การศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่มีต่อมูลค่าของอีเธอเรียม

EFFECT OF ETHEREUM WHALE WALLET TO ETHEREUM PRICING

ปุณพัฒน์กฤช ลิมวรนุสรณ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

EFFECT OF ETHEREUM WHALE WALLET TO ETHEREUM PRICING

PUNNAPATKIT LIMWORRANUSORN

A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

หัวข้อปัญหาพิเศษ การศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่มีต่อมูลค่าอีเธอเรียม
EFFECT OF ETHEREUM WHALE WALLET TO ETHEREUM PRICING

ชื่อนักศึกษา นายปุณพัฒน์กฤช ลิมวรนุสรณ์ 58050332

ปริญญา วิทยาศาสตรบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ดร.อัคเดช อุดมชัยพร

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2561

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.ศรัณย์ อินทโกสุม **ประธานกรรมการ**	
ดร.กุลสวัสดิ์ จิตขจรวานิช **กรรมการ**	
ดร.อัคเดช อุดมชัยพร **กรรมการและอาจารย์ที่ปรึกษา**	

หัวข้อปัญหาพิเศษ การศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่มีต่อมูลค่าอีเธอเรียม
ชื่อนักศึกษา นายปุณพัฒน์กฤช ลิมวรนุสรณ์ รหัสนักศึกษา 58050332
ปริญญา วิทยาศาสตรบัณฑิต
ภาควิชา วิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา 2561
อาจารย์ที่ปรึกษา ดร.อัคเดช อุดมชัยพร

บทคัดย่อ

แนวโน้มการเปลี่ยนแปลงมูลค่าของสกุลเงินอีเธอเรียมนั้นมีปัจจัยเกี่ยวข้องที่หลากหลายและมีความผันผวน จึงเป็นไปได้ยากที่จะทำนายทิศทางการขึ้นลงของมูลค่าอีเธอเรียม ซึ่งในปัจจุบันยังไม่มีอัลกอริทึมใดที่สามารถทำนายแนวโน้มราคาของอีเธอเรียมได้อย่างแม่นยำ ปัญหาพิเศษนี้จึงได้นำเสนอการศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่มีผลต่อมูลค่าอีเธอเรียม โดยใช้เทคนิคการทำเหมืองข้อมูล เพื่อทำนายแนวโน้มของมูลค่าอีเธอเรียมและหาความสัมพันธ์ของผู้ถืออีเธอเรียมรายใหญ่ที่มีต่อมูลค่าอีเธอเรียม การทดลองในงานวิจัยนี้ใช้ชุดข้อมูลจาก Google BigQuery และข้อมูลจากเว็บไซต์ Crytocompare ซึ่งดึงข้อมูลจากบล็อคเชน (Blockchain) เพื่อทำการทดลองและเปรียบเทียบประสิทธิภาพในแต่ละเทคนิคของการทำเหมืองข้อมูลที่ใช้ดังกล่าว จากการทดลองพบว่าข้อมูลที่ได้นำมาใช้ทำนายแนวโน้มมูลค่าอีเธอเรียมและการหากฎความสัมพันธ์นั้นยังไม่สามารถทำนายและหากฎความสัมพันธ์ได้ดีเท่าที่ควร โดยคาดว่ายังจำเป็นต้องใช้คุณลักษณะอื่น ๆ มาใช้ประกอบการวิเคราะห์ รวมถึงการใช้อัลกอริทึมหรือเทคนิคที่มีประสิทธิภาพสูงกว่าในการทำนายแนวโน้มมูลค่าและหากฎความสัมพันธ์ของอีเธอเรียม

คำสำคัญ : การหากฎความสัมพันธ์, การทำนายแนวโน้มมูลค่าอีเธอเรียม

Title	EFFECT OF ETHEREUM WHALE WALLET TO ETHEREUM PRICING
Students	Mr.Punnapatkit Limworranusorn Student ID 58050332
Degree	Bachelor of Science
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang
Academic Year	2018
Advisor	Dr. Akadej Udomchaiporn

Abstract

Ethereum is a popular cryptocurrency but its exchange rate is fluctuated depending on many internal and external factors. Therefore, it's not easy to predict the trends of Ethereum exchange rate and there is no well-known existing algorithm that is able to predict that trends. This special problem thus proposes a study of relationships between Ethereum exchange rate and the top shareholders using data mining techniques. The project consists of two phases: (1) Data acquisition from Blockchain and (2) Data mining and visualization. The outcomes of the study are shown in a web application and can suggest some relationships between Ethereum exchange rate and the top shareholders

Keywords : association rule mining, ethereum value predictive

กิตติกรรมประกาศ

การจัดทำปัญหาพิเศษเรื่องการศึกษาผลกระทบของผู้อีเธอเรียมรายใหญ่ที่มีผลต่อมูลค่าอีเธอเรียมนี้สำเร็จลุล่วงไปได้ด้วยดีคณะผู้จัดทำต้องขอขอบพระคุณอาจารย์ที่ปรึกษา ดร. อัคเดช อุดมชัยพร ที่ช่วยให้คำปรึกษาและคำแนะนำที่ดีแก่คณะผู้จัดทำในการปรับปรุงปัญหาพิเศษนี้

ขอขอบพระคุณอาจารย์ผู้ควบคุมการสอบปัญหาพิเศษ ผศ.ดร. ศรัณย์ อินทโกสุม และ ดร.กุลสวัสดิ์ จิตขจรวานิช ที่มีส่วนช่วยในการตรวจสอบ และให้คำแนะนำ ทำให้ปัญหาพิเศษนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณบุคคลต่าง ๆ ที่เกี่ยวข้องที่ได้ให้การช่วยเหลือในการทำปัญหาพิเศษนี้ และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้มอบโอกาสให้ได้เข้ารับการศึกษาในสถาบันแห่งนี้

ปุณพัฒน์กฤช ลิมวรนุสรณ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญ 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตของงานวิจัย 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2
1.5 ขั้นตอนการดำเนินงาน 2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3
2.1 บล็อคเชน (Blockchain) 3
2.2 อีเธอเรียม (Ethereum) 4
2.3 การทำเหมืองข้อมูล (Data Mining) 5
2.3.1 การหากฎความสัมพันธ์ (Association Rules Mining) 5
2.3.2 การจำแนกข้อมูล (Data Classification) 6
2.3.3 การประเมินผลการทำเหมืองข้อมูล 7
2.4 บริการฐานข้อมูล Google BigQuery 9
บทที่ 3 วิธีดำเนินการวิจัย 10
3.1 ระเบียบวิธีวิจัย 10
3.1.1 ศึกษากระบวนการหาความสัมพันธ์ของข้อมูลในรูปแบบต่าง ๆ 11
3.1.2 ทดลองหาความสัมพันธ์ของข้อมูลตามวิธีการต่างๆ 11
3.1.3 ทดลองปรับค่าพารามิเตอร์ต่างๆที่ใช้ในการทดลอง 11
3.1.4 ประเมินผลการทดลองและทดสอบทางสถิติ 11
3.1.5 วิเคราะห์และสรุปผลการวิจัย 11
3.2 ขั้นตอนวิธีการทดลองการหาความสัมพันธ์ของข้อมูล 11

สารบัญ(ต่อ)

หน้า

3.2.1 การนำเข้าข้อมูล 13
3.2.1.1 การนำเข้าฐานข้อมูลโดยใช้ Google BigQuery 13
3.2.1.2 การนำข้อมูลเข้าฐานข้อมูลโดยใช้โปรแกรมไพธอน 17
3.2.2 การเตรียมข้อมูล (Preprocessing) 22
3.2.2.1 Data Integration 22
3.2.2.2 Data Cleaning 22
3.2.2.3 Data Transformation 23
3.2.2.4 Data Reduction 26
3.2.3 การวิเคราะห์ความสัมพันธ์ของข้อมูล 28
3.2.3.1 ขั้นตอนวิธีการทดลองโดยใช้วิธีการจำแนกกลุ่มข้อมูล 28
3.2.3.2 ขั้นตอนวิธีการทดลองโดยใช้วิธีการวิเคราะห์การถดถอยของข้อมูล 29
3.2.3.3 ขั้นตอนวิธีการทดลองโดยใช้การหากฎความสัมพันธ์ของข้อมูล 38
3.2.3.4 ขั้นตอนวิธีการทดลองโดยใช้การวิเคราะห์การถดถอยเชิงเส้นด้วย Scikit-learn Framework 39
3.2.3.5 ขั้นตอนวิธีการทดลองโดยใช้การหากฎความสัมพันธ์ของข้อมูลส่วนของข้อมูลนำเข้า 42
3.2.3.6 ขั้นตอนวิธีการทดลองโดยใช้การหากฎความสัมพันธ์ของข้อมูลด้วย Rapidminer Studio 43
3.2.4 การประเมินผล 46
3.2.4.1 การประเมินผลโมเดลสำหรับปัญหาการจำแนกประเภทข้อมูล 46
3.2.4.2 การประเมินผลโมเดลสำหรับปัญหาการวิเคราะห์การถดถอย 47
3.2.5 การปรับค่าพารามิเตอร์ต่าง ๆ 48
บทที่ 4 ผลการดำเนินงานและการอภิปราย 49
4.1 ผลการดำเนินงาน 49
4.1.1 ใช้การวิเคราะห์การถดถอยในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework 49

สารบัญ(ต่อ)

หน้า

4.1.2 ใช้ต้นไม้ตัดสินใตในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework.....53
4.1.3 ใช้ต้นไม้ตัดสินใจในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio.....54
4.1.4 ใช้วิธีเนอีฟเบย์ในการทดลองกับชุดข้อมูลด้วย Scikit-learn Framework.....54
4.1.5 ใช้วิธีเนอีฟเบย์ในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio.....56
4.1.6 ใช้การเรียนรู้เชิงลึกในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio.....56
4.1.7 ใช้การหากฎความสัมพันธ์ในการหาความสัมพันธ์ของข้อมูลด้วย Rapidminer Studio 57
4.2 การอภิปรายผลการดำเนินงาน.....60
4.2.1 การเปรียบเทียบค่าความแม่นยำที่ทดลองด้วย Scikit-learn Framework.....60
4.2.2 การเปรียบเทียบค่าความแม่นยำที่ทดลองด้วย Rapidminer Studio.....60
4.3 ปัญหาที่พบในการดำเนินงาน.....61
4.3.1 ข้อจำกัดทางทรัพยากร.....61
4.3.2 การประมวลผลข้อมูลประเภทข้อความ.....61
4.3.3 ชุดข้อมูลที่นำมาทดลองเป็นคนละประเภท.....61
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....63
5.1 สรุปผลการดำเนินงาน.....63
5.2 ข้อเสนอแนะ.....64
บรรณานุกรม.....65

สารบัญตาราง

ตารางที่ หน้า

3.1 ตารางข้อมูลยอดเงินในบัญชี (balance) 26
3.2 ตารางข้อมูลราคาของค่าเงินในแต่ละวัน (price_hostory_day) 27
3.3 ตารางข้อมูลธุรกรรมการซื้อขาย (transaction) 28
4.1 ผลการทดลองของการวิเคราะห์การถดถอย 51
4.2 ค่าความแม่นยำของวิธีต้นไม้ตัดสินใจด้วย Scikit-learn Framework 53
4.3 ผลการทดลองโดยใช้ต้นไม้ตัดสินใจด้วย Rapidminer Studio 54
4.4 ผลการทดลองโดยใช้วิธีเนอีฟเบย์ด้วย Scikit-learn Framework 55
4.5 ผลการทดลองโดยใช้วิธีเนอีฟเบย์ด้วย Rapidminer Studio 56
4.6 ผลการทดลองโดยใช้วิธีการเรียนรู้เชิงลึกด้วย Rapidminer Studio 57
4.7 ผลการทดลองหากฎความสัมพันธ์ของข้อมูลด้วย Rapidminer Studio 58
4.8 การเปรียบเทียบค่าความแม่นยำของวิธีการต่างๆจากการทดลองด้วย Scikit-learn Framework 60
4.9 การเปรียบเทียบค่าความแม่นยำของวิธีการต่างๆจากการทดลองด้วย Rapidminer Studio 61

สารบัญรูป

รูปที่ หน้า

2.1 ตัวอย่างข้อมูลในบล็อคเริ่มต้น....3
2.2 ตัวอย่างข้อมูลในบล็อคเชน....4
2.3 กระบวนการทำงานของอีเธอเรียม....4
2.4 ขั้นตอนวิธีของการทำเหมืองข้อมูล....5
2.5 การเรียนรู้จากข้อมูลเพื่อสร้างโมเดลในการจำแนกข้อมูล....7
2.6 ตารางเมทริกซ์ความสับสน....8
3.1 แผนภาพแสดงระเบียบวิธีวิจัยการศึกษาผลกระทบของผู้ถืออีเธอเรียมที่มีต่อมูลค่าของอีเธอเรียม ...10
3.2 แผนภาพแสดงขั้นตอนวิธีทดลองหาความสัมพันธ์ของข้อมูล....12
3.3 ตัวอย่างการ Query ข้อมูลจากฐานข้อมูล Ethereum_blockchain ผ่าน Google BigQuery....13
3.4 การเลือกบันทึกผลลัพธ์จากการ Query....14
3.5 บันทึกผลลัพธ์ที่ได้จากการ Query ไปยังชุดข้อมูลที่เลือกไว้....14
3.6 นำข้อมูลออกมาเป็นไฟล์ csv....15
3.7 การเชื่อมต่อ MySQL กับโปรแกรม DataGrip....15
3.8 การนำไฟล์เข้าไปยังฐานข้อมูล....16
3.9 ผลลัพธ์จากการนำข้อมูลเข้าฐานข้อมูล....16
3.10 การเชื่อมต่อภาษาไพธอนกับฐานข้อมูล....17
3.11 แสดงถึงการนำรายชื่อผู้ถืออีเธอเรียมจากเว็บไซต์มาใช้....18
3.12 การอ่านไฟล์ csv มาใช้....18
3.13 การนำข้อมูลเข้าฐานข้อมูล....19
3.14 ผลลัพธ์จากการนำข้อมูลเข้าฐานข้อมูล....19
3.15 การเรียกข้อมูลที่ต้องการผ่าน API ของเว็บไซต์ CryptoCompare....20
3.16 เตรียมข้อมูลเพื่อนำเข้าฐานข้อมูล MySQL....21
3.17 การนำข้อมูลเข้าฐานข้อมูล MySQL....21
3.18 ฟังก์ชันปรับค่าของข้อมูล....21
3.19 คำสั่งตรวจสอบข้อมูลสูญหาย....23
3.20 คำสั่งลบแถวข้อมูลขาดหาย....23
3.21 การแบ่งช่วงข้อมูล....24
3.22 ฟังก์ชันแปลงข้อมูลเป็นตัวเลข....25

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.23 ข้อมูลที่เป็นข้อความและข้อมูลตัวเลข....25
3.24 ฟังก์ชันแปลงข้อมูลเป็นวันฮ็อต....26
3.25 ตัวอย่างข้อมูลในแต่ละคุณลักษณะ....29
3.26 คำสั่งใช้งานฟังก์ชันการเตรียมข้อมูล....30
3.27 คำสั่งเรียกใช้งาน Scikit-learn Framework....30
3.28 คำสั่งแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ....31
3.29 คำสั่งสร้างต้นไม้ตัดสินใจสำหรับค่า criterion เป็น entropy....31
3.30 คำสั่งเรียกใช้งานการทดสอบความแม่นยำของต้นไม้ตัดสินใจและผลลัพธ์การทดสอบ....32
3.31 โครงสร้างการประมวลผลข้อมูล....32
3.32 โครงสร้างการฝึกสอนและทดสอบข้อมูล....33
3.33 ผลลัพธ์การใช้ต้นไม้ตัดสินใจ....33
3.34 คำสั่งเรียกใช้งานโมเดล....34
3.35 คำสั่งแบ่งข้อมูลไปยังตัวแปร....34
3.37 คำสั่งสร้างโมเดลเนอีฟเบย์....35
3.38 คำสั่งแสดงผลลัพธ์ของโมเดล....35
3.39 โครงสร้างการประมวลผลข้อมูล....36
3.40 โครงสร้างการฝึกสอนและการทดสอบ....36
3.41 ผลลัพธ์การใช้เนอีฟเบย์....37
3.42 โครงสร้างการฝึกสอนและทดสอบโดยใช้การเรียนรู้เชิงลึก....37
3.43 ผลลัพธ์การใช้การเรียนรู้เชิงลึก....38
3.44 ตัวอย่างข้อมูลในแต่ละคุณลักษณะ....38
3.45 ตัวอย่างคำสั่งเรียกใช้ Scikit-learn Framework....39
3.46 คำสั่งเลือกคุณลักษณะไปยังตัวแปร....40
3.47 คำสั่งเรียกใช้การถดถอยเชิงเส้นและการสร้าง....40
3.48 คำสั่งแสดงผลลัพธ์ของโมเดล....41
3.49 แสดงผลลัพธ์ของโมเดล....41
3.50 คำสั่งคำนวณค่าสหสัมพันธ์....41

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.51 ตัวอย่างตารางสหสัมพันธ์ 42
3.52 ตัวอย่างชุดข้อมูล 43
3.53 คำสั่งแสดงรายละเอียดข้อมูล 44
3.54 คำสั่งแบ่งช่วงข้อมูล 44
3.55 การแปลงข้อมูลเป็นวันฮ็อต 45
3.56 โครงสร้างการประมวลผลหากฎความสัมพันธ์ 45
3.57 กฎความสัมพันธ์ 46
4.1 กราฟเส้นตรงของโมเดลที่ 1 51
4.2 กราฟเส้นตรงของโมเดลที่ 2 52
4.3 กราฟเส้นตรงของโมเดลที่ 3 52
4.4 ต้นไม้ตัดสินใจ 54

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ระบบการเงินในอดีต ธนาคารมีบทบาทสำคัญอย่างมากในการเป็นศูนย์กลางในการทำธุรกรรมทางการเงินเช่น การฝากเงินกับธนาคาร การโอนเงินและการรับเงินผ่านธนาคารซึ่งปัจจุบันผู้ใช้บริการที่ใช้เงินสดลดน้อยลง ทำให้การทำธุรกรรมการเงินออนไลน์มีมากยิ่งขึ้น และความเสี่ยงจากการถูกเจาะข้อมูลทางการเงินก็มากขึ้นด้วยเช่นกัน จึงทำให้เกิดการคิดค้นเทคโนโลยีบล็อคเชนขึ้นมา บล็อคเชนเป็นระบบบันทึกธุรกรรมออนไลน์ที่มีลักษณะเป็นเครือข่าย ที่สามารถเก็บข้อมูลการทำธุรกรรมทางการเงินและสินทรัพย์อื่น ๆ ได้โดยไม่มีตัวกลางคือ สถาบันการเงิน ซึ่งด้วยระบบของบล็อคเชนก่อให้เกิดสกุลเงินดิจิตอลต่าง ๆ เช่น บิตคอยน์ (Bitcoin) หรืออีเธอเรียม (Ethereum) เป็นต้น

สกุลเงินอิเล็กทรอนิกส์ที่เกิดจากอีเธอเรียม มีหลากหลายเช่น BNB, VeChain (VEN) หรือ Maker (MKR) เป็นต้น ซึ่งมูลค่าของสกุลเงินอิเล็กทรอนิกส์แต่ละสกุลเงินมีมูลค่าที่แตกต่างกันและผู้ถือสกุลเงินอิเล็กทรอนิกส์ในแต่ละสกุลเงินก็มีจำนวนที่แตกต่างเช่นกัน อย่างไรก็ตามการลงทุนในสกุลเงินอิเล็กทรอนิกส์นั้นย่อมมีความเสี่ยงเช่นเดียวกับการลงทุนหุ้นหรือการลงทุนประเภทอื่น ๆ ซึ่งมูลค่าของสกุลเงินอิเล็กทรอนิกส์แต่ละสกุลเงินมีความผันผวนขึ้นลงของราคา ปัญหาด้านความเสี่ยงนี้เป็นสิ่งที่นักลงทุนต้องพิจารณาอย่างถี่ถ้วน หากนักลงทุนวิเคราะห์ความเสี่ยงดังกล่าวได้ไม่ละเอียดเท่าที่ควร อาจทำให้นักลงทุนต้องเผชิญกับสภาวะการขาดทุนอย่างต่อเนื่องได้ ประกอบกับข้อมูลที่มีขนาดใหญ่มากในตลาดสกุลเงินของอีเธอเรียม ทำให้การวิเคราะห์ความเสี่ยงอาจไม่ครอบคลุม และมีความคลาดเคลื่อนได้ โดยในปัญหาพิเศษนี้นำเทคโนโลยีทางการทำเหมืองข้อมูลมาใช้ในการวิเคราะห์ข้อมูลการขึ้นหรือลงของมูลค่าเงินอิเล็กทรอนิกส์อีเธอเรียมเพื่อนำไปใช้ในการประกอบการตัดสินใจของผู้ลงทุนโดยใช้การจำแนกกลุ่มของข้อมูล (Classification) และการจัดกลุ่มของข้อมูล (Clustering) เพื่อศึกษาตัวแปรที่มีผลต่อความผันผวนของมูลค่าของสกุลเงินอิเล็กทรอนิกส์ของอีเธอเรียม

1.2 วัตถุประสงค์ของงานวิจัย

ศึกษาและทดลองวิธีการต่าง ๆ ในการทำนายมูลค่าของเงินดิจิตอลสกุลอีเธอเรียมโดยการพิจารณาความสัมพันธ์ของผู้ถืออีเธอเรียมรายใหญ่ 1000 อันดับแรก กับมูลค่าของอีเธอเรียมและแสดงผลความสัมพันธ์ในรูปแบบของกราฟได้

1.3 ขอบเขตของงานวิจัย

1. วิเคราะห์แนวโน้มมูลค่าของอีเธอเรียมได้โดยพิจารณจากผู้ถืออีเธอเรียมรายใหญ่ 1000 อันดับแรก
2. หาความสัมพันธ์ของผู้ถืออีเธอเรียมรายใหญ่ 1000 อันดับแรกและมูลค่าของอีเธอเรียม
3. ข้อมูลที่นำมาวิเคราะห์เป็นข้อมูลของอีเธอเรียมในช่วงปี 2016-2018 เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ทราบถึงประสิทธิภาพและประสิทธิผลของวิธีการวิเคราะห์ข้อมูลแบบต่างๆ
2. ได้ทราบองค์ความรู้ที่ได้มาจากการศึกษาและวิเคราะห์ผลลัพธ์ที่ได้
3. ได้ทราบความสัมพันธ์ของผู้ถืออีเธอเรียมรายใหญ่ 1000 อันดับแรก กับมูลค่าของอีเธอเรียมในช่วงปี 2016-2018

1.5 ขั้นตอนในการดำเนินงาน

1. ศึกษาการทำงานของวิธีการจำแนกข้อมูลและการหากฎความสัมพันธ์ของข้อมูล
2. กำหนดและรวบรวมชุดข้อมูลที่จำเป็นในการทดลอง
3. ออกแบบโครงสร้างและขั้นตอนการเตรียมข้อมูล
4. เริ่มทำการทดลองและทดสอบโมเดล
5. ประเมินผลและปรับปรุงผลการทดลอง
6. วิเคราะห์และสรุปผลการทดลอง
7. จัดทำเอกสารประกอบโครงงานปัญหาพิเศษ

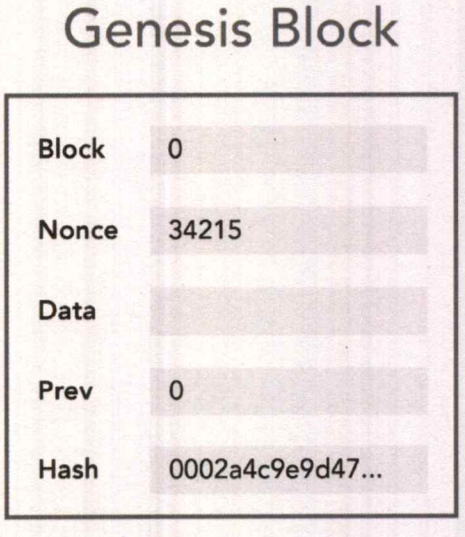
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง ซึ่งประกอบไปด้วย บล็อคเชน อีเธอเรียม การทำเหมืองข้อมูล เช่น ขั้นตอนการหากฎความสัมพันธ์ ขั้นตอนการจำแนกข้อมูล ขั้นตอนการจัดกลุ่มข้อมูล การประเมินผลการทำเหมืองข้อมูล และเว็บแอปพลิเคชัน โดยจะแสดงรายละเอียดในหัวข้อที่ 2.1 ถึง 2.4

2.1 บล็อคเชน (Blockchain)

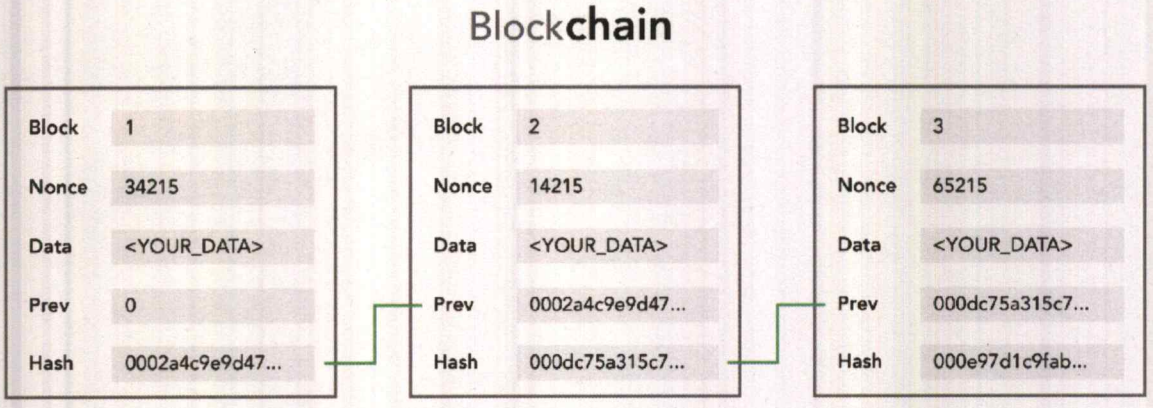
บล็อคเชน คือฐานข้อมูล (Database) ประเภทหนึ่งที่เก็บไว้ในเครื่องของผู้ใช้ในระบบ โดยไม่ใช้ตัวกลางในการดูแลระบบ การทำงานของบล็อคเชนในขั้นแรกจะต้องมีบล็อคเริ่มเต้น (Genesis block) เกิดขึ้นเสมอ ซึ่งเป็นบล็อค (Block) แรกที่เกิดขึ้นบนสายบล็อคเชนโดยจะถูกสร้างขึ้นเพื่อนำไปอ้างอิงในบล็อคต่อ ๆ ไป



รูปที่ 2.1 ตัวอย่างข้อมูลในบล็อคเริ่มต้น

หลังจากบล็อคเริ่มต้นถูกสร้างขึ้นมาในระบบแล้ว เมื่อเวลาผ่านไประยะหนึ่งบล็อคเริ่มต้นจะถูกบล็อคที่ถูกสร้างใหม่มาต่อโดยบล็อคที่ถูกสร้างใหม่เกิดจากทรานเซคชัน (Transaction) ที่อยู่บนระบบบล็อคเชน เช่น บิตคอยน์ทรานเซคชัน (Bitcoin transaction) คือข้อมูลการโอนสกุลเงินของบิตคอยน์

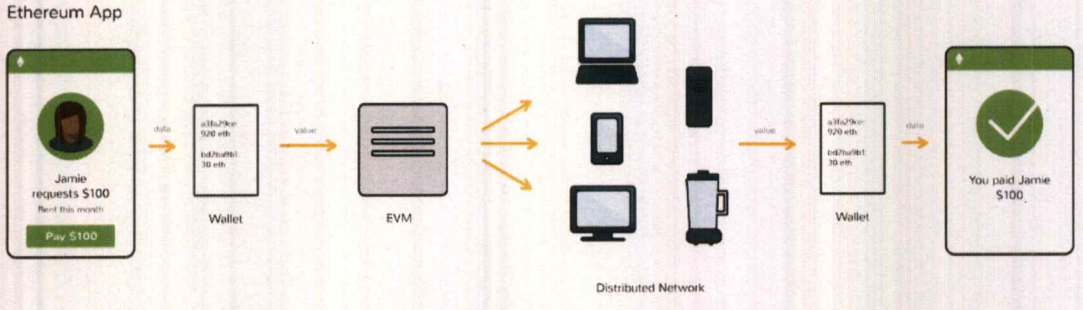
(BTC) ของผู้ใช้งาน เป็นต้น เมื่อเวลาผ่านไประยะหนึ่งจะเกิดทรานเซคชันมากมายในระบบ ทรานเซคชันทั้งหมดจะถูกรวบรวมเป็นบล็อค โดยระยะเวลาการเกิดบล็อคจะขึ้นอยู่กับระบบในแต่ละบล็อคเชน



รูปที่ 2.2 ตัวอย่างข้อมูลในบล็อคเชน

2.2 อีเธอเรียม (Ethereum)

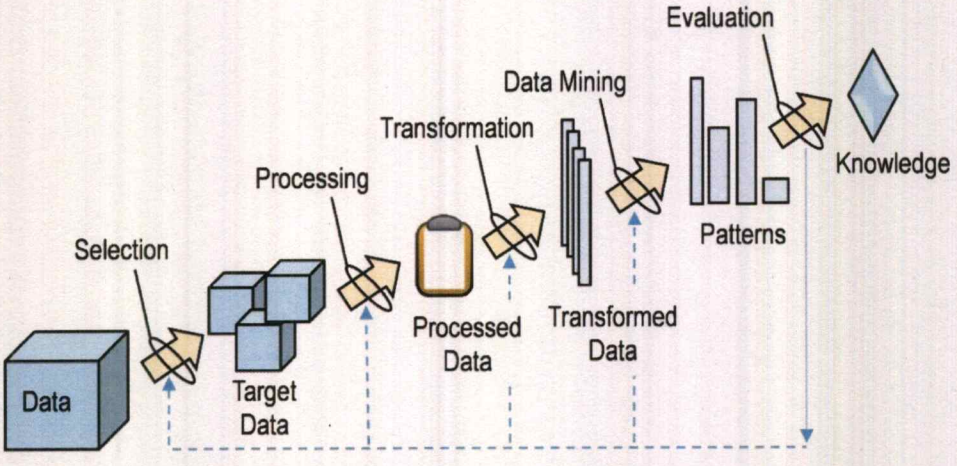
อีเธอเรียม เป็นแพลตฟอร์มแบบเปิดของเทคโนโลยีบล็อคเชน ที่ทำให้ทุกคนสามารถสร้างและใช้งานแอปพลิเคชันได้แบบกระจายข้อมูล (Decentralized) ซึ่งจะทำงานบนเทคโนโลยีบล็อคเชนได้ โดยที่อีเธอเรียมมีความคล้ายคลึงกับบิตคอยน์คือไม่สามารถควบคุมหรือเป็นเจ้าของอีเธอเรียมได้ เนื่องจากอีเธอเรียมเป็นโครงสร้างแบบโอเพนซอร์ส (Open-Source Project) ที่สร้างขึ้นโดยผู้คนจำนวนมากจากทั่วโลก ซึ่งอีเธอเรียมถูกออกแบบมาเพื่อให้สามารถปรับตัวและมีความหยืดหยุ่นในการทำงาน โดยอีเธอเรียมมีโปรโตคอลเครือข่ายแบบ peer-to-peer เช่นเดียวกับบล็อคเชนทั้งนี้ ฐานข้อมูลบล็อคเชนของอีเธอเรียมจะถูกเก็บรักษาและพัฒนาปรับปรุง โดยโหนดหลาย ๆ โหนดที่เชื่อมต่อกันเป็นเครือข่ายโดยแต่ละโหนดในเครือข่ายจะสามารถใช้ Ethereum Virtual Machine (EVM) ซึ่งสามารถประมวลผลที่มีอัลกอริทึมที่ซับซ้อนได้



รูปที่ 2.3 กระบวนการทำงานของอีเธอเรียม

2.3 การทำเหมืองข้อมูล (Data Mining)

การทำเหมืองข้อมูล คือกระบวนการค้นหาความรู้จากข้อมูลจำนวนมากเพื่อค้นหาภาพแบบและความสัมพันธ์ที่อยู่ในฐานข้อมูลขนาดใหญ่ จากนั้นจึงนำความรู้ที่ได้ไปใช้ประโยชน์ในการตัดสินใจ การทำเหมืองข้อมูลประกอบขึ้นด้วยการนำกระบวนการทางสถิติและการเรียนรู้ผ่านระบบคอมพิวเตอร์ และนำมาสร้างภาพแบบ กฎเกณฑ์ การพยากรณ์และความรู้จากฐานข้อมูลขนาดใหญ่ โดยการทำเหมืองข้อมูลมีขั้นตอนการดำเนินงานหลายวิธีซึ่งต้องอาศัยเทคนิคหรือวิธีการต่าง ๆ เช่น การหากฎความสัมพันธ์ การจำแนกข้อมูล การจัดกลุ่มข้อมูล เป็นต้น



รูปที่ 2.4 ขั้นตอนวิธีของการทำเหมืองข้อมูล

2.3.1 การหากฎความสัมพันธ์ (Association Rules Mining)

เป็นการค้นหากฎความสัมพันธ์ของข้อมูล โดยจะค้นหาความสัมพันธ์ที่เกิดขึ้นได้ทั้งหมดของข้อมูล ซึ่งข้อมูลที่จะนำมาใช้จะอยู่ในภาพของ Nominal หรือ Ordinal เท่านั้น ภาพแบบทั่วไปของการค้นหากฎความสัมพันธ์จะอยู่ในภาพแบบ A⟶B โดยที่ A เป็นเงื่อนไขและ B เป็นผลลัพธ์ที่เกิดขึ้น การประเมินค่าของกฎจะใช้ค่าสนับสนุน (Support) และค่าความเชื่อมั่น (Confidence)

โดยที่ค่าสนับสนุน คือเปอร์เซ็นต์ของข้อมูลที่มีเงื่อนไขและผลลัพธ์ที่สอดคล้องตามกฎต่อจำนวนข้อมูลทั้งหมด สามารถเขียนสมการได้ดังนี้

$$\text{Support}(A, B) = \frac{\text{จำนวนของ Transaction (A,B)}}{\text{จำนวน Transaction ทั้งหมด}} \qquad \textbf{(สมการที่ 2.1)}$$

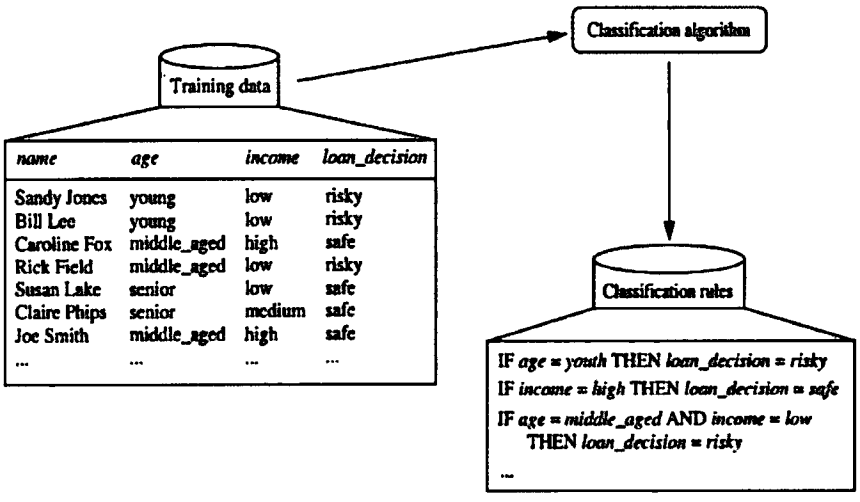
จากสมการที่ 2.1 เป็นสมการของค่าสนับสนุนโดยที่ A คือเหตุการณ์ที่ใช้เป็นเงื่อนไขในการหาผลลัพธ์ในส่วนของ B คือเหตุการณ์ที่เป็นผลลัพธ์ และTransaction (A,B) คือเหตุการณ์ที่ประกอบด้วยเหตุการณ์ A และ B

$$\text{Confidence}(A, B) = \frac{\text{จำนวนของ Transaction (A,B)}}{\text{จำนวน Transaction (A)}} \quad \textbf{(สมการที่ 2.2)}$$

จากสมการที่ 2.2 เป็นสมการของค่าความเชื่อมั่นโดยที่ Transaction (A) คือเหตุการณ์ที่ประกอบด้วยเหตุการณ์ A อย่างเดียวในการเลือกกฎใดนั้นจะต้องพิจารณาค่าสนับสนุนและค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าสนับสนุนและค่าความมั่นใจที่กำหนด โดยกำหนดค่าสนับสนุนต่ำสุด (Minimum Support) และค่าความเชื่อมั่นต่ำสุด (Minimum Confidence) โดยทั่วไปจะกำหนดค่าสนับสนุนต่ำสุดเป็น 5-10% และค่าความเชื่อมั่นต่ำสุดเป็น 50-100% ซึ่งอัลกอริทึม (Algorithm) ที่ใช้ในการค้นหาความสัมพันธ์เช่น Apriori Algorithm หรือ FP-growth Algorithm

2.3.2 การจำแนกข้อมูล (Data Classification)

เป็นวิธีในการจำแนกกลุ่มข้อมูลด้วยคุณลักษณะต่าง ๆ ของข้อมูลที่ได้มีการกำหนดไว้แล้ว วิธีนี้เหมาะกับการสร้างโมเดลจำแนกข้อมูล (Classifier) เพื่อทำนายหมวดหมู่ของข้อมูล (Class) จากการที่ได้จำแนกข้อมูลตัวอย่างไว้แล้ว ซึ่งในลักษณะดังกล่าวเรียกว่าการเรียนรู้แบบมีผู้สอน (Supervised Learning) วิธีการจำแนกกลุ่มเป็นกระบวนการสร้างโมเดลเพื่อจำแนกข้อมูลให้อยู่ในกลุ่มที่กำหนด เช่น การแบ่งประเภทลูกค้าที่มีแนวโน้มจะซื้อเครื่องใช้ไฟฟ้า ซึ่งเป็นการสร้างตัวแบบโดยการเรียนรู้จากข้อมูลที่ได้กำหนดไว้เรียบร้อยแล้ว เทคนิคที่ใช้ในการจำแนกกลุ่มข้อมูล เช่น การสร้างต้นไม้ตัดสินใจ (Decision Tree) เนอีฟเบย์ (Naïve-Bayes) เคเนียร์เรสเนเบอร์ (K-Nearest Neighbors : KNN) การวิเคราะห์การถดถอยเชิงเส้น (Linear Regression) โครงข่ายประสาทเทียม (Neural Networks) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine : SVM) เป็นต้น



รูปที่ 2.5 การเรียนรู้จากข้อมูลเพื่อสร้างโมเดลในการจำแนกข้อมูล

จากรูปที่ 2.5 แสดงถึงกระบวนการเรียนรู้จากข้อมูลเพื่อใช้สำหรับสร้างตัวจำแนกข้อมูล โดยชุดของข้อมูลที่ใช้ในการเรียนรู้จะบอกถึงคุณลักษณะต่าง ๆ ของข้อมูลซึ่งจะมีคุณลักษณะหนึ่งที่บ่งบอกถึงหมวดหมู่ของข้อมูล

2.3.3 การประเมินผลการทำเหมืองข้อมูล

การวัดประสิทธิภาพของโมเดลเป็นหนึ่งสิ่งที่สำคัญในการทำเหมืองข้อมูล ซึ่งการวัดประสิทธิภาพจะช่วยบ่งบอกถึงความเหมาะสมของโมเดลต่าง ๆ ที่นำมาใช้ในการจำแนก เช่น เมทริกซ์ความสับสน (Confusion Matrix) ค่าความถูกต้อง (Accuracy) ค่าความจำ (Recall) ค่าความแม่นยำ (Precision) และ ค่าเฉลี่ยประสิทธิภาพโดยรวม (F-measure)

1) เมทริกซ์ความสับสน (Confusion Matrix)

เป็นมาตรวัดสำหรับการประเมินผลลัพธ์การจำแนกของข้อมูลเปรียบเทียบกับผลลัพธ์จริง ๆ

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

รูปที่ 2.6 ตารางเมทริกซ์ความสับสน

จากรูปที่ 2.6 เป็นตารางเมทริกซ์ความสับสนโดยที่ TP คือผลการทำนายที่ถูกต้องตามค่าที่คาดหวังที่เป็น Positive TN คือผลการทำนายที่ถูกต้องตามค่าที่คาดหวังที่เป็น Negative FP คือผลการทำนายที่ไม่ถูกต้องตามค่าที่คาดหวังที่เป็น Positive และ FN คือผลการทำนายที่ไม่ถูกต้องตามค่าที่คาดหวังที่เป็น Negative

2) ค่าความถูกต้อง (Accuracy)

เป็นมาตรวัดที่นิยมใช้ในการวัดประสิทธิภาพในการทำนายผลสำหรับการจำแนกประเภทของข้อมูล

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad \textbf{(สมการที่ 2.3)}$$

3) ค่าความจำ (Recall)

เป็นมาตรวัดความถูกต้องของโมเดล โดยที่จะพิจารณาความถูกต้องของข้อมูลทีละคลาส

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad \textbf{(สมการที่ 2.4)}$$

4) ค่าความแม่นยำ

เป็นมาตรวัดความแม่นยำของข้อมูล โดยจะพิจารณาความแม่นยำทีละคลาสของข้อมูล

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FN}} \qquad \textbf{(สมการที่ 2.5)}$$

5) ค่าเฉลี่ยประสิทธิภาพโดยรวม

เป็นมาตรวัดประสิทธิภาพโดยเฉลี่ยของค่าความแม่นยำและค่าความจำ ดังสมการที่ 2.5 และสมการที่ 2.6

$$\text{F} - \text{measure} = \frac{\text{2 x Precision x Recall}}{\text{Precision+Recall}} \qquad \textbf{(สมการที่ 2.6)}$$

2.4 บริการฐานข้อมูล Google BigQuery

Google BigQuery เป็นบริการฐานข้อมูลขนาดใหญ่ของกูเกิลบนกูเกิลคราวแพลทฟอร์ม (Google Cloud Platform) โดยทำหน้าที่ในการวิเคราะห์และประมวลผลข้อมูลที่มีขนาดใหญ่ เพื่อหาผลลัพธ์ที่ต้องการได้อย่างมีประสิทธิภาพและรวดเร็ว โดยการใช้งานผู้ใช้สามารถใช้ความรู้พื้นฐานของภาษาเอสคิวแอล (SQL) ในการเรียกข้อมูลที่ต้องการได้ทันที

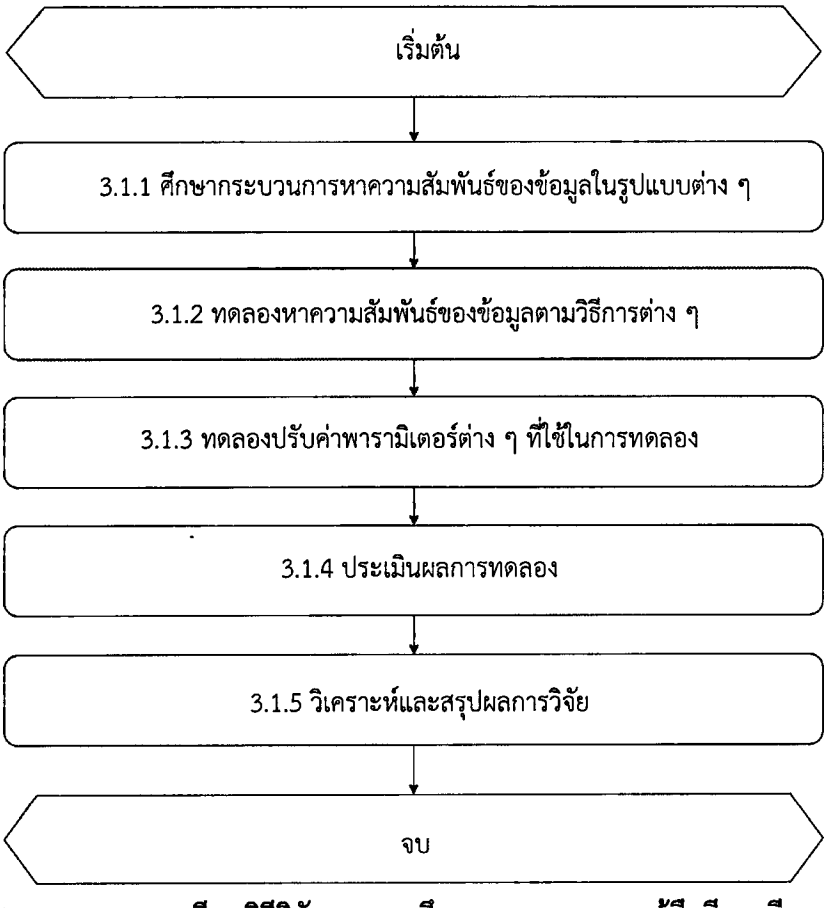
บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงการดำเนินการวิจัยเรื่องการศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่มีต่อมูลค่าของอีเธอเรียม โดยเนื้อหาประกอบด้วยระเบียบวิธีวิจัย และขั้นตอนวิธีการทดลองหาความสัมพันธ์ของข้อมูล

3.1 ระเบียบวิธีวิจัย

การศึกษาวิธีการต่าง ๆ ของการศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่มีต่อมูลค่าของอีเธอเรียม ซึ่งมีลำดับการดำเนินงานดังแสดงในตารางรูปที่ 3.1 โดยที่รายละเอียดของแต่ละขั้นตอน ได้อธิบายในหัวข้อที่ 3.1.1 ถึง 3.1.5



รูปที่ 3.1 แผนภาพระเบียบวิธีวิจัยของการศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่มีต่อมูลค่าของอีเธอเรียม

3.1.1 ศึกษากระบวนการหาความสัมพันธ์ของข้อมูลในรูปแบบต่าง ๆ

ศึกษาวิธีต่าง ๆ ที่ใช้สำหรับหาความสัมพันธ์ของข้อมูล เช่น ขั้นตอนหากฎความสัมพันธ์ (Association Rule miner) ขั้นตอนการจำแนกข้อมูล (Classification) หรือขั้นตอนการวิเคราะห์ความถดถอย (Regression Analysis) เป็นต้น

3.1.2 ทดลองหาความสัมพันธ์ของข้อมูลตามวิธีการต่าง ๆ

ทดลองการหาความสัมพันธ์ของข้อมูลตามวิธีการต่าง ๆ ที่ได้ศึกษามา โดยนำชุดข้อมูลที่ได้จาก Google BigQuery และ เว็บไซต์ Ethescan.io

3.1.3 ทดลองปรับค่าพารามิเตอร์ต่าง ๆ ที่ใช้ในการทดลอง

ทดลองปรับค่าพารามิเตอร์ของวิธีการต่าง ๆ ที่ได้ศึกษามา เพื่อทดสอบดูชุดของพารามิเตอร์และเลือกชุดของพารามิเตอร์ที่ให้ค่าผลลัพธ์ที่ดีที่สุด สำหรับการทดลอง

3.1.4 ประเมินผลการทดลอง

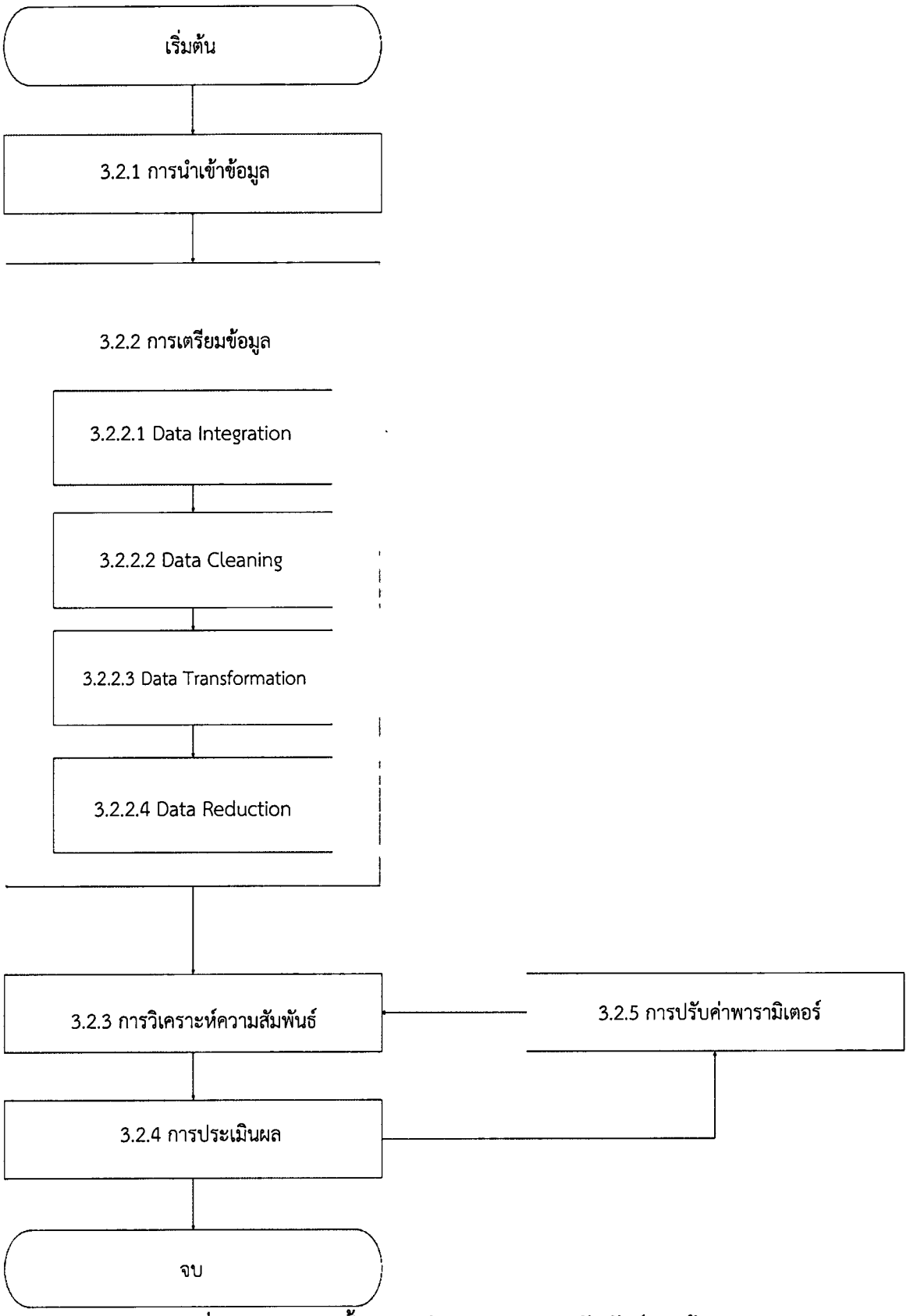
ประเมินผลการทดลอง โดยจะเน้นการวัดความแม่นยำ (Accuracy) ได้จากการทดลอง

3.1.5 วิเคราะห์และสรุปผลการวิจัย

ในขั้นตอนนี้จะนำผลลัพธ์ที่ได้จากการทดลองมาวิเคราะห์ เพื่อหาองค์ความรู้ที่ได้รับจากการทดลองโดยจะหาปัจจัยที่ส่งผลต่อผลลัพธ์ของการทดลอง

3.2 ขั้นตอนวิธีการทดลองการหาความสัมพันธ์ของข้อมูล

ในหัวข้อนี้จะกล่าวถึงการดำเนินการทดลองหาความสัมพันธ์ของข้อมูล ซึ่งมีขั้นตอนวิธีการทดลองดังแสดงในรูปที่ 3.2



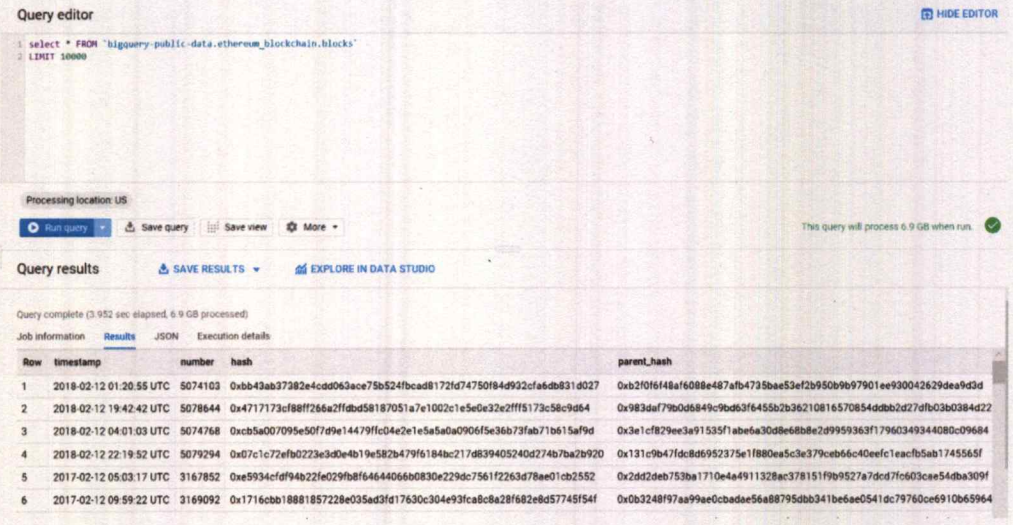
รูปที่ 3.2 แผนภาพขั้นตอนวิธีทดลองหาความสัมพันธ์ของข้อมูล

3.2.1 การนำเข้าข้อมูล

เป็นขั้นตอนการนำข้อมูลจากแหล่งต่าง ๆ ที่สนใจมาเป็นรวบรวมในฐานข้อมูล โดยการนำข้อมูลเข้าฐานข้อมูลมี 2 วิธีประกอบด้วย การนำข้อมูลเข้าฐานข้อมูลโดยใช้ Google BigQuery และการนำข้อมูลเข้าฐานข้อมูลโดยใช้โปรแกรมภาษาไพธอน (python)

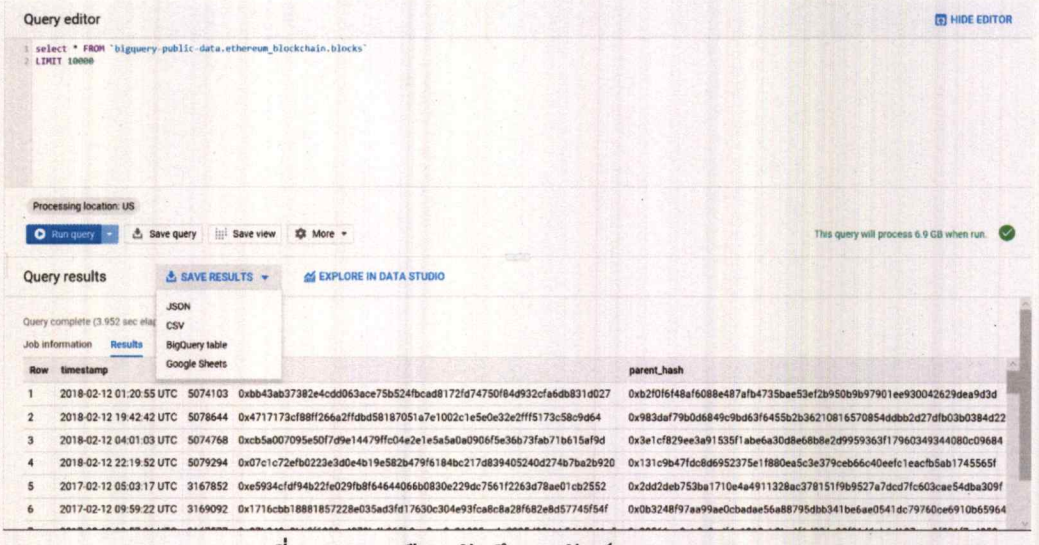
3.2.1.1 การนำข้อมูลเข้าฐานข้อมูลโดยใช้ Google BigQuery

การนำข้อมูลเข้าจะใช้ชุดข้อมูลที่ Query จากฐานข้อมูล Ethereum _blockchain ผ่าน Google BigQuery และนำข้อมูลเข้าฐานข้อมูลของ MySQL โดยใช้โปรแกรม JetBrains DataGrip 2018



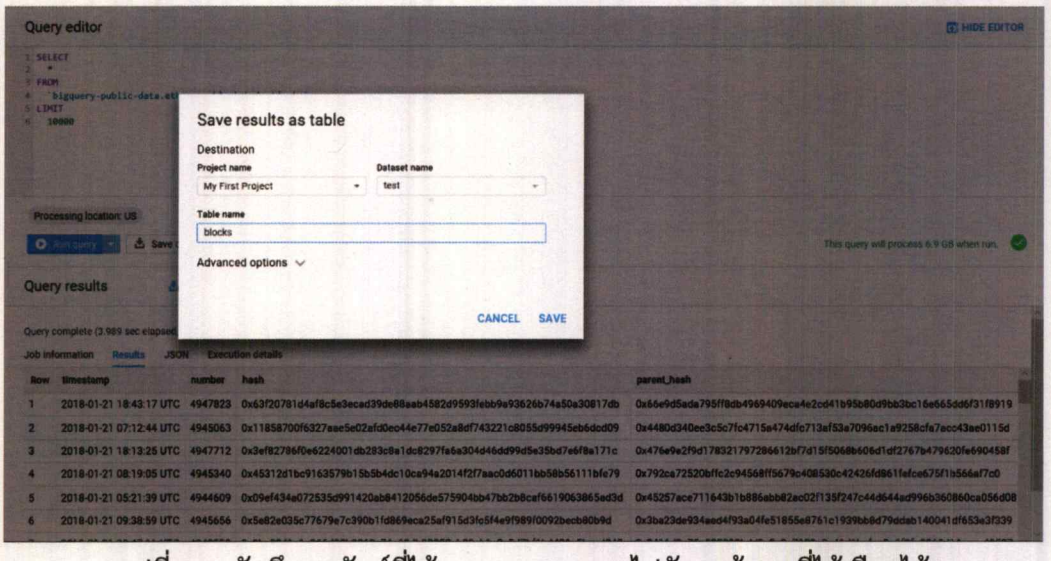
รูปที่ 3.3 ตัวอย่างการ Query ข้อมูลจากฐานข้อมูล Ethereum_blockchain ผ่าน Google BigQuery

จากรูปที่ 3.3 แสดงถึงตัวอย่างการ Query ข้อมูลทั้งหมดในตาราง blocks จากฐานข้อมูล Ethereum_blockchain ผ่าน Google BigQuery



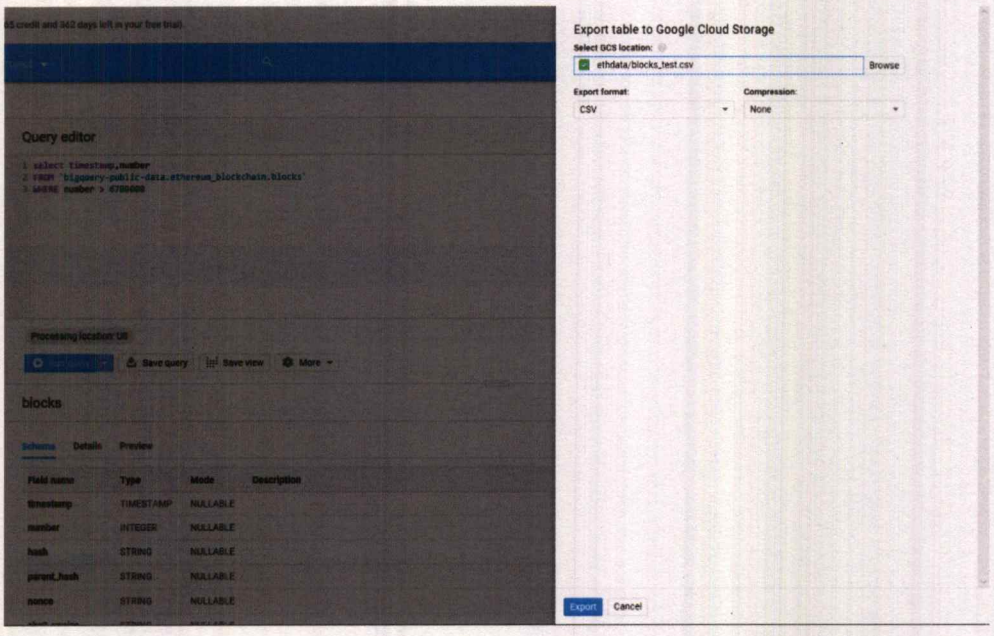
รูปที่ 3.4 การเลือกบันทึกผลลัพธ์จากการ Query

จากรูปที่ 3.4 แสดงถึงเป็นการเลือกบันทึกผลลัพธ์จากการ Query ที่ได้ทำในรูปที่ 3.2 เพื่อนำไปใช้ในขั้นตอนต่อ ๆ ไป



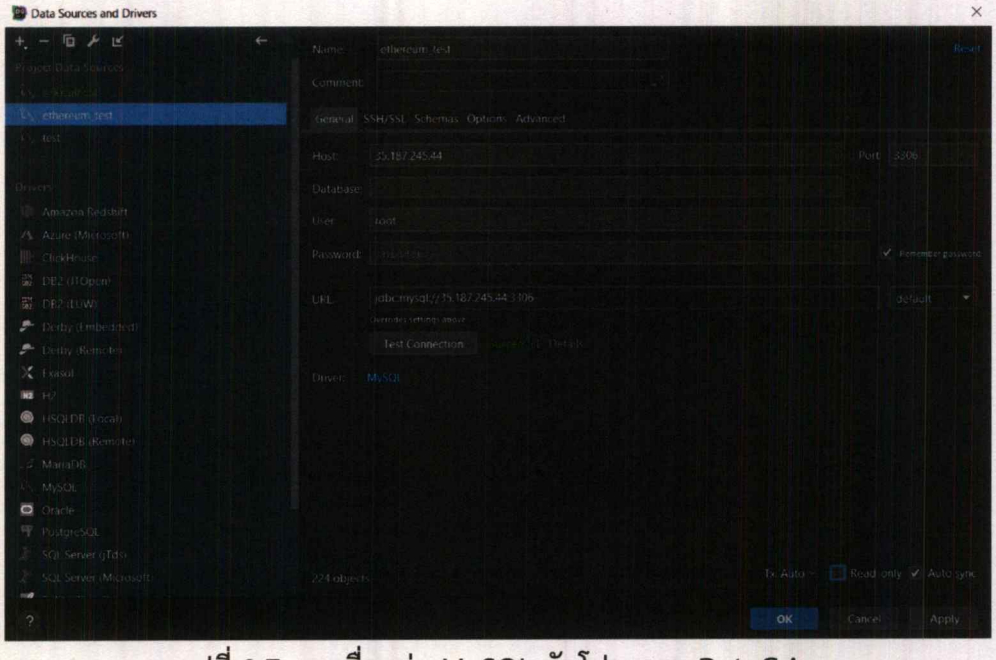
รูปที่ 3.5 บันทึกผลลัพธ์ที่ได้จากการ Query ไปยังชุดข้อมูลที่ได้เลือกไว้

จากรูปที่ 3.5 แสดงถึงการบันทึกผลลัพธ์ที่ได้จากรูปที่ 3.3 ไปยังชุดข้อมูลที่ได้เลือกไว้ เพื่อนำไปจัดเก็บในฐานข้อมูล



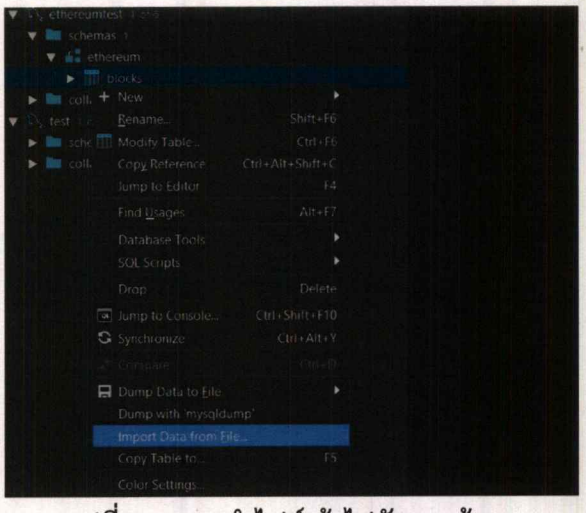
รูปที่ 3.6 นำข้อมูลออกมาออกมาเป็นไฟล์ csv

จากรูปที่ 3.6 แสดงถึงการนำชุดข้อมูลที่ได้จากรูปที่ 3.4 ออกมาเป็นไฟล์ csv เพื่อนำไปเข้าฐานข้อมูล



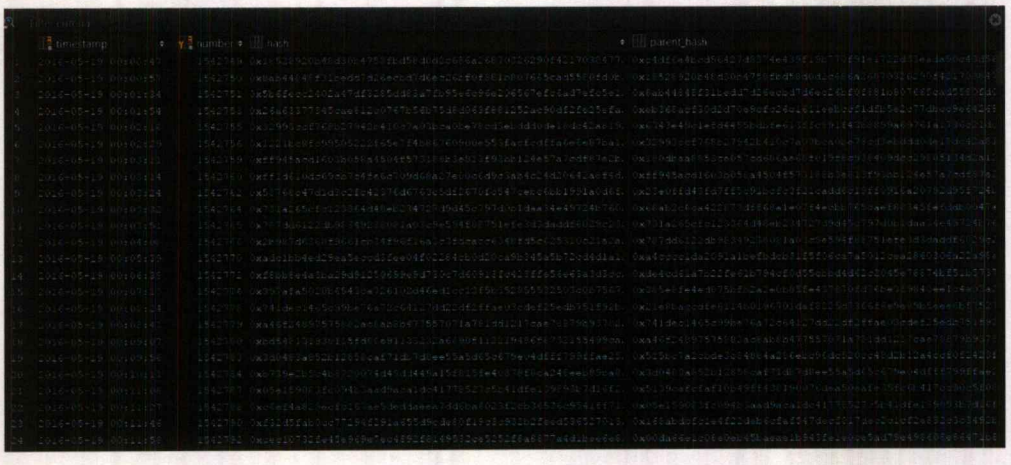
รูปที่ 3.7 การเชื่อมต่อ MySQL กับโปรแกรม DataGrip

จากรูปที่ 3.7 แสดงถึงการเชื่อมต่อ MySQL กับโปรแกรม DataGrip เพื่อจัดการข้อมูลในฐานข้อมูล



รูปที่ 3.8 การนำไฟล์เข้าไปยังฐานข้อมูล

จากรูปที่ 3.8 แสดงถึงการนำข้อมูลชนิดไฟล์ csv ที่ได้จากรูปที่ 3.5 นำเข้าไปยังฐานข้อมูล MySQL



รูปที่ 3.9 ผลลัพธ์จากการนำข้อมูลเข้าฐานข้อมูล

จากรูปที่ 3.9 แสดงถึงผลลัพธ์ที่ได้จากการนำข้อมูลลงฐานข้อมูลในรูปที่ 3.7 โดยแสดงข้อมูลทั้งหมดที่อยู่ในฐานข้อมูล

3.2.1.2 การนำข้อมูลเข้าฐานข้อมูลโดยใช้โปรแกรมภาษาไพธอน

เป็นการนำข้อมูลเข้าฐานข้อมูลโดยใช้ไลบรารี่ (Library) ของภาษาไพธอน โดยการนำข้อมูลเข้าฐานข้อมูลโดยใช้โปรแกรม แบ่งออกเป็น 2 ส่วนคือ การนำข้อมูลเข้าฐานข้อมูล balance และการนำข้อมูลเข้าฐานข้อมูล price_history_day

1) การนำข้อมูลเข้าฐานข้อมูล balance

การนำข้อมูลเข้าจะใช้ข้อมูลรายชื่อผู้ถืออีเธอเรียมที่อยู่ในเว็บไซต์ etherscan.io และใช้ข้อมูลหมายเลขบล็อคเชน บล็อคแรกในแต่ละวัน โดยการ Query หมายเลขบล็อคเชนจากฐานข้อมูล Ethereum_blockchain ผ่าน Google BigQuery โดยจะเก็บข้อมูลในรูปแบบไฟล์ csv เพื่อนำไปใช้ในการค้นหาข้อมูลผู้ถืออีเธอเรียมและนำข้อมูลเข้าฐานข้อมูล MySQL

```
class DB:

    @staticmethod
    def ethereum(db_name='ethereum'):
        return 'mysql+pymysql://root:1234@35.187.245.44/{}'.format(db_name)
```

รูปที่ 3.10 การเชื่อมต่อภาษาไพธอนกับฐานข้อมูล

จากรูปที่ 3.10 แสดงถึงการตั้งค่าเชื่อมต่อเข้าใช้งานของภาษาไพธอนเข้ากับฐานข้อมูล MySQL

```
def get_address():
    result = []
    s = requests.Session()

    for page in tqdm(range(1, 101)):
        r = s.get('https://etherscan.io/accounts/{}?ps=100'.format(page))
        html = r.text
        soup = BeautifulSoup(html, 'html.parser')
        td = soup.select('td[width="330px"]')

        for t in td:
            result.append(t.text)

    return result
```

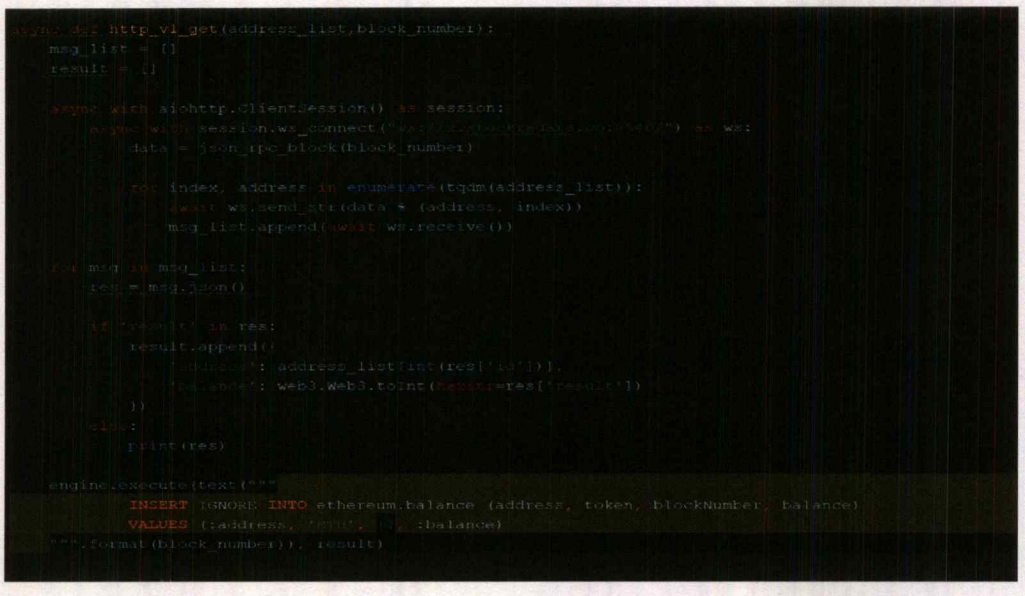
รูปที่ 3.11 แสดงถึงการนำรายชื่อผู้ถืออีเธอเรียมจากเว็บไซต์มาใช้

จากรูปที่ 3.11 การนำรายชื่อของผู้ถืออีเธอเรียมรายใหญ่จากเว็บไซต์ etherscan.io มาใช้ในการหาข้อมูลของผู้ถืออีเธอเรียม

```
def get_numblocks():

    numblock = []
    with open('numblocks.csv') as csv_file:
        csv_reader = csv.reader(csv_file)
        print(csv_reader)

        for row in csv_reader:
            numblock.append(row[0])
        """ Convert str to int list"""
        numblock = list(map(int, numblock))

    return numblock
```

รูปที่ 3.12 การอ่านไฟล์ csv มาใช้

จากรูปที่ 3.12 เป็นการนำข้อมูลหมายเลขบล็อคเชนจากไฟล์ csv เพื่อนำมาใช้ในการดึงข้อมูลจากบล็อคหมายเลขนั้น ๆ



```
async def http_v1_get(address_list,block_number):
    msg_list = []
    result = []

    async with aiohttp.ClientSession() as session:
        async with session.ws_connect("wss://[redacted]") as ws:
            data = json_rpc_block(block_number)

            for index, address in enumerate(tqdm(address_list)):
                await ws.send_str(data % (address, index))
                msg_list.append(await ws.receive())

    for msg in msg_list:
        res = msg.json()

        if "result" in res:
            result.append({
                "index": address_list[int(res['id'])],
                "eth_bal": web3.Web3.toInt(hexstr=res['result'])
            })
        else:
            print(res)

    engine.execute(text("""
            INSERT IGNORE INTO ethereum.balance (address, token, blockNumber, balance)
            VALUES (:address, '0x0', [redacted], :balance)
    """.format(block_number)), result)
```

รูปที่ 3.13 การนำข้อมูลเข้าฐานข้อมูล

จากรูปที่ 3.13 เป็นการนำข้อมูลเข้าฐานข้อมูล โดยเชื่อมต่อกับเซิฟเวอร์ของ StockRadars และใช้ข้อมูลรายชื่อผู้ถืออีเธอเรียมดังรูปที่ 3.10 รวมถึงหมายเลขบล็อคแรกในแต่วันดังรูปที่ 3.11 มาใช้ในการหาข้อมูลและนำข้อมูลเข้าฐานข้อมูล MySQL

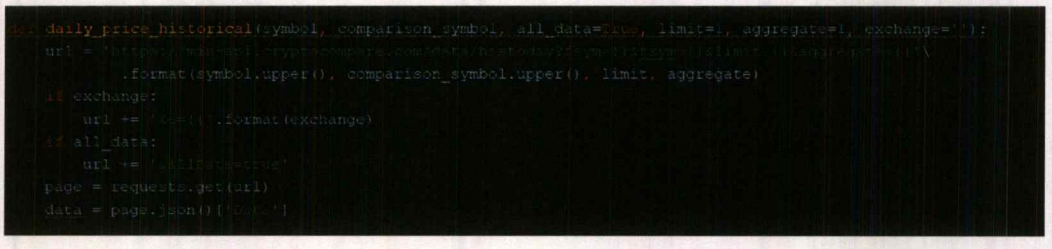
	N_address	I_order ▲ 1	N_balance	Z_percentage
1	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.5050
2	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.5050
3	0xfe9e8709d3215310075d67e3ed32a380ccf451c	1	1,538,662.1235	1.4990
4	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4980
5	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4980
6	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4970
7	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4970
8	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4970
9	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4970
10	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4960
11	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4960
12	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4960
13	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4950
14	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4950
15	0x281055afc982d96fab65b3a49cac8b878184cb1	1	1,538,423.1066	1.4950
16	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,583,220.5681	1.5380
17	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,613,603.7923	1.5660
18	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,680,613.5574	1.6310
19	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,691,212.6025	1.6410
20	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,661,212.6020	1.6120
21	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,661,212.5978	1.6110
22	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,661,212.5970	1.6110
23	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,661,212.5970	1.6110
24	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,661,212.5970	1.6100
25	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,614,820.5092	1.5650
26	0x742d35cc6634c0532925a3b844bc454e4438f44	1	1,614,820.5092	1.5650

รูปที่ 3.14 ผลลัพธ์จากการนำข้อมูลเข้าฐานข้อมูล

จากรูปที่ 3.14 แสดงถึงผลลัพธ์ที่ได้จากการนำข้อมูลลงฐานข้อมูลในรูปที่ 3.12 โดยแสดงข้อมูลทั้งหมดที่อยู่ในฐานข้อมูล

2) การนำข้อมูลเข้าฐานข้อมูล price_history_day

การนำข้อมูลเข้าฐานข้อมูลโดยเรียกข้อมูลที่ต้องการผ่าน API ของเว็บไซต์ CryptoCompare และนำข้อมูลเข้าฐานข้อมูล MySQL



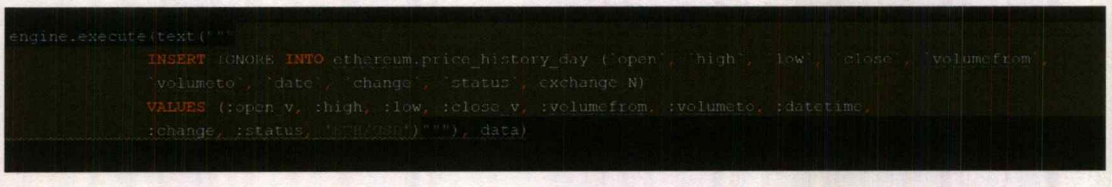
รูปที่ 3.15 การเรียกข้อมูลที่ต้องการผ่าน API ของเว็บไซต์ CryptoCompare

จากรูปที่ 3.15 เป็นการเรียกข้อมูลผ่าน API ของเว็บไซต์ CryptoCompare โดยข้อมูลที่ได้เป็นข้อมูลในรูปแบบ JSON

```
d_dict = []

for d in tqdm(data):
    status = ''
    change = d['open']-d['close']
    if change >= 7:
        status = 'S_high'

    elif change >= 0:
        status = 'high'

    elif change <= -7:
        status = 'S_low'

    elif change < 0:
        status = 'low'


    d_dict.append({
        'datetime': datetime.datetime.fromtimestamp(d['time']).strftime('%Y-%m-%d'),
        'change': change,
        'status': status,
        'open_y': d['open'],
        'close_y': d['close'],
        'high': d['high'],
        'low': d['low'],
        'volumefrom': d['volumefrom'],
        'volumeto': d['volumeto']
    })
```

รูปที่ 3.16 เตรียมข้อมูลเพื่อนำเข้าฐานข้อมูล MySQL

จากรูปที่ 3.16 เป็นการนำข้อมูลที่ได้จากเว็บไซต์ CryptoCompare ดังรูปที่ 3.14 มาเตรียมเพื่อนำเข้าฐานข้อมูล MySQL โดยมีการจัดโครงสร้างของข้อมูลเพื่อให้ข้อมูลมีความถูกต้องมากยิ่งขึ้น



```
engine.execute(text("""
        INSERT IGNORE INTO ethereum.price_history_day (`open`, `high`, `low`, `close`, `volumefrom`,
        `volumeto`, `date`, `change`, `status`, exchange_N)
        VALUES (:open_y, :high, :low, :close_y, :volumefrom, :volumeto, :datetime,
        :change, :status, 'ETH/USD')"""), data)
```

รูปที่ 3.17 การนำข้อมูลเข้าฐานข้อมูล MySQL

จากรูปที่ 3.17 เป็นการนำข้อมูลที่ได้จากเว็บไซต์ CryptoCompare ดังรูปที่ 3.14 และเตรียมข้อมูลเพื่อนำเข้าฐานข้อมูล MySQL ดังรูปที่ 3.15

3.2.2 การเตรียมข้อมูล (Preprocessing)

เป็นขั้นตอนในการเตรียมข้อมูลเพื่อนำไปประมวลผล ซึ่งขั้นตอนนี้จะทำให้ข้อมูลดิบหรือข้อมูลเริ่มต้น มีความถูกต้องแม่นยำและมีขนาดตรงตามที่ต้องการมากยิ่งขึ้น โดยกระบวนการเตรียมข้อมูลนั้น จะมีกระบวนการย่อย ๆ ดังต่อไปนี้

3.2.2.1 Data Integration

เป็นขั้นตอนการรวบรวมข้อมูลจากหลายแหล่งข้อมูล เพื่อนำมาใช้ในการประมวลผล เช่น การรวมไฟล์ที่มีลักษณะเหมือนกัน โดยรายละเอียดการรวบรวมข้อมูลได้กล่าวในหัวข้อที่ 3.2.1.1 และ 3.2.1.2

3.2.2.2 Data Cleaning

เป็นขั้นตอนการจัดการข้อมูลให้มีความถูกต้อง เช่น การจัดการข้อมูลขาดหาย (Nan หรือ NULL) หรือการจัดการข้อมูลที่มีค่าต่างจากข้อมูลส่วนใหญ่มากเกินไป (Outlier)

1) การปรับค่าของข้อมูล

เป็นการปรับค่าความแปรปรวน หรือค่าเฉลี่ยของข้อมูลให้มีค่าเท่ากันก่อนที่จะนำข้อมูลไปวิเคราะห์

```
from sklearn import preprocessing
ss = preprocessing.StandardScaler()
df[cols] = ss.fit_transform(df[cols])
```

รูปที่ 3.18 ฟังก์ชันปรับค่าของข้อมูล

จากรูปที่ 3.18 เป็นตัวอย่างการปรับค่าของข้อมูล โดยใช้ฟังก์ชัน preprocessing.StanderScaler() ในการปรับค่าของข้อมูลให้เท่ากันเพื่อประสิทธิภาพในการวิเคราะห์ข้อมูล

2) การจัดการข้อมูลขาดหาย

เป็นทำความสะอาดข้อมูลโดยจะตรวจสอบข้อมูลขาดหาย และจัดการข้อมูลขาดหายดังกล่าว โดยจะแสดงตัวอย่างการตรวจสอบดังรูปที่ 3.19 และ 3.20

```
# check count missing value
df.isnull().sum()
```

```
from_address        0
to_address      24000
value               0
date                0
balance             0
status              0
open                0
high                0
low                 0
close               0
volumefrom          0
dtype: int64
```

รูปที่ 3.19 คำสั่งตรวจสอบข้อมูลสูญหาย

จากรูปที่ 3.19 เป็นตัวอย่างคำสั่งตรวจสอบข้อมูลขาดหายในชุดข้อมูล และตรวจสอบจำนวนข้อมูลขาดหาย โดยจะจัดการข้อมูลสูญหายดังรูปที่ 3.20

```
# drop row missing value from index
index_df = df[df.to_address.isna()].index
df = df.drop(index_df, axis=0)
```

รูปที่ 3.20 คำสั่งลบแถวข้อมูลขาดหาย

จากรูปที่ 3.20 เป็นตัวอย่างคำสั่งลบแถวที่มีข้อมูลขาดหาย เนื่องจากชุดข้อมูลมีปริมาณข้อมูลที่มากจึงสามารถลบแถวที่มีข้อมูลขาดหายได้

3.2.2.3 Data Transformation

เป็นขั้นตอนสำหรับการแปลงข้อมูลดิบเป็นข้อมูลที่สามารถนำไปใช้ในการประมวลผลได้อย่างมีประสิทธิภาพ สำหรับการแปลงประเภทข้อมูลเพื่อให้ข้อมูลสามารถนำไปวิเคราะห์ในวิธีต่าง ๆ ได้โดยแบ่งการแปลงข้อมูลออกเป็น 3 แบบคือ การแบ่งช่วง

ข้อมูล การแปลงข้อมูลที่เป็นข้อความเป็นตัวเลข (Label Encoder) และการแปลงข้อมูลที่เป็นข้อความเป็นวันฮ็อท (One-hot Encoding)

1) การแบ่งช่วงข้อมูล

เป็นการแบ่งช่วงของข้อมูลที่เป็นตัวเลข และแปลงข้อมูลที่แบ่งดังกล่าวเป็นข้อความโดยจะแสดงตัวอย่างคำสั่งการแบ่งช่วงของข้อมูลดังรูปที่ 3.21

```
low = mean - std
high = mean + std
med = high - low
print(f"low={low}")
print(f"high={high}")
print(f"med={med}")
```

```
low=-1.0680733197238659e+18
high=5.473181009614352e+18
med=6.541254329338218e+18
```

```
def ordinal(value):
    y = value
    if -1.0686997692411543e+18 >= y:
        return "low"
    elif -1.0686997692411543e+18 < y < 5.473347679261484e+18:
        return "med"
    else:
        return "high"
```

รูปที่ 3.21 การแบ่งช่วงข้อมูล

จากรูปที่ 3.21 เป็นตัวอย่างการแบ่งช่วงข้อมูลของคอลัมน์ value โดยแบ่งข้อมูลออกเป็น 3 ช่วงคือ low med และ high

2) การแปลงข้อมูลที่เป็นข้อความเป็นตัวเลข (Label Encoder)

เป็นการแปลงข้อมูลประเภทข้อความให้เป็นตัวเลข โดยจะแสดงตัวอย่างคำสั่งการแปลงข้อมูลดังรูปที่ 3.22

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df['status'] = le.fit_transform(df['status'])
```

รูปที่ 3.22 ฟังก์ชันแปลงข้อมูลเป็นตัวเลข

จากรูปที่ 3.22 เป็นการแปลงข้อมูลข้อความให้เป็นตัวเลข โดยใช ฟังก์ชัน preprocessing.LabelEncodel() ในการแปลงข้อมูลให้เป็นตัวเลขเพื่อนำข้อมูลดังกล่าวไปใช้ในการวิเคราะห์ต่อไป

```
df['status'].head()
```

```
0    S_low
1    S_low
2      low
3      low
4      low
Name: status, dtype: object
```

```
df['status'].head()
```

```
0    1
1    1
2    3
3    3
4    3
Name: status, dtype: int64
```

รูปที่ 3.23 ข้อมูลที่เป็นข้อความและข้อมูลตัวเลข

จากรูปที่ 3.23 แสดงตัวอย่างข้อมูลก่อนแปลงข้อมูล จากรูปที่ 3.21 และตัวอย่างผลลัพธ์ที่ได้จากการแปลงข้อมูล จากรูปที่ 3.22

3) การแปลงข้อมูลที่เป็นข้อความเป็นวันฮ็อท (One-hot Encoding)

เป็นการแปลงข้อมูลประเภทข้อความให้เป็นตัวเลขศูนย์และหนึ่ง โดยที่ค่าหนึ่งหมายถึงมีข้อมูลชนิดนั้น และค่าศูนย์หมายถึงไม่มีข้อมูลชนิดนั้น ซึ่งการแปลงข้อมูลวิธีดังกล่าวทำให้ข้อมูลยังคงความสัมพันธ์

```
# convert to One-Hot-Encoder
def cv_ohe(df):
    cv_from(df)
    cv_to(df)
    df = pd.get_dummies(df)
    return df
```

รูปที่ 3.24 ฟังก์ชันแปลงข้อมูลเป็นวันฮ็อต

จากรูป 3.24 เป็นการแปลงข้อมูลที่เป็นข้อความให้เป็นวันฮ็อต โดยใช้ฟังก์ชัน pandas.get_dummies() ในการแปลงข้อมูลเพื่อนำข้อมูลดังกล่าวไปใช้ในการหาความสัมพันธ์

3.2.2.4 Data Reduction

เป็นขั้นตอนการลดความซ้ำซ้อนของข้อมูล โดยกระบวนการเตรียมข้อมูล ที่กล่าวมาข้างต้นสามารถแสดงรายละเอียดฐานข้อมูลดังต่อไปนี้

1) พจนานุกรมข้อมูล (Data Dictionary)

ในการวิเคราะห์และออกแบบข้อมูลนั้นต้องมีการเขียนคำอธิบายข้อมูล (Data Description) หรือพจนานุกรมข้อมูล ซึ่งเป็นสิ่งที่จัดเก็บรายละเอียดของข้อมูลการพัฒนาระบบเนื่องจากทุกฐานข้อมูลจะมีการจัดเก็บรายละเอียดต่าง ๆ เกี่ยวกับข้อมูลภายในฐานข้อมูล

ตารางที่ 3.1 ตารางข้อมูลยอดเงินในบัญชี (balance)

Key	ชื่อแอตทริบิวต์	ประเภทข้อมูล	คำอธิบาย
PK	address	Varchar	หมายเลขบัญชีของผู้ใช้
	token	Varchar	ชื่อของสกุลเงินอิเล็กทรอนิกส์
PK, FK	blockNumber	Varchar	หมายเลขของบล็อคเชน
	balance	Decimal	จำนวนเงินคงเหลือในบัญชี

ตารางที่ 3.2 ตารางข้อมูลราคาของค่าเงินในแต่ละวัน (price_history_day)

Key	ชื่อแอตทริบิวต์	ประเภทข้อมูล	คำอธิบาย
	exchange_N	Varchar	ประเภทอัตราการแลกเปลี่ยนเงิน
	open	Decimal	ค่าเงินเปิดตัวของสกุลเงินอีเธอเรียม
	high	Decimal	ค่าเงินสูงสุดในช่วงวันของสกุลเงินอีเธอเรียม
	low	Decimal	ค่าเงินต่ำสุดในช่วงวันของสกุลเงินอีเธอเรียม
	close	Decimal	ค่าเงินปิดตัวของสกุลเงินอีเธอเรียม
	volumefrom	Double	ปริมาณของเงินซื้อขายในมุมมองสกุลเงินดอลลาร์ (USD)
	volumeto	Double	ปริมาณของเงินซื้อขายในมุมมองสกุลเงินอีเธอเรียม (ETH)
FK	date	Date	เวลาของค่าเงิน ณ เวลานั้น
	change	Decimal	อัตราการเปลี่ยนแปลงของค่าเงิน
	status	Varchar	สถานะของราคาอีเธอเรียม

ตารางที่ 3.3 ตารางข้อมูลธุรกรรมการซื้อขาย (transaction)

Key	ชื่อแอตทริบิวต์	ประเภทข้อมูล	คำอธิบาย
	from_address	Varchar	หมายเลขบัญชีผู้ส่ง
	to_address	Varchar	หมายเลขบัญชีผู้รับ
	value	BigInt	จำนวนเงิน
FK	block_timestamp	Timestamp	เวลาที่เกิดบล็อคเชน ณ เวลานั้น
FK	block_number	Int	หมายเลขบล็อคเชน

3.2.3 การวิเคราะห์ความสัมพันธ์ของข้อมูล

เป็นขั้นตอนการนำข้อมูลที่ได้จากหัวข้อที่ 3.2.1.2 ดังที่กล่าวไปข้างต้น มาวิเคราะห์หาความสัมพันธ์ของข้อมูลโดยใช้เทคนิคการทำเหมืองข้อมูล เช่น การจำแนกกลุ่มของข้อมูล การวิเคราะห์การถดถอยของข้อมูล การหากฎความสัมพันธ์ของข้อมูล

3.2.3.1 ขั้นตอนวิธีการทดลองโดยใช้วิธีการจำแนกกลุ่มข้อมูล

การนำข้อมูลเข้าจะใช้ชุดข้อมูลจาก balance price_history_day และ transactions ดังที่กล่าวไปแล้วในหัวข้อ 3.2.2.1 โดยนำข้อมูลรวมกันเป็นชุดเดียวซึ่งตัวอย่างข้อมูลที่นำมาใช้งานมีดังนี้

1) ส่วนของข้อมูลนำเข้า

ข้อมูลของชุดข้อมูล balance price_history_day และ transactions ที่รวมกันเป็นชุดเดียวประกอบไปด้วยคุณลักษณะของข้อมูล 10 คุณลักษณะคือ from_address, to_address, value, date, balance, status, open, high, low, close และ volmefrom

from_address	to_address	value	date	balance	status	open	high	low	close	volumefrom
0xfbb1b73c4f0bda4f67dca266ce6ef42f520fbb98	0xb47002a89e6bb9769cf5ef2f5910f8e379bfc37d	0.000000e+00	24-08-17	6.000000e+22	S_low	317.40	328.65	315.94	325.28	352256.90
0xfbb1b73c4f0bda4f67dca266ce6ef42f520fbb98	0x960b236a07cf122663c4303350609a66a7b288c0	0.000000e+00	24-08-17	6.000000e+22	S_low	317.40	328.65	315.94	325.28	352256.90
0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be	0xef850775cb1a2c5a53e7a3702ceb9973f2566b39	1.422360e+17	25-08-17	6.000000e+22	low	325.28	337.24	324.69	330.06	428182.32
0xd7394025387eb5bdff4ef568b863ef4994137b3f	0x8d12a197cb00d4747a1fe03395095ce2a5cc6819	0.000000e+00	25-08-17	6.000000e+22	low	325.28	337.24	324.69	330.06	428182.32
0x933153cb57b715907aa041ad25ffc343bbf657c2	0x8d12a197cb00d4747a1fe03395095ce2a5cc6819	0.000000e+00	25-08-17	6.000000e+22	low	325.28	337.24	324.69	330.06	428182.32

รูปที่ 3.25 ตัวอย่างข้อมูลในแต่ละคุณลักษณะ

จากรูปที่ 3.25 เป็นชุดตัวอย่างข้อมูลในแต่ละคุณลักษณะ ที่เกิดจากการรวมของชุดข้อมูล 3 ชุดคือ balance price_history_day และ transactions โดยเลือกคุณลักษณะที่เหมาะสมในการทดลอง

2) ส่วนของการประมวลผล

ในการประมวลผลจะปรับ criterion ในทดลองเพื่อให้มีค่าความแม่นยำมากที่สุด

3) ส่วนของผลลัพธ์การคำนวณ

ผลลัพธ์ที่ได้จะนำไปเปรียบเทียบกับวิธีต่าง ๆ ของการทดลองจากในงานวิจัยนี้ เพื่อให้ทราบถึงวิธีที่เหมาะสมสำหรับการวิเคราะห์ความสัมพันธ์ของข้อมูล โดยในการเขียนคำสั่งเพื่อคำนวณผลลัพธ์จะใช้ Scikit-learn Framework และใช้โปรแกรม Rapidminer Studio ทดสอบชุดข้อมูลดังที่กล่าวในรูปที่ 3.25

3.2.3.2 ขั้นตอนการทดลองโดยใช้ต้นไม้ตัดสินใจด้วย Scikit-learn Framework

เนื่องจากในการเรียกใช้ Scikit-learn Framework นั้นไม่สามารถวิเคราะห์คุณลักษณะที่เป็นข้อความได้โดยตรง จึงต้องทำการแปลงข้อมูลนำเข้าที่เป็นตัวอักษรให้เป็นตัวเลขเสียก่อนโดยจะใช้ฟังก์ชัน preprocess. LabelEncoder() ของ Scikit-learn Framework ผลลัพธ์ของกระบวนการชุดนี้คือชุดตัวเลขที่แทนตัวอักษรในคุณลักษณะดังกล่าว ซึ่งคำสั่งที่ใช้แสดงดังรูปที่ 3.26 และสำหรับการเรียกใช้ Scikit-learn

Framework สามารถพิมพ์คำสั่ง import sklearn ได้ในโปรแกรมโดยจะหมายถึงการเรียกใช้ทุกฟังก์ชันทั้งหมดของ Scikit-learn แต่ในปัญหาพิเศษนี้จะเลือกใช้ฟังก์ชันบางส่วนตามรูปที่ 3.27

```
# Convert to label
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df['from_address_le'] = le.fit_transform(df['from_address'])
df['to_address_le'] = le.fit_transform(df['to_address'])
```

รูปที่ 3.26 คำสั่งใช้งานฟังก์ชันการเตรียมข้อมูล

จากรูปที่ 3.26 เป็นตัวอย่างคำสั่งการเตรียมข้อมูลโดยใช้ฟังก์ชัน LabelEncoder() ในการแปลงตัวอักษรเป็นตัวเลขทั้งหมด

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

รูปที่ 3.27 คำสั่งเรียกใช้งาน Scikit-learn Framework

จากรูปที่ 3.27 เป็นตัวอย่างฟังก์ชันบางส่วนของ Scikit-learn Framework ที่นำไปใช้สร้างต้นไม้ตัดสินใจประกอบไปด้วยคำสั่ง from sklearn.model_selection import train_test_split คือการเรียกใช้ฟังก์ชันการแบ่งข้อมูล from sklearn.tree import DecisionTreeClassifier หมายถึง การเรียกใช้โมเดล DecisionTreeClassifier

ข้อมูลที่จะใช้ฝึกสอนและทดสอบโมเดลจะถูกแบ่งออกเป็นสองส่วนได้แก่ ส่วนข้อมูลนำเข้าและคลาสคำตอบโดยคำสั่งที่ใช้แสดงดังรูปที่ 3.27

```
cols = ['value', 'balance', 'open', 'high', 'low', 'close',
        'volumefrom']
X = df[cols]
y = df['status']
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3)
```

รูปที่ 3.28 คำสั่งแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

จากรูปที่ 3.28 ตัวแปร X จะเก็บข้อมูลนำเข้า และตัว Y เก็บคลาสคำตอบที่ใช้สำหรับการฝึกสอนและการทดสอบโมเดล โดยแบ่งข้อมูลการฝึกสอน 70% เก็บไว้ในตัวแปร X_train กับ y_train และข้อมูลการทดสอบ 30% เก็บไว้ในตัวแปร X_test กับ y_test

ขั้นตอนต่อไปนี้คือการสร้างต้นไม้ตัดสินใจโดยคำสั่งที่ใช้จะแสดงได้ดังรูปที่ 3.29

```
# build model
model = DecisionTreeClassifier(criterion='entropy')
model.fit(X_train, y_train)
prediction = model.predict(X_test)
```

รูปที่ 3.29 คำสั่งสร้างต้นไม้ตัดสินใจสำหรับค่า criterion เป็น entropy

จากรูปที่ 3.29 เป็นตัวอย่างคำสั่งการสร้างต้นไม้ตัดสินใจโดยกำหนดค่า criterion เป็น entropy และทดสอบความแม่นยำ โดยคำสั่งจะแสดงดังรูปที่ 3.28

```
from sklearn.metrics import accuracy_score,classification_report
print(accuracy_score(y_test, prediction))
print(classification_report(y_test, prediction))
```

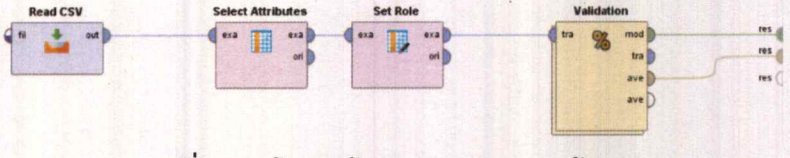
```
1.0
              precision    recall  f1-score   support

     S_high       1.00      1.00      1.00   1035817
      S_low       1.00      1.00      1.00   1215262
       high       1.00      1.00      1.00    476742
        low       1.00      1.00      1.00    613499

   accuracy                           1.00   3341320
  macro avg       1.00      1.00      1.00   3341320
weighted avg      1.00      1.00      1.00   3341320
```

รูปที่ 3.30 คำสั่งเรียกใช้งานการทดสอบความแม่นยำของต้นไม้ตัดสินใจและผลลัพธ์การทดสอบ

จากรูปที่ 3.30 เป็นตัวอย่างคำสั่งคำนวณค่าความแม่นยำและผลลัพธ์ของต้นไม้ตัดสินใจเมื่อทดสอบกับชุดข้อมูล โดยมีค่าความแม่นยำและค่า f1-score ที่ 1.00

1) ขั้นตอนการทดลองโดยใช้ต้นไม้ตัดสินใจด้วย Rapidminer Studio

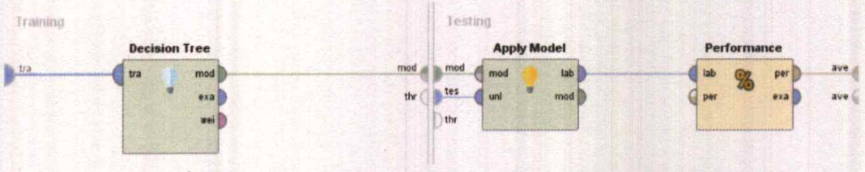
ในการใช้งาน Rapidminer Studio นั้นสามารถกำหนดโครงสร้างการประมวลผลข้อมูลในโปรแกรมได้เลย โดยขั้นตอนต่อไปนี้เป็นการสร้างการประมวลผลข้อมูล ดังรูปที่ 3.31



รูปที่ 3.31 โครงสร้างการประมวลผลข้อมูล

จากรูปที่ 3.31 แสดงโครงสร้างการประมวลผลข้อมูลประกอบไปด้วยคำสั่ง read csv คือการนำเข้าชุดข้อมูลจากไฟล์ csv ในส่วนของการเลือกคุณลักษณะและการกำหนดเซทคำตอบประกอบไปด้วย คำสั่ง select attributes คือการเลือกคุณลักษณะที่ใช้ในการประมวลผล คำสั่ง set role คือ

การกำหนดคลาสคำตอบในการทำนายผล และในส่วนถัดมาจะใช้คำสั่ง validation คือการแบ่งข้อมูลเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ



รูปที่ 3.32 โครงสร้างการฝึกสอนและทดสอบข้อมูล

จากรูปที่ 3.32 แสดงโครงสร้างการฝึกสอนและทดสอบข้อมูล โดยใช้ต้นไม้ตัดสินใจประกอบด้วย คำสั่ง decision tree คือวิธีที่ใช้ในการฝึกสอนข้อมูล สำหรับในการทดสอบโมเดลจะใช้คำสั่ง apply model คือการทดสอบข้อมูล และวัดประสิทธิภาพของโมเดลโดยใช้คำสั่ง performance คือการประมวลผลลัพธ์ของโมเดล

หลังจากการฝึกสอนและการทดสอบข้อมูลดังรูปที่ 3.32 ผลลัพธ์ที่ได้จากขั้นตอนนี้จะแสดงได้ดังรูปที่ 3.33

accuracy: 89.14%

	true high	true low	true S_low	true S_high	class precision
pred. high	327340	11783	7845	19236	89.39%
pred. low	115624	569289	18251	37296	76.88%
pred. S_low	3855	24328	1171483	21658	96.92%
pred. S_high	42484	21901	47421	982910	89.79%
class recall	66.90%	90.75%	94.10%	92.63%	

รูปที่ 3.33 ผลลัพธ์การใช้ต้นไม้ตัดสินใจ

จากรูปที่ 3.33 แสดงผลลัพธ์ของการใช้ต้นไม้ตัดสินใจเมื่อทดสอบกับชุดข้อมูลที่ได้แบ่งไว้ก่อนหน้า โดยมีค่าความแม่นยำที่ 89.14%

2) ขั้นการทดลองโดยใช้เนอีฟเบย์ด้วย Scikit-learn Framework

เนื่องจากในการเรียกใช้ Scikit-learn Framework นั้นไม่สามารถวิเคราะห์คุณลักษณะที่เป็นข้อความได้โดยตรง จึงต้องทำการแปลงข้อมูลนำเข้า

ที่เป็นตัวอักษรให้เป็นตัวเลข โดยจะใช้ฟังก์ชัน preprocess.LabelEncoder() ของ Scikit-learn Framework ซึ่งผลลัพธ์ของกระบวนการชุดนี้จะเป็นชุดตัวเลขที่แทนตัวอักษรในคุณลักษณะดังกล่าว ซึ่งคำสั่งที่ใช้แสดง ดังรูปที่ 3.20 และเรียกใช้ฟังก์ชันบางส่วนของ Scikit-learn Framework ดังรูปที่ 3.34

```
from sklearn.naive_bayes import GaussianNB
```

รูปที่ 3.34 คำสั่งเรียกใช้งานโมเดล

จากรูปที่ 3.34 แสดงคำสั่งเรียกใช้งานโมเดลที่จะนำไปสร้างโมเดลเนอีฟเบย์ประกอบไปด้วยคำสั่ง from sklearn.navie_bayes import GaussianNB คือการเรียกใช้โมเดล GaussianNB ในการฝึกสอนข้อมูล

ข้อมูลที่จะใช้ฝึกสอนและทดสอบโมเดลจะถูกแบ่งออกเป็นสองประเภทได้แก่ ส่วนข้อมูลนำเข้าและคลาสคำตอบโดยใช้คำสั่ง ดังรูปที่ 3.35

```
cols = ['from_address', 'to_address', 'value', 'balance',
        'open', 'high', 'low', 'close', 'volumefrom']
X = df[cols]
y = df['labels']
```

รูปที่ 3.35 คำสั่งแบ่งข้อมูลไปยังตัวแปร

จากรูปที่ 3.35 ตัวแปร X จะเก็บข้อมูลนำเข้า และตัวแปร y จะเก็บคลาสคำตอบที่จะใช้เพื่อการฝึกสอนและการทดสอบโมเดล

ในขั้นตอนต่อไปคือการแบ่งข้อมูลจากตัวแปร X และ y ให้เป็นชุดข้อมูลฝึกสอน 70% และชุดข้อมูลทดสอบ 30% และสุดท้าย คือ การสร้างโมเดลเนอีฟเบย์ ดังรูปที่ 3.36 และ 3.37

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3)
```

รูปที่ 3.36 คำสั่งแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

จากรูปที่ 3.36 ตัวแปร X และ y จะถูกแบ่งออกเป็นชุดข้อมูลฝึกสอน 70% เก็บไปที่ตัวแปร X_train กับ y_train ส่วนชุดข้อมูลทดสอบ 30% เก็บที่ตัวแปร X_test และ y_test

```
model = GaussianNB()
model.fit(X_train, y_train)
pred = model.predict(X_test)
```

รูปที่ 3.37 คำสั่งสร้างโมเดลเนอีฟเบย์

จากรูปที่ 3.37 เป็นตัวอย่างคำสั่งสร้างโมเดลเนอีฟเบย์และทดสอบโมเดล โดยผลลัพธ์ที่ได้แสดง ดังรูปที่ 3.38

```
from sklearn import metrics

print("Accuracy:",metrics.accuracy_score(y_test, pred))
```

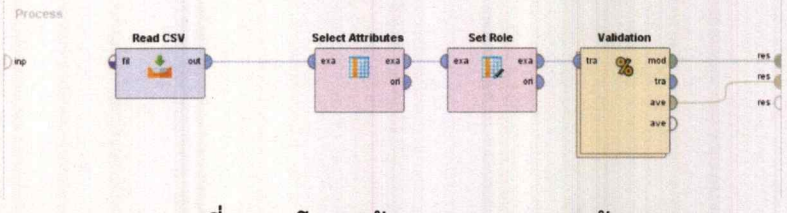
```
Accuracy: 0.4124133575952019
```

รูปที่ 3.38 คำสั่งแสดงผลลัพธ์ของโมเดล

จากรูปที่ 3.38 เป็นตัวอย่างการแสดงผลลัพธ์ของโมเดล ดังรูปที่ 3.37 โดยประกอบด้วยคำสั่ง ค่าความแม่นยำ หมายถึง ค่าความแม่นยำในการทดสอบของโมเดล โดยมีค่าความแม่นยำที่ 0.41

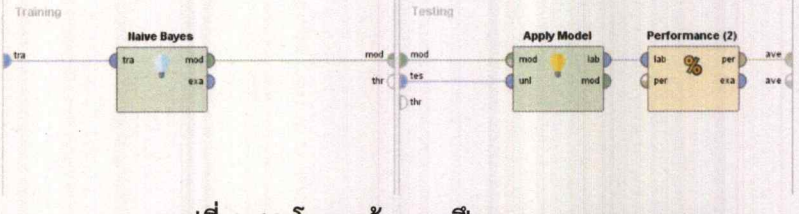
3) ขั้นการทดลองโดยใช้เนอีฟเบย์ด้วย Rapidminer Studio

การใช้งาน Rapidminer Studio นั้นสามารถกำหนดโครงสร้างการประมวลผลข้อมูลในโปรแกรมได้ โดยขั้นตอนต่อไปนี้เป็นการสร้างการประมวลผลข้อมูล แสดงดังรูปที่ 3.39



รูปที่ 3.39 โครงสร้างการประมวลผลข้อมูล

จากรูปที่ 3.39 แสดงการกำหนดโครงสร้างการประมวลผลข้อมูล ประกอบไปด้วยคำสั่ง read csv คือการนำเข้าชุดข้อมูลจากไฟล์ csv ในส่วนของการเลือกคุณลักษณะจะใช้คำสั่ง select attributes คือการเลือกคุณลักษณะที่ใช้ในการประมวลผล ในการกำหนดคลาสคำตอบจะคำสั่ง set role คือการกำหนดคลาสคำตอบสำหรับการทำนาย และส่วนของการฝึกสอนและทดสอบจะใช้คำสั่ง validation คือการแบ่งข้อมูลเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ



รูปที่ 3.40 โครงสร้างการฝึกสอนและการทดสอบ

จากรูปที่ 3.40 แสดงโครงสร้างการฝึกสอนและทดสอบข้อมูล โดยใช้วิธีเนอีฟเบย์ตัดสินใจประกอบด้วยคำสั่ง Naïve Bayes คือวิธีที่ใช้ในการฝึกสอนข้อมูล สำหรับการฝึกสอนข้อมูลจะใช้คำสั่ง apply model คือการทดสอบข้อมูล และวัดประสิทธิของโมเดลด้วยคำสั่ง performance คือการประมวลผลลัพธ์ของโมเดล

หลังจากที่ได้ฝึกสอนและทดสอบข้อมูลดังรูปที่ 3.40 ผลลัพธ์ที่ได้จากการฝึกสอนจะแสดงดังรูปที่ 3.41

accuracy: 90.26%

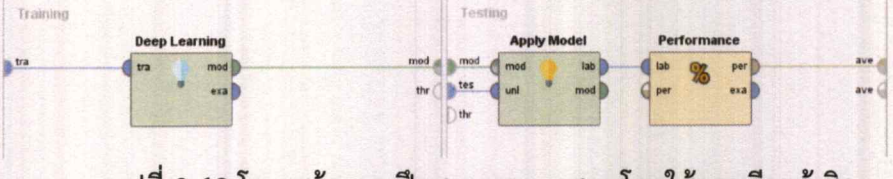
	true high	true low	true S_low	true S_high	class precision
pred. high	463257	43297	35483	38143	79.85%
pred. low	8415	540838	26749	25486	89.92%
pred. S_low	14982	32355	1146831	59128	91.51%
pred. S_high	2646	10811	35937	938343	95.00%
class recall	94.68%	86.22%	94.10%	88.43%	

รูปที่ 3.41 ผลลัพธ์การใช้เนอีฟเบย์

จากรูปที่ 3.41 เป็นผลลัพธ์การใช้เนอีฟเบย์เมื่อทดสอบกับชุดข้อมูลที่ได้แบ่งเอาไว้ดังรูปที่ 3.42

4) ขั้นการทดลองโดยใช้การเรียนรู้เชิงลึกด้วย Rapidminer Studio

ในการทดลองการเรียนรู้เชิงลึกด้วย Rapidminer Studio สามารถกำหนดโครงสร้างในการประมวลผลข้อมูลได้ โดยโครงการนำเข้าข้อมูลและการประมวลผลจะเหมือนดังรูปที่ 3.39 ซึ่งโครงสร้างการฝึกสอนและทดสอบโดยใช้การเรียนรู้เชิงลึกจะแสดงดังรูปที่ 3.42



รูปที่ 3.42 โครงสร้างการฝึกสอนและทดสอบโดยใช้การเรียนรู้เชิงลึก

จากรูปที่ 3.42 แสดงโครงสร้างการฝึกสอนและทดสอบข้อมูล โดยใช้ต้นไม้ตัดสินใจประกอบด้วยคำสั่ง Deep learning คือวิธีที่ใช้ในการฝึกสอนข้อมูล ในส่วนของการทดสอบข้อมูลใช้คำสั่ง apply model คือการทดสอบข้อมูล และในการประมวลผลใช้คำสั่ง performance คือการประมวลผลลัพธ์ของโมเดล

ผลลัพธ์ที่ได้จากการฝึกสอนและทดสอบที่กล่าวมาข้างต้นจะแสดงดังรูปที่ 3.43

accuracy: 97.53%

	true high	true low	true S_low	true S_high	class precision
pred. high	430830	13490	21	9732	94.88%
pred. low	57820	612927	1964	0	91.11%
pred. S_low	0	882	1243015	0	99.93%
pred. S_high	650	2	0	1051368	99.94%
class recall	88.05%	97.71%	99.84%	99.08%	

รูปที่ 3.43 ผลลัพธ์การใช้การเรียนรู้เชิงลึก

จากรูปที่ 3.43 เป็นผลลัพธ์ที่ได้จากการฝึกสอนและทดสอบโดยใช้การเรียนรู้เชิงลึก

3.2.3.3 ขั้นตอนวิธีการทดลองโดยใช้วิธีการวิเคราะห์การถดถอยของข้อมูล

ในการนำข้อมูลเข้าจะใช้ชุดข้อมูลจาก balance price_history_day และ transactions ดังที่กล่าวไปแล้วในหัวข้อ 3.2.2.1 โดยนำข้อมูลรวมกันเป็นชุดเดียวซึ่งตัวอย่างข้อมูลที่นำมาใช้งานมีดังนี้

1) ส่วนของข้อมูลนำเข้า

ข้อมูลของชุดข้อมูล balance price_history_day และ transactions ที่รวมกันเป็นชุดเดียวประกอบไปด้วยคุณลักษณะของข้อมูล 7 คุณลักษณะคือ value, balance, open, high, low, close และ volumefrom

	value	balance	open	high	low	close	volumefrom
0	1.028580e+17	0.0	5.22	5.41	4.50	5.20	21713.64
1	8.808750e+18	0.0	4.25	4.78	3.50	3.86	48722.06
2	7.754630e+17	0.0	5.60	6.00	5.32	5.70	29377.55
3	7.512680e+17	0.0	5.70	6.50	5.51	6.23	34361.90
4	1.006020e+18	0.0	6.23	6.64	5.55	5.93	48900.99

รูปที่ 3.44 ตัวอย่างข้อมูลในแต่ละคุณลักษณะ

จากรูปที่ 3.44 เป็นชุดตัวอย่างข้อมูลในแต่ละคุณลักษณะ ที่เกิดจากการรวมของชุดข้อมูล 3 ชุดคือ balance price_history_day และ transactions โดยเลือกคุณลักษณะที่เหมาะสมสำหรับการทดลอง

2) ส่วนของการประมวลผล

ในการประมวลผลจะปรับลดคุณลักษณะ เพื่อให้มีค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง รากที่สองของความคลาดเคลื่อนกำลังสอง และค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์มีค่าน้อยที่สุด

3) ส่วนของผลลัพธ์การคำนวณ

ผลลัพธ์ที่ได้จะนำไปเปรียบเทียบกับวิธีต่าง ๆ ของการทดลองจากในงานวิจัยนี้ เพื่อให้ทราบถึงวิธีที่เหมาะสมสำหรับการวิเคราะห์การถดถอยของข้อมูล โดยในการเขียนคำสั่งเพื่อคำนวณผลลัพธ์จะใช้ Scikit-learn Framework ทดสอบชุดข้อมูลดังที่กล่าวในรูปที่ 3.44

3.2.3.4 ขั้นตอนวิธีการทดลองโดยใช้การวิเคราะห์การถดถอยเชิงเส้นด้วย Scikit-learn Framework

ในหัวข้อนี้จะกล่าวถึงการใช้ Scikit-learn Framework ในการวิเคราะห์การถดถอยเชิงเส้นกับชุดข้อมูลที่ได้กล่าวไปในหัวข้อข้างต้น ในการวิเคราะห์ข้อมูลสามารถเลือกคุณลักษณะที่เป็นตัวเลขได้โดยตรง สำหรับการเรียกใช้ Scikit-learn Framework สามารถพิมพ์คำสั่งดังรูปที่ 3.45

```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.model_selection import train_test_split
```

รูปที่ 3.45 ตัวอย่างคำสั่งเรียกใช้ Scikit-learn Framework

จากรูปที่ 3.45 เป็นตัวคำสั่งสำหรับเรียกใช้งาน Scikit-learn Framework ที่ใช้สำหรับการวิเคราะห์การถดถอยเชิงเส้นประกอบไปด้วยคำสั่ง from sklearn.linear_model import LinearRegrssion คือการเรียกใช้โมเดล DecisionTreeClassifier ในการฝึกสอนข้อมูล คำสั่ง from sklearn import metrics คือการเรียกใช้มาตรวัดโมเดล และคำสั่ง from sklearn.model_selection import train_test_split คือการเรียกใช้ฟังก์ชันการแบ่งข้อมูล

ข้อมูลที่จะใช้ฝึกสอนและทดสอบโมเดลจะถูกแบ่งออกเป็นสองส่วนได้แก่ ส่วนข้อมูลนำเข้าและคลาสคำตอบโดยคำสั่งที่ใช้แสดงดังภาพที่ 3.46

```
X = df[['open','high','low','volumefrom']]
y = df['close']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

รูปที่ 3.46 คำสั่งเลือกคุณลักษณะไปยังตัวแปร

จากรูปที่ 3.46 ตัวแปร X จะเก็บข้อมูลนำเข้า และตัวแปร Y เก็บคำตอบที่ใช้สำหรับการฝึกสอนและการทดสอบโมเดล โดยแบ่งข้อมูลการฝึกสอน 70% เก็บไว้ในตัวแปร X_train กับ y_train และข้อมูลการทดสอบ 30% เก็บไว้ในตัวแปร X_test กับ y_test

```
# build the model using sklearn
lm = LinearRegression()
lm.fit(X_train,y_train)
```

รูปที่ 3.47 คำสั่งเรียกใช้การถดถอยเชิงเส้นและการสร้าง

จากรูปที่ 3.47 เป็นตัวอย่างการเรียกใช้และสร้างโมเดลการวิเคราะห์การถดถอยเชิงเส้น

หลังจากสร้างโมเดลดังรูปที่ 3.47 ในรูปถัดมาจะเป็นขั้นตอนการวัดประสิทธิภาพของโมเดลจะแสดงดังรูปที่ 3.48

```
r2 = lm.score(X_train,y_train)
predictions = lm.predict(X_test)
print("R-squared :",r2)
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
print(r2_score(y_test, predictions))
```

รูปที่ 3.48 คำสั่งแสดงผลลัพธ์ของโมเดล

จากรูปที่ 3.48 เป็นคำสั่งแสดงผลลัพธ์ของโมเดลที่ได้จากการทดสอบ โดยค่าของผลลัพธ์จะแสดงดังรูปที่ 3.49

```
R-squared : 0.9971671326809295
MAE: 10.153945886929971
MSE: 250.02315202982135
RMSE: 15.812120415359267
```

รูปที่ 3.49 แสดลงผลลัพธ์ของโมเดล

จากรูปที่ 3.49 เป็นการแสดงผลลัพธ์ของโมเดล โดยประกอบไปด้วย R-squared คือประสิทธิภาพของโมเดล สำหรับค่า MAE MSE และ RMSE คือค่าความผิดพลาดของโมเดลที่ใช้การวัดประสิทธิภาพของโมเดล

สำหรับการปรับลดคุณลักษณะที่ใช้สำหรับการสร้างโมเดลจะใช้การหาสหสัมพันธ์ของข้อมูลจะแสดงดังรูปที่ 3.50

```
corr = X.corr()
corr
```

รูปที่ 3.50 คำสั่งคำนวณค่าสหสัมพันธ์

จากรูปที่ 3.50 เป็นตัวอย่างการใช้คำสั่งการคำนวณค่าสหสัมพันธ์ของข้อมูล โดยใช้ฟังก์ชัน pandas.corr() ในการคำนวณค่าสหสัมพันธ์เพื่อสร้างตารางความสัมพันธ์เพื่อใช้ในการตัดสินใจเลือกคุณลักษณะที่เหมาะสมสำหรับการวิเคราะห์ข้อมูลซึ่งตารางสหสัมพันธ์จะแสดงดังรูปที่ 3.51

	open	high	low	volumefrom
open	1.000000	0.996562	0.987525	0.010483
high	0.996562	1.000000	0.988951	0.026703
low	0.987525	0.988951	1.000000	-0.074699
volumefrom	0.010483	0.026703	-0.074699	1.000000

รูปที่ 3.51 ตัวอย่างตารางสหสัมพันธ์

จากรูปที่ 3.51 เป็นตัวอย่างตารางแสดงค่าสหสัมพันธ์ของข้อมูล โดยการปรับลดคุณลักษณะของข้อมูลจะเลือกข้อมูลที่มีค่าสหสัมพันธ์ไม่เกิน 0.7

ในการทดลองการวิเคราะห์การถดถอยเชิงเส้นหลังจากการปรับลดคุณลักษณะที่กล่าวไปก่อนหน้านี้แล้ว จะนำคุณลักษณะที่ได้เลือกไว้ไปทดลอง ที่ได้กล่าวไปในขั้นตอนแรก และนำผลลัพธ์ที่ได้มาเปรียบเทียบประสิทธิภาพของโมเดลเพื่อหาโมเดลที่ดีที่สุด

3.2.3.5 ขั้นตอนวิธีการทดลองโดยใช้การหากฎความสัมพันธ์ของข้อมูลส่วนของข้อมูลนำเข้า

การทดลองในขั้นตอนนี้จะนำข้อมูลเข้าโดยใช้ชุดข้อมูลจาก balance price_history_day และ transactions ดังที่กล่าวไปแล้วในหัวข้อ 3.2.2.1 โดยนำข้อมูลรวมกันเป็นชุดเดียวซึ่งตัวอย่างข้อมูลที่นำมาใช้งานมีดังนี้

1) ส่วนของการประมวลผล

ข้อมูลของชุดข้อมูล balance price_history_day และ transactions ที่รวมกันเป็นชุดเดียวประกอบไปด้วยคุณลักษณะของข้อมูล 10 คุณลักษณะคือ

from_address, to_address, value, balance, status, open, high, low, close และ volumefrom

from_address	to_address	value
0x3886edb2a1678f1ace39146db3d5ed59a87a3a59	0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be	1.117390e+18
0xd24400ae8bfebb18ca49be86258a3c749cf46853	0x22859869c24cbc73e54e5cd170a2fc67cc478149	1.000000e+17
0xceceaa8edc0830c7cec497e33bb3a3c28dd55a32	0x2a0c0dbecc7e4d658f48e01e3fa353f44050c208	0.000000e+00
0xfbb1b73c4f0bda4f67dca266ce6ef42f520fbb98	0x78333643a97427fc485cf43e0e5fa824e1e9b5e3	0.000000e+00
0x932692889a3ae9b5db8c43cb6b7646ebee81a93e	0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be	2.096710e+18

รูปที่ 3.52 ตัวอย่างชุดข้อมูล

จากรูปที่ 3.52 เป็นตัวอย่างชุดข้อมูลบางส่วนที่ใช้สำหรับการหากฎความสัมพันธ์ของข้อมูล

2) ส่วนของการประมวลผล

ในการประมวลผลจะกำหนดค่าสนับสนุน และค่าความเชื่อมั่น เพื่อใช้ในการหากฎความสัมพันธ์ที่ถูกต้องมากยิ่งขึ้น

3) ส่วนของผลลัพธ์การคำนวณ

ผลลัพธ์ที่ได้จะนำไปสรุปเป็นกฎความสัมพันธ์ของข้อมูล โดยในการคำนวณหากฎความสัมพันธ์ของข้อมูลจะใช้ pandas Library ในการเตรียมข้อมูล และใช้ Rapidminer Studio ในการหากฎความสัมพันธ์ของชุดข้อมูล

3.2.3.6 ขั้นตอนวิธีการทดลองโดยใช้การหากฎความสัมพันธ์ของข้อมูลด้วย Rapidminer Studio

เนื่องจากในการเรียกใช้ Rapidminer Studio เพื่อหากฎความสัมพันธ์นั้นไม่สามารถวิเคราะห์คุณลักษณะที่เป็นข้อความได้โดยตรง จึงต้องทำการแปลงข้อมูลนำเข้าที่เป็นตัวอักษรให้เป็นวันฮ็อทเสียก่อนโดยจะใช้ฟังก์ชัน pandas.get_dummies() ของ Pandas library ผลลัพธ์ของกระบวนการชุดนี้คือชุดตัวเลขที่แทนตัวอักษรใน

คุณลักษณะดังกล่าว ซึ่งคำสั่งที่ใช้แสดงดังรูปที่ 3.21 และสำหรับการแบ่งช่วงของข้อมูลของคุณลักษณะต่าง ๆ จะแสดงคำสั่งที่ใช้ดังรูปที่ 3.53 และ 3.54

```
data['value'].describe()
```

```
Out[11]:
count    2.045020e+05
mean     2.195344e+18
std      3.266921e+18
min      0.000000e+00
25%      0.000000e+00
50%      4.995800e+17
75%      2.920498e+18
max      9.223370e+18
Name: value, dtype: float64
```

รูปที่ 3.53 คำสั่งแสดงรายละเอียดข้อมูล

จากรูปที่ 3.53 เป็นตัวอย่างคำสั่งแสดงรายละเอียดของคุณลักษณะ เพื่อพิจารณาข้อมูล

```
mean = data['value'].mean()
std = data['value'].std()
```

```
In [13]:
low = mean - std
high = mean + std
med = high - low
print(f"low={low}")
print(f"high={high}")
print(f"med={med}")
```

รูปที่ 3.54 คำสั่งแบ่งช่วงข้อมูล

จากรูปที่ 3.54 เป็นตัวอย่างคำสั่งการแบ่งช่วงข้อมูลโดยใช้ค่าที่ได้แสดงดังรูปที่ 3.54 ในการแบ่งช่วงข้อมูลสำหรับใช้ในการหากฎความสัมพันธ์

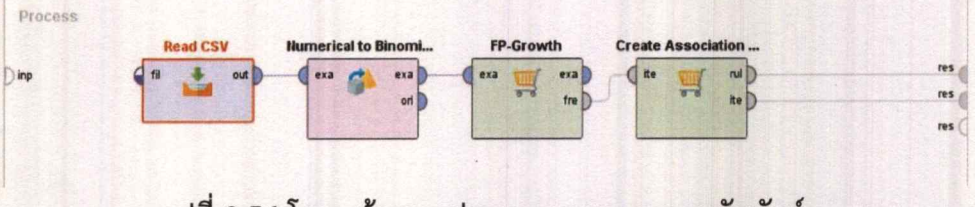
หลังจากแบ่งช่วงของข้อมูล ต่อมาจะเป็นขั้นตอนการแปลงข้อมูลเป็นวันฮ้อต เพื่อทำให้ข้อมูลสามารถนำไปหากฎความสัมพันธ์ได้โดยจะแสดงดังรูปที่ 3.55

```
data_ohe = pd.get_dummies(data['N_value'])
data = pd.concat([data,data_ohe],axis=1)
```

รูปที่ 3.55 การแปลงข้อมูลเป็นวันฮ็อต

จากรูปที่ 3.55 เป็นตัวอย่างการแปลงข้อมูลเป็นวันฮ็อตเพื่อใช้ในการหากฎความสัมพันธ์ของข้อมูล

สำหรับขั้นตอนต่อมา เป็นการนำข้อมูลที่ได้จากการเตรียมข้อมูลเพื่อนำไปหากฎความสัมพันธ์โดยใช้โปรแกรม Rapidminer Studio โดยโครงสร้างของการประมวลผลจะแสดงดังรูป 3.56



รูปที่ 3.56 โครงสร้างการประมวลผลหากฎความสัมพันธ์

จากรูปที่ 3.56 เป็นโครงสร้างการประมวลผลในการหากฎความสัมพันธ์ของข้อมูลประกอบไปด้วย คำสั่ง read csv คือการนำเข้าชุดข้อมูลจากไฟล์ csv หลังนำชุดข้อมูลเข้าโปรแกรมถัดมาเป็นการแปลงข้อมูลโดยใช้คำสั่ง numerical to binominal คือการแปลงข้อมูลตัวเลขให้เป็น Boolean เลือกใช้คำสั่ง fp-growth คือวิธีในการหากฎความสัมพันธ์ และใช้คำสั่ง create association rule คือการสร้างกฎความสัมพันธ์ของข้อมูล

โดยผลลัพธ์ที่ได้จากการหากฎความสัมพันธ์จะแสดงดังรูปที่ 3.57

AssociationRules

```
Association Rules
[volumefrom_med, S_low] --> [value_med] (confidence: 0.809)
[balance_med, volumefrom_med, S_low] --> [value_med] (confidence: 0.809)
[value_med, S_low] --> [balance_med, volumefrom_med] (confidence: 0.811)
[balance_med] --> [volumefrom_med] (confidence: 0.811)
[balance_med, S_low] --> [value_med] (confidence: 0.811)
[S_low] --> [value_med] (confidence: 0.811)
[S_low] --> [balance_med, volumefrom_med] (confidence: 0.813)
[value_med] --> [volumefrom_med] (confidence: 0.814)
[balance_med, value_med] --> [volumefrom_med] (confidence: 0.814)
[balance_med] --> [value_med] (confidence: 0.821)
[volumefrom_med] --> [value_med] (confidence: 0.824)
[balance_med, volumefrom_med] --> [value_med] (confidence: 0.824)
[balance_med, S_high] --> [value_med] (confidence: 0.825)
[S_high] --> [value_med] (confidence: 0.825)
[to_address_0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be] --> [balance_med, volumefrom_med] (confidence: 0.828)
[volumefrom_med, S_high] --> [value_med] (confidence: 0.831)
[value_med, S_low] --> [volumefrom_med] (confidence: 0.844)
[balance_med, value_med, S_low] --> [volumefrom_med] (confidence: 0.845)
[S_low] --> [volumefrom_med] (confidence: 0.846)
[balance_med, S_low] --> [volumefrom_med] (confidence: 0.847)
[balance_med, to_address_0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be] --> [volumefrom_med] (confidence: 0.862)
[to_address_0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be] --> [volumefrom_med] (confidence: 0.862)
[value_med, S_low] --> [balance_med] (confidence: 0.959)
[S_low] --> [balance_med] (confidence: 0.959)
[value_med] --> [balance_med] (confidence: 0.959)
[volumefrom_med] --> [balance_med] (confidence: 0.960)
[value_med, volumefrom_med] --> [balance_med] (confidence: 0.960)
[volumefrom_med, to_address_0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be] --> [balance_med] (confidence: 0.960)
[volumefrom_med, S_high] --> [balance_med] (confidence: 0.961)
[volumefrom_med, S_low] --> [balance_med] (confidence: 0.961)
[value_med, S_high] --> [balance_med] (confidence: 0.961)
[to_address_0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be] --> [balance_med] (confidence: 0.961)
[value_med, to_address_0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be] --> [balance_med] (confidence: 0.961)
[value_med, volumefrom_med, S_low] --> [balance_med] (confidence: 0.961)
[S_high] --> [balance_med] (confidence: 0.961)
```

รูปที่ 3.57 กฎความสัมพันธ์

จากรูปที่ 3.57 เป็นกฎความสัมพันธ์ของข้อมูลที่ได้จากขั้นตอนที่กล่าวไปข้างต้น

3.2.4 การประเมินผล

เป็นขั้นตอนการประเมินผลโมเดลที่ได้จากการทำเหมืองข้อมูล เพื่อวัดประสิทธิภาพของโมเดลสำหรับการนำไปใช้จริง หรือนำไปปรับปรุงแก้ไขโดยแบ่งออกเป็น 2 ส่วนคือ

3.2.4.1 การประเมินผลโมเดลสำหรับปัญหาการจำแนกประเภทข้อมูล

โดยจะใช้มาตรวัดประสิทธิภาพของโมเดลที่ใช้ในการจำแนกประเภทข้อมูลเช่น ค่าความถูกต้อง (Accuracy)

1) ค่าความถูกต้อง

เป็นมาตรวัดประสิทธิภาพของโมเดลที่นิยมใช้ในการจำแนกประเภทของข้อมูล

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{(สมการที่ 3.1)}$$

3.2.4.2 การประเมินผลโมเดลสำหรับปัญหาการวิเคราะห์การถดถอย

ใช้มาตรวัดประสิทธิภาพของโมเดลที่ใช้ในการวิเคราะห์การถดถอยของข้อมูล เช่น รากที่สองของความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Mean Squared Error) และค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์ (Mean Absolute Error)

1) รากที่สองของความคลาดเคลื่อนกำลังสอง

เป็นมาตรวัดประสิทธิภาพของโมเดลที่ใช้ในการวิเคราะห์การถดถอยของข้อมูล

$$RMSE = \sqrt{\frac{1}{n} * \sum(prediction - actual)^2} \quad \text{(สมการที่ 3.2)}$$

2) ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง

เป็นมาตรวัดประสิทธิภาพของโมเดลใช้สำหรับการวิเคราะห์การถดถอยของข้อมูล

$$MSE = \frac{1}{n} * \sum(prediction - actual)^2 \quad \text{(สมการที่ 3.3)}$$

3) ค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์

เป็นมาตรวัดประสิทธิภาพของโมเดลใช้สำหรับการวิเคราะห์การถดถอยของข้อมูล

$$MAE = \frac{1}{n} * \sum |prediction - actual| \quad \text{(สมการที่ 3.4)}$$

3.2.5 การปรับค่าพารามิเตอร์ต่าง ๆ

เป็นขั้นตอนการปรับค่าตัวแปรต่าง ๆ ที่มีผลต่อการหาความสัมพันธ์ของข้อมูล เพื่อให้โมเดลที่ได้จากการทำเหมืองข้อมูลมีประสิทธิภาพมากยิ่งขึ้น

บทที่ 4

ผลการดำเนินงานและการอภิปรายผล

ในบทนี้จะกล่าวถึง ผลการดำเนินงาน การอภิปรายผลการดำเนินงาน การทดสอบทางสถิติและปัญหาที่พบในการดำเนินงาน

4.1 ผลการดำเนินงาน

การดำเนินงานวิจัยนี้ได้ทำการแบ่งการทดลองออกเป็น 7 การทดลองประกอบไปด้วย

1) ใช้การวิเคราะห์การถดถอยในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework
2) ใช้ต้นไม้ตัดสินใจในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework
3) ใช้ต้นไม้ตัดสินใจในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio
4) ใช้วิธีเนอีฟเบย์ในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework
5) ใช้วิธีเนอีฟเบย์ในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio
6) ใช้การเรียนรู้เชิงลึกในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio
7) ใช้การหากฎความสัมพันธ์ในการหาความสัมพันธ์ของข้อมูลด้วย Rapidminer Studio

4.1.1 ใช้การวิเคราะห์การถดถอยในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework

ในการทดลองนี้จะนำการวิเคราะห์การถดถอยทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework เพื่อหารากที่สองของความคลาดเคลื่อนกำลังสอง ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง R-Squared และค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์ ซึ่งใช้เมทริกซ์สหสัมพันธ์ (Correlation Matrix) เพื่อเลือกคุณลักษณะที่เหมาะการสร้างแบบจำลอง และเพื่อป้องกันปัญหา Multicollinearity สำหรับการเลือกคุณลักษณะที่เหมาะสำหรับการสร้างแบบจำลอง โดยจะเลือกคุณลักษณะที่มีค่าสหสัมพันธ์น้อยกว่า 0.7 และข้อมูลนำซึ่งระกอบไปด้วย จำนวนเงินที่ส่ง (value), ยอดเงินคงเหลือ (balance), ค่าเงินเปิดตัวของสกุลเงินอีเธอเรียม (open), ค่าเงินสูงสุดของสกุลเงินอีเธอเรียม (high), ค่าต่ำสุดของสกุลเงินอีเธอเรียม (low) และปริมาณค่าเงินซื้อขายในมุมมองสกุลเงินดอลลาร์

(volumefrom) โดยคุณลักษณะดังกล่าวใช้เพื่อทำนายราคาปิดโดยได้ออกแบบการทดลองไว้เป็น 3 โมเดลดังนี้

1) **โมเดลที่ 1**

ข้อมูลนำเข้าทราใช้ในการสร้างโมเดลจะใช้คุณลักษณะทั้งหมดที่กล่าวไปก่อนหน้านี้ในการสร้างโมเดลเพื่อทำนายราคาปิด

2) **โมเดลที่ 2**

ข้อมูลนำเข้าที่ใช้ในการสร้างโมเดลจะใช้ค่าสหสัมพันธ์ในการเลือกคุณลักษณะเพื่อทำนายราคาปิดประกอบด้วย จำนวนเงินที่ส่ง , ยอดเงินคงเหลือ , ค่าเงินเปิดตัวของสกุลเงินอีเธอเรียม และปริมาณค่าเงินซื้อขายในมุมมองสกุลเงินดอลลาร์

3) **โมเดลที่ 3**

ข้อมูลนำเข้าที่ใช้ในการสร้างโมเดลโดยใช้ข้อมูลที่ไม่เกี่ยวข้องกับผู้ถือสกุลเงินอีเธอเรียมเพื่อทำนายราคาปิดประกอบด้วย ค่าเงินเปิดตัวของสกุลเงินอีเธอเรียม , ค่าเงินสูงสุดของสกุลเงินอีเธอเรียม , ค่าเงินต่ำสุดของสกุลเงินอีเธอเรียม และปริมาณค่าเงินซื้อขายในมุมมองสกุลเงินดอลลาร์

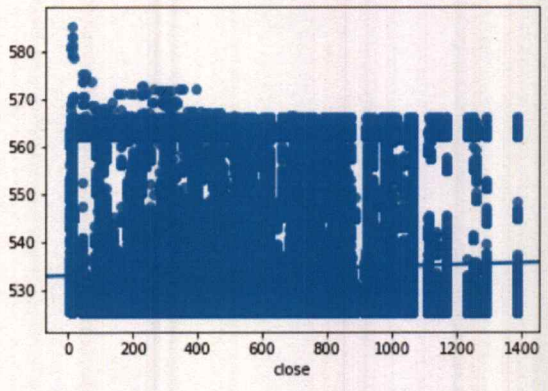
ก่อนนำข้อมูลเข้าเพื่อทดลองการวิเคราะห์การถดถอย ข้อมูลนำเข้าจะถูกแบ่งออกเป็นสองส่วน (Split validation) ส่วนหนึ่งเพื่อใช้สร้างแบบจำลอง (Model Training) จำนวน 70% และอีกส่วนเพื่อใช้ทดสอบแบบจำลอง (Model Testing) จำนวน 30%

โดยตารางสรุปผลการทดลองของการวิเคราะห์การถดถอย เมื่อทดสอบกับชุดข้อมูลนำเข้าที่ได้กล่าวไปข้างต้น คือ ข้อมูลนำเข้าที่ใช้ในการสร้างโมเดลก่อนใช้ค่าสหสัมพันธ์ในการเลือกคุณลักษณะและข้อมูลนำเข้าที่ใช้ในการสร้างโมเดลหลังใช้ค่าสหสัมพันธ์ในการเลือกคุณลักษณะ ซึ่งจากการทดลอง ค่ารากที่สองของความคลาดเคลื่อนกำลังสอง ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง และค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์ มีค่าที่ไม่แตกต่างกันมาก แต่ข้อมูลนำเข้าที่ใช้ในการสร้างโมเดลโดยใช้ข้อมูลที่ไม่เกี่ยวข้องกับผู้ถือสกุลอีเธอเรียม มีค่าทดลอง รากที่สองของความคลาดเคลื่อนกำลังสอง ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง และค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์ที่ต่ำ ซึ่งมีค่า R-Squared เท่ากับ 0.997 โดยจะแสดงค่าดังกล่าวในตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองของการวิเคราะห์การถดถอย

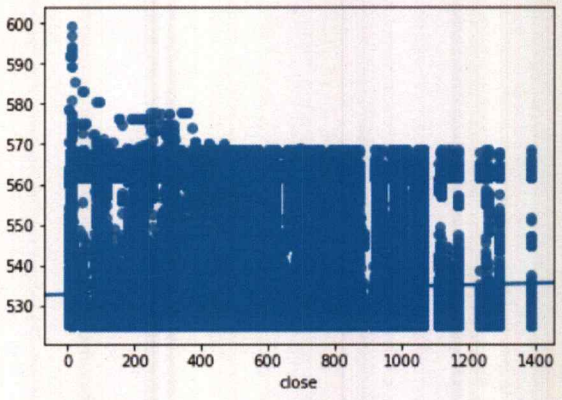
MODEL	R-Squared	MAE	MSE	RMSE
1	0.001	246.38	88041.84	296.71
2	0.001	246.24	87939.89	296.54
3	0.997	10.14	249.49	15.79

จากตารางที่ 4.1 ค่า R-Squared และค่ามาตรวัดอื่น ๆ มีค่าไม่แตกต่างกันในโมเดลที่ 1 และ 2 จะสังเกตได้ว่าโมเดลที่ 3 มีค่า R-Squared รากที่สองของความคลาดเคลื่อนกำลังสอง ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง และค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์ที่ค่อนข้างสูงแตกต่างจากโมเดลที่ 1 และ 2



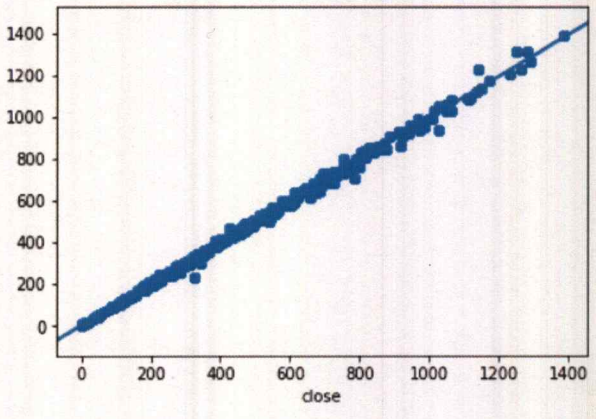
รูปที่ 4.1 กราฟเส้นตรงของโมเดลที่ 1

จากรูปที่ 4.1 สังเกตได้ว่ากราฟเส้นตรงของโมเดลที่ 1 ข้อมูลมีการกระจายตัวมากทำให้ไม่สามารถระบุแนวโน้มของข้อมูลได้ โดยที่แกน x เป็นค่าที่แบ่งไว้สำหรับทดสอบและแกน y เป็นค่าที่ทำนายได้ซึ่งแสดงถึงแนวโน้มของราคาปิดตัวไปในทิศทางที่เป็นบวก



รูปที่ 4.2 กราฟเส้นตรงของโมเดลที่ 2

จากรูปที่ 4.2 สังเกตได้ว่ากราฟเส้นตรงของโมเดลที่ 2 ข้อมูลมีการกระจายตัวมากทำให้ไม่สามารถระบุแนวโน้มของข้อมูลได้เช่นเดียวกับโมเดลที่ 1 ดังรูปที่ 4.1 โดยที่แกน x เป็นค่าที่แบ่งไว้สำหรับทดสอบและแกน y เป็นค่าที่ทำนายได้ซึ่งแสดงถึงแนวโน้มของราคาปิดตัวไปในทิศทางที่เป็นบวก



รูปที่ 4.3 กราฟเส้นตรงของโมเดลที่ 3

จากรูปที่ 4.3 กราฟเส้นตรงแสดงผลการทำนายของโมเดลที่ 3 โดยที่แกน x เป็นค่าที่แบ่งไว้สำหรับทดสอบและแกน y เป็นค่าที่ทำนายได้ซึ่งแสดงถึงแนวโน้มของราคาปิดตัวไปในทิศทางที่เป็นบวก

4.1.2 ใช้ต้นไม้ตัดสินใจในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework

ในการทดลองนี้ใช้ต้นไม้ตัดสินใจในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework เพื่อหาค่าความแม่นยำ และจัดกลุ่มสถานะราคาอีเธอเรียม โดยได้ออกแบบการทดลองไว้ดังนี้

1) กำหนดชุดคำตอบให้ "ผลลัพธ์ = S_high" เมื่อสถานะราคาอีเธอเรียมมีค่าสูงมาก "ผลลัพธ์ = high" เมื่อผลลัพธ์ราคาอีเธอเรียมมีค่าสูง "ผลลัพธ์ = low" เมื่อสถานะราคาอีเธอเรียมมีค่าต่ำ และ"ผลลัพธ์ = S_low" เมื่อสถานะราคาอีเธอเรียมมีค่าต่ำมาก
2) สำหรับข้อมูลการนำเข้าที่ใช้ต้นไม้ตัดสินใจประกอบด้วย หมายเลขผบัญชีผู้ส่ง (from_address) , หมายเลขบัญชีผู้รับ (to_address) , เวลาของค่าเงิน ณ เวลานั้น (date) , จำนวนเงินที่ส่ง , ยอดเงินคงเหลือ , ราคาเปิดตัวของสกุลเงินอีเธอเรียม , ค่าเงินสูงสุดของสกุลเงินอีเธอเรียม , ค่าเงินต่ำสุดของสกุลเงินอีเธอเรียม , ค่าเงินซื้อขายในมุมมองสกุลเงินดอลลาร์
3) การทดลองข้อมูลแบ่งออกเป็นสองส่วนได้แก่ แบบที่ 1: กำหนด criterion เป็น gini index และ แบบที่ 2: กำหนด criterion เป็น information gain
4) แปลงข้อมูลที่เป็นตัวอักษรให้เป็นตัวเลขด้วยฟังก์ชัน LabelEncoder ดังที่กล่าวไปในบทที่ 3 (หัวข้อที่ 3.2.2.2)

ในการทดลอง ข้อมูลทั้งหมดถูกแบ่งออกเป็นสองส่วน ส่วนหนึ่งใช้เพื่อสร้างแบบจำลองจำนวน 70% และอีกส่วนเพื่อใช้ทดสอบแบบจำลองที่ถูกสร้างขึ้น 30% และใช้มาตรวัด 2 อย่างในการวัดประสิทธิภาพได้แก่ ค่าความแม่นยำ และค่าเฉลี่ยประสิทธิภาพโดยรวม

ตารางที่ 4.2 ค่าความแม่นยำของวิธีต้นไม้ตัดสินใจด้วย Scikit-learn Framework

Criterion	Accuracy	F1-Score
Gini index	1.00	1.00
Information gain	1.00	1.00

จากตารางที่ 4.2 ค่าความแม่นยำของทั้งสามแบบ มีค่าไม่แตกต่างกันที่ 1.00 โดยทั้งสองวิธีได้ลดคุณลักษณะที่คาดว่าไม่สำคัญออกแล้วค่าที่ได้ก็ยังไม่แตกต่างจากเดิม ซึ่งอาจเกิดจากการที่คุณลักษณะของข้อมูลยังไม่เพียงพอต่อการทดลอง

รูปที่ 4.4 ต้นไม้ตัดสินใจ

จากรูปที่ 4.4 แสดงต้นไม้ตัดสินใจที่ได้จากการทดลองของ criterion gini index โดยใช้ Scikit-learn Framework

4.1.3 ใช้ต้นไม้ตัดสินใจในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio

ในการทดลองนี้จะกล่าวถึงการใช้ต้นไม้ตัดสินใจโดยใช้ Rapidminer Studio ซึ่งข้อมูลทั้งหมดถูกแบ่งออกเป็นสองส่วน ส่วนหนึ่งใช้เพื่อสร้างแบบจำลองจำนวน 70% และอีกส่วนเพื่อใช้ทดสอบแบบจำลองที่ถูกสร้างขึ้น 30% และใช้มาตรวัด 2 อย่างในการวัดประสิทธิภาพได้แก่ ค่าความแม่นยำ และค่าเฉลี่ยประสิทธิภาพโดยรวม

ในการทดลองจะแบ่งออกเป็นสองส่วนได้แก่ โมเดลที่ใช้ Criterion gini index กับโมเดลที่ใช้ Criterion information gain และใช้มาตรวัดประสิทธิภาพได้แก่ ค่าแม่นยำของข้อมูล โดยผลการทดลองจะแสดงในตารางที่ 4.3

ตารางที่ 4.3 ผลการทดลองโดยใช้ต้นไม้ตัดสินใจด้วย Rapidminer Studio

Criterion	Accuracy
Gini index	0.884
Information gain	0.891

จากตารางที่ 4.3 สังเกตได้ว่าการทดลองโดยใช้ต้นไม้ตัดสินใจด้วย Rapidminer Studio ที่ใช้ Criterion เป็น information gain มีค่าความแม่นยำ 0.891 ซึ่งมีค่าสูงกว่าโมเดลที่ใช้ Criterion เป็น Gini index

4.1.4 ใช้วิธีเนอีฟเบย์ในการทดสอบกับชุดข้อมูลด้วย Scikit-learn Framework

สำหรับในการทดลองนี้ใช้วิธีเนอีฟเบย์ในการทดสอบชุดข้อมูลด้วย Scikit-learn Framework เพื่อหาค่าความแม่นยำ โดยได้ออกแบบการทดลองไว้ดังนี้

1) สำหรับข้อมูลการนำไปใช้ในวิธีเนอีฟเบย์ประกอบด้วย หมายเลขผบัญชีผู้ส่ง, หมายเลขบัญชีผู้รับ, จำนวนเงินเงินที่ส่ง, ยอดเงินคงเหลือ, ราคาเปิดตัวของสกุลเงินอีเธอเรียม, ค่าเงินสูงสุดของสกุลเงินอีเธอเรียม, ค่าเงินต่ำสุดของสกุลเงินอีเธอเรียม และค่าเงินซื้อขายในมุมมองสกุลเงินดอลลาร์
2) ในการแปลงข้อมูลที่เป็นตัวอักษรให้เป็นตัวเลขด้วยฟังก์ชัน LabelEncoder ดังที่กล่าวไปในบทที่ 3 (หัวข้อที่ 3.2.2.2)
3) การทดลองแบ่งออกเป็นสองส่วนได้แก่

   แบบที่ 1: ใช้ข้อมูลทั้งหมดโดยไม่ปรับใด ๆ

   แบบที่ 2: ไม่เลือกใช้คุณลักษณะ เวลาของค่าเงิน ณ เวลานั้น

   แบบที่ 3: ไม่เลือกใช้คุณลักษณะ เวลาของค่าเงิน ณ เวลานั้น หมายเลขบัญชีผู้ส่ง และหมายเลขบัญชีผู้รับ

   แบบที่ 4: ปรับค่าของข้อมูลจากแบบที่ 2 และแบบที่ 5: ปรับค่าของข้อมูลจากแบบที่ 3
4) ข้อมูลทั้งหมดถูกแบ่งออกเป็นสองส่วน ส่วนที่หนึ่งใช้เพื่อสร้างแบบจำลองจำนวน 70% และอีกส่วนเพื่อใช้ทดสอบแบบจำลองที่ถูกสร้างขึ้น 30% และใช้ค่าความแม่นยำ ในการวัดประสิทธิภาพของข้อมูล

ตารางที่ 4.4 ผลการทดลองโดยใช้วิธีเนอีฟเบย์ด้วย Scikit-learn Framework

Model	Accuracy
1	0.36
2	0.36
3	0.35
4	0.41
5	0.41

จากตารางที่ 4.4 ค่าความแม่นยำมีค่าเพิ่มขึ้นเมื่อมีการปรับค่าของข้อมูลโดยโมเดลที่ 4และ 5 มีค่าความแม่นยำที่มากที่สุดอยู่ที่ 0.41

4.1.5 ใช้วิธีเนอีฟเบย์ในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio

ในการทดลองนี้จะกล่าวถึงการใช้วิธีเนอีฟเบย์ในการทดสอบข้อมูลโดยใช้ Rapidminer Studio ซึ่งข้อมูลทั้งหมดจะถูกแบ่งออกเป็นสองส่วน ส่วนหนึ่งใช้เพื่อสร้างแบบจำลองจำนวน 70% และอีกส่วนเพื่อใช้ทดสอบแบบจำลองที่ถูกสร้างขึ้น 30% และใช้ค่าความแม่นยำในการวัดประสิทธิภาพของข้อมูล โดยการทดสอบข้อมูลจะถูกแบ่งออกเป็นสามส่วนได้แก่

แบบที่ 1: ใช้คุณลักษณะทั้งหมดในการทดสอบ

แบบที่ 2: ไม่เลือกใช้คุณลักษณะ เวลาของค่าเงิน ณ เวลานั้น

แบบที่ 3: ไม่เลือกใช้คุณลักษณะ เวลาของค่าเงิน ณ เวลานั้น หมายเลขบัญชีผู้ส่ง และหมายเลขบัญชีผู้รับ

ตารางที่ 4.5 ผลการทดลองโดยใช้วิธีเนอีฟเบย์ด้วย Rapidminer Studio

Model	Accuracy
1	0.90
2	0.90
3	0.41

จากตารางที่ 4.5 ค่าความแม่นยำที่ได้จากการทดลองโดยใช้วิธีเนอีฟเบย์ด้วย Rapidminer Studio มีค่าความแม่นยำที่ 0.90 สังเกตได้ว่ามีค่าความแม่นยำสูงกว่าการทดลองโดยใช้วิธีเนอีกเบย์ด้วย Scikit-learn Framework ดังตารางที่ 4.4

4.1.6 ใช้การเรียนรู้เชิงลึกในการทดสอบกับชุดข้อมูลด้วย Rapidminer Studio

ในการทดลองนี้จะนำข้อมูลชุดเดียวกับการทดลองในหัวข้อมูล 4.1.1 ทดสอบกับวิธีการเรียนรู้เชิงลึก โดยได้ออกแบบการทดลองนี้ดังนี้ ข้อมูลทั้งหมดจะถูกแบ่งออกเป็นสองส่วนได้แก่ ส่วนที่หนึ่งสำหรับการฝึกสอนจำนวน 70% และอีกส่วนสำหรับการทดสอบ 30% โดยใช้ค่าความแม่นยำในการวัดประสิทธิภาพของข้อมูล และการทดสอบข้อมูลจะถูกแบ่งออกเป็นสามส่วนได้แก่ แบบที่ 1: ใช้คุณลักษณะทั้งหมดในการทดสอบ , แบบที่ 2: ไม่เลือกใช้คุณลักษณะ เวลาของค่าเงิน ณ เวลานั้น และแบบที่ 3: ไม่เลือกใช้คุณลักษณะ เวลาของค่าเงิน ณ เวลานั้น หมายเลขบัญชีผู้ส่ง และหมายเลขบัญชีผู้รับ

ตารางที่ 4.6 ผลการทดลองโดยใช้วิธีการเรียนรู้เชิงลึกด้วย Rapidminer Studio

Model	Accuracy
1	0.99
2	0.99
3	0.98

จากตารางที่ 4.6 สังเกตได้ว่าเมื่อลดคุณลักษณะลงทำให้ค่าความแม่นยำมีค่าที่ต่ำลงไม่มาก และค่าความแม่นยำที่มีค่าสูงมากจนผิดปกติ

4.1.7 ใช้การหากฎความสัมพันธ์ในการหาความสัมพันธ์ของข้อมูลด้วย Rapidminer Studio

ในการทดลองนี้ได้มีแปลงข้อมูลที่เป็นตัวอักษรให้เป็นวันฮ้อท ดังที่กล่าวไปในบทที่ 3 (หัวข้อที่ 3.2.2.2) ทำให้จำนวนคุณลักษณะมีจำนวน 2,004 คุณลักษณะ แต่เนื่องจากข้อจำกัดด้านทรัพยากรในการประมวลผล ดังนั้นในการทดลองนี้จะสุ่มใช้ชุดข้อมูลจากชุดข้อมูลที่ถูกแปลงเป็นวันฮ้อตดังกล่าว จำนวน 50% ในการประมวลผล

ตารางที่ 4.7 ผลการทดลองหากฎความสัมพันธ์ของข้อมูลด้วย Rapidminer Studio

ลำดับ	กฎ	ค่าสนับสนุน	ค่าความเชื่อมั่น	ค่าลิฟท์
1	balance_med → volumefrom_med **ถ้า** เงินในบัญชีอยู่ในช่วงปานกลาง **แล้ว** ปริมาณซื้อขายมีความน่าจะเป็นที่จะอยู่ในช่วงปานกลาง	0.778	0.811	1.000
2	value_med → volumefrom_med **ถ้า** จำนวนเงินที่ส่งอยู่ในช่วงปานกลาง **แล้ว** ปริมาณซื้อขายมีความน่าจะเป็นที่จะอยู่ในช่วงปานกลาง	0.668	0.814	1.003
3	balance_med, value_med → volumefrom_med **ถ้า** เงินในบัญชีอยู่ในช่วงปานกลาง และจำนวนเงินที่ส่งอยู่ในช่วงปานกลาง **แล้ว** ปริมาณซื้อขายมีความน่าจะเป็นที่จะอยู่ในช่วงปานกลาง	0.641	0.814	1.003
4	balance_med → value_med **ถ้า** เงินในบัญชีอยู่ในช่วงปานกลาง **แล้ว** จำนวนเงินมีความน่าจะเป็นที่ส่งอยู่ในช่วงปานกลาง	0.788	0.821	1.000
5	volumefrom_med → value_med **ถ้า** ปริมาณซื้อขายมีอยู่ในช่วงปานกลาง **แล้ว** จำนวนเงินมีความน่าจะเป็นที่ส่งอยู่ในช่วงปานกลาง	0.668	0.824	1.003

ลำดับ	กฎ	ค่าสนับสนุน	ค่าความเชื่อมั่น	ค่าลิฟท์
6	balance_med, volumefrom_med → value_med **ถ้า** เงินในบัญชีอยู่ในช่วงปานกลาง และปริมาณซื้อขายมีอยู่ในช่วงปานกลาง **แล้ว** จำนวนเงินที่ส่งมีความน่าจะเป็นที่ส่งอยู่ในช่วงปานกลาง	0.641	0.824	1.003
7	value_med → balance_med **ถ้า** จำนวนเงินที่ส่งอยู่ในช่วงปานกลาง **แล้ว** เงินในบัญชีมีความน่าจะเป็นที่จะอยู่ในช่วงปานกลาง	0.788	0.959	1.000
8	volumefrom_med → balance_med **ถ้า** ปริมาณซื้อขายมีอยู่ในช่วงปานกลาง **แล้ว** เงินในบัญชีมีความน่าจะเป็นที่จะอยู่ในช่วงปานกลาง	0.778	0.960	1.000
9	value_med, volumefrom_med → balance_med **ถ้า** จำนวนเงินที่ส่งอยู่ในช่วงปานกลาง และ ปริมาณซื้อขายมีอยู่ในช่วงปานกลาง **แล้ว** เงินในบัญชีมีความน่าจะเป็นที่จะอยู่ในช่วงปานกลาง	0.641	0.960	1.000

จากตารางที่ 4.7 กฎความสัมพันธ์ได้จากการทดลองมี 9 กฎ โดยมีค่าสนับสนุนอยู่ที่ 0.5 และค่าความเชื่อมั่น 0.8 และใช้เวลาในการสร้างกฎความสัมพันธ์ 45 นาที่ 25 วินาที

4.2 การอภิปรายผลการดำเนินงาน

จากการทดลองในหัวข้อที่ 4.1.1 ถึง 4.1.7 เราจะนำผลลัพธ์ที่ได้มาเปรียบเทียบและภิปรายในรายละเอียด ซึ่งวิธีที่นำมาเปรียบเทียบประกอบไปด้วย ผลการทดลองการจำแนกข้อมูลในหัวข้อ 4.1.2 ถึง 4.1.6 สำหรับการเปรียบเทียบผลการทดลองโดยจะแบ่งออกเป็นสองส่วนได้แก่ การเปรียบเทียบค่าความแม่นยำที่ทดลองด้วย Scikit-learn Framework และการเปรียบเทียบค่าความแม่นยำที่ทดลองด้วย Rapidminer Studio โดยทั้งสองส่วนจะกล่าวถึงในหัวข้อที่ 4.2.1 ถึง 4.2.2

4.2.1 การเปรียบเทียบค่าความแม่นยำที่ทดลองด้วย Scikit-learn Framework

เนื่องจากการทดลองโดยใช้การเรียนรู้เชิงลึก ไม่สามารถใช้ Scikit-learn Framework ในการทดลองได้ การเปรียบเทียบค่าความแม่นยำของวิธีต่าง ๆ จึงเปรียบเทียบได้เพียบหัวข้อที่ 4.1.2 และ 4.1.3 ซึ่งจะได้ผลลัพธ์ตามตารางที่ 4.8

ตารางที่ 4.8 การเปรียบเทียบค่าความแม่นยำของวิธีการต่าง ๆ จากการทดลองด้วย Scikit-learn Framework

Accuracy	
Decision tree	Naïve bayes
<u>1.00</u>	0.41

จากตารางที่ 4.8 สังเกตได้ว่าค่าความแม่นยำที่ดีที่สุดคือ 1.00 ได้จากการทดลองด้วย Scikit-learn Framework โดยใช้ต้นไม้ตัดสินใจ

4.2.2 การเปรียบเทียบค่าความแม่นยำที่ทดลองด้วย Rapidminer Studio

จากการทดลองในหัวข้อที่ 4.1.3 4.1.5 และ 4.1.6 ค่าความแม่นยำที่ดีที่สุดคือ ซึ่งได้มาจากวิธี และเมื่อนำมาเปรียบเทียบค่าความแม่นยำที่ดีที่สุดของวิธีต่าง ๆ จะได้ผลลัพธ์ตามตารางที่ 4.9

ตารางที่ 4.9 การเปรียบเทียบค่าความแม่นยำของวิธีการต่าง ๆ จากการทดลองด้วย Rapidminer Studio

Accuracy		
Decision tree	Naïve bayes	Deep learning
0.891	0.90	**0.99**

จากตารางที่ 4.9 พบว่าการทดลองที่ใช้การเรียนรู้เชิงลึกมีค่าความแม่นยำที่ดีที่สุดคือ 0.99 เมื่อเปรียบเทียบกับการทดลองด้วยวิธีการต่าง ๆ

4.3 ปัญหาที่พบในการดำเนินงาน

4.3.1 ข้อจำกัดทางทรัพยากร

เนื่องจากจำนวนชุดข้อมูล ที่นำมาใช้ประกอบไปด้วยข้อมูลการแลกเปลี่ยนเงิน ยอดเงินคงเหลือของผู้ถือสกุล และราคาของสกุลเงินอีเธอเรียมจำนวน 11,409,004 รายการ เป็นข้อมูลประเภทข้อความ และตัวอักษร ซึ่งส่งผลต่อเวลา และหน่วยความจำในการประมวลผล เนื่องจากทรัพยากรที่มีจำกัด ในการทดลองจึงจำเป็นต้องเลือกใช้วิธีทดลองบางวิธีที่สามารถทดลองได้จากข้อจำกัดดังกล่าว และในหัวข้อที่ 4.1.7 จึงใช้วิธีการลดจำนวนข้อมูลที่นำมาทดลอง โดยการสุ่มข้อมูล 50% ของข้อมูลทั้งหมดก่อนที่จะนำชุดข้อมูลมาทดลองในขั้นตอนต่อไป

4.3.2 การประมวลผลข้อมูลประเภทข้อความ

ในการประมวลผลข้อมูลประเภทข้อความไม่สามารถใช้ข้อมูลที่เป็นตัวอักษรกับการทดลองด้วย Scikit-learn Framework ได้โดยตรง เนื่องจากการทำงานของ Scikit-learn Framework ต้องใช้ข้อมูลนำเข้าประเภทตัวเลขเท่านั้น จึงต้องทำการแปลงข้อมูลนำเข้าที่เป็นตัวอักษรให้เป็นตัวเลขเสียก่อน โดยจะใช้คำสั่งการเตรียมข้อมูล LabelEncoder ของ Scikit-learn Framework ก่อนที่จะนำชุดข้อมูลไปประมวลผลต่อไป

4.3.3 ชุดข้อมูลที่นำมาทดลองเป็นคนละประเภท

ชุดข้อมูลที่จะนำมาใช้กับการวิเคราะห์การถดถอย และการหากฎความสัมพันธ์ของข้อมูลในการทดลองต้องเป็นข้อมูลประเภทเดียวกัน เนื่องด้วยชุดข้อมูลที่นำมาทดสอบมีข้อมูล

ประเภทข้อความ และตัวเลข จึงต้องแปลงชนิดของข้อมูลก่อนที่จะนำชุดข้อมูลมาดำเนินงานในขั้นตอนต่อไป

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงกการสรุปผลการดำเนินงานและข้อเสนอแนะ ซึ่งการสรุปผลการดำเนินงานจะถูกนำเสนอในหัวข้อ 5.1 และข้อเสนอแนะจะถูกนำเสนอในหัวข้อ 5.2

5.1 สรุปผลการดำเนินงาน

ปัญหาพิเศษนี้นำเสนอการศึกษาผลกระทบของผู้ถืออีเธอเรียมรายใหญ่ที่ส่งผลกับราคาอีเธอเรียม โดยใช้เทคนิคการวิเคราะห์การถดถอย เทคนิคต้นไม้ตัดสินใจ เทคนิคเนอีฟเบย์ เทคนิคการเรียนรู้เชิงลึก และเทคนิคการหากฎความสัมพันธ์ ซึ่งช่วยในการหาความสัมพันธ์ของข้อมูล และวิเคราะห์แนวโน้มของราคาอีเธอเรียมได้ โดยแบ่งการทดลองเป็นสองแบบคือ การทดลองด้วย Scikit-learn Framework และการทดลองด้วย Rapidminer Studio จากการทดลองพบว่าการทดลองโดยใช้การเรียนรู้เชิงลึกด้วย Rapidminer Studio มีประสิทธิภาพดีกว่า เนื่องจาก Scikit-learn Framework ต้องปรับพารามิเตอร์และปรับข้อมูลบางส่วนซึ่งอาจทำให้ข้อมูลเกิดความผิดพลาดได้ ทั้งนี้ในการทดลองในแต่ละวิธีนั้นใช้เวลาทดลองที่แตกต่างกันเนื่องจากข้อมูลที่มีชุดขนาดใหญ่ และประเภทข้อมูลนำเข้าในแต่ละวิธีแตกต่างกันจึงทำให้ใช้เวลาในการทดลองแต่ละวิธีแตกต่างกันเช่น การทดลองจำแนกกลุ่มข้อมูลโดยใช้วิธีเนอีฟเบย์ การทดลองโดยใช้การเรียนรู้เชิงลึก และการทดลองโดยใช้ต้นไม้ตัดสินใจ ใช้เวลาในการทดลองที่แตกต่างกันมากโดยการเรียนรู้เชิงลึกใช้เวลาในการทดลองมากที่สุด แต่สังเกตได้ว่าผลลัพธ์ที่ได้จากการเรียนเรียนรู้เชิงลึกให้ประสิทธิภาพที่ดีกว่าเมื่อใช้ชุดข้อมูลที่มีขนาดเท่ากันทั้งหมดในการทดลอง สำหรับประสิทธิภาพในการทดลองของแต่ละวิธีเกิดจากการปรับค่าพารามิเตอร์ต่าง ๆ ในแต่ละวิธีเพื่อเพิ่มประสิทธิภาพของโมเดลให้มีความแม่นยำมากยิ่งขึ้น และนำผลลัพธ์ที่ได้จากการปรับค่าพารามิเตอร์ดังกล่าวมาเปรียบเทียบเพื่อหาวิธีที่ดีที่สุดในการทดลอง แต่เนื่องด้วยข้อจำกัดด้านทรัพยากรจึงไม่สามารถทดลองโดยใช้เทคนิคในการจำแนกข้อมูลบางเทคนิค และข้อมูลทั้งหมดในการหากฎความสัมพันธ์ได้

5.2 ข้อเสนอแนะ

จากการดำเนินงานของปัญหาพิเศษนี้ทำให้ทราบทิศทางงานวิจัยที่สามารถต่อยอดได้ในอนาคต โดยทางทีมวิจัยได้นำเสนอเกี่ยวกับการเพิ่มคุณลักษณะ (Attributes) อื่นที่มีความหลากหลายมากยิ่งขึ้น เช่น ข่าวของการเปิดตัวสกุลเงินอื่น ๆ หรือข่าวแนวโน้มของราคา รวมถึงการวิจัยสกุลเงินเหรียญอื่น ๆ ของอีเธอเรียม และการใช้ Cloud Computing Services.หรือบริการประมวลผลโดยใช้คลาด์ ในการประมวลผลข้อมูลเช่น ใช้ Google Cloud Platform ในการประมวลผลเพื่อช่วยการทำงานมีประสิทธิภาพมากขึ้น

บรรณานุกรม

"Ethereum json-rpc for python" [Online]. https://github.com/ConsenSys/ethjsonrpc (16 January 2019)

"Crytocompare api" [Online].https://min-api.cryptocompare.com/ (18 January 2019)

"Google bigquery cloud" [Online]. https://cloud.google.com/bigquery/docs/ (23 January 2019)

"Understanding ethereum and smart contact " [Online]. https://nuuneoi.com/blog.php?read_id=939&fbclid=IwAR2YvxyZCD0uIbyVD8OUaDPA0sslzgNPHBwkLkx9ptqU_VrwpJPJK7UDhjc /. (16 January 2019)

"What is Ethereum" [Online]. https://siamblockchain.com/ethereum-%e0%b8%84%e0%0%b8%ad%e0%b8%ad%e0%b8%b0%e0%b9%84%e0%b8%a3/?%2F/. (20 January 2019)

"Rapidminer Studio" [Online].https://rapidminer.com (13 March 2019)

"Scikit-learn Framework" [Online].https://scikit-learn.org/ stable/modules/classes.html (5 March 2019)

"Evaluation metrics" [Online].https://datarockie.com/2019/03/30/top-ten-machine-learning-metrics/ (5 March 2019)

"Machine Learning with python" [Online].https://medium.com/@sirawit0676/machine-learning-01-data-preprocessing-python-coding-basic-687aee03c478 (5 March 2019)

"Decision Tree for Machine Learing" [Online]. https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956/. (7 March 2019)

"Data Preprocessing for Machine Learing" [Online]. https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d (7 March 2019)

"Blockchain" [Online].https://medium.com/dcen/blockchain-pow-system-7a722f4b8261 (9 March 2019)