BTC Price Forecasting using Time Series Analysis

Sitthi Ngowwattana 645162020022

Introduction:

Bitcoin คือสกุลเงินหนึ่งใน Cryptocurrency ที่มีมูลค่าสูงที่สุดและมีการเทรดอย่างแพร่หลาย มีหลายประเทศให้การยอมรับ และมีผู้คนให้ความสนใจจำนวนมาก จนนักลงทุนบางกลุ่มบอกว่านี่คือทางเลือกใหม่ของการลงทุน และในการทดลองนี้จะเป็นการทำนายราคาของ Bitcoin ในอนาคตด้วย time series analysis

Time series analysis เป็นวิธีการหนึ่งที่สามารถใช้ในการวิเคราะห์ข้อมูลเปรียบเทียบกับเวลา โดยการเรียนรู้จาก patterns และ trends ในอดีต ซึ่งสามารถช่วยเราในการพยากรณ์แนวโน้มในอนาคตได้เพื่อเป็นตัวช่วยในการตัดสินใจ

การทำนายราคาของผลิตภัณฑ์ทางการเงิน การลงทุน เช่น หุ้น, ทองคำ หรือแม้กระทั่งสินค้าอุปโภค บริโภค เช่น น้ำมันดิบ น้ำมันปาล์ม มีหลายหลายวิธีไม่ว่าจะเป็น ARIMA หรือจะเป็น Machine Learning อย่าง Linear Regression, Random Forest รวมทั้ง Deep Learning เช่น MLP, LSTM จะเห็นได้ว่ามีหลากหลายวิธีให้เลือกใช้ตามความเหมาะสมของปัญหา

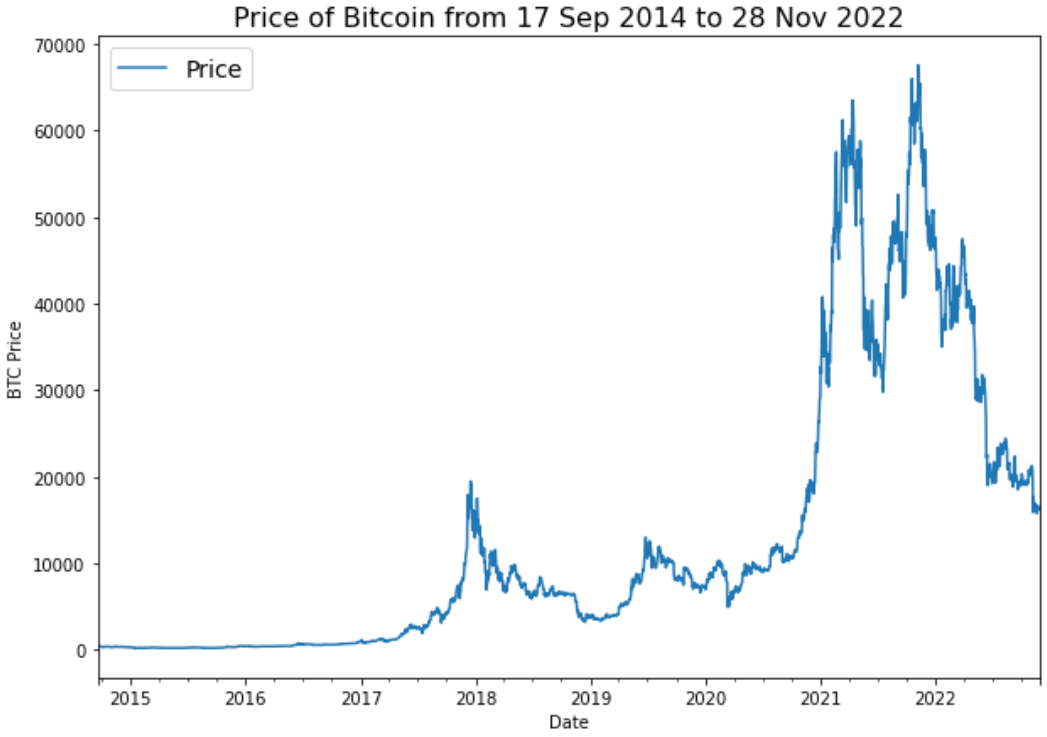
การทดลองในครั้งนี้เลือกใช้ Deep Learning เพื่อที่จะทำนาย Closing Price จากข้อมูในอดีต ด้วยโมเดลที่หลากหลาย นำมาเปรียบเทียบกัน จากนั้นวัดผลโมเดลด้วย **MAE** (mean absolute error), **RMSE** (root mean square error), **MAPE** (mean absolute percentage error), **sMAPE** (symmetric mean absolute percentage error) และ **MASE** (mean absolute scaled error) เพื่อที่จะได้ทราบว่าโมเดลใดมีความน่าสนใจที่สุดในกรณีศึกษาครั้งนี้

