

Final project: Prediction of stock price direction

6520422002 อภิชัย พุแก้ว

6520422022 นริศ จันทร์ไพบุลย์กิจ

6520422025 ภควัต รักษาสิล

เป็นการสร้างโมเดลทำนายราคาหุ้นที่คาดว่าจะซื้อหรือขาย (ราคาปิดของ Period ข้างหน้ามากกว่า Period ปัจจุบัน โดย Period ปัจจุบันจะถูกบันทึกว่าต้องซื้อ(1)) ในทุกๆ 15 นาที กับหุ้น 3 ตัว ได้แก่ ERW, TISCO, SPRC โดยดูจาก feature ข้อมูลการเปลี่ยนแปลงราคาในรอบ 15 นาที ที่สูงที่สุดที่โมเดลทำนายได้ของแต่ละหุ้น โดยข้อมูลที่ใช้ในการ train และ test โมเดลเป็นข้อมูลราคาหุ้น ในอดีตย้อนหลัง ประมาณ 60 วันโดยประมาณ ตั้งแต่ช่วงวันที่ทำการรันข้อมูลห้กลบย้อนกลับไป 60 วัน

ขั้นตอนการทำโมเดล prediction

1. Import all necessary packages

เป็นการดึง library ที่ต้องใช้เข้ามาก่อนได้แก่ pandas_datareader, pandas, yfinance, numpy, seaborn, sklearn, lightgbm และอื่น ๆ เป็นต้น

2. Get data

- ดึงข้อมูลราคาหุ้น ERW.BK, TISCO.BK และ SPRC.BK
- ดึงข้อมูลสินค้าโภคภัณฑ์ ได้แก่ ทองคำ (Gold), น้ำมัน (Crude Oil), SET Index, Dow Jones, S&P 500

โดยข้อมูลแต่ละค่าจะมีค่า Volume, Open, High, Low, Close และ Adj_Close

3. Feature engineering

3.1. Feature Generating

โดยจะเป็นการสร้าง Indicator จากข้อมูล ปริมาณและราคาหุ้นที่มีโดยการอาศัย Library ta และมีการทำ Feature อื่นๆเพิ่มเติมได้แก่ Signal features ที่บอกว่าราคาตอนนั้นควรซื้อหรือขายและในบาง Feature จะคำนึงถึงความเป็น Time Series ของ Feature นั้นๆด้วยโดยจะมี window หลายๆแบบหรือมองย้อนไป k period ซึ่งจะทำให้บาง observations เป็นค่า null ในข้อมูลช่วงวันแรกๆที่ดึงออกมาทางเราจึงจะตัดสินใจลบแถวที่มีค่า Null ออกเพื่อไม่ให้มีปัญหาในขั้นตอนของการเรียนรู้ของโมเดล โดย มีจำนวน Feature ที่สร้างขึ้นมา 185 ตัวซึ่งมีทั้ง Numerical Features และ Categorical Features ตัวอย่างเช่น Buy or Sell signal by technical indicator eg. Stoch_Sell, Stoch_Buy, RSI_Buy, RSI_Sell และมี Feature ที่เกี่ยวกับเวลาอย่าง วันของสัปดาห์ และชั่วโมงของวัน ณ observation นั้นๆ

Feature	Description
SMA_{window}	Simple Moving Average สำหรับช่วงเวลา x, เป็นค่าเฉลี่ยของราคาปิดในช่วงเวลานั้นๆ
EMA_{window}	Exponential Moving Average สำหรับช่วงเวลา x, คล้ายกับ SMA แต่ให้น้ำหนักมากขึ้นกับข้อมูลล่าสุด
Stoch_%K_{window}	Stochastic Oscillator %K สำหรับช่วงเวลา x, ชีววัดโมเมนตัมของราคา
Stoch_%D_{window}	Stochastic Oscillator %D สำหรับช่วงเวลา x, เป็นค่าเฉลี่ยเคลื่อนที่ของ %K
Stoch_Buy_{window}	สัญญาณซื้อจาก Stochastic Oscillator สำหรับช่วงเวลา x
Stoch_Sell_{window}	สัญญาณขายจาก Stochastic Oscillator สำหรับช่วงเวลา x
RSI_{window}	Relative Strength Index สำหรับช่วงเวลา x, วัดโมเมนตัมของการเปลี่ยนแปลงราคา
RSI_Buy_{window}	สัญญาณซื้อจาก RSI สำหรับช่วงเวลา x ($RSI < 30$)
RSI_Sell_{window}	สัญญาณขายจาก RSI สำหรับช่วงเวลา x ($RSI > 70$)
WPR_{window}	Williams %R สำหรับช่วงเวลา x, วัดระดับราคาปัจจุบันเทียบกับช่วงราคาในอดีต
ATR_{window}	Average True Range สำหรับช่วงเวลา x, วัดความผันผวนของราคา
CCI_{window}	Commodity Channel Index สำหรับช่วงเวลา x, วัดการเปลี่ยนแปลงของราคาเทียบกับค่าเฉลี่ย
Bollinger_mavg_{window}	ค่าเฉลี่ยเคลื่อนที่ของ Bollinger Bands สำหรับช่วงเวลา x
Bollinger_hband_{window}	แถบบนของ Bollinger Bands สำหรับช่วงเวลา x
Bollinger_lband_{window}	แถบล่างของ Bollinger Bands สำหรับช่วงเวลา x
VWAP_{window}	Volume-Weighted Average Price สำหรับช่วงเวลา x

ตาราง (ต่อ)

Feature	Description
VWAP_Buy_{window}	สัญญาณซื้อจาก VWAP สำหรับช่วงเวลา x
VWAP_Sell_{window}	สัญญาณขายจาก VWAP สำหรับช่วงเวลา x
CMF_{window}	Chaikin Money Flow สำหรับช่วงเวลา x, วัดการไหลเข้าออกของเงินลงทุน
CMF_Buy_{window}	สัญญาณซื้อจาก CMF สำหรับช่วงเวลา x
CMF_Sell_{window}	สัญญาณขายจาก CMF สำหรับช่วงเวลา x
Parabolic_SAR_{window}	Parabolic SAR สำหรับช่วงเวลา x, ใช้เพื่อหาจุดกลับตัวของแนวโน้ม
Fibonacci_Retracement_23.6_{window}	Fibonacci Retracement 23.6% สำหรับช่วงเวลา x
Fibonacci_Retracement_38.2_{window}	Fibonacci Retracement 38.2% สำหรับช่วงเวลา x
ROC	Price Rate of Change, วัดการเปลี่ยนแปลงของราคาเทียบกับช่วงเวลาก่อนหน้า
ROC_Buy	สัญญาณซื้อจาก ROC ($ROC > 0$ และก่อนหน้านั้น < 0)
ROC_Sell	สัญญาณขายจาก ROC ($ROC < 0$ และก่อนหน้านั้น > 0)
MACD	Moving Average Convergence Divergence, วัดแนวโน้มและโมเมนตัม
MACD_signal	สัญญาณของ MACD
MACD_diff	ความแตกต่างระหว่าง MACD กับสัญญาณของมัน
MACD_Buy	สัญญาณซื้อจาก MACD ($MACD > MACD_signal$)
MACD_Sell	สัญญาณขายจาก MACD ($MACD < MACD_signal$)
Momentum	โมเมนตัมของราคา, วัดด้วยการเปลี่ยนแปลงของราคา
OBV	On-Balance Volume, วัดปริมาณการซื้อขายสะสม

ตาราง (ต่อ)

Feature	Description
Close_pct_change	เปอร์เซ็นต์การเปลี่ยนแปลงของราคาปิด
hour_x_cate	One-hot encoding ของชั่วโมงที่ x ของวัน
day_x_cate	One-hot encoding ของวันที่ x ของสัปดาห์

หมายเหตุ: {window} หมายถึงระยะเวลาที่กำหนดสำหรับการคำนวณค่าเฉลี่ยเคลื่อนที่ โดยกำหนดให้มีค่า 5,10,15,20,25

4. Feature selection

ได้ทำ Feature selection ออกมา 5 วิธีเพื่อนำไปทำการทดลองกับทุกๆโมเดลที่นำมาฝึกฝนได้แก่

4.1. Random Forest Feature Importance, Multi-Collinearity, Cramer's V

ขั้นตอนคือนำ Feature ทั้งหมดไปหา Feature Importance ออกมาแล้วคัดเลือกเฉพาะ Feature ที่มีคะแนนอยู่ที่ quantile ที่ 0.9 ขึ้นไปแล้วไล่พิจารณาแยกระหว่าง Numerical Feature กับ Categorical Feature โดย Numerical Feature จะดูว่าระหว่าง Feature ที่คัดมาจาก Feature Importance คู่ใดๆ (multi comparison) มี Correlation กันสูงเกินค่า Threshold พิจารณาที่ 0.5 ก็จะเลือกตัดออก 1 ตัวที่มีค่า Feature Importance Score ที่ต่ำกว่าออกและคงอีกตัว ทำจนครบทุกๆ ตัว ในด้าน Categorical Features ก็จะทำเช่นเดียวกันแต่จะพิจารณา Cramer's V แทนโดยที่พิจารณาค่า Threshold ที่ 0.15

4.2. Recursive Feature Elimination (RFE)

โดยใช้ Library sklearn.feature_selection RFE ในการเลือก Feature โดยตั้งจำนวน Feature ที่ต้องการที่ 20

4.3. Principal Component Analysis (PCA)

โดยนำ Feature ที่ผ่านการทำ Scaling แล้วยก (ยกเว้น Categorical Data) ที่มีทั้งหมดมาลด มิติข้อมูลลงเพื่อนำไปใช้กับโมเดลที่เป็น Linear ได้อย่างมีประสิทธิภาพและป้องกันการเกิด Overfitting ได้ดีขึ้น โดยทดลองใช้จำนวน component ที่ 5, 15, 20

4.4. Domain Knowledge Selection

เลือก Feature โดยเข้ามาใช้โดยพิจารณาลำดับของ Feature Importance ประกอบกับการหาข้อมูลเพิ่มเติมว่านักลงทุนนิยมใช้อะไรในการตัดสินใจซื้อหรือขายในช่วงเวลานั้นๆ สำหรับการซื้อขายหุ้นส่วนใหญ่ดังตารางด้านล่าง

Feature	Description
Stoch_%K_5	Stochastic Oscillator %K ที่มีระยะเวลา 5 ข้อมูลย้อนหลัง คือตัววัดเทรนด์ของราคาที่เปลี่ยนแปลงในช่วงระยะย้อนหลัง 5 ข้อมูล
Stoch_Buy_5	สัญญาณหรือเงื่อนไขที่บ่งชี้ถึงโอกาสในการซื้อที่เป็นไปได้ ขึ้นอยู่กับ Stochastic Oscillator %K ที่มีระยะย้อนหลัง 5 ข้อมูล
Stoch_Sell_5	สัญญาณหรือเงื่อนไขที่บ่งชี้ถึงโอกาสในการขายที่เป็นไปได้ ขึ้นอยู่กับ Stochastic Oscillator %K ที่มีระยะย้อนหลัง 5 ข้อมูล
RSI_5	ดัชนี Relative Strength Index (RSI) ที่มีระยะเวลา 5 ข้อมูลย้อนหลัง ที่วัดความเร็วและการเปลี่ยนแปลงของราคา
RSI_Buy_5	สัญญาณหรือเงื่อนไขที่บ่งชี้ถึงโอกาสในการซื้อที่เป็นไปได้ ขึ้นอยู่กับดัชนี Relative Strength Index (RSI) ที่มีระยะย้อนหลัง 5 ข้อมูล
RSI_Sell_5	สัญญาณหรือเงื่อนไขที่บ่งชี้ถึงโอกาสในการขายที่เป็นไปได้ ขึ้นอยู่กับดัชนี Relative Strength Index (RSI) ที่มีระยะย้อนหลัง 5 ข้อมูล
SET_Close	ราคาปิดของดัชนี SET (ตลาดหลักทรัพย์แห่งประเทศไทย) หรือหลักทรัพย์ที่เกี่ยวข้อง
Close_pct_change	การเปลี่ยนแปลงเปอร์เซ็นต์ของราคาปิดเมื่อเปรียบเทียบกับวันก่อนหน้า
Oil_Close	ราคาปิดของน้ำมัน
Gold_Close	ราคาปิดของทอง
MACD_Buy	สัญญาณหรือเงื่อนไขที่บ่งชี้ถึงโอกาสในการซื้อที่เป็นไปได้ ขึ้นอยู่กับดัชนี Moving Average Convergence Divergence (MACD)
MACD_Sell	สัญญาณหรือเงื่อนไขที่บ่งชี้ถึงโอกาสในการขายที่เป็นไปได้ ขึ้นอยู่กับดัชนี Moving Average Convergence Divergence (MACD)
SET50_Close	ราคาปิดของดัชนี SET50 ซึ่งแทนบริษัทที่ได้รับการจดทะเบียน 50 อันดับแรกใน SET
VWAP_25	ราคาเฉลี่ยที่น้ำหนักด้วยปริมาณการซื้อขาย (Volume Weighted Average Price) ระยะย้อนหลัง 25 ข้อมูล
hour_(number of hour)_cate	ตัวแปรแบบหมวดหมู่ที่แทนช่วงเวลาต่าง ๆ ของวันการซื้อขาย
day_(number of day)_cate	ตัวแปรแบบหมวดหมู่ที่แทนวันต่าง ๆ ของสัปดาห์การซื้อขาย
close_col	ราคาปิดหรือค่าปิดของหลักทรัพย์ใด ๆ

อ้างอิงจาก : <https://knowledge.bualuang.co.th/knowledge-base/howtouseindicators/>

การวิเคราะห์กราฟหุ้น ด้วย Indicators

Indicators หรือ เครื่องมือวัดความแกว่งตัว (Oscillators) คือ เครื่องมือที่นำข้อมูลตัวเลขของทั้งในปัจจุบันและอดีตของหุ้นมาคำนวณผ่านหลักการทางคณิตศาสตร์ เพื่อทำให้เกิดเป็นข้อมูลที่จะเป็นตัวช่วยให้นักลงทุนสามารถวิเคราะห์ ทั้งแนวโน้มและความผันผวนของราคาหุ้นในอนาคตได้ โดย Indicator หลัก ๆ

MACD (Moving Average Convergence Divergence) เป็นเครื่องมือค่าเฉลี่ยเคลื่อนที่ ที่ใช้สังเกตแนวโน้มของราคาหุ้นว่าเป็นแนวโน้มขาขึ้น หรือขาลงและยังสามารถใช้ในการบอกสัญญาณซื้อขาย.

RSI (Relative Strength Index) เป็นเครื่องมือที่นำมาใช้วัดแนวโน้ม หรือ Momentum ของราคาหุ้น รวมถึงความเร็วในการเกิดแนวโน้มของราคาสำหรับนักลงทุนในช่วงหนึ่ง เพื่อใช้ประกอบการดูแนวโน้มของราคา โดยจะมีค่าระหว่าง 0-100 เป็นตัววัดความแกว่งของราคาหุ้น เพื่อดูสถานะซื้อมากเกินไปหรือขายมากเกินไป

EMA (Exponential Moving Average) สามารถตอบสนองต่อการเปลี่ยนแปลงของราคาไวกว่าการคำนวณแบบอื่น ๆ เนื่องจากมองความสัมพันธ์ของราคาหุ้นย้อนหลังแบบถ่วงน้ำหนักในรูปแบบของเลขชี้กำลัง โดยให้ความสำคัญกับราคาสุดท้ายมากที่สุด

อ้างอิงจาก : <https://knowledge.bualuang.co.th/knowledge-base/stochastic-oscillator/>

STO (Stochastic Oscillator) เป็นที่รู้จักกันอย่างกว้างขวาง และนิยมอย่างมาก เพราะเป็นเครื่องมือที่ช่วยตัดสินใจในการลงทุน สำหรับนักลงทุนสายเทคนิคที่เน้นดูกราฟโดยจะให้สัญญาณซื้อขายที่รวดเร็วและเชื่อถือได้ในระดับหนึ่ง แต่ Stochastic จะเหมาะกับการวิเคราะห์ในตลาดที่เป็นกรอบไซด์เวย์ เพื่อดูแนวโน้มซื้อ-ขาย และเพื่อการเก็งกำไรในระยะสั้น

อ้างอิงจาก : รัตนภรณ์ แซ่หลี่ และ สุมาลย์ ปานคำ. (2564). ปัจจัยที่ส่งผลต่อราคาหลักทรัพย์ใน

กลุ่มธนาคารพาณิชย์ในตลาดหลักทรัพย์แห่งประเทศไทย. วิทยาลัยนวัตกรรมการศึกษาและเทคโนโลยี มหาวิทยาลัยรังสิต.

ผลการศึกษาพบว่าปัจจัยที่ส่งผลต่อราคาหลักทรัพย์ในกลุ่มธนาคารพาณิชย์ในตลาดหลักทรัพย์แห่งประเทศไทย คือ ดัชนีอุตสาหกรรมดาวโจนส์ (DJIA) อัตราแลกเปลี่ยนดอลลาร์สหรัฐต่อบาทไทย (USD/THB) ดัชนีราคาหุ้นตลาดหลักทรัพย์แห่งประเทศไทย (SET Index) ดัชนีราคาผู้บริโภค (CPI) และอัตราเงินเฟ้อ (INF)

อ้างอิงจาก : อภิรดี พรหมสาเพชร. (2564). ปัจจัยที่ส่งผลต่อดัชนีราคาหุ้นกลุ่มปิโตรเคมีและ

เคมีภัณฑ์ในตลาดหลักทรัพย์แห่งประเทศไทย. มหาวิทยาลัยกรุงเทพ

ผลการวิจัยพบว่า ปัจจัยที่ส่งผลกระทบต่อทิศทางเดียวกันอย่างมีนัยสำคัญทางสถิติกับดัชนีราคาหุ้นกลุ่มปิโตรเคมีและเคมีภัณฑ์ในตลาดหลักทรัพย์แห่งประเทศไทย ได้แก่ ราคาน้ำมันดิบดูไบ ปริมาณเงิน M2 และดัชนีความเชื่อมั่นทางธุรกิจ ส่วนปัจจัยที่ส่งผลกระทบต่อทิศทางตรงข้ามอย่างมีนัยสำคัญทางสถิติกับดัชนีราคาหุ้นกลุ่มปิโตรเคมีและเคมีภัณฑ์ในตลาดหลักทรัพย์แห่งประเทศไทย ได้แก่ ดัชนีราคาผู้บริโภค

4.5 Stepwise Selection

โดยการนำ Domain Knowledge ที่คัดเลือกไว้ก่อนหน้านี้มาลองทำ Stepwise Selection กับโมเดลต่างๆดูว่าเมื่อทำการคัดเลือก Feature เข้าและออกทำให้โมเดลทำนายได้ดีขึ้นหรือไม่โดยทดลองทำกับโมเดลดังต่อไปนี้ LGBM, Gradient Boosting, Random Forest, Logistic Regression(L1, L2, Elasticnet), GaussianNB, SVM โดยใช้ Parameter ดังเดิมเพื่อประหยัดทรัพยากร

4. Transformation

Categorical Data

- Signal
เพื่อดูสำหรับขายหรือซื้อ ซึ่งได้แก่ RSI, MACD, STO และ CMF
- Time
วันที่เกิดการซื้อขาย, เวลาชั่วโมงของวันของข้อมูลหุ้นนั้นๆ
- โดยใช้ One hot encoding เพื่อให้ข้อมูล Categorical สามารถนำไปฝึกฝนในโมเดลได้

Numerical Data

Robust Scaler

เพื่อให้ข้อมูลมีความทนทานต่อค่าผิดปกติ (outliers) **จึงใช้ Robust Scaling** เพื่อให้ข้อมูลมีการกระจายที่น่าเสถียรแม้ว่าจะมีค่าผิดปกติ (outliers) หรือค่าที่ผิดปกติอยู่ในข้อมูลด้วย โดยมีการใช้ median และ interquartile range (IQR) เป็นหลักการในการปรับปรุง และง่ายต่อการนำไปใช้ในกระบวนการต่อไป

5. Modeling

นำข้อมูลที่ผ่านมาการทำ Transform, Feature Selection ทั้ง 5 แบบมาทำการแบ่งส่วน Train test split โดยคำนึงถึงช่วงเวลาก่อนหน้าด้วยโดยวิธีการแบ่งที่เลือกจะใช้วิธี TimeSeriesSplit แทนการใช้ TrainTestSplit ธรรมดาเพราะคำนึงถึงความเป็น TimeSeries ของเวลาหุ่นน่าจะเป็นข้อมูลที่มีน้ำหนักจากข้อมูลในอดีตใกล้เคียงมากกว่าอดีตที่ผ่านมาไกลๆ ซึ่งจะแบ่งข้อมูลไว้ ประมาณ 3 วันหรือประมาณ 50 observations และนำข้อมูลส่วนที่เหลือไปทำการฝึกฝนกับโมเดลซึ่งประกอบไปด้วย LGBM, Gradient Boosting, Random Forest, Logistic Regression(L1, L2, Elasticnet), GaussianNB, SVM

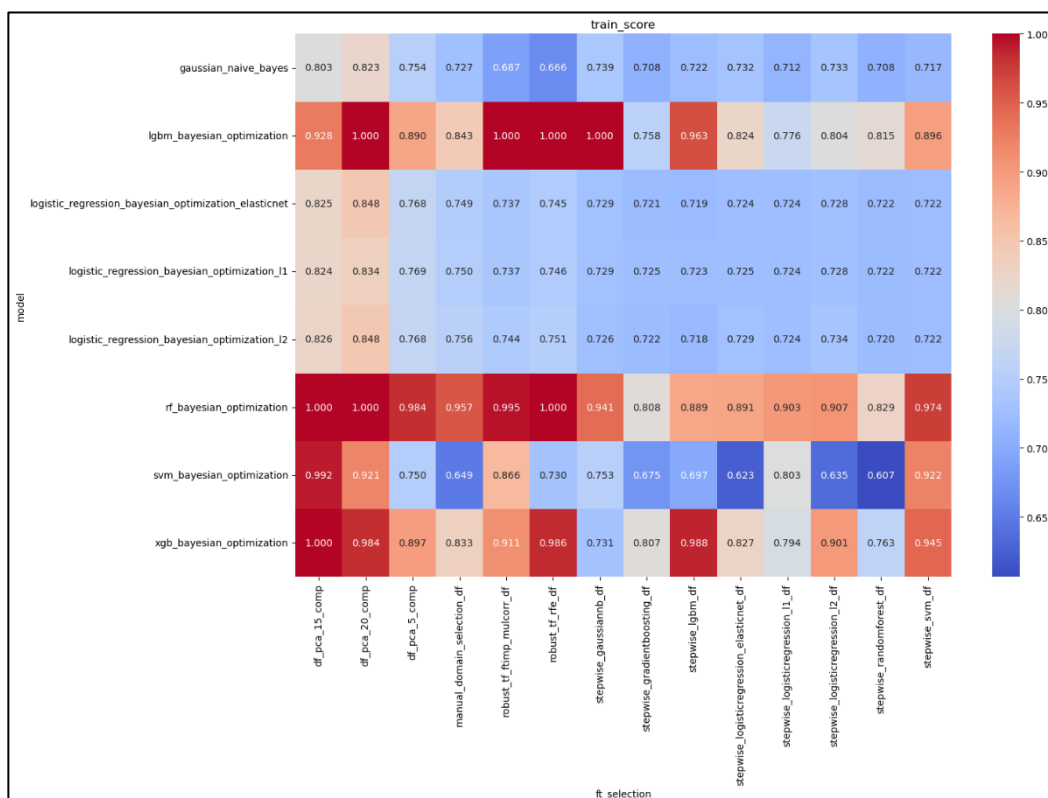
ใน Function ที่ทำการฝึกฝนกับโมเดลแต่ละอันจะทำการแบ่ง CrossValidationTimeSeriesSplit อีกครั้งเพื่อป้องกันปัญหาของการ Overfitting และหลังจากได้โมเดลในแต่ละชนิดที่มี Hyperparameter ดีที่สุด โดยหลักการทำ Hyperparameter Tuning ที่ใช้จะเลือกใช้วิธี BayesianOptimization จาก byes_opt ซึ่งใช้ความน่าจะเป็นก่อนหน้าเพื่อปรับค่าพารามิเตอร์ในรอบถัดไปแทนที่จะใช้ GSV, RGSV ซึ่งกินทรัพยากรเยอะกว่า จะได้โมเดลที่พร้อมในการ ทำนายกับ Test Data ที่แบ่งเอาไว้ในตอนแรกจะได้ โดย Function จะโชว์ Test ROC AUC ออกมาให้หลักจากทำงานเสร็จ

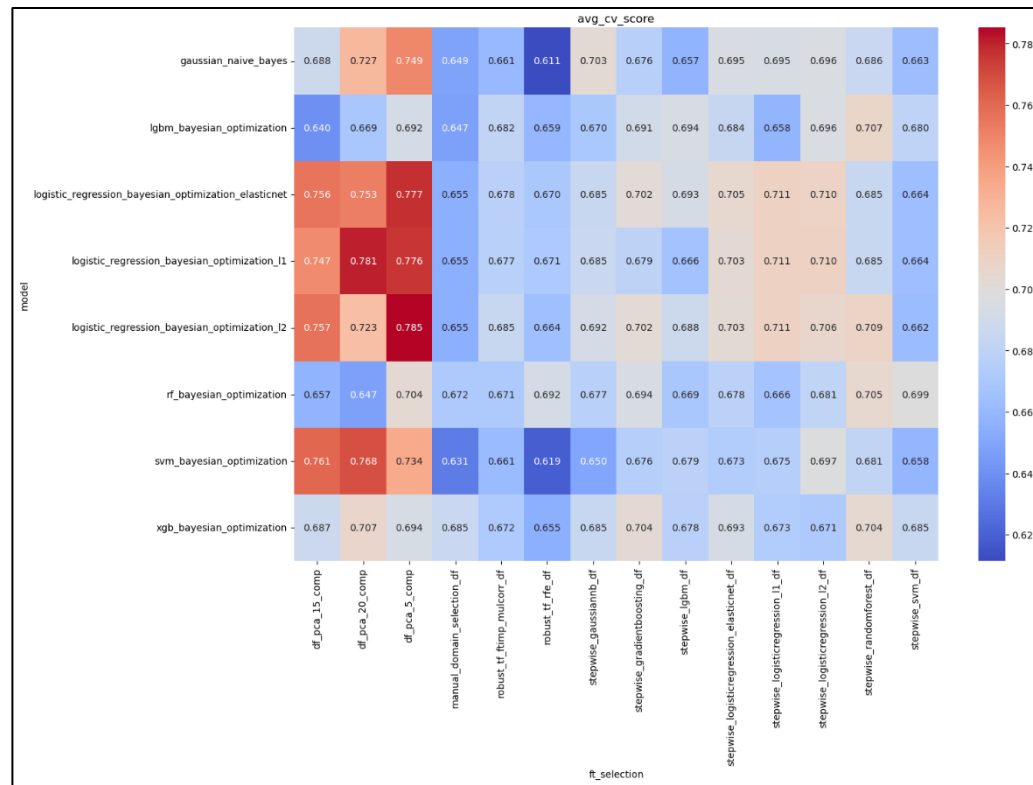
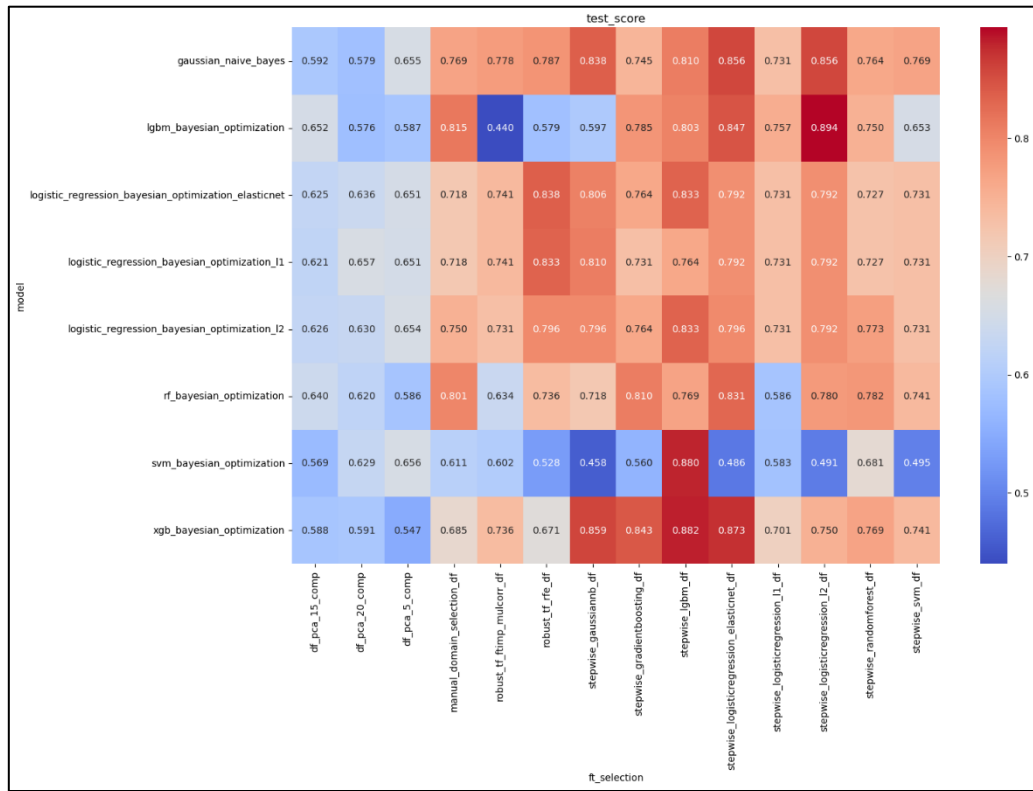
6. Evaluation

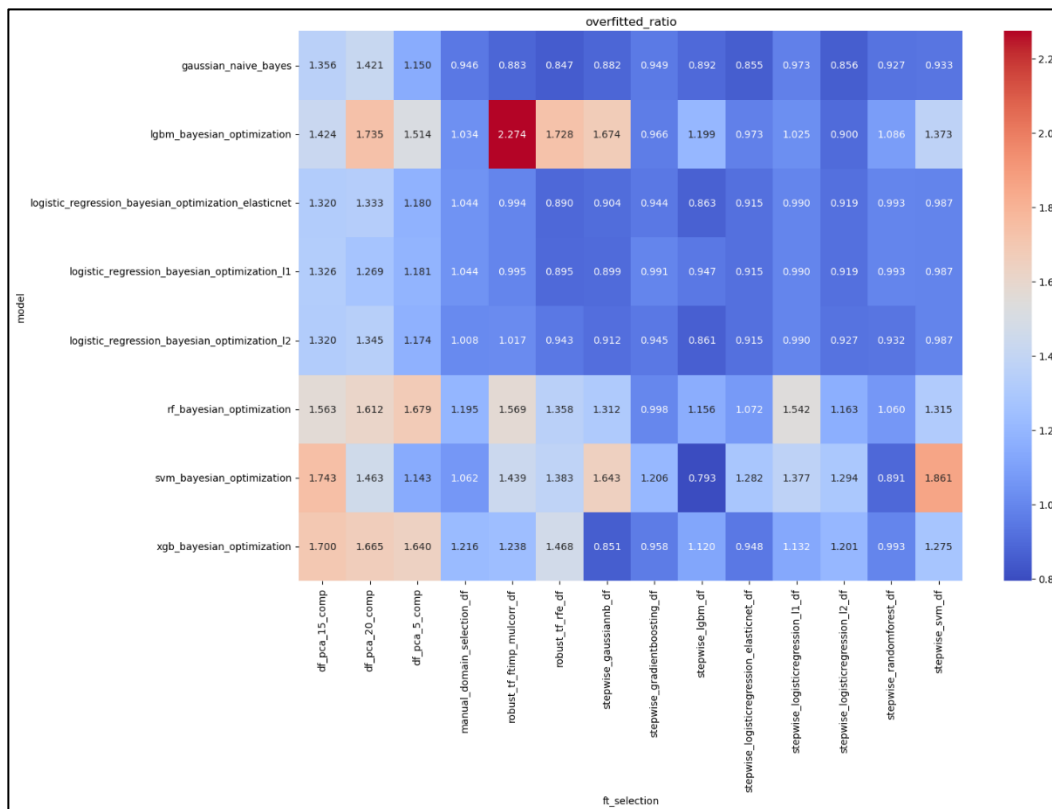
สำหรับการเลือก Model ที่เหมาะสมของหุ่นทั้ง 3 ตัว โดยวัดค่าจาก Train Score, Test Score ที่สูง และไม่ต่างกันมากเกินไป (Overfitting) โดยคำนึงถึงค่า Overfitted ratio รวมถึงค่า Avg. CV score ของ หุ่นแต่ละตัวเพื่อป้องกันการทำนายที่ Overfitting สรุปดังนี้

1. ERW

- **Model:** Gaussian naïve bayes
- **Feature:** Stepwise_Gaussian
- **Train_Score:** 0.739
- **Test_Score:** 0.838
- **Avg. CV score:** 0.703
- **erfitted_ratio:** 0.882

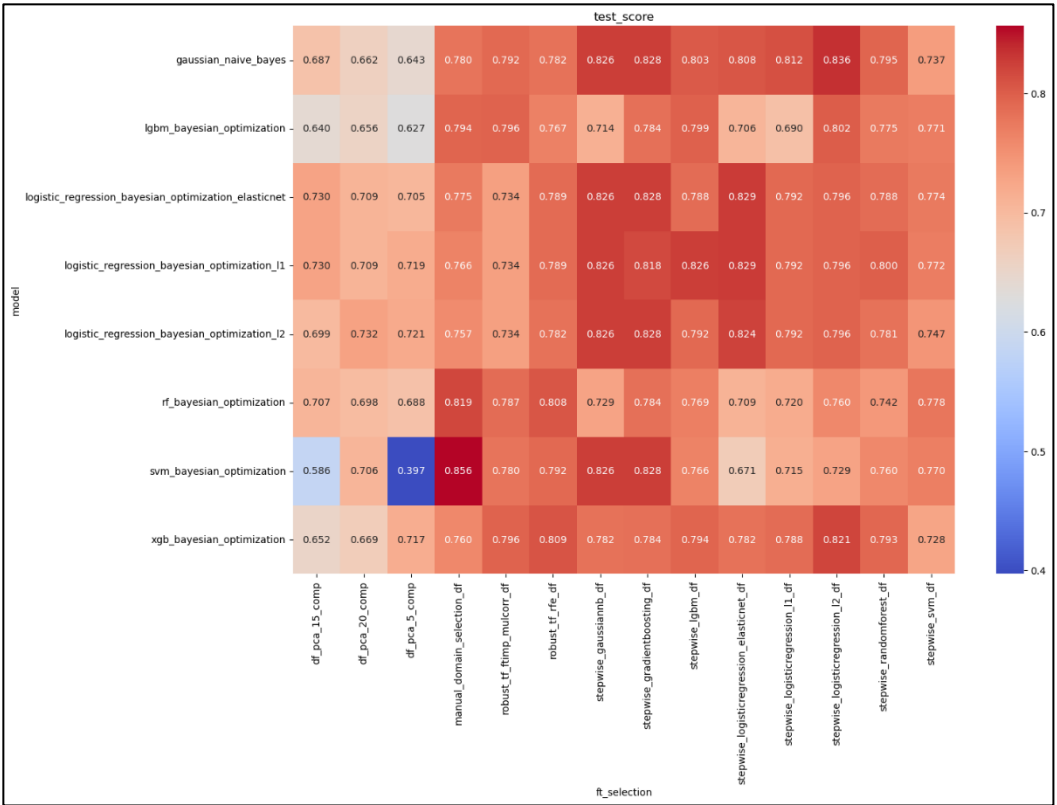
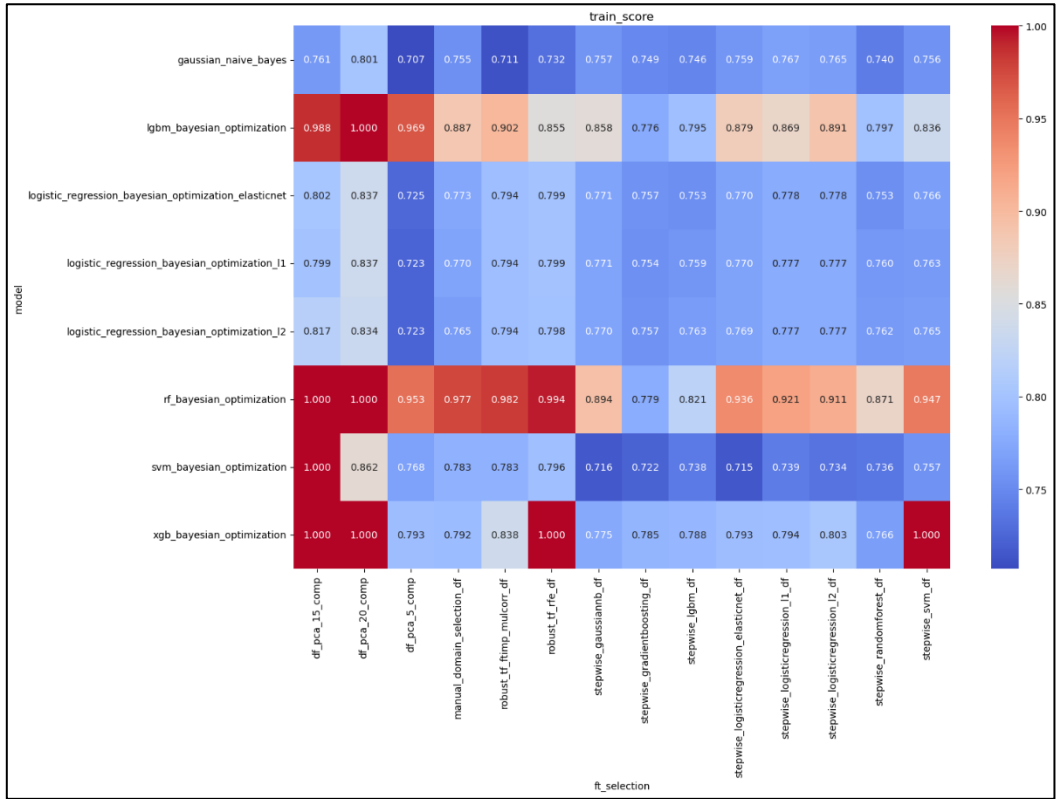


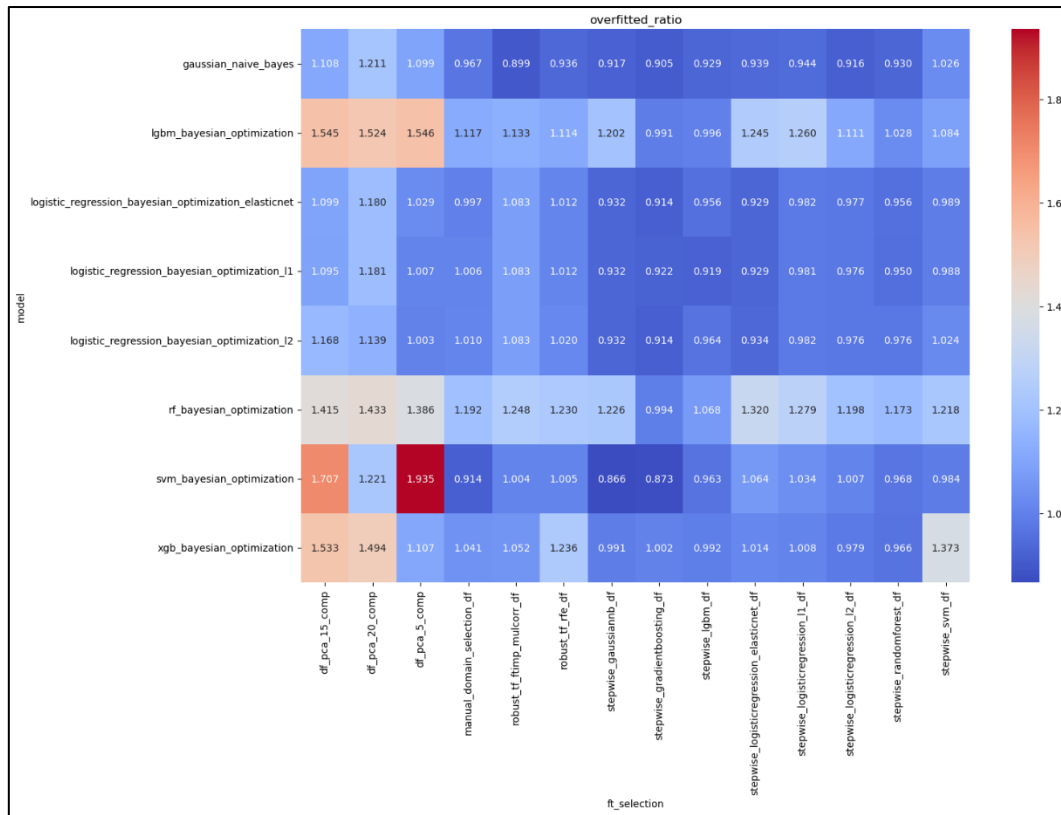
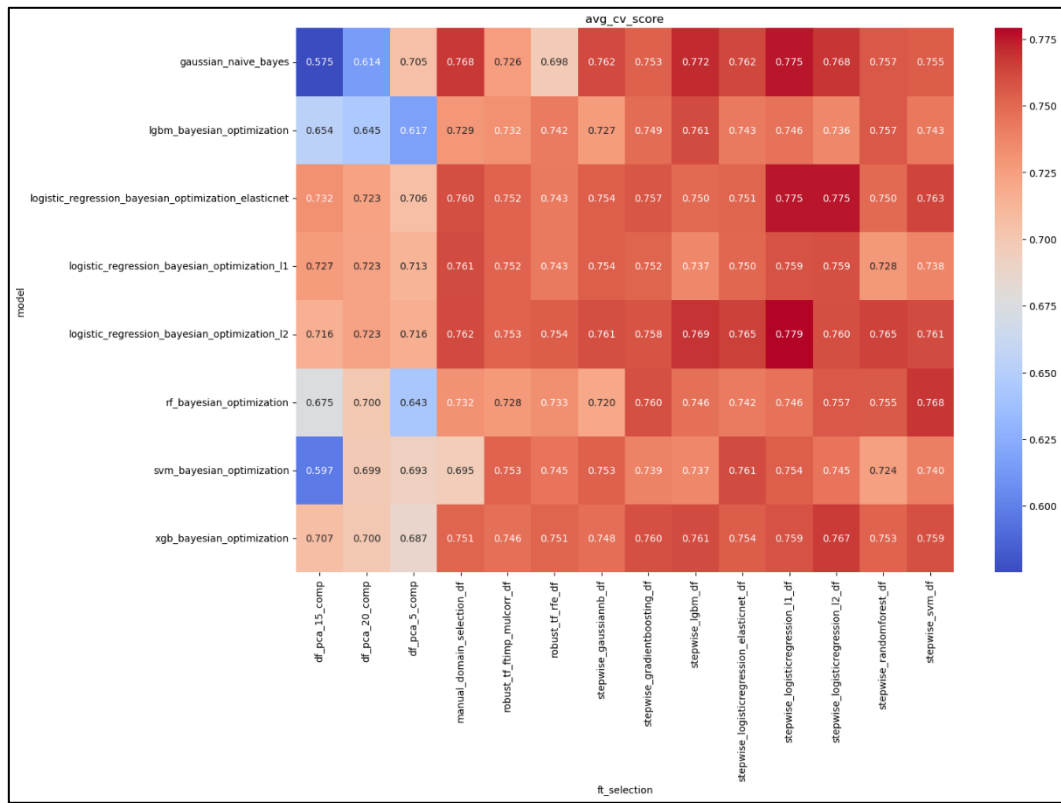




2. TISCO

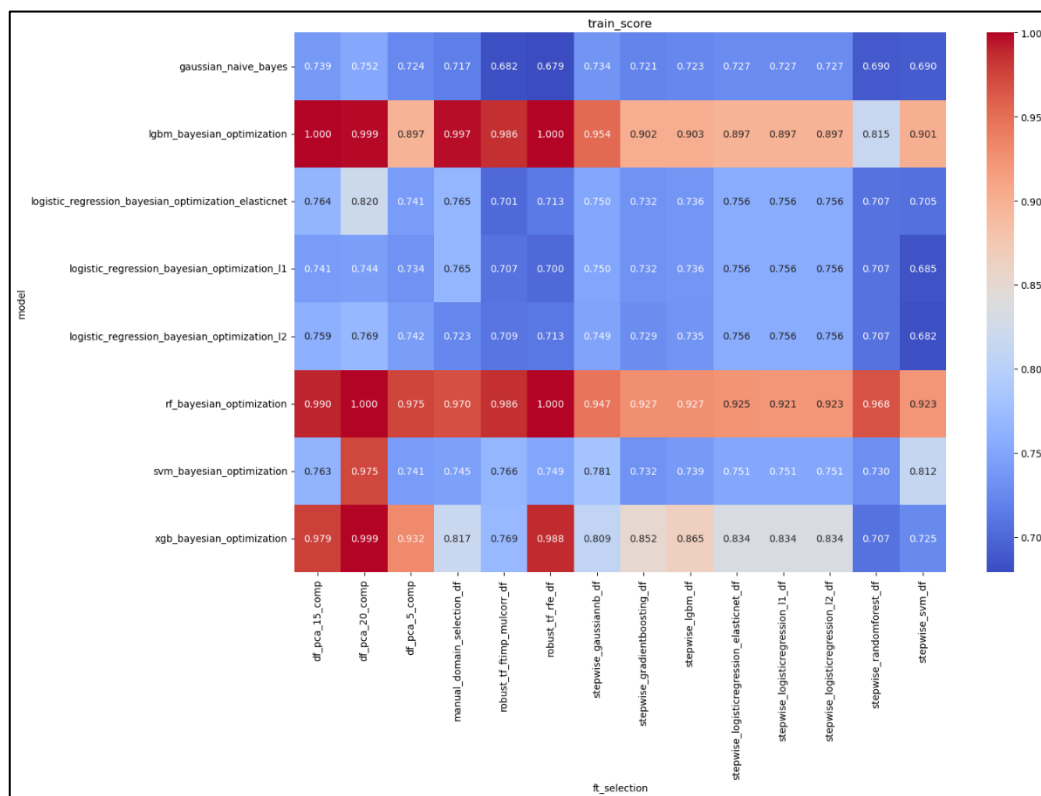
- **Model:** xgb bayesian
- **Feature:** Stepwise_gradientboosting
- **Train_Score:** 0.785
- **Test_Score:** 0.784
- **Avg. CV score:** 0.760
- **Overfitted_ratio:** 1.002

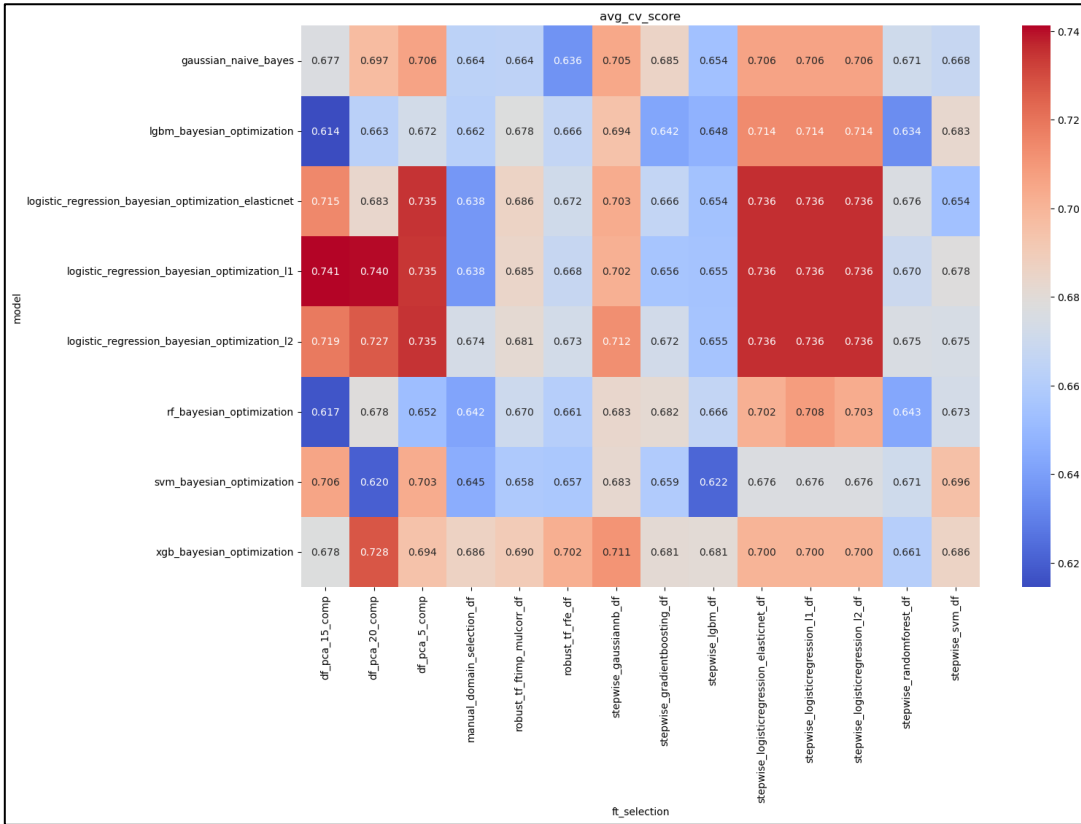
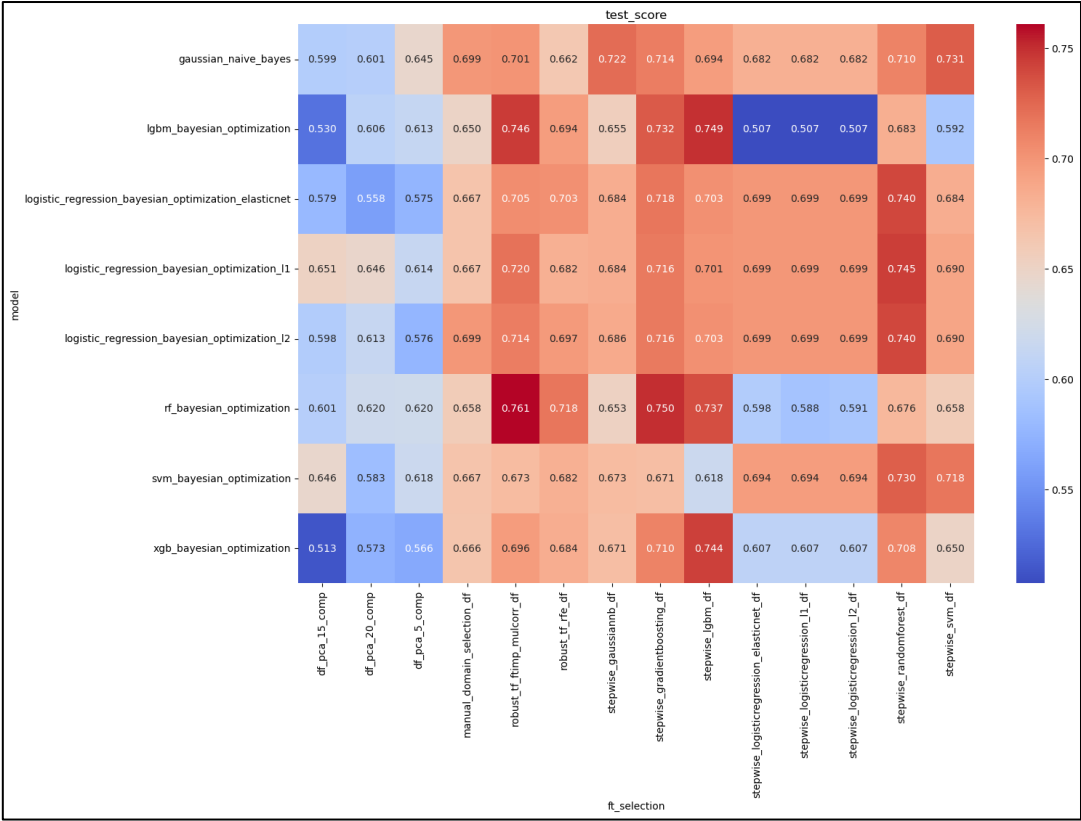


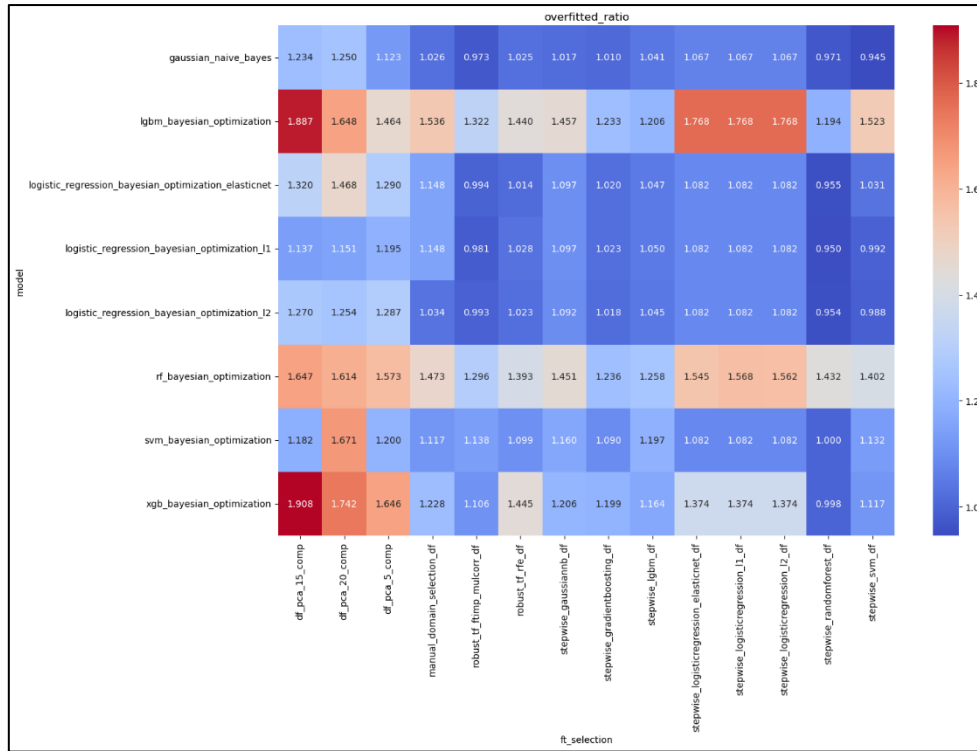


3. SPRC

- **Model:** Gaussian naïve bayes
- **Feature:** Stepwise_Gaussian
- **Train_Score:** 0.734
- **Test_Score:** 0.722
- **Avg. CV score:** 0.705
- **Overfitted_ratio:** 1.017







7. Notebook walkthrough

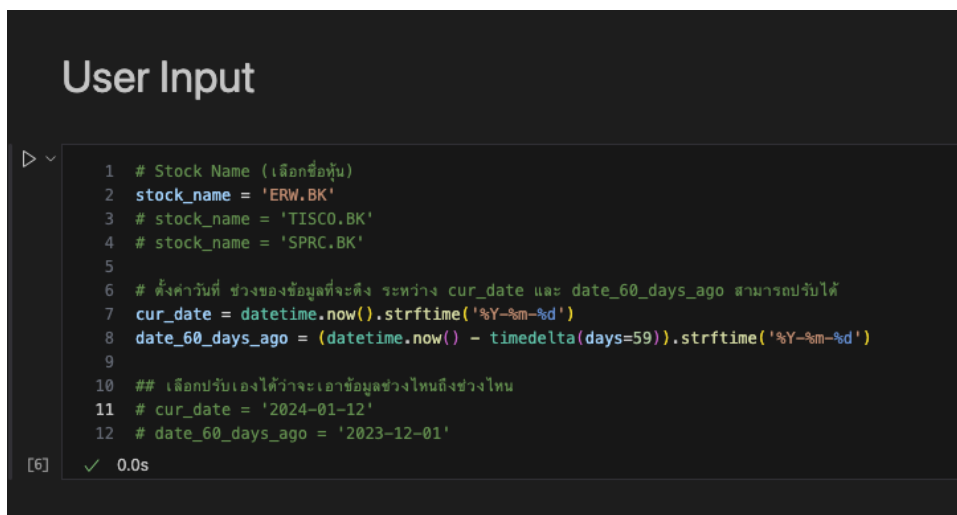
แนะนำการใช้ workbook สำหรับทำ Prediction

ไฟล์ชื่อ final_model_predict.ipynb ในไฟล์จะประกอบไปด้วย input ที่ให้ user ใส่เข้ามา

ทางทีมได้ปรึกษากันว่าจะให้ Model เรียนรู้ข้อมูลใหม่และปรับจูน Parameter เมื่อมีการเปลี่ยนแปลงช่วงเวลาของราคาหุ้นแต่ละตัวจึงไม่มีการบันทึกโมเดลไว้เป็นไฟล์แต่จะทำการลือค Feature ที่เราคิดว่าเหมาะกับ Model นั้นๆ ไว้ให้เท่านั้นเพราะอาจจะใช้เวลาค่อนข้างสูงถ้าจะต้องทำ Feature selection ใหม่

ขั้นตอนการกรอก Input มี 2 ส่วน

Input ที่ User Input (ส่วนต้นของ Notebook)



```
User Input

1 # Stock Name (เลือกชื่อหุ้น)
2 stock_name = 'ERW.BK'
3 # stock_name = 'TISCO.BK'
4 # stock_name = 'SPRC.BK'
5
6 # ตั้งค่าวันที่ ช่วงของข้อมูลที่จะดึง ระหว่าง cur_date และ date_60_days_ago สามารถปรับได้
7 cur_date = datetime.now().strftime('%Y-%m-%d')
8 date_60_days_ago = (datetime.now() - timedelta(days=59)).strftime('%Y-%m-%d')
9
10 ## เลือกปรับเองได้ว่าจะเอาข้อมูลช่วงไหนถึงช่วงไหน
11 # cur_date = '2024-01-12'
12 # date_60_days_ago = '2023-12-01'

[6] ✓ 0.0s
```

Input ที่ต้องใส่คือ

stock_name: ชื่อหุ้นที่ต้องการ predict.

cur_date: วันปัจจุบัน หรือกรอกวันสุดท้ายที่ต้องการให้อยู่ในข้อมูล

date_60_days_ago: วันที่ย้อนจากวันปัจจุบันไป 60 วัน หรือวันแรกที่ต้องการให้มีอยู่ในข้อมูล

Note

cur_date, date_60_days_ago สามารถกรอกเองได้หรือใช้ค่า default

Input ที่ Final Model (ส่วนท้ายของ Notebook)

train_split_index: ก่อนวันที่นี่จะเป็นข้อมูลที่ใช้ในการ Train

test_split_index : หลังวันที่นี่จะเป็นวันที่ใช้การ Test

Note

- วันที่เหล่านี้ควรอยู่ในช่วง ระหว่าง date_60_days_ago ถึง cur_date
- และข้อมูลวันแรกๆอาจจะถูกตัดไปบางส่วนเพราะว่าการทำ Time window feature จะมีข้อมูลที่ไม่อยู่ใน Window นั้นไม่สามารถคิดว่าจะออกมาได้เนื่องจาก period ก่อนหน้ามีจำนวนน้อยกว่า ค่า window เพราะฉะนั้น Test data ก็จำเป็นต้องอาศัยข้อมูลในอดีตมาสร้าง Feature ด้วย
- ไม่ควร Train data กับข้อมูลอนาคตเพื่อมาใช้ในการพยากรณ์ข้อมูลในอดีตเพราะว่ามี Feature ที่เป็นรูปแบบ Timeseries ด้วยควรจะพยากรณ์ไปข้างหน้าเท่านั้น

ขั้นตอนการ Run Notebook

เมื่อทำการเปลี่ยน setting ใดๆก็ตามให้กด Run-all เพื่อ Update Data โดยในแต่ละรอบจะสามารถประเมิน model ได้เพียงตัวเดียวจาก stock_name

Final Model

```
1 print('วันที่อยู่ใน data ทั้งหมด', min(df.index.date) , 'ถึง' , max(df.index.date))
2 print('""""train_split_index, test_split_index วันที่พลางกันจะพลางอยู่ในช่วง date_60_days_ago - cur_date เท่านั้น""")
3 print(f"date_60_days_ago : {date_60_days_ago} - cur_date : {cur_date} ")
```

[30] ✓ 0.0s

```
... วันที่อยู่ใน data ทั้งหมด 2023-11-17 ถึง 2024-01-12
train_split_index, test_split_index วันที่สองอันนี้จะต้องอยู่ในช่วง date_60_days_ago - cur_date เท่านั้น
date_60_days_ago :2023-11-16 - cur_date : 2024-01-14
```

```

2 | global train_split_index, test_split_index
3 |
4 | # train_split_index = (datetime.now() - timedelta(days=4)).strftime('%Y-%m-%d')
5 | # test_split_index = (datetime.now() - timedelta(days=3)).strftime('%Y-%m-%d')
6 |
7 | train_split_index = '2024-01-07'
8 | test_split_index = '2024-01-08'
9 |
10 | print('df max date : ',max(df.index.date))
11 | print('df_test min date : ',min(df_test.index.date))
12 |
13 | # Check
14 | # print(df.loc[train_split_index , :].index.date)
15 | # print()
16 | # print(set(df.loc[test_split_index: , :].index.date))
17 |
18 | print('Train Period :')
19 | print(f"{'date_60_days_ago' - {train_split_index}}")
20 |
21 | print('Test Period :')
22 | print(f"{'test_split_index' - {cur_date}}")

```

[31] ✓ 0.0s

```
... df max date : 2024-01-12
df_test min date : 2024-01-10
Train Period :
2023-11-16 - 2024-01-07
Test Period :
2024-01-08 - 2024-01-14
```

ERV

```
1 erw_feature_cols = ['Stoch_MK_5', 'RSI_Buy_5', 'MACD_Buy', 'hour_12_cat', 'y']
2 df_erw = df.loc[:,erw_feature_cols]
3
4 # ERW<br>
5 # Model - gaussian_naive_bayes <br>
6 # Feature Selection - stepwise_gaussiannb_df<br>
7 if stock_name == 'ERW.BK':
8     gaussian_naive_bayes(df_erw, 'y', 'df_erw')
```

```
8] ✓ 0.0s
```

```
Cross-Validation Scores: [0.81551363 0.71327684 0.72955975 0.78694332 0.72696078]
Average CV Score: 0.754508631144058
Train ROC AUC: 0.7446649774774774
Test ROC AUC: 0.8238095238095238
Model performance is balanced.
```

TIS®

```

1 tisco_feature_cols = ['Stoch_MK_5', 'RSI_Buy_5', 'MACD_Buy', 'hour_12_cate', 'y']
2 df_tisco = df.loc[:,tisco_feature_cols]
3
4 # TISCO <br>
5 # Model - xgb_bayesian_optimization <br>
6 # Feature Selection - stepwise_gradientboosting_df <br>
7 # ['Stoch_MK_5', 'RSI_Buy_5', 'MACD_Buy', 'hour_12_cate', 'y'] = 5 ตัวนี้ที่ ต้อง
8
9 if stock_name == 'TISCO.BK':
10     xgb_bayesian_optimization(df_tisco, 'y', 'df_tisco')

```

5] ✓ 1m 3.1s

```
df_tisco @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
Cross-Validation Scores: [0.8348155 0.71680791 0.69968553 0.75455466 0.77156863]  
Average CV Score: 0.7553996557759648  
Train ROC AUC: 0.7681306306306306  
Test ROC AUC: 0.7984126984126984  
Model performance is balanced.
```