



การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์ข่าวตลาด forex gold spot

An Application of Natural Language Processing on forex gold spot News Analysis

นายชวัลชัย อภิชาตรูติวรณ์

6210110646

โครงการวิศวกรรมคอมพิวเตอร์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

2565



การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์ข่าวตลาด forex gold spot

An Application of Natural Language Processing on forex gold spot News Analysis

นายชวัลชัย อภิชาตรูติวรณ์

6210110646

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิគวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

ชื่อโครงการ การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์ข่าว
ตลาด forex gold spot

ผู้จัดทำ นายชวลชัย อภิชาตธิติวรรณ รหัส 6210110646

สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2565

อาจารย์ที่ปรึกษาโครงการ

คณะกรรมการสอบ

(รศ.ดร. มนตรี กาญจนะเดชะ)

(ผศ.ดร. วัชรินทร์ แก้วอภิชัย)

(ผศ.ดร. ธเนศ เคารพาพงศ์)

โครงการนี้เป็นส่วนหนึ่งของรายวิชาโครงการวิศวกรรมคอมพิวเตอร์ 1 และ 2
ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

(รศ.ดร. พิชญา ตันทัยย์)

หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์

หนังสือรับรองความเป็นเอกสารลักษณ์

ข้าพเจ้าผู้ลงนามท้ายนี้ ขอรับรองว่ารายงานฉบับนี้เป็นรายงานที่มีความเป็นเอกสารลักษณ์โดยที่ข้าพเจ้ามิได้การคัดลอกมาจากการที่ได้เนื้อหาในรายงานทั้งหมดถูกรวบรวมจากการพัฒนาในขั้นตอนต่างๆ ของการจัดทำโครงการ หากมีส่วนหนึ่งส่วนใดที่จำเป็นต้องนำข้อความจากผลงานของบุคคลอื่นใดที่ไม่ใช่ตัวข้าพเจ้า ข้าพเจ้าได้ทำการอ้างอิงถึงเอกสารเหล่านั้นไว้อย่างเหมาะสม และขอรับรองว่ารายงานฉบับนี้ไม่เคยเสนอต่อสถาบันใดมาก่อน

ผู้จัดทำ

(นายชวัลชัย อภิชาติ^{ธุ}ติวรณ์)

ชื่อโครงการ การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์
ข่าวตลาด forex gold spot

ผู้จัดทำ นายชวพลชัย อภิชาตธิติวรรณ รหัส 6210110646

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ปีการศึกษา 2565

บทคัดย่อ

การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์ข่าวตลาด forex gold spot ในโครงการนี้เราดำเนินการสร้าง ML NLP และ การทำ Model Naivebaye เพื่อการวิเคราะห์ข่าวสาร และข้อมูลต่างๆที่เกี่ยวกับ Forex Gold sport และ เราได้ทำการเปรียบเทียบให้เห็นถึงประสิทธิภาพ ของNLP และความรู้เบื้องต้นที่จะได้รับจากโครงการนี้คือ ภาษา Python ซึ่งเป็นภาษายอดนิยมในปัจจุบัน

คำสำคัญ NLP, ML, Forex

Project	An Application of Natural Language Processing on forex gold spot News Analysis
Author	Mr.Chualchai Apichatthitiworn ID 6210110646
Major Program	Computer Engineering
Academic Year	2020

Abstract

Above the natural language banners to provide forex gold spot market news, this will help you to create ML NLP and don't forget to use Model Naivebaye for sandy, news for Forex Gold sport and we collect various information. In order to reach the effectiveness of NLP and basic knowledge that will be gained here. which is the Python language itself, which is what needs to be done nowadays.

Keywords: NLP, ML, Forex

กิตติกรรมประกาศ

ข้าพเจ้าผู้จัดทำโครงการนวัตกรรม เรื่อง การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์ข่าวตลาด forex gold spot ขอขอบพระคุณบุคลากรทุกท่าน ที่ได้ให้คำปรึกษาและชี้แนะแนวทางการทำโครงการ ทั้งในด้านวิชาการและการดำเนินโครงการ ดังนี้

รองศาสตราจารย์ ดร.มนตรี กัญจนะเดชะ อาจารย์ที่ปรึกษาโครงการหลัก ซึ่งเคยให้คำปรึกษาชี้แนะแนวทางในการทำงาน และแก้ไขปัญหาและค่อยสนับสนุนการทำโครงการ ของข้าพเจ้า ไปจนถึงการตรวจสอบรายงานเพื่อให้ผลสำเร็จ

และขอบคุณบุคลากรของภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ ที่อย่างมุ่งมั่นพยายามสุดๆ ในด้านต่างๆ และปรึกษาการทำโครงการ ตลอดระยะเวลาในการทำโครงการ

ชื่อผู้จัดทำ

ชวลอชัย อภิชาตธิวัณ

สารบัญ

บทคัดย่อ.....	๑
Abstract.....	๒
กิตติกรรมประกาศ	๓
สารบัญ.....	๔
รายการภาพประกอบ	๕
รายการตาราง	๖
รายการคำย่อ	๗
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตโครงการ.....	2
1.4 แผนการดำเนินงาน	3
บทที่ 2 ทฤษฎีและความรู้พื้นฐาน.....	5
2.1 Forex gold spot.....	5
2.1.1 ปัจจัยที่ส่งผลกระทบต่อราคาทองคำ	5
2.2 Natural Language Processing NLP	10
2.2.1 How does natural language processing work.....	10
2.2.2 เหตุใดการประมวลผลภาษาธรรมชาติจึงมีความสำคัญ	11
2.2.3 เทคนิคและวิธีการประมวลผลภาษาธรรมชาติ.....	12
2.3 Machine Learning	15
2.4 Extracting features from text	16
2.4.1 Count-based strategies.....	16
2.4.2 Advanced strategies	18
2.5 Supervised learning on text.....	19
2.5.1 Supervised learning	19
2.6 Naive Bayes Classification	21
2.7 PyWeblO	22
2.7.1 คุณสมบัติของPyWeblO	22

บทที่ 3 รายละเอียดการดำเนินงาน.....	23
3.1 ขั้นตอนและวิธีการดำเนินงาน	23
3.1.1 ขั้นตอนการทำงานของระบบ	23
3.1.2 ขั้นตอน INPUT (แหล่งข้อมูลข่าวสาร)	23
3.1.3 ขั้นตอน Processing	23
3.1.4 ขั้นตอน Output.....	24
3.2 ขั้นตอนการดำเนินงาน	24
บทที่ 4 ความก้าวหน้าการดำเนินงาน.....	25
4.1 ความก้าวหน้า 1 ศึกษา Library Python NLP และทดสอบ	25
4.1.1 Library Python NLP	25
4.1.2 รายละเอียดการทดลอง	26
4.2 ความก้าวหน้า 2 ทดสอบ Library NLTK.....	27
4.2.1 รายละเอียดการทดลอง	27
4.3 ความก้าวหน้า 3 ทดสอบ Library NLTK.....	32
4.3.1 รายละเอียดการทดลอง	32
4.4 ความก้าวหน้า 4 สร้างเมืองข้อมูล และทดสอบ	37
4.5 ความก้าวหน้า 5 ทดสอบ Model <i>Training and Test datasets</i>	38
4.5.1 Step 1 - Loading the Required Libraries and Modules	38
4.5.2 Step 2 - Loading the Data and Performing Basic Data Checks.....	38
4.5.3 Step 3 – Pre-processing the Raw Text and Getting It Ready for Machine Learning	39
4.5.4 Step 4 - Creating the Training and Test Datasets	41
4.5.5 Step 5 - Converting Text to Word Frequency Vectors with TfidfVectorizer.	42
4.5.6 Step 6 - Create and Fit the Classifier.....	43
4.5.7 Step 7 - Computing the Evaluation Metrics	44
4.6 Building Random Forest Classifier	44
4.7 ความก้าวหน้า 6 ทดลองทำ Sentiment Analysis Naïve Bayes Classifier	45
4.8 ความก้าวหน้า 7 ปรับแก้ไข Sentiment Analysis Naïve Bayes Classifier.....	49
4.8.1 การเปรียบเทียบ โดยใช้ Model Naïve baye	49
4.8.2 การเปรียบเทียบโดยใช้ข่าวสารที่ไม่ได้รับการ tren โดยใช้ Model Naïve baye	50

4.9	ความก้าวหน้า 8 pywebio.....	51
บทที่ 5 สรุป.....		53
5.1	สรุปผลการดำเนินงาน.....	53
5.2	ปัญหาและอุปสรรค	53
5.3	งานที่จะดำเนินการต่อไป.....	54
บรรณานุกรม.....		55

รายการภาพประกอบ

ภาพที่ 1 เป็นการทำ Tokenization [6]	12
ภาพที่ 2 รูปแบบของการใช้งาน Stemming และ lemmatization [6]	13
ภาพที่ 3 ตัวอย่างการติดแทรกจัดกลุ่มข้อความ POS-taggers [6].....	14
ภาพที่ 4 การติดแทรกที่อ้างอิงถึงวัตถุเฉพาะ Named entity [6]	14
ภาพที่ 5 ความแตกต่างระหว่างการเรียนรู้แบบมีผู้ดูแล และ ไม่มีผู้ดูแล [6].....	15
ภาพที่ 6 เป็นขั้นตอนการเคลียร์ข้อความ ตัดอักขระออก Sentence [6]	16
ภาพที่ 7 เป็นสร้างโครงคำศัพท์เพื่อแยกคำศัพท์ตัวใหม่มีการใช้บ่อຍ หลักการของ BoW [6]	16
ภาพที่ 8 เป็นการจัดความสำคัญของคำเทียบกับความยาวประโยค [6].....	17
ภาพที่ 9 แสดงถึงความสำคัญของแต่ละประโยค [6].....	17
ภาพที่ 10 เป็นการแสดงบริบทที่คล้ายกันจัดอยู่ในกลุ่มเดียวกัน [6].....	18
ภาพที่ 11 workflow supervised [6]	19
ภาพที่ 12 เป็นการแสดงขั้นตอนการทำงานของ NLP	23
ภาพที่ 13 ข่าวสารตั้งเดิม	49
ภาพที่ 14 ข่าวสาร ผ่านกระบวนการNLP	50
ภาพที่ 15 ข่าวสารตั้งเดิม	50
ภาพที่ 16 ข่าวสาร ผ่านกระบวนการNLP	51

รายการตาราง

ตารางที่ 1	4
ตารางที่ 2 แหล่งข่าวสาร	9
ตารางที่ 3	30

รายการคำย่อ

MS Microsoft

LO LibreOffice

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันสังคมมีการลงทุนในส่วนของหุ้น เทรดหุ้น forex หรือ cryptocurrency มาขึ้นและมีอาชีพเกิดขึ้นมากมายในวงการของการเล่นหุ้น หรือ Stock และมีการใช้ AI ต่าง ๆ เพื่อวิเคราะห์หัวแนวโน้ม หรือคาดการณ์ล่วงหน้าของกราฟหุ้น ซึ่งบางครั้งย่อมเกิดปัญหา AI วิเคราะห์ได้แค่ทฤษฎีของกราฟและเครื่องมือต่าง ๆ ที่ใช้กัน และการวิเคราะห์ข่าวนั้นย่อมคาดการณ์ได้ยาก เพราะข่าวสารที่มามากมายย่อมส่งผลกระทบต่อหุ้นนั้น ๆ

ดังนั้นการลงทุนในส่วนของหุ้น เトレดหุ้น หรือ cryptocurrency ควรต้องมีการใช้เครื่องมือวิเคราะห์ข่าวเข้ามาเป็นส่วนหนึ่งของการคาดการณ์จากการเกร็งกำไร หรือ ลงทุน ซึ่งเห็นได้ชัดว่า AI ที่มีอยู่แล้ว เช่น EA เป็นตัวช่วยให้เราไม่ต้องมาเทรดเองโดยเป็นการเซทค่าจาก เครื่องมือต่าง ๆ ตามเทคนิคของเราว่า ซึ่งในบางครั้งเรื่องของเทคนิคก็ผิดพลาด เพราะบางหลักทรัพย์มีความผันผวนสูง ทำให้เทคนิคที่เราใช้ไว้อาจเกิดการผิดพลาดได้สูง และส่งผลให้เราขาดทุน

ในการทำ AI วิเคราะห์ข่าวหุ้นได้มีการนำ Machine Learning NLP มาวิเคราะห์ ซึ่งเป็นตัวช่วยทำให้คอมพิวเตอร์วิเคราะห์ข่าวสารหรือข้อความได้อย่างง่ายดาย NLP เป็นสาขานึงในการเรียนรู้ของเครื่องด้วยความสามารถของคอมพิวเตอร์ในการทำความเข้าใจ วิเคราะห์ จัดการ และสร้างภาษาอนุษฐานได้ ในปัจจุบันเทคโนโลยี Machine Learning NLP เป็นที่นิยมในการนำมาทำ AI เช่น การถึงข้อมูล การแปลภาษา การทำให้ข้อความง่ายขึ้น การวิเคราะห์ให้ความรู้สึกของผู้ใช้ การสรุปข้อความ ตัวกรองสแปม คาดการณ์ผลการค้นหาของผู้ใช้ แก้ไขข้ามิติ ตัวต่อตัว เป็นต้น Natural Language Processing(NLP) for Machine Learning หรือการประมวลผลภาษาธรรมชาติด้วย Python ซึ่งภาษา Python เป็นภาษาที่รวดเร็วและในการทำ NLP จะใช้ Natural Language Toolkit (NLTK) เป็น Library Opensource ยอดนิยมใน Python

1.2 วัตถุประสงค์ของโครงงาน

1. พัฒนาระบบที่ใช้หลักการของ NLP
2. วิเคราะห์ข่าวต่าง ๆ ที่เกี่ยวกับตลาด forex gold spot ซึ่งเป็นข่าวที่อยู่ในรูปแบบออนไลน์ เพื่อใช้เป็นข้อมูลสำหรับระบบ Robot Trader

1.3 ขอบเขตโครงงาน

1. สร้าง Machine Learning AI วิเคราะห์ข่าว โดยใช้ Language Processing(NLP)
2. เดพาช forex gold spot
3. วิเคราะห์แนวโน้มขึ้นหรือลง แสดงเป็น 1 0 และ -1
4. วิเคราะห์เฉพาะข่าวสำคัญที่อยู่ในตารางปฏิทิน
5. วิเคราะห์เฉพาะตลาดอเมริกา ช่วงเวลา 7.00 pm – 3.00 am ตามเวลาประเทศไทย
6. จัดทำให้แสดงข้อมูลบนเว็บไซต์

1.4 แผนการดำเนินงาน

เวลา	PROJECT 1					PROJECT 2			
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
กิจกรรม	65	65	65	65	65	65	66	66	66
(1) พัฒนา อัลกอริทึม									
(2) พัฒนา เขียนโปรแกรม ด้วย Python									
(3) เทคนิค กล									
(4) ทดสอบ ระบบ									

ตารางที่ 1

บทที่ 2 ทฤษฎีและความรู้พื้นฐาน

2.1 Forex gold spot

Forex คือ ตลาดแลกเปลี่ยนเงินตราต่างประเทศ (หรือที่เรียกว่า forex หรือ FX) หมายถึงตลาดที่ซื้อขายกันโดยตรง (OTC) ระดับโลกซึ่งเทรดเดอร์ นักลงทุน สถาบัน และธนาคารจะแลกเปลี่ยน เก็งกำไร ซื้อและขายสกุลเงินของโลกการเทรดจะทำขึ้นใน ‘ตลาดระหว่างธนาคาร’ ซึ่งเป็นช่องทางทางออนไลน์ที่มีการเทรดสกุลเงิน 24 ชั่วโมงต่อวัน ห้าวันต่อสัปดาห์ Forex เป็นหนึ่งในตลาดการเทรดที่ใหญ่ที่สุดโดยมีเงินหมุนเวียนทั่วโลกในแต่ละวันโดยประมาณมากกว่า 5 ล้านล้านดอลลาร์สหรัฐฯ

Gold Spot คือตลาดสากลในการซื้อขายทองคำทั่วโลก เป็นตลาดที่มี Volume สูงมาก เพราะเป็นการซื้อขายทองคำจากทั่วโลก มักเรียกกันในอีกชื่อหนึ่งว่า “การเทรดทองคำในตลาดโลก” โดยจะเป็นการซื้อขายในรูปแบบสัญญาหรือใบรับประกัน ไม่ได้มีการจัดส่งทองคำแท่งให้ผู้ซื้อ