Problem Statement:

ต้องการทราบว่าแต่ละ time series model มีความแม่นยำเท่าไร และเมื่อนำมาเปรียบกันด้วยวิธีการต่างๆ โมเดลไหนจะเป็นโมเดลที่น่าสนใจที่สุดในการทดลองนี้ โดยที่ไม่ได้มองแค่ความแม่นยำเพียงอย่างเดียว แต่รวมทั้งการที่จะนำไปทดลองต่อในการปรับพารามิเตอร์ต่างๆ และนอกจากนี้ยังเป็นตัวช่วยในการตัดสินใจในการลงทุน ไม่ใช่เฉพาะกับ BTC หรือ Cryptocurrency อื่นๆ แต่ยังสามารถนำไปประยุกต์กับตราสารทุนอื่นๆ หรือปัญหาต่างๆที่สามารถใช้ time series Analysis ได้

Data Exploration:

ใช้ข้อมูลราคา BTC (USD) ช่วงวันที่ 17 กันยายน 2014 ถึงวันที่ 28 พฤศจิกายน 2022 จากเว็บไซต์ Yahoo! Finance มีข้อมูลทั้งสิ้น 2,995 วัน โดยข้อมูลที่มีประกอบไปด้วย Date, Open, High, Low, Close, Adj Close และ Volume และได้ทำการสำรวจข้อมูลด้วยการ Plot ดูข้อมูลทั้งช่วงที่มี จะพบว่า BTC นั้นมีขาขึ้นในช่วงปี 2018 และ 2021 จากข้อมูลราคาของ BTC มีความผันผวนมากในช่วงปี 2021-2022 โดยที่ Close Price สูงสุดคือ $67,566.83 เกิดขึ้นในวันที่ 8 พฤศจิกายน 2021

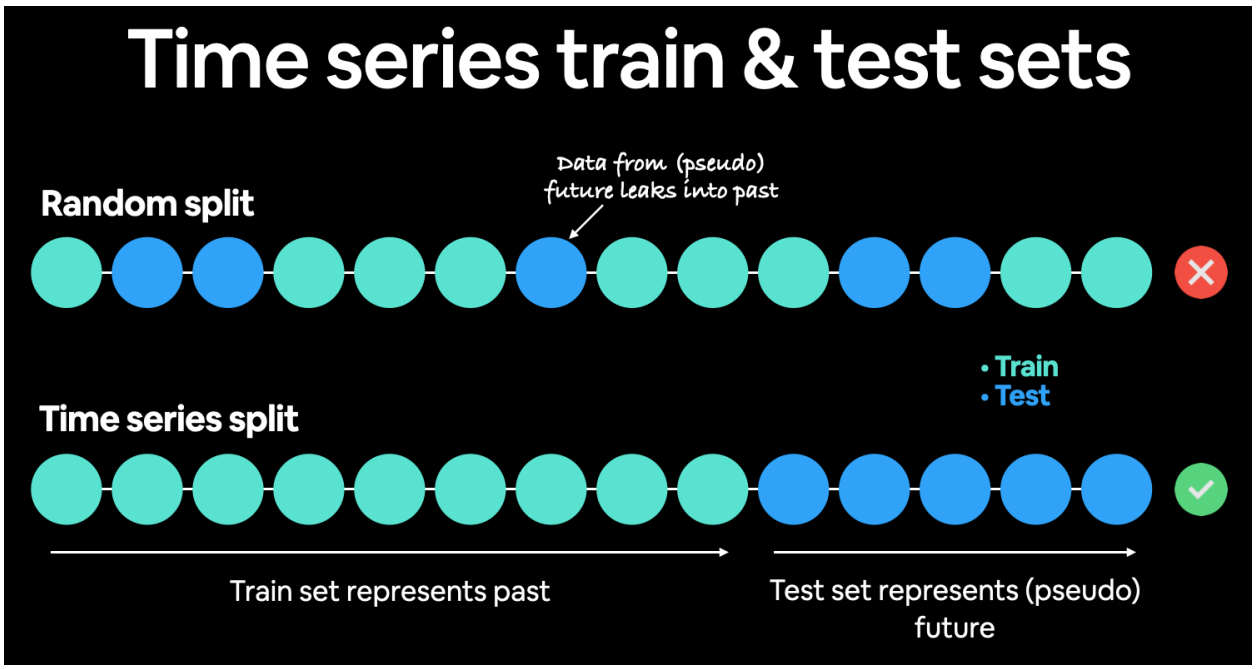


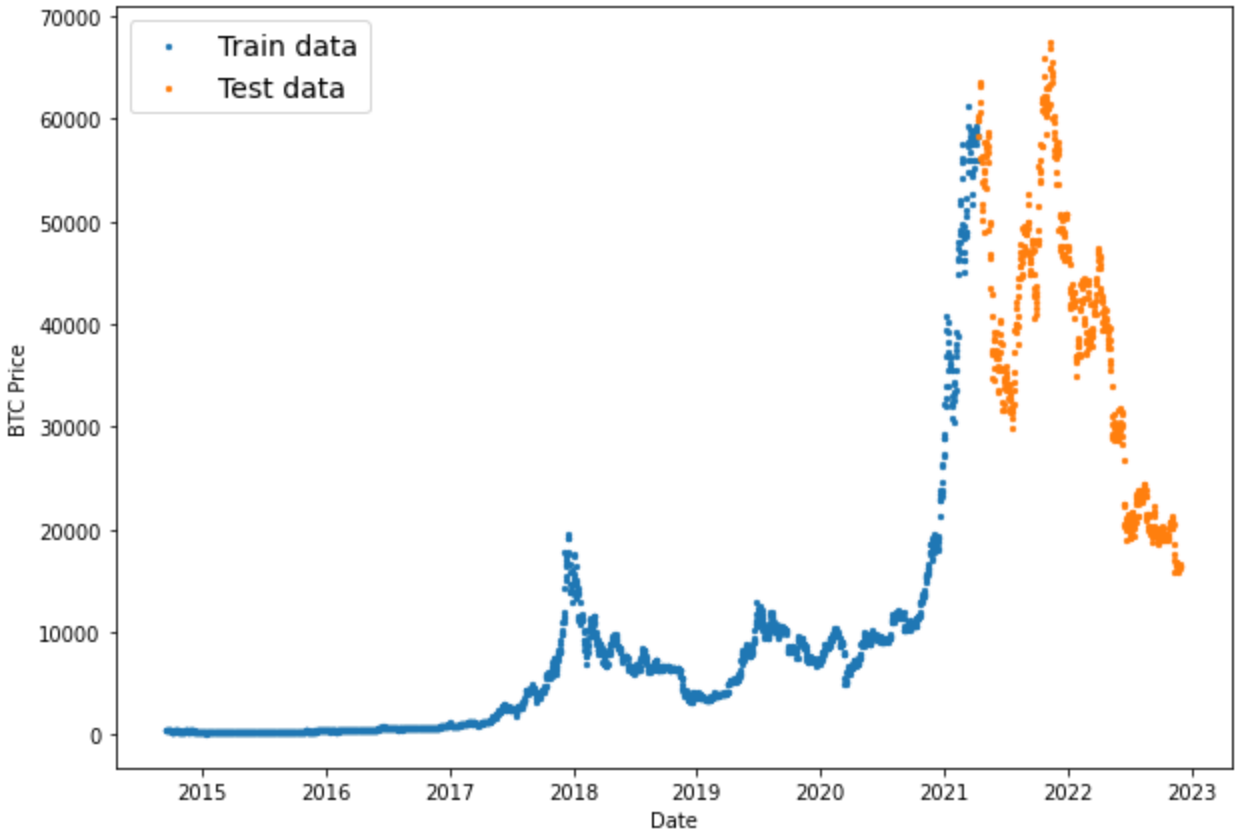
Data Pre-processing:

1. Mount Google Drive
2. Load ข้อมูลราคา BTC(USD) จาก Yahoo! Finance โดย Pandas
    a. ทำการ Parse date column ให้เป็น datetime และ set index
    b. เช็คว่ามี Null หรือไม่ และนับจำนวน record ของข้อมูลชุดนี้
    c. ทำการ Drop column ที่ไม่จำเป็นออก การทดลองนี้ใช้ Date และ Close เท่านั้น
3. จัดเตรียมข้อมูล ให้เหมาะสมกับปัญหา time series ได้แก่

a. สร้าง Training set และ Test set ในรูปแบบ Time series split

เราจะทำการแบ่ง Train/Test เป็น 80/20 โดยการแบ่งข้อมูลใน time serie จะไม่ใช่การ Random split เหมือนวิธีการอื่นๆ แต่จะเป็นการแบ่งช่วงเวลาในอดีตไกลๆเป็น Training set ในการทดลองนี้ใช้ 80% แรกของข้อมูล และที่เหลืออีก 20% เป็น Test Set และเมื่อแบ่งข้อมูลเสร็จแล้วก็ทำการ Plot จะได้ตามกราฟด้านล่าง





4. ทำการ window ข้อมูล

เป็นการกำหนดให้โมเดลนำข้อมูลก่อนหน้า มาช่วยกันทำนายใน timestep ถัดไป เช่น นำข้อมูลก่อนหน้า 7 timestep มาทำนาย timestep ต่อไป ดังตัวอย่างข้างล่าง นำข้อมูลวันที่ 0 ถึงวันที่ 6 มาทำนายผลของวันที่ 7 และจะทำการ slide window ไปเรื่อยๆ นอกจากนี้เรายังสามารถกำหนดได้ทั้ง window และ horizon

- Window คือจำนวนของ timestep ก่อนหน้าที่จะนำมาทำนาย timestep ต่อไป (horizon)
- Horizon คือจำนวนของ timestep ถัดไปที่เราจะทำนายในอนาคต

```
Window for one week (univariate time series)

[0, 1, 2, 3, 4, 5, 6] -> [7]
[1, 2, 3, 4, 5, 6, 7] -> [8]
[2, 3, 4, 5, 6, 7, 8] -> [9]
```

5. สร้างตัวแปรในการทำ multivariate

ทำ feature engineering สร้างตัวแปรใหม่ขึ้นมา เพื่อใช้เป็น input เพิ่มเติมในการทำการทดลอง ตัวแปรที่สร้างขึ้นมาใหม่จะเป็น Block Reward ซึ่ง Block Reward คือ ผลตอบแทนสำหรับ Miner ที่คำนวณแฮ

ชอย่างถูกต้องสำเร็จในระหว่างกระบวนการขุด Crypto asset ผลตอบแทนยังเพื่อให้ Miner ได้เกิดความมีส่วนร่วมสร้างความเคลื่อนไหว ไปจนถึงได้เป็นส่วนหนึ่งในการรับผิดชอบดูแลความปลอดภัยในระบบ Blockchain ดังตารางข้างล่าง

Block Reward	Start Date
50	3 January 2009 (2009-01-03)
25	28 November 2012
12.5	9 July 2016
6.25	11 May 2020
3.125	TBA (expected 2024)
1.5625	TBA (expected 2028)

6. สร้าง Model Checkpoint เพื่อเก็บผลของ Epoch ที่ดีที่สุดไปเปรียบเทียบกับผลของโมเดลอื่นๆ เพื่อที่จะได้นำผลการทดลองมาสรุปในภายหลัง

Methodology / Approach:

Model Number	Model Type	Horizon size	Window size	Extra data
0	Naïve model (baseline)	NA	NA	NA
1	Dense model	1	7	NA
2	Dense model	1	30	NA
3	Dense model	7	30	NA
4	Conv1D	1	7	NA
5	LSTM	1	7	NA
6	Dense model (multivariate data)	1	7	Block reward size
7	N-BEATs Algorithm	1	7	NA

1. Naive model

Naive model เป็นโมเดล baseline สำหรับการทำนายผลของ time series ซึ่งเป็นโมเดลที่ง่ายที่สุด ไม่จำเป็นต้องการการ training นั่นหมายความว่า naive model นั้นใช้เพียงแค่ timestep ก่อนหน้าที่จะใช้ในการทำนายเท่านั้น

2. Dense (fully-connected) networks

Dense หรือ fully-connected networks คือประเภทของ neural network ที่แต่ละ neuron ในหนึ่งเลเยอร์นั้นเชื่อมต่อกับทุก neuron ในเลเยอร์ถัดไป นั่นหมายความว่าแต่ละ neuron จะได้รับ input จากทุก neuron ในเลเยอร์ก่อนหน้า และ output จะถูกส่งต่อไปยังเลเยอร์ถัดไป ซึ่งการกระทำเหล่านี้จะเรียกว่า interconnected manner ซึ่งจะช่วยให้ network นี้สามารถที่จะเรียนรู้สิ่งที่มีความซับซ้อนได้จากความสัมพันธ์ระหว่าง input และ output

Fully-connected networks ถูกใช้อย่างกว้างขวางรวมทั้ง image and speech recognition, NLP และอื่นๆอีกมากมาย และอื่นๆจุดแข็งของ fully-connected network คือสามารถทำความเข้าใจได้ง่าย เทรนได้ง่าย เมื่อเปรียบเทียบกับ neural networks ประเภทอื่น

3. Sequence models (LSTM and 1D CNN)

Sequence models คือ machine learing model ประเภทหนึ่งที่ถูกออกแบบเพื่อให้ทำนายข้อมูลที่เป็นลำดับ เช่น ข้อมูล time series และ natural language text เป็นต้น sequence model ที่เป็นที่นิยมได้แก่ long short-term memory (LSTM) และ one-dimensional convolutional neural network (1D CNN)

Long short-term memory (LSTM) เป็นโครงข่ายประเภท RNN รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียรและมีประสิทธิภาพมากขึ้น โดยมีหลักการทำงานคือ สามารถเก็บสถานะ หรือข้อมูลของแต่ละโหนดเอาไว้เพื่อที่เวลาย้อนกลับไปดูจะได้ทราบถึงที่มาของข้อมูลค่าดังกล่าวว่าเดิมเป็นค่าอะไร และจุดเด่นของแบบจำลอง LSTM คือฟังก์ชันพิเศษที่มีหน้าที่เสมือนประตู(Gate) ที่คอยความคุมข้อมูลที่จะเข้าในแต่ละโหนด ซึ่งประกอบด้วย Forget gate layer, Input gate layer และ Output gate layer

Convolutional Neural Network จัดว่าเป็นโครงสร้าง Neural Network แบบพิเศษที่ความสามารถในการเรียนรู้ที่ จะสกัดคุณลักษณะขึ้นมาเอง (Feature Engineering) จากข้อมูลที่เป็น Input โดยแทนที่จะใช้ Activation Function แบบปกติ CNN จะใช้ Convolution และ Pooling Function แทน

