความถูกต้องของข้อมูลที่ได้จากการวิเคราะห์และแก้ไขข้อผิดพลาดของโปรแกรม
8. สรุปผลการดำเนินการ และจัดทำเอกสารประกอบโครงการ

ตารางเวลาดำเนินการ

ขั้นตอนการดำเนินการ	ปี พ.ศ. 2561					ปี พ.ศ. 2562			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย
1. ศึกษาวิธีและค้นคว้าข้อมูลที่เกี่ยวกับโครงงาน และรวบรวมข้อมูลความคิดเห็นจากเฟสบุ๊ค									
2. ศึกษาข้อมูลเกี่ยวกับซอฟต์แวร์และเทคนิคการดำเนินงาน									
3. กำหนดขอบเขต ขั้นตอนการดำเนินงาน ออกแบบโปรแกรม									
4. พัฒนาส่วนการดึงข้อมูลความคิดเห็นจากเพจบนเฟสบุ๊ค									
5. รวบรวมข้อมูลความคิดเห็นจากหน้าเพจบนเฟสบุ๊ค									
6. พัฒนาส่วนวิเคราะห์ข้อมูล									
7. ตรวจสอบความถูกต้องของผลวิเคราะห์และแก้ไขข้อผิดพลาด									
8. สรุปผลการดำเนินการและจัดทำเอกสารประกอบ									

ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ต่อนิสิต

1. ได้เรียนรู้วิธีเขียนโปรแกรมโดยใช้ภาษา Python
2. ฝึกฝนการพัฒนาโปรแกรม
3. ได้เรียนรู้เกี่ยวกับวิธีการจัดการการดำเนินงาน
4. ได้ความรู้เกี่ยวกับการนำความรู้เรื่อง NLP, Data Mining และ Sentiment Analysis มาใช้ร่วมกัน

ประโยชน์ที่ได้ต่อโครงงาน

1. นำข้อมูลที่ได้จากการเก็บไปใช้ในการวางแผนงานในอนาคต
2. สามารถวิเคราะห์เสียงตอบรับของผู้ใช้จากการให้บริการ
3.ข้อมูลที่ได้สามารถนำไปใช้ประโยชน์ในการพัฒนาการให้บริการของร้านอาหารหรือห้างร้านได้

อุปกรณ์และเครื่องมือที่ใช้

ฮาร์ดแวร์

1. เครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการวินโดวส์ 10
    - Processor: Intel Core i7-6700HQ QUAD-CORE
    - Graphic card: NVIDIA GEFORCE GTX 960M (4GB GDDR5)
    - Memory: 8 GB
    - Hard disk: 1TB 9.5MM 5400RPM
2. เครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการวินโดวส์ 10
    - Processor: Intel Core i5-8250U
    - Graphic card: NVIDIA GeForce MX150 (2GB GDDR5)
    - Memory: 8 GB
    - SSD: 256GB

ซอฟแวร์

1. Python IDEs
2. NTLK version 3.3
3. Facebook Graph API

งบประมาณ

1. Solid State Drive (SSD) ความจุ 256 GB จำนวน 1 ตัว
   ราคา 2,000 บาท
2. Solid State Drive (SSD) ความจุ 512 GB จำนวน 1 ตัว
   ราคา 4,000 บาท
3. Ram Notebook ความจุ 8 GB จำนวน 2 ตัว
   ราคา 2,000 * 2 = 4,000 บาท
   ราคารวม 10,000 บาท

ภาคผนวก ข

ตัวอย่างส่วนของโปรแกรมที่ใช้ในการพัฒนาระบบ

1. ส่วนของโปรแกรมสำหรับอ่านข้อมูล

```
data_pos = []
with open('combipos.txt', 'r', encoding='utf-8') as omega:
    for line in omega:
        data_pos.append([line.strip(), 'pos'])
```

2. ส่วนของโปรแกรมสำหรับดึงข้อมูลจากเพจบนเฟสบุ๊ค

```
import facebook
import codecs


token = [Your page access token]


graph = facebook.GraphAPI(access_token=token, version="3.1")
info = graph.get_object(id='me', fields='ratings{review_text}')
rating = info['ratings']
data = rating['data']
with codecs.open("data\\rating.txt", "w", "utf-8") as outputFile:
    for review in data:
        outputFile.write(review['review_text']+'\n')
```

3. ส่วนของโปรแกรมสำหรับแบ่งข้อมูลเพื่อ train, test

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    data_tokens,
    data_labels,
    test_size=0.25,
    random_state=1412)
```

4. ส่วนของโปรแกรมการจำแนกความคิดเห็นด้วย NLTK Naive Bayes