2.1.1 ปัจจัยที่ส่งผลกระทบต่อราคาทองคำ

ในปัจจุบัน ราคาทองคำมีความผันผวนค่อนข้างต่ำในระยะยาว จึงเป็นสินทรัพย์ที่ปลอดภัยถึงแม้ว่าทองคำจะเป็นสินทรัพย์ที่ปลอดภัย แต่ก็จะมีความเสี่ยงและมีปัจจัยที่ต้องพิจารณาด้วย เช่นปัจจัยที่จะมีผลต่อทิศทางการเคลื่อนไหวของราคาทองคำและส่งผลต่อกำไรที่นักลงทุนจะได้รับ ในปัจจุบันทองคำยังได้รับความนิยมอยู่ เนื่องจากเป็นสินทรัพย์ที่มีความสามารถในการป้องกันความเสี่ยงในรูปแบบต่าง ๆ ได้ เช่น ความเสี่ยงจากการเงินเพื่อ ความผันผวนของอัตราแลกเปลี่ยนเงินตราต่างประเทศ ภาวะเศรษฐกิจทั่วไป จนถึงการเปลี่ยนแปลงทางการเมือง เพราะทองคำเป็นสิ่งที่มีคุณค่าในตัวเองอยู่ตลอดเวลา จึงทำให้การลงทุนในทองคำสามารถกระจายความเสี่ยงของพอร์ตการลงทุน และยังนำไปใช้สร้างผลกำไรหากจับจังหวะซื้อขายได้ถูกทางอย่างไรก็ตาม ทองคำมีความเสี่ยงจึงต้องทำการวิเคราะห์ถึงปัจจัยที่มีผลต่อทิศทางราคาทองคำ โดยหลัก ๆ แล้ว ควรพิจารณาปัจจัยใน 3 สัญญาณ ได้แก่ สัญญาณระยะยาว สัญญาณระยะกลาง และสัญญาณระยะสั้น และอีกหนึ่งอย่างที่สำคัญคือข่าวสาร

2.1.1.1 สัญญาณระยะยาว

นักลงทุนควรพิจารณาราคาทองคำย้อนหลังในอดีตไปประมาณ 7-10 ปี เพื่อเห็นภาพทิศทางราคาทองคำที่แม่นยำมากขึ้น ตัวอย่างเช่น สถิติราคาทองคำ 7 ปีย้อนหลัง ตั้งแต่ปี 2558 จนถึงต้นปี 2564 พบร่ว่าราคาทองคำ (Gold Spot) มีระดับต่ำสุดของแต่ละปีสูงขึ้นเรื่อย ๆ (ภาษาในตลาดทองคำเรียกว่า การยก

ฐานราคาในระดับต่ำสุดขึ้น) ซึ่งเหตุการณ์นี้ทำให้นักวิเคราะห์มองคำหัวโลกประเมินว่าทิศทางราคายังเป็นขาขึ้นในระยะยาวค่อนข้างชัดเจน

โดยสภาพของคำโลก ได้อธิบายถึงปัจจัยที่ทำให้ทิศทางราคายังคงเป็นขาขึ้นต่อไปนั้นคือ ในช่วงวิกฤติ COVID-19 ที่ผ่านมา ทองคำเป็นสินทรัพย์ที่มีราคาผันผวนน้อย ขณะเดียวกันก็ให้ผลตอบแทนค่อนข้างสม่ำเสมอ ประกอบกับกลุ่มยังคงมองว่าการลงทุนในสินทรัพย์อื่นยังคงมีความเสี่ยงสูง ขณะที่ อัตราดอกเบี้ยหัวโลกยังคงอยู่ในระดับต่ำและเศรษฐกิจโลกอยู่ในภาวะชะลอตัว จึงทำให้กระแสเงินลงทุนไหลเข้าสู่ตลาดทองคำอย่างต่อเนื่อง โดยเฉพาะความต้องการจากผู้บริโภคชาวจีนและอินเดีย ซึ่งล้วนแล้วแต่เป็นปัจจัยบวกต่อทิศทางราคายังคงคำหัวที่สิ้น

สำหรับนักลงทุนที่กำลังตัดสินใจลงทุนหรือว่าถือทองคำอยู่แล้วและเน้นลงทุนระยะยาว ยังสามารถลงทุนและถือต่อไปได้ โดยพฤติกรรมการลงทุนทองคำในระยะยาวจะลงทุนตั้งแต่ 6 เดือนขึ้นไป (มากกว่า 2 ไตรมาส) หรืออาจถือข้ามปี ซึ่งกลยุทธ์ในการลงทุน ก็คือ ทยอยซื้อในช่วงต้นปีหรือช่วงครุฑ์จีน จนกันนั้นให้ถือและรอจังหวะทยอยขายในช่วงปลายไตรมาส 3 หรือ ก่อนสิ้นปี นอกจากนี้ ยังมีกลุ่มนักลงทุนทองคำที่ทยอยลงทุนไปเรื่อย ๆ และถือเป็นระยะเวลาหลายปีหรือสะสมเพื่อเป็นมรดก เพราะเชื่อว่าการถือทองคำเกิน 10 ปี จะมีแต่กำไร

2.1.1.2 สัญญาณระยะปานกลาง

นักลงทุนควรพิจารณาคาดการณ์ของปี ในช่วงไตรมาส 3 และคาดการณ์ของปี ในช่วงไตรมาส 4 และไตรมาส 1 ของปีถัดไป เช่น ราคาปรับขึ้นไปที่ระดับ 2,075 เหรียญสหรัฐต่อออนซ์ในช่วงเดือนสิงหาคม ปี 2563 หลังจากนั้นราคาเริ่มอ่อนตัวลง และล่าสุดไตรมาส 1 ปี 2564 ราคาทองคำอ่อนตัวลงสู่ระดับ 1,767 เหรียญสหรัฐต่อออนซ์ และปรับลดลงสู่ระดับต่ำสุดที่บริเวณ 1,676 เหรียญสหรัฐต่อออนซ์ และเมื่อเข้าสู่ไตรมาส 2 ราคาอาจจะมีสัญญาณค่อย ๆ ปรับขึ้น โดยประเมินว่าในไตรมาส 3 ปีนี้ ราคาทองคำอาจจะปรับขึ้นไปที่ระดับสูงสุดของปีใหม่อีกดีดี

หากมองจากปัจจัยพื้นฐานจะพบว่า ช่วงต้นปีนักลงทุนเริ่มคาดการณ์ว่าเศรษฐกิจโลกจะฟื้นตัวอย่างชัดเจนหลังจากวัคซีน COVID-19 มีประสิทธิภาพ จึงเห็นการปรับประมาณการการเติบโตทางเศรษฐกิจ ถือเป็นปัจจัยกดดันให้ราคายังคงค้ำ ซึ่งเป็นสินทรัพย์ปลอดภัย (Safe Haven) อ่อนตัวลง

อย่างไรก็ตาม หลังจากการแพร่ระบาด COVID-19 รอบล่าสุด นักลงทุนประเมินว่าเศรษฐกิจโลกในปีนี้จะฟื้นตัวในลักษณะค่อยเป็นค่อยไป ประกอบกับ มาตรการผ่อนคลายทางการเงินของธนาคารกลางต่าง ๆ ทั่วโลก จะไม่เปลี่ยนแปลงในช่วงครึ่งปีหลัง จึงเริ่มเห็นการเปลี่ยนทิศทางของราคายังคงค้ำที่มีสัญญาณฟื้นตัวขึ้น จากแนวโน้มที่จะยังคงมีเม็ดเงินถูกอัดฉีดเข้ามาเพื่อกระตุนเศรษฐกิจ ส่งผลผลักดันให้ราคายังคงค้ำปรับตัวขึ้น

สำหรับนักลงทุนท่องค้าในระยะปานกลางจะเน้นลงทุนเป็นรายเดือนและไม่เกิน 3 เดือน (1 ไตรมาส) โดยพยายามจับจังหวะการแก้วงตัวของราคายังคงค้ำเพื่อหาจังหวะซื้อและขายเพื่อทำกำไร ซึ่งกลยุทธ์ในการลงทุน ก็คือ รอจังหวะลงทุนเมื่อเห็นราคายังคงค้ำอ่อนตัวลง และรอขายทำกำไรเมื่อราคามีปรับขึ้น

2.1.1.3 สัญญาณระยะสั้น

นักลงทุนจะวิเคราะห์ราคายังคงค้ำเป็นรายวันด้วยการพิจารณาราคасินทรัพย์อื่น ๆ ประกอบ เพื่อดูทองคำกับสินทรัพย์อื่น ๆ ว่ามีความเคลื่อนไหวด้านราคาอย่างไร โดยจะพิจารณาใน 2 ปัจจัย ได้แก่ ปัจจัยที่ส่งผลไปในทิศทางตรงข้ามกับราคายังคงค้ำ และปัจจัยที่ส่งผลไปในทิศทางเดียวกันกับราคา ทองคำ เช่น ตลาดหุ้นสหรัฐอเมริกา อัตราผลตอบแทนพันธบัตรรัฐบาล และค่าเงินสกุลดอลลาร์สหรัฐ

2.1.1.3.1 ตลาดหุ้นสหรัฐอเมริกา

หากตลาดหุ้นสหรัฐฯ ปรับขึ้น ราคายังคงค้ำมีแนวโน้มปรับตัวลดลง
เนื่องจากตลาดหุ้นเป็นสินทรัพย์เสี่ยง ขณะที่ทองคำเป็นสินทรัพย์ปลอดภัย ราคางานเคลื่อนไหวในทิศทางตรงกันข้าม

2.1.1.3.2 อัตราผลตอบแทนพันธบัตรรัฐบาล

จะพิจารณาจากอัตราผลตอบแทนพันธบัตรรัฐบาลสหรัฐอเมริกา อายุ 10 ปี ซึ่งถือเป็นปัจจัยสำคัญที่สะท้อนภาวะเศรษฐกิจและทิศทางอัตราดอกเบี้ยในตลาดเงินและตลาดทุน

ของสหรัฐฯ หากอัตราผลตอบแทนพันธบัตรรัฐบาลสหรัฐอเมริกาอายุ 10 ปีปรับขึ้น ส่วนใหญ่ค่าเงินดอลลาร์จะแข็งค่าขึ้นตามไปด้วย รวมทั้งนักลงทุนประเมินว่าเศรษฐกิจสหรัฐอเมริกาจะเติบโตและอัตราดอกเบี้ยมีแนวโน้มเป็นขาขึ้น ก็จะส่งผลให้ราคาทองคำปรับลดลง เนื่องจากทองคำไม่มีผลตอบแทนอยู่ในรูปของดอกเบี้ย ดังนั้น เมื่ออัตราดอกเบี้ยปรับตัวขึ้น การลงทุนในทองคำจึงถูกลดความน่าสนใจลง

2.1.1.3.3 ค่าเงินสกุลดอลลาร์สหรัฐ

จะมีความสัมพันธ์ในเชิงลบกับราคากองคำโลก กล่าวคือ ถ้าค่าเงินดอลลาร์สหรัฐอ่อนลง เมื่อเทียบกับเงินสกุลสำคัญของโลก เช่น เงินยูโร เงินเยน หรือพิจารณาจาก US Dollar Index ก็ได้เช่นกัน ราคากองคำโลกจะสูงขึ้น เพราะราคากองคำซื้อขายเป็นสกุลเงินดอลลาร์สหรัฐ เมื่อค่าเงินดอลลาร์อ่อนลง ทองคำจะมีราคาถูกลง เมื่อเทียบกับเงินสกุลอื่นที่นักลงทุนถือไว้ จึงสร้างแรงซื้อเข้ามาดันให้ราคาทองคำปรับตัวเพิ่มสูงขึ้น

2.1.1.4 ข่าวสาร

ข่าวสารเป็นสิ่งสำคัญอีกอย่างที่ราคาทองคำจะมีความผันผวนเนื่องจากข่าวสารจะมีทั้งข่าวดีที่ส่งผลดีกับทองคำแล้ว แต่ก็มีข่าวสารที่ไม่ได้ส่งผลต่อทองคำในทิศทางลงส่วนใหญ่ข่าวสารที่กระทบถึงทองคำ เช่น ข่าวสารเกี่ยวกับน้ำมัน ข่าวสารเกี่ยวกับเศรษฐกิจ อัตราการว่างงาน หรือโรงงานต่าง ภาวะเงินเฟ้อ หรือ เงินฟื้ด และข่าวสารที่สำคัญมากสำหรับทองคำจะเป็นข่าวสารจากธนาคารโลก หรือ ธนาคารกลางต่าง ๆ

ข่าวสาร	ข่าวประเภท	อ้างอิงค์
1. Census Bureau	กระทรวงพาณิชย์สหรัฐ	[7]
2. Us Department of labo	กระทรวงแรงงาน	[8]
3. Energy information Administration	ข้อมูลด้านพลังงาน	[9]
4. กองทุน SPDR	การซื้อขายทองคำของ กองทุน SPDR	[10]
5. Federal Reserve	ธนาคารกลางสหรัฐ	[11]
6. Bloomberg	รายงานข่าวทั่วไปรอบโลก	[12]
7. Twitter	ข่าวทั่วไป	[13]

ตารางที่ 2 แหล่งข่าวสาร

ในปัจจุบัน เทคโนโลยี Machine Learning NLP เป็นที่นิยมในการนำมาทำ AI เช่น การดึงข้อมูล การแปลภาษา การทำให้ข้อความง่ายขึ้น การวิเคราะห์ให้ความรู้สึกของผู้ใช้ การสรุปข้อความ ตัวกรองสแปม คาดการณ์ผลการค้นหาของผู้ใช้ แก้ไขข้าพิดอัตโนมัติ เป็นต้น Natural Language Processing(NLP) for Machine Learning หรือการประมวลผลภาษาธรรมชาติด้วย Python ซึ่งภาษา Python เป็นภาษาที่รวดเร็วและในการทำ NLP จะใช้ Natural Language Toolkit (NLTK) เป็น Library Opensource ยอดนิยมใน Python ซึ่งมีรายละเอียดดังนี้

2.2 Natural Language Processing NLP

การประมวลผลภาษาธรรมชาติ (NLP) คือความสามารถของโปรแกรมคอมพิวเตอร์ในการทำความเข้าใจภาษาตามนุ悔ยในขณะที่พูดและเขียน ซึ่งเรียกว่าภาษาธรรมชาติ เป็นส่วนประกอบของปัญญาประดิษฐ์

2.2.1 How does natural language processing work

NLP ช่วยให้คอมพิวเตอร์เข้าใจภาษาธรรมชาติเหมือนที่มนุษย์เข้าใจ ไม่ว่าจะเป็นภาษาพูดหรือเขียน การประมวลผลภาษาธรรมชาติใช้ปัญญาประดิษฐ์เพื่อป้อนข้อมูลในโลกแห่งความเป็นจริง ประมวลผลและทำความเข้าใจในลักษณะที่คอมพิวเตอร์สามารถเข้าใจได้ เช่นเดียวกับที่มนุษย์มีเช่นเชอร์ที่แตกต่างกัน เช่น หูสำหรับได้ยินและตาที่มองเห็น คอมพิวเตอร์มีโปรแกรมสำหรับอ่านและไมโครโฟนเพื่อรับรวมเสียง และเช่นเดียวกับที่มนุษย์มีสมองในการประมวลผลอินพุตนั้น คอมพิวเตอร์ก็มีโปรแกรมสำหรับประมวลผลอินพุตที่เกี่ยวข้อง ในบางจุดของการประมวลผล อินพุตจะถูกแปลงเป็นโค้ดที่คอมพิวเตอร์สามารถเข้าใจได้

มีสองขั้นตอนหลักในการประมวลผลภาษาธรรมชาติ: การประมวลผล ข้อมูลล่วงหน้า และการพัฒนาอัลกอริธึม

การประมวลผลข้อมูลล่วงหน้าเกี่ยวข้องกับการเตรียมและ "การล้าง" ข้อมูลข้อความสำหรับเครื่องเพื่อให้สามารถวิเคราะห์ได้ การประมวลผลล่วงหน้าทำให้ข้อมูลอยู่ในรูปแบบที่ใช้การได้และเน้นคุณลักษณะในข้อความที่อัลกอริธึมสามารถทำงานได้ สามารถทำได้หลายวิธี ได้แก่:

Tokenization คือเมื่อข้อความถูกแบ่งออกเป็นหน่วยย่อยเพื่อใช้งาน

Stop word คือเมื่อคำทั่วไปถูกลบออกจากข้อความเพื่อให้คำที่ไม่ซ้ำกันซึ่งให้ข้อมูลส่วนใหญ่เกี่ยวกับข้อความยังคงอยู่

