

# การหาแนวโน้มของตลาดหุ้น S&P 500

ปริญญา อบอูน

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

## บทคัดย่อ

โปรเจกต์นี้นำเสนอเกี่ยวกับระบบหาแนวโน้มของตลาดหุ้นดัชนี S&P 500 โดยใช้เทคนิคการประมวลผลสัญญาณดิจิทัล ได้แก่ เส้นค่าเฉลี่ยเคลื่อนที่ (Moving Average) เส้นค่าเฉลี่ยเคลื่อนที่แบบเอ็กซ์โพเนนเชียล (EMA) และ model Auto-Regressive Integrated Moving Average สำหรับการพยากรณ์ทิศทางตลาด การศึกษานี้ใช้ข้อมูลปิดรายวันของดัชนี S&P 500 ตั้งแต่ปี 2015-2025 จำนวนข้อมูล 2,516 จุด โดยข้อมูลจะผ่านการทดสอบ Stationarity ด้วย ADF และ KPSS test และหาค่า parameter ที่เหมาะสม Grid Search 100x100 combination จากการทดลองพบว่า EMA(11) ตัดกับ SMA(99) EMA(11) x SMA(99) ให้ประสิทธิภาพสูงสุดที่ Accuracy 53.45% Precision 54.90% และ Recall 77.24% ในส่วนของ EMA ตัดกับ SMA ในส่วนของ EMA(11) ตัดกับ EMA(99) ให้ผลที่ดีกว่าสำหรับ EMAfast ตัดกับ EMAslow คือ Accuracy 52.83% และ Precision 54.35% และ Recall 78.77% และที่โมเดล ARIMA(1,0,2) ให้ Accuracy 52.19% ทั้งสามวิธีนี้มีประสิทธิภาพสูงกว่าแบบการทำนายการสุ่มอย่างมีนัยสำคัญ เป็นการแสดงให้เห็นความไปได้ในการประยุกต์ใช้หลักการทางด้านการประมวลผลสัญญาณดิจิทัลเพื่อสร้าง Technical Indicator เพื่อมีประสิทธิภาพในการวิเคราะห์ตลาดหุ้น

## Abstract

This project present Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

## 1 บทนำ

### 1.1 ที่มาและความสำคัญ

ในสถานการณ์ปัจจุบันโลกนั้นได้เจอกับการเปลี่ยนแปลงโดยการมีมาตรการจากนโยบายการเงินที่ผ่อนคลายเชิงระบบ (Quantative Easing) ของธนาคารกลางทั่วโลกทำให้เกิดการขยายตัวของปริมาณเงินในระบบหรือที่เรียกว่าเงินเฟ้อ โดยข้อมูลจาก Federal Reserve แสดงให้เห็นว่าเงิน M2 เพิ่มขึ้นอย่างมาก

ดังนั้นถ้าอยากให้ความสามารถในการจับจ่ายใช้สอยได้เท่าเดิมเราต้องเอาเงินของเราไปลงทุนเพื่อที่จะทำให้เงินของเราอกเงย

โครงการชิ้นนี้จะพัฒนาและวิเคราะห์โดยใช้ความรู้ทางด้าน Digital signal processing เพื่อสร้างตัวชี้วัด (indicator) ที่สามารถตรวจวัดการเปลี่ยนแปลงของตลาด

### 1.2 วัตถุประสงค์

1. พัฒนาระบบวิเคราะห์แนวโน้มของตลาดหุ้น S&P 500 โดยใช้เทคนิค Moving Average Crossover และ statical model (ARIMA)
2. สร้าง Technical Indicator ที่มีความแม่นยำมากกว่าการเดาสุ่ม (50%) โดยต้องมากกว่าแบบมีนัยสำคัญ
3. วิเคราะห์คุณสมบัติของ stationarity ของข้อมูล S&P 500 index ด้วย ADF และ KPSS test

### 1.3 ขอบเขต

โครงการนี้มีขอบเขตดังต่อไปนี้ :

1. ขอบเขตด้านปฏิบัติ
  - ใช้ข้อมูลราคาปิดของกราฟรายวัน (daily) ของ S&P 500 index (^GSPC) จาก Yahoo Finance
  - ช่วงเวลา: 1 มกราคม 2015 ถึง 1 มกราคม 2025 (เป็นระยะเวลา 10 ปี)
  - จำนวนจุดข้อมูล: 2,516 observations
  - ประเภทข้อมูล: ราคาปิด (close price) ในรูปแบบ time series ที่สัญญาณเป็น non-stationary
2. ขอบเขตด้านวิธีการ
  - Technical analysis: ศึกษาแค่ Moving average indicator (SMA และ EMA)
  - Optimization: Grid Search 100x100 (10,000 combination)
  - Evaluation: เน้นความแม่นยำ accuracy ไม่รวม profit/loss
3. ขอบเขตด้านการทดสอบ
  - ทดสอบประสิทธิภาพด้วย rolling window prediction (one-step ahead)
  - ใช้ entire dataset ไม่มีการแบ่ง train / test

## 2 ทฤษฎีที่นำเสนอ

### 2.1 ทฤษฎีที่เกี่ยวข้อง (Theoretical Background)

#### 2.1.1 เส้นค่าเฉลี่ยเคลื่อนที่ (Moving average)

**นิยาม** Moving average (MA) เป็นตัวกรองแบบ Finite Impulse Response (FIR) ที่ใช้ในทางด้าน Digital Signal Processing โดยตัว Moving Average (MA) ทำหน้าที่เป็น Low-pass Filter เพื่อลดความถี่สูงของสัญญาณรบกวน (high-frequency noise) และทำให้เห็นแนวโน้มพื้นฐาน (underlying trend) ของอนุกรมเวลา

โดยทำการเฉลี่ยค่าล่าสุดในหน้าต่างขนาด  $n$

$$MA_w(t) = \left(\frac{1}{w}\right) \sum_{i=t-w+1}^t x_k$$

โดยที่:

- $MA_w(t)$  คือ ค่าเฉลี่ยเคลื่อนที่ (Moving average) ณ เวลาที่ด้วยหน้าต่างต่าง  $w$
- $w$  คือ ขนาดหน้าต่าง
- $x_i$  คือค่าที่เวลา  $i$
- $t$  คือ เวลาปัจจุบัน

คุณสมบัติทางคณิตศาสตร์ (Mathematical Properties):

1. Linear Time-Invariant(LTI) System: MA เป็นระบบ LTI ที่มีคุณสมบัติ causality และ stability
2. Phase Delay: ทำให้เกิด phase delay เท่ากับ  $\frac{w-1}{2}$  samples
3. Smoothing Factor: ระดับการปรับเรียบผกผันกับ  $w$

**Implement:** ใน rust ระบบใช้ module sma.rs ที่ implement โดยใช้  $O(n)$  สำหรับการคำนวณ

การใช้ MA ช่วยลดความผันผวนของสัญญาณและช่วยให้มองเห็นแนวโน้มที่ชัดเจนยิ่งขึ้นแม้ว่าหุ้นนั้นจะแกว่งมากก็ตาม

#### 2.1.2 เส้นค่าเฉลี่ยเคลื่อนที่แบบเอ็กซ์โพเนนเชียล (Exponential moving average)

**นิยาม** Exponential Moving Average (EMA) เป็นตัวกรองแบบ Infinite Impulse Response (IIR) ที่ให้น้ำหนักแบบ exponential decaying กับข้อมูล

ในอดีต ทำให้มีการตอบสนอง (response) ที่รวดเร็วกว่า MA

$$\text{EMA}_w(t) = x_k \left( \frac{S}{1+\text{Days}} \right) + \text{EMA}_{t-1} \left( 1 - \left( \frac{S}{1+\text{Days}} \right) \right)$$

โดยที่:

- $\alpha = \frac{S}{1+\text{Days}}$  = smoothing factor
- $S$  = smoothing constant (ใช้ค่า 2 ในโปรเจกต์)
- **Days** = จำนวน periods (windows)
- $x_t$  = ค่าปัจจุบัน
- $\text{EMA}_{t-1}$  = ค่า EMA ก่อนหน้า

คุณสมบัติทางศาสตร์ (Mathematical Properties):

1. Memory Effect: EMA มี infinite memory แต่ให้น้ำหนักลดแบบ exponential:  $w_k = \alpha(1 - \alpha)^k$
2. Effective Window: หน้าต่างทำให้เกิด phase delay เท่ากับ  $\frac{2}{\alpha} - 1$  periods
3. Lag Reduction: มี lag น้อยกว่า SMA เท่ากับ  $\frac{\omega-1}{2}$  periods

**Implementation:** ema.rs ใช้ iterative approach ด้วย time complexity  $O(n)$  และ space complexity  $O(1)$

จะเป็นสูตรที่ตอบสนองต่อการเปลี่ยนแปลงของข้อมูลได้เร็วกว่าจะได้รู้ว่า trend ของตลาดไปทางไหนได้เร็วกว่าแบบคือ MA

### 2.1.3 Auto-Regressive Integrated Moving Average

ARIMA คือโมเดลสำหรับวิเคราะห์อนุกรมเวลา โดยรวมองค์ประกอบของการถดถอยอัตโนมัติ (Auto-Regressive) และค่าเฉลี่ยเคลื่อนที่ (Mean Average) เข้าด้วยกัน พร้อมกับการทำให้ข้อมูลเป็นสถิติก่อนด้วยการหาค่าต่าง (differencing) จำนวน  $d$  ครั้ง โมเดล ARIMA สามารถเขียนเป็น:

$$Y_t = \alpha + \beta_{(1)} Y_{t-1} + \beta_2 Y_{t-2} +$$

$$.. + \beta_p Y_{t-p} \varepsilon_{t-1} + \Phi_1 \varepsilon_{t-1} + \Phi_2 \varepsilon_{t-2} + \Phi_q \varepsilon_{t-q}$$

เลือก parameter โดยการ โดยที่:

- $Y_t$  = ค่าที่สังเกต ณ เวลา  $t$  (หลัง differencing)
- $c$  = ค่าคงที่ (drift term)
- $\Phi_i$  = สัมประสิทธิ์ Auto-Regressive(AR) ที่ lag  $i$  ( $|\Phi_i| < 1$  สำหรับ stationarity)
- $\Theta_j$  = สัมประสิทธิ์ Moving-average (MA) ที่ lag  $j$  ( $|\Theta_j| < 1$  สำหรับ invertibility)
- $\varepsilon_t$  = white noise error term  $\sim N(0, \sigma^2)$
- $L$  = lag Operator ( $L^k \cdot Y_t = Y_{t-k}$ )

การระบุ Parameters (model Identification):

1. Differencing Order (d):
  - ใช้ Augmented Dickey-Fuller (ADF) test
  - KPSS test สำหรับ trend stationarity
2. AR Order (p) ผ่าน PACF:
  - ใช้ OLS regression
  - Cut-off criterion:  $|\text{PACF}(K)| < \frac{1.96}{\sqrt{n}}$
3. MA Order (q) ผ่าน ACF:
  - ใช้ FFT-base Compute สำหรับงาน efficiency
  - Autocorrelation function ที่ lag  $k$

### 2.1.4 การแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform)

เป็น algorithm ที่ optimal ที่สุดในเวลานี้สำหรับการคำนวณ Discrete Fourier Transform (DFT):

$$X[k] = \sum_{n=0}^{N-1} \left( x[n] * e^{\frac{-j2\pi kn}{N}} \right)$$

การนำมาประยุกต์:

1. Spectral Density Estimation
2. ACF via Winer-Khinchin Theorem
3. Periodicity Detection

## 2.2 การประเมินประสิทธิภาพของระบบ

1. Accuracy (ความถูกต้องโดยรวม):

$$\text{Accuracy} = \frac{TP + TN}{\text{Total}}$$

2. Precision (ความแม่นยำ):

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. Recall (ความไว):

$$\text{Recall} = \frac{TP}{TP + FN}$$

## 2.3 วิธีการแก้ปัญหา

### 2.3.1 โครงสร้างระบบ (System Architecture)

src/module/

- └─ data
- └─ eval.rs
- └─ indicator
- └─ model
- └─ mod.rs
- └─ plot
- └─ util
- └─ workflow.rs

Design Principles:

1. Vibe Coding

### 2.3.1 Algorithm Implementation Details

EMA Crossover Strategy:

- Fast Ema (period สั้น) ตัดบน Slow Ema (period ยาว) = Buy Signal
- Zero-padding สำหรับ boundary conditions
- Time Complexity:  $O(n)$  per prediction

ARIMA (Auto-Regressive integrate moving average):

1. Stationary testing -> ADF/KPSS test
2. Parameter Selection -> PADF/ACF analysis
3. Model fitting -> CSSoptimization
4. Rolling Prediction -> One-step ahead forecasts

5. Performance Evalution -> Direactional accuracy

## 3 ผลการทดลอง

การทดสอบใช้ข้อมูลของหุ้น S&P 500 Index (^GSPC) จาก Yahoo Finanace ระหว่างวันที่ 1 มกราคม 2015 ถึง 1 มกราคม 2025 ทั้งหมดคือ 2,516 จุดรายวัน (daily) โดยใช้ตัวแปรคือราคาปิดของวันนั้น

ข้อมูลทั้งหมดที่เป็นความต่างของข้อมูลวันนี้น้และเมื่อวาน (logs return) จะถูกนำไปใช้ใน ARIMA เพื่อการพยากรณ์ว่าจะขึ้นหรือลงวันพรุ่งนี้ที่ (T+1) โดยต้องนำไปทดสอบเพื่อหา parameter คือ ค่า d ว่าเท่ากับเท่าไร ถ้าเป็น stationary จะให้ค่า d = 0 ถ้าไม่ใช่ stationary จะให้ค่า d = 1

การทดสอบ	ค่าสถิติ	ค่าวิกฤต (1%)	ผลการทดสอบ
ADF Test (Diff)	$t = -41.160$	-3.460	Stationary
KPSS Test (Diff)	0.175	0.739	Level-stationary

สรุป:ผลการทดสอบ ADF แสดงค่า t-statistic ที่ต่ำกว่าค่าวิกฤตอย่างมีนัยสำคัญ ( $p < 0.01$ ) ทำให้ปฏิเสธสมมติฐานหลักของ unit root ได้ ขณะที่ KPSS test ไม่สามารถปฏิเสธสมมติฐานหลักของ stationarity ได้ แสดงว่าข้อมูลมีคุณสมบัติ stationary เหมาะสมสำหรับการใช้งาน

จากผลดังตารางเราจะให้ค่า d = 0 เป็น stationary จะเรียกว่า ARMA(p,q)

### การเลือกพารามิเตอร์ของโมเดล

การวิเคราะห์ Partial Autocorrelation Function (PACF) ด้วยวิธี Levinson-Durbin และ OLS

regression ให้ผลสอดคล้องกันที่  $p = 1$  โดยใช้เกณฑ์ cutoff ที่ระดับความเชื่อมั่น 95% ( $\pm \frac{1.96}{\sqrt{n}} = \pm 0.0391$ )

### การหาค่า MA Order (q) ด้วย ACF

การวิเคราะห์ Autocorrelation Function ด้วย FFT-based algorithm แสดงผลดังนี้:

- First-drop criterion:  $q = 2$
- Largest significant lag:  $q = 10$

จะพิจารณาเลือกใช้ first drop

### 3.4.1 Grid Search Optimization

จะทำการค้นหาค่าที่ดีที่สุด (accuracy) สูงสุดด้วยการทดสอบ 100x100 ของ ema/sma period ได้ผลดังนี้

Strategy	Parameters	Accuracy	Coverage
Best	EMA(11) × SMA(99)	53.45%	96.10%

จากการทดลองแสดงให้เห็นว่า

โมเดล EMA(11) × SMA(99) ให้ความแม่นยำสูงสุดที่ 53.45% ด้วย precision 54.90% และ recall 77.24% ตามมาด้วย EMA(11) × EMA(99) ที่ให้ความแม่นยำ 52.83% (precision 54.36%, recall 78.77%) และ ARIMA(1,0,2) ให้ความแม่นยำ 52.19% (precision 53.80%, recall 78.16%)

การวิเคราะห์รายละเอียดของ Best Model (EMA(11)×SMA(99)) จากจำนวนข้อมูลทั้งหมด 2,417 จุด แบ่งเป็น 4 กลุ่มดังนี้ True Positive (ทำนายขึ้นและขึ้นจริง) 1,008 ครั้ง คิดเป็น 41.7% ของทั้งหมด True Negative (ทำนายลงและลงจริง) 284 ครั้ง คิดเป็น 11.8% False Positive (ทำนายขึ้นแต่ลงจริง) 297 ครั้ง คิดเป็น 12.3% และ False Negative (ทำนายลงแต่ขึ้นจริง) 828 ครั้ง คิดเป็น 34.3%

โมเดล ARIMA(1,0,2) ที่ได้จากการประมาณค่าด้วย Conditional Sum of Squares ให้สมการพยากรณ์ดังนี้  $Y_t = 9.020 + 0.998Y_{t-1} + \varepsilon_t - 0.043\varepsilon_{t-1} + 0.016\varepsilon_{t-2}$  ค่าสัมประสิทธิ์ AR ที่ 0.998 โดยความถูกต้องโดยรวมอยู่ที่ 52.19% และ ความแม่นยำอยู่ที่ 53.80%

## 4 การวิเคราะห์และสรุปผล

1. ประสิทธิภาพของ technical indicator: ด้วยวิธี EMA crossover จะได้ประสิทธิภาพที่ EMA(11) × EMA(99) ได้ผลลัพธ์ที่ดีที่สุดโดยวัดจากความแม่นยำ accuracy คือ 53.35% และ precision ที่ 54.90%
2. การเลือกใช้ period ที่เหมาะสมสำหรับการทำ indicator อาจจะปรับให้มากกว่านี้เช่น EMA(11) โดยเลข 11 ยังเป็นเลขที่มีนัยสำคัญมากในการวิเคราะห์โดยการใช้ indicator ครั้งที่
3. ข้อจำกัดของ ARIMA: ตลาดหุ้นเป็น non linear และ deterministic ทำให้ ARIMA ไม่สามารถประมาณค่าได้แม่นยำมาก โดยในการทดลองนี้ใกล้เคียงกับ moving average crossover

โครงการนี้แสดงให้เห็นการประยุกต์โดยใช้เทคนิคทาง digital signal processing เพื่อสร้าง threading indicator ที่มีประสิทธิภาพเหนือกว่าการสุ่มแบบโยนเหรียญเล็กน้อย

## 5 กิตติกรรมประกาศ

## 6 เอกสารอ้างอิง

- [1] Algorithmic-Oriented Digital Signal Processing for Computer Engineers 01204496, “บท 5: การประมวลผลสัญญาณปรับตัว (Adaptive Signal Processing),” lecture notes, [in Thai].
- [2] Anthropic, “Claude AI [Large language model],” Sep. 2025. [Online]. Available: <https://www.anthropic.com/claude>

[3] P. S. R. Diniz, Signal Processing and Machine Learning Theory. Academic Press, 2024.

[4] G. Gundersen, “Returns and Log Returns,” Sep. 2022. [Online]. Available: <https://gregorygundersen.com/blog/2022/02/06/log-returns/>

[5] OpenAI, “ChatGPT [Large language model],” Sep. 2025. [Online]. Available: <https://chatgpt.com>

[6] S. Prabhakaran, “Time Series Analysis in Python – A Comprehensive Guide with Examples,” MachineLearningPlus (ML+), Feb. 13, 2019. [Online]. Available: <https://www.machinelearningplus.com/time-series/time-series-analysis-python/>

[7] The Rust Project Developers, The Rustdoc Book. [Online]. Available: <https://doc.rust-lang.org/rustdoc/>

[8] Machine Learning Plus, “ARIMA Model — Time Series Forecasting in Python.” [Online]. Available: <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>. Accessed: Sep, 2025.