



## รายงานฉบับสมบูรณ์

การทำนายราคาบิตคอยน์ (BTC) แบบ Real-time

โดยใช้ เทคนิคการวิเคราะห์ Time Series และ Machine Learning

### จัดทำโดย

อริสรา อินทแสง 6610412001

พรไพลิน ไทยสุริโย 6610412006

จักร์พันธ์ นิ่มทอง 6610412008

### เสนอ

ผู้ช่วยศาสตราจารย์ ดร.เอกรัฐ รัฐกาญจน์

รายงานฉบับนี้เป็นส่วนหนึ่งรายวิชาการวิเคราะห์ข้อมูล

ณ ขณะปัจจุบันและกระแสดำเนินการของข้อมูล รหัสรายวิชา DADS6005

คณะสถิติประยุกต์ สาขาการวิเคราะห์ข้อมูลและวิทยาการข้อมูล (DADS)

สถาบันบัณฑิตพัฒนบริหารศาสตร์

ปีการศึกษา 1 / 2567

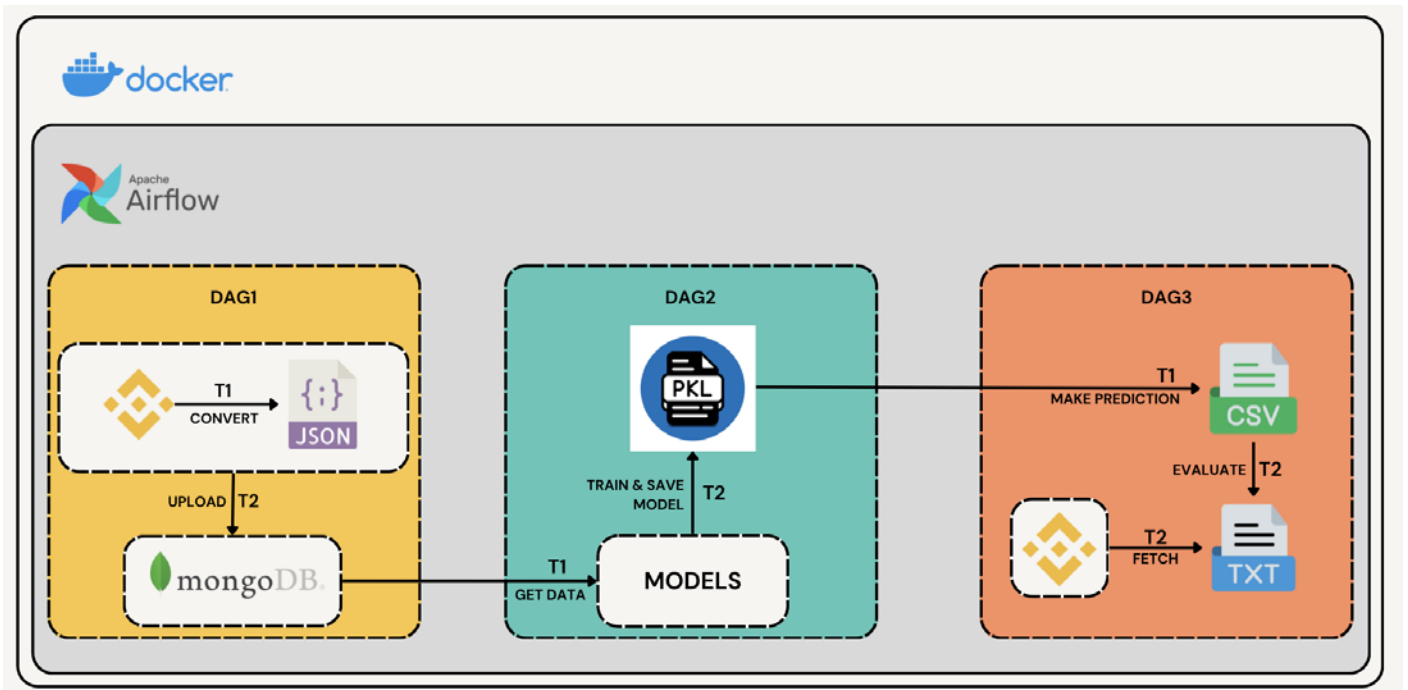
## Introduction

ในปัจจุบัน การพยากรณ์ราคาของ Bitcoin (BTC) แบบเรียลไทม์เป็นความท้าทายสำคัญในยุคที่ข้อมูลมีความซับซ้อนและเปลี่ยนแปลงอย่างรวดเร็ว ด้วยตลาดคริปโตเคอเรนซีมีลักษณะการเคลื่อนไหวที่ไม่สามารถคาดเดาได้ง่าย เนื่องจากมีความผันผวนสูงและได้รับผลกระทบจากปัจจัยหลายประการ เช่น สถานะเศรษฐกิจ, ข่าวสาร, และกิจกรรมของนักลงทุนทั่วโลก ส่งผลให้ราคาของสกุลเงินดิจิทัลสามารถเปลี่ยนแปลงได้อย่างรวดเร็วภายในเวลาสั้น ๆ นำไปสู่การที่นักลงทุนต้องเผชิญกับความเสี่ยงและความไม่มั่นคง ฉะนั้น การพยากรณ์ราคาของ Bitcoin จึงเป็นโจทย์ท้าทายในการวิเคราะห์ข้อมูลที่มีความซับซ้อนสูงและการใช้เทคนิคที่เหมาะสมในการสร้างโมเดลที่สามารถทำนายราคาที่ผันผวนในระยะสั้นได้อย่างแม่นยำ