Lemmatization and stemming คือเวลาที่คำถูกลดขนาดให้อยู่ในรูปแบบเดียว ประมวลผล

Part-of-speech tagging ทำการทำเครื่องหมายคำตามส่วนของคำพูด เช่น คำนาม กริยา และคำคุณศัพท์

เมื่อข้อมูลได้รับการประมวลผลล่วงหน้า อัลกอริทึมจะได้รับการพัฒนาเพื่อประมวลผล มีอัลกอริธึมการประมวลผลภาษาธรรมชาติที่แตกต่างกันมากมาย แต่โดยทั่วไปจะใช้สองประเภทหลัก:

Rules-based system. ระบบนี้ใช้กฎทางภาษาที่ออกแบบมาอย่างดี แนวทางนี้ใช้ในช่วงเริ่มต้นในการพัฒนาการประมวลผลภาษาธรรมชาติ และยังคงใช้อยู่

Machine learning-based system. อัลกอริธึมการเรียนรู้ของเครื่องใช้บริทางสถิติ พากเขาระบุเรียนรู้ที่จะดำเนินการตามข้อมูลการฝึกอบรมที่ได้รับ และปรับวิธีการของตนเมื่อมีการประมวลผลข้อมูลมากขึ้น อัลกอริธึมการประมวลผลภาษาธรรมชาติใช้การผสมผสานระหว่างแมชชีนเลิร์นนิ่ง การเรียนรู้เชิงลึก และ~~โครงข่ายประสาทเทียม~~ ผ่านการประมวลผลและการเรียนรู้ซ้ำๆ

2.2.2 เทคนิคการประมวลผลภาษาธรรมชาติที่มีความสำคัญ

ธุรกิจใช้ข้อมูลที่ไม่มีโครงสร้างและมีข้อความจำนวนมากและต้องการวิธีในการประมวลผลอย่างมีประสิทธิภาพ ข้อมูลจำนวนมากที่สร้างขึ้นทางออนไลน์และจัดเก็บไว้ในฐานข้อมูลเป็นภาษาบนุษย์ตามธรรมชาติ และจักระทั้งเมื่อไม่นานมานี้ ธุรกิจต่างๆ ก็ไม่สามารถวิเคราะห์ข้อมูลนี้ได้อย่างมีประสิทธิภาพ นี่คือจุดที่การประมวลผลภาษาธรรมชาติมีประโยชน์

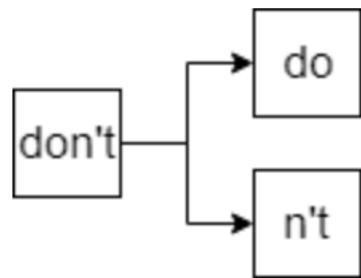
ประโยชน์ของการประมวลผลภาษาธรรมชาติสามารถเห็นได้เมื่อพิจารณาจากสองข้อความต่อไปนี้: "การประกันการประมวลผลแบบคลาวด์ควรเป็นส่วนหนึ่งของขั้นตอนการให้บริการทุกฉบับ" และ "TLA ที่ดี ช่วยให้นอนหลับสบายตลอดคืน แม้จะอยู่ในระบบคลาวด์" หากผู้ใช้อาศัยการประมวลผลภาษาธรรมชาติในการค้นหา โปรแกรมจะรับรู้ว่าการประมวลผลแบบคลาวด์เป็นอนาคตคลาวด์นั้นเป็นรูปแบบย่อของคลาวด์คอมพิวติ้ง และ TLA นั้นเป็นคำย่อของอุตสาหกรรมสำหรับขั้นตอนการให้บริการ

เหล่านี้เป็นประเภทขององค์ประกอบที่คุณเครื่องมือมักปรากฏในภาษาบนุษย์และอัลกอริธึมการเรียนรู้ด้วยเครื่องมีการติดตามที่ไม่ดีในอดีต ด้วยการปรับปรุงวิธีการเรียนรู้เชิงลึกและแมชชีนเลิร์นนิ่ง อัลกอริธึมสามารถติดตามความได้อย่างมีประสิทธิภาพ การปรับปรุงเหล่านี้ช่วยขยายความกว้างและความลึกของข้อมูลที่สามารถวิเคราะห์ได้

2.2.3 เทคนิคและวิธีการประมวลผลภาษาธรรมชาติ

ทั่วไปสำหรับข้อความก่อนการประมวลผลประกอบด้วย 5 อย่าง

2.2.3.1 Sentence segmentation ในขั้นตอนแรกของการเตรียมประโภคข้อความจะถูกแบ่งออกเป็นประโภคข้อความที่ใช้ เช่น ภาษาอังกฤษ เครื่องหมายวรรคตอน โดยเฉพาะอักขระหยุด เครื่องหมายอัศจรรย์ และเครื่องหมายคำนามสามารถใช้ระบุจุดสิ้นสุดของประโภคได้อย่างไรก็ตาม อักขระจุดยังสามารถใช้เป็นตัวย่อของข้อความได้ เช่น Ms. หรือ UK ซึ่งในกรณีนี้ อักขระหยุดไม่ได้หมายถึงจุดสิ้นสุดของประโภค ในกรณีเหล่านี้ใช้อักขระย่อเพื่อหลีกเลี่ยงการแบ่งประเภทของเขตประโภคที่ไม่ถูกต้อง เมื่อข้อความมีคำศัพท์เฉพาะ เราจะต้องสร้างพจนานุกรมคำย่อเพิ่มเติมเพื่อหลีกเลี่ยงการทำเครื่องหมายผิดหลักธรรมชาติตัวอย่างการทำให้เป็นมาตรฐาน



ภาพที่ 1 เป็นการทำ Tokenization [6]

2.2.3.2 Tokenization คือ การแบ่งข้อความออกเป็นคำและเครื่องหมายวรรคตอนที่เป็นเครื่องหมาย เช่น เดียวกับการแบ่งประโภคเครื่องหมายวรรคตอน ตัวอย่างเช่น U.K. ควรจะเป็นเครื่องหมาย และ don't ควรแบ่งออกเป็นสองเครื่องหมาย do และ n't

2.2.3.3 Stemming และ lemmatization เป็นส่วนสำคัญของกระบวนการทำให้เป็นมาตรฐาน การทำให้เป็นมาตรฐานประกอบด้วยการสกัดคำที่ต้องระบุต้นคำโดยการลบต่อท้าย เช่น -ed และ -ing ไม่จำเป็นต้องเป็นคำ ในทำนองเดียวกัน lemmatization เกี่ยวข้องกับการลบคำนำหน้าและส่วนต่อท้าย

ความแตกต่างที่สำคัญคือผลที่ได้คือภาษา ผลที่ได้นี้เรียกว่าการอ้างอิง ตัวอย่างของ Stemming และ lemmatization ดังรูปที่ 2

	Word 1	Word 2	Word 3
Original	studies	playing	best
Stem	studi	play	best
Lemma	study	play	good

ภาพที่ 2 รูปแบบของการใช้งาน Stemming และ lemmatization [6]

ทั้ง 2 เทคนิคที่กล่าวมานี้ช่วยในการลดสัญญาณรบกวนในข้อความโดยแปลงคำให้อยู่ในรูปแบบพื้นฐาน เช่น การประเภทข้อความหรือการจัดกลุ่มเอกสาร ซึ่งการรักษาความหมายของคำเป็นสิ่งสำคัญ ควรใช้ lemmatization มากกว่าการวิเคราะห์ ตัวอย่างเช่น คำนามและคำกริยา ซึ่งทำให้สัญเสียงความหมายตั้งเดิมไป เทคนิคการทำให้เป็นมาตรฐานอื่น ๆ ได้แก่ การขยายคำย่อ การลบตัวเลขและเครื่องหมายวรรคตอน การแก้ไขคำผิดพลาดทางไวยากรณ์ การดำเนินการเหล่านี้ส่วนใหญ่สามารถทำได้โดยใช้尼พจน์ทั่วไป

2.2.3.4 Part of speech tagging ขั้นตอนนี้จะเป็นการแบ่งเครื่องหมายเป็น part of speech (POS) หรือที่เรียกว่าคำศัพท์ หรือ หมวดหมู่คำศัพท์ คำที่ประกอบด้วยคำนาม, คำกริยา, คำบุพบท, คำกริยาวิเศษณ์ ดังตารางต่อไปนี้จะแสดงคำและตัวอย่าง ส่วนสัญลักษณ์ จะใช้ lemmatization ซึ่งเป็นสิ่งจำเป็นสำหรับการตั้งชื่อ บุคคล

Lexical category	Example
Noun	book, girl, forest, moss
Verb	play, study, write, choose
Adjective	happy, short, brown, cool
Preposition	at, about, over, on
Determiner	the, a, this, those
Conjunction	and, but, or, if
Pronoun	I, she, you, they

ภาพที่ 3 ตัวอย่างการติดแทรกจัดกลุ่มข้อความ POS-taggers [6]

POS-taggers มี 3 ประเภท ได้แก่ ตามกฎสถิติและตามการเรียนรู้เชิงลึก กฎตามเครื่องหมายขั้นอยู่กับว่ากฎที่ชัดเจนเพื่อทำเครื่องหมาย เช่น บทความต้องตามด้วยคำนาม เพื่อกำหนดเครื่องหมาย ตามกฎสถิติใช้แบบจำลองความน่าจะเป็นในการมาร์คแต่ละคำหรือลำดับของคำ กฎตามแท็กตามกฎนั้นแม่นยำมาก แต่ก็ยังขึ้นอยู่กับภาษาด้วย การขยาย tagger เพื่อรับรู้ภาษาอื่น ๆ ตัวติดแท็กภาษาอังกฤษนั้นสร้างได้ง่ายกว่าและไม่ขึ้นกับภาษา และมีการใช้วิธีสมมผานของแบบจำลองตามกฎและแบบจำลองทางสถิติ โดยที่แบบจำลองจะได้รับการฝึกอบรมเกี่ยวกับชุดประโยคที่ติดแท็กล่วงหน้า วิธีการแบบไฮบริดและการเรียนรู้เชิงลึกจะสามารถปรับปรุงการติดแท็กได้ตามบริบท

2.2.3.5 Named entity Recognition คือการแบ่งกลุ่มของเครื่องหมาย การแบ่งกลุ่มหมายถึงการติดแท็ก หนึ่งในกลุ่มคำที่ใช้มากที่สุด คือกลุ่มคำนามที่ประกอบด้วยตัวกำหนด คำคุณศัพท์ และคำนาม เช่น a happy unicorn ประโยค He found a happy unicorn ประกอบด้วยสองส่วน he และ a happy unicorn

Barack Obama PERSON is an American NORP attorney and politician who
 PERSON served as the 44th ORDINAL president of the United States GPE from
 2009 to 2017 DATE .

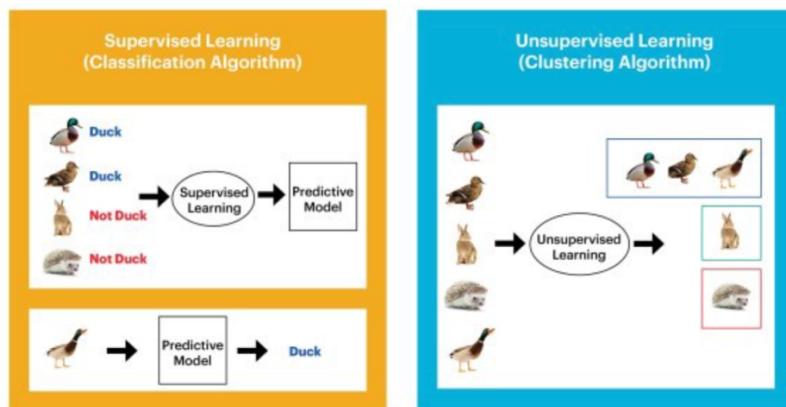
ภาพที่ 4 การติดแทรกที่อ้างอิงถึงวัตถุเฉพาะ Named entity [6]

Named entity เป็นคำนามที่อ้างอิงถึงวัตถุเฉพาะ เช่น บุคคล องค์กร สถานที่ วันที่ และหน่วยทางภูมิศาสตร์ เป้าหมายของขั้นตอน Named entity Recognition คือการระบุชื่อนิติบุคคลที่กล่าวถึงในข้อความ

2.3 Machine Learning

As Brink และคนอื่น ๆ ให้คำจำกัดความไว้ว่า Machine Learning(ML) คือการใช้ประโยชน์จากรูปแบบของข้อมูลในอดีตเพื่อตัดสินใจเกี่ยวกับข้อมูลใหม่ หรือเป็นทาง Google หัวหน้านักวิทยาศาสตร์ด้านการตัดสินใจ หรือ Cassie Kozyrkov ซึ่งให้เห็นว่า Machine Learning เป็นเพียงตัวติดฉลาก อธิบายไว้เกี่ยวกับบางสิ่งบางอย่างและบอกให้รู้ว่าควรได้รับฉลากอะไร การใช้เทคนิค ML มีประโยชน์เมื่อปัญหานั้นซับซ้อนเกินกว่าจะแก้ไขด้วยการเขียนโปรแกรม เช่น แยกแยะสายพันธุ์แมวต่าง บนรูปภาพ หรือโซลูชันจำเป็นต้องปรับเปลี่ยนเมื่อเวลาผ่านไป เช่น การจดจำข้อความที่เขียนด้วยลายมือ

โดยทั่วไปแล้ว Machine Learning จะแบ่งออกเป็น Machine Learning ที่จะต้องดูแล และ ไม่ต้องดูแล เราสามารถการเรียนรู้ภายใต้การดูแลเมื่อข้อมูลการฝึกอบรมในอดีตของเรามีป้ายกำกับ (เช่น duck และ no duck ในรูปตัวอย่างด้านล่าง) ในทางกลับกันการเรียนรู้แบบไม่มีผู้ดูแลจะถูกนำมาใช้เมื่อไม่มีป้ายกำกับในข้อมูล วิธีการเรียนรู้ของเครื่องที่ไม่ได้รับการดูแลมีเป้าหมายเพื่อสรุปหรือบีบอัดข้อมูลการฝึกอบรมพร้อมป้ายกำกับกับ ส่วน/ไม่ส่วน ในกรณีหลัง เราจะต้องตรวจหาอีเมลผิดปกติตามชุดการฝึกอบรมของอีเมล



ภาพที่ 5 ความแตกต่างระหว่างการเรียนรู้แบบมีผู้ดูแล และ ไม่มีผู้ดูแล [6]

2.4 Extracting features from text

อัลกอริทึม ของ Machine Learning ทั้งหมดต้องการข้อมูลดิจิตอลเป็นอินพุต ซึ่งหมายความว่าข้อมูลและข้อความจะต้องถูกแปลงเป็นตัวเลข ขั้นตอนการแยกคุณลักษณะของ NLP

2.4.1 Count-based strategies

เป็นวิธีที่ง่ายที่สุดในการแปลงข้อความเป็นเวกเตอร์ตัวเลขคือการใช้วิธี Bag-of-Words (BoW) หลักการของ BoW คือการแยกคำที่ไม่ซ้ำกันทั้งหมดจากข้อความและสร้างคลังข้อความที่เรียกว่าคำศัพท์ การใช้คำศัพท์แต่ละประโยคสามารถแสดงเวกเตอร์ประกอบด้วย 1 และ 0 ขึ้นอยู่กับว่ามีคำศัพท์อยู่ในประโยคหรือไม่ รูปด้านล่างแสดงตัวอย่างของเมทริกที่สร้างขึ้นโดยใช้วิธี BoW ในประโยคห้าประโยคที่ทำให้เป็นมาตรฐาน

```
[ 'Rabbit jumped over a large fox.',  
  'Unicorns are magical creatures living in dark forests.',  
  'Unicorns and rabbits live in forests.',  
  'Google is being sued by European Union',  
  'Apple and Google are some of the biggest companies in the world']
```

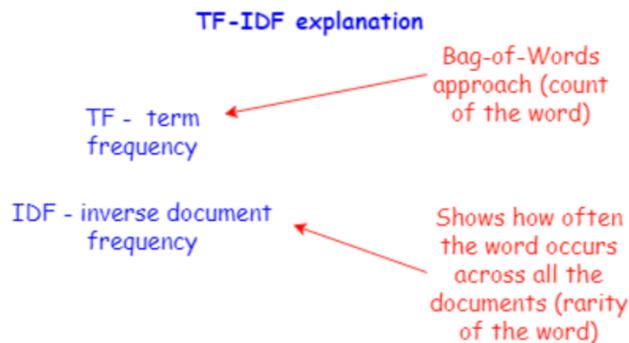
ภาพที่ 6 เป็นขั้นตอนการเคลี่ยงข้อความ ตัดอักขระออก Sentence [6]

	apple	big	company	creature	dark	european	forest	fox	google	jump	large	live	magical	rabbit	sue	unicorn	union	world
0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0
1	0	0	0	0	1	1	0	1	0	0	0	1	1	0	0	1	0	0
2	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0
3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0
4	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

ภาพที่ 7 เป็นสร้างครั้งคำศัพท์เพื่อยกเว้นคำศัพท์ตัวไหนมีการใช้บ่อย หลักการของ BoW [6]

จะสามารถจัดกลุ่มแท็กเข้าด้วยกันเพื่อเพิ่มบริบทเพิ่มเติมไปยังคำศัพท์ วิธีนี้เรียกว่า N-gram วิธี N-gram คือลำดับของเครื่องหมาย N เช่น 2-gram คำลำดับของคำสองคำ ในขณะที่ trigram คือลำดับของสามเมื่อเลือกคำศัพท์แล้ว ไม่ว่าจะเป็น 1-, 2-, หรือ 3- gram จะต้องนับจำนวน gram เราชารถใช้วิธี

BoW ได้ ข้อเสียของแนวทางนี้คือคำที่นิยมมีความสำคัญเกินไป ดังนั้น วิธีที่นิยมใช้กันมากที่สุดจึงเรียกว่า term frequency - inverse document frequency (TF-IDF)



ภาพที่ 8 เป็นการจัดความสำคัญของคำเทียบกับความยาวประโยค [6]

TF-IDF ประกอบด้วย term frequency (TF) เป็นการจัดความสำคัญของคำเทียบกับความยาวประโยคและ inverse document frequency (IDF) ซึ่งจัดจำนวนแ雷อเอกสารที่ gram เกิดเมื่อเทียบกับจำนวนของแ雷อสารเพื่อเน้นความหมายของคำ ตามที่คิดไว้ คำที่ปรากฏอยู่บ่อยในเอกสารแต่ไม่ค่อยปรากฏในเอกสารทั้งหมด คำหนึ่งจะมีคะแนน TF-IDF สูงกว่า หากพบบ่อยในเอกสารแต่จะไม่พบบ่อยในชุดเอกสารทั้งหมด ดังรูปที่ 9 ตัวอย่างของเมทริกซ์ที่สร้างขึ้นโดยใช้วิธี TF-RDF ในประโยคตัวอย่างที่เห็นก่อนหน้านี้ สังเกตว่าคะแนนของคำว่า fox แตกต่างจากคะแนนที่กำหนดให้กับ rabbit

	apple	big	company	creature	dark	european	forest	fox	google	jump	large	live	magical	rabbit	sue	unicorn	union	world
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.52	0.52	0.00	0.00	0.42	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.45	0.45	0.00	0.36	0.00	0.00	0.00	0.00	0.36	0.45	0.00	0.00	0.36	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.50	0.00	0.50	0.00	0.50	0.00	0.50	0.00
3	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.00	0.42	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.52	0.00
4	0.46	0.46	0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.46

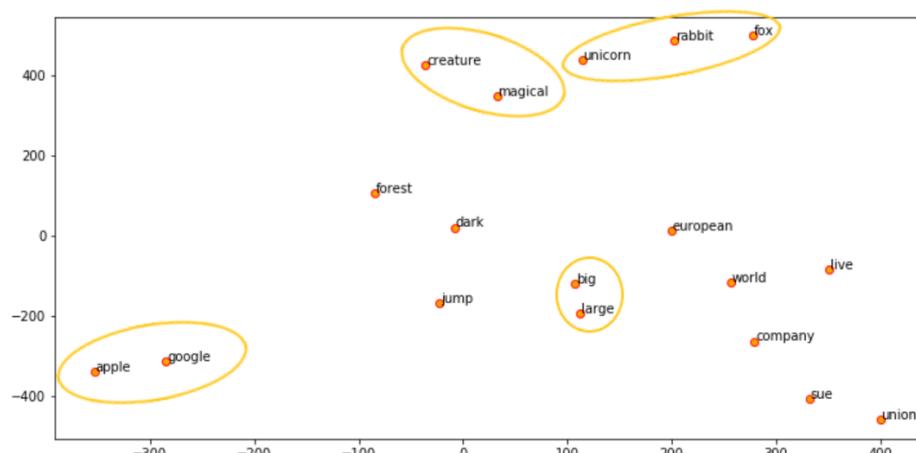
ภาพที่ 9 แสดงถึงความสำคัญของแต่ละประโยค [6]

2.4.2 Advanced strategies

วิธีการ count-based แม้ส่วนสามารถใช้เพื่อจัดลำดับของคำ (N-grams) แต่ก็ไม่ได้จัดบริบททางความหมายของคำซึ่งเป็นแกนหลักของแอปพลิเคชัน NLP จำนวนมาก เทคนิคการฝังคำใช้เพื่อแก้ปัญหานี้ การใช้การฝังคำคำศัพท์จะถูกแปลงเป็นเวกเตอร์เพื่อให้คำที่มีบริบทคล้ายกันอยู่ใกล้กัน

Word2Vec เป็นเฟรมเวิร์คจาก Google ที่ใช้โครงข่ายประสาทเทียมแบบตื้นเพื่อฝึกโมเดลฝังคำสั้น อัลกอริธึม โดย Word2Vec มี 2 ประเภท ประเภทที่ 1 Skip-gram ซึ่งใช้เพื่อทำนายบริบทรอบ ๆ คำที่กำหนด ในขณะที่โถดูล Continuous Bag of Words (CBOW) ใช้เพื่อทำนายคำถัดไปตามบริบทที่กำหนด

วิธีที่ 2 GloVe วิธี Global Vector ใช้สถิติการเกิดขึ้นร่วมเพื่อสร้างช่องว่างเวกเตอร์ วิธีนี้เป็นส่วนขยายต่อมาจากการฝังคำที่มีแนวโน้มว่าจะให้การฝังคำที่ดีกว่า

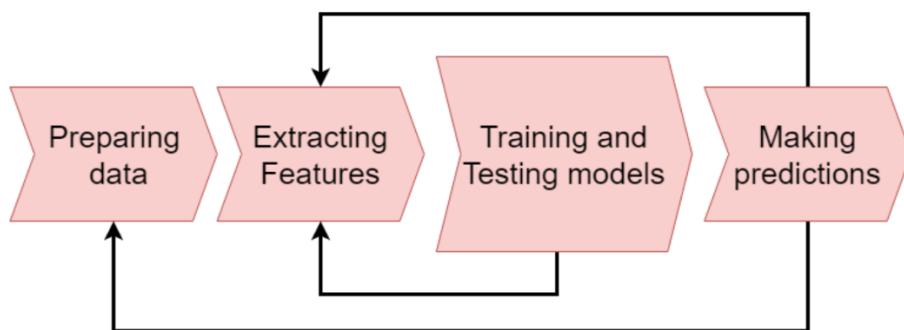


ภาพที่ 10 เป็นการแสดงบริบทที่คล้ายกันจัดอยู่ในกลุ่มเดียวกัน [6]

2.5 Supervised learning on text

2.5.1 Supervised learning

การดูแลเครื่องการเรียนรู้งานแบ่งออกเป็นสองส่วน ตามรูปแบบของป้ายกำกับ (เรียกอีกอย่างว่าเป้าหมาย) หากเป้าหมายคือการจำแนกค่า (cat/dog) แสดงว่าเป็นปัญหาการจัดหมวดหมู่ ในทางกลับกัน หากเป้าหมายเป็นตัวเลข (ราคาของบ้าน) แสดงว่าปัญหาการทดสอบ เมื่อต้องจัดการกับข้อความปัญหาที่ตามมาจะเป็นการจำแนกประเภท



ภาพที่ 11 workflow supervised [6]

จากรูปด้านบนแสดง workflow ทั่วไปของระบบการจัดการประเภทข้อความ เราเริ่มต้นด้วยการแบ่งข้อมูลออกเป็นชุดฝึกอบรม และชุดทดสอบ ข้อมูลชุดฝึกและข้อมูลทดสอบต้องได้รับการประมวลผลล่วงหน้าและทำให้เป็นมาตรฐาน หลังจากนั้นจึงจะสามารถดึงคุณลักษณะออก มาได้ เทคนิคการแยกคุณลักษณะยังนิยมสำหรับข้อมูลประเภทข้อความครอบคลุมอยู่ในส่วนก่อนหน้านี้ เมื่อข้อมูลข้อความถูกแปลงเป็นรูปแบบตัวเลขแล้ว สามารถใช้อัลกอริธึมการเรียนรู้ของเครื่องได้ กระบวนการนี้เรียกว่าฝึกโมเดล โดยการใช้ พารามิเตอร์โมเดลผ่านกระบวนการที่เรียกว่าการปรับแต่งไฮเปอร์พารามิเตอร์ แบบจะลงผลลัพธ์จะถูกประเมินบนข้อมูลการทดสอบที่มองไม่เห็นก่อนหน้า ประสิทธิภาพของโมเดลวัดโดยใช้เมตริกต่าง ๆ เช่น ความแม่นยำ การเรียกคืนค่าคะแนน F1 และอื่นๆ อัลกอริธึมที่ใช้สำหรับการจัดประเภทข้อความเช่น

2.5.1.1 Multinomial Naive Bayes อูปในตรากูลของอัลกอริทึม Naïve Bayes ซึ่งสร้างขึ้นจากการใช้ทฤษฎีของ Bayes โดยใช้สมมติฐานที่มีป้ายกำกับต่างกันมากกว่าสองป้ายที่แตกต่างกัน

2.5.1.2 Logistic Regression เป็นอัลกอริทึมที่ใช้พังก์ชัน Sigmoid เพื่อคำนวณค่าการจำแนก แพคเกจซอฟแวร์ที่ได้รับความนิยม SKLearn จะอนุญาติให้ปรับพารามิเตอร์ของโมเดลในลักษณะที่อัลกอริทึมสามารถใช้สำหรับการจำแนกประเภทหลายป้ายกำกับได้ เช่น กัน

2.5.1.3 Support Vector Machines (SVM) อัลกอริธึมที่ใช้เส้นหรือไฮเปอร์เพลน (ในกรณีที่ มีคุณสมบัติมากกว่าสองอย่าง จึงสร้างพื้นที่หลายมิติ) เพื่อแยกคลาส

2.5.1.4 Random Forest การบูรณาการวิธีการฝึกต้นไม้ตัดสินใจหลายนานในเชิงย่อย ข้อมูลที่แตกต่างกัน

2.5.1.5 Gradient Boosting Machine (GBM) ชุดของวิธีการแบบบูรณาการที่ใช้ในการฝึก ชุดของผู้เรียนที่อ่อนแอก่อน เช่นต้นไม้การตัดสินใจที่จะได้รับผลลัพธ์ที่ถูกต้อง XGboost เป็นหนึ่งในการใช้งานที่ นิยมมากที่สุดของชุดนี้

Random Forest และ Random Forest เป็นอัลกอริทึมการจัดหมวดหมู่ เป็น วิธีการแบบบูรณาการซึ่งใช้ขั้นตอนวิธีการพยากรณ์หลายเพื่อให้บรรลุทั่วไปดีกว่า ผลของวิธีการตั้งค่ามักจะ เฉลี่ยมากกว่ารุ่นเดียวและมีประสิทธิภาพมากขึ้นในชุดข้อมูลที่มีขนาดใหญ่กว่า อย่างไรก็ตามเป็น sarkar ได้ พิสูจน์แล้วว่าใน วิธีการบูรณาการไม่จำเป็นต้องจัดการกับข้อมูลข้อความดีกว่า

2.6 Naive Bayes Classification

เป็นการจัดหมวดหมู่โดยใช้หลักความน่าจะเป็นเข้ามาช่วยคำนวณ

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood Class Prior Probability
 $P(c|x)$ = $\frac{P(x|c)P(c)}{P(x)}$
 ↓ ↓
 Posterior Probability Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

อธิบายสมการ แทนตัวแปร 3 ตัว คือ

c คือ Class

x คือ Attribute

P คือ Probability (ความน่าจะเป็น)

$P(c|x)$ Posterior probability คือ

ความน่าจะเป็นที่ข้อมูลที่มีแอ็ตทริบิวต์เป็น x จะมีคลาส C

$P(x|c)$ Likelihood คือ ความน่าจะเป็นที่ข้อมูลที่มีคลาส C และมีแอ็ตทริบิวต์ x

$P(c)$ Prior probability คือ จำนวน Class ที่อาจจะเกิดขึ้น / จำนวน Class ทั้งหมด

หรือความน่าจะเป็นของ Class C

$P(x)$ Predictor Prior probability คือ จำนวน Attribute ทั้งหมด

2.7 PyWebIO

มีชุดฟังก์ชันที่จำเป็นเพื่อรับอินพุตและเอาต์พุตของผู้ใช้บนเบราว์เซอร์ เปลี่ยนเบราว์เซอร์ให้เป็นเทอร์มินัลข้อความที่มีรูปแบบ" และสามารถใช้เพื่อสร้างเว็บแอปพลิเคชันอย่างง่ายหรือแอปพลิเคชัน GUI บนเบราว์เซอร์โดยไม่จำเป็นต้องมีความรู้ HTML และ JS นอกจากนี้ยังสามารถรวมเข้ากับบริการบนเว็บที่มีอยู่ได้อย่างง่ายดาย หมายเหตุสำคัญคือ PyWebIO ไม่ได้สนับสนุนการสร้าง UI ที่ซับซ้อน

PyWebIO เป็นเครื่องมือในหมวด **Low Code Platforms** ของกลุ่มเทคโนโลยี

PyWebIO เป็นเครื่องมือโอเพ่นซอร์สที่มี GitHub ดาวและส้อม GitHub นี้คือลิงค์ไปยังที่เก็บโอเพ่นซอร์สของ PyWebIO บน [GitHub](#)

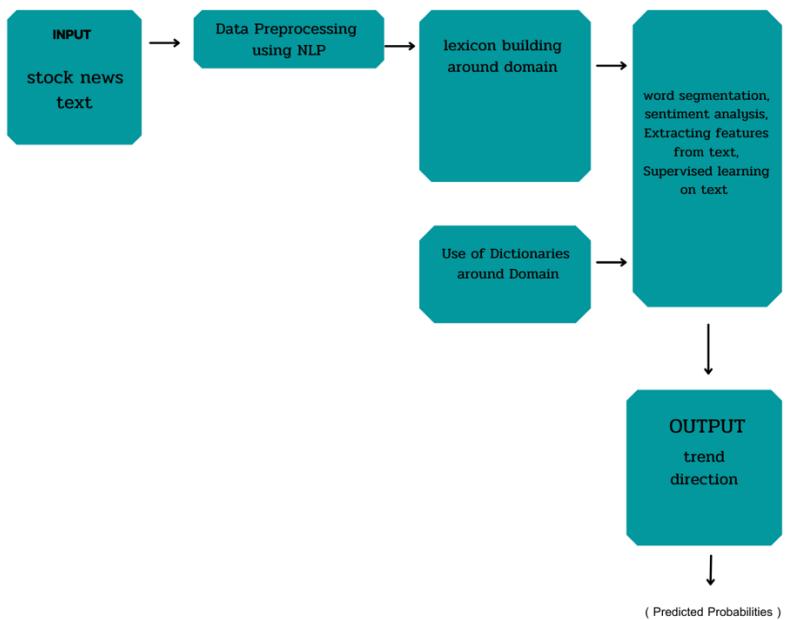
2.7.1 คุณสมบัติของ PyWebIO

- ฟรีและโอเพ่นซอร์ส
- รหัสตัว
- ประสิทธิภาพสูง
- รองรับทั้งการบล็อกอินพุตและการโทรกลับ
- ผ่อนรวมกับเว็บเฟรมเวิร์กที่ใช้ Python, flask, Django, FastAPI และอื่นๆ ได้ง่าย
- ทำงานร่วมกับไลบรารีเกือบทั้งหมดในระบบปฏิบัติการ Python

บทที่ 3 รายละเอียดการดำเนินงาน

3.1 ขั้นตอนและวิธีการดำเนินงาน

3.1.1 ขั้นตอนการทำงานของระบบ



ภาพที่ 12 เป็นการแสดงขั้นตอนการทำงานของ NLP

3.1.2 ขั้นตอน INPUT (แหล่งข้อมูลข่าวสาร)

ดังรูปที่ 1

3.1.3 ขั้นตอน Processing

1. การแยกตัวย่อและคำจำกัดความ
2. แยกหน่วยงาน (เช่น คน บริษัท ผลิตภัณฑ์ จำนวนเงิน สถานที่ ฯลฯ)
3. ดึงข้อมูลอ้างอิงไปยังเอกสารอื่น ๆ
4. การแยกอารมณ์ความรู้สึก (ข่าวเชิงบวก/เชิงลบและการอ้างอิง)
5. ดึงคำพูดจากบุคคลที่มีการอ้างอิงถึงผู้เขียน

6. สถิติเงื่อนไขสัญญา
7. เลือกอัลกอริทึมที่จะมาใช้ในการทำนาย
8. เทคนิค

3.1.4 ขั้นตอน Output

1. สรุปข้อความ
2. คาดการณ์ล่วงหน้า
3. แสดงทิศทางแนวโน้มราคาก้าวขึ้นหรือลง
4. แสดงขั้นบนเว็บไซต์

3.2 ขั้นตอนการดำเนินงาน

1. พัฒนาอัลกอริทึม
2. พัฒนาเขียนโปรแกรมด้วย Python
3. เทคนิคสมองกล
4. ทดสอบระบบ
5. ทดลองนำไปใช้จริง กับบัญชี Forex demo
6. ปรับปรุงแก้ไขอุปกรณ์ให้สมบูรณ์
7. จัดทำเว็บไซต์
8. จัดทำรายงานให้สมบูรณ์

บทที่ 4 ความก้าวหน้าการดำเนินงาน

4.1 ความก้าวหน้า 1 ศึกษา Library Python NLP และทดสอบ

4.1.1 Library Python NLP

4.1.1.1 NLTK Natural Language Toolkit

อินเทอร์เฟซที่ใช้งาน

1. Tokenisation
2. Stemming
3. Tagging
4. Parsing
5. Semantic reasoning
6. Wrappers for industrial-strength NLP libraries WordNet

4.1.1.2 TextBlob ศึกเป็น Library ของภาษา Python

อินเทอร์เฟซที่ใช้งาน

1. Tagging
2. noun phrase extraction
3. sentiment analysis
4. classification
5. language translation
6. word inflection, parsing
7. n-grams
8. WordNet integration.

4.1.2 รายละเอียดการทดลอง

4.1.2.1 Tokenizing Words and Sentences with NLTK



The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

MacBook-Pro-khng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 52987 -- /Users/whatbest/project/nltknp1.py
['Hello Mr. Smith, how are you doing today?', 'The weather is great, and Python is awesome.', 'The sky is pinkish-blue.', "You shouldn't eat cardboard."]
['Hello', 'Mr', '.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The', 'weather', 'is', 'great', ',', 'and', 'Python', 'is', 'awesome', '.', 'The', 'sky', 'is', 'pinkish-blue', '.', 'You', 'shouldn', 't', 'eat', 'cardboard', '.']

MacBook-Pro-khng-Chualchai:project whatbest$
```

Sentences _Tokenizing – การแยกประโยคและคำอกรจากเนื้อความของข้อความ
Words_ Tokenizing เป็นการแยกคำอกรมาจากการประยุกต์

4.1.2.2 Stop words with NLTK

```
stopwords.py > ...
1  from nltk.corpus import stopwords
2  from nltk.tokenize import word_tokenize
3
4  example_sent = "This is a sample sentence, showing off the stop words filtration."
5
6  stop_words = set(stopwords.words('english'))
7
8  word_tokens = word_tokenize(example_sent)
9
10 filtered_sentence = [w for w in word_tokens if not w in stop_words]
11
12 filtered_sentence = []
13
14 for w in word_tokens:
15     if w not in stop_words:
16         |   filtered_sentence.append(w)
17 print("stop_words = ",stop_words)
18 print("word_tokens = ",word_tokens)
19 print("filtered_sentence = ",filtered_sentence)

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

MacBook-Pro-khxng-Chualchai:project whatbests cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 53097 — /Users/whatbest/project/stopwords.py
stop_words = {'hadn', 'itself', 'just', 'shan', 'you', 'as', 'an', 'here', 'how', 'thatll', 'below', 'very', 'where', 'the', 'doesn', 'theirs', 'on', 'wouldnt', 'is', 'any', 'nor', 've', 'he', 'them', 'yourselves', 'arent', 'their', 'about', 'myself', 'itself', 'while', 'y
weren', 'werent', 'wasnt', 'arent', 'hadnt', 'hasnt', 'had', 'has', 'havin', 'will', 'it', 'these', 'having', 'few', 'ma', 'or', 'herself', 'yourself', 'we', 'shouldve', 'mightn', 'she', 'ourselves', 'wont', 'are', 'did', 'so', 'couldnt', 'for', 'wa
s', 'off', 'our', 'only', 'who', 'u', 'it', 'when', 'can', 'werent', 'o', 'had', 'am', 'll', 'that', 'from', 'they', 'about', 'counl', 'on
l', 'weren', 'werent', 'now', 'further', 'and', 'haven', 'hers', 'not', 'should', 'most', 'i', 'm', 're', 'wasn', 'once', 'durin', 'myself', 'on
mustnt', 'her', 'than', 'needn', 'shouldnt', 'in', 'again', 'between', 'haven', 'his', 'into', 'then', 'youd', 'but', 'themselves', 'don', 'does', 'at', 'on', 'you', 're', 'recievin', 'herself', 'down', 'about', 'other', 'neednt', 'isnt', 'has', 'after', 'shes', 'of', 'ot
'each', 'this', 'right', 'its', 'through', 'doing', 'um', 'sh', 'shan', 'what', 'whose', 'more', 'd', 'ain', 'mustn', 'same', 'which', 'didn', 'out', 'hadn', 'before', 'don', 'against', 'under', 'shoul', 'woudn', 'ours', 'didn', 'until', 'there', 'yours', 'y', 'by', 'be', 'being', 'hasn', 't', 'such', 'dosen', 've', 'those', 'with'
MacBook-Pro-khxng-Chualchai:project whatbests █
```

เป็นการใช้ คำหยุด โดยอ้างองค์จากคลังข้อมูลของภาษาอังกฤษ คำหยุด คือ คำที่ เป็นคำลีหรือเป็นคำที่เสริมทำให้ประโยคดีขึ้น แต่ในเชิงคอมพิวเตอร์ คำหยุด คือคำที่ไม่มีความหมาย เหตุผล ที่ต้องลบคำหยุดเนื่องจากประโยคที่ยารื้นจะกินทรัพยากรและเวลาในการประมวลผล

4.2 ความก้าวหน้า 2 ทดสอบ Library NLTK

4.2.1 รายละเอียดการทดลอง

4.2.1.1 Stemming words with NLTK

```
stem.py > ...
1  from nltk.stem import PorterStemmer
2  from nltk.tokenize import sent_tokenize, word_tokenize
3
4  ps = PorterStemmer()
5
6  example_words = ["python","pythoner","pythoning","pythonly","feeling"]
7
8  for w in example_words:
9      print(ps.stem(w))
10
11 new_text = "It is important to by very pythonly while you are pythoning with python. All pythoners have pythoned poorly at least onc
12 words = word_tokenize(new_text)
13
14 for w in words:
15     print(ps.stem(w))
16
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
MacBook-Pro-khang-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 53446 — /Users/whatbest/project/stem.py
python
python
python
python
python,feel
it
is
import
to
by
veri
pythonli
while
you
are
python
with
python
all
all
python
have
python
portli
at
least
one
.
```

MacBook-Pro-khang-Chualchai:project whatbest\$ []

Stemming หรือที่เรียกว่า suffix stripping เป็นเทคนิคที่ใช้ในการลดขนาดข้อความ ต้นกำเนิดยังเป็นประเภทของข้อความที่ทำให้เป็นมาตรฐานที่ช่วยให้สร้างมาตรฐานของคำบางคำให้เป็นนิพจน์เฉพาะ ข้อเสีย ในไฟล์ไอล์การทำเหมือนข้อความหลายๆ แบบ มีตัวเลือกมากมายที่เกี่ยวข้องซึ่งอาจทำให้ข้อมูลสูญหายได้

4.2.1.2 Part of Speech Tagging with NLTK

POS Tagging (ส่วนหนึ่งของการแท็กคำพูด) เป็นกระบวนการในการทำ

เครื่องหมายคำในรูปแบบข้อความสำหรับส่วนได้ส่วนหนึ่งของคำพูดตามคำจำกัดความและบริบท มีหน้าที่รับผิดชอบในการอ่านข้อความในภาษาและกำหนดโทนเสียง (Parts of Speech) ให้กับแต่ละคำ เรียกว่า กองคำ อ่านว่า การติดแท็กทั้งไวยากรณ์

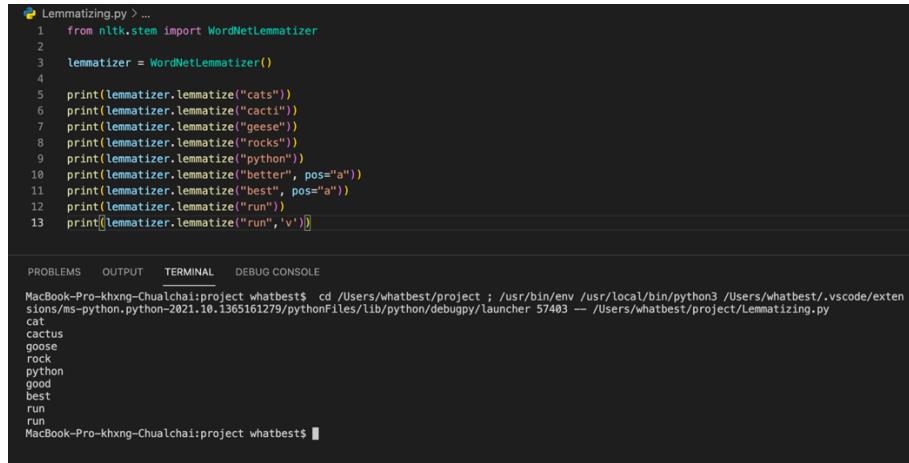
ตัวอย่างแท็ก NLTK POS มีดังนี้:

Abbreviation	Meaning
CC	coordinating conjunction
CD	cardinal digit
DT	determiner
EX	existential there
FW	foreign word
IN	preposition/subordinating conjunction
JJ	This NLTK POS Tag is an adjective (large)
JJR	adjective, comparative (larger)
JJS	adjective, superlative (largest)
LS	list marker
MD	modal (could, will)
NN	noun, singular (cat, tree)
NNS	noun plural (desks)
NNP	proper noun, singular (sarah)
NNPS	proper noun, plural (indians or americans)
PDT	predeterminer (all, both, half)
POS	possessive ending (parent\ 's)

PRP	personal pronoun (hers, herself, him, himself)
PRP\$	possessive pronoun (her, his, mine, my, our)
RB	adverb (occasionally, swiftly)
RBR	adverb, comparative (greater)
RBS	adverb, superlative (biggest)
RP	particle (about)
TO	infinite marker (to)
UH	interjection (goodbye)
VB	verb (ask)
VBG	verb gerund (judging)
VBD	verb past tense (pleaded)
VBN	verb past participle (reunified)
VBP	verb, present tense not 3rd person singular(wrap)
VBZ	verb, present tense with 3rd person singular (bases)
WDT	wh-determiner (that, what)
WP	wh- pronoun (who)
WRB	wh- adverb (how)

ตารางที่ 3

4.2.1.3 Lemmatizing with NLTK



```
Lemmatizing.py > ...
1  from nltk.stem import WordNetLemmatizer
2
3  lemmatizer = WordNetLemmatizer()
4
5  print(lemmatizer.lemmatize("cats"))
6  print(lemmatizer.lemmatize("cacti"))
7  print(lemmatizer.lemmatize("geese"))
8  print(lemmatizer.lemmatize("rocks"))
9  print(lemmatizer.lemmatize("python"))
10 print(lemmatizer.lemmatize("better", pos="a"))
11 print(lemmatizer.lemmatize("best", pos="a"))
12 print(lemmatizer.lemmatize("run"))
13 print([lemmatizer.lemmatize("run", 'v')])

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 57403 -- /Users/whatbest/project/Lemmatizing.py
cat
cactus
goose
rock
python
good
best
run
run
MacBook-Pro-khxng-Chualchai:project whatbest$
```

Lemmatization เป็นกระบวนการของการจัดกลุ่มคำในรูปแบบผันแปรต่างๆ เพื่อให้สามารถวิเคราะห์เป็นรายการเดียวได้ Lemmatization คล้ายกับการกำเนิด แต่นำบริบทมาสู่คำ ดังนั้นจึงเชื่อมโยงคำที่มีความหมายคล้ายกันเป็นคำเดียว

4.3 ความก้าวหน้า 3 ทดสอบ Library NLTK

4.3.1 รายละเอียดการทดลอง

4.3.1.1 Wordnet with NLTK

```
Wordnet with NLTK.py > ...
1  from nltk.corpus import wordnet
2  syns = wordnet.synsets("program")
3  print(syns[0].name())
4  print(syns[0].lemmas()[0].name())
5  print(syns[0].definition())
6  print(syns[0].examples())
7  synonyms = []
8  antonyms = []
9
10 for syn in wordnet.synsets("good"):
11     for l in syn.lemmas():
12         synonyms.append(l.name())
13         if l.antonyms():
14             antonyms.append(l.antonyms()[0].name())
15
16 print(set(synonyms))
17 print(set(antonyms))

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 57441 -- "/Users/whatbest/project/Wordnet with NLTK.py"
plan.n.01
plan
  A series of steps to be carried out or goals to be accomplished
  ['they drew up a six-step plan', 'they discussed plans for a new bond issue']
{'near', 'effective', 'proficient', 'respectable', 'skilful', 'secure', 'unspoilt', 'estimable', 'beneficial', 'goodness', 'upright', 'right', 'in_effect', 'expert', 'honorable', 'dependable', 'ripe', 'serious', 'soundly', 'just', 'undecomposed', 'safe', 'full', 'dear', 'in_force', 'thoroughly', 'adept', 'sound', 'salutary', 'skillful', 'trade_good', 'commodity', 'good', 'well', 'honest'}
{'evil', 'bad', 'evilness', 'badness', 'ill'}
```

WordNet เป็นฐานข้อมูลคำศัพท์สำหรับภาษาอังกฤษ ซึ่งสร้างโดย Princeton และเป็นส่วนหนึ่งของคลังข้อมูล NLTK

4.3.1.2 Text Summarization with NLTK in Python

main ▾ Project / summarize / Auto-Summarize an article.py / <> Jump to ▾ Go to file ⋮

Chualchai Apichatitiworn commit Latest commit 44eb6a6 28 minutes ago ⌂ History

All 0 contributors

51 lines (42 sloc) | 1.67 KB

```
1 import bs4 as bs
2 import urllib.request
3 import re
4 import nltk
5
6 scraped_data = urllib.request.urlopen('https://www.eia.gov/petroleum/weekly/')
7 article = scraped_data.read()
8
9 parsed_article = bs.BeautifulSoup(article,'lxml')
10
11 paragraphs = parsed_article.find_all('p')
12
13 article_text = ""
14
15 for p in paragraphs:
16     article_text += p.text
17 # Removing Square Brackets and Extra Spaces
18 article_text = re.sub(r'\[[0-9]*\]', ' ', article_text)
19 article_text = re.sub(r'\s+', ' ', article_text)
20 # Removing special characters and digits
21 formatted_article_text = re.sub('[^a-zA-Z]', ' ', article_text )
22 formatted_article_text = re.sub(r'\s+', ' ', formatted_article_text)
23 sentence_list = nltk.sent_tokenize(article_text)
24 stopwords = nltk.corpus.stopwords.words('english')
25
26 word_frequencies = {}
27 for word in nltk.word_tokenize(formatted_article_text):
28     if word not in stopwords:
29         if word not in word_frequencies.keys():
30             word_frequencies[word] = 1
31         else:
32             word_frequencies[word] += 1
33 maximum_frequency = max(word_frequencies.values())
34
35 for word in word_frequencies.keys():
36     word_frequencies[word] = (word_frequencies[word]/maximum_frequency)
37 sentence_scores = {}
38 for sent in sentence_list:
39     for word in nltk.word_tokenize(sent.lower()):
40         if word in word_frequencies.keys():
41             if len(sent.split(' ')) < 30:
42                 if sent not in sentence_scores.keys():
43                     sentence_scores[sent] = word_frequencies[word]
44                 else:
45                     sentence_scores[sent] += word_frequencies[word]
46 import heapq
47 summary_sentences = heapq.nlargest(7, sentence_scores, key=sentence_scores.get)
48
49 summary = ' '.join(summary_sentences)
50
51 print(summary)
```

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

MacBook-Pro-khxng-Chualchai:Project whatbest$ cd /Users/whatbest/Documents/GitHub/Project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 60443 -- "/Users/whatbest/Documents/GitHub/Project/summarize/Auto-Summarize an article .py"
In 2021, we estimate the average Eagle Ford formation well continued to produce more oil at 21,900 barrels in its first month (Figure 3). We developed these production decline curves for Eagle Ford formation wells from approximately 750 sub-county areas, called grids, which are approximately 14 square miles. This is based on the known number of wells already drilled in a grid, their past decline profile, and developing all future potential well sites. Virtually all of this production has occurred in 16 of the 30 counties and within a producing subset of that total area of approximately 7.2 million acres. Without a price capable of providing a return on investment, producers will not invest capital in drilling a well. A substantially larger amount of the area becomes more profitable as a result of higher prices in 2022. However, not all possible acreage will be developed because of future surface infrastructure considerations or leased acreage that is unavailable for development.
MacBook-Pro-khxng-Chualchai:Project whatbest$

```

ในสคริปต์ด้านบน นำเข้าไลบรารีที่สำคัญที่จำเป็นสำหรับการดึงข้อมูลจากเว็บ ก่อน ใจนั้นใช้`urlopen`ฟังก์ชันจาก`urllib.request`ยูทิลิตี้เพื่อขอข้อมูล ต่อไป ต้องเรียก`read`ใช้ฟังก์ชันบน วัตถุที่ส่งคืนโดย`urlopen`ฟังก์ชันเพื่ออ่านข้อมูล ในการแยกวิเคราะห์ข้อมูล ใช้`BeautifulSoup`อ้อบเจ็กต์และ ส่งผ่านอ้อบเจ็กต์ข้อมูลที่คัดลอกมา เช่น`article`และ`xml`ตัวแยกวิเคราะห์

`article_text`มีข้อความโดยไม่มีจังหวะ อย่างไรก็ตาม จะไม่ลบอะไรมากจากบทความ ต้นฉบับ จะไม่ลบตัวเลข เครื่องหมายวรรคตอน และสัญลักษณ์พิเศษอื่นๆ ออกจากข้อความนี้ เนื่องจากเราจะ ใช้ข้อความนี้เพื่อสร้างบทสรุปและความถี่ของคำแบบต่อว่าหนัก

มีสองอ้อบเจ็กต์`article_text`ซึ่งประกอบด้วยบทความต้นฉบับและบทความ`formatted_article_text`ที่ จัดรูปแบบ จะใช้`formatted_article_text`เพื่อสร้างอีสโต้แกรมความถี่ต่อว่าหนักสำหรับคำนั้นๆ และจะ แทนที่ความถี่ต่อว่าหนักเหล่านี้ด้วยคำใน`article_text`อ้อบเจ็กต์

Converting Text To Sentences ได้ทำการประมวลผลข้อมูลต่อว่าหนักแล้ว ต่อไป ต้องแปลงบทความให้เป็นประโยค จะใช้`article_text`วัตถุสำหรับ tokenizing บทความเป็นประโยคเนื่องจาก มีการหยุดเติม`formatted_article_text`ไม่มีเครื่องหมายวรรคตอนใดๆ ดังนั้นจึงไม่สามารถแปลงเป็นประโยค โดยใช้จุดเติมเป็นพารามิเตอร์ได้

Converting Text To Sentences ในการหาความถี่ของการเกิดขึ้นของแต่ละคำ ใช้`formatted_article_text`ตัวแปร ใช้ตัวแปรนี้เพื่อค้นหาความถี่ของการเกิดเนื่องจากไม่มีเครื่องหมายวรรค ตอน ตัวเลข หรืออักษรพิเศษอื่นๆ

ขั้นแรกจะเก็บคำหยุดภาษาอังกฤษทั้งหมดจาก`nltk`ไลบรารีลงใน`stopwords`ตัวแปร ต่อไป จะวนรอบประโยค ทั้งหมดแล้วตามด้วยคำที่เกี่ยวข้องเพื่อตรวจสอบว่าเป็นคำหยุดหรือไม่ หากไม่เป็นเช่นนั้น จะดำเนินการ

ตรวจสอบว่าคำนั้นมีอยู่ในword_frequencyพจนานุกรมหรือไม่ เช่นword_frequenciesหรือไม่ หากพบคำนี้ เป็นครั้งแรก คำนั้นจะถูกเพิ่มลงในพจนานุกรมเป็นคีย์และตั้งค่าเป็น 1 มิฉะนั้น หากคำนั้นมีอยู่ในพจนานุกรม ก่อนหน้านี้ ค่าของคำนั้นจะถูกอัปเดตเพียง 1

Calculating Sentence Scores ตอนนี้ได้คำนวณความถี่ต่อหน้าหนักสำหรับคำทั้งหมดแล้ว ตอนนี้เป็นเวลาที่จะคำนวณคะแนนสำหรับแต่ละประโยคโดยการเพิ่มความถี่ต่อหน้าหนักของคำที่เกิดขึ้นในประโยคนั้นโดยเฉพาะ

ขั้นแรกจะสร้างsentence_scoresพจนานุกรม เปล่า ถูกแจ้งของพจนานุกรมนี้จะเป็นตัวประโยคเอง และค่าจะเป็นคะแนนที่สอดคล้องกันของประโยค ต่อไป จะวนรอบแต่ละประโยคใน the sentence_listและแปลงประโยคเป็นคำ

จากนั้นจะตรวจสอบว่าคำนั้นมีอยู่ในword_frequencyพจนานุกรมหรือไม่ การตรวจสอบนี้ดำเนินการเนื่องจากสร้างsentence_listsรายการจากarticle_textวัตถุ ในทางกลับกัน ความถี่ของคำถูกคำนวณโดยใช้formatted_article_textขอบเขต ซึ่งไม่มีคำหยุด ตัวเลข ฯลฯ

ไม่ต้องการประโยคที่ยาวมากในการสรุป ดังนั้นจึงคำนวณคะแนนสำหรับประโยคที่มีคำน้อยกว่า 30 คำเท่านั้น (แม้ว่าจะปรับแต่งพารามิเตอร์นี้สำหรับกรณีการใช้งานของเองได้ก็ตาม) ต่อไปจะตรวจสอบว่าประโยคนั้นมีอยู่ในsentence_scoresพจนานุกรมหรือไม่ หากไม่มีประโยคดังกล่าว จะเพิ่มลงในsentence_scores พจนานุกรมเป็นคีย์และกำหนดความถี่ต่อหน้าหนักของคำแรกในประโยคเป็นค่าของประโยค ในทางตรงกันข้าม หากประโยคนั้นมีอยู่ในพจนานุกรม เพียงแค่เพิ่มความถี่ต่อหน้าหนักของคำนั้นให้กับค่าที่มีอยู่

Getting the Summary ตอนนี้มีsentence_scoresพจนานุกรมที่มีประโยคที่มีคะแนนตรงกัน เพื่อสรุปทความสามารถใช้ประโยค N อันดับแรกที่มีคะแนนสูงสุด ศรีปต์ต่อไปนี้ดึงประโยค 7 อันดับแรก

ใช้ไลบรารี heapq และเรียกใช้ largestฟังก์ชันเพื่อดึงประโยค 7 อันดับแรกที่มีคะแนนสูงสุด

4.3.1.3 Sentiment Analysis

The screenshot shows a code editor window with a dark theme. At the top, there's a status bar with tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. Below the status bar is a code editor containing the following Python script:

```
1  from textblob import TextBlob
2
3  # Preparing an input sentence
4  sentence = """The platform provides universal access to the world's best education, partnering with top universities and organizations to bring high-quality learning experiences to people around the globe. We believe that everyone deserves a chance to succeed, regardless of their background or circumstances. That's why we've made our platform accessible to anyone with an internet connection, and we're constantly working to improve it. Our mission is to make education more affordable and accessible than ever before, so that everyone can reach their full potential. We're excited to see where this journey takes us, and we invite you to join us on the ride!"""
5
6  analysisPol = TextBlob(sentence).polarity
7  analysisSub = TextBlob(sentence).subjectivity
8
9  print(analysisPol)
10 print(analysisSub)
```

Below the code editor is a terminal window showing the execution of the script. The command run was `python sentimentddd.py`. The output shows the polarity and subjectivity values:

```
MacBook-Pro-khxng-Chualchai:Project whatbest$ cd /Users/whatbest/Documents/GitHub/Project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/Documents/GitHub/Project/textbox/sentimentddd.py
0.5
0.2666666666666666
MacBook-Pro-khxng-Chualchai:Project whatbest$
```

อาท์พุทธของ TextBlob สำหรับงานวิเคราะห์การวิเคราะห์จะภายในช่วง[-1.0, 1.0] ที่-1.0เป็นข้อลบและ1.0เป็นบวก คะแนนนี้ยังสามารถเท่ากับ0ซึ่งหมายถึงการประเมินที่เป็นกลางของคำสั่ง เนื่องจากไม่มีค่าใด ๆ จากชุดการฝึก

ในขณะที่งานการ ระบุ อัตโนมัติ / ความเป็นวัตถุรายงานการลอยตัวภายในช่วง[0.0, 1.0]ที่0.0เป็นประโยชน์ที่เป็น กลางและ1.0เป็นอัตโนมัติมาก

เมื่อนำเข้าแล้ว เราจะโหลดประโยชน์เพื่อวิเคราะห์และสร้างอินสแตนซ์ของTextBlobวัตถุ รวมทั้งกำหนด sentimentคุณสมบัติให้กับของเรางานanalysis:

คุณสมบัติsentimentเป็น a ของ แบบnamedtupleฟอร์มSentiment(polarity, subjectivity)

ผลลัพธ์ที่คาดหวังของการวิเคราะห์คือ:

สิ่งที่ยอดเยี่ยมอย่างหนึ่งเกี่ยวกับ TextBlob คือช่วยให้ผู้ใช้สามารถเลือกอัลกอริธึมสำหรับการใช้งาน NLP ระดับสูงได้:

PatternAnalyzer- ตัวแยกประเภทเริ่มต้นที่สร้างขึ้นบนไลบรารีรูปแบบ

NaiveBayesAnalyzer- โมเดล NLTK ที่ได้รับการฝึกอบรมเกี่ยวกับคลังบทวิจารณ์ภาษาไทย

4.4 ความก้าวหน้า 4 สร้างเหมืองข้อมูล และทดสอบ

ชุดข้อมูลนี้รวมมาจากแหล่งข่าวต่างๆ และได้คำอธิบายประกอบโดยนักบันทึกย่อที่เป็นมนุษย์สามคนซึ่งเป็นผู้เชี่ยวชาญในเรื่อง แต่ละพาดหัวข่าวได้รับการประเมินในมิติต่างๆ เช่น หากพาดหัวข่าวเป็นข่าวที่เกี่ยวข้องกับราคา ทิศทางของการเคลื่อนไหวของราคาที่พูดถึงคืออะไร ไม่ว่าพาดหัวข่าวจะพูดถึงอดีตหรืออนาคต รายการข่าวกำลังพูดถึงการเปรียบเทียบสินทรัพย์หรือไม่ เป็นต้น

Dates	URL	News	Price Direction Up	Price Direction Constant	Price Direction Down	Asset Comparision	Past Information	Future Information	PriceSentiment
0 28/1/16	http://www.marketwatch.com/story/april-gold-down-20-cents-to-settle-at-\$1,116.1...	april gold down 20 cents to settle at \$1,116.1...	0.0	0.0	1.0	0.0	1.0	0.0	negative
1 13/9/17	http://www.marketwatch.com/story/gold-prices-s...	gold suffers third straight daily decline	0.0	0.0	1.0	0.0	1.0	0.0	negative
2 26/7/16	http://www.marketwatch.com/story/gold-futures-...	Gold futures edge up after two-session decline	1.0	0.0	0.0	0.0	1.0	0.0	positive
3 28/2/18	https://www.metaldaily.com/link/277199/dent-re...	dent research : is gold's day in the sun comin...	0.0	0.0	0.0	0.0	0.0	1.0	none
4 6/9/17	http://www.marketwatch.com/story/gold-steadies...	Gold snaps three-day rally as Trump, lawmakers...	0.0	0.0	1.0	0.0	1.0	0.0	negative
5 16/8/16	http://www.marketwatch.com/story/dec-gold-clim...	Dec. gold climbs \$9.40, or 0.7%, to settle at ...	1.0	0.0	0.0	0.0	1.0	0.0	positive
6 24/9/13	https://economictimes.indiatimes.com/markets/c...	gold falls by rs 25 on sluggish demand, global...	0.0	0.0	1.0	0.0	1.0	0.0	negative

ในช่วงไม่กี่ปีที่ผ่านมา มีการใช้วิธีการเรียนรู้ด้วยเครื่องเพื่อดึงข้อมูลจากกระasseข่าวในโดเมนการเงิน อย่างไรก็ตาม ข้อมูลนี้ส่วนใหญ่อยู่ในรูปแบบของความรู้สึกทางการเงินที่มีอยู่ในหัวข้อข่าว โดยเฉพาะราคาหุ้น ในงานปัจจุบันของเรา เราขอเสนอว่าสามารถดึงข้อมูลมิติอื่นๆ ที่หลากหลายจากหัวข้อข่าว ซึ่งจะเป็นที่สนใจของนักลงทุน ผู้กำหนดนโยบาย และผู้ปฏิบัติงานอื่นๆ เราเสนอกรอบการทำงานที่ดึงข้อมูล เช่น การเคลื่อนไหวในอดีตและทิศทางที่คาดหวังในด้านราคา การเปรียบเทียบสินทรัพย์ และข้อมูลทั่วไปอื่นๆ ที่ข่าวกล่าวถึง เราใช้กรอบการทำงานนี้กับสินค้าโภคภัณฑ์ "ทองคำ" และฝึกโมเดลการเรียนรู้ของเครื่องโดยใช้ชุดข้อมูลหัวข่าวที่มีคำอธิบายประกอบโดยมนุษย์ 11,412 รายการ (เผยแพร่พร้อมกับการศึกษา) รวมรวมตั้งแต่ช่วง พ.ศ.

2543-2562 เรายอดลองเพื่อตรวจสอบผลกระทบของกระแสข่าวที่มีต่อราคาทองคำ และสังเกตว่าข้อมูลที่ผลิตจากกรอบของเรางานส่งผลกระทบอย่างมีนัยสำคัญต่อราคาทองคำในอนาคต

4.5 ความก้าวหน้า 5 ทดสอบ Model Training and Test datasets.

ขั้นตอนการทำงาน มี 7 ขั้นตอน

Step 1 - Loading the required libraries and modules.

Step 2 - Loading the data and performing basic data checks.

Step 3 - Pre-processing the raw text and getting it ready for machine learning.

Step 4 - Creating the Training and Test datasets.

Step 5 - Converting text to word frequency vectors with TfidfVectorizer.

Step 6 - Create and fit the classifier.

Step 7 - Computing the evaluation metrics.

4.5.1 Step 1 - Loading the Required Libraries and Modules

```
# Import required libraries
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.utils import shuffle

%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

4.5.2 Step 2 - Loading the Data and Performing Basic Data Checks

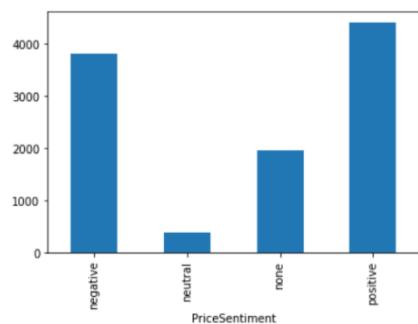
โค้ด บรรทัดแรกอ่านในข้อมูลเป็น data frame ของ pandas ในขณะที่บรรทัดที่สองพิมพ์รูปร่าง – 10,570 การสังเกตจาก 10 ตัวแปร บรรทัดที่สามพิมพ์ข้อสังเกตห้าข้อแรก

```
df = pd.read_csv('gold-dataset-sinha-khandait.csv')
print(df.shape)
df.head()
```

(10570, 10)

	Dates	URL	News	Price Direction Up	Price Direction Constant	Price Direction Down	Asset Comparision	Past Information	Future Information	PriceSentiment
0	28/1/16	http://www.marketwatch.com/story/april-gold-down-20-cents-to-settle-at-\$1,116.1...	april gold down 20 cents to settle at \$1,116.1...	0	0	1	0	1	0	negative
1	13/9/17	http://www.marketwatch.com/story/gold-prices-s...	gold suffers third straight daily decline	0	0	1	0	1	0	negative
2	26/7/16	http://www.marketwatch.com/story/gold-futures-...	Gold futures edge up after two-session decline	1	0	0	0	1	0	positive
3	28/2/18	https://www.metalsdaily.com/link/277199/dent-re...	dent research : is gold's day in the sun comin...	0	0	0	0	0	1	none
4	6/9/17	http://www.marketwatch.com/story/gold-steadies...	Gold snaps three-day rally as Trump, lawmakers...	0	0	1	0	1	0	negative

```
df.groupby(df['PriceSentiment']).News.count().plot.bar(ylim=0)
plt.show()
print(4533/10570) #Baseline accuracy
```



0.4288552507095553

4.5.3 Step 3 – Pre-processing the Raw Text and Getting It Ready for Machine Learning

ขั้นตอนก่อนการประมวลผลทั่วไปคือ:

การลบเครื่องหมายวรรคตอน - หลักการทั่วไปคือการลบทุกอย่างที่ไม่อยู่ในรูปแบบ x,y,z

การนำคำหยุดออก - คำเหล่านี้เป็นคำที่ไม่มีประโยชน์ เช่น 'the', 'is', 'at' ลิ่งเหล่านี้ไม่มีประโยชน์ เพราะ

ความถี่ของคำหยุดนั้นอยู่ในคลังข้อมูลสูง แต่ก็ไม่ได้ช่วยในการสร้างความแตกต่างของคลาสเป้าหมาย การลบ Stopwords ยังช่วยลดขนาดข้อมูลอีกด้วย

การแปลงเป็นตัวพิมพ์เล็ก - คำเช่น 'คลินิก' และ 'คลินิก' จำเป็นต้องถือเป็นคำเดียวกัน ดังนั้นลิ่งเหล่านี้จะถูกแปลงเป็นตัวพิมพ์เล็ก

Stemming - เป้าหมายของ Stemming คือการลดจำนวนรูปแบบการผันคำที่ปรากฏในข้อความ ซึ่งทำให้คำต่างๆ เช่น "argue", "argued", "arguing", "argues" ถูกลดthonเป็นคำว่า "argu" ซึ่งช่วยในการลดขนาดของ

พื้นที่คำศัพท์ มีหลายวิธีในการดำเนินการ Stemming วิธีที่นิยมเป็นวิธี “Porter Stemmer” โดย Martin Porter

สำหรับการทำตามขั้นตอนที่กล่าวไว้ข้างต้นให้เสร็จสิ้น เราจะต้องโหลดแพ็คเกจ nltk ซึ่งทำในโค้ดบรรทัดแรก ด้านล่าง บรรทัดที่สองดาวน์โหลดรายการ 'คำหยุด' ในแพ็คเกจ nltk

```
import nltk
nltk.download('stopwords')
```

โค้ด ที่ลีช์บรรทัดที่หกทำหน้าที่ประมวลผลข้อความล่วงหน้าที่กล่าวถึงข้างต้น

```
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import re

stemmer = PorterStemmer()
words = stopwords.words("english")

df['processedtext'] = df['News'].apply(lambda x: " ".join([stemmer.stem(i) for i in re.sub("[^a-zA-Z]", " ", x).split() if i not in words])
```

ตอนนี้เราจะมีชุดข้อมูลที่ประมวลผลล่วงหน้าซึ่งมีคอลัมน์ 'processedtext' ใหม่

```

print(df.shape)
df.head(10)

```

	Dates	URL	News	Price Up	Price Direction Constant	Price Down	Asset Comparision	Past Information	Future Information	Price Sentiment	processed
0	28/1/16	http://www.marketwatch.com/story/april-gold-d... do...	april gold down 20 cents to settle at \$1,116.1...	0	0	1	0	1	0	negative	april gold set
1	13/9/17	http://www.marketwatch.com/story/gold-prices-s... s...	gold suffers third straight daily decline	0	0	1	0	1	0	negative	gold si third stra daili d
2	26/7/16	http://www.marketwatch.com/story/gold-futures-... s...	Gold futures edge up after two-session decline	1	0	0	0	1	0	positive	gold futur two ses di
3	28/2/18	https://www.metalsdaily.com/link/277199/dent-r... s...	dent research : is gold's day in the sun comin...	0	0	0	0	0	1	none	dent rese gold day come :
4	6/9/17	http://www.marketwatch.com/story/gold-steadies... s...	Gold snaps three-day rally as Trump, lawmakers...	0	0	1	0	1	0	negative	gold : three day trump law reac
5	16/8/16	http://www.marketwatch.com/story/dec-gold-clim... s...	Dec. gold climbs \$9.40, or 0.7%, to settle at ...	1	0	0	0	1	0	positive	dec gold c set
6	24/9/13	https://economictimes.indiatimes.com/markets/c... s...	gold falls by rs 25 on sluggish demand, global...	0	0	1	0	1	0	negative	gold f. slug demand gl
7	23/9/16	http://www.marketwatch.com/story/gold-futures-... s...	Gold futures fall for the session, but gain fo...	1	0	1	0	1	0	positive	gold futu session v
8	21/10/12	https://www.thehindubusinessline.com/opinion/c... s...	Gold struggles; silver slides, base metals falter	0	1	0	1	1	0	neutral	gold str silver base n f
9	16/3/18	http://www.marketwatch.com/story/april-gold-ho... s...	april gold holds slight gain, up \$2.50, or 0.2...	1	0	0	0	1	0	positive	april gold slight ga

4.5.4 Step 4 - Creating the Training and Test Datasets

โค้ด บรรทัดแรกด้านล่างนำเข้าโมดูลสำหรับสร้างชุดข้อมูลการฝึกอบรมและทดสอบ บรรทัดที่สองสร้างอาร์เรย์ของตัวแปรเป้าหมาย เรียกว่า 'เป้าหมาย'

บรรทัดที่สามสร้างอาร์เรย์การฝึกอบรม (X_{train} , y_{train}) และชุดการทดสอบ (X_{test} , y_{test}) โดยจะเก็บข้อมูลไว้ 30% สำหรับการทดสอบไม่เดล อาร์กิวเมนต์ 'random_state' ช่วยให้แน่ใจว่า ผลลัพธ์สามารถทำซ้ำได้

บรรทัดที่สี่พิมพ์รูปร่างของชุดข้อมูลโดยรวม การฝึกอบรม และการทดสอบ ตามลำดับ

```

from sklearn.model_selection import train_test_split
target = df['PriceSentiment']
X_train, X_test, y_train, y_test = train_test_split(df['processedtext'], target, test_size=0.30, random_state=100)
print(df.shape); print(X_train.shape); print(X_test.shape)

(10570, 11)
(7399,)
(3171,)

```

4.5.5 Step 5 - Converting Text to Word Frequency Vectors with TfidfVectorizer.

เราได้ประมวลผลข้อความแล้ว แต่เราต้องแปลงเป็นเวกเตอร์ความถี่คำเพื่อสร้างแบบจำลองการเรียนรู้ของเครื่อง มีหลายวิธีในการทำเช่นนี้ เช่น การใช้ CountVectorizer และ HashingVectorizer และ TfidfVectorizer เป็นวิธีที่ได้รับความนิยมมากที่สุด

TF-IDF เป็นตัวย่อที่ย่อมาจาก 'Term Frequency-Inverse Document Frequency' มันถูกใช้เป็นปัจจัยถ่วงน้ำหนักในแอปพลิเคชันการทำเหมืองข้อมูล

Term Frequency (TF): สรุป Term Frequency ที่เป็นมาตรฐานภายในเอกสาร

ความถี่เอกสารผกผัน (IDF): วิธีนี้จะช่วยลดน้ำหนักของคำที่ปรากฏบ่อยในเอกสารต่างๆ พูดง่ายๆ ก็คือ TF-IDF พยายามเน้นคำสำคัญที่มักพบในเอกสารแต่ไม่อยู่ในเอกสาร เราจะพยายามสร้างเวกเตอร์ TF-IDF สำหรับเอกสารของเรา

บรรทัดแรกของโค้ดด้านล่างนำเข้า TfidfVectorizer จากโมดูล 'sklearn.feature_extraction.text' บรรทัดที่สองเริ่มต้นรัวๆ TfidfVectorizer ที่เรียกว่า 'vectorizer_tfidf'

บรรทัดที่สามพอดีและแปลงข้อมูลการฝึก โค้ดบรรทัดที่สี่จะเปลี่ยนข้อมูลการทดสอบ ในขณะที่บรรทัดที่ห้าจะพิมพ์คุณลักษณะ 10 รายการแรก

```

from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer_tfidf = TfidfVectorizer(stop_words='english', max_df=0.7)
train_tfidf = vectorizer_tfidf.fit_transform(X_train.values.astype('U'))
test_tfidf = vectorizer_tfidf.transform(X_test.values.astype('U'))
print(vectorizer_tfidf.get_feature_names()[:10])
['aayog', 'abat', 'abbrevi', 'abc', 'abn', 'acacia', 'acceler', 'access', 'account', 'accredit']

```

```

print(train_tfidf.shape); print(test_tfidf.shape)
(7399, 2698)
(3171, 2698)

```

4.5.6 Step 6 - Create and Fit the Classifier.

ตอนนี้ เราจะสร้างแบบจำลองการจัดประเภทข้อความ อัลกอริทึมที่เราจะเลือกคือ Naive Bayes Classifier ซึ่งมักใช้สำหรับปัญหาการจัดประเภทข้อความ เนื่องจากขึ้นอยู่กับความน่าจะเป็น มันง่ายและมีประสิทธิภาพในการตอบคำถามเช่น "ให้คำเฉพาะในเอกสารอะไรเป็นโอกาส (ความน่าจะเป็น) ที่จะเป็นของชั้นเรียนนั้น"

เราเริ่มต้นด้วยการนำเข้าโมดูลที่จำเป็นซึ่งเสร็จสิ้นในโค้ดสองบรรทัดแรกด้านล่าง บรรทัดที่สามสร้างตัวแยกประเภท Multinomial Naive Bayes เรียกว่า 'nb_classifier' รหัสบรรทัดที่สี่หมายถึงกับตัวแยกประเภทในข้อมูลการฝึกอบรม

สุดท้าย โมเดลของเราได้รับการฝึกฝนและพร้อมที่จะสร้างการคาดการณ์เกี่ยวกับข้อมูลที่มองไม่เห็น การดำเนินการนี้ดำเนินการในโค้ดบรรทัดที่ห้าในขณะที่บรรทัดที่หกจะพิมพ์คลาสที่คาดการณ์ไว้สำหรับระเบียน 10 รายการแรกในข้อมูลการทดสอบ

การประเมิน Naïve Bayes Model

```

from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics

nb_classifier = MultinomialNB()
nb_classifier.fit(train_tfidf, y_train)

pred2 = nb_classifier.predict(test_tfidf)
print(pred2[:10])
['positive' 'negative' 'positive' 'positive' 'negative' 'none' 'none'
 'positive' 'negative' 'neutral']

```

4.5.7 Step 7 - Computing the Evaluation Metrics

ตอนนี้เราพร้อมที่จะประเมินประสิทธิภาพของแบบจำลองของเรา กับข้อมูลการทดสอบแล้ว การใช้ฟังก์ชัน 'metrics.accuracy_score' เรากำหนดความถูกต้องในบรรทัดแรกของโค้ดด้านล่าง และพิมพ์ผลลัพธ์โดยใช้โค้ดบรรทัดที่สอง เราเห็นว่าความแม่นยำอยู่ที่ 73% ซึ่งเป็นคะแนนที่ดี นอกจากนี้เรายังสามารถคำนวณความถูกต้องผ่านตัวชี้วัดความสับสน โค้ดบรรทัดที่สามด้านล่างสร้างตัวชี้วัดความสับสน ซึ่งอธิบายเม้นต์ 'labels' ใช้เพื่อบ่งบอกถึงคลาสเป้าหมาย ('positive', 'negative', 'none', 'neutral' ในกรณีของเรา) บรรทัดที่สี่พิมพ์ตัวชี้วัด

```
# Calculate the accuracy score: score
accuracy_tfidf = metrics.accuracy_score(y_test, pred2)
print(accuracy_tfidf)

Conf_metrics_tfidf = metrics.confusion_matrix(y_test, pred2, labels=['positive', 'negative', 'none', 'neutral'])
print(Conf_metrics_tfidf)

0.7303689687795648
[[1142 151 29 0]
 [ 297 785 37 0]
 [ 172 64 377 0]
 [ 62 40 3 12]]
```

4.6 Building Random Forest Classifier

โค้ดสองบรรทัดแรกด้านล่างนำเข้าโมดูลที่จำเป็น บรรทัดที่สามสร้าง Random Forest Classifier ในขณะที่บรรทัดที่สี่หมายความว่าจะทำการฝึกอบรม

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix

classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 100)
classifier.fit(train_tfidf, y_train)
```

เมื่อการฝึกโมเดลเสร็จสิ้น เราใช้โมเดลเพื่อสร้างการคาดการณ์เกี่ยวกับข้อมูลการทดสอบ ซึ่งจะทำในโค้ดบรรทัดแรกด้านล่าง บรรทัดที่สองพิมพ์คลาสที่คาดการณ์ไว้สำหรับ 10 ระเบียนแรกในข้อมูลการทดสอบ โค้ดบรรทัดที่สามและสี่จะคำนวณและพิมพ์คะแนนความถูกต้องตามลำดับ เราเห็นว่าความแม่นยำเพิ่มขึ้นเหลือ 77.2%

```

predRF = classifier.predict(test_tfidf)
print(predRF[:10])

# Calculate the accuracy score
accuracy_RF = metrics.accuracy_score(y_test, predRF)
print(accuracy_RF)

Conf_metrics_RF = metrics.confusion_matrix(y_test, predRF, labels=['positive', 'negative', 'none', 'neutral'])
print(Conf_metrics_RF)

['positive' 'negative' 'positive' 'positive' 'negative' 'none' 'none'
 'negative' 'positive' 'neutral']
0.7723115736360769
[[1042  206   66    8]
 [ 173  891   50    5]
 [  95   58  456   4]
 [  23   29    5  60]]

```

4.7 ความก้าวหน้า 6 ทดลองทำ Sentiment Analysis Naïve Bayes Classifier

Read into Python

Let's first read the required data from CSV file using Pandas library.

```

import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.utils import shuffle

import numpy as np          #linear algebra
import pandas as pd         # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt #For Visualisation
import seaborn as sns        #For better Visualisation
from bs4 import BeautifulSoup #For Text Parsing

%matplotlib inline
import warnings
warnings.filterwarnings('ignore')

data = pd.read_csv('gold-dataset-sinha-khandait_test.csv')
print(data.shape)
data.head(7)

```

	Dates	URL	News	Price Direction Up	Price Direction Constant	Price Direction Down	Asset Comparision	Past Information	Future Information	PriceSentiment
0	28/1/16	http://www.marketwatch.com/story/april-gold-down-20-cents-to-settle-at-\$1,116.1...	april gold down 20 cents to settle at \$1,116.1...	0.0	0.0	1.0	0.0	1.0	0.0	negative
1	13/9/17	http://www.marketwatch.com/story/gold-prices-s...	gold suffers third straight daily decline	0.0	0.0	1.0	0.0	1.0	0.0	negative
2	26/7/16	http://www.marketwatch.com/story/gold-futures-...	Gold futures edge up after two-session decline	1.0	0.0	0.0	0.0	1.0	0.0	positive
3	28/2/18	https://www.metalsdaily.com/link/277199/dent-re...	dent research : is gold's day in the sun comin...	0.0	0.0	0.0	0.0	0.0	1.0	none
4	6/9/17	http://www.marketwatch.com/story/gold-steadies...	Gold snaps three-day rally as Trump, lawmakers...	0.0	0.0	1.0	0.0	1.0	0.0	negative
5	16/8/16	http://www.marketwatch.com/story/dec-gold-clim...	Dec. gold climbs \$9.40, or 0.7%, to settle at ...	1.0	0.0	0.0	0.0	1.0	0.0	positive
6	24/9/13	https://economictimes.indiatimes.com/markets/c...	gold falls by rs 25 on sluggish demand, global...	0.0	0.0	1.0	0.0	1.0	0.0	negative

```
data.isnull().sum()
```