```
word_features = get_word_features(get_words_in_reviews(features_data.tolist()))
train_features = nltk.classify.apply_features(extract_features, features_data.tolist())
classifier = nltk.NaiveBayesClassifier.train(train_features

Xy_test = []
for z in range(len(y_test)):
    Xy_test.append([X_test[z],y_test[z]])

features_data2 = numpy.array(Xy_test)
refset2 = []
testset2 = []

out_features = nltk.classify.apply_features(extract_features, features_data2.tolist())
for i, (feats, label) in enumerate(out_features):
    observed = classifier.classify(feats)
    #print(observed)

    refset2.append(label)
    testset2.append(observed)
```

```
print('Accuracy = ',accuracy_score(refset2,testset2))
print(classification_report(refset2,testset2))
print(confusion_matrix(refset2,testset2,labels=['pos','neg']))
```

5. ส่วนของโปรแกรมการจำแนกความคิดเห็นด้วย Support Vector Machine

```
from sklearn import svm
svm = svm.SVC(kernel='linear',C=1.0,gamma=0.1)
svm.fit(onehot_.transform(X_train), y_train)

print('Accuracy = ',accuracy_score(y_test,svm.predict(onehot_.transform(X_test))))
print(classification_report(y_test,svm.predict(onehot_.transform(X_test))))
print(confusion_matrix(y_test,svm.predict(onehot_.transform(X_test)),labels=['pos','neg']))
```

6. ส่วนของโปรแกรมการจำแนกความคิดเห็นด้วย Scikit-learn Naive Bayes

```
from sklearn.naive_bayes import BernoulliNB
bnbc = BernoulliNB(binarize=None)
bnbc.fit(onehot_.transform(X_train), y_train)

print('Accuracy = ',accuracy_score(y_test,bnbc.predict(onehot_.transform(X_test))))
print(classification_report(y_test,bnbc.predict(onehot_.transform(X_test))))
print(confusion_matrix(y_test,bnbc.predict(onehot_.transform(X_test)),labels=['pos','neg']))
```

7. ส่วนของโปรแกรมการจำแนกความคิดเห็นด้วย Neural Network

```
import tensorflow as tf
tf.reset_default_graph()
```

```
vocab_len = len(onehot_nn.classes_)
inputs_ = tf.placeholder(dtype=tf.float32, shape=[None, vocab_len], name="inputs")
targets_ = tf.placeholder(dtype=tf.float32, shape=[None, 2], name="targets")

h1 = tf.layers.dense(inputs_, 500, activation=tf.nn.relu)
#h2 = tf.layers.dense(h1, 500, activation=tf.nn.relu)
#h3 = tf.layers.dense(h2, 500, activation=tf.nn.relu)
logits = tf.layers.dense(h1, 2, activation=tf.nn.relu)
output = tf.nn.sigmoid(logits)
loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=logits,
labels=targets_))
optimizer = tf.train.AdamOptimizer(0.001).minimize(loss)
correct_pred = tf.equal(tf.argmax(logits, 1), tf.argmax(targets_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32), name='accuracy')

test_result = sess.run(output,feed_dict={
    inputs_:onehot_nn.transform(X_test)
})
i=0
nn_pred = []
for stuff in test_result:
    if test_result[i][0]>test_result[i][1]:
        nn_pred.append('pos')
    else:
        nn_pred.append('neg')
    i=i+1

print('Accuracy = ',accuracy_score(y_test, nn_pred))
print(classification_report(y_test, nn_pred))
print(confusion_matrix(y_test, nn_pred,labels=['pos','neg']))
```

8. ส่วนของโปรแกรมการจำแนกความคิดเห็นด้วย Logistic Regression

```
from sklearn.linear_model import LogisticRegression
log_model = LogisticRegression()
log_model = log_model.fit(X=lrx_train, y=lry_train)

lry_pred = log_model.predict(lrx_test)
print(accuracy_score(lry_test, lry_pred))

print('Accuracy = ',accuracy_score(lry_test, lry_pred))
print(classification_report(lry_test, lry_pred))
print(confusion_matrix(lry_test, lry_pred,labels=['pos','neg']))
```

ประวัติผู้เขียน

Mr. Thanakrit Ngamsoonthornwong

นาย ธนกฤต งามสุนทรวงศ์

วัน เดือน ปี เกิด: 11 กรกฎาคม 2540

อีเมลล์: mosza1107@gmail.com

การศึกษา: คณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

Mr. Pooriwat Meechaicharoenying

นาย ภูริวัฒน์ มีชัยเจริญยิ่ง

วัน เดือน ปี เกิด: 14 กรกฎาคม 2538

อีเมลล์: Pooriwat1995@gmail.com

การศึกษา: คณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย