**1. อธิบายความแตกต่างของผลการ predict ราคาหุ้นล่วงหน้า เมื่อปรับค่าระยะเวลาการคาดการข้อมูลราคาหุ้นล่วงหน้า (Next\_N-day) ที่แตกต่างกัน**

- ในเริ่มแรก Next\_N-day เท่ากับ 30 พบว่าการทำนายราคาหุ้นของโมเดลทั้ง 4 โมเดล โมเดล SVR(rbf) มีค่าความแม่นยำเฉลี่ยมาที่สุด 87.7% รองลงมาคือ Linear Regression 86.26%, SVR(lin) 83.6% และ SVR(poly) 71%

- ลองปรับ Next\_N-day เท่ากับ 15 พบว่าการทำนายราคาหุ้นของโมเดลทั้ง 4 โมเดล โมเดล SVR(rbf) มีค่าความแม่นยำเฉลี่ยมาที่สุด 92.26% รองลงมาคือ Linear Regression 92.23 %, SVR(lin) 90.89% และ SVR(poly) 78.87%

- และสุดท้ายลองปรับ Next\_N-day เท่ากับ 60 พบว่าการทำนายราคาหุ้นของโมเดลทั้ง 4 โมเดล โมเดล SVR(rbf) มีค่าความแม่นยำเฉลี่ยมาที่สุด 78.45% รองลงมาคือ Linear Regression 75.39%, SVR(lin) 72.50% และ SVR(poly) 56.76%

จากการทดลองปรับค่า Next\_N-day เป็น 15, 30 และ 60 พบว่าที่ค่า 15 ให้ความแม่นยำมากที่สุด เนื่องจากราคาหุ้นที่ทำการทำนายอาจมีการเปลี่ยนแปลงเทรนในช่วงประมาณ 15 วัน ไม่มากเท่ากับ Next\_N-day ที่ 30 และ 60 วัน

**2. แสดงพารามิเตอร์ของ Linear regression model และ Support Vector Regression (Linear kernel) ที่ได้ (zeta\_1 = slope, zeta\_0 = bias or y-intercept) พร้อมยกตัวอย่างค่า input x และ ผลลัพธ์การ predict y หรือ h\_zeta**

- Linear regression model จะได้สมการ y = 0.9329(x) + 0.0199 เมื่อลองแทนค่า x = 0.0826 จะได้ค่า y มีค่าประมาณ 0.097

- Support Vector Regression (linear) จะได้สมการ y = 0.8428 (x) + 0.071 เมื่อลองแทนค่า x = 0.0826 จะได้ค่า y มีค่าประมาณ 0.14

**3. ถ้าต้องเพิ่ม feature ใน Regression Model จะต้องปรับส่วนใดใน code บ้าง เพราะเหตุใด**

ถ้าต้องการปรับ feature ให้เพิ่มขึ้นเช่นการใช้ราคาหุ้นวันต่อไป เป็นต้น โดยจะต้องทำการปรับ code ในส่วนดังนนี้

* ส่วนการแบ่งข้อมูลชุด train validate และ test จะต้องทำการเพิ่ม column Close\_norm\_next\_1 เป็นต้น

stock\_MSFT\_2 = stock\_MSFT\_Close.iloc[:-1].assign(Close\_norm\_2 = stock\_MSFT\_Close.iloc[1:].values) เพื่อให้ค่าถัดไปเป็น feature ต่อไป

* จากนั้นต้องทำการแบ่งข้อมูล เป็นชุด train validate และ test โดย X จะต้องเป็น 2 column

train, valid = model\_selection.train\_test\_split(

stock\_MSFT,

test\_size=0.2, random\_state=42,

shuffle=True)

# ข้อมูล test ให้ใช้ข้อมูล ‘GT’ ช่วง 60 วันท้าย

test\_df = stock\_MSFT.iloc[len(stock\_MSFT) - 60:]

X\_test = np.array(test\_df.iloc[:, :2])

y\_test = test\_df.GT

# # เตรียมข้อมูล train, validate

train, valid = model\_selection.train\_test\_split(stock\_MSFT, test\_size=0.2, random\_state=42,

shuffle=True)

X\_train = train.iloc[:, :-1].values

X\_valid = valid.iloc[:, :-1].values

y\_train = train.iloc[:, -1:].values

y\_valid = valid.iloc[:, -1:].values

* จากนั้นทำตามขั้นตอนเดิม คือ cross\_val\_score เพื่อวัดค่าความแม่นยำของแต่ละ fold
* และสุดท้ายนำข้อมูลไป fit กับโมเดลต่างๆ