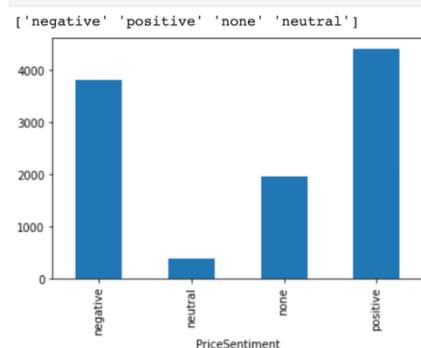
```
Dates          14
URL           14
News          14
Price Direction Up    14
Price Direction Constant 14
Price Direction Down   14
Asset Comparision     14
Past Information      14
Future Information     14
PriceSentiment        14
dtype: int64
```

```
data=data.dropna()
data.isnull().sum()
```

```
Dates          0
URL           0
News          0
Price Direction Up    0
Price Direction Constant 0
Price Direction Down   0
Asset Comparision     0
Past Information      0
Future Information     0
PriceSentiment        0
dtype: int64
```

```
Sentiment = data['PriceSentiment'].unique()
print(Sentiment)

data.groupby(data['PriceSentiment']).News.count().plot.bar(ylim=0)
plt.show()
```



```

import nltk
nltk.download('stopwords')

[nltk_data] Downloading package stopwords to
[nltk_data]      /Users/whatbest/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
True

from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import re

stemmer = PorterStemmer()
words = stopwords.words("english")

data['processedtext'] = data['News'].apply(lambda x: " ".join([stemmer.stem(i) for i in re.sub("[^a-zA-Z]", " ", x).split() if i not in words]))

```

Dates	URL	News	Price Up	Direction	Price Constant	Direction	Price Down	Asset Comparision	Past Information	Future Information	PriceSentiment	processedtext
28/1/16	http://www.marketwatch.com/story/april-gold-doesnt-settle-at-new-high-11161116	april gold down 20 cents to settle at \$1,116.1...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	negative	april gold cent settl oz	
13/9/17	http://www.marketwatch.com/story/gold-prices-suffer-third-straight-daily-decline-1500913	gold suffers third straight daily decline	0.0	0.0	1.0	0.0	1.0	0.0	0.0	negative	gold suffer third straight daili declin	
26/7/16	http://www.marketwatch.com/story/gold-futures-edge-up-after-two-session-decline-141426	Gold futures edge up after two-session decline	1.0	0.0	0.0	0.0	1.0	0.0	0.0	positive	gold futur edg two session declin	
28/2/18	https://www.metaldaily.com/link/277199/dent-research-is-golds-day-in-the-sun-coming-soon	dent research : is gold's day in the sun comin...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	none	dent research gold day sun come soon	
6/9/17	http://www.marketwatch.com/story/gold-steadies-149606	Gold snaps three-day rally as Trump, lawmakers...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	negative	gold snap three day ralli trump lawmak reach d...	
16/8/16	http://www.marketwatch.com/story/dec-gold-climbs-140816	Dec. gold climbs \$9.40, or 0.7%, to settle at ...	1.0	0.0	0.0	0.0	1.0	0.0	0.0	positive	dec gold climb settl oz	
24/9/13	https://economictimes.indiatimes.com/markets/commodities/gold-falls-by-rs-25-on-sluggish-demand-global-cue-130924	gold falls by rs 25 on sluggish demand, global...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	negative	gold fall rs sluggish demand global cue	
23/9/16	http://www.marketwatch.com/story/gold-futures-fall-for-the-session-but-gain-for-the-week-140923	Gold futures fall for the session, but gain fo...	1.0	0.0	1.0	0.0	1.0	0.0	0.0	positive	gold futur fall session gain week	

Pre-process Data

We need to remove package name as it's not relevant. Then convert text to lowercase for CSV data. So, this is data pre-process stage.

```
def preprocess_data(data):
    # Remove package name as it's not relevant
    #data = data.drop('News', axis=1)

    # Convert text to lowercase
    data['processedtext'] = data['processedtext'].str.strip().str.lower()

    return data

data = preprocess_data(data)
```

Splitting Data

First, separate the columns into dependent and independent variables (or features and label). Then you split those variables into train and test set.

```
df = data
# Split into training and testing data
x = data['processedtext']
y = data['PriceSentiment']
x, x_test, y, y_test = train_test_split(x,y, stratify=y, test_size=0.25, random_state=42)
```

Vectorize text reviews to numbers.

```
# Vectorize text reviews to numbers
vec = CountVectorizer(stop_words='english')
x = vec.fit_transform(x).toarray()
x_test = vec.transform(x_test).toarray()
```

Model Generation

After splitting and vectorize text reviews to number, we will generate a random forest model on the training set and perform prediction on test set features.

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x, y)

MultinomialNB()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

Evaluating Model

After model generation, check the accuracy using actual and predicted values.

```
model.score(x_test, y_test)*100
75.89996210685865
```

Then check prediction...

```
model.predict(vec.transform(['gold reversal buoys bay street']))
array(['positive'], dtype='<U8')

print(8/14*100)
57.14285714285714

import joblib
joblib.dump(model, 'model.pkl')

['model.pkl']
```

4.8 ความก้าวหน้า 7 ปรับแก้ไข Sentiment Analysis Naïve Bayes Classifier

4.8.1 การเปรียบเทียบ โดยใช้ Model Naïve baye

โดยการเปรียบเทียบระหว่างใช้ ข่าวสารที่นำมาผ่านกระบวนการ NLP และ ข่าวสารดังเดิมจะมี % ความแม่นยำที่แตกต่างกัน โดยที่ % ความแม่นยำจากการ Test โดยใช้ Model Naïve baye ข่าวสารที่นำมาผ่านระบบได้วันการ NLP จะได้ความแม่นยำอยู่ที่ 75.55% และ ข่าวสารดังเดิม จะได้ความแม่นยำอยู่ที่ 76.35%

```
# df = data
# Split into training and testing data
x = data['News']
y = data['PriceSentiment']
x_train, x_test, y_train, y_test = train_test_split(x,y, stratify=y, test_size=0.2, random_state=100)

[187]: ✓ 0.6s Python

Vectorize text reviews to numbers.

# Vectorize text reviews to numbers
vec = CountVectorizer(stop_words='english')
x = vec.fit_transform(x).toarray()
x_test = vec.transform(x_test).toarray()

[188]: ✓ 0.2s Python

from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x, y)

[189]: ✓ 1.6s Python
... + MultinomialNB
MultinomialNB()

[190]: model.score(x_test, y_test)*100
[190]: ✓ 0.7s Python
... 76.35
```

ภาพที่ 13 ข่าวสารดังเดิม

```

df = data
# Split into training and testing data
x = data['processedtext']
y = data['PriceSentiment']
x, x_test, y, y_test = train_test_split(x,y, stratify=y, test_size=0.2, random_state=100)

[124] ✓ 0.7s Python

Vectorize text reviews to numbers.

# Vectorize text reviews to numbers
vec = CountVectorizer(stop_words='english')
x = vec.fit_transform(x).toarray()
x_test = vec.transform(x_test).toarray()

[125] ✓ 0.1s Python

from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x, y)

[126] ✓ 0.8s Python
... + MultinomialNB
MultinomialNB()

model.score(x_test, y_test)*100

[127] ✓ 0.1s Python Python
... 79.55

```

ภาพที่ 14 ข่าวสาร ผ่านกระบวนการNLP

4.8.2 การเปรียบเทียบโดยใช้ข่าวสารที่ไม่ได้รับการเทรน โดยใช้ Model Naïve baye

โดยการเปรียบเทียบเป็นแบบข้อมูลจากระบบ Data set โดยที่ Data set มีข้อมูลทั้งหมด

10,570 record โดย 10,000 record ทำการใช้ในการ Train 80% และ test 20% และ 570 record ใช้ใน การทดสอบจริง โดยความแม่นยำที่ได้ ข่าวสารดังเดิม 75.26% และ ข่าวสาร ผ่านกระบวนการNLP 64.35%

```

Then check prediction...

from itertools import count
import pandas as pd
df = pd.read_csv('gold-dataset-sinha-khandait.csv', sep=',', header=None)
start = 10000
end = 10570
df = df[start - 1:end - 1]
correct = 0
for i in range(len(df)):
    (df.values[i][2])
    (model.predict(vec.transform([df.values[i][2]])), df.values[i][9] == model.predict(vec.transform([df.values[i][2]])))

    if df.values[i][9] == model.predict(vec.transform([df.values[i][2]])):
        correct += 1
print(correct / len(df) * 100)

[166] ✓ 0.5s Python
... 75.26315789473685

```

ภาพที่ 15 ข่าวสารดังเดิม

```

    ▷ ✓ from itertools import count
      import pandas as pd
      df = pd.read_csv('gold-dataset-sinha-khandait.csv', sep=',', header=None)
      start = 10000
      end = 10570
      df = df[start - 1:end - 1]
      correct = 0
      for i in range(len(df)):
          (df.values[i][2])
          (model.predict(vec.transform([df.values[i][2]])), df.values[i][9] == model.predict(vec.transform([df.values[i][2]])))
      if df.values[i][9] == model.predict(vec.transform([df.values[i][2]])):
          correct += 1
      print(correct / len(df) * 100 )
[181] ✓ 0.5s
... 64.3859649122807

```

ภาพที่ 16 ข่าวสาร ผ่านกระบวนการNLP

4.9 ความก้าวหน้า 8 pywebio

ได้ทำการแปลงไฟล์ Jupyter เป็น Python เนื่องจากการที่จะทำ Web Application จะต้องใช้ไฟล์ Python ก่อน

```

64 | ##pywebio##
65 | from pywebio.input import input, FLOAT, TEXT
66 | from pywebio.output import put_text
67 | def main():
68 |     ## Model Generation
69 |     from sklearn.naive_bayes import MultinomialNB
70 |
71 |     model = MultinomialNB()
72 |     model.fit(x, y)
73 |     from itertools import count
74 |     import pandas as pd
75 |     df = pd.read_csv('gold-dataset-sinha-khandait.csv', sep=',', header=None)
76 |     start = 10000
77 |     end = 10570
78 |     df = df[start - 1:end - 1]
79 |     correct = 0
80 |     str = input('This is label', type=TEXT, placeholder='This is placeholder',
81 |                help_text='This is help text', required=True)
82 |     put_text(model.predict(vec.transform([str])))
83 |     for i in range(len(df)):
84 |         print(df.values[i][2])
85 |         put_text(model.predict(vec.transform([df.values[i][2]])), df.values[i][9] == model.predict(vec.transform([df.values[i][2]])))
86 |
87 |         if df.values[i][9] == model.predict(vec.transform([df.values[i][2]])):
88 |             correct += 1
89 |
90 |     print(correct / len(df) * 100 )
91 |
92 |
93 | if __name__ == '__main__':
94 |     main()
95 |
96 |

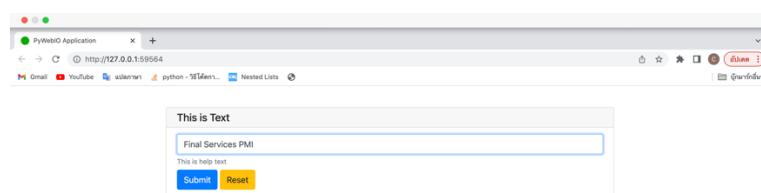
```

บรรทัดที่ 65 และ 66 เป็นการเรียกใช้ pywebio input, output

บรรทัดที่ 80 เป็นการเรียกใช้ Input คือการสร้าง Label เพื่อให้ป้อนข้อมูลที่จะนำไปใช้กับ Model

บรรทัดที่ 82, 85 เป็นการเรียกใช้ Output คือการแสดงผลหลังจากที่รับ Input เข้าไป

ผลการทดลอง



```
['negative']
['none'] [ True]
['positive'] [ True]
['positive'] [ True]
['negative'] [ True]
['positive'] [ True]
['positive'] [ True]
['none'] [ True]
```

A screenshot of a web browser window titled 'PyWebIO Application'. The URL is 'http://127.0.0.1:59564'. The page displays a series of alternating strings: 'negative', 'none', 'positive', 'positive', 'negative', 'positive', 'positive', and 'none'. Each string is followed by a small square containing a checkmark or an 'X'.

บทที่ 5 สรุป

5.1 สรุปผลการดำเนินงาน

1. สามารถสรุปบทความได้ ใช้ Library NLTK NLP
2. สามารถวิเคราะห์ บทความได้ระดับเบื้องต้น
3. จัดทำเหมืองข้อมูล Dataset
4. เปรียบเทียบอัลกอริทึมระหว่าง Random Forest Classifier กับ Naïve Bayes Classifier
5. ทำ Sentiment Analysis Naïve Bayes Classifier
6. ทำการเปรียบเทียบระหว่างข่าวสารตั้งเดิม และ ข่าวสารที่ผ่านระบบ NLP
7. มีการนำ pywebio เข้ามาในการทำ Web Application

5.2 ปัญหาและอุปสรรค

1. ยังใช้ภาษา Python ได้ไม่ชำนาญ
2. Library ที่จะใช้ค่อนข้างเยอะ จึงทำให้เริ่ม Project ช้า
3. มีปัญหาเรื่อง Version ที่จะติดตั้ง Library
4. ไม่สามารถนำ ตัวเลขที่ข่าวประกาศไปวิเคราะห์ หรือ เก็บข้อมูลได้เนื่องจากตอนนี้การทราบนิ่ง ต้องใช้ตัวอักษรภาษาอังกฤษเท่านั้น
5. ผลจากการทำ Sentiment Analysis ผลที่ได้รับ ไม่ต้องกับที่เหล่านะในระบบ
6. เนื่องจากการทำ Web Application โดยใช้ Python มีตัวเลือก Framework ให้ใช้น้อย
7. เนื่องจากข่าวสารที่ใช้ในการทดสอบ มีตัวเลขและบางประโยคสำคัญทำให้ NLP ตัดบางส่วนที่สำคัญไป

5.3 งานที่จะดำเนินการต่อไป

1. ปรับปรุงแก้ไขการทำ sentiment Analysis
2. เทคนิคข้อมูลที่เกี่ยวกับ Forex
3. ทดสอบ Algorithm ระหว่าง SpaCy กับ NLTK
4. ศึกษาและทดลองอัลกอริทึมอื่นๆ
5. จัดทำเนื้อหาข่าวสารเพื่อมาแทน ทำให้ประสิทธิภาพดียิ่งขึ้น
6. จัดทำ Web Application ให้สมบูรณ์
7. ศึกษา Framework Django

บรรณานุกรม

- [1] S. Shalev-Shwartz, S. Ben-David, [Understanding Machine Learning: From Theory to Algorithms](#) (2014), Cambridge University Press , เข้าถึงล่าสุด 15 มกราคม 2565
- [2] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado, and J. Dean. [Distributed Representations of Words and Phrases and their Compositionality](#) (2013), Advances in Neural Information Processing Systems 26 เข้าถึงล่าสุด 15 มกราคม 2565
- [3] J. Pennington, R. Socher, and C. D. Manning, [GloVe: Global Vectors for Word Representation](#) (2014), In EMNLP. เข้าถึงล่าสุด 16 มกราคม 2565
- [4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. [Enriching word vectors with subword information](#) (2016), arXiv preprint เข้าถึงล่าสุด 17 มกราคม 2565
- [5] NLP Implementations : URL :
<https://towardsdatascience.com/gentle-start-to-natural-language-processing-using-python-6e46c07addf3> เข้าถึงล่าสุด 18 มกราคม 2565
- [6] The theory you need to know before you start an NLP : URL :
<https://towardsdatascience.com/the-theory-you-need-to-know-before-you-start-an-nlp-project-1890f5bbb793> เข้าถึงล่าสุด 12 มีนาคม 2565
- [7] Us Department of labor : URL : <https://www.dol.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565
- [8] Energy information Administration : URL : <https://www.eia.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565
- [9] กองทุน SPDR : URL : <https://traderider.com/forex/spdr-%E0%B8%81%E0%B8%AD%E0%B8%87%E0%B8%97%E0%B8%B8%E0%B8%99%E0%>

[B8%97%E0%B8%AD%E0%B8%87%E0%B8%84%E0%B8%B3%E0%B9%81%E0%B8%97
%E0%B9%88%E0%B8%87](#) เข้าถึงล่าสุด 12 มีนาคม 2565

[10] Federal Reserve : URL : <https://www.federalreserve.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565

[11] Bloomberg : URL : <https://www.bloomberg.com/asia> เข้าถึงล่าสุด 12 มีนาคม 2565

[12] Twitter : URL : <https://twitter.com/> เข้าถึงล่าสุด 12 มีนาคม 2562

[13] <https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP> เข้าถึงล่าสุด 12 สิงหาคม 2562

[14] Sentiment Analysis of Commodity News (Gold) : URL :
<https://www.kaggle.com/datasets/ankurzing/sentiment-analysis-in-commodity-market-gold> เข้าถึงล่าสุด 1 กันยายน 2562