โปรเจกต์นี้มีเป้าหมายในการพยากรณ์ราคาของ Bitcoin (BTC) แบบเรียลไทม์ โดยดึงข้อมูลจาก API ของ Binance ตามเวลาที่กำหนด และบันทึกลงใน MongoDB เพื่อใช้ในการพยากรณ์ราคา Bitcoin ซึ่งจะทำ การพยากรณ์ราคาทุก ๆ 5 นาที ของแต่ละวันต่อเนื่องกันเป็นจำนวน 3 วัน ดังนี้ วันที่ 21 , 22 , และ 23 ธันวาคม 2567 ในช่วงเวลา 22.00 น. ถึง 06.00 น. โดยมีการปรับปรุงโมเดลและกระบวนการในแต่ละวันเพื่อลดค่า MAPE (Mean Absolute Percentage Error) ซึ่งเป็นตัวชี้วัดประสิทธิภาพการพยากรณ์ และความแม่นยำของโมเดลที่ใช้ในการทำนายราคา ด้วยวัตถุประสงค์หลักเพื่อให้สามารถตัดสินใจลงทุนได้อย่างมีประสิทธิภาพในตลาดที่มีความผันผวนสูง

ในโปรเจกต์ครั้งนี้ ได้ทดลองใช้โมเดลต่าง ๆ ได้แก่ ARIMA, Random Forest Regressor (RF) และ Extra Trees Regressor (ET) เพื่อตรวจสอบว่าโมเดลใดสามารถทำงานได้ดีที่สุดในเงื่อนไขที่มีการเปลี่ยนแปลงของข้อมูลอย่างรวดเร็วและมีความไม่เสถียรสูงอย่าง Bitcoin ทั้งนี้ การทดลองใช้โมเดลเหล่านี้จะช่วยให้สามารถเลือกโมเดลที่มีความแม่นยำสูงสุดในการพยากรณ์ราคาของ Bitcoin เพื่อสนับสนุนการตัดสินใจทางธุรกิจและการลงทุนได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

## Methodology



จากรูปคือแผนภาพการทำงานโปรแกรมทำนายราคา BTCUSDT โดยทั้งหมดทำงานอยู่ภายใน Docker Container ทำงานไปกับ Apache Airflow ช่วยให้สามารถใช้งาน Directed Acyclic Graph (DAG) หลายตัว ทำให้สามารถวิเคราะห์ข้อมูลที่ไหลเข้ามาแบบ Real Time ได้อย่างมีประสิทธิภาพ

โปรแกรมของเราจะมี 3 DAG โดยแต่ละ DAG มีหน้าที่ดังนี้

1. DAG1 มีหน้าที่ในการดึงข้อมูลราคา BTCUSDT จาก Binance API แล้ว convert ให้เป็นไฟล์สกุล json จากนั้น upload ขึ้น mongoDB
2. DAG2 มีหน้าที่ ดึงข้อมูลที่ ต้องใช้ในการ train model จาก mongoDB หลังจาก train เสร็จเรียบร้อยแล้วจะมีการเซฟ model ดังกล่าว เป็นไฟล์สกุล pkl
3. DAG3 มีหน้าที่ดึง pkl ที่ผ่านการ train มาทำการ predict ราคา BTCUSDT จากนั้นเซฟใส่ไฟล์ csv จากนั้นจะดึงราคา BTCUSDT จริงจาก Binance API เพื่อมาเทียบกับ แล้วทำการคำนวณค่า MAPE เก็บใส่ไฟล์ txt

DAG2 ของกลุ่มเราได้มี 3 รูปแบบ เพื่อใช้สำหรับทดลองโมเดลต่างๆ ที่กลุ่มเราสนใจ โดยโมเดลที่กลุ่มเราสนใจ ได้แก่ ARIMA, Random Forest Regressor และ Extra Tree Regressor รายละเอียดของแต่ละโมเดล และผลการทดลอง เราจะอธิบายในบทถัดไป

## Model Results

### 1. ARIMA (Autoregressive Integrated Moving Average)

กลุ่มของเราเลือกใช้ ARIMA เป็นหนึ่งในโมเดลสำหรับทดสอบการทำนายราคา BTCUSDT ด้วยเหตุผลว่าเงื่อนไขในการทำนายสำหรับรายงานนี้ ถูกตั้งไว้ให้ใช้ ตัวแปรต้น(x) คือ วันและเวลา ส่วนตัวแปรตาม(y) คือ ระดับราคาของ BTCUSDT ด้วยความถี่ระดับนาที่เท่านั้น ซึ่งเป็นข้อมูลประเภทอนุกรมเวลา (time series) เราจึงได้ให้สมมติฐานว่า ราคา BTC ในอนาคตจะถูกผลกระทบจากราคา BTC ในอดีต

ข้อดีของ ARIMA คือ ARIMA เป็นหนึ่งในกลุ่ม Moving Average ที่มีความสามารถในการจัดการพวกความผันผวน (Noise) หรือข้อผิดพลาด (Errors) ต่างๆ ได้ดี และมีความยืดหยุ่นสูง เพราะสามารถปรับค่า parameter ได้ โดยสูตรการคำนวณ ARIMA คือ

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q} + \epsilon_t$$

โดย ARIMA มีค่า parameter ให้ปรับอยู่ 3 ตัวแปรดังนี้

1. p คือส่วนของ Autoregressive เป็นตัวแปรที่ใช้กำหนด lags ว่าต้องการใช้ข้อมูลย้อนหลังกี่ period เพื่อนำไปใช้ในการพยากรณ์ y ในกรณีนี้คือระดับราคาในแต่ละ 1 นาที่

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p}$$

2. d คือส่วนของ Integrated เป็นตัวแปรที่ใช้กำหนดจำนวนครั้งที่ทำการ Differential ให้ข้อมูลเป็น Stationary

$$y'_t = y_t - y_{t-1}$$

3. q คือส่วนของ Moving Average เป็นตัวแปรที่ใช้กำหนด lags ว่าต้องการใช้ค่าผิดพลาด (Errors) ย้อนหลังกี่ period

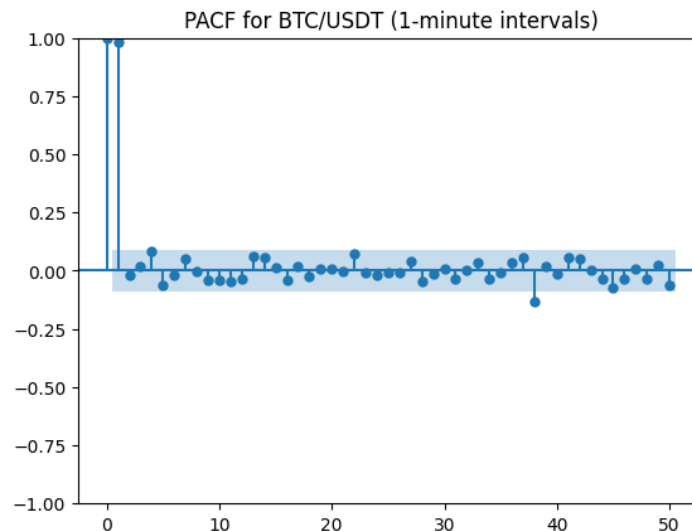
$$y_t = c - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q} + \epsilon_t$$

การ train model ของ ARIMA จะไม่ได้ใช้ Moving Absolute Percentage Error (MAPE) ในการ train model แต่เป็น Maximum Likelihood Estimation (MLE) แทน ซึ่งจะมีความซับซ้อนในการคำนวณมากกว่า MLE มีสูตรคำนวณดังนี้

$$L(\theta, \phi, \sigma^2 | y) = \prod_{t=1}^T f(y_t | \text{past data, parameters})$$

การกำหนดค่าของตัวแปร p, d, q ของกลุ่มเรา ได้จากการวิเคราะห์ ดังนี้

1. ใช้ Partial Autocorrelation Function (PACF) ในการช่วยกำหนดค่า p (Autoregressive)



ค่า Lag ที่ 1 และ 2 มีค่าสูงมาก เกินกรอบของ confidence level (พื้นที่สีฟ้าอ่อน) จึงตัดสินใจให้  $p = 2$

2. เราใช้ Augmented Dickey-Fuller (ADF) ช่วยกำหนดค่า d (Integrated)

```
--- Augmented Dickey-Fuller Test ---  
ADF Statistic: -1.0269895486109608  
p-value: 0.7432936614860095  
Data is not stationary. Differencing is needed.
```

ได้ตั้งสมมติฐานในการทดสอบ ดังนี้

$H_0$ : ข้อมูลไม่เป็น Stationary (มีแนวโน้มหรือความผันผวนที่ไม่คงที่)

$H_1$ : ข้อมูลเป็น Stationary (ไม่มีแนวโน้มหรือความผันผวนคงที่)

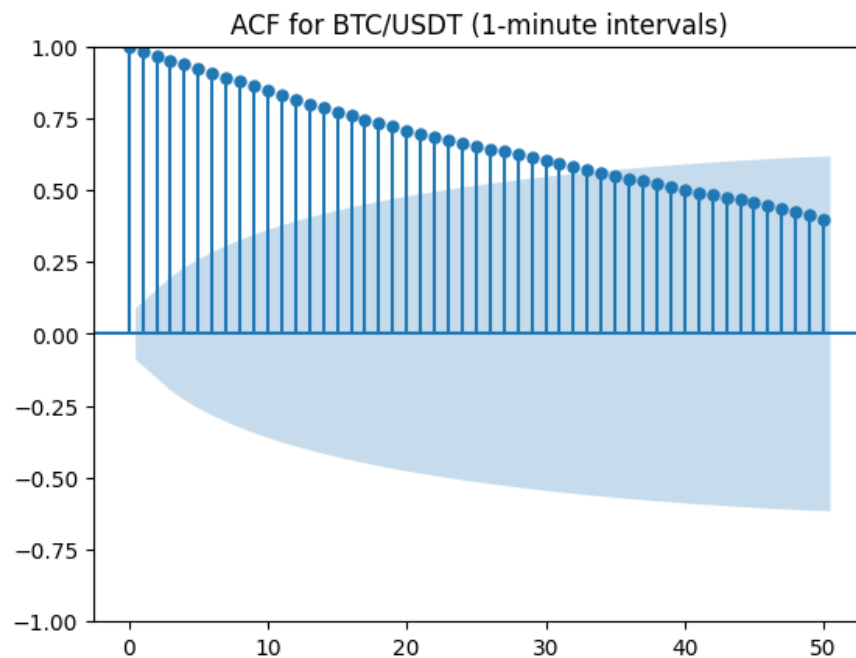
โดยผลที่ได้คือ ค่า  $p\text{-value} > 0.05$  จึงไม่สามารถ reject null hypothesis ได้ และยอมรับว่าข้อมูลที่มีความผันผวนไม่คงที่ จึงตั้งให้ค่า  $d = 1$

ทดสอบ ADF อีกรอบด้วยการ differencing 1 ครั้ง

```
ADF Statistic: -10.965432950179345
p-value: 8.12623777004688e-20
Data is stationary.
```

ผลที่ได้คือข้อมูลมีความผันผวนคงที่แล้วจึงตัดสินใจใช้ค่า  $d = 1$

3. ใช้ Autocorrelation Function (ACF) ช่วยกำหนดค่า  $q$  (Moving Average)



เนื่องจากค่า ACF มีการลดลงอย่างช้าๆ และต่อเนื่อง จึงไม่จำเป็นต้องมีส่วนของ Moving Average กลุ่มของเราจึงตั้งค่าให้  $q = 0$

ดังนั้น ใน code ของกลุ่มเราในส่วนของ การ train model ARIMA จึงเป็นดังนี้:

```
model = ARIMA(series, order=(2, 1, 0))
```

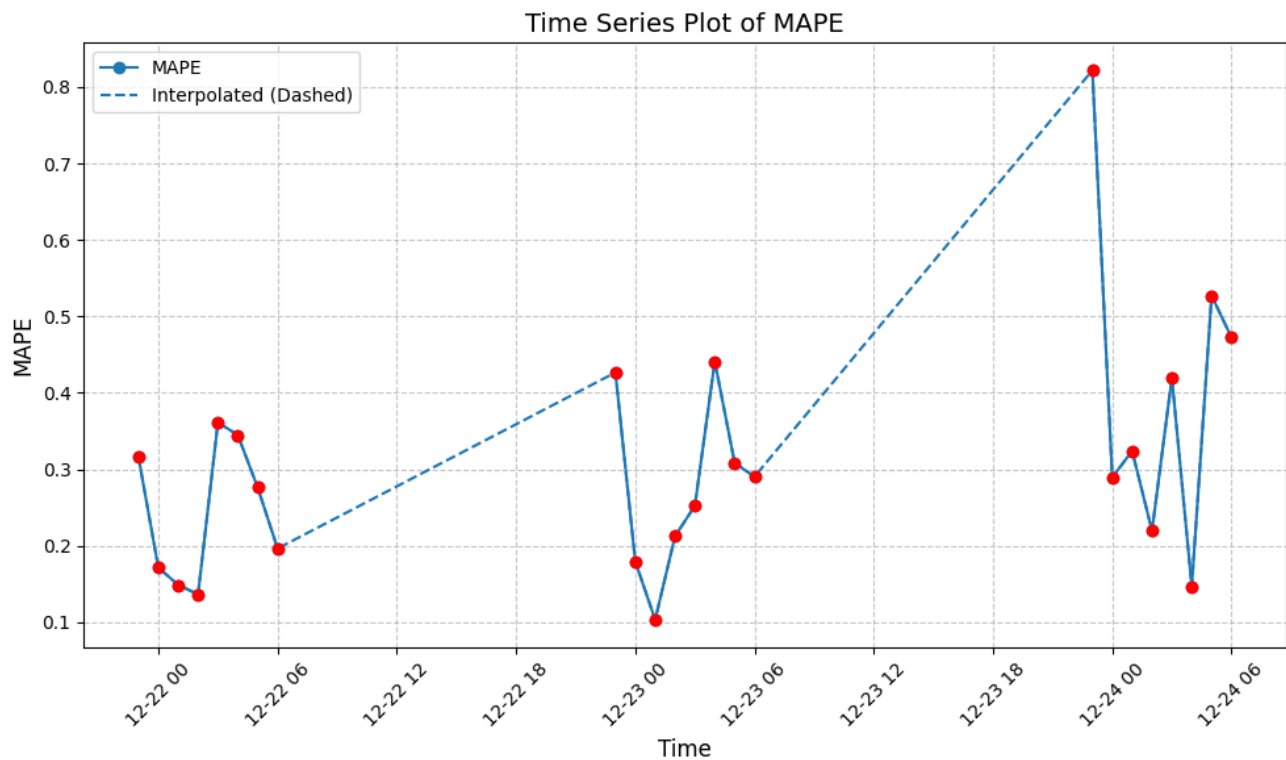
ในการทดสอบทั้งสามวัน โมเดล ARIMA ไม่ได้มีการปรับเปลี่ยน code เพิ่มเติม เนื่องจาก performance ของโมเดล ที่สังเกตจากค่า MAPE นั้น อยู่ในระดับที่ดีมาก เมื่อเปรียบเทียบกับโมเดลอื่นที่รันพร้อมกัน จึงเลือกใช้เวลาในการปรับปรุงโมเดลอื่นแทน

### Output :

วันที่	23:00	24:00	01:00	02:00	03:00	04:00	05:00	06:00	MAPE เฉลี่ย
21/12	0.3162	0.1714	0.1486	0.1364	0.3613	0.3447	0.2761	0.1963	22.82%
22/12	0.4266	0.1793	0.1031	0.2124	0.2522	0.4412	0.3081	0.2904	27.66%
23/12	0.8219	0.2892	0.3240	0.2195	0.4196	0.1469	0.5271	0.4723	40.26%

จากตารางข้างต้น ทำให้ได้ข้อสรุป ดังนี้:

วันที่ 21 ธันวาคม ค่า MAPE ลดลงต่อเนื่องจนถึงช่วง 02:00 น. ซึ่งมีค่าต่ำสุดที่ 0.1364 และเพิ่มขึ้นในช่วงเวลา 03:00 น. ถึง 05:00 น. ก่อนจะลดลงเล็กน้อยในเวลา 06:00 น. สำหรับวันที่ 22 ธันวาคม ค่า MAPE เริ่มต้นสูงขึ้นจากวันก่อนหน้า และหลังจากเวลา 01:00 น. ค่า MAPE มีความผันผวนเพิ่มขึ้น โดยเฉพาะเวลา 04:00 น. (0.4412) ท้ายที่สุด วันที่ 23 ธันวาคม ค่า MAPE มีค่าที่สูงขึ้นอย่างชัดเจนซึ่งเป็นค่าสูงที่สุดในทั้ง 3 วัน แม้ว่าจะลดลงในบางช่วง เช่น เวลา 04:00 น. แต่ในภาพรวม ค่า MAPE ยังคงอยู่ในระดับสูง





## 2. Random Forest Regressor

Random Forest Regressor (RF) เป็นอีกหนึ่งโมเดลที่ใช้งานง่ายและมีความสามารถในการทำงานกับข้อมูลที่มีความผันผวน (Noise) สูง เช่น ราคาของ Bitcoin (BTC) จากการกระจายการเรียนรู้ในหลายต้นไม้ ซึ่งช่วยลดการ overfitting และเพิ่มความแม่นยำในการทำนาย โดยสูตรการคำนวณ Random Forest Regressor จากการตัดสินใจผ่านการเฉลี่ยผลลัพธ์จากทุก decision trees คือ

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N f(T_i(X))$$

การฝึกโมเดล Random Forest Regressor ในการพยากรณ์ราคาของ Bitcoin แบบเรียลไทม์ช่วงวันที่ 21-23 ธันวาคม 2567 มีการปรับปรุงโมเดลเพื่อพยายามลดค่า MAPE ดังนี้

### วันที่ 1: การสร้างโมเดล Random Forest Regressor ตั้งต้น

- ข้อมูลถูกดึงจาก API ของ Binance ทุก 5 นาที ผ่าน DAG “Get\_BTC\_and\_Save\_to\_MongoDB” และบันทึกลงใน MongoDB เพื่อใช้ในการฝึกโมเดล Random Forest Regressor ด้วยฟีเจอร์ closeTime สำหรับการพยากรณ์ราคาปิด (lastPrice) โดยตั้งค่าพารามิเตอร์พื้นฐาน n\_estimators = 100 และ random\_state = 42
- ข้อเสีย:** การใช้พารามิเตอร์เริ่มต้นอาจไม่เหมาะสมซึ่งส่งผลต่อประสิทธิภาพของโมเดล

### วันที่ 2: การปรับปรุงโมเดลด้วย GridSearchCV

- พยายามปรับปรุงประสิทธิภาพของโมเดล Random Forest Regressor โดยการค้นหาพารามิเตอร์ที่เหมาะสมที่สุดผ่าน GridSearchCV มีชุดพารามิเตอร์ที่ทดลอง ได้แก่ n\_estimators: [50, 100, 200], max\_depth: [None, 10, 20, 30] และ min\_samples\_split: [2, 5, 10] รวมถึงการใช้ Cross-Validation แบบ 5-Fold เพื่อเลือกพารามิเตอร์ที่ลดค่า Error ได้ดีที่สุด
- ข้อเสีย:** ใช้เวลาในการประมวลผลมากขึ้นเมื่อเทียบกับวันที่ 1 และค่า MAPE ที่ได้ไม่ลดลงจากวันแรก

### วันที่ 3: ปรับกลับไปใช้โมเดลพื้นฐาน และเพิ่มความถี่การดึงข้อมูล

- ปรับปรุงกระบวนการดึงข้อมูล โดยเพิ่มความถี่ในการดึงข้อมูลจาก ทุก 5 นาที เป็น ทุก 1 นาที เพื่อให้โมเดลได้รับข้อมูลที่ละเอียดขึ้น และกลับมาใช้โมเดล Random Forest Regressor ที่ตั้งค่าพื้นฐานของวันที่ 1 แทน

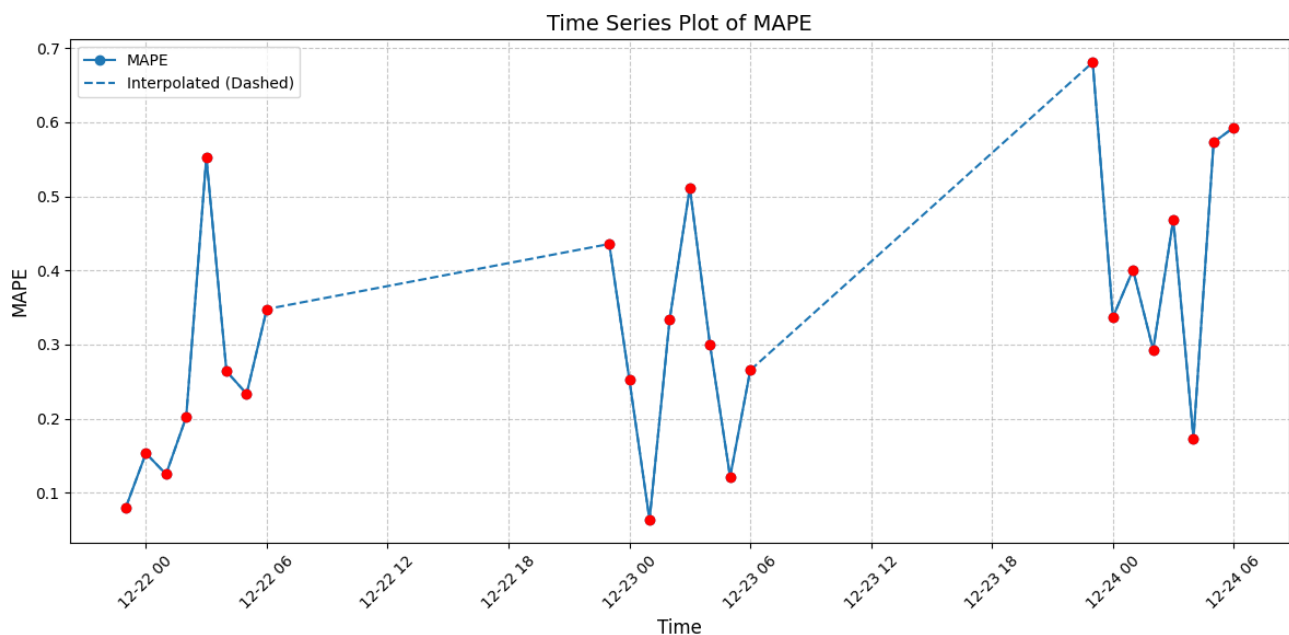
- **ข้อดี:** ข้อมูลที่ละเอียดขึ้นช่วยให้โมเดลเรียนรู้ข้อมูลราคา Bitcoin ที่เปลี่ยนแปลงเร็วได้ดีขึ้น รวมถึงลดความซับซ้อนของโมเดลและกระบวนการฝึก ทว่าการเพิ่มความถี่ในการดึงข้อมูลเพิ่มภาระให้กับเซิร์ฟเวอร์

### Output :

วันที่	23:00	24:00	01:00	02:00	03:00	04:00	05:00	06:00	MAPE เฉลี่ย
21/12	0.0798	0.1531	0.1248	0.2021	0.5525	0.2637	0.2334	0.3478	24.46%
22/12	0.4356	0.2521	0.0633	0.3340	0.5116	0.2996	0.1213	0.2652	28.53%
23/12	0.6808	0.3367	0.4007	0.2927	0.4687	0.1728	0.5728	0.5930	43.98%

จากตารางข้างต้น ทำให้ได้ข้อสรุป ดังนี้:

ในวันที่ 21 ธันวาคมและ 22 ธันวาคม มีค่าความคลาดเคลื่อน (MAPE) สูงสุดที่เวลา 03:00 (0.5525 และ 0.5116 ตามลำดับ) ทว่าในวันที่ 21 ธันวาคม ค่า MAPE ต่ำสุดอยู่ที่เวลา 23:00 (0.0798) ขณะที่ในวันที่ 22 ธันวาคม ค่า MAPE ต่ำสุดอยู่ที่เวลา 01:00 (0.0633) ส่วนในวันที่ 23 ธันวาคม ค่า MAPE สูงขึ้นอย่างเห็นได้ชัดในทุกช่วงเวลา โดยสูงสุดที่เวลา 23:00 (0.6808) ทั้งนี้ สังเกตได้ว่าค่า MAPE เพิ่มสูงขึ้นเรื่อยๆ จากวันที่ 21 ไปจนถึงวันที่ 23 และเพิ่มสูงขึ้นที่สุดในช่วงเวลา 03:00 น. ถึง 06:00 น.



### 3. Extra Tree Regressor

เป็นโมเดลที่ใช้ การตัดสินใจในรูปแบบของต้นไม้ (decision trees) โดยจะมีการสุ่มเลือกคุณลักษณะ ที่ช่วยลดการ overfitting และ มีความแม่นยำในการทำนาย โดยการคำนวณของ Extra Trees Regressor จะใช้จากแต่ละการตัดสินใจแล้วเฉลี่ยผลลัพธ์ ซึ่งสูตรที่ใช้เป็นดังนี้ :

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

ซึ่งการฝึกโมเดลโดยใช้ Extra Tree Regressor เนื่องจาก ข้อมูล Bitcoin (BTC) มีความผันผวนสูง ซึ่ง Extra Trees Regressor สามารถจัดการกับความไม่เป็นเส้นตรงนี้ได้ดี และมีการปรับปรุงโมเดลแต่ละวันของค่า MAPE ดังนี้

#### วันที่ 1 : การสร้างโมเดล Extra Tree Regressor

- ดึงข้อมูลจาก MongoDB แล้วฝึกโมเดล Machine Learning ในการพยากรณ์ราคาของ Bitcoin โดยใช้ Extra Tree Regressor โดยจะมีการดึงข้อมูลจาก API ของ Binance ทุก 5 นาที ผ่าน DAG จาก task Get\_BTC\_from\_Mongodb นำไปใช้ในการฝึกโมเดล ExtraTreesRegressor โดยใช้ฟีเจอร์ closeTime ในการพยากรณ์ lastPrice และมีการตั้งค่าให้ทำงานทุกชั่วโมง และมีการตั้งค่าพารามิเตอร์เริ่มต้น `n_estimators = 100` และ `random_state = 42`
- ข้อเสีย:** การใช้พารามิเตอร์เริ่มต้น (เช่น `n_estimators = 100` และ `random_state = 42`) อาจไม่เหมาะสมกับทุกกรณีหรือข้อมูล ซึ่งอาจส่งผลต่อค่า MAPE ซึ่งใช้ในการประเมินความแม่นยำของโมเดล

#### วันที่ 2 : ทำการปรับปรุงโมเดลโดย RandomizedSearchCV

- ทำการปรับปรุง Hyperparameters โดย **RandomizedSearchCV** เพื่อค้นหาชุดพารามิเตอร์ที่ดีที่สุดของ ExtraTreesRegressor โดยกำหนด `param_grid = {'n_estimators': [100, 200, 300], 'max_depth': [10, 20, None], 'min_samples_split': [2, 5, 10]}` ซึ่งการใช้ **RandomizedSearchCV** เป็นทางเลือกที่ดีเมื่อมี hyperparameters หลายตัว และหาค่าที่ดีที่สุดโดยไม่ต้องใช้เวลานาน
- ข้อเสีย:** เป็นเครื่องมือที่มีประสิทธิภาพในการปรับแต่ง hyperparameters แต่ก็มีข้อจำกัด ในด้านการสุ่มเลือกค่าที่อาจทำให้พลาดการหาค่าที่ดีที่สุด โดยเฉพาะในกรณีที่ต้องการความแม่นยำสูง จึงทำให้ค่า MAPE มีค่าลดลงจากวันที่ 1

### วันที่ 3 : ปรับกลับไปใช้โมเดลพื้นฐาน

- เนื่องจาก **RandomizedSearchCV** มีข้อเสียเกี่ยวกับการสุ่มเลือกพารามิเตอร์ในช่วงที่กำหนด ซึ่งหากการสุ่มจำนวนครั้งไม่เพียงพอหรือช่วงพารามิเตอร์ที่เลือกไม่เหมาะสม อาจทำให้ไม่สามารถหาค่าพารามิเตอร์ที่ดีที่สุดได้ ผลลัพธ์ที่ได้จากการใช้ **RandomizedSearchCV** และ **Extra Tree Regressor** แสดงให้เห็นว่า MAPE มีการลดลงอย่างมากทำให้สามารถสรุปได้ว่าโมเดลนี้ อาจไม่เหมาะสมกับลักษณะของข้อมูลเมื่อเปรียบเทียบกับโมเดลอื่น ๆ

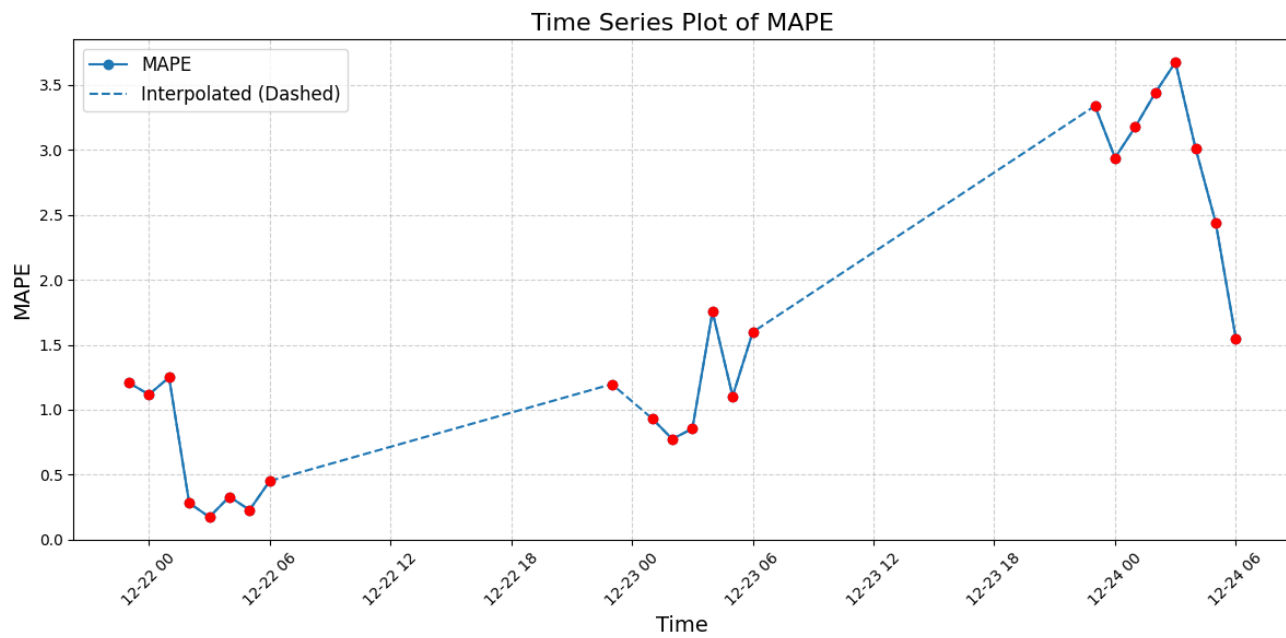
#### Output :

วันที่	23:00	24:00	01:00	02:00	03:00	04:00	05:00	06:00	MAPE เฉลี่ย
21/12	1.2073	1.1430	1.2483	0.2823	0.1742	0.3306	0.2281	0.4514	63.32%
22/12	1.1967	1.0912	0.9322	0.7744	0.8541	1.7543	1.1046	1.5971	116.31%
23/12	3.3379	2.9380	3.1752	3.4387	3.6737	3.0112	2.4404	1.5502	294.58%

จากตารางข้างต้น ทำให้ได้ข้อสรุป ดังนี้ :

ในวันที่ 21 ธันวาคม มีค่าความคลาดเคลื่อนต่ำสุดอยู่ที่เวลา 03:00 (0.1742) และมีค่าเฉลี่ย MAPE อยู่ที่ 63.32% ซึ่งแสดงให้เห็นว่าการพยากรณ์ในวันนี้ค่อนข้างแม่นยำ ขณะที่ในวันที่ 22 ธันวาคม ค่าความคลาดเคลื่อนต่ำสุดอยู่ที่เวลา 02:00 (0.7744) แต่ค่าเฉลี่ย MAPE กลับเพิ่มขึ้นเป็น 116.31% ซึ่งสะท้อนให้เห็นถึงความคลาดเคลื่อนที่สูงขึ้นเมื่อเทียบกับวันก่อนหน้า ส่วนในวันที่ 23 ธันวาคม ค่าความคลาดเคลื่อนสูงอย่างชัดเจนในทุกช่วงเวลา โดยสูงสุดที่เวลา 03:00 (3.6737) และค่าเฉลี่ย MAPE พุ่งสูงถึง 294.58% ซึ่งบ่งบอกถึงความคลาดเคลื่อนในการพยากรณ์ที่รุนแรง

ทั้งนี้ สังเกตได้ว่าความคลาดเคลื่อนเพิ่มสูงขึ้นเรื่อย ๆ จากวันที่ 21 ไปจนถึงวันที่ 23 โดยเฉพาะวันที่ 23 ธันวาคม ที่มีความคลาดเคลื่อนสูงผิดปกติ ซึ่งอาจเกิดจากปัจจัยอื่น



## Conclusion

วันที่	MAPE เฉลี่ย (ARIMA)	MAPE เฉลี่ย (RF)	MAPE เฉลี่ย (ET)	Best Model
21/12/2567	22.82%	24.46%	63.32%	ARIMA
22/12/2567	27.66%	28.53%	116.31%	ARIMA
23/12/2567	40.26%	43.98%	294.58%	ARIMA

จากการเปรียบเทียบประสิทธิภาพของ ARIMA (Autoregressive Integrated Moving Average), Random Forest Regressor (RF) และ Extra Trees Regressor (ET) ในการพยากรณ์ราคาของ Bitcoin (BTC) โดยคำนึงถึงค่า Mean Absolute Percentage Error (MAPE) เฉลี่ยจากวันที่ 21 ถึง 23 ธันวาคม 2567 ได้ว่า

ค่า MAPE เฉลี่ยของทุกโมเดลเพิ่มขึ้นในแต่ละวัน ซึ่งวันที่ 21 ธันวาคมเป็นวันที่มีประสิทธิภาพดีที่สุด (ค่า MAPE เฉลี่ยต่ำที่สุด) ขณะที่วันที่ 23 ธันวาคมมีค่า MAPE เฉลี่ยสูงที่สุด โดยค่า MAPE เฉลี่ยของ ARIMA ต่ำที่สุดในทุกวันที่ทำการทดสอบ โดยเฉพาะในวันแรก หรือ วันที่ 21 ธันวาคม โดยให้ผลลัพธ์ที่ดีที่สุด (22.82%) เมื่อเทียบกับโมเดลอื่น แม้ว่า ARIMA จะมีค่า MAPE ที่สูงขึ้นจากเดิมในวันที่ 22 และ 23 ธันวาคม (27.66% และ 40.26% ตามลำดับ) แต่ยังคงมีประสิทธิภาพดีกว่าโมเดล Random Forest Regressor และ Extra Trees Regressor อย่างเห็นได้ชัด

ขณะที่ ค่า MAPE เฉลี่ยของ Random Forest Regressor อยู่ในระดับที่ใกล้เคียงกับ ARIMA คือ 24.46% เทียบกับ 22.82% ในวันที่ 21 ธันวาคม และ 28.53% เทียบกับ 27.66% ในวันที่ 22 ธันวาคม ทว่าในวันสุดท้าย ค่า MAPE เฉลี่ยของ Random Forest Regressor เพิ่มขึ้นอย่างชัดเจนเป็น 43.98% แม้จะมีการปรับพารามิเตอร์ผ่าน GridSearchCV ในวันที่ 22 ธันวาคม ผลลัพธ์กลับไม่ดีขึ้น ซึ่งอาจสรุปได้ว่า Random Forest Regressor ไม่สามารถปรับตัวได้ดีกับข้อมูลที่มีความผันผวนสูง

ตลอดจน Extra Trees Regressor แสดงให้เห็นว่าไม่สามารถจัดการกับความผันผวนของราคา Bitcoin ได้ดีเท่าโมเดลอื่น ๆ โดยมีค่า MAPE สูงมากที่สุดในทุกวัน การใช้ Extra Trees Regressor ในการพยากรณ์ราคาของ Bitcoin อาจไม่ได้ผลดีในกรณีนี้ เนื่องจากโมเดลนี้มีความซับซ้อนในการเรียนรู้จากข้อมูลที่มีลักษณะไม่เสถียรและความผันผวนสูง

จากการทดสอบโมเดลทั้งสาม พบว่า ARIMA เป็นโมเดลเหมาะสมที่สุดสำหรับการพยากรณ์ราคาของ Bitcoin (BTC) เนื่องจากมีค่า MAPE เฉลี่ยที่ต่ำที่สุดของทุกวัน ด้วยความเหมาะสมกับข้อมูล Time Series และสามารถจัดการกับข้อมูลที่มีความผันผวนสูงโดยไม่จำเป็นต้องปรับแต่งพารามิเตอร์อย่างซับซ้อน