4. Multivariate model

ในการทดลองนี้จะเป็นการทำ feature enginnering เพื่อสร้างตัวแปรใหม่ขึ้นมาใช้เป็น output อีกตัวหนึ่ง มาใช้กับโมเดล fully-connected networks ซึ่งสามารถทำให้ model มีค่า error ที่ลดลง

5. N-BEATS

N-BEATS เป็น neural network architecture สำหรับการทำนายข้อมูล time series ถูกออกแบบมาเพื่อจัดการกับการทำนายข้อมูลหลายแบบ และมันเหมาะสำหรับการทำนายข้อมูลแบบ long-term forecasting โครงข่ายนี้มีสองส่วนหลักคือ Backcast component และ Forecast component ส่วน Backcast

component จัดการกับการโมเดลปฏิบัติการของข้อมูลในอดีต และส่วน Forecast component จัดการกับการทำนายค่าของข้อมูลอนาคต หนึ่งในประโยชน์หลักของ N-BEATS คือ สามารถเรียนรู้แนวโน้มระยะยาวและการเปลี่ยนแปลงในขณะที่มีข้อมูล time series มันทำเช่นนี้โดยแยกข้อมูลเวลาเป็นกลุ่มของการซิงค์ตามช่วงเวลาต่าง ๆ ซึ่งช่วยให้สามารถจับคู่แนวโน้มของช่วงเวลาต่าง ๆ นี้ขึ้นได้ ซึ่งทำให้เหมาะสำหรับงาน long-term forecasting

Experimental Results:

หน่วยที่ใช้ในการวัดผลการทดลองได้แก่

Scale-dependent errors

- MAE (mean absolute error)
- RMSE (root mean square error)

Percentage errors

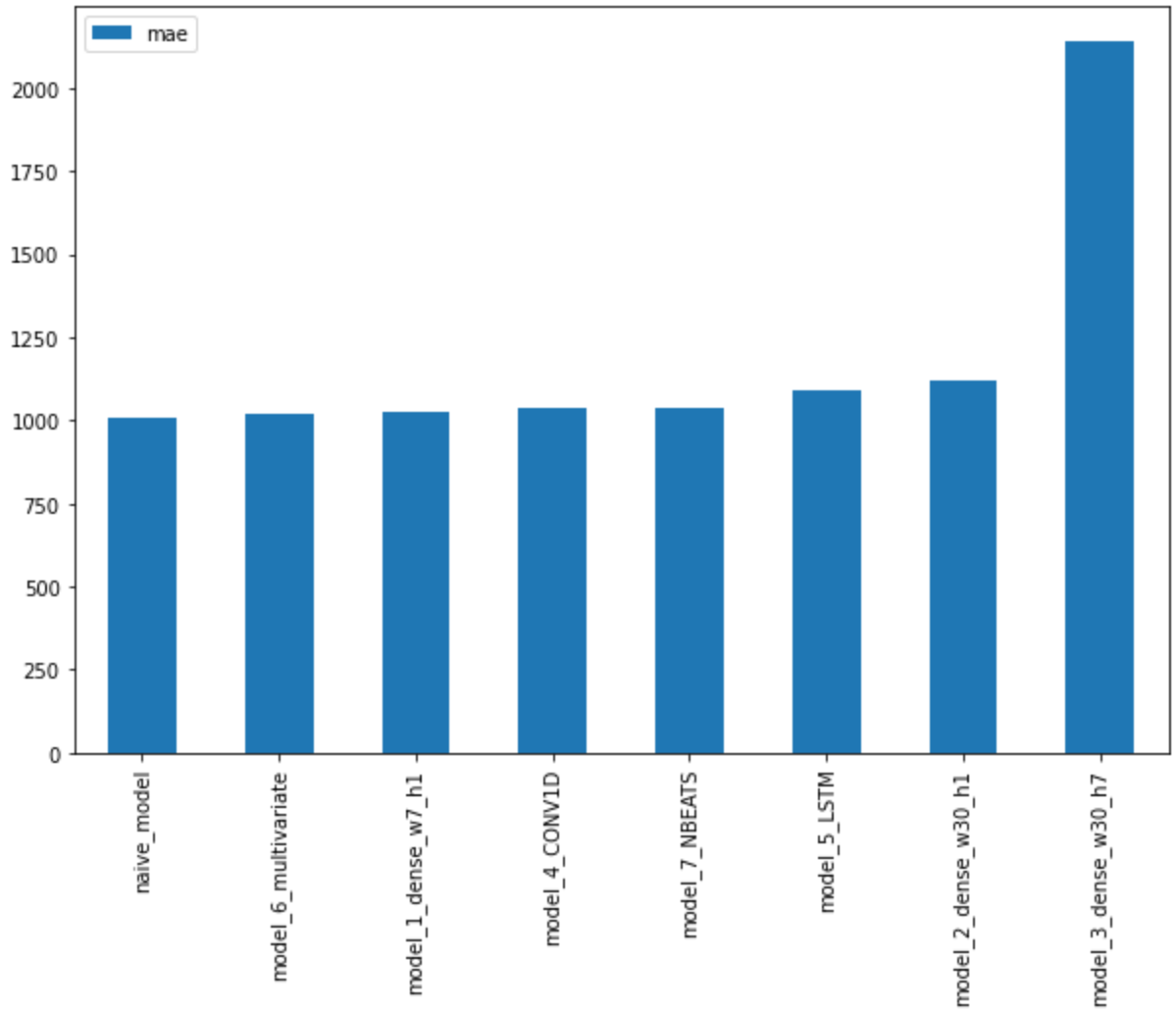
- MAPE (mean absolute percentage error)
- sMAPE (symmetric mean absolute percentage error)

Scaled errors

- MASE (mean absolute scaled error)

ผลการทดลองสามารถดูได้จากตารางด้านล่าง พบว่า naive model (baseline) มีค่า error น้อยที่สุด

	mae	mse	rmse	mape	mase
naive_model	1006.620422	2170030.25	1473.102173	2.656193	1.000902
model_1_dense_w7_h1	1023.898926	2217592.50	1489.158325	2.711378	1.018082
model_2_dense_w30_h1	1119.237793	2494597.50	1579.429443	2.987905	1.112698
model_3_dense_w30_h7	2138.294922	9105924.00	2443.218018	5.738970	2.116955
model_4_CONV1D	1037.942261	2254313.75	1501.437256	2.756398	1.032046
model_5_LSTM	1093.163208	2384758.75	1544.266479	2.901080	1.086953
model_6_multivariate	1019.881165	2195179.25	1481.613770	2.697946	1.014087
model_7_NBEATS	1038.754150	2247351.50	1499.116943	2.739160	1.032853



Discussion / Conclusion / Future Work:

จากผลการทดลองในหัวข้อที่ผ่านมา พบว่า naive model มีค่า error ที่น้อยที่สุด แต่อย่างไรก็ตามโมเดล multivariate ก็เป็นที่น่าสนใจถ้าเราได้ทำการ feature engineering เพื่อหาตัวแปรใหม่ๆมาเป็น input เพิ่มเติม เพื่อให้โมเดลเกิดการเรียนรู้ และหาความสัมพันธ์เพื่อมาช่วยในการทำนายผล รวมถึงการพัฒนาโมเดลอื่นๆ ได้แก่ CONV1D, LSTM, N-BEATS ให้เป็น multivariate model

แนวทางที่จะปรับปรุงในอนาคต จะเน้นไปที่การค้นคว้าเพิ่มเติมเพื่อให้มีมุมมองต่างๆ และแนวความคิดใหม่ๆ ที่จะช่วยในการทำ Feature Engineer รวมถึงการทดลองเพิ่ม Layer และปรับแต่งพารามิเตอร์ ให้เหมาะสมกับปัญหาที่ทำการทดลอง