**4. การทำ Data Normalization ด้วยวิธีที่แตกต่างกัน จะให้ผลลัพธ์ในการทำ Regression แตกต่างกันอย่างไร**

- ในเริ่มแรกทำการ Normalizationด้วย MinMaxScaler พบว่าการทำนายราคาหุ้นของโมเดลทั้ง 4 โมเดล โมเดล SVR(rbf) มีค่าความแม่นยำเฉลี่ยมาที่สุด 87.7% รองลงมาคือ Linear Regression 86.26%, SVR(lin) 83.6% และ SVR(poly) 71% และมีค่า Mean Squared Error และ R^2 ของชุด validate และ test ดังนี้

โมเดล Linear Regression

Mean Squared Error LRM validation set: 0.005677160806793135

Mean Squared Error LRM test set: 0.0015042986909177731

R2 LRM validation set: 0.8500628098047724

R2 LRM test set: -4.091487970372274

โมเดล SVR(lin)

Mean Squared Error LRM validation set: 0.006633253428575268

Mean Squared Error LRM test set: 0.000771908373613604

R2 LRM validation set: 0.8248118355669336

R2 LRM test set: -1.6126208991683018

โมเดล SVR(rbf)

Mean Squared Error LRM validation set: 0.0049604046790096504

Mean Squared Error LRM test set: 0.00029300299986255787

R2 LRM validation set: 0.8689927650257824

R2 LRM test set: 0.00829452415926668

โมเดล SVR(poly)

Mean Squared Error LRM validation set: 0.011825285776561341

Mean Squared Error LRM test set: 0.006525102264828718

R2 LRM validation set: 0.6876871762251979

R2 LRM test set: -21.08502864983233

=======================================================================

- จากนั้น Normalizationด้วย StandardScaler พบว่าการทำนายราคาหุ้นของโมเดลทั้ง 4 โมเดล โดยสามารถรันได้เพียง linear regression เท่านั้น ซึ่งอีก 3 โมเดลที่เหลือใช้วลานานมาก โดย linear regression มีค่าเฉลี่ยความแม่นยำอยู่ที่ 86.26% และมีค่า Mean Squared Error และ R^2 ของชุด validate และ test ดังนี้

โมเดล Linear Regression

Mean Squared Error LRM validation set: 0.1454118589954793

Mean Squared Error LRM test set: 0.03853032820015887

R2 LRM validation set: 0.8500628098047722

R2 LRM test set: -4.091487970372248

โมเดล Linear Regression เมื่อลองเปรียบเทียบการ Normalizationพบว่าค่าเฉลี่ยความแม่นยำมีค่าเท่ากัน ส่วนค่า Mean Squared Error ของ MinMaxScaler มีค่าน้อยกว่า StandardScaler อยู่มาก แต่ค่า R^2 มีค่าที่เท่ากัน โดยคาดว่าอีกทั้ง 3 โมเดลน่าจะได้ค่าความแม่นยำและค่า R^2 เหมือนหรือใกล้เคียงกับการ Normalizationด้วย MinMaxScaler

**5. ถ้าเพิ่มการ Clean Data ก่อนนำไปสอน Regression Model จะให้ผลการ predict แตกต่างอย่างไร (ยกตัวอย่าง เทคนิคการ Clean data และอธิบายผลที่ได้)**

จากการตรวจสอบข้อมูลค่า Close พบว่า

มีจำนวน 8076 ข้อมูล

ค่าเฉลี่ย 59.87

ค่าการกระจาย 32.56

ค่าต่ำสุด 15.15

ค่าสูงสุด 179.94

25% 28.94

50% 54.25

75% 83.87

โดยข้อมูลที่ได้มีวันไม่ครบ จึงได้ทำการ resample ข้อมูลด้วย วัน ดังนี้

clean\_df = clean\_df.resample('D').pad()

ซึ่งจะนำข้อมูลค่าด้านบนมาใส่ ตามวันที่เพิ่มเข้ามา

จากการตรวจสอบค่าต่างๆ ของข้อมูล พบว่าในวันใกล้ๆกันจะมีค่าใกล้เคียงกัน รวมถึงค่าต่ำสุด และค่าสูงสุด จะมีการไล่ระดับตามวัน ซึ่งจะไม่เป็นค่าโดด จึงไม่ได้ทำการตัดออก

