

Deep Learning Stock Trading Advisor

โครงการหมายเลข 46

นายวิศรุต	ดวงมรกต	57070501040	wisarut.bank@gmail.com
นายโชติธัช	ดวงเจริญทิพย์	57070501059	shotitouch@gmail.com
นางสาวพิชญ์สินี	สินเจริญพงศ์	57070501073	phitsinee.sincharoen@gmail.com

ที่ปรึกษาโครงการ

ดร.ขจรพงษ์ อัครจิตสกุล

ที่ปรึกษาร่วม

ผศ.ดร.มารอง ผดุงสิทธิ์

วันที่ 20 พฤษภาคม 2561

ข้าพเจ้าได้อ่านรายงานและตรวจเนื้อหาของรายงานเรียบร้อยแล้ว

Project Title	Deep Learning Stock Trading Advisor
Project Credit	3 credits
Project Participant	Mr. Wisarut Duangorakot Mr. Shotitouch Tuangcharoentip Mrs. Phitsinee Sincharoenpong
Advisor	Dr. Khajonpong Akkarajitsakul
Co-Advisor	Dr. Marong Phadoongsidhi
Degree of Study	Bachelor's Degree
Department	Computer Engineering
Academic Year	2017

Abstract

Nowadays, Artificial Intelligence (AI) has been involved and become significant in many fields of the country's development. However, only a small part of this technology has been applied to finance technology. Moreover, another problem is that most of Thais consider stock trading as difficult and risky investment.

Thus, this project was started due to the aforementioned reason. It is intended to be a new stock trading product which applies the concept of Artificial Intelligence. In this project, we developed models using Machine Learning and made them return as much profit as possible in stock market. Each model calculated good buy/sell signals of stocks in SET50. Our aim was to gain more investment return than what an investor will get by using buy-and-hold strategy. We used back-testing in order to test our proposed algorithms' performance with testing data collected from 2016 to 2017.

This project was developed by using 3 kinds of machine learning techniques which are Tree learning, Neural network, and Reinforcement learning.

As a result, we were able to build a model which yield higher return than buy-and-hold strategy by 10.53 percent during testing period. We also found that the more complex our model was, the higher return yielded.

หัวข้อโครงการ	Deep Learning Stock Trading Advisor
หน่วยกิตของโครงการ	3 หน่วยกิต
จัดทำโดย	นายวิศรุต ดวงมรกต นายโชติธัช ดวงเจริญทิพย์ นางสาวพิชญ์สินี สีนเจริญพงศ์
อาจารย์ที่ปรึกษา	ดร.ขจรพงษ์ อัครจิตสกุล
อาจารย์ที่ปรึกษาร่วม	ผศ.ดร.มารอง ผดุงสิทธิ์
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2560

บทคัดย่อ

ปัจจุบันนี้ เทคโนโลยีปัญญาประดิษฐ์กำลังเข้ามามีบทบาทสำคัญต่อการพัฒนาประเทศ อย่างไรก็ตาม มีเพียงส่วนน้อยเท่านั้นที่ถูกนำไปใช้กับสาขาเทคโนโลยีการเงิน นอกจากนี้ อีกปัญหาที่เกิดขึ้นคือคนไทยส่วนมากยังเห็นว่าการลงทุนในหุ้นเป็นเรื่องที่ยากและมีความเสี่ยง

ดังนั้น โครงการนี้จึงเกิดขึ้นเนื่องมาจากปัญหาที่กล่าวมาข้างต้น ซึ่งเป็นผลงานที่นำเทคโนโลยีปัญญาประดิษฐ์มาใช้สร้างผลิตภัณฑ์ที่ช่วยการลงทุนในหุ้น ผลิตภัณฑ์โครงการนี้เกิดจากคณะผู้จัดทำใช้เทคโนโลยีการเรียนรู้ของเครื่องในการสร้างโมเดลปัญญาประดิษฐ์ที่สามารถส่งสัญญาณซื้อขายที่ดีใน SET50 และสามารถสร้างผลตอบแทนที่มากกว่าวิธีการถือยาว คณะผู้จัดทำใช้ระบบทดสอบย้อนหลังในการวัดประสิทธิภาพระบบเทรดต่างๆ ด้วยข้อมูลหุ้นปี พ.ศ.2559 - 2560

โครงการนี้จะนำการเรียนรู้ของเครื่อง 3 ประเภทมาใช้ในการสร้างระบบเทรด ได้แก่ การเรียนรู้แบบต้นไม้ตัดสินใจ การเรียนรู้แบบโครงข่ายประสาทเทียม และการเรียนรู้แบบเสริมกำลัง

จากผลการวิจัยของโครงการ คณะผู้จัดทำสามารถสร้างปัญญาประดิษฐ์ที่สามารถเทรดหุ้นแล้วให้ผลตอบแทนมากกว่าการถือยาวในช่วงเวลาการทดสอบที่กำหนด ร้อยละ 10.53 นอกจากนี้ คณะผู้จัดทำยังพบว่า ยิ่งโมเดลที่นำมาใช้มีความซับซ้อนเท่าไรก็สามารถให้ผลตอบแทนมากขึ้นเท่านั้น

กิตติกรรมประกาศ

โครงการ Deep Learning Stock Trading Advisor สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาจากอาจารย์ขจรพงษ์ อัครจิตสกุล อาจารย์ที่ปรึกษาโครงการ และอาจารย์มารอง ผดุงสิทธิ์ อาจารย์ที่ปรึกษาร่วมของโครงการ ที่ได้ให้ทั้งความรู้ คำแนะนำและคำปรึกษา รวมถึงการปรับปรุงแก้ไขข้อบกพร่องต่างๆ ของโครงการด้วยความเอาใจใส่เป็นอย่างดี ผู้จัดทำตระหนักถึงความตั้งใจและความทุ่มเทของอาจารย์ จึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณ คุณอนนต์ อุ่นสินมัน ผู้อำนวยการศูนย์ทดสอบและพัฒนาฟินเทคแห่งชาติ (National Fintech Sandbox - F13) ที่ได้เอื้อเฟื้อสถานที่ในการศึกษาค้นคว้าและพัฒนาโครงการ บริเวณชั้น 13 ของอาคาร KX (Knowledge Exchange) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี นอกจากนี้ ยังให้คำแนะนำและข้อมูลที่เป็นต่อการทำโครงการ

ขอขอบคุณรุ่นพี่ภาควิชาวิศวกรรมคอมพิวเตอร์ที่ทำงานอยู่บริษัท Market Anyware ที่ได้ให้ความรู้และคำแนะนำต่างๆ ในการทำโครงการเรื่องนี้

ขอขอบคุณรุ่นพี่ในห้องแอดมิน ที่ให้ยืมอุปกรณ์ในการทำโครงการ

สุดท้ายนี้ ขอขอบคุณเพื่อนๆ นักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สำหรับการแลกเปลี่ยนความรู้และความคิดเห็นในการทำโครงการเรื่องนี้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ที่มาของปัญหาและแนวทางการแก้ปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน	2
บทที่ 2 ที่มาทฤษฎีและงานวิจัยที่เกี่ยวข้อง	6
2.1 แนวคิดทางทฤษฎี	6
2.1.1 หุ่น	6
2.1.2 การเรียนรู้ของเครื่อง (Machine learning)	7
2.1.3 ระบบทดสอบย้อนหลัง (Backtest)	10
2.2 โครงสร้างข้อมูลและอัลกอริธึม	11
2.2.1 โครงข่ายประสาท (Neural network)	11
2.2.2 การเรียนรู้ต้นไม้ตัดสินใจ (Decision tree learning)	11
2.2.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement learning)	12
2.2.4 การเรียนรู้เชิงลึก (Deep learning)	13
2.3 ภาษาโปรแกรมและเครื่องมือซอฟต์แวร์ที่ใช้	14
2.3.1 ภาษาไพทอน (Python)	14
2.3.2 แพลตฟอร์มจูปิเตอร์โน้ตบุ๊ก (Jupyter notebook IDE)	15
2.3.3 โปรแกรมอานาคอนด้า (Anaconda)	16
2.4 ประเภทของงานวิจัยที่เกี่ยวข้อง	16
2.4.1 เครื่องมือหุ่นเอาวา (AVA)	16
2.4.2 เครื่องมือหุ่นบัวหลวงไอโปรแกรมเทรด (Bualuang iProgram Trade)	18
บทที่ 3 การออกแบบและระเบียบวิธีวิจัย	19
3.1 หลักการทำงานของ Deep Learning Stock Trading Advisor	19
3.2 การออกแบบและสร้างโมเดล	20
3.3 ตัวอย่างแอปพลิเคชันสามารถนำโครงการไปปรับใช้ได้	23
3.4 Usecase diagram	29

สารบัญ

หน้า

บทที่ 4 ผลการวิจัยและอภิปรายผล	30
4.1 การสร้างระบบการทดสอบย้อนหลัง	30
4.2 ข้อมูลที่ใช้ในการสร้างโมเดล	31
4.3 การทดสอบโมเดล	31
4.4 ขั้นตอนการสร้างและผลการทดสอบของแต่ละโมเดล	32
4.5 สรุปผล	47
4.6 อภิปรายผลการทดลอง	47
บทที่ 5 สรุปผล	49
5.1 ผลสำเร็จของโครงการ	49
5.2 ปัญหาที่พบในการทำงาน	49
บรรณานุกรม	51

รายการรูปประกอบ

หน้า

รูปที่ 2.1 หลักการทำงานของการเรียนรู้แบบมีผู้สอน	8
รูปที่ 2.2 หลักการทำงานของการเรียนรู้แบบไม่มีผู้สอน	9
รูปที่ 2.3 หลักการทำงานของ Reinforcement Learning	12
รูปที่ 2.4 Deep Learning สำหรับการเรียนรู้จดจำใบหน้า	13
รูปที่ 2.5 สัญลักษณ์ภาษา Python	14
รูปที่ 2.6 สัญลักษณ์ Jupyter-notebook IDE	15
รูปที่ 2.7 ตัวอย่างการใช้งาน Jupyter-notebook IDE	16
รูปที่ 2.8 โปรแกรม Anaconda	16
รูปที่ 2.9 แอปพลิเคชัน AVA Advisor	17
รูปที่ 2.10 ตัวอย่างของแอปพลิเคชัน AVA Advisor	17
รูปที่ 2.11 โปรแกรม Bualuang iProgram Trade	18
รูปที่ 3.1 ระบบการทำงานภาพรวมของ Deep Learning Stock Trading Advisor	19
รูปที่ 3.2 ภาพแสดงขั้นตอนการทำงานภาพรวมของระบบ	21
รูปที่ 3.3 หน้า Log In ของเว็บแอปพลิเคชัน	23
รูปที่ 3.4 หน้าดูระบบเทรดที่มีอยู่	24
รูปที่ 3.5 หน้ากำหนดจำนวนเงินต้น	25
รูปที่ 3.6 หน้ากำหนดหุ้นที่ต้องการพิจารณา	25
รูปที่ 3.7 หน้ากำหนดเงื่อนไขการคัดกรองหุ้น	26
รูปที่ 3.8 หน้ากำหนดเงื่อนไขสัญญาณการซื้อหุ้น	27
รูปที่ 3.9 หน้ากำหนดเงื่อนไขสัญญาณการขายหุ้น	27
รูปที่ 3.10 หน้าแสดงผลการทดสอบย้อนหลัง	28
รูปที่ 3.11 Usecase diagram แสดงการใช้งานที่ผู้ใช้สามารถทำกับระบบได้	29
รูปที่ 3.4 หน้าแจ้งผลลัพธ์การทดสอบย้อนหลัง	23
รูปที่ 3.5 ภาพ Usecase diagram แสดงการใช้งานที่ผู้ใช้สามารถทำกับระบบได้	24
รูปที่ 4.1 Raw Data ของหุ้นที่จะนำมาสร้างโมเดลและทดสอบ	27
รูปที่ 4.2 ผลการทดสอบโมเดล DecisionTreeClassifier	33
รูปที่ 4.3 ผลการทดสอบโมเดล RandomForestClassifier	35

รายการรูปประกอบ

หน้า

รูปที่ 4.4 ผลลัพธ์ของข้อมูลหุ้น เมื่อใช้ GaussianMixture ในการจัดกลุ่มข้อมูล	36
รูปที่ 4.5 ผลการทดสอบโมเดล RandomForest ร่วมกับ GaussianMixture	38
รูปที่ 4.6 ผลการทดสอบโมเดล Neural Network	40
รูปที่ 4.7 ตัวอย่างราคาหุ้นที่ RNN regressor predict ได้ กับ ราคาจริงของหุ้น	41
รูปที่ 4.8 ผลการทดสอบโมเดล Recurrent Neural Network	42
รูปที่ 4.9 ผลการทดสอบโมเดล Deep Double Q Reinforcement learning	46

สารบัญตาราง

หน้า

ตารางที่ 1.1 แผนการดำเนินงาน	4
ตารางที่ 4.1 ตารางสรุปผลทดสอบเปรียบเทียบแต่ละโมเดลรวมถึง SET 50 index	47
ตารางที่ 5.1 ตารางแสดงสถานะความสำเร็จของงานในแต่ละส่วน	49

บทที่ 1

บทนำ

1.1 ที่มาของปัญหาและแนวทางการแก้ปัญหา

การเติบโตของบริษัทในปัจจุบันนี้ต้องอาศัยเงินทุนเป็นจำนวนมาก ซึ่งการนำบริษัทเข้าตลาดหลักทรัพย์นั้นจะทำให้เกิดกลไกการลงทุนที่สำคัญ นั่นก็คือการซื้อขายหุ้น ประเทศไทยมีบริษัทที่จดทะเบียนในตลาดหลักทรัพย์แห่งประเทศไทย (SET) ทั้งสิ้น 529 บริษัท รวมถึงบริษัทที่ขับเคลื่อนประเทศไทยหลายบริษัท ไม่ว่าจะเป็น การปิโตรเลียมแห่งประเทศไทย (PTT), บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด (AIS), ธนาคารไทยพาณิชย์ (SCB), ธนาคารกรุงไทย (KBANK), บริษัท ปูนซิเมนต์ไทย จำกัด (SCC) และอื่นๆ อีกมากมายล้วนแต่อยู่ในตลาดหลักทรัพย์ทั้งสิ้น โดยมูลค่าตลาดรวมของ SET เมื่อเดือนกันยายนที่ผ่านมามีมูลค่าสูงถึง 16,471,116.51 ล้านบาท เรียกได้ว่าบริษัทที่สำคัญส่วนมากในประเทศไทยสามารถดำเนินการต่อได้ด้วยเงินทุนจากตลาดหุ้น

ประเทศไทย ณ ปัจจุบันนี้ นักลงทุนสามารถลงทุนในตลาดหุ้นได้ง่ายขึ้น มีระบบการสั่งซื้อซื้อขายหุ้นในอินเทอร์เน็ต มีโบรกเกอร์หลายแห่งให้เลือกใช้บริการ มีแหล่งข้อมูลให้ศึกษาและติดตาม อย่างไรก็ตาม การซื้อขายหุ้นให้มีประสิทธิภาพนั้น จำเป็นต้องอาศัยการศึกษาให้เกิดความเข้าใจอย่างถ่องแท้ อีกทั้งยังต้องคัดกรองข้อมูลให้ถูกต้อง และต้องคอยเฝ้าติดตามข้อมูลหุ้นอยู่ตลอดเวลา สิ่งเหล่านี้กลายมาเป็นปัญหาใหญ่สำหรับนักลงทุนหลายคน ซึ่งก็คือ การใช้เวลาจำนวนมากในการศึกษาและติดตามหุ้น หากว่านักลงทุนไม่ได้ศึกษาเกี่ยวกับหุ้นให้ดีแล้วการลงทุนนั้นจะมีความเสี่ยงเป็นอย่างมาก การลงทุนในกองทุนที่ให้มีอาชีพเป็นผู้ดูแลจึงเป็นการลงทุนที่มีความเสี่ยงต่ำกว่า

ข้อมูลจากสำนักงานสถิติแห่งชาติเปิดเผยว่า คนไทยมีเวลาว่างเฉลี่ยเพียงคนละ 3.6 ชั่วโมงต่อวันเท่านั้น ซึ่งเวลาเหล่านี้มักถูกใช้ไปกับการผ่อนคลายต่างๆ แต่การซื้อขายหุ้นต้องใช้เวลามาก สมมติว่าใช้เวลาเพียงวันละ 1 ชั่วโมง ก็เท่ากับใช้เวลาไปถึง 1 ใน 3 ของเวลาว่างทั้งหมดแล้ว คนส่วนมากจึงเลือกที่จะไม่ซื้อขายหุ้น อย่างไรก็ตาม ยังมีคนบางกลุ่มที่ต้องการซื้อขายหุ้นอย่างจริงจังแล้วหันมาซื้อขายหุ้นเป็นอาชีพหลัก และกลุ่มคนที่ซื้อขายหุ้นเป็นงานอดิเรก ซื้อขายหุ้นเป็นบางเวลา ไม่ได้ศึกษาหุ้นอย่างจริงจังและไม่ใช้เงินลงทุนมาก

โครงการนี้จึงเกิดขึ้นเพื่อแก้ไขปัญหของการไม่มีเวลาที่จะศึกษาและลงทุนในหุ้น โดยโครงการนี้เป็นโครงการประเภทวิจัยและผลิตภัณฑ์ทางการค้า ที่ได้นำแนวคิดของ Machine Learning มาพัฒนาซอฟต์แวร์ที่สามารถใช้งานได้จริง ซึ่งจะให้โรบอทที่มีความชำนาญซื้อขายหุ้นแทนนักลงทุนที่ไม่ต้องการเสียเวลาเพื่อซื้อขายหุ้นมากนัก แต่ยังคงได้ผลตอบแทนที่ดีเหมือนการสละเวลาเพื่อศึกษาด้วยตนเอง

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อออกแบบและสร้างแอปพลิเคชันที่สามารถทำผลตอบแทนชนะตลาดได้ในการทดสอบย้อนหลัง 1 ปี ในปี 2559
2. เพื่อออกแบบและสร้างระบบแจ้งเตือนสัญญาณการซื้อขายหุ้น
3. เพื่อออกแบบและสร้างระบบ backtest ผลการเทรดหุ้นด้วยโมเดลต่างๆ
4. เพื่อเป็นแนวทางสำหรับการพัฒนาโรบอทเทรดที่สามารถใช้งานได้ในตลาดจริงต่อไปในอนาคต

1.3 ขอบเขตของโครงการ

1. สิ่งที่จะได้รับจากผลของโครงการ
 - โมเดลที่ใช้สำหรับทำนายผลว่า ผู้ใช้งานควรจะต้องซื้อขายหุ้นในวันพรุ่งนี้หรือไม่
 - ระบบแจ้งเตือนสัญญาณการซื้อขายหุ้นให้แก่ผู้ใช้งาน
 - ระบบ backtest ผลการเทรดหุ้นด้วยโมเดลต่างๆ
2. ขอบจำกัดของโครงการ
 - อัลกอริธึมทางด้าน Machine Learning 3 ประเภท ได้แก่ โมเดลตระกูล Neural Network โมเดลตระกูล Tree และ Reinforcement learning Model รวมทั้งหมด 10 โมเดล
 - ทดสอบการซื้อขายย้อนหลัง (Backtest) กับหุ้น 50 ตัวใน SET50 ในปี 2558-2560 ซึ่ง SET50 หมายถึง หุ้นสามัญ 50 ตัวที่มีมูลค่าตามราคาตลาด (Market Capitalization) สูง การซื้อขายมีสภาพคล่องสูงอย่างสม่ำเสมอ และมีสัดส่วนผู้ถือหุ้นรายย่อยผ่านเกณฑ์ที่กำหนด 50 ลำดับแรก
 - ข้อมูลที่นำไปให้ AI มาเรียนรู้ มาจากข้อมูลของหุ้นตั้งแต่ปี 2543 - 2558 โดยได้มาจาก เว็บไซต์ Siamchart.com และ Yahoo Finance API

1.4 ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน

1.4.1 ศึกษาข้อมูลเกี่ยวกับหุ้นและบอทเทรดหุ้น

หาข้อมูล ทดลองและทำความเข้าใจเกี่ยวกับตลาดหลักทรัพย์, การเทรดหุ้น, บอทเทรดหุ้น, Machine Learning และความรู้ที่เกี่ยวข้องอื่นๆ โดยศึกษาค้นคว้าข้อมูลจากหนังสือ, บทความ, วิดีโอ และแหล่งข้อมูลอื่นๆ ที่เกี่ยวข้อง

1.4.2 สร้างระบบ Backtest ที่นำโมเดลข้างต้นไปใช้เพื่อทดสอบผลย้อนหลัง

สร้างระบบที่เมื่อนำโมเดลมาใส่แล้ว สามารถบอกได้ว่า โมเดลนั้นสามารถเทรดจากข้อมูลเก่าได้ผลตอบแทนอย่างไรบ้าง

1.4.3 สร้าง ทดสอบ และปรับเปลี่ยนโมเดลการเทรดหุ้นด้วยข้อมูลตลาดหุ้นในอดีต โดยใช้วิธีการ Machine Learning

ขั้นตอนในการสร้างโมเดลเทรดหุ้นด้วย Machine Learning มีดังนี้

- 1) นำข้อมูลที่ได้จาก Siamchart และ Yahoo โดยข้อมูลที่ได้จะมีค่า Date, Open, High, Low, Close, Adjusted Close, Volume ตั้งแต่ปี 1970-2017 มาทำ Data Cleansing และ EDA
- 2) กำหนดข้อมูลที่จะมาเป็น feature ซึ่งจะเป็นข้อมูลที่เกี่ยวข้องกับหุ้น ไม่ว่าจะเป็น High, Low, Open, Close, Volume, Adjusted Close, Financial State, Volume ของหุ้นที่มีความสัมพันธ์, Indicator ต่างๆ, ประวัติการเคลื่อนไหวของตลาดและอื่นๆ ที่เรา ต้องการจะให้โมเดลพิจารณา
- 3) กำหนดข้อมูลที่จะนำมา label ว่าเป็นผลลัพธ์ในอดีต ซึ่งในที่นี้คือจุดซื้อขายในอดีต
- 4) รวบรวมข้อมูลที่ต้องการและจัดการข้อมูลให้อยู่ในรูปแบบที่นำไปใช้เทรดต่อได้
- 5) เลือกวิธีการที่ใช้ทำการเทรด โดยกลุ่มของเราเลือกวิธีการดังนี้
 - ตระกูล Neural Network
 - ตระกูล Tree
 - Reinforcement Learning
- 6) สร้างโมเดลโดยการเทรดข้อมูลกับ label
- 7) ทดสอบโมเดลด้วยการ backtest

ทดลองกับหุ้น 20 ตัว ที่มีความคล่องตัวสูง ซึ่งข้อมูลของหุ้นได้มาจากเว็บไซต์ Siamchart และ Yahoo โดยข้อมูลที่ได้จะมีค่า Date, Open, High, Low, Close, Adj Close และ Volume ตั้งแต่ปี 1970-2017
- 8) ปรับปรุงโมเดลโดยอาจย้อนไปทำข้อ 1 2 และ 4 เพื่อที่นำไป backtest แล้วได้

1.4.4 สรุปโครงงาน

จัดทำรูปเล่มรายงานและนำเสนอโครงงาน

แผนการปฏิบัติงาน	ส.ค.				ก.ย.				ต.ค.				พ.ย.				ธ.ค.				ผู้รับผิดชอบ
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1. คิดหัวข้อโครงการ วัตถุประสงค์และขอบเขตของโครงการ																					วิศรุต โชติธัช พิชญ์สินี
2. ศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องกับโครงการ																					
3. สร้างระบบ Backtest ที่สามารถนำไปทดสอบผลตอบแทนของโมเดลได้																					
4. กำหนด Features, Label และรวบรวมข้อมูล																					
5. สร้างและทดสอบ Neural network model																					วิศรุต
6. สร้างและทดสอบ Tree model																					พิชญ์สินี
7. สร้างและทดสอบ Reinforcement learning model																					โชติธัช

แผนการปฏิบัติงาน	ม.ค.				ก.พ.				มี.ค.				เม.ย.				พ.ค.				ผู้รับผิดชอบ
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
8. สร้างและทดสอบโมเดลตระกูล Neural network																					วิศรุต
9. สร้างและทดสอบโมเดลตระกูล Tree																					พิชญ์สินี
10. สร้างและทดสอบ Reinforcement Learning model																					โชติธัช
11. เปรียบเทียบ ทดสอบ และปรับปรุงให้ได้โมเดลที่สามารถทำผลตอบแทนได้ดีที่สุด																					วิศรุต โชติธัช พิชญ์สินี
12. สรุปผลโครงการ																					วิศรุต โชติธัช พิชญ์สินี

ตารางที่ 1.1 แผนการดำเนินงาน

■ แผนที่วางไว้ ■ ปฏิบัติจริง

1. ผลการดำเนินงานในภาคการศึกษาที่ 1

ในภาคการศึกษาที่ 1 มีการศึกษาและค้นคว้าข้อมูลที่เกี่ยวข้องกับโครงงาน กำหนดข้อมูลที่จะนำมาเป็น Feature ซึ่งจะเป็นข้อมูลที่เกี่ยวข้องกับหุ้น ไม่ว่าจะเป็น High, Low, Open, Close, Volume, Adjusted Close, Financial State, Volume ของหุ้นที่มีความสัมพันธ์, Indicator ต่างๆ, ประวัติการเคลื่อนไหวของตลาดและอื่นๆ ที่ต้องการจะให้โมเดลพิจารณา และกำหนดข้อมูลที่จะนำมา Label ซึ่งในที่นี้คือจุดซื้อขายในอดีต อีกทั้งยังทำการออกแบบและสร้างระบบ Backtest ที่สามารถนำไปทดสอบผลตอบแทนของโมเดลได้

2. ผลการดำเนินงานในภาคการศึกษาที่ 2

ในภาคการศึกษาที่ 2 จะเป็นการสร้างบอทเทรดหุ้นให้สำเร็จ โดยเน้นไปที่การสร้างและทดสอบโมเดลทั้งหมด 3 ประเภท ได้แก่ โมเดลตระกูล Neural network, โมเดลตระกูล Tree และ Reinforcement Learning model จากนั้นจึงทำการเปรียบเทียบเพื่อให้ได้โมเดลที่ให้ผลตอบแทนดีที่สุดตามวัตถุประสงค์ของโครงงาน คือ สามารถทำผลตอบแทนได้มากกว่าดัชนีตลาดหุ้นในการทดสอบย้อนหลัง 1 ปี ในปี 2559 พร้อมทั้งทำเอกสารสรุปผลโครงงาน

บทที่ 2

ที่มาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ผู้จัดทำโครงงานวิจัยได้มีการศึกษาทฤษฎีทางด้านวิศวกรรมคอมพิวเตอร์และงานด้านอื่นๆ ที่เกี่ยวข้องกับโครงงานนี้ อีกทั้งผู้จัดทำยังได้ทำการทบทวนวรรณกรรม เพื่อนำมาเป็นแนวทางที่สามารถนำมาประยุกต์ใช้ในการพัฒนาโครงงานให้สำเร็จลุล่วงไปได้ โดยได้ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องดังนี้

2.1 แนวคิดทางทฤษฎี

ในงานวิจัยได้มีการศึกษาและค้นคว้าแนวคิดทางทฤษฎีที่เกี่ยวข้องกับโครงงาน โดยได้นำแนวคิดทางทฤษฎีเกี่ยวกับเรื่องหุ้น การเรียนรู้ของเครื่อง (Machine learning) และระบบทดสอบย้อนหลัง (Backtest) มาใช้ในการพัฒนาโครงงานในส่วนของการออกแบบและสร้างระบบ Backtest ที่สามารถนำไปทดสอบผลตอบแทนของโมเดลได้ และการสร้างโมเดลเทรดหุ้นที่สามารถทำผลตอบแทนได้มากกว่าดัชนีตลาดหุ้นในการทดสอบย้อนหลัง 1 ปี ในปี 2559

2.1.1 หุ้น

หุ้นเป็นตราสารที่กิจการออกให้แก่ผู้ถือเพื่อระดมเงินทุนไปใช้ในกิจการ โดยผู้ถือตราสารทุกคนจะมีฐานะเป็นเจ้าของกิจการ ซึ่งจะมีส่วนได้ส่วนเสียหรือมีสิทธิในทรัพย์สินและรายได้ของกิจการ รวมทั้งมีโอกาสดำเนินการผลตอบแทนเป็นเงินปันผล ซึ่งขึ้นอยู่กับผลกำไรและข้อตกลงของกิจการนั้นๆ

- หลักการของการเทรดหุ้น

หุ้นแต่ละตัวจะมีราคาของหุ้น ซึ่งราคาหุ้นจะเปลี่ยนแปลงตามอุปสงค์กับอุปทาน หากว่ามีคนต้องการซื้อหุ้น A ที่ราคา 40.25 บาท 40,000 หุ้น มีคนต้องการขายหุ้น A ที่ราคา 40.50 บาท 30,000 หุ้น หากว่าเราต้องการซื้อหุ้นนี้ที่ 40.25 บาทด้วย เราก็ต้องไปต่อคิวเป็นหุ้นที่ 40,001 เป็นต้นไป แต่ถ้าหากเราต้องการจะซื้อเลย ก็จะได้ซื้อได้ไม่เกิน 30,000 หุ้นที่ราคา 40.50 บาท หากว่ามีคนที่ซื้อที่ราคา 40.50 บาทจนหมด 30,000 หุ้นแล้ว ราคาถัดไปก็จะขยับไปที่ 40.75 บาท (ราคาช่วงนี้จะหุบที่ละ 0.25 บาท) ซึ่งก็จะมีคนที่ต้องการขายที่ราคา 40.75 อยู่อีกเป็นจำนวนหนึ่ง ในทางกลับกัน หากมีความต้องการขายที่ราคา 40.25 บาทจนความต้องการซื้อได้รับหมดแล้ว ราคาถัดไปที่จะตั้งขายได้ก็คือที่ราคา 40.00 บาท จะเห็นได้ว่าในกรณีแรกมีความต้องการซื้อเยอะ ราคาที่ขึ้น กรณีที่สองมีความต้องการขายเยอะ ราคาที่ลง ทั้งนี้การขึ้นลงของราคาหุ้นเกิดจากหลายปัจจัยไม่ว่าจะเป็นผลตอบแทนที่คนที่ผู้ลงทุนอื่นคาดว่าจะได้รับทั้งในรูปของเงินปันผลและส่วนต่างราคา และความเสี่ยงที่ผู้ลงทุนอื่นต้องเผชิญจากการลงทุนในหุ้นนี้

การทำการกำไรจากการเทรดหุ้นนั้นคือหลักการซื้อขายทั่วไป ซื้อถูกขายแพง หากว่าผู้ลงทุนสามารถซื้อหุ้นตัวหนึ่งที่ราคาถูก แต่สามารถขายหุ้นนั้นตอนราคาแพงได้ ส่วนต่างนั้นก็จะเป็นการกำไรของผู้ลงทุน

- ค่า End of day ของหุ้น
 1. Date คือวันที่
 2. Open คือราคาเปิด
 3. High คือราคาสูงสุด
 4. Low คือราคาต่ำสุด
 5. Close คือราคาปิด
 6. Adjusted Close คือราคาปิดแบบปรับค่าแล้ว
 7. Volume คือปริมาณการซื้อขาย

- นักลงทุน 2 แบบ

หลักการวิเคราะห์ว่าราคาหุ้นจะไปทางใด รวมถึงเทคนิคการซื้อขายต่างๆ ก็จะแตกต่างกันไปตามแต่ละบุคคล ส่วนมากจะแบ่งได้เป็น 2 แบบ คือ แบบ technical ที่ใช้ค่าทางคณิตศาสตร์นำข้อมูลราคาในอดีตมาทำเป็น indicator ต่างๆ เพื่อหาจุดซื้อขาย แบบที่สองคือ value invest ที่วิเคราะห์งบการเงิน ประวัติ ราคาในอดีต ข่าวสารต่างๆ จนมั่นใจว่าหุ้นที่ตั้งใจจะซื้อนี้มีพื้นฐานดี ราคาจะขึ้นแน่นอน

- อินดิเคเตอร์ (Indicator)

อินดิเคเตอร์ คือ ค่าที่เกิดจากการคำนวณทางคณิตศาสตร์ ส่วนมากสามารถทำออกมาเป็นกราฟได้ จากกราฟนี้ ผู้ลงทุนจะสามารถนำอินดิเคเตอร์ไปวิเคราะห์ได้และหาจุดซื้อขาย หรือดูสภาวะของราคาต่างๆ ตัวอย่างอินดิเคเตอร์ที่เป็นที่นิยมได้แก่ EMA, MACD, OBV, Bollinger band และ RSI

- งบการเงิน

งบการเงินเปรียบเสมือนผลสรุปการดำเนินงานของกิจการหนึ่งๆ สามารถบอกค่าต่างๆ ได้ เช่น กำไรต่อหุ้นหรือมูลค่ากิจการ

2.1.2 การเรียนรู้ของเครื่อง (Machine learning)

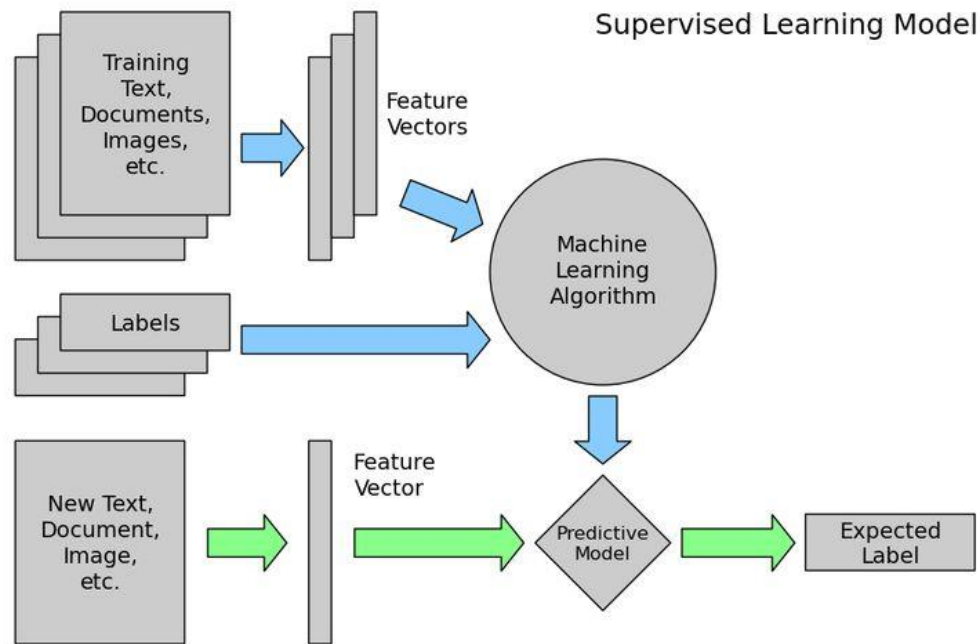
การเรียนรู้ของเครื่อง (Machine learning) เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่ถูกพัฒนามาจากการศึกษาการรู้จำแบบ โดยเป็นศาสตร์ที่ทำให้คอมพิวเตอร์มีความสามารถในการเรียนรู้ได้ด้วยตนเอง เมื่อมีข้อมูลเข้ามา สามารถทำนายหรือตัดสินใจได้โดยปราศจากการทำงานตามลำดับคำสั่งของโปรแกรม

การเรียนรู้ของเครื่องเป็นการรวมของศาสตร์หลายแขนง ไม่ว่าจะเป็นวิทยาการคอมพิวเตอร์, วิศวกรรมและโดยเฉพาะอย่างยิ่งสถิติศาสตร์ นอกจากนี้การเรียนรู้ของเครื่องยังสามารถนำไปประยุกต์ใช้งานได้หลากหลาย เช่น การกรองอีเมลขยะ การรู้จำตัวอักษร เครื่องมือค้นหาและคอมพิวเตอร์วิทัศน์

การเรียนรู้ของเครื่องแบ่งออกได้เป็น 2 ประเภท ได้แก่ Supervised learning และ Unsupervised learning

1) การเรียนรู้แบบมีผู้สอน (Supervised learning)

เป็นเทคนิคการเรียนรู้ของเครื่องที่เรียนรู้จากข้อมูลสอน (Training data) โดยมีการแบ่งกลุ่มข้อมูลจากลักษณะหรือคุณสมบัติที่มีคล้ายกัน ซึ่งส่วนของข้อมูลที่ใช้เรียกแทนกลุ่มข้อมูลแต่ละกลุ่มเรียกว่า Label และคุณสมบัติของข้อมูลเรียกว่า Feature



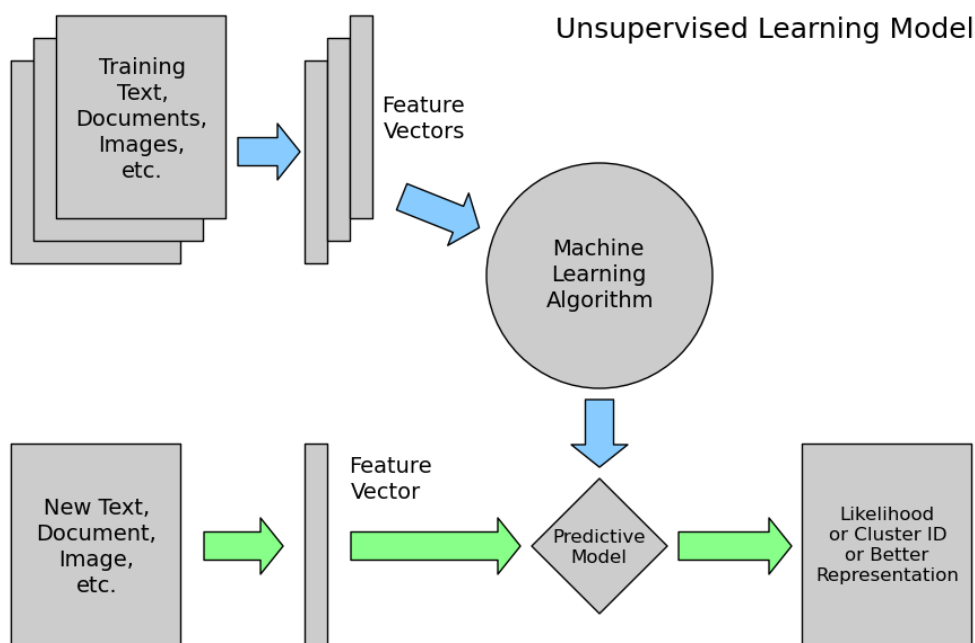
รูปที่ 2.1 หลักการทำงานของการเรียนรู้แบบมีผู้สอน

[ที่มา: <http://www.allprogrammingtutorials.com>]

จากรูปที่ 2.1 ลูกศรสีฟ้าแสดงถึงกระบวนการเรียนรู้ของเครื่อง โดยเริ่มจากการนำข้อมูลสอน (Training data) มาทำการคัดแยกคุณสมบัติ (Features) ว่าข้อมูลแต่ละข้อมูลมีคุณสมบัติอย่างไรบ้าง หลังจากนั้นจึงทำการวิเคราะห์และจัดกลุ่มข้อมูลจากคุณสมบัติที่มีเหมือนกันให้อยู่กลุ่มเดียวกัน (Label) เมื่อออกแบบอัลกอริธึมเสร็จแล้วจะได้โมเดลสำหรับทำนายผล ลูกศรสีเขียวแสดงถึงกระบวนการทดสอบโมเดล โดยจะนำข้อมูลกลุ่มใหม่ที่ไม่ใช่ข้อมูลสอน (Test data) มาทำการคัดแยกคุณสมบัติแล้วป้อนข้อมูลเหล่านี้เข้าสู่โมเดล โมเดลจะให้ผลการทำนายจากสิ่งที่ได้เรียนรู้มาว่า ข้อมูลที่มีคุณสมบัติลักษณะนี้เป็นข้อมูลกลุ่มใด

2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning)

การเรียนรู้ไม่ได้มีการ Label ที่บอกว่าข้อมูลนั้นคืออะไร แต่การเรียนรู้จะจัดข้อมูลนำเข้า (Input) เป็นกลุ่ม (Cluster) บนพื้นฐานของความเหมือน (Similarities) และความแตกต่าง (Differences) ระหว่างรูปแบบของข้อมูลนำเข้า (Input patterns)



รูปที่ 2.2 หลักการทำงานของการเรียนรู้แบบไม่มีผู้สอน

[ที่มา: <http://www.allprogrammingtutorials.com>]

จากรูปที่ 2.2 ลูกศรสีฟ้าแสดงถึงกระบวนการเรียนรู้ของเครื่องแบบไม่มีผู้สอน โดยเริ่มจากการนำข้อมูลสอน (Training data) มาทำการคัดแยกคุณสมบัติ (Features) เช่นเดียวกันกับการเรียนรู้แบบมีผู้สอน หลังจากนั้นโมเดลจะทำการเรียนรู้ข้อมูลจากคุณสมบัติของข้อมูลที่มีความเหมือนกันให้อยู่กลุ่มเดียวกัน และกำหนดสิ่งที่ใช้เรียกแทนแต่ละกลุ่มไว้ เช่น แทนแต่ละกลุ่มด้วยตัวเลข ส่วนลูกศรสีเขียวแสดงถึงกระบวนการทดสอบโมเดล โดยจะนำข้อมูลกลุ่มใหม่ที่ไม่ใช่ข้อมูลสอน (Test data) มาทำการคัดแยกคุณสมบัติแล้วป้อนข้อมูลเหล่านี้เข้าสู่โมเดล โมเดลจะให้ผลการทำนายจากสิ่งที่ได้เรียนรู้มาว่า ข้อมูลที่มีคุณสมบัตินี้เป็นข้อมูลกลุ่มใดตามที่โมเดลกำหนดไว้

โดยการเรียนรู้ที่ใช้ในโครงงานนี้มีอยู่ 3 ประเภท ได้แก่

1. โครงข่ายประสาท (Neuron Network)

2. การเรียนรู้ต้นไม้ตัดสินใจ (Decision Tree)
3. การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

2.1.3 ระบบทดสอบย้อนหลัง (Backtest)

ในการเทรดหุ้น การทดสอบย้อนหลังคือการทดสอบแผนการเทรดหุ้นหนึ่งๆ กับข้อมูลในอดีตเพื่อชี้วัดประสิทธิภาพของแผนการเทรดนั้น หลักการของการทดสอบย้อนหลังคือการนำแผนการเทรดหุ้นไปลองเทรดจริงในช่วงเวลาหนึ่งในอดีต โดยสามารถทำเช่นนี้ได้จากข้อมูลหุ้นในอดีตที่เราได้เก็บมา เรียกได้ว่าเป็นการจำลองการทำงานของแผนการ ดูผลลัพธ์หากว่า ณ เวลานั้นได้ใช้แผนการนี้จะเป็นเช่นไร สำหรับ Machine Learning ช่วงข้อมูลที่นำมาทำการทดสอบนั้น จะเป็นช่วงที่ไม่ได้นำไปเทรดกับตัวโมเดลมาก่อน มิฉะนั้นจะเกิดการทดสอบที่ไม่ยุติธรรมได้ เนื่องจากข้อมูลเป็นสิ่งที่โมเดลเคยเห็น ดังนั้น ก่อนจะเทรดโมเดลและทดสอบย้อนหลัง ข้อมูลจะถูกแบ่งเป็น 2 ส่วนสำหรับเทรดส่วนหนึ่งและทดสอบส่วนหนึ่ง การทดสอบย้อนหลังของ Machine Learning สำหรับเทรดหุ้นสามารถบอกสิ่งต่างๆ ได้ดังต่อไปนี้

- ค่าทาง Machine Learning

1. Confusion Matrix

- 1.1 True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่าจริงและมนุษย์บอกว่าจริง
- 1.2 True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริงและมนุษย์บอกว่าไม่จริง
- 1.3 False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่าจริง แต่มนุษย์บอกว่าไม่จริง
- 1.4 False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง แต่มนุษย์บอกว่าจริง

2. Accuracy คือ ค่าที่บอกว่าโปรแกรมสามารถทำนายถูกจากข้อมูลที่ทำนายว่าเป็นสิ่งที่พิจารณา

3. Precision คือ ค่าที่บอกว่าโปรแกรมทำนายว่าจริง ถูกต้องเท่าไร

4. Recall คือ ค่าที่บอกว่าโปรแกรมสามารถทำนายถูกต้องจากที่ถูกต้องทั้งหมดเท่าไร

5. F-Measure คือค่าเฉลี่ยระหว่าง Precision กับ Recall

- ค่าทางพอร์ตหุ้น

1. Total return คือ ค่าที่บอกว่าผลตอบแทนโดยรวมคิดเป็นกี่ปอร์เซ็นต์
2. Benchmark return คือ ค่าที่บอกว่า set index มีผลตอบแทนเท่าใด
3. Alpha คือ ค่าที่บอกว่าผลตอบแทนของแผนการเทรดแตกต่างกับ set index เท่าใด
4. Beta คือ ค่าที่บอกว่าผลตอบแทนของแผนการเทรดเป็นกี่เท่าของ set index
5. Sharpe ratio คือ ค่าที่บอกถึงความเสี่ยงและผลตอบแทนของพอร์ต
6. Sortino ratio เหมือน Sharpe ratio แต่จะไม่คิดส่วนการเบี่ยงเบนขาขึ้น

7. Volatility คือ ค่าที่บอกความเสี่ยงด้วยค่าเบี่ยงเบน
8. Max drawdown คือ ค่าที่บอกจุดขาดทุนสูงสุดเป็นเปอร์เซ็นต์
9. Transaction detail เป็นลิสต์ของข้อมูลการเทรดแต่ละครั้ง
10. Daily return เป็นกราฟที่บอกผลตอบแทนต่อวันของแผนการเทรดที่ถูกทดสอบ

2.2 โครงสร้างข้อมูลและอัลกอริธึม

2.2.1 โครงข่ายประสาท (Neuron network)

โครงข่ายประสาทเทียมเป็นวิธีการที่เลียนแบบลักษณะการเรียนรู้ของมนุษย์ โดยปฏิกิริยาเรียนรู้จากเซลล์ประสาทในสมอง เมื่อเซลล์ได้รับกระแสไฟฟ้าจากสิ่งเร้าหรือจากเซลล์อื่นๆ เซลล์ประสาทเหล่านี้จะมีกระบวนการในการตัดสินใจว่าจะส่งกระแสไฟฟ้าต่อหรือไม่ หากกระแสที่เข้ามามีความแรงพอ เซลล์ประสาทก็จะส่งกระแสไฟฟ้าต่อไปยังเซลล์ประสาทข้างเคียงต่อไป

โครงข่ายประสาทเทียมได้หยิบเอาแนวคิดนี้มาใช้ในการสร้างระบบเรียนรู้ โดยการสร้างกลุ่มประมวลผลย่อยเพื่อใช้ในการตัดสินใจเช่นเดียวกับที่สมองมีเซลล์ประสาทหลายเซลล์ แต่ละหน่วยประมวลผลหรือเรียกว่า นิวรอน จะมี Weight ที่เอาไว้กำหนดความสำคัญของ Input แต่ละ Input หลังจากคำนวณ Input ทั้งหมดแล้ว หากค่าการตัดสินใจมีมากเกินระดับหนึ่ง นิวรอนก็จะส่ง Output ออกไปยังนิวรอนตัวอื่นๆ ตามลักษณะเครือข่ายที่กำหนดไว้ ซึ่ง Output ของนิวรอนตัวนี้จะเป็น Input ของนิวรอนอีกตัว ในการเทรน 1 ครั้งจะเกิดการส่งข้อมูลหลายครั้ง ซึ่งในแต่ละครั้งนิวรอนจะค่อยๆ ปรับ Weight จนเมื่อสิ้นสุดการเทรนข้อมูล นิวรอนทุกตัวก็จะสามารถตัดสินใจได้ดีมากขึ้น

การทำนิวรอนโดยมากจะใช้เทคนิคที่เรียกว่า Back-propagation มาคำนวณในส่วนของการปรับ Weight เทคนิคนี้คือการคิด gradient ของแต่ละนิวรอนทั้งขาไปและขากลับ ซึ่งการคิดนี้คือการปรับ Weight เพื่อหาสัดส่วนที่เหมาะสมในการตัดสินใจจาก Input เทคนิคนี้ทำให้โครงข่ายประสาทเทียมมีประสิทธิภาพสูงขึ้นมาก

โครงข่ายประสาทรูปแบบหนึ่งที่ควรกล่าวถึงคือ RNN (Recurrent Neuron network) เป็นโครงข่ายประสาทเทียมที่ใช้สำหรับการวิเคราะห์ข้อมูลแบบเป็นลำดับขั้นตอนโดยเฉพาะ หลักการของระบบนี้จะเป็นการนำข้อมูลเข้าลำดับก่อนหน้าจำนวนหนึ่ง เข้ามาเป็นปัจจัยในการพิจารณาข้อมูลเข้า ณ ปัจจุบันด้วย ดังนั้น ระบบนี้จึงสามารถทำนายผลลัพธ์จากข้อมูลที่ลำดับมีผลต่อผลลัพธ์

2.2.2 การเรียนรู้ต้นไม้ตัดสินใจ (Decision tree learning)

การเรียนรู้แบบต้นไม้ตัดสินใจ เป็นวิธีการที่ทำออกมาเป็น Tree ได้ โดยแต่ละกิ่งก้านจะบอก Probability ว่าข้อมูลนี้สามารถถูกตัดสินใจเป็นคำตอบต่างๆ ได้ด้วยความน่าจะเป็นเท่าใด

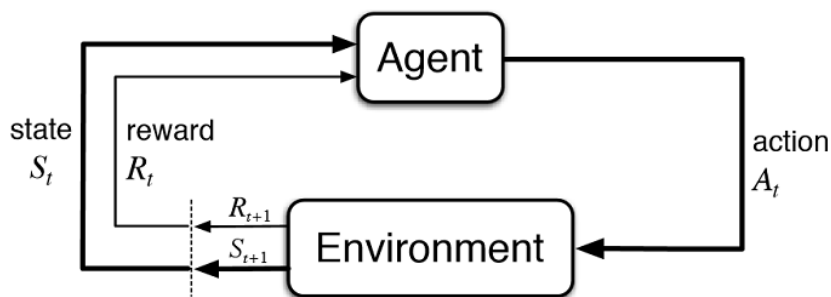
การเรียนรู้แบบนี้จะอาศัย Attribute แต่ละประเภทในการจำแนก เมื่อเรียนรู้แล้วต้นไม้จะสามารถบอกได้ว่าควรดูจาก Attribute ไດก่อนเป็นหลัก และ Attribute แต่ละตัวนั้นมีความสำคัญต่อการตัดสินใจแค่ไหน

2.2.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement learning)

การเรียนรู้แบบเสริมกำลังหรือ Reinforcement learning เป็นการเรียนรู้ที่คอมพิวเตอร์จะสนใจต่อสิ่งแวดล้อมเป็นพิเศษ เช่น AlphaGo ของ Google ทุกครั้งที่ผู้เล่นเดินหมาก AlphaGo จะคำนวณและหาวิธีเดินใหม่ตามสิ่งแวดล้อม ซึ่งก็คือการเล่นของผู้เล่นที่เปลี่ยนไป ในอีกแง่มุมหนึ่ง Reinforcement learning ก็เหมือนการ Trial-and-error คือ ลองผิดลองถูก เพราะการเรียนรู้แบบนี้คือการให้รางวัลและการลงโทษ

การนิยามการแก้ปัญหาด้วยวิธี Reinforcement learning มีสิ่งที่ต้องกำหนดดังต่อไปนี้

1. สถานะ (S) เช่น บอทสำหรับเล่นเกมเอ็กซ์โอ สถานะคือภาพกระดานที่กำลังเล่นอยู่ในปัจจุบัน
2. การกระทำ (A) ซึ่งอาจจะไม่เหมือนกันสำหรับแต่ละสถานะ เช่น ตอนต้นเกมของเอ็กซ์โอ ผู้เล่นย่อมมีทางเลือกในการเล่นมากกว่าตอนท้ายเกม
3. รางวัล (R) ขึ้นอยู่กับแต่ละสถานะหรือขึ้นอยู่กับการกระทำ หรือทั้งคู่ก็ได้ โดยทั่วไปสำหรับการเล่นเกมแล้วก็คือผลลัพธ์ของการเล่น เช่น ชนะอาจจะเป็นเลขบวก (+) แพ้อาจจะเป็นเลขลบ (-) หากเสมอก็อาจจะเป็นศูนย์ (0) รางวัลประเภทนี้เรียกว่า รางวัลที่ไม่ได้รับทันที เนื่องจากต้องเล่นให้จบเกมเสียก่อน รางวัลยังมีอีกประเภท คือ รางวัลที่ได้รับทันที เช่น โปรแกรมควบคุมหุ่นยนต์ให้เดินได้เร็วๆ รางวัลอาจจะเป็นความเร็วของหุ่นยนต์ ถ้าเดินได้เร็วก็จะได้รับรางวัลทันที



รูปที่ 2.3 หลักการทำงานของ Reinforcement learning

[ที่มา: <https://www.tractica.com>]

หลักการทำงานของการเรียนรู้แบบเสริมกำลัง คือ ในแต่ละสถานะ (State) ตัวแทนของระบบ (Agent) จะสังเกตสิ่งแวดล้อม จากนั้นจึงเลือกการกระทำ (Action) จากการกระทำทั้งหมดที่สามารถทำได้ เมื่อเลือกการ

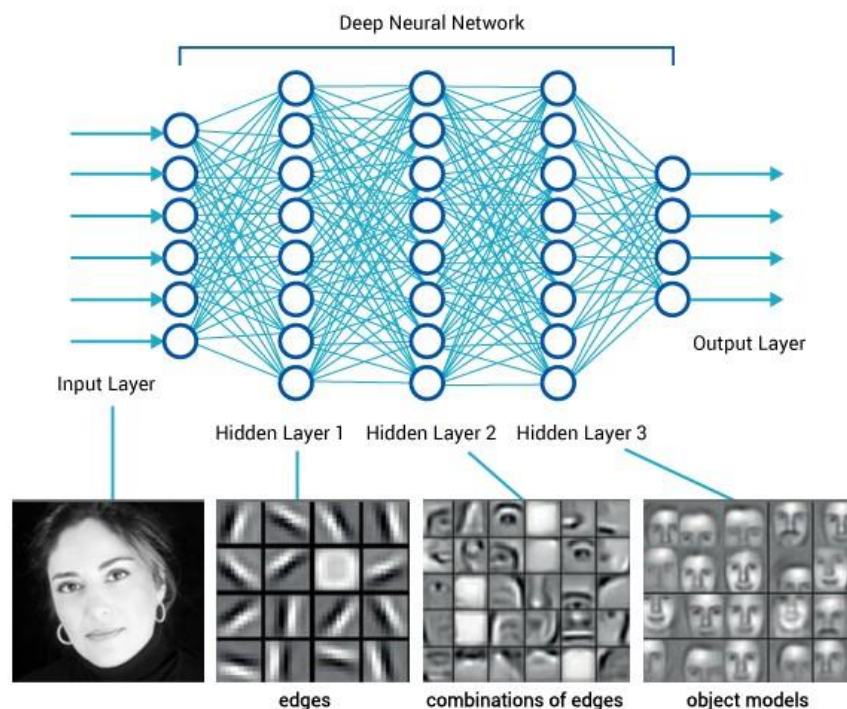
กระทำแล้ว การกระทำนั้นจะส่งผลต่อสิ่งแวดล้อมทำให้สถานะเปลี่ยนไป และตัวแทนของระบบจะได้รับรางวัลจากการกระทำที่เลือก โดยใน Reinforcement learning นั้นต้องการสร้างอัลกอริธึมที่มีนโยบาย (Policy) ที่ดีที่สุด นั่นคือได้รางวัลรวมที่คาดหวัง (Maximum Sum of Expected Rewards) มากที่สุด

2.2.4 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึกมีพื้นฐานมาจากโครงข่ายประสาทเทียม แต่การเรียนรู้เชิงลึกนี้จะมีโครงสร้างที่ทำให้สามารถเรียนรู้ระดับที่สูงยิ่งกว่าได้ โดยการเพิ่มชั้น (Layer) เข้าไป ซึ่งชั้นแต่ละชั้นจะมีหน้าที่และวิธีการของตนเอง ชั้นถัดไปจะนำผลลัพธ์จากชั้นก่อนไปทำการเรียนรู้ต่อ

การเรียนรู้โดยผ่านนิเวศหลายชั้น แต่ละชั้นจะส่งผลกระทบและเรียนรู้กับชั้นอื่นด้วยเช่นกัน ยิ่งผ่านนิเวศมากๆ ข้อมูลในชั้นสูงๆ จะมีความซับซ้อนมากยิ่งขึ้น ในปัจจุบันการเรียนรู้เชิงลึกถือเป็นการเรียนรู้ที่มีประสิทธิภาพสูงสุดกับข้อมูลขนาดใหญ่ สามารถเรียนรู้สิ่งที่ซับซ้อนได้หากมีข้อมูลปริมาณมากพอ

หลักการของการเรียนรู้เชิงลึก ก็จะเป็นโครงข่ายประสาทเทียมที่มีโหนดหลายๆ ชั้นและใช้การประมวลผลแบบขนาน (Parallel Processing) ทำให้สามารถประมวลผลได้ครั้งละจำนวนมาก ช่วยให้การเรียนรู้ของเครื่องสามารถให้ผลลัพธ์ในการตัดสินใจและคาดการณ์ได้ดีมากยิ่งขึ้น



รูปที่ 2.4 Deep Learning สำหรับการเรียนรู้จดจำใบหน้า

[ที่มา: <https://medium.com/@athivvat/>]

Deep learning ถูกนำมาประยุกต์ใช้ในงานต่างๆ มากมาย เช่น การแยกแยะใบหน้าแต่ละคน ตัวอย่างเช่นในการติดแท็กรูปภาพเพื่อนใน Facebook หรือการแยกวัตถุที่ไม่ใช่คน หรือใช้เป็นส่วนหนึ่งในระบบรถยนต์ไร้คนขับ เป็นต้น

2.3 ภาษาโปรแกรมและเครื่องมือซอฟต์แวร์ที่ใช้

ในหัวข้อนี้เป็นการศึกษาภาษาโปรแกรมและเครื่องมือซอฟต์แวร์ที่ใช้ในการพัฒนาโครงการ ผู้จัดทำเลือกใช้ภาษา Python บนซอฟต์แวร์ Jupyter-notebook ซึ่งเป็นชุดเครื่องมือหนึ่งในซอฟต์แวร์ Anaconda โดยใช้ภาษา Python ทั้งในส่วนของการทดสอบย้อนหลังผลการเทรดหุ้นและการสร้างโมเดลทั้งหมด 3 ตระกูล ได้แก่ โมเดลตระกูล Neural network, โมเดลตระกูล Tree และ Reinforcement Learning model

2.3.1 Python



รูปที่ 2.5 สัญลักษณ์ภาษา Python

[ที่มา: <https://www.python.org/community/logos/>]

ภาษา Python เป็นภาษาที่ทำงานโดยการแปลคำสั่งทีละคำสั่งด้วยตัวแปลภาษาไพทอน (Python Interpreter) เป็นภาษาที่เข้าใจง่ายและเหมาะสำหรับการทำ Machine Learning Model เนื่องจากมี Library ให้เลือกใช้งานจำนวนมาก ซึ่งในการพัฒนาโครงการนี้จะใช้ Python เวอร์ชัน 3.6 และใช้ Library ดังนี้

- Talib Library

เป็น Library ในภาษา Python ที่ช่วยในการคำนวณเพื่อหาค่าทางคณิตศาสตร์เกี่ยวกับหุ้น (Indicator) ต่างๆ เช่น Moving Average Convergence Divergence (MACD) ค่าที่สามารถบอกแนวโน้มราคาที่เกิดขึ้น, บอก Momentum ของราคาหุ้น, Relative Strength Index (RSI) ค่าที่สามารถบอกการแกว่งตัวของราคาหุ้น เพื่อดูภาวะการซื้อมากเกินไป (OVERBOUGHT) หรือขายมากเกินไป (OVERSOLD) และ indicator อื่นๆ อีกมากมาย

- Scikit-learn library

เป็น Library สำหรับช่วยในการทำ Machine Learning และ Data Mining บนภาษา python

- Keras Library

เป็น Library สำหรับช่วยในการทำ Deep Learning บนภาษา python

- Pandas Library

เป็น Library สำหรับช่วยในการสร้าง แก้ไข อ่านและบันทึก Datasets รวมไปถึงการดาวน์โหลด Datasets จากเว็บไซต์หลายเว็บไซต์ได้โดยตรงบนภาษา Python

- Matplot Library

เป็น Library ที่ใช้ในการสร้างและแสดงผลของข้อมูลในรูปแบบกราฟเพื่อวิเคราะห์ข้อมูลบน ภาษา Python

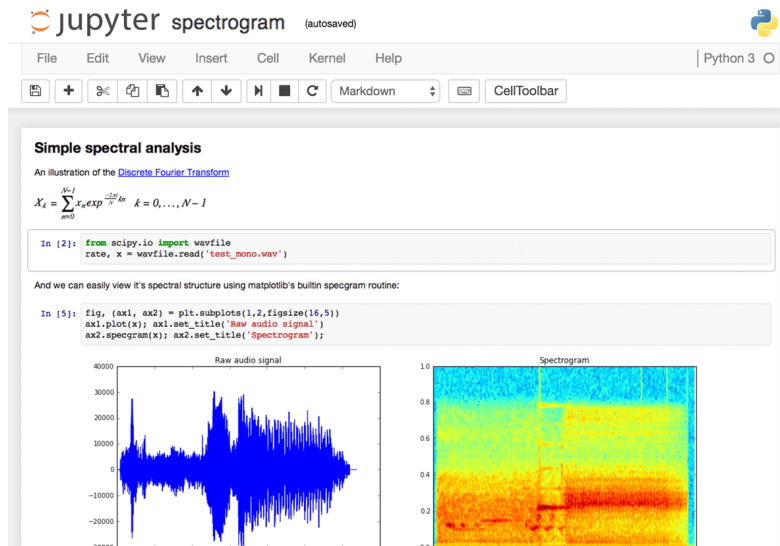
2.3.2 Jupyter notebook IDE



รูปที่ 2.6 สัญลักษณ์ Jupyter-notebook IDE

[ที่มา: <https://avatars3.githubusercontent.com/u/7388996?s=400&v=4>]

Jupyter notebook เป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรมภาษา Python ซึ่งทำงานบนเว็บเบราว์เซอร์ โดยสามารถเขียนคำสั่งร่วมกับการเขียนอธิบายและแทรกรูปภาพประกอบได้ นอกจากนี้ยังสามารถดูผลลัพธ์ของคำสั่งได้เลยเมื่อพิมพ์คำสั่งเสร็จ และยังสามารถเก็บข้อมูลงาน ไม่ว่าจะเป็นรูปภาพหรือ Dataset ต่างๆ ได้อีกด้วย



รูปที่ 2.7 ตัวอย่างการใช้งาน Jupyter-notebook IDE
[ที่มา: <https://www.dataquest.io>]

2.3.3 Anaconda



รูปที่ 2.8 โปรแกรม Anaconda
[ที่มา: <http://seanlaw.github.io>]

โปรแกรม Anaconda เป็นชุดแจกจ่ายอย่างหนึ่งของภาษา Python โดยได้รวบรวมเครื่องมือทางวิทยาศาสตร์เข้ามาด้วย ไม่ว่าจะเป็นงาน Data Analytics งานด้านวิทยาศาสตร์ เช่น โมดูล IPython, Pandas, NumPy, Qt/PySide, NLTK และอื่นๆ ซึ่งในปัจจุบัน IPython ก็คือ Jupyter notebook นั่นเอง ในการทำโครงการนี้จะใช้โปรแกรม Anaconda เวอร์ชัน 3.6

2.4 ประเภทของงานวิจัยที่เกี่ยวข้อง

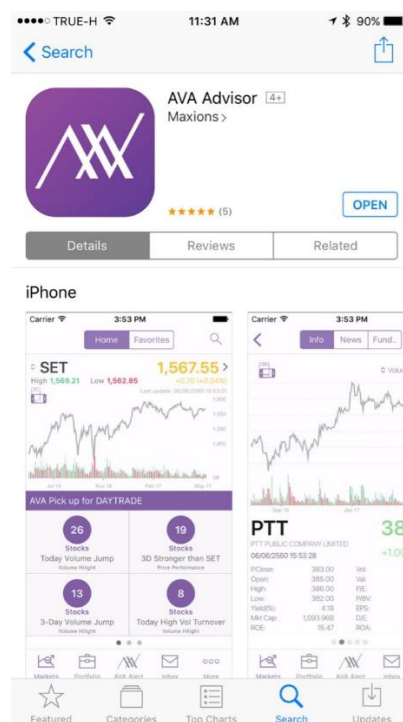
2.4.1 AVA



รูปที่ 2.9 แอปพลิเคชัน AVA Advisor

[ที่มา: <https://lh3.googleusercontent.com>]

เป็นแอปพลิเคชันที่นำ AI เข้ามาใช้ในการลงทุน เป็น Robot Advisor ในชื่อ AVA (เอวา) ที่ช่วยให้ผู้ใช้งานสามารถลงทุนได้ง่ายขึ้น โดยสิ่งที่ AVA ให้แก่ผู้ใช้งานคือ คำแนะนำจาก AI การคัดเลือหุ้นที่น่าสนใจและการตรวจจับการผันผวนของตลาดหุ้น ซึ่งสามารถปรับแต่งได้ตามความต้องการของผู้ใช้งาน



รูปที่ 2.10 ตัวอย่างของแอปพลิเคชัน AVA Advisor

[ที่มา: <https://f.ptcdn.info>]

เมื่อเปรียบเทียบกับโครงการนี้แล้ว AVA เป็นการให้คำแนะนำการเทรด การคัดเลือกหุ้นที่น่าสนใจ และการตรวจจับการผันผวนของตลาดหุ้น แต่โครงการของเราเป็นการแจ้งเตือนสัญญาณการเทรดให้ผู้ใช้งานได้ทราบเลย ซึ่งช่วยลดขั้นตอนการซื้อขายหุ้นให้ผู้ใช้งานมากกว่า เนื่องจากการได้รับสัญญาณแจ้งเตือนจาก AVA ผู้ใช้งานจะต้องแปลสัญญาณว่าควรซื้อขายหรือไม่

2.4.2 Bualuang iProgram Trade



รูปที่ 2.11 โปรแกรม Bualuang iProgram Trade

[ที่มา: <https://www.bualuang.co.th>]

บริการส่งคำสั่งซื้อขายอัตโนมัติแบบครบวงจร (Program Trading) โดยผู้ลงทุนเป็นผู้กำหนดกลยุทธ์การลงทุนร่วมกับทีมงานมืออาชีพจากหลักทรัพย์บัวหลวง ที่จัดเตรียมชุดคำสั่งคอมพิวเตอร์ เพื่อให้ระบบทำการคำนวณและจับสัญญาณการลงทุน ตลอดจนส่งคำสั่งซื้อขายแบบอัตโนมัติ ช่วยให้ผู้ลงทุนประหยัดเวลา โดยไม่จำเป็นต้องเทรดเองช่วยให้ช่วยเพิ่มประสิทธิภาพในการบริหารเงินลงทุนได้ แต่ว่าหลักการของ Bualuang iProgram Trade นี้ ไม่ได้ใช้หลักการของ Machine Learning ใช้เพียงแค่เงื่อนไขทั่วไปเท่านั้น

เมื่อเปรียบเทียบกับโครงการนี้แล้ว Bualuang iProgram Trade ไม่ได้ใช้หลักการเรียนรู้ด้วยตัวเองของ Machine Learning โดยผู้ใช้งานจะต้องกำหนดกลยุทธ์เป็นเงื่อนไขการซื้อขาย (Rule-based) เอง ซึ่งอาจจะได้ผลตอบแทนที่ไม่มากพอ และใช้กับหุ้นทุกตัวไม่ได้

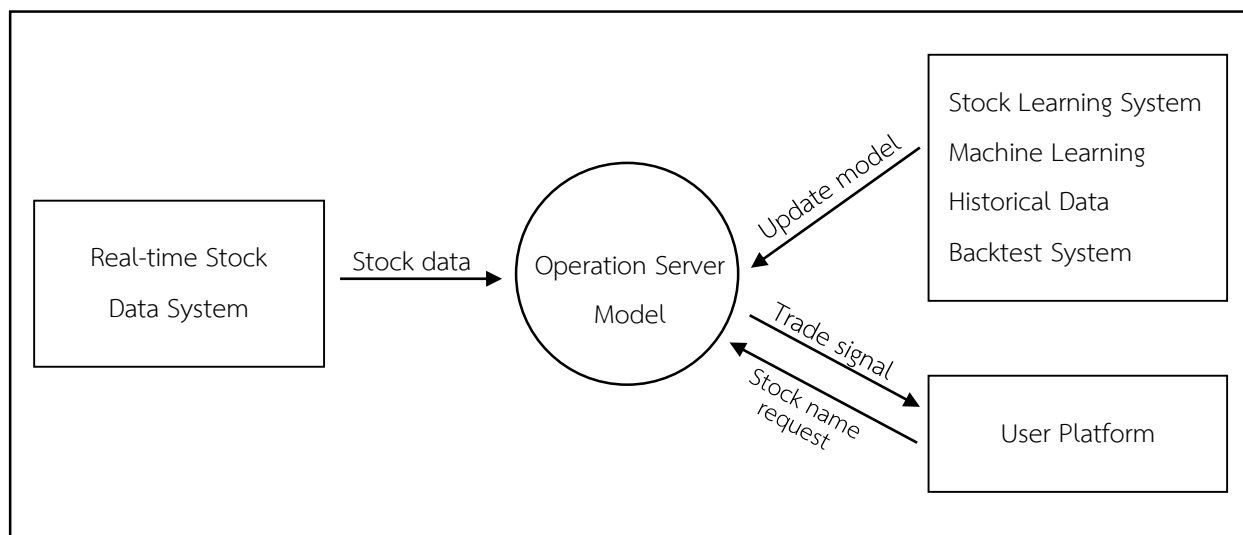
บทที่ 3

การออกแบบและระเบียบวิธีวิจัย

ในบทที่ 3 นี้จะเป็นบทที่กล่าวถึงการออกแบบและการพัฒนา Deep Learning Stock Trading Advisor ซึ่งสามารถแบ่งการดำเนินงานได้เป็น 2 ส่วนหลักคือ 3.1 หลักการทำงานของ Deep Learning Stock Trading Advisor เป็นการออกแบบการทำงานของระบบทั้งหมด โดยอธิบายให้เห็นถึงภาพรวมของโครงงานที่ทำให้ง่ายต่อการเข้าใจ และ 3.2 การออกแบบและสร้างโมเดล ในส่วนนี้จะเป็นการออกแบบและสร้างโมเดลที่ใช้สำหรับเทรดหุ้น ซึ่งจะรวมถึงการออกแบบและสร้างระบบ Backtest, การทำ Cleansing Data, การทำ Label และการกำหนด Feature ด้วย โดยจะกล่าวถึงรายละเอียดในแต่ละส่วนดังต่อไปนี้

3.1 หลักการทำงานของ Deep Learning Stock Trading Advisor

ในส่วนแรกนี้จะกล่าวถึงวิธีการทำงานของ Deep Learning Stock Trading Advisor ทั้งหมด โดยจะอธิบายให้เห็นถึงภาพรวมของโครงงาน ซึ่งทำให้เข้าใจขั้นตอนการทำงานและความสัมพันธ์ของส่วนประกอบต่างๆ ของระบบ Deep Learning Stock Trading Advisor



รูปที่ 3.1 ระบบการทำงานภาพรวมของ Deep Learning Stock Trading Advisor

ระบบการทำงานโดยรวมของ Deep Learning Stock Trading Advisor สามารถแบ่งส่วนของการทำงานย่อยได้เป็น 4 ส่วน ได้แก่

1. Operation Server Model

2. Stock Learning System
3. Real-time Stock Data System
4. User Platform

การทำงานของระบบมีศูนย์กลางอยู่ที่โมเดล Operation Server ซึ่งทำหน้าที่เก็บและประมวลผลโมเดลที่ได้ทำการเทรนแล้วหลายรูปแบบ โมเดลเหล่านี้จะต้องได้รับการพัฒนาและอัปเดตโดย Stock Learning System ประกอบกับได้รับข้อมูลหุ้นแบบ Real-time จาก Real-time Stock Data System ที่คอยให้ข้อมูลหุ้นแก่โมเดล Operation Server เพื่อให้สามารถทำการทำนายและส่งสัญญาณการซื้อขายหุ้น (Trade signal) กลับไปหาผู้ใช้งานได้ผ่าน User platform โดยในแต่ละส่วนการทำงานมีรายละเอียดดังต่อไปนี้

3.1.1 Model operation server

Model operation server คือ server ที่มีหน้าที่ 2 อย่างคือ เป็นแหล่งเก็บโมเดลหลากหลายรูปแบบที่ผ่านการเทรนจากข้อมูลหุ้นมาแล้ว และทำหน้าที่ประมวลผลโมเดลเหล่านี้ ได้ผลลัพธ์เป็นสัญญาณการเทรดหุ้นที่ผู้ใช้งานต้องการทราบบนการเทรดหุ้นของตน

3.1.2 Stock learning system

Stock learning system คือ ระบบที่รับผิดชอบในส่วนของโมเดลที่นำมาใช้ใน Server ไม่ว่าจะเป็นการสร้าง การอัปเดตหรือแก้ไขต่างๆ เมื่อโมเดลใดเสร็จสมบูรณ์แล้ว จะต้องผ่านขั้นตอนการทดสอบด้วย Backtest ก่อน หากผลลัพธ์เป็นที่น่าพอใจจึงจะนำโมเดลเข้าสู่ Server เพื่อนำไปใช้จริง

3.1.3 Real-time stock data system

Real-time stock data system คือ ระบบที่ทำหน้าที่คอยให้ข้อมูลที่ Server ต้องการแบบ Real-time ระบบจะทำการ clean data และจัดรูปแบบข้อมูลให้เป็นไปตามรูปแบบที่ Model สามารถนำไปใช้งานได้

3.1.4 User platform

User platform คือตัว Application ที่เชื่อมกับ Server โดยผู้ใช้งานสามารถตั้งค่าหุ้นที่ต้องการสัญญาณการเทรดได้ ตัว platform จะทำหน้าที่ติดต่อกับ Server เพื่อนำสัญญาณการเทรดของหุ้นที่ผ่านการประมวลผลจากฝั่ง Server มาแจ้งเตือนผู้ใช้งาน

3.2 การออกแบบและสร้างโมเดล

ขั้นตอนในการสร้างโมเดลเทรดหุ้นด้วย Machine Learning มีดังนี้

- 1) นำข้อมูลที่ได้จาก Siamchart และ Yahoo โดยข้อมูลที่ได้จะมีค่า Date, Open, High, Low, Close, Adjusted Close, Volume ตั้งแต่ปี 1970-2017 มาทำ Data Cleansing และ EDA

2) กำหนดข้อมูลที่จะมาเป็น feature ซึ่งจะเป็นข้อมูลที่เกี่ยวข้องกับหุ้น ไม่ว่าจะเป็น High, Low, Open, Close, Volume, Adjusted Close, Financial State, Volume ของหุ้นที่มีความสัมพันธ์, Indicator ต่างๆ, ประวัติการเคลื่อนไหวของตลาดและอื่นๆ ที่เรา ต้องการจะให้โมเดลพิจารณา

3) กำหนดข้อมูลที่จะนำมา label ว่าเป็นผลลัพธ์ในอดีต ซึ่งในที่นี้คือจุดซื้อขายในอดีต

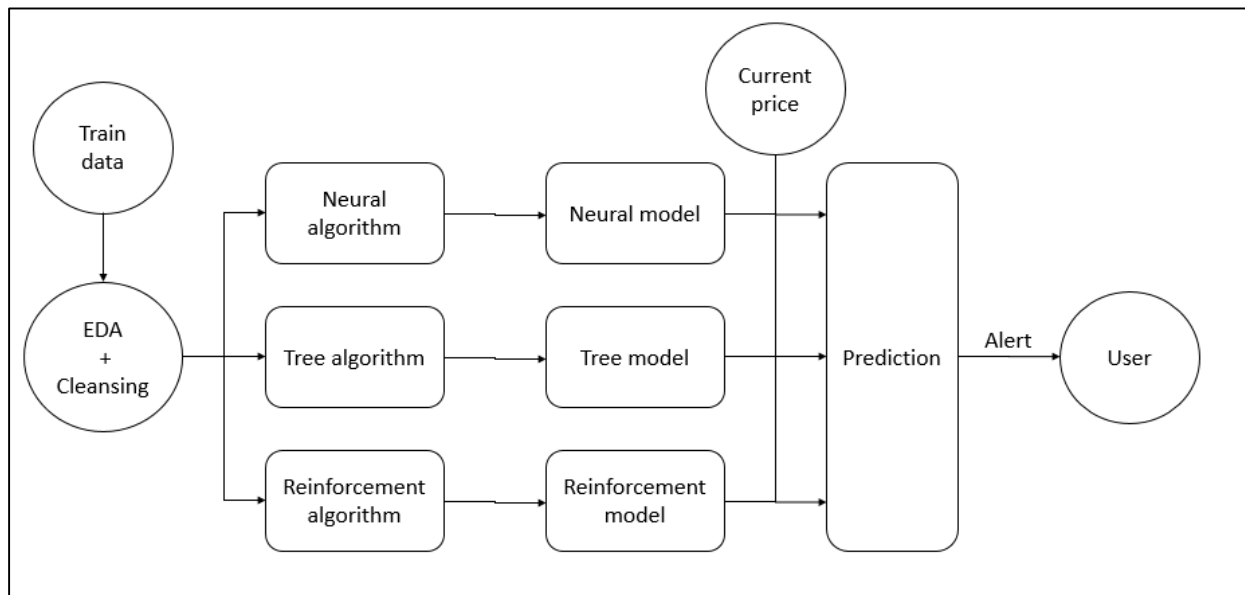
4) รวบรวมข้อมูลที่ต้องการและจัดการข้อมูลให้อยู่ในรูปแบบที่นำไปใช้เทรนต่อได้

5) ทำการเทรน โดยกลุ่มของเราเลือกโมเดลในการเทรนดังนี้

- ตระกูล Neural Network
- ตระกูล Tree
- Reinforcement Learning

6) ทดสอบโมเดลด้วยการ backtest โดยทดลองกับหุ้น 50 ตัว ที่มีความคล่องตัวสูง ซึ่งข้อมูลของหุ้นได้มาจากเว็บไซต์ Siamchart และ Yahoo โดยข้อมูลที่ได้จะมีค่า Date, Open, High, Low, Close, Adj Close และ Volume ตั้งแต่ปี 1970-2017

7) ปรับปรุงโมเดลโดยอาจย้อนไปทำ ตั้งแต่ข้อ 2 และ 4 เพื่อที่นำไป backtest



รูปที่ 3.2 ภาพแสดงขั้นตอนการทำงานภาพรวมของระบบ

จากรูปที่ 3.2 แสดงขั้นตอนการทำงานภาพรวมของระบบ ซึ่งแต่ละส่วนการทำงานมีรายละเอียดดังต่อไปนี้

3.2.1 Train Data หมายถึง ข้อมูล EOD ของหุ้นที่ได้ทำการรวบรวมมา เป็นข้อมูลดิบที่ยังไม่ผ่านการจัดสรรหรือการทำความสะอาดข้อมูล

3.2.2 Data Cleansing เป็นการนำ Train Data มาตรวจสอบและทำความสะอาดในส่วนต่างๆ สิ่ง ข้อมูลชุดนี้ยังคงมีปัญหาอยู่ คือ เรื่องความครบถ้วนสมบูรณ์ของข้อมูลรายวัน ตลาดหลักทรัพย์เปิดให้บริการวันจันทร์ถึงศุกร์ และอาจมีเหตุการณ์บางอย่างที่ทำให้ตลาดต้องหยุดการซื้อขายในบางวัน ซึ่งข้อมูลหุ้นของวันหยุดเหล่านั้นจะถูกบันทึกไว้เป็นค่า NaN ซึ่งหมายถึงการไม่มีค่า เราจึงต้องทำการเติมค่าในวันนั้น โดยหลักการการเติมค่าราคาหุ้นจะเป็นการทำให้ราคาวันที่ไม่ทำการเป็นราคาเดินของวันทำการล่าสุด ตัวอย่างเช่น ราคาปิดวันศุกร์อยู่ที่ 30.00 บาท ราคาปิดวันเสาร์และวันอาทิตย์ที่เป็นวันหยุดทำการก็จะทำการเปลี่ยนจาก NaN เป็น 30.00 บาทคงเดิม ที่ไม่ทำให้ราคาค่อยๆ เพิ่มขึ้นแบบเส้นตรงจนไปถึงราคาวันจันทร์นั้น เพราะจะเป็นการนำราคาในอนาคตมา ซึ่งไม่สะท้อนความเป็นจริง หากนำข้อมูลแบบนี้ไปเรียนรู้อาจจะเกิดความผิดพลาดเพราะโมเดลรู้อนาคตในช่วงวันเหล่านั้น

3.2.3 EDA เป็นการนำข้อมูลหุ้นมาลองวิเคราะห์และลอง Visualize ในรูปแบบกราฟและอื่นๆ เพื่อหาประเด็นที่อาจจะเป็นประโยชน์ต่อการทำความเข้าใจลักษณะของข้อมูลชุดนี้

3.2.4 Neural Algorithm และ Neural Model เป็นการนำข้อมูลที่ถูกทำความสะอาดตามกระบวนการแล้วมาใช้ในการทำโมเดล Neural Network การจะสร้างโมเดลขึ้นมาได้นั้น อันดับแรกต้องนำข้อมูลมาสร้างเป็น Feature ก่อน ตั้งแต่ขั้นตอนนี้เราสามารถปรับแต่งสิ่งต่างๆ ได้ว่าเราต้องการ Feature แบบไหน ต่อมาจึงกำหนดพารามิเตอร์ของอัลกอริธึมแล้วจึงนำข้อมูลมาเทรนโมเดล ข้อมูลแต่ละตัวจะถูกใช้ในการปรับค่าตัวแปรน้ำหนักจนได้โมเดลที่เหมาะสมต่อการไปทำนายผลจากข้อมูลใหม่

3.2.5 Tree Algorithm และ Tree model เป็นการนำข้อมูลที่ถูกทำความสะอาดตามกระบวนการแล้วมาใช้ในการทำโมเดลประเภท Tree

3.2.6 Reinforcement Algorithm และ Reinforcement Model เป็นการนำข้อมูลที่ผ่านมาผ่านการทำความสะอาดแล้วมาใช้ในการสร้างตัวแปรสภาพแวดล้อม และนำโมเดลมาทดลองเทรดหุ้นหลายพันครั้ง แต่ละครั้งโมเดลจะเรียนรู้จากความผิดพลาด เมื่อเทรนจนจบแล้วโมเดลจะสามารถทำอะไรได้สูงเพราะได้เรียนรู้ข้อมูลจำนวนมาก

3.2.7 Current Price หมายถึง ราคาหุ้นที่เป็นข้อมูลขาเข้า ณ ปัจจุบัน โมเดลจะต้องใช้ข้อมูลนี้ในการพิจารณาหาค่าทำนาย เราต้องดำเนินการกับข้อมูลนี้ให้กลายเป็น Feature เหมือนข้อมูลที่นำมาเทรนก่อน

3.2.8 Predict หมายถึง การนำข้อมูลขาเข้า ณ ปัจจุบันมาเข้าโมเดลแต่ละโมเดล และให้โมเดลทำนายผลลัพธ์ ซึ่งผลลัพธ์นี้จะถูกนำไปใช้ซื้อขายหุ้นต่อไป หากนำไปใช้ในผลิตภัณฑ์จริง ผลลัพธ์ส่วนนี้จะถูกส่งเป็นข้อความเตือนให้ผู้ใช้งาน

3.3 ตัวอย่างแอปพลิเคชันที่สามารถนำโครงงานไปปรับใช้ได้

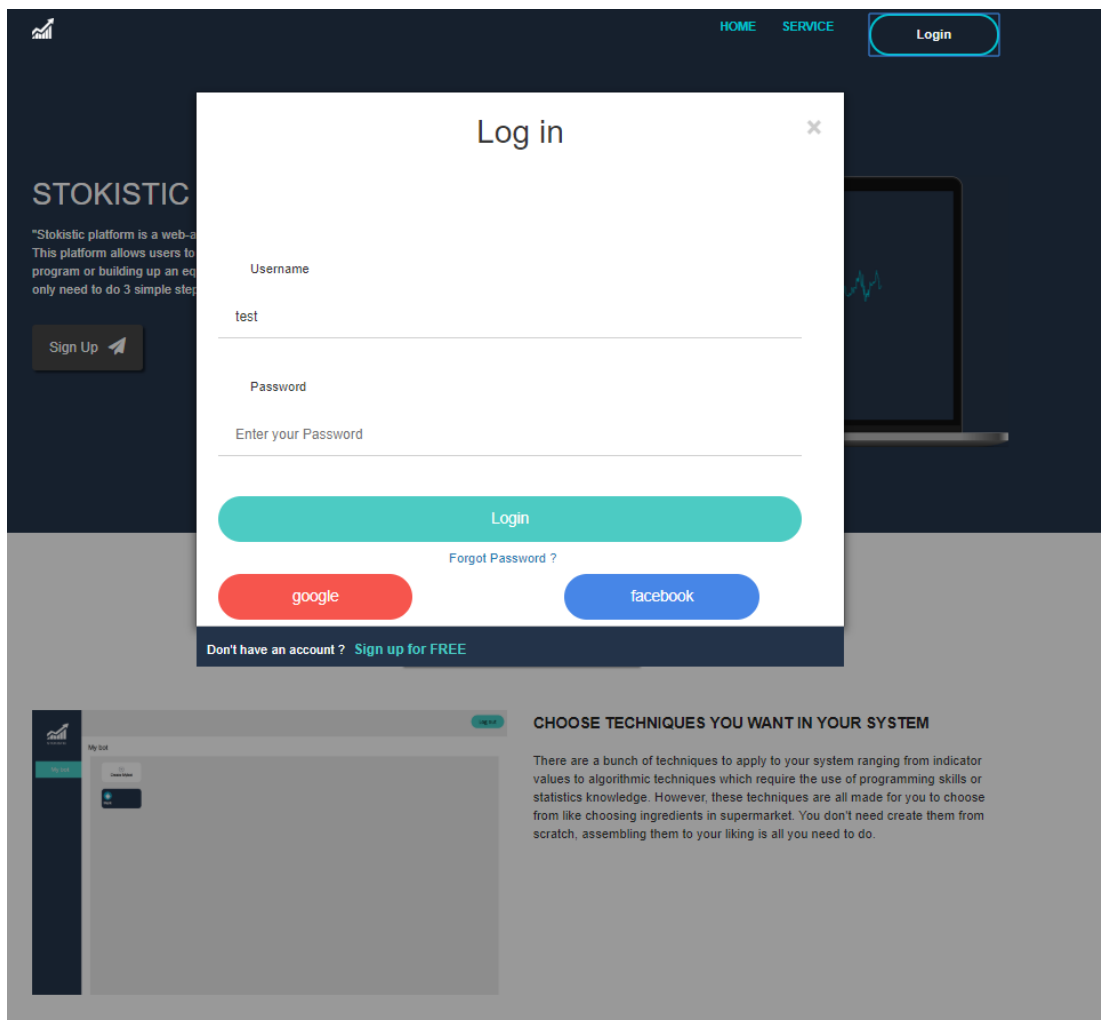
ตัว AI ที่ได้ถูกพัฒนาขึ้นนี้สามารถนำไปปรับใช้ให้เกิดเป็นผลิตภัณฑ์ได้ โดยคณะผู้จัดทำได้ทดลองสร้าง Web Application ที่เปรียบเสมือน MVP เพื่อทดสอบการใช้งานในกรณีที่ผลิตภัณฑ์ถูกพัฒนาขึ้นจริง

ผลิตภัณฑ์นี้จะช่วยให้ผู้ใช้งานสามารถสร้างระบบเตือนสัญญาณเทรดเชิงเทคนิคได้ โดยไม่จำเป็นต้องเขียนโปรแกรมหรือมีความรู้ความสามารถเฉพาะทาง ผลิตภัณฑ์นี้จะรวบรวมเอาเทคนิคต่างๆ ที่ถูกเขียนเป็นโปรแกรมแล้ว นำมาให้ผู้ใช้งานเลือกและปรับแต่งค่าบางอย่างที่มีความสำคัญต่อเทคนิคนั้น

ส่วนประกอบและการใช้งานผลิตภัณฑ์

1. การลงชื่อเข้าใช้โปรแกรม

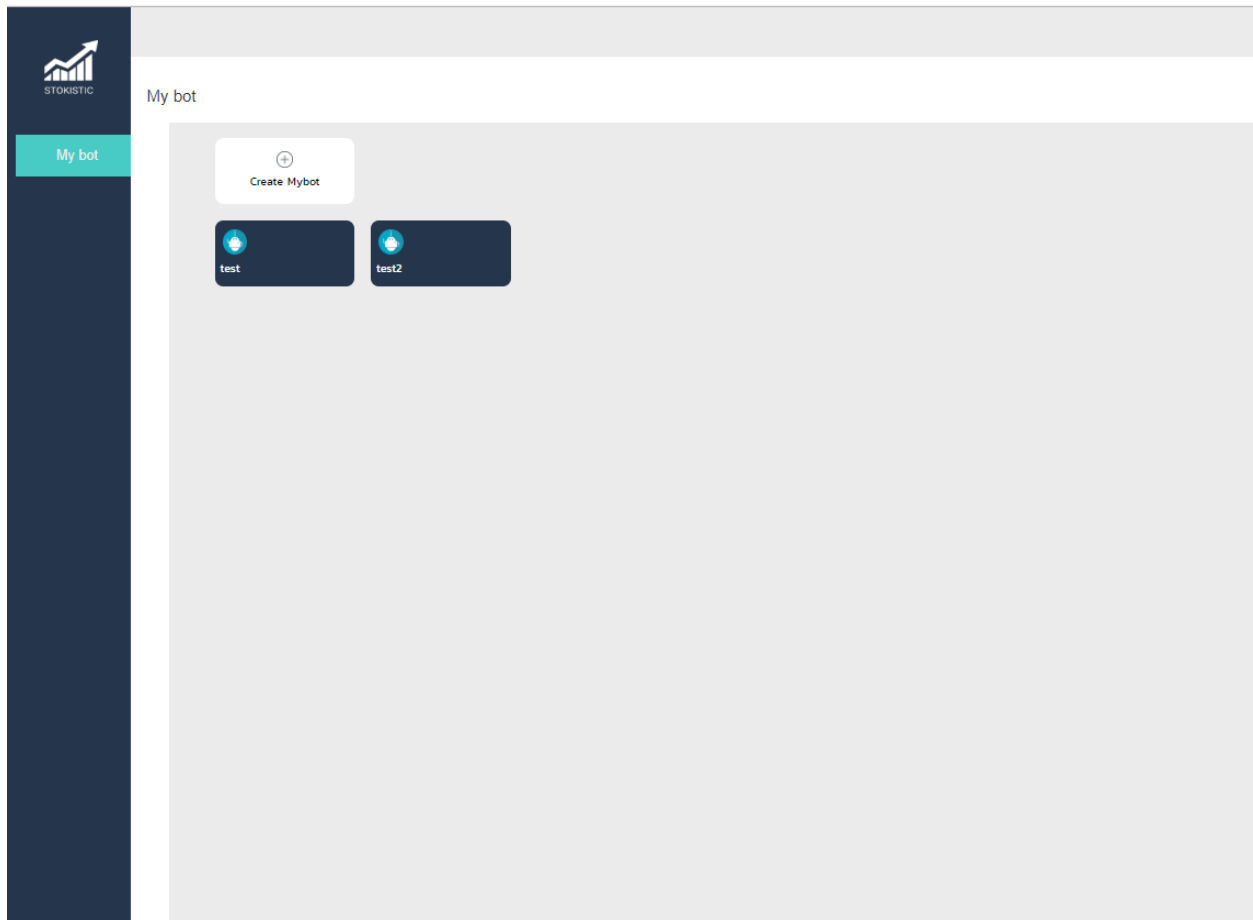
เป็นการล็อกอินเพื่อเข้าใช้งาน ผู้ใช้งานแต่ละคนจะมีระบบเทรดที่เคยสร้างไว้แล้วแตกต่างกัน



รูปที่ 3.3 หน้า Log In ของเว็บแอปพลิเคชัน

2. การดูระบบที่มีอยู่

หน้าแรกจะแสดงระบบเทรดที่เคยสร้างไว้แล้ว ผู้ใช้งานสามารถเข้าไปดูเทคนิคที่ใช้ได้ หรือเลือกที่จะสร้างระบบใหม่จากหน้านี้ก็ได้



รูปที่ 3.4 หน้าดูระบบเทรดที่มีอยู่

3. การสร้างระบบใหม่

ผู้ใช้งานสามารถสร้างระบบแจ้งเตือนใหม่ได้โดยกดที่สัญลักษณ์บวก “Create Mybot” เมื่อเข้ามาในหน้าสร้าง AI แล้ว ผู้ใช้งานจะสามารถกำหนดเทคนิคที่จะใช้ในแต่ละขั้นตอนได้ โดยมีทั้งหมด 5 ขั้นตอน ดังจะกล่าวถึงต่อไปนี้

เริ่มจากกำหนดจำนวนเงินลงทุน ผู้ใช้งานสามารถใส่เงินลงทุนลงไปได้เพื่อบอกกับระบบว่ามีเงินเริ่มต้นเท่าไร

My bot

Bot Name : test

เงินลงทุน	+
จำนวนเงิน : 20000 บาท	
เลือกหุ้น	+
คัดกรองหุ้น	+
เงื่อนไขการซื้อหุ้น	+
เงื่อนไขการขายหุ้น	+

Cancel Save Back tests Delete

รูปที่ 3.5 หน้ากำหนดจำนวนเงินต้น

ขั้นตอนถัดมาเป็นการเลือกหุ้นที่ต้องการให้ระบบนี้นำหุ้นอะไรมาพิจารณาบ้าง มีให้เลือกเป็นรายตัวหรือจะเลือกเป็น SET50 SET100 ก็ได้

Bot Name : test

เงินลงทุน	+
เลือกหุ้น	-

Search...

ALL	2S
50	7UP
	A
100	✓ AAV
	ABICO
	ABM
	ABPIF
	ACAP
	ACC
	ADAM
	ADB

คัดกรองหุ้น +

เงื่อนไขการซื้อหุ้น +

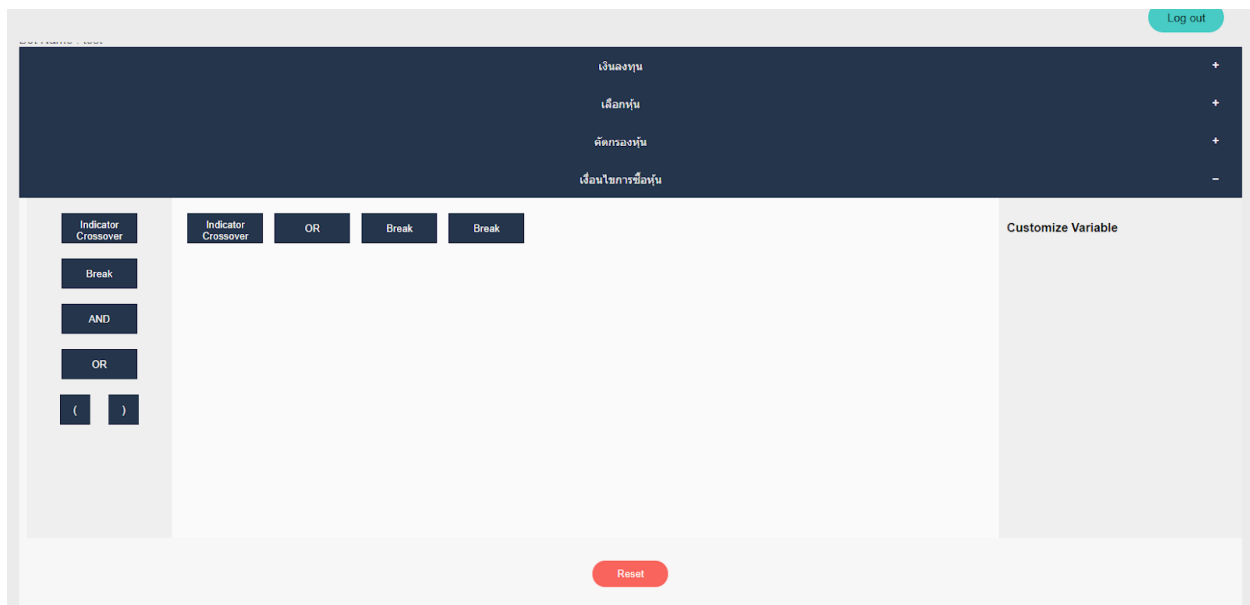
รูปที่ 3.6 หน้ากำหนดหุ้นที่ต้องการพิจารณา

เมื่อเลือกหุ้นแล้ว ขั้นตอนถัดไปจะเป็นส่วนการคัดกรองหุ้น ในส่วนนี้ผู้ใช้งานสามารถเลือกเทคนิคที่จะใช้ในการคัดกรองได้ เช่น การเลือกตามลำดับค่า volume หรือเลือกเฉพาะตัวที่มีค่า RSI เกิน 70 โดยลักษณะการกำหนดเงื่อนไขนั้นอยู่ในรูปของสมการทางตรรกศาสตร์ หากว่าเงื่อนไขเป็นจริงก็จะนำหุ้นเหล่านั้นมาทำการกำหนดสัญญาณซื้อขายต่อไป

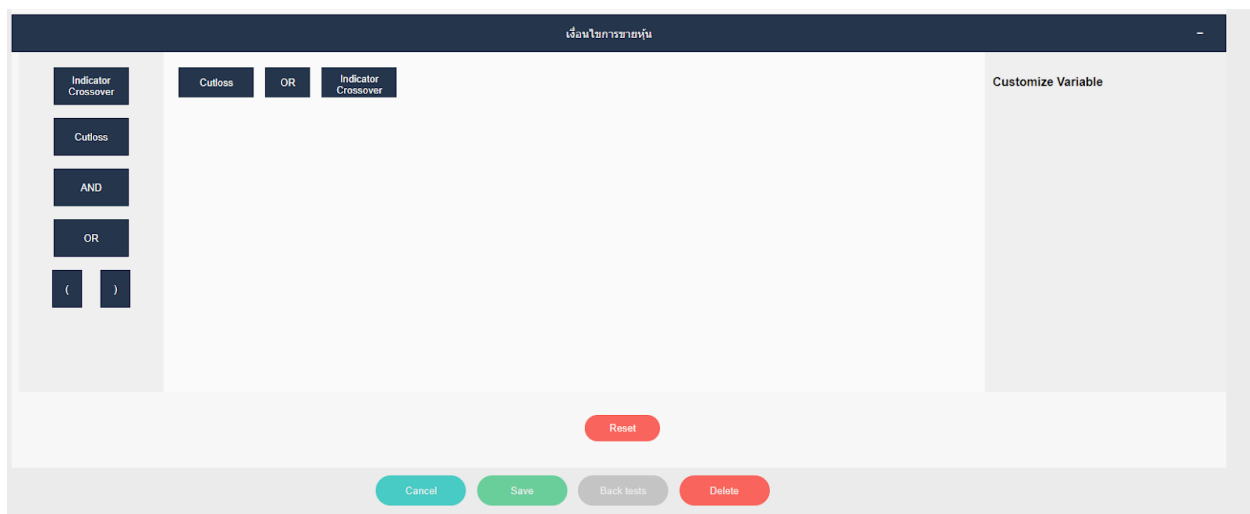
The screenshot shows a web interface for configuring a trading bot named 'test'. At the top right is a 'Log out' button. Below the bot name, there are three rows of configuration options: 'เงินลงทุน' (Investment) with a '+' button, 'เลือกหุ้น' (Select stock) with a '+' button, and 'คัดกรองหุ้น' (Filter stock) with a '-' button. The main area contains a logic builder with buttons for 'Ranking', 'Threshold', 'AND', 'OR', '(', and ')'. A sample logic expression is shown: '(Ranking AND Threshold) OR Threshold'. On the right, a 'Customize Variable' section shows 'volume' selected from a dropdown, with input fields for '1' and '2' separated by 'to', and 'Save' and 'Delete' buttons. A 'Reset' button is located at the bottom center.

รูปที่ 3.7 หน้ากำหนดเงื่อนไขการคัดกรองหุ้น

ในส่วนขั้นตอนเงื่อนไขซื้อและเงื่อนไขขายนั้น จะเป็นการกำหนดด้วยเทคนิคต่างๆ ว่า เมื่อใดจึงควรจะแจ้งเตือนไปยังผู้ใช้งานว่าควรซื้อหุ้นหรือขายหุ้น โดยการกำหนดเงื่อนไขก็ยังคงอยู่ในรูปแบบของสมการทางตรรกศาสตร์เช่นเดิม ผลงานจากโครงการของเราจะถูกนำมาใช้ในส่วนนี้ AI ถูกแปลงให้เป็นหนึ่งในเทคนิคให้ผู้ใช้งานเลือก เนื่องจากอย่างไรก็ตาม ผลลัพธ์ของ AI ก็เป็นคำแนะนำที่ให้ซื้อหรือขายเช่นเดียวกับเทคนิคที่ไม่ได้ใช้การเรียนรู้ของเครื่องอื่นๆ หมายความว่า AI แต่ละตัวสามารถนำมาใช้กับสมการทางตรรกศาสตร์ได้เช่นกัน



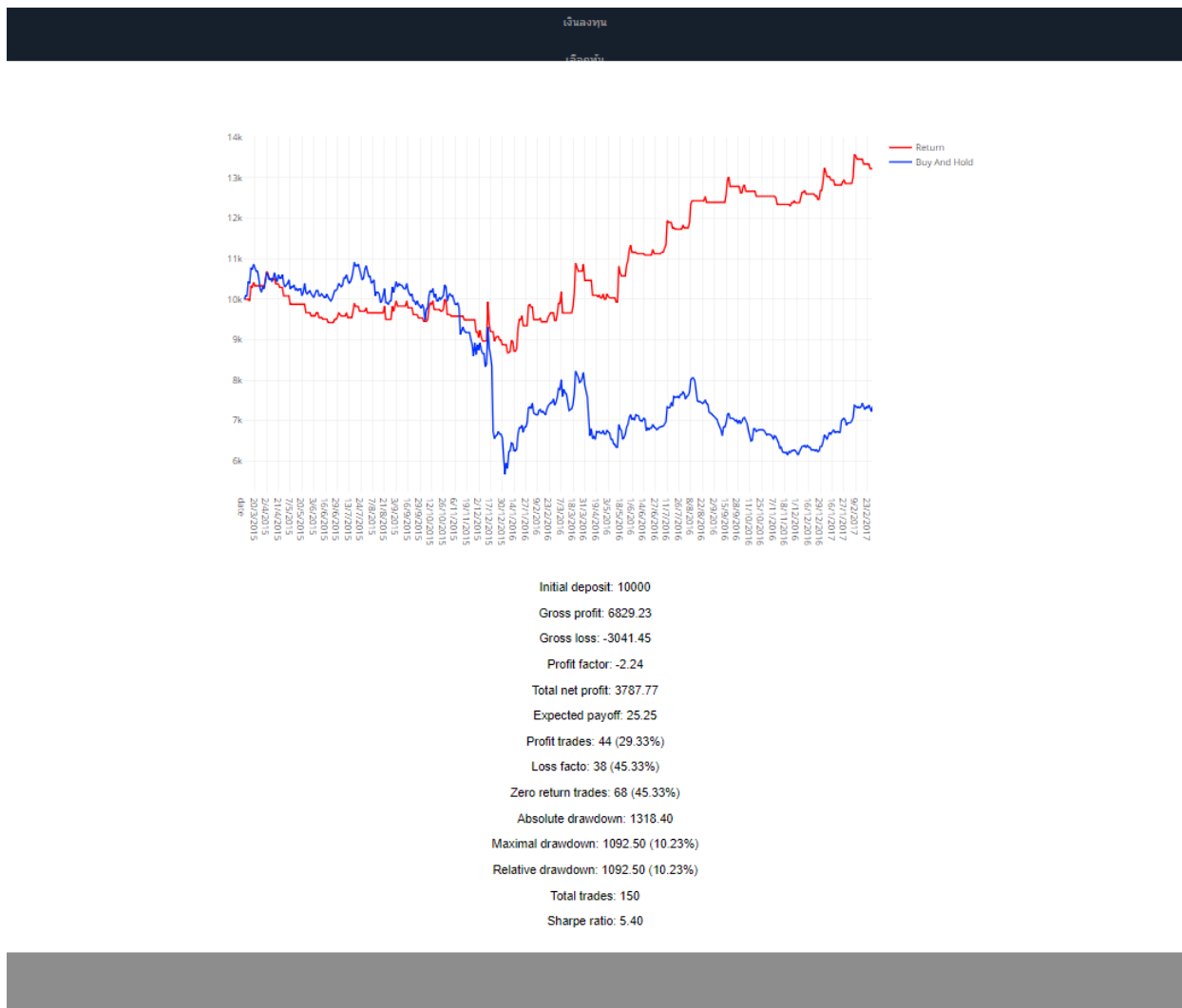
รูปที่ 3.8 หน้ากำหนดเงื่อนไขสัญญาณการซื้อหุ้น



รูปที่ 3.9 หน้ากำหนดเงื่อนไขสัญญาณการขายหุ้น

4. การใช้งานระบบการทดสอบย้อนหลัง

ผู้ใช้งานสามารถทดสอบประสิทธิภาพของระบบเทรดที่สร้างขึ้นได้ โดยใช้ระบบทดสอบย้อนหลังช่วยวัดผลให้ เพียงกดที่ “Backtest ” หลังจากกำหนดเทคนิคในขั้นตอนก่อนหน้าเสร็จสิ้น ระบบจะแสดงผลการทดสอบย้อนหลังในรูปแบบกราฟผลตอบแทนเปรียบเทียบระหว่างระบบเทรดกับการ buy and hold และค่าตัวเลขต่างๆ ดังรูปที่ 3.10

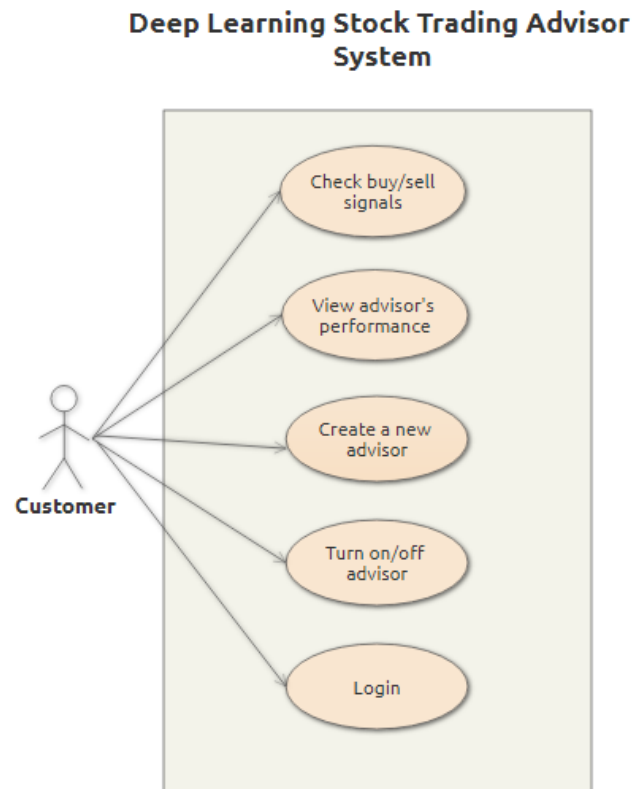


รูปที่ 3.10 หน้าแสดงผลการทดสอบย้อนหลัง

5. ระบบแจ้งเตือนสัญญาณเทรดหุ้นอัตโนมัติ

สุดท้ายเมื่อผู้ใช้งานได้สร้างระบบเทรดที่พอใจแล้ว ระบบจะทำการเก็บข้อมูลหุ้นใหม่ในทุกวัน และเมื่อราคาหุ้นวันใดเข้าข่ายสัญญาณซื้อขาย ระบบจะส่งสัญญาณเตือนไปถึงผู้ใช้งานในรูปแบบของข้อความตามช่องทางที่ผู้ใช้งานกำหนดไว้ โดยข้อความจะระบุรายละเอียดการซื้อขายและบอกเงื่อนไขที่ทำให้เกิดสัญญาณนี้ขึ้นมา

3.4 Usecase Diagram



รูปที่ 3.11 Usecase diagram แสดงการใช้งานที่ผู้ใช้สามารถทำกับระบบได้

บทที่ 4

ผลการวิจัยและอภิปรายผล

คณะผู้จัดทำได้ทดลองสร้างโมเดล Machine Learning เพื่อใช้ในการคำนวณจุดซื้อขายหุ้นที่เหมาะสมเป็นตัวต้นแบบโดยใช้อัลกอริธึม Random Forest Classifier ซึ่งอยู่ในตระกูลของ Decision Tree และ Reinforcement learning โมเดล Random forest สามารถทำผลตอบแทนได้ดีพอสมควร แต่ไม่สามารถทำผลตอบแทนได้ดีกับหุ้นทุกตัวใน SET50

4.1 การสร้างระบบการทดสอบย้อนหลัง

ระบบการทดสอบย้อนหลังถือเป็นหลักสำคัญในการประเมินประสิทธิภาพของบอทเทรดที่ได้สร้างขึ้น ระบบทดสอบย้อนหลังสามารถบอกได้ว่า ถ้าหากนำบอทนี้ไปเทรดหุ้นในอดีต ผลลัพธ์จะเป็นเช่นไร ค่าที่ใช้ในการวัดผลก็จะมีทั้งค่าทางสถิติและค่าทางพอร์ตหุ้น ดังที่ได้บอกไว้แล้ว ในบทก่อนหน้านี้ อย่างไรก็ตาม ค่าที่ได้จากการทดสอบนี้ไม่สามารถสะท้อนการทำงานในอนาคตของบอทได้แม่นยำ 100 เปอร์เซ็นต์ เนื่องจากอนาคตเป็นสิ่งที่ไม่แน่นอน อาจเกิดการเปลี่ยนแปลงที่ทำให้หุ้นนั้นๆ เปลี่ยนพฤติกรรมได้

ในช่วงแรก เราได้เขียนระบบทดสอบย้อนหลังมาก่อน 1 เวอร์ชัน แต่หลังจากนั้นเราได้พบข้อผิดพลาดว่าการทดสอบไม่ได้สะท้อนความเป็นจริง เราจึงได้แก้ไขและพัฒนา ระบบทดสอบย้อนหลังเวอร์ชัน 2 เพื่อให้การทดสอบสะท้อนความเป็นจริงมากขึ้น

4.2 ข้อมูลที่ใช้ในการสร้างโมเดล

ใช้ข้อมูลราคาปิดหุ้นวันต่อวัน คือ Open, High, Low, Close, Adjusted Close และ Volume ซึ่งได้มาจาก Yahoo financial API ใน SET50 ตั้งแต่ปี 2000-2018

	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-01-04	2.04167	2.10833	2.00000	2.06667	0.688321	11910000.0
1	2000-01-05	1.91667	1.94167	1.85833	1.89167	0.630035	11983200.0
2	2000-01-06	1.87500	1.87500	1.77500	1.81667	0.605056	6333600.0
3	2000-01-07	1.83333	1.89167	1.77500	1.87500	0.624483	9670800.0
4	2000-01-10	1.89167	1.89167	1.81667	1.83333	0.610605	3608400.0
5	2000-01-11	1.81667	1.81667	1.75000	1.79167	0.596730	2916000.0
6	2000-01-12	1.70833	1.91667	1.70833	1.91667	0.638362	7065600.0
7	2000-01-13	1.94167	1.95833	1.85833	1.85833	0.618931	7958400.0
8	2000-01-14	1.79167	1.85833	1.77500	1.83333	0.610605	8380800.0
9	2000-01-17	1.79167	1.83333	1.79167	1.83333	0.610605	4696800.0
10	2000-01-18	1.83333	1.94167	1.83333	1.91667	0.638362	23371200.0
11	2000-01-19	1.91667	1.91667	1.85833	1.87500	0.624483	20796000.0
12	2000-01-20	1.87500	1.98333	1.87500	1.89167	0.630035	16411200.0
13	2000-01-21	1.89167	1.94167	1.83333	1.83333	0.610605	8991600.0
14	2000-01-24	1.85833	1.85833	1.81667	1.85833	0.618931	7954800.0
15	2000-01-25	1.79167	1.89167	1.79167	1.89167	0.630035	2275200.0
16	2000-01-26	1.91667	1.91667	1.79167	1.81667	0.605056	10819200.0
17	2000-01-27	1.81667	1.85833	1.77500	1.83333	0.610605	4020000.0
18	2000-01-28	1.81667	1.81667	1.79167	1.81667	0.605056	6882000.0
19	2000-01-31	1.79167	1.89167	1.79167	1.87500	0.624483	20373600.0
20	2000-02-01	1.85833	1.87500	1.77500	1.85833	0.618931	40005600.0
21	2000-02-02	1.83333	1.83333	1.79167	1.83333	0.610605	2016000.0
22	2000-02-03	1.81667	1.81667	1.77500	1.77500	0.591178	6480000.0
23	2000-02-04	1.81667	1.81667	1.77500	1.79167	0.596730	2836800.0
24	2000-02-07	1.79167	1.81667	1.77500	1.81667	0.605056	1088400.0
--	----	----	----	----	----	----	----

รูปที่ 4.1 Raw Data ของหุ้นที่จะนำมาสร้างโมเดลและทดสอบ

โดยนำ Data ที่ได้มาทำการ Preprocessing เพื่อให้สามารถนำไปใช้ได้ ดังนี้

1. Fill forward, Fill backward และลบ sample ที่มีค่า NAN
2. แบ่งข้อมูลออกเป็น train dataset ในปี 2000 - 2016 และ test dataset ในปี 2016 - 2018

4.3 การทดสอบโมเดล

- วัดจากการนำเอาโมเดลไปทดสอบเทรดบนหุ้นทั้งหมด 50 ตัวใน SET50 ในปัจจุบัน โดยใช้วิธีการทดสอบย้อนหลังตั้งแต่ พฤษภาคม 2016 - พฤษภาคม 2018
- ใช้ Return เป็นตัววัดผลโดยเปรียบเทียบกับ Baseline คือ การ Buy and hold ซึ่งการ Buy and hold ในช่วงพฤษภาคม 2016 - พฤษภาคม 2018 สามารถทำ Return ได้ 30.95%

4.4 ขั้นตอนการสร้างและผลการทดสอบของแต่ละโมเดล

4.4.1 DecisionTreeClassifier

DecisionTreeClassifier เป็นอัลกอริทึมที่ได้รับความนิยมสูง และสามารถแก้ปัญหได้อย่างหลากหลาย เราจึงใช้โมเดลนี้เพื่อทำการทดสอบเป็นวิธีการแรก

4.4.1.1 โมเดล DecisionTreeClassifier เวอร์ชันที่ 1

ทดลองใช้ข้อมูลแค่ Open, High, Low, Close, Adj Close, Volume เป็น Features และ Label โดยใช้วิธี Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น log return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ให้ label เป็น 0 และใช้ parameter ในการเทรนดังนี้

```
criterion='gini',splitter='best',max_depth=None, min_samples_split=2,min_samples_leaf=1,
min_weight_fraction_leaf=0.0,max_features=None, random_state=None,max_leaf_nodes=None,
min_impurity_decrease=0.0,min_impurity_split=None, class_weight=None, presort=False
```

ผลการทดสอบ

Return 14.32% ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50 เนื่องจากโมเดลไม่สามารถที่จะเรียนรู้และทำนายได้ ซึ่งโมเดลจะทำนายทุกวันว่าข้อมูลหุ้นจะขึ้นทุกวันหรือลงทุกวัน เนื่องจากข้อมูลไม่ได้ผ่านการ normalize และ raw data ที่ใช้เป็น features เพียงวันเดียวไม่สามารถที่จะเห็นรูปแบบในการทำนายการซื้อขายได้

วิธีการแก้ไข

เพิ่ม features ให้มากกว่านี้ เนื่องจาก features น้อยเกินไป

4.4.1.2 โมเดล DecisionTreeClassifier เวอร์ชันที่ 2

เพิ่มข้อมูลที่ใช้เป็น features จากเวอร์ชันก่อนหน้า โดยคำนวณ Indicator เพิ่มเข้าไป ซึ่งได้แก่ SMA5, SMA12, SMA26, EMA5, EMA12, EMA26, RSI7, RSI14, MACD12, MACD26 และใช้โมเดลเดิมเหมือนกับเวอร์ชันที่ 1

ผลการทดสอบ

Return 16.11% ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50

วิธีการแก้ไข

เพิ่ม features ให้มากกว่านี้ คือ เพิ่มราคาหุ้นวันย้อนหลังก่อนหน้าเข้าไปด้วย เนื่องจากเวลาที่คนเราเทรดหุ้นก็จะดูส่วนนี้ประกอบและลบบาง features ออก คือ SMA5, SMA12, SMA26, EMA5, EMA12, EMA26 ไม่สามารถทำให้โมเดลทำงานได้ดีขึ้น แต่กลับทำให้โมเดลแย่ลงจากการที่ได้ลองทดสอบ อย่างไรก็ตาม Indicator อย่าง RSI สามารถทำให้โมเดลดีขึ้นได้ จึงได้เพิ่ม Indicator ADX เข้าไปด้วย เนื่องจาก เป็น Indicator ที่ผ่านการ normalize มาแล้ว และนำ Open - Close และ High - Low ใส่เข้าไปด้วย ซึ่งจากการทดสอบพบว่า

สามารถทำให้โมเดลทำงานได้แม่นยำขึ้น นอกจากนี้เราจะนำข้อมูลทั้งหมด normalize ก่อน เนื่องจากได้ทดสอบมาแล้วว่าการทำ normalization ข้อมูลก่อนจะทำให้โมเดลสามารถทำงานได้มีประสิทธิภาพที่ดีกว่า

4.4.1.3 โมเดล DecisionTreeClassifier เวอร์ชันที่ 3

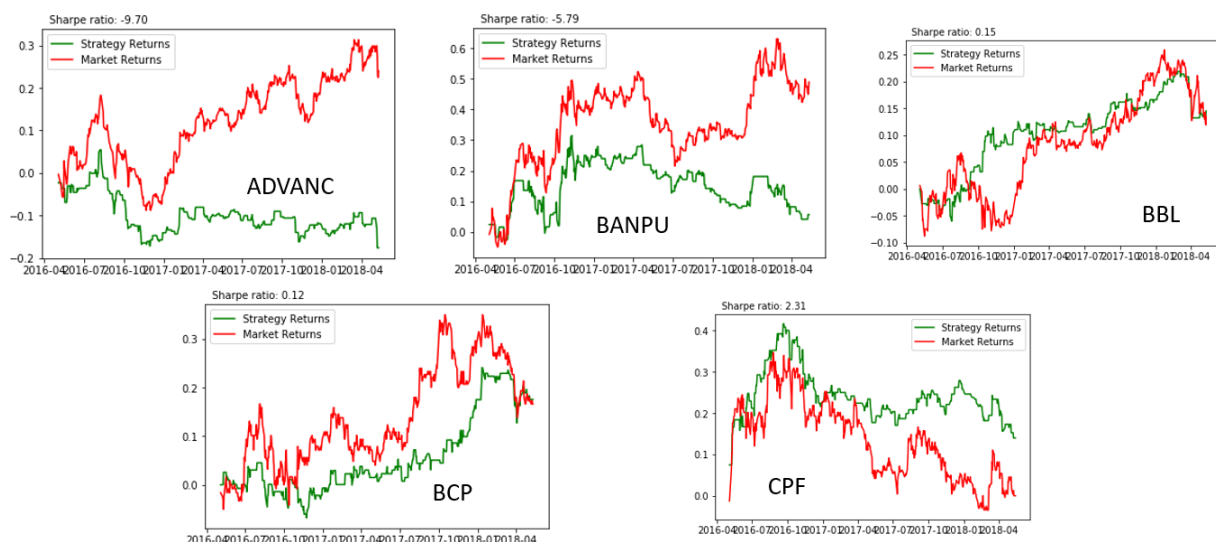
ขั้นตอนการสร้างโมเดล

1. ทำข้อมูลทั้งหมดให้เป็น standard โดยลบด้วย mean และ scale ให้เป็น unit variance โดยคิดจาก train dataset
2. Insert Indicator เพิ่มเข้าไปดังนี้ RSI, ADX
3. นำ open, high, low, close, open-close, high-low, RSI 7, RSI 14, ADX7, ADX14 ของวันย้อนหลัง 5 วันมา insert ในแต่ละ sample
4. Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น log return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ ให้ label เป็น 0
5. ใช้พารามิเตอร์เหล่านี้ในการเทรน

criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False

ผลการทดสอบ

Return 23.43 % ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50



รูปที่ 4.2 ผลการทดสอบโมเดล DecisionTreeClassifier

จากรูปที่ 4.2 เป็นผลการทดสอบโมเดล DecisionTreeClassifier เวอร์ชันที่ 3 เมื่อกราฟสีแดงคือ Return ของตลาด และกราฟสีเขียวคือ Return ของโมเดล DecisionTreeClassifier จะสังเกตได้ว่าโมเดลยังไม่สามารถให้ Return ได้มากกว่า Return ของตลาดในหุ้นบางตัว อย่างไรก็ตาม ยังมีหุ้นที่โมเดลสามารถทำงานได้ดีอย่างหุ้น BBL และ CPF

4.4.1.4 วิธีการพัฒนาโมเดล

ทางคณะผู้จัดทำคิดว่า เราจะสามารถเพิ่มประสิทธิภาพของโมเดล Decision Tree ได้โดยใช้วิธีการ Ensemble ซึ่งเป็นโมเดลในหัวข้อถัดไป

4.4.2. RandomForestClassifier

Random forest classifier เป็นโมเดลที่ใช้เพิ่มประสิทธิภาพของโมเดลอย่าง Decision Tree และสามารถทำ Ensemble กันได้อย่างง่าย เพื่อให้ Decision Tree มีประสิทธิภาพมากขึ้นโดย

4.4.2.1 โมเดล RandomForestClassifier เวอร์ชันที่ 1

ใช้ Open, High, Low, Close, Open-Close, High-Low, RSI 7 , RSI 14, ADX7, ADX14 ของวันปัจจุบันและย้อนหลัง 5 วันมา insert ในแต่ละ sample Label โดยใช้วิธี Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น log return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ให้ label เป็น 0 และใช้ parameter ในการเทรนดังนี้

```
n_estimators=10,criterion='gini',
max_depth=None,min_samples_split=2,min_samples_leaf=1,
min_weight_fraction_leaf=0.0,max_features='auto',
max_leaf_nodes=None,min_impurity_decrease=0.0, min_impurity_split=None,
bootstrap=True, oob_score=False, n_jobs=1, random_state=None, verbose=0,
warm_start=False, class_weight=None
```

ผลการทดสอบ

Return แต่ละครั้งได้ค่าไม่เท่ากัน เนื่องจากการ Random Decision Tree ซึ่งมี Features แตกต่างกัน ซึ่งความไม่แน่นอนนี้ทำให้ไม่สามารถนำไปใช้จริงได้

วิธีการแก้ไข

Tune model โดยเพิ่ม n_estimators และปรับพารามิเตอร์ n_jobs=-1 เพื่อใช้จำนวน process ของ CPU ทั้งหมดในการคำนวณเพื่อให้ได้ผลลัพธ์ที่เร็วขึ้น

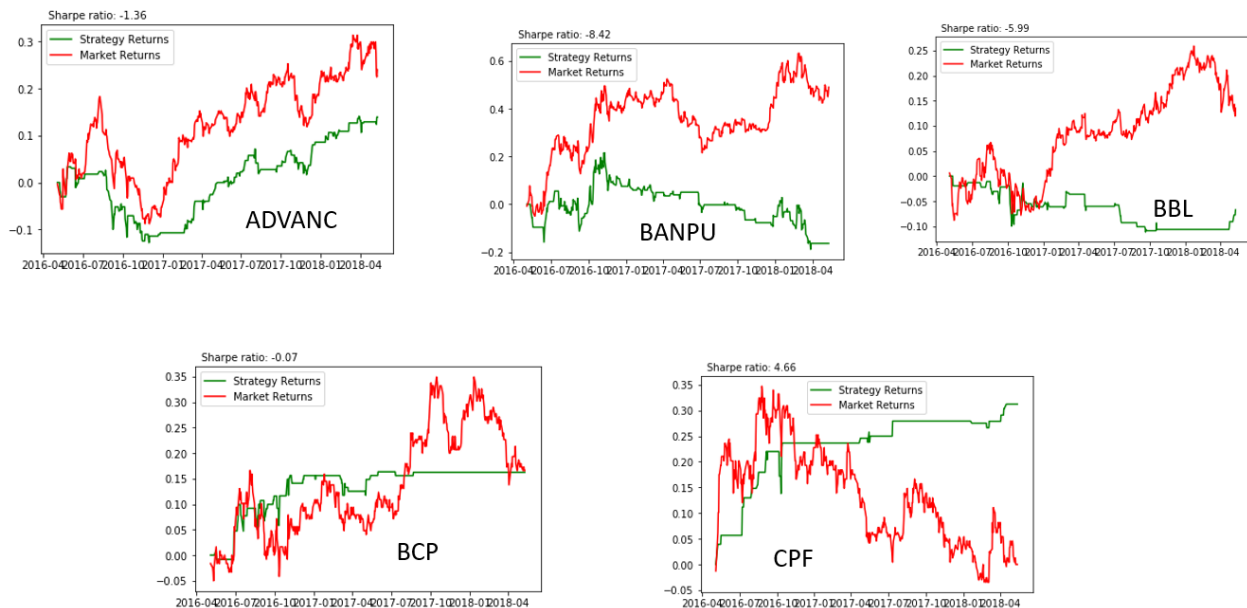
4.4.2.2 โมเดล RandomForestClassifier เวอร์ชันที่ 2

ขั้นตอนการสร้างโมเดล

1. ทำข้อมูลทั้งให้เป็น standard โดยลบด้วย mean และ scale ให้เป็น unit variance โดยคิดจาก train dataset
2. Insert Indicator เพิ่มเข้าไปดังนี้ RSI, ADX
3. นำ open, high, low, close, open-close, high-low, RSI 7 , RSI 14, ADX7, ADX14 ของวันย้อนหลัง 5 วันมา insert ในแต่ละ sample
4. Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ ให้ label เป็น 0
5. ใช้พารามิเตอร์ `n_estimators=4000, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=-1, random_state=None, verbose=0, warm_start=False, class_weight=None`

ผลการทดสอบ

Return 21.48% ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50



รูปที่ 4.3 ผลการทดสอบโมเดล RandomForestClassifier

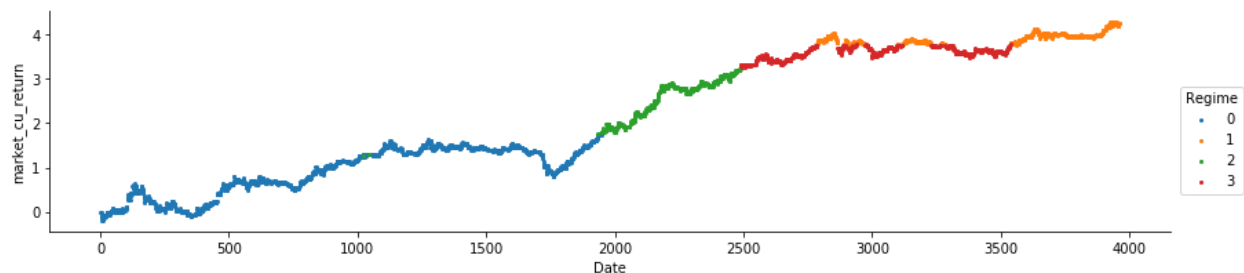
จากรูปที่ 4.3 เป็นผลการทดสอบโมเดล RandomForestClassifier เวอร์ชันที่ 2 เมื่อกราฟสีแดงคือ Return ของตลาด และกราฟสีเขียวคือ Return ของโมเดล RandomForestClassifier จะสังเกตได้ว่าโมเดลนี้ยังไม่สามารถให้ Return ได้มากกว่า Return ของตลาด โดยให้ Return เพียง 21.48% ซึ่งมีประสิทธิภาพน้อยกว่าโมเดล DecisionTreeClassifier ที่ให้ Return 23.43%

4.4.2.3 วิธีการพัฒนาโมเดล

ทางคณะผู้จัดทำคิดว่า เราจะสามารถเพิ่มประสิทธิภาพได้โดยใช้ Random Forest ร่วมกับโมเดลอื่น โดยเฉพาะร่วมกับโมเดลของ Unsupervised learning

4.4.3. RandomForest ร่วมกับ GaussianMixture (Unsupervised learning)

ใช้ RandomForest เพื่อเพิ่มประสิทธิภาพของ Decision Tree รวมกับใช้ GaussianMixture ที่เป็น unsupervised learning ในการสร้าง Features ที่เรียกว่า trends ขึ้นมา



รูปที่ 4.4 ผลลัพธ์ของข้อมูลหุ้น เมื่อใช้ GaussianMixture ในการ cluster

4.4.3.1 โมเดล RandomForest ร่วมกับ GaussianMixture เวอร์ชันที่ 1

ใช้ GaussianMixture ในการ Cluster ข้อมูลออกเป็น 4 ส่วนโดยแบ่งเป็น ข้อมูลส่วน trend up, trend down, sideways, และ unknow ซึ่งแทนด้วยค่า categorical number เป็น features และใช้ Open, High, Low, Close, Open-Close, High-Low, RSI 7, RSI 14, ADX7, ADX14 ของวันปัจจุบันและย้อนหลัง 5 วันมา insert ในแต่ละ sample เป็น features ใช้วิธี Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น log return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ให้ label เป็น 0 และใช้ parameter ในการเทรนดังนี้

```
n_estimators=10,criterion='gini',
max_depth=None,min_samples_split=2,min_samples_leaf=1,
min_weight_fraction_leaf=0.0,max_features='auto',
max_leaf_nodes=None,min_impurity_decrease=0.0, min_impurity_split=None,
```

```
bootstrap=True, oob_score=False, n_jobs=1, random_state=None, verbose=0,  
warm_start=False, class_weight=None
```

ผลการทดสอบ

ได้ Return ที่สูงมากเกินไปแบบผิดปกติ เนื่องจากผู้จัดทำไม่ได้คำนึงถึงว่า เมื่อนำข้อมูลทั้งหมดผ่านโมเดล GaussianMixture แล้ว ข้อมูลทั้งหมดจะถูก cluster ซึ่งขัดกับความเป็นจริง คือ เราไม่สามารถจะนำข้อมูลทั้งหมดมา cluster ในการเทรดจริงได้

วิธีการแก้ไข

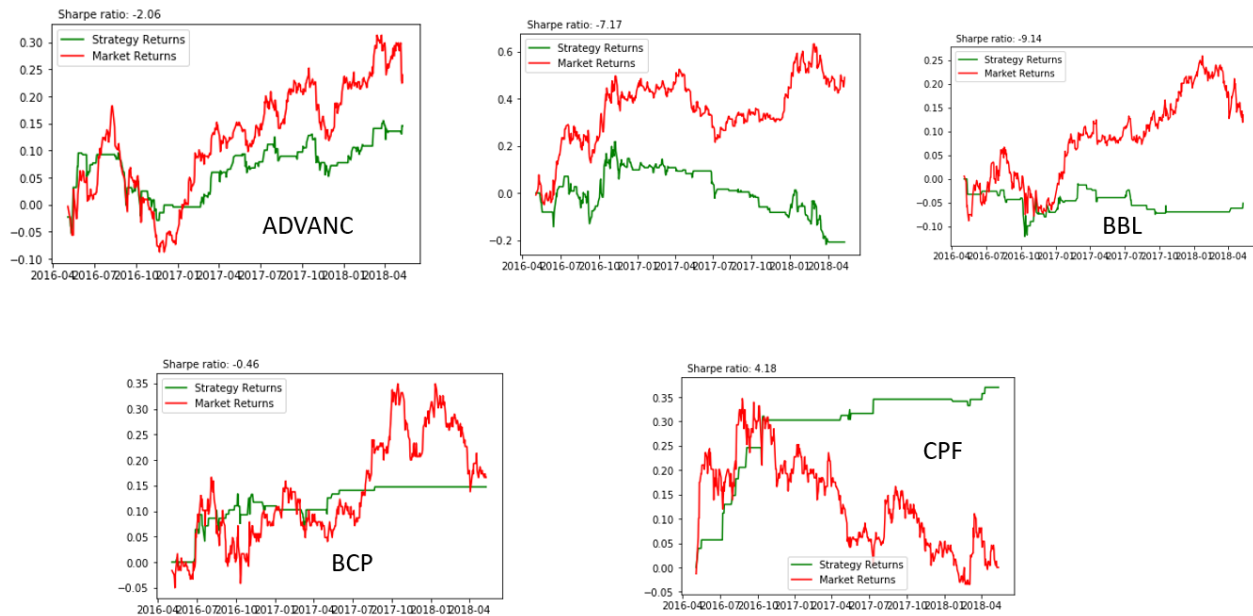
นำข้อมูล feed เข้าแต่ละวันหลังจากนั้นจึง cluster ตั้งแต่วันแรกถึงวันที่ข้อมูล feed เข้ามาล่าสุด

4.4.3.2 โมเดล RandomForest ร่วมกับ GaussianMixture เวอร์ชันที่ 2

ขั้นตอนการสร้างโมเดล

1. ทำข้อมูลทั้งให้เป็น standard โดยลบด้วย mean และ scale ให้เป็น unit variance โดยคิดจาก train dataset
2. Insert Indicator เพิ่มเข้าไปดังนี้ RSI, ADX
3. นำ open, high, low, close, open-close, high-low, RSI 7 , RSI 14, ADX7, ADX14 ของวันย้อนหลัง 5 วันมา insert ในแต่ละ sample
4. Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ ให้ label เป็น 0
5. ใช้ GaussianMixture ที่เป็น Unsupervised learning ในการหา trends ของ ราคาหุ้นว่าเป็นแบบใด ซึ่งแบ่งได้ 4 category คือ trends up, trends down, no trends, unknown
6. ใช้ parameters ดังนี้ n_estimators=4000,criterion='gini',
max_depth=None,min_samples_split=2,min_samples_leaf=1,
min_weight_fraction_leaf=0.0,max_features='auto',
max_leaf_nodes=None,min_impurity_decrease=0.0,
min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=-1,
random_state=None, verbose=0, warm_start=False, class_weight=None

ผลการทดสอบ



รูปที่ 4.5 ผลการทดสอบโมเดล RandomForest ร่วมกับ GaussianMixture

Return 21.47% ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50 ซึ่งได้ Return ที่ต่ำกว่าแบบไม่ใช้ Unsupervised learning รวมด้วย และนอกจากนี้ยังใช้การเทรนโมเดลที่นานมาก เพราะต้องทำ Clustering ทุกครั้งและทุกวัน

4.4.4. Neural network

ลองใช้ Multi-layer Perceptron อย่างง่ายเพื่อ Classify และทดสอบว่าโมเดลที่ใช้ Neural Network นี้สามารถทำผลได้ดีหรือไม่

4.4.4.1 โมเดล Neural network เวอร์ชันที่ 1

ใช้ Recurrent Neural Networks ในการ classification ส่วน features และใช้ Open, High, Low, Close, Open-Close, High-Low, RSI 7 , RSI 14, ADX7, ADX14 ของวันปัจจุบันและย้อนหลัง 5 วันมา insert ในแต่ละ sample เป็น features ใช้วิธี Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น log return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ให้ label เป็น 0 และใช้ parameter ในการเทรนดังนี้ โดย Architecture จะใช้ RNN 100 node ในเลเยอร์แรกและ RNN 100 node ในเลเยอร์ที่สอง หลังจากนั้นเลเยอร์สุดท้ายเป็น output ที่มี dense 2 node

ผลการทดสอบ

โมเดลบอกกับหุ้นทุกวันทุกตัวว่า หุ้นราคาจะลงในวันรุ่งขึ้น อาจเนื่องมาจากโมเดลมี parameter มากเกินไป และข้อมูลหุ้นที่มีจำนวนน้อยเช่นกัน จึงทำให้ไม่สามารถเทรน network นี้ได้

วิธีการแก้ไข

เปลี่ยน architecture ใหม่ให้โมเดลมี parameter น้อยลง โดยเปลี่ยนเป็น feed forward

ทั้งหมด

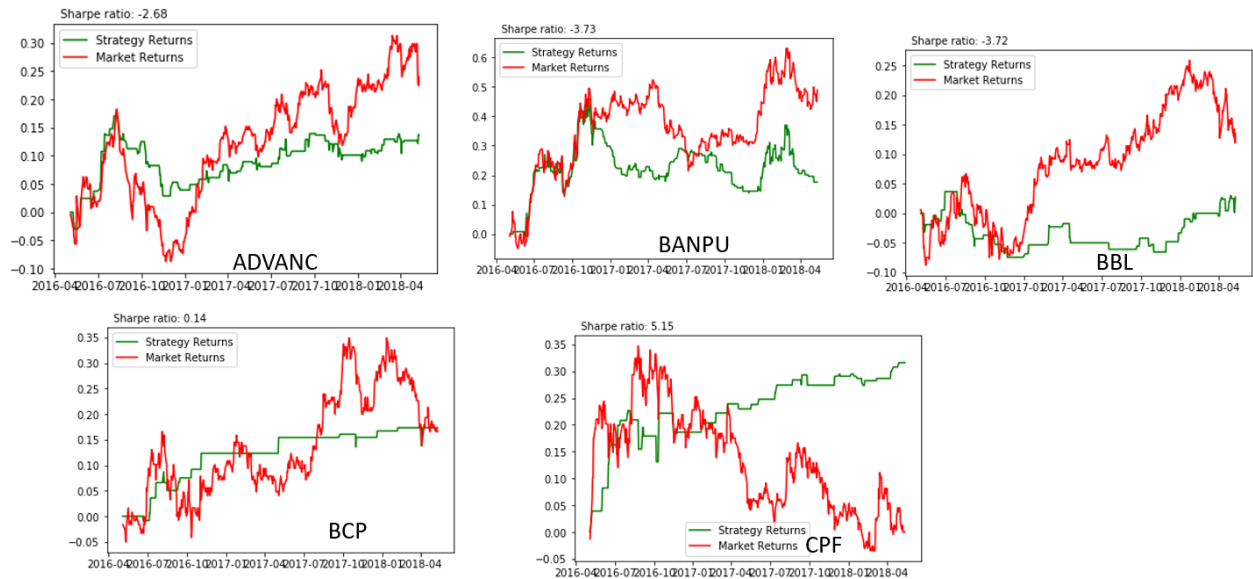
4.4.4.2 โมเดล Neural network เวอร์ชันที่ 2

ขั้นตอนการสร้างโมเดล

1. ทำข้อมูลทั้งให้เป็น standard โดยลบด้วย mean และ scale ให้เป็น unit variance โดยคิดจาก train dataset
2. Insert Indicator เพิ่มเข้าไปดังนี้ RSI, ADX
3. นำ open, high, low, close, open-close, high-low, RSI 7 , RSI 14, ADX7, ADX14 ของวันย้อนหลัง 5 วันมา insert ในแต่ละ sample
4. Label Data โดยใช้ log return ถ้าวันรุ่งขึ้น return มากกว่า 0 ให้ Label เป็น 1 แต่ถ้าไม่ใช่ให้ label เป็น 0
5. ใช้พารามิเตอร์ดังนี้(activation='relu',alpha=1e-05,batch_size='auto',beta_1=0.9, beta_2=0.999,early_stopping=False,epsilon=1e-08, hidden_layer_sizes=(100,100),learning_rate='constant',learning_rate_init=0.001,max_iter=200,momentum=0.9,nesterovs_momentum=True, power_t=0.5, random_state=1,shuffle=True,solver='lbfgs', tol=0.0001,validation_fraction=0.1, verbose=False,warm_start=False)

ผลการทดสอบ

Return 16.88 % ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50 ซึ่งได้ผลที่น้อยกว่า Decision Tree



รูปที่ 4.6 ผลการทดสอบโมเดล Neural Network

4.4.5. Recurrent Neural Network (regression)

ใช้ Recurrent Neural Network ในการทำ Regression ทำนายราคาหุ้นของแต่ละวัน

4.4.5.1 โมเดล Recurrent Neural Network เวอร์ชันที่ 1

ใช้ RNN 2 layer ซึ่งเลเยอร์ที่ 1 และเลเยอร์ที่ 2 มีขนาด 100 node เลเยอร์ที่ 3 เป็น output ขนาด 1 node ใช้ loss function เป็น MSE และใช้ Open, High, Low, Close, Open-Close, High-Low, RSI 7, RSI 14, ADX7, ADX14 ของวันปัจจุบันและย้อนหลัง 5 วันมา insert ในแต่ละ Sample Label ซึ่งใช้วิธี Label Data โดยใช้ Close วันรุ่งขึ้น

ผลการทดสอบ

โมเดลมีค่า MSE ที่สูง เนื่องจากการใส่ค่า timeseries ย้อนหลังให้กับการทำโมเดล regression จะทำให้โมเดลมีข้อผิดพลาด เนื่องจากโมเดลจำเป็นต้องปรับ weight เดิมและ weight ก่อนหน้าเข้าไป มา และ RNN ก็สามารถทำสิ่งนี้ได้อัตโนมัติ

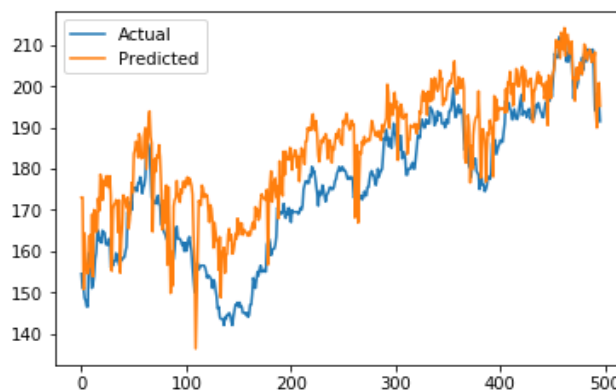
วิธีการแก้ไข

ลบ Features ที่เป็นค่าย้อนหลังทั้งหมดออก ลดจำนวนเลเยอร์ออก เพื่อลดจำนวน parameters ที่จะต้องปรับ

4.4.5.2 โมเดล Recurrent Neural Network เวอร์ชันที่ 2

ขั้นตอนการสร้างโมเดล

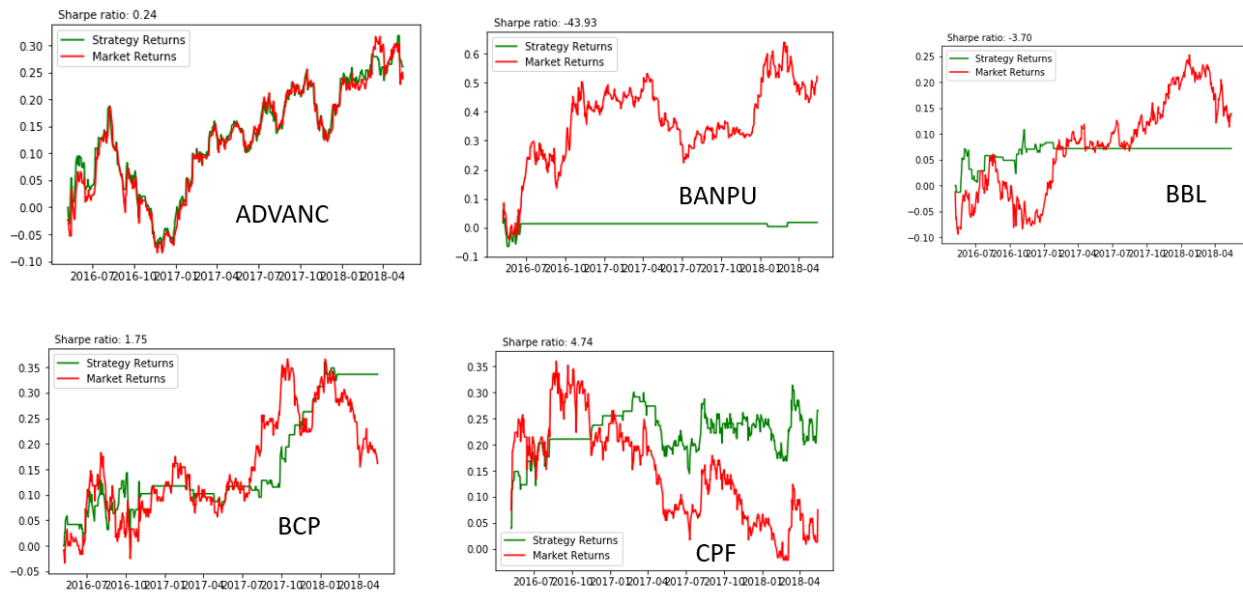
1. ทำข้อมูลทั้งให้เป็น standard โดยลบด้วย mean และ scale ให้เป็น unit variance โดยคิดจาก train dataset
2. Insert Indicator เพิ่มเข้าไปดังนี้ RSI 7 , RSI 14, ADX 7, ADX 14
3. Label Data โดยใช้ราคาวันรุ่งขึ้น
4. Architecture ใช้ RNN 50 node ใน layer แรก และ layer ที่สองเป็น Dense Output ขนาด 1 node
5. ใช้ lose function เป็น MSE
6. Strategy การซื้อขาย คือ ถ้าค่าที่ predict ได้วันนี้มากกว่า close ของวันนี้ให้ทำการซื้อ นอกจากนั้นให้ขาย



รูปที่ 4.7 ตัวอย่างราคาที่ RNN regressor predict ได้กับราคาจริงของหุ้น

ผลการทดสอบ

Return 29.20% ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50



รูปที่ 4.8 ผลการทดสอบโมเดล Recurrent Neural Network

จากรูปที่ 4.8 เป็นผลการทดสอบโมเดล Recurrent Neural Network เวอร์ชันที่ 2 เมื่อกราฟสีแดงคือ Return ของตลาด และกราฟสีเขียวคือ Return ของโมเดล Recurrent Neural Network จะสังเกตได้ว่า โมเดลสามารถทำงานได้ดี โดยให้ Return 29.20% ซึ่งใกล้เคียงกับ Baseline ที่ให้ Return 30.95% อย่างไรก็ตาม โมเดลไม่สามารถทำงานได้กับหุ้นบางตัวอย่าง BANPU

4.4.6. Reinforcement Learning (Q-learning)

ใช้อัลกอริทึม q learning ในการ predict หุ้น โดย implement บน โดยสร้าง environment จำลองที่สร้างขึ้นมาโดยใช้ข้อมูลในปี 2000 - 2016 และหลังจากที่ neural network สามารถทำงานได้ดีแล้วจะทำการทดสอบย้อนในปี 2016-2018

ขั้นตอนการสร้างโมเดล

1. สร้าง Environment ขึ้นมา ซึ่งประกอบไปด้วย Observation, Action และ reward

- Observation เป็น state ส่ง input ให้กับโมเดลรายวันรายตัวโดยประกอบไปด้วย EOD ของหุ้น ย้อนหลัง 5 วัน ซึ่งประกอบไปด้วย Open, High, Low, Close, Adjusted Close, Volume รวมไปถึง Indicator คือ RSI7, RSI14, EMA5, EMA12, EMA26, SMA5, SMA12, SMA26

- Reward ค่าที่ environment ส่งกลับให้กลับโมเดลเพื่อทำการ update policy ในแต่ละครั้งของการเทรด ซึ่ง environment จะ ส่งค่า reward กลับเมื่อผลของกำไรและขาดทุนเมื่อทำการขาย - เท่ากับเปอร์เซ็นต์ของผลกำไรและขาดทุนครั้งนั้น
- Action มี 2 action คือ 1. เลือกที่จะถือหุ้น 2. เลือกที่จะไม่ถือหุ้น

2. Update Q-table ในแต่ละรอบด้วย reward ที่ได้รับ

ผลการทดสอบ

ไม่สามารถใช้งานได้ เนื่องจากไม่สามารถ optimize reward function ได้และใช้เวลานานในการเทรน คณะผู้จัดทำคิดว่าอาจจะต้องปรับ Environment และ Reward ให้เหมาะสมยิ่งขึ้น

4.4.7 Deep Double Q Reinforcement Learning

ใช้อัลกอริธึม q learning พร้อมกับการใช้ Actor-critic เพื่อเพิ่มประสิทธิภาพให้โมเดลสามารถเรียนรู้ได้เร็วขึ้น ในการ predict หุ้น โดย implement บน Neural network โดยสร้าง environment จำลองที่สร้างขึ้นมาโดยใช้ข้อมูลในปี 2000 - 2016 และหลังจากที่ neural network สามารถทำงานได้ดีแล้วจะทำการทดสอบย้อนในปี 2016-2018

4.4.7.1 โมเดล Deep Double Q Reinforcement learning เวอร์ชันที่ 1

สร้าง Environment ขึ้นมา จำลองให้เหมือนการเทรดหุ้นจริงๆ ซึ่งประกอบไปด้วย Observation, Action และ Reward

- Observation เป็น state ส่ง input ให้กับ model รายวันรายตัวโดยประกอบไปด้วย EOD ของหุ้น ย้อนหลัง 15 วัน ซึ่งประกอบไปด้วย Open, High, Low, Close, Adjusted Close, Volume ที่ผ่านการทำให้เป็น standard แล้ว รวมไปถึง Indicator คือ RSI7, RSI14, ADX7 และ ADX14, ราคาหุ้นที่ถืออยู่, output ก่อนหน้า 5 วัน
- Environment จะส่ง reward กลับเมื่อมีการซื้อหุ้นหรือขายหุ้น ซึ่ง $\text{reward} = \text{money} + (\text{เปอร์เซ็นต์ของกำไรหรือขาดทุนที่มีการซื้อขาย} * \text{money})$ โดย money มีค่าเริ่มต้นเท่ากับ 100
- Action มี 2 actions คือ เลือกที่จะถือหุ้น หรือ เลือกที่จะไม่ถือหุ้น

ผลการทดสอบ

ไม่สามารถ optimize reward function ได้

วิธีการแก้ไข

แก้ไขการ return reward ให้มีความเหมาะสมยิ่งขึ้น

4.4.7.2 โมเดล Deep Double Q Reinforcement learning เวอร์ชันที่ 2

สร้าง Environment ขึ้นมา จำลองให้เหมือนการเทรดหุ้นจริงๆ ซึ่งประกอบไปด้วย Observation, Action และ Reward

- Observation เป็น state ส่ง input ให้กับ model รายวันรายตัวโดยประกอบไปด้วย EOD ของหุ้น ย้อนหลัง 15 วัน ซึ่งประกอบไปด้วย Open, High, Low, Close, Adjusted Close, Volume ที่ผ่านการทำให้เป็น standard แล้ว รวมไปถึง Indicator คือ RSI7, RSI14, ADX7 และ ADX14, ราคาหุ้นที่ถืออยู่, output ก่อนหน้า 5 วัน
- Reward ค่าที่ environment ส่งกลับให้กลับโมเดลเพื่อทำการ update policy ในแต่ละครั้งของการเทรด ซึ่ง environment จะ ส่งค่า reward กลับเมื่อ
 - แต่ละวันผ่านไปโดยที่ model เลือกที่จะไม่ถือหุ้นจะได้รับ reward = -0.02
 - ราคาที่ถือวันนี้จะมีราคาสูงขึ้นในวันพรุ่งนี้จะได้ reward เท่ากับเปอร์เซ็นต์ของราคาที่จะสูงขึ้น หาดด้วย 4
 - ผลของกำไรและขาดทุนเมื่อทำการขายจะได้ reward เท่ากับเปอร์เซ็นต์ต่อผลกำไรและขาดทุนครั้งนั้น
- Action มี 2 action คือ เลือกที่จะถือหุ้น หรือ เลือกที่จะไม่ถือหุ้น

โมเดลมี architecture ดังนี้

1. Dense 64 node
2. Dense 32 node
3. Dropout (0.25)
4. Dense 2 node (output node)

ผลการทดสอบ

Return 22.48% ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50

วิธีการแก้ไข

อาจจะใช้ Actor-critic เข้ามาช่วยในการเรียนรู้ของโมเดล เนื่องจากเป็นวิธีที่มักจะถูกนำมาใช้ร่วมกับการใช้ Deep Q reinforcement learning และจะช่วยได้ผลดีขึ้นอยู่เสมอ

4.4.7.3 โมเดล Deep Double Q Reinforcement learning เวอร์ชันที่ 3

ขั้นตอนการสร้างโมเดล

1. สร้าง Environment ขึ้นมา จำลองให้เหมือนการเทรดหุ้นจริงๆ ซึ่งประกอบไปด้วย Observation, Action และ reward

- Observation เป็น state ส่ง input ให้กับ model รายวันรายตัวโดยประกอบไปด้วย EOD ของหุ้น ย้อนหลัง 15 วัน ซึ่งประกอบไปด้วย Open, High, Low, Close, Adjusted Close, Volume ที่ผ่านการทำให้เป็น standard แล้ว รวมไปถึง Indicator คือ RSI7, RSI14, ADX7 และ ADX14, ราคาหุ้นที่ถืออยู่, output ก่อนหน้า 5 วัน
- Reward ค่าที่ environment ส่งกลับให้กลับโมเดลเพื่อทำการ update policy ในแต่ละครั้งของการเทรด ซึ่ง environment จะ ส่งค่า reward กลับเมื่อ
 - แต่ละวันผ่านไปโดยที่ model เลือกที่จะไม่ถือหุ้นจะได้รับ reward = -0.02
 - ราคาที่ถือวันนี้จะมีราคาสูงขึ้นในวันพรุ่งนี้จะได้ reward เท่ากับเปอร์เซ็นต์ของราคาที่จะสูงขึ้น หารด้วย 4
 - ผลของกำไรและขาดทุนเมื่อทำการขายจะได้ reward เท่ากับเปอร์เซ็นต์ของผลกำไรและขาดทุนครั้งนั้น
- Action มี 2 action คือ เลือกที่จะถือหุ้น หรือ เลือกที่จะไม่ถือหุ้น

2. สร้างโมเดล 2 โมเดลคือ คือ โมเดลที่ใช้ในการ predict action และ โมเดลที่ใช้ในการ predict reward ที่เกิดขึ้นของแต่ละ state เพื่อกำหนดว่าโมเดลควรจะไปอยู่ที่ state ใดที่เรียกว่า Actor critic

3. Train network ทั้งสิ้น 1,000 epoch

โมเดลแรกมี architecture ดังนี้

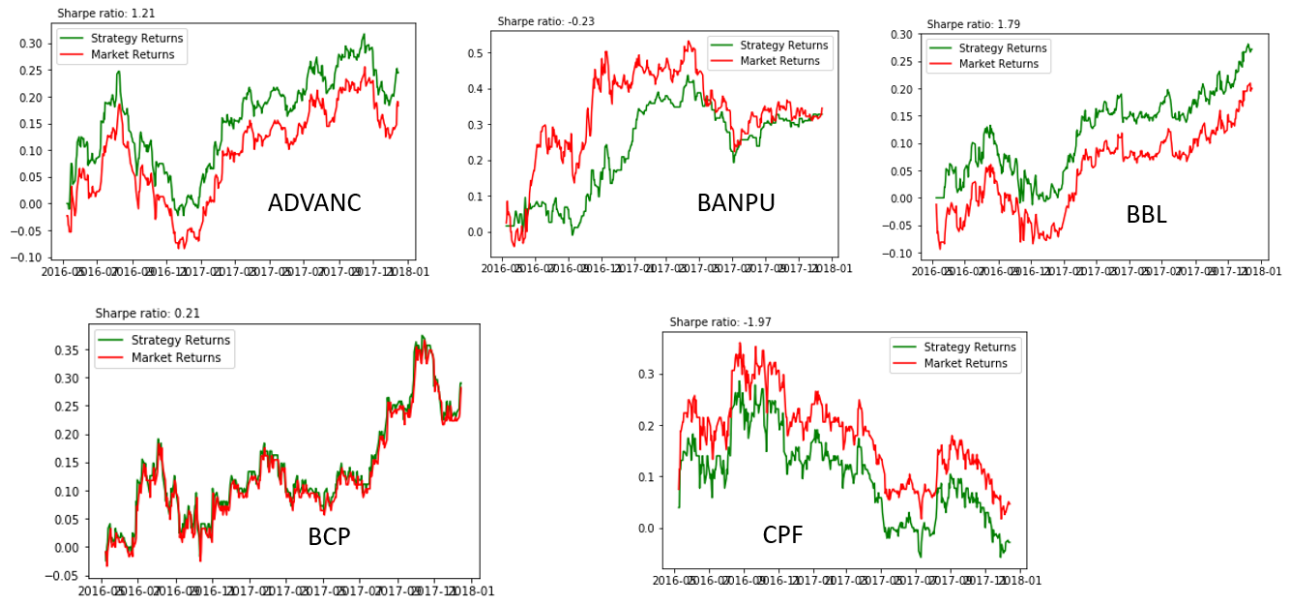
1. Dense 64 node
2. Dense 32 node
3. Dropout (0.25)
4. Dense 2 node (output node)

โมเดลที่สองมี architecture ดังนี้

1. Dense 32 node
2. Dense 16 node
3. Dropout (0.25)
4. Dense 1 node (output node)

ผลการทดสอบ

Return 41.48% ในการทดสอบบน Backtest ย้อนหลัง 2 ปี จากหุ้นทุกตัวใน SET50 ซึ่งมีค่ามากกว่าดัชนีตลาดหุ้นบน SET50 ของปี 2016 - 2018



รูปที่ 4.9 ผลการทดสอบโมเดล Deep Double Q Reinforcement learning

4.4.7.4 วิธีการพัฒนาโมเดล

ปรับการให้ reward ใหม่ ซึ่งทางคณะผู้จัดทำคิดว่าสามารถปรับการให้ reward ได้ดีกว่านี้ รวมไปถึงบางหุ่นโมเดลนี้จะ overfit กับ training dataset จึงเสนอให้อาจจะนำโมเดลนี้เป็นการ label ให้กับการใช้ Decision Tree เพื่อทำนายผล

4.5 สรุปผลการทดลอง

Model	Return (SET50) MAY 2016 - MAY 2018
Buy and hold on SET50 index (Baseline)	30.95%
1. DecisionTreeClassifier	23.43%
2. RandomForestClassifier	21.48%
3. RandomForest ร่วมกับ GaussianMixture	21.47%
4. Neural Network Classifier	16.88%
5. RNN regressor	29.20 %
6. Reinforcement Q learning	-
7. Deep Double Q Reinforcement learning	41.48 %

ตารางที่ 4.1 ตารางสรุปผลทดสอบเปรียบเทียบแต่ละโมเดลรวมถึง SET 50 index

จากผลการทดลอง สามารถสรุปได้ว่าทางคณะผู้จัดทำสามารถสร้างโมเดลที่สามารถชนะ SET 50 index ได้จริงในการทดสอบ Backtest ในปี 2559 - 2560 ซึ่งเป็นไปตามจุดประสงค์ โดยโมเดลที่ดีที่สุดคือ Deep Double Q Reinforcement learning ซึ่งสามารถทำผลตอบแทนได้มากกว่าการ buy and hold ถึง 10.53 % แต่โมเดลอื่นยังไม่สามารถทำผลตอบแทนเทียบเท่าการ Buy and hold ได้

4.6 อภิปรายผลการทดลอง

คณะผู้จัดทำได้ทดลองสร้างโมเดลการเทรดหุ้นจากการใช้เทคโนโลยีการเรียนรู้ของเครื่องบนข้อมูล EOD ของหุ้น ในการสร้างโมเดลการตัดสินใจของปัญญาประดิษฐ์มาทั้งหมด 7 ประเภท ซึ่งในแต่ละประเภทผ่านการทดสอบโดยระบบ Backtest ที่สามารถบอกผลตอบแทนจากการเทรดในอดีตได้ และได้ใช้ค่าจากการทดสอบในการประเมินประสิทธิภาพแต่ละโมเดลที่ถูกสร้างขึ้น โดยผลการทดสอบได้ถูกนำเสนอไปแล้วในหัวข้อ 4.3

จากผลการทดลองข้างต้นสามารถสรุปได้ว่า โมเดลการเทรดหุ้นที่มีประสิทธิภาพสูงสุดคือ Deep Double Q Reinforcement learning ซึ่งสามารถทำกำไรได้ร้อยละ 41.48 สูงกว่าการ buy and hold ถึงร้อยละ 10.47 ในขณะที่โมเดลอื่นนั้น ไม่สามารถทำกำไรสูงกว่าการ buy and hold ได้เลย ทั้งนี้สาเหตุอาจมาจาก ระดับความซับซ้อนของโมเดลที่มีความซับซ้อนสูงสุด ประกอบกับความเป็น Deep Reinforcement ที่สามารถเรียนรู้จากการทดลองเทรดจริงจำนวนหลายรอบ จึงสามารถสร้างผลกำไรจากข้อมูลหุ้นที่มีความไม่แน่นอนสูงได้ จะสังเกตได้ว่า โมเดลที่มีความซับซ้อนเป็นอันดับสองอย่าง RNN Regressor สามารถทำกำไรได้เกือบเทียบเท่า baseline และโมเดลอื่นที่ความซับซ้อนน้อยกว่าก็ทำกำไรได้น้อยลงตามลำดับ

อย่างไรก็ตาม ผลการทดลองยังคงมีประเด็นเรื่องข้อจำกัดต่างๆ อยู่ ขอบเขตของการทดสอบ ณ ตอนนี้ ยังคงเป็นเพียงการเทรดแบบเจาะจงหุ้นบางตัวเท่านั้น โมเดลการตัดสินใจยังไม่สามารถตัดสินใจในส่วนของการเลือกหุ้นที่จะเทรดได้ อีกระยะในการเทรดในระบบทดสอบจึงถูกจำกัดไว้ระดับหนึ่ง และการทดลองนี้เป็นการทดสอบย้อนหลังเท่านั้น จึงไม่สามารถบอกได้ว่าเมื่อนำโมเดลดังกล่าวไปใช้จริงหรือทดสอบจริงในอนาคตจะสามารถทำผลตอบแทนได้มากกว่า SET index อย่างที่ผลการทดสอบบอกได้จริงหรือไม่

ข้อมูลที่มีน้อยเกินไปก็เป็นอีกหนึ่งประเด็นหลัก เนื่องจากข้อมูลที่สามารถหามาได้ยังคงมี Feature ที่น้อย คือ มีเพียง EOD เท่านั้น จึงทำให้ข้อมูลชุดนี้อาจไม่มีความเหมาะสมต่อโมเดลบางตัวอย่าง Neural Network ที่จะทำงานได้ดีมากขึ้นหากจำนวน Feature มีมาก

บทที่ 5

บทสรุป

5.1 ผลสำเร็จของโครงการ

ฟังก์ชัน	สถานะการดำเนินงาน
หาข้อมูล EOD ของหุ้น SET100 ย้อนหลังถึงปี 2000	เสร็จแล้ว
Data cleansing	เสร็จแล้ว
ทดลองโมเดลตระกูล Neural network	เสร็จแล้ว
ทดลองโมเดลตระกูล Decision Tree	เสร็จแล้ว
ทดลองโมเดลตระกูล Reinforcement	เสร็จแล้ว
สร้างโมเดลที่ให้ผลตอบแทนชนะตลาด	เสร็จแล้ว

ตารางที่ 5.1 ตารางแสดงสถานะความสำเร็จของงานในแต่ละส่วน

จากการทดสอบโมเดลหลากหลายประเภท พบว่าโมเดลที่มีประสิทธิภาพสูงสุดคือ Deep Double Q Reinforcement learning ที่สามารถทำกำไรสูงกว่าการ buy and hold ปกติได้ถึงร้อยละ 10.47 ทั้งนี้การทดลองยังมีข้อจำกัดด้านระบบ Backtest และข้อมูลในการเทรนโมเดล

5.2 ปัญหาที่พบในการทำงาน

1. ปัญหาความล่าช้าของการได้มาซึ่งข้อมูลที่มีคุณภาพ

การได้มาซึ่งข้อมูลหุ้นที่มีแหล่งผู้ให้บริการแบบไม่มีค่าใช้จ่ายนั้นสามารถทำได้ง่ายตาย เช่น ข้อมูล EOD จาก Yahoo Finance ที่สามารถดาวน์โหลดได้ผ่านอินเทอร์เน็ต แต่ข้อมูลจาก Yahoo Finance ก็ให้ข้อมูลหุ้นไทยได้เพียงแค่อีโอดีของหุ้นหลังปี ค.ศ. 2000 เท่านั้น และไม่สามารถให้ข้อมูลอื่นๆ นอกจาก EOD ได้

การที่จะได้มาซึ่งข้อมูลที่กว้างขวางนอกจาก EOD นั้นส่วนมากต้องมีค่าใช้จ่าย เช่น การซื้อข้อมูลจาก Settrade Reuter Bisnews ซึ่งข้อมูลส่วนนี้มีทั้งในส่วนของ Fact Sheet งบการเงิน Bid/Offer ข่าวสำคัญต่างๆ และอื่นๆ ที่ไม่สามารถหาได้จากช่องทางที่ไม่เสียค่าใช้จ่าย

ข้อมูลในส่วนแรกนั้น ผู้จัดทำสามารถหาได้อย่างไม่มีปัญหา ส่วนข้อมูลที่มีค่าใช้จ่ายนั้น ทางผู้จัดทำได้รับการสนับสนุนจากสมาคมฟินเทคแห่งประเทศไทยที่จะคอยจัดหาข้อมูลเหล่านี้ให้เพื่อการศึกษา ปัญหาที่เกิดขึ้นก็คือการจะได้มาซึ่งข้อมูลจากทางสมาคมนั้นต้องมีขั้นตอนที่ใช้เวลา หรือข้อมูลบางตัวนั้นเปิดเผยได้ยาก จึงทำให้เกิดความล่าช้าในการนำข้อมูลในส่วนนี้มาใช้ทำโครงการ

2. ปัญหาแหล่งอ้างอิงเกี่ยวกับบทเทรตหุ้น Machine learning มีน้อย

ปัญหานี้เกิดตอนที่ผู้จัดทำต้องการหาข้อมูลและตัวอย่างของผลงานด้านหุ้นกับ machine learning ซึ่งงานที่ทำเฉพาะด้านใดด้านหนึ่งนั้นสามารถหาได้ง่าย แต่งานที่ผสมผสานทั้ง 2 อย่างเข้าด้วยกันนั้นมีน้อย นอกจากนี้ เรื่องชิ้นงานด้านการเทรตหุ้นนั้นถือเป็นความลับโดยส่วนใหญ่ เพราะการเผยแพร่ข้อมูลการทำงานทั้งหมดถือเป็นการทำร้ายผลิตภัณฑ์ตนเอง การเข้าถึงข้อมูลเหล่านี้จึงได้เพียงผิวเผิน หรืออาจจะไม่สามารถหาข้อมูลบางส่วนเจอ

3. ปัญหาผู้ทำมีประสบการณ์ด้านหุ้นไม่เพียงพอ

ปัญหานี้เกิดจากการที่ผู้จัดทำไม่มีประสบการณ์ในการลงทุนในหุ้นมากนักมาก่อนจะทำโครงการ ผู้จัดทำเคยลงทุนในช่องทางบางช่องทาง แต่ไม่ได้จริงจังและมีประสบการณ์ขนาดเข้าใจและเทรตหุ้นเป็น ถึงแม้การทำบทเทรตหุ้นจะไม่ได้อาศัยความเข้าใจด้านหุ้นขนาดมืออาชีพ แต่ก็ต้องใช้ความรู้และประสบการณ์ระดับหนึ่ง ผู้จัดทำจึงเร่งศึกษาและหาประสบการณ์ การทำโครงการจึงต้องใช้เวลาไปกับการศึกษาเพิ่มเติมและปรึกษากันอยู่บ่อยครั้ง

4. ปัญหาความผิดพลาดในโปรแกรมทดสอบประสิทธิภาพ

ระบบทดสอบประสิทธิภาพหรือที่เรียกว่า Backtest เป็นระบบที่ใช้ทดสอบผลได้จากการเทรตหุ้นโดยการใช้อัลกอริธึมการเทรตแบบต่างๆ ที่ได้คิดค้นขึ้น ระบบ Backtest ได้ถูกพัฒนาขึ้นเป็นอันดับแรกเพื่อใช้วัดประสิทธิภาพของโมเดลที่จะถูกสร้างขึ้นในเวลาต่อมา และคณะผู้จัดทำก็ได้ใช้ระบบตัวนี้วัดผลกำไร ความแม่นยำ และค่าอื่นๆ มาโดยตลอด อย่างไรก็ตาม คณะผู้จัดทำได้พบข้อผิดพลาดในตัวโปรแกรม Backtest นี้ที่มีผลทำให้การเทรตหุ้นตามข้อมูลในอดีตไม่เป็นไปตามความเป็นจริง หมายความว่า ระบบเทรตหุ้นบางตัวที่เคยทดสอบมา อาจไม่ได้ถูกทดสอบอย่างถูกต้อง ความผิดพลาดครั้งนี้ส่งผลให้การพัฒนาหยุดชะงักเป็นช่วงระยะเวลาหนึ่ง จึงทำให้การทำงานล่าช้ายิ่งขึ้นไปอีก

บรรณานุกรม

1. การเรียนรู้เชิงลึก [ออนไลน์], แหล่งอ้างอิง: <https://th.wikipedia.org/wiki/การเรียนรู้เชิงลึก>
[29 กันยายน 2560]
2. ศัพท์เฉพาะและคำจำกัดความของ Dataset [ออนไลน์], แหล่งอ้างอิง:
<https://khasathan.in.th/archives/1197> [29 กันยายน 2560]
3. ศุภวัฒน์ สุภักดวงศ์. 2560. *Think ALGO เทรดหุ้นยุคใหม่ ให้โรบอททำเงินแทน*. กรุงเทพฯ: stock2morrow.
4. เสริมโชค ไชยเลิศ. 2558. *เทวอลามเขย่าหุ้น Season 1*. กรุงเทพฯ: สำนักพิมพ์ เกรท ไอเดีย.
5. Arpan Chakraborty and Tucker Balch (2015), Machine Learning for Trading by Georgia Tech [ออนไลน์], แหล่งอ้างอิง: <https://www.udacity.com/course/machine-learning-for-trading--ud501> [27 สิงหาคม 2560]
6. Eakasit Pacharawongsakda (2015), Evaluation metrics [ออนไลน์], แหล่งอ้างอิง:
<https://www.slideshare.net/sitake/evaluation-metrics-precision-recall-fmeasure-roc> [10 กันยายน 2560]
7. Katie Malone and Sebastian Thrun, Intro to Machine Learning [ออนไลน์], แหล่งอ้างอิง:
<https://www.udacity.com/course/intro-to-machine-learning--ud120>
[21 สิงหาคม 2560]
8. Napong Wanichayapong (2010), Confusion Matrix [ออนไลน์], แหล่งอ้างอิง:
<https://plagad.wordpress.com/2010/08/26/confusion-matrix/> [10 กันยายน 2560]
9. Neural network [ออนไลน์], แหล่งอ้างอิง: <https://www.gotoknow.org/posts/163433>
[29 กันยายน 2560]

10. Reinforcement Learning [ออนไลน์], แหล่งอ้างอิง: <https://medium.com/o-v-e-r-f-i-t-t-e-d/เริ่มต้นกับ-reinforcement-learning-และตัวอย่างพื้นฐาน-b6a9a1167820>
[29 กันยายน 2560]