จากการ clean data ผลลัพธ์ที่ได้เป็นดังนี้

โมเดล LRM

Accuracy avg : 89.35

Mean Squared Error LRM validation set: 0.004120886130581286

Mean Squared Error LRM test set: 0.001019239760679745

R2 LRM validation set: 0.8955478981043663

R2 LRM test set: -3.1510877304493015

โมเดล svr\_lin

Accuracy avg : 88.03

Mean Squared Error LRM validation set: 0.004603248670288253

Mean Squared Error LRM test set: 0.000664855845124113

R2 LRM validation set: 0.8833214546765297

R2 LRM test set: -1.707777941641138

โมเดล svr\_rbf

Accuracy avg : 90.13

Mean Squared Error LRM validation set: 0.0037870123337675017

Mean Squared Error LRM test set: 0.0005595338189887199

R2 LRM validation set: 0.904010597324875

R2 LRM test set: -1.2788298302123655

โมเดล svr\_poly

Accuracy avg : 74.88

Mean Squared Error LRM validation set: 0.009676701843783452

Mean Squared Error LRM test set: 0.005359582812503306

R2 LRM validation set: 0.7547246356797632

R2 LRM test set: -20.828130447414914

จากผลการทดลองพบว่า โดมเดลมีความแม่นยำเพิ่มมากขึ้น แต่ในบางโมเดลค่า R^2 มีค่าเป็นลบมากขึ้น

เพิ่มเติม จากการแสดงกราฟค่า Close ด้วยวัน พบว่าเทรนของข้อมูลมีความแตกต่างกัน ระหว่างปี 1986-1999 และ ปี 2000 เป็นต้นไป อาจทำการตัดข้อมูลที่ล้าหลังออกได้ คือข้อมูลในช่วงปี 1986-1999

**6. แสดง รายละเอียดข้อมูลหุ้นที่เลือก ระยะเวลาในการดึงข้อมูลย้อนหลัง ชนิดของ best model, best parameter ที่ได้พร้อมค่า performance measurement (MSE, R-square)**

หุ้นที่ทำการเลือกคือ MSFT หรือ Microsoft โดยเป็นข้อมูลระหว่างวันที่ 13 มีนาคม 1986 ถึง 27 มีนาคม 2018 รวมเป็นจำนวน 8076 ข้อมูล ประกอบด้วย ค่า

* Open ราคาหุ้นที่เปิดวันนั้น
* High ราคาหุ้นสูงที่สุดวันนั้น
* Low ราคาหุ้นต่ำที่สุดวันนั้น
* Close ราคาหุ้นที่ปิดวันนั้น
* Volume ปริมาณการซื้อขายหุ้น
* Ex-Dividend ผู้ลงทุนที่ซื้อหุ้นจะไม่มีสิทธิรับเงินปันผล ราคาที่ซื้อขายในวันนี้
* อื่นๆ คือ Split Ratio, Adj. Open, Adj. High, Adj. Low, Adj. Close, Adj. Volume

Best model : SVR

Best parameter : C = 1 , gamma = 1.0, kernel = rbf

Performance measurement :

Mean Squared Error validation set: 0.004918668762892873

Mean Squared Error test set: 0.00030224367628290386

R2 validation set: 0.8700950353693114

R2 test set: -0.02298170649648812

**7. วิเคราะห์ผลของการปรับค่าพารามิเตอร์สำหรับ Regression Model ทั้ง 4 แบบ ที่ทำการทดลอง**

ทำการทดสอบกับค่าพารามิเตอร์ด้านล่าง โดยทดสอบกับ SVR โมเดล kernel ต่างๆ ได้ค่าดังนี้

ค่าพารามิเตอร์ :

c\_param = [1, 10, 100, 1000]

gamma = [0.01, 0.5, 1.0]

Best parameter : C = 1 , gamma = 1.0, kernel = rbf

คำนวณค่าตัววัดประสิทธิภาพของการทำนายได้ดังนี้ :

Mean Squared Error validation set: 0.004918668762892873

Mean Squared Error test set: 0.00030224367628290386

R2 validation set: 0.8700950353693114

R2 test set: -0.02298170649648812

Best parameter : C = 1000 , gamma = 0.01, kernel = linear

คำนวณค่าตัววัดประสิทธิภาพของการทำนายได้ดังนี้ :

Mean Squared Error validation set: 0.006633253428575268

Mean Squared Error test set: 0.000771908373613604

R2 validation set: 0.8248118355669336

R2 LM test set: -1.6126208991683018

Best parameter : C = 1 , gamma = 0.5, kernel = poly

คำนวณค่าตัววัดประสิทธิภาพของการทำนายได้ดังนี้ :

Mean Squared Error validation set: 0.01991124396549425

Mean Squared Error test set: 0.021381661339061813

R2 validation set: 0.4741322158946777

R2 test set: -71.368